

Agentes Inteligentes y Sistemas Multiagentes

Tutorial
MICAÍ 2006

Dr. Leonardo Garrido
Centro de Sistemas Inteligentes
Tecnológico de Monterrey
E-mail: leonardo.garrido@itesm.mx
Web page: <http://www-csi.mty.itesm.mx/~lgarrido>

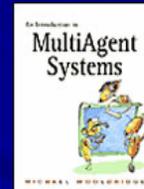
Instructor

- Instructor:
 - Dr. Leonardo Garrido
- Afiliación:
 - Cátedra de Investigación en Agentes Inteligentes y Sistemas Multiagentes
 - Centro de Sistemas Inteligentes
 - Tecnológico de Monterrey
- E-mail:
 - leonardo.garrido@itesm.mx
- Web page:
 - <http://csi.mty.itesm.mx/~lgarrido>



Panorama general

- Objetivo del tutorial:
 - Introducción a los Agentes Inteligentes y Sistemas Multiagentes
- Tutorial basado en:
 - El libro de texto “An introduction to MultiAgent Systems” de Michael Wooldridge:
 - <http://www.csc.liv.ac.uk/~mjw/pubs/imas/>
 - Las diapositivas del libro disponibles en:
 - <http://www.csc.liv.ac.uk/~mjw/pubs/imas/distrib/>
 - Y el software de programación multiagente “NetLogo”:
 - <http://ccl.northwestern.edu/netlogo/>



Contenido

- Introducción
 - Introducción a los Agentes
 - Agentes Inteligentes y Sistemas Multiagentes
- Arquitecturas de Agentes
 - Agentes deliberativos con razonamiento práctico
 - Agentes con arquitecturas reactivas e híbridas
- Sistemas Multiagentes
 - Interacción y negociación entre agentes
 - Comunicación y coordinación entre agentes
- Aplicaciones de Sistemas Multiagentes
 - Pasado, presente y futuro de los sistemas de agentes
- Ejemplos y ejercicios en NetLogo

Contenido

- **Introducción**
 - **Introducción a los Agentes**
 - **Agentes Inteligentes y Sistemas Multiagentes**
- **Arquitecturas de Agentes**
 - Agentes deliberativos con razonamiento práctico
 - Agentes con arquitecturas reactivas e híbridas
- **Sistemas Multiagentes:**
 - Interacción y negociación entre agentes
 - Comunicación y coordinación entre agentes
- **Aplicaciones de Sistemas Multiagentes**
 - Pasado, presente y futuro de los sistemas de agentes
- **Ejemplos y ejercicios en NetLogo**

¿Sistemas Inteligentes?



¿Qué es un agente?

- Pero, no un agente de espionaje!
- ... ni un agente de tránsito, tampoco un agente de seguros, ni un agente de viajes ...
- ¿Qué es un **Agente Inteligente de Software**?



**Es un proceso autónomo
con un comportamiento
reactivo, proactivo e inteligente!**

Agentes Inteligentes



Un primer ejemplo ...

- Cuando una nave espacial hace un viaje largo en el espacio, el centro de control en tierra usualmente requiere ir rastreando continuamente su progreso y decidir como solucionar eventualidades. Esto es muy costoso y si las decisiones tienen que tomarse rápidamente, entonces esto simplemente no es práctico.
- Por esta razón, organizaciones como la NASA están seriamente investigando la posibilidad de hacer naves más autónomas e inteligentes.

Nave espacial Deep Space 1

- Lanzada en Cabo Cañaveral el 24 de octubre de 1998.
- Probó en forma autónoma 12 tecnologías avanzadas de alto riesgo.
- Además, cumplió con una misión extra imprevista: encontrar el cometa Borrelly y regresar imágenes y otros datos científicos.
- La nave fue retirada de órbita en diciembre de 2001.
- Más información: <http://nmp.jpl.nasa.gov/ds1/>



Agentes para tareas especializadas

- Agentes (y sus instancias físicas en robots) tienen un rol muy importante en situaciones de alto riesgo, imposibles o no adecuados para seres humanos.
- El grado de autonomía difiere y depende de cada situación (control remoto por humanos es una alternativa, pero no siempre)

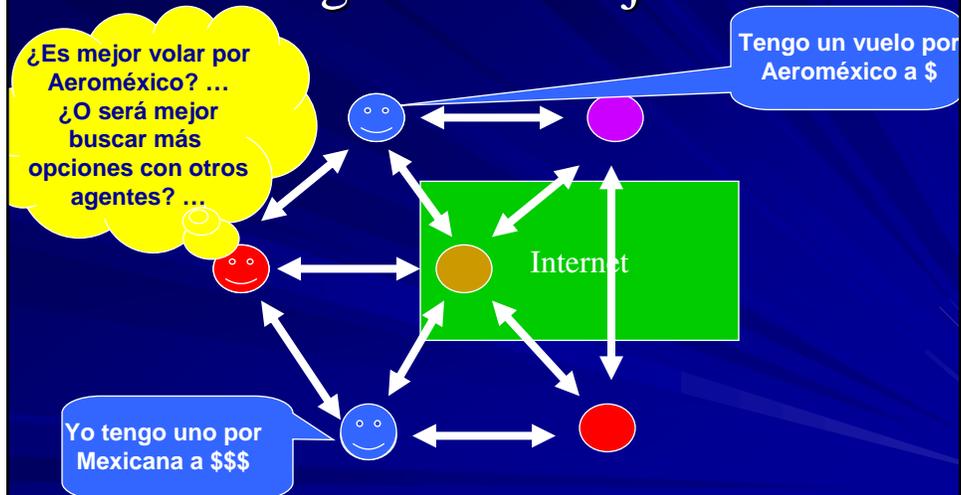
Sistemas Multiagentes



Otros ejemplos ...

- Si tenemos un conjunto de aspiradoras robots limpiando un edificio deben ser capaces de auto coordinarse en forma inteligente para recorrer y aspirar toda el área de tal forma que optimicen sus recursos.
- Si tenemos varios agentes de software “viviendo” en Internet, estos deben ser capaces de poder comunicarse entre sí, interactuar, negociar y coordinarse para lograr sus objetivos.
- Los objetivos que busquen estos agentes (ya sea en sus instancias robóticas o de software) pueden ser colectivos donde cada agente está interesado en colaborar, pero también pueden ser objetivos personales e individuales.

Agentes de viajes



Una definición de agentes

Un Agente es un sistema computacional que es capaz de realizar acciones en forma independiente para beneficio de su usuario o dueño (ideando lo que necesita ser hecho para satisfacer sus objetivos de diseño, en lugar de que constantemente se le esté diciendo que hacer paso a paso).

Autonomía en los agentes

- El principal punto acerca de los Agentes es que son autónomos: capaces de actuar independientemente, teniendo control de su propio estado interno.
- Por lo tanto: un Agente es un sistema computacional de acción autónoma en algún medio ambiente que busca satisfacer los objetivos con los que fue diseñado.



Inteligencia en los agentes

- Agentes triviales (no-interesantes):
 - Termostato
 - Demonio de Unix
- *Un Agente Inteligente es un sistema computacional capaz de realizar acciones flexibles y autónomas en algún medio ambiente*
- Por *flexible*, queremos decir:
 - *Reactivo*
 - *Proactivo*
 - *Social*

Una definición de Sistemas Multiagentes

- Un sistema multiagente es un sistema que consiste de un número de agentes, en donde ellos interactúan unos con otros.
- En el caso más general, los agentes estarán actuando para sus respectivos usuarios con objetivos y motivaciones diferentes.
- Para interactuar exitosamente, los agentes requieren la habilidad de cooperar, coordinarse y negociar unos con otros, tal como lo hacen los seres humanos.

Investigación en Sistemas Multiagentes

- Las preguntas de investigación son del tipo:
 - ¿Cómo pueden emerger conductas cooperativas en sociedades de agentes con intereses egoístas?
 - ¿Qué tipos de lenguajes pueden usar los agentes para comunicarse?
 - ¿Cómo pueden los agentes egoístas reconocer conflictos y luego cómo lograr acuerdos entre ellos?
 - ¿Cómo pueden los agentes autónomos coordinar sus actividades para lograr sus objetivos cooperativamente?

Áreas relacionadas con los Agentes Inteligentes y los Sistemas Multiagentes

- El área de la Inteligencia Artificial
- Las tecnologías de objetos
- Los sistemas distribuidos
- El cómputo paralelo
- Las sociedades naturales
- La sociología y psicología

Una definición de Inteligencia Artificial

“Podemos llamar a un programa inteligente si exhibe comportamientos que podrían pensarse inteligentes si éstos fueran exhibidos por seres humanos”

Herbert Simon.

La Inteligencia Artificial

- Es el campo de las Ciencias Computacionales concerniente con el diseño de Sistemas Inteligentes que tenga comportamientos inteligentes, tales como:
 - Razonamiento acerca de diferentes posibilidades
 - Toma decisiones autónomamente
 - Aprendizaje de nuevas situaciones y errores del pasado
 - Planeación y programación de tareas
 - Pronóstico de situaciones futuras, etc.
- Para esto la IA cuenta con diversas áreas que estudian:
 - Técnicas de Razonamiento Automático
 - Técnicas de Aprendizaje Automático
 - Algoritmos de planificación y programación de tareas
 - Algoritmos de coordinación
 - Protocolos de negociación, etc.

Las tecnologías de objetos

- Clases
- Herencia
- Encapsulamiento
- Mensajes entre objetos



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Los sistemas distribuidos

- Distribución de datos y procesos
- Conectividad, Redes, Protocolos
- Interoperabilidad
- Internet e intranets

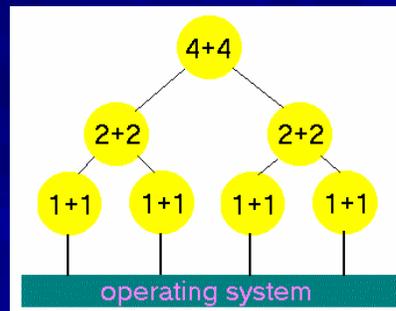


Dr. Leonardo Garrido

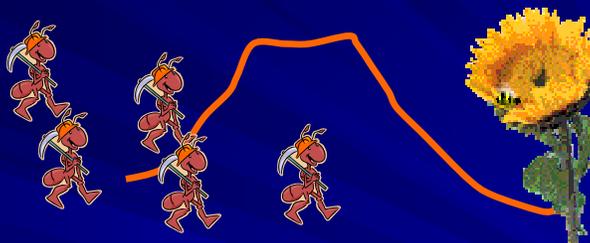
Agentes Inteligentes y Sistemas Multiagentes

El cómputo paralelo

- Multiprocesadores
- División de tareas
- El método de divide y vencerás (“divide and conquer”)
- Las tareas por lo general son homogéneas



Las sociedades naturales



- Coordinación y comunicación
- Comportamientos emergentes

La sociología y psicología

- Estudio de sociedades humanas
- Organización y coordinación
- Cooperación y competencia
- Psicología social



Agentes y Sistemas Multiagentes



**Procesos autónomos
con comportamientos
reactivos, proactivos
e inteligentes!**

Contenido

- Introducción
 - Introducción a los Agentes
 - Agentes Inteligentes y Sistemas Multiagentes
- **Arquitecturas de Agentes**
 - **Agentes deliberativos con razonamiento práctico**
 - **Agentes con arquitecturas reactivas e híbridas**
- Sistemas Multiagentes
 - Interacción y negociación entre agentes
 - Comunicación y coordinación entre agentes
- Aplicaciones de Sistemas Multiagentes
 - Pasado, presente y futuro de los sistemas de agentes
- Ejemplos y ejercicios en NetLogo

Arquitecturas de Agentes

- Existen básicamente tres tipos de arquitecturas de agentes inteligentes:
 - Las simbólicas, lógicas o deliberativas
 - Las arquitecturas reactivas
 - Y las arquitecturas híbridas

Arquitecturas

- Maes define una arquitectura de agentes como:
- Una metodología particular para la construcción de agentes. Ésta especifica como el agente puede ser descompuesto en un conjunto de módulos o componentes y sus interacciones para proveer respuestas a las preguntas de cómo los datos de los sensores y el estado interno del agente determina sus acciones y también sus futuros estados internos.
- Una arquitectura también incluye las técnicas y algoritmos que soportan esta metodología.

Arquitecturas

- Kaelbling considera una arquitectura de agentes como:
- Una colección de módulos de software y hardware diseñadas típicamente como cajas con flechas entre módulos. Una vista más abstracta de una arquitectura es como una metodología general para diseñar módulos para tareas particulares.

Arquitecturas

- Originalmente (1956-1985) muchas de las arquitecturas de agentes fueron diseñadas con técnicas de razonamiento simbólico.
- En su más pura expresión proponían que los agentes usaran razonamiento lógico explícito para decidir que hacer.
- Problemas con este enfoque llevaron al surgimiento del enfoque reactivo (a partir de 1985).
- A partir de 1990, un número de alternativas han emergido: las arquitecturas híbridas que intentan combinar lo mejor de ambos enfoques.

Arquitecturas de agentes



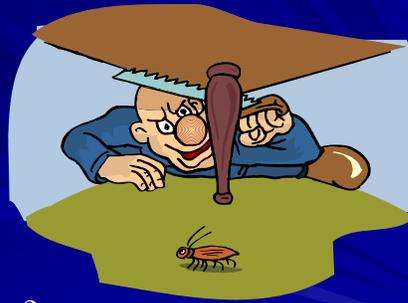
- Agentes reactivos
 - Estímulo - respuesta



- Agentes deliberativos
 - Estado interno
 - Proceso de deliberación

¿Reacción o deliberación?

- ¿Una cucaracha se pone a pensar realmente en cómo escapar?
- ¿O simplemente escapa de forma reactiva?
- ¿Es el ser humano igual de reactivo?
- ¿O es un ser más deliberativo?



Razonamiento práctico

- El razonamiento práctico es razonamiento que es dirigido hacia acciones:
 - Los procesos de decidir que hacer.
 - El proceso de considerar diferentes opciones conflictivas, donde las consideraciones relevantes son provistas por lo que el agente desea o por lo que le importa o por sus valores acerca de las creencias del propio agente
- El razonamiento práctico se distingue del razonamiento teórico en que este último es dirigido hacia las creencias del agente.

Razonamiento práctico

- El razonamiento práctico humano consiste en dos actividades:
 - *Deliberación*:
Decidir *cuál* estado queremos alcanzar
 - *Razonamiento “means-ends”*:
Decidir el *cómo* lograr alcanzar dicho estado.
- Las salidas del proceso de deliberación son las *intenciones*.

Intenciones

- Las intenciones son problemas que los agentes deben resolver.
- Las intenciones proveen un “filtro” para adoptar otras intenciones (que no deben causar conflictos con las primeras).
- Los agentes mantienen una lista de intenciones logradas y se inclinan a seguir tratando aquellas que no han logrado hasta el momento.
- Los agentes creen que sus intenciones son posibles
- No es racional para un agente adoptar intenciones que él cree que son imposibles.
- Tampoco es racional adoptar intenciones que se creen que son inevitables.
- Los agentes tampoco tienen la intención de lograr todos los efectos secundarios de sus intenciones.

Intenciones y deseos

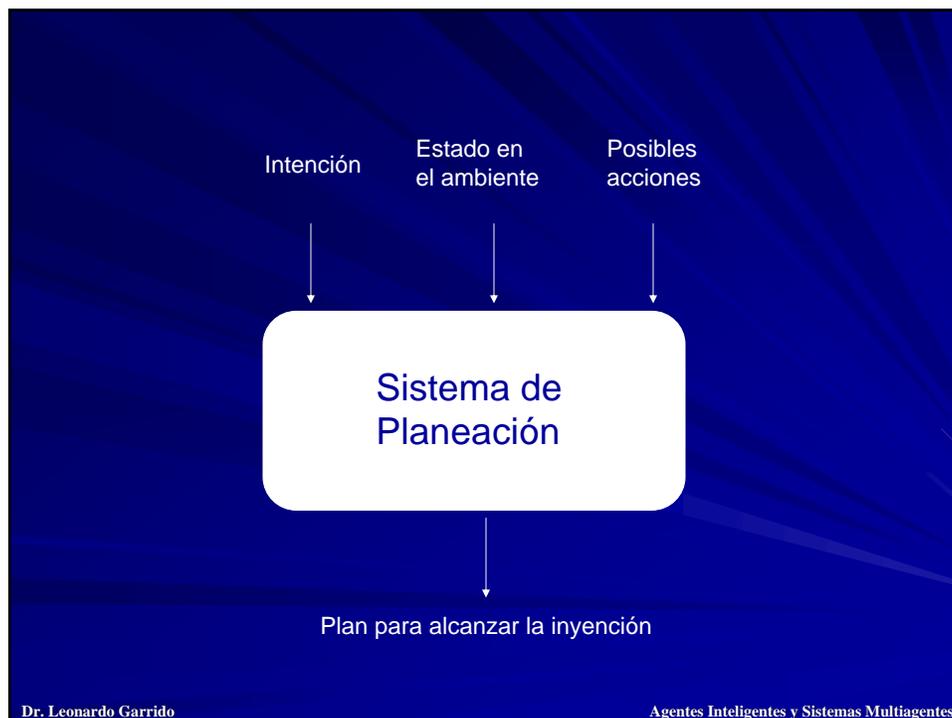
- Hay que notar que las intenciones son algo más fuerte que meros deseos.
- Mi deseo de programar un sistema esta tarde es sólo una influencia potencial de mi conducta para esta tarde. Mi conducta también puede estar influenciada por otros deseos. Entonces yo considero todos ellos, veo sus pros y sus contras, y entonces me decido por una de las opciones y entonces ya tengo una intención. Luego sólo tengo que razonar cómo alcanzar mi objetivo, ejecutar mis intenciones.

Planeación

- Desde principio de los 70's, la comunidad de IA se ha preocupado por el diseño de algoritmos de planeación.
- Planeación es esencialmente programación automática: el diseño de un curso de acción que logrará un objetivo deseado.
- Dentro de la comunidad simbólica de IA siempre ha asumido que un sistema de planeación será el componente central de cualquier agente artificial.
- El resultado es la creación de muchos algoritmos de planeación y una teoría de planeación ha sido muy bien desarrollada.

Razonamiento “Means-Ends”

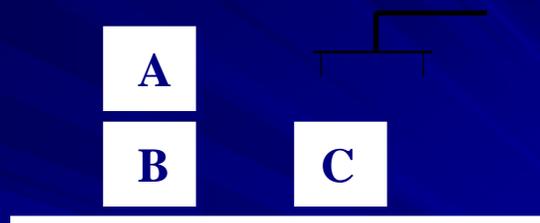
- La idea básica es darle al agente representaciones de:
 - Los objetivos o intenciones a lograr.
 - Las acciones que puede ejecutar.
 - El medio ambiente.
- Luego sólo hay que generar un *plan* para lograr las intenciones.
- Esencialmente es una forma de:
 - **Programación automática!**



Representación en planeación

- Cómo debemos representar:
 - Los objetivos a alcanzar?
 - Los estados del medio ambiente?
 - Las acciones disponibles del agente?
 - El plan por sí mismo?

El mundo de los bloques



- Este mundo de los bloques contiene un brazo robótico, 3 bloques (A, B, and C) de igual tamaño y la superficie horizontal de una mesa.

La ontología en este mundo

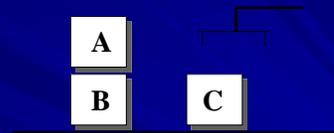
- Para representar al mundo de los bloques, necesitamos una *ontología*:

$On(x, y)$	obj x está encima del obj y
$OnTable(x)$	obj x está encima de la mesa
$Clear(x)$	nada está encima del obj x
$Holding(x)$	el brazo está agarrando el obj x

Representación del mundo

- Un ejemplo de una representación del mundo con esta ontología y su descripción gráfica:

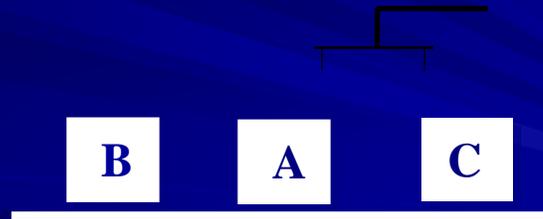
- $Clear(A)$
 $On(A, B)$
 $OnTable(B)$
 $OnTable(C)$



- Generalmente se usa la *suposición del mundo cerrado*:
 - Cualquier cosa que no se describe explícitamente en el mundo se supone que es *falso*.

Intenciones

- Una *intención u objetivo* se representa con un conjunto de fórmulas en lógica de predicados:
- Este sería entonces el objetivo:
 $OnTable(A) \wedge OnTable(B) \wedge OnTable(C)$



Acciones

- Las *acciones* basadas en el sistema STRIPS.
- Cada acción tiene:
 - Un *nombre* que tiene argumentos
 - Una *lista de pre-condiciones* o hechos que deben ser verdaderos para poder ejecutar la acción
 - Una *lista de hechos a borrar* que ya no serán verdaderos cuando la acción sea ejecutada
 - Una *lista de hechos a añadir* que se volverán verdaderos al ejecutar la acción
- Cada una de estas puede contener *variables*

Operadores



■ Ejemplo 1:

La acción *stack* ocurre cuando el brazo coloca el objeto x que está agarrando encima del objeto y

– $Stack(x, y)$

precondiciones
lista a borrar
lista a añadir

$Clear(y) \wedge Holding(x)$
 $Clear(y) \wedge Holding(x)$
 $ArmEmpty \wedge On(x, y)$

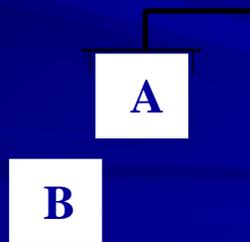
Operadores

■ Ejemplo 2:

La acción *unstack* ocurre cuando el brazo agarra un objeto x de encima de otro objeto y

– $UnStack(x, y)$

precondiciones $On(x, y) \wedge Clear(x) \wedge ArmEmpty$
lista a borrar $On(x, y) \wedge ArmEmpty$
lista a añadir $Holding(x) \wedge Clear(y)$



Operadores

■ Ejemplo 3:

La acción *pickup* ocurre cuando el brazo recoge un objeto x de encima de la mesa.

– *Pickup(x)*

precondiciones $Clear(x) \wedge OnTable(x) \wedge ArmEmpty$

lista a borrar $OnTable(x) \wedge ArmEmpty$

lista a añadir $Holding(x)$

■ Ejemplo 4:

La acción *putdown* ocurre cuando el brazo pone un objeto x encima de la mesa.

– *Putdown(x)*

precondiciones $Holding(x)$

lista a borrar $Holding(x)$

lista a añadir $Clear(x) \wedge OnTable(x) \wedge ArmEmpty$

Implementación del razonamiento práctico

■ Algoritmo global de un agente con razonamiento práctico:

```
Agent Control Loop Version 1
1. while true
2.   observe the world;
3.   update internal world model;
4.   deliberate about what intention to achieve
   next;
5.   use means-ends reasoning to get a plan for the
   intention;
6.   execute the plan
7. end while
```

Deliberación

- Cómo delibera un agente?
 - Primero empieza tratando de entender cuales son las *opciones* disponibles.
 - *Escoge* entre estas opciones y se compromete con alguna(s).
- Las opciones escogidas son las intenciones del agente inteligente.

Deliberación

- La función de *deliberación* puede ser descompuesta en otros dos componentes funcionales:
 - *Generador de opciones*
 - *Genera las posibles opciones (o deseos del agente).*
 - *Filtrado de opciones*
 - *Decide entre los posibles deseos, ve sus pros y contras y decide finalmente por comprometerse con alguna(s) intención(es).*

Arquitecturas reactivas

- Existen muchos problemas no resueltos por la IA simbólica.
- Estos problemas han llevado a cuestionar la viabilidad del enfoque simbólico y así al desarrollo de este nuevo enfoque de arquitecturas reactivas
- El principal iniciador de este enfoque fue: Rodney Brooks

El enfoque Brooks

- Brooks propuso tres ideas básicas:
 1. La conducta inteligentes puede ser generada sin usar representaciones explícitas del tipo que usa la IA simbólica.
 2. La conducta inteligente puede ser generada sin usar razonamiento abstracto explícito del tipo usado por la IA simbólica.
 3. La inteligencia es una propiedad emergente de ciertos sistemas complejos.

El enfoque de Brooks

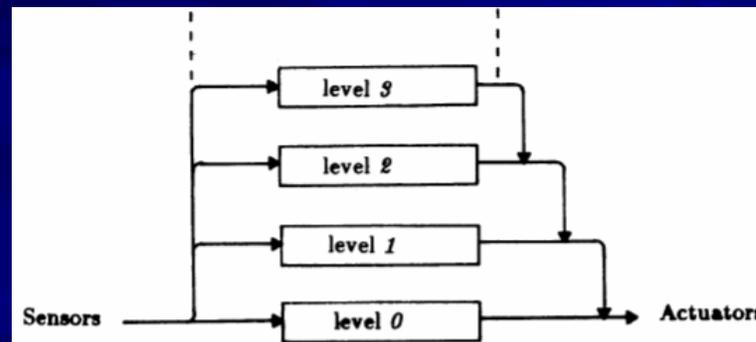
- Él identifica dos puntos clave:
 1. *“Situatèdness”* y *“embodiment”*: Inteligencia real es “situada” en el mundo y no sistemas “sin cuerpo” como los demostradores de teoremas.
 2. *“Intelligence”* y *“emergence”*: La conducta “Inteligente” viene como resultado “emergente” de las interacciones del agente con su medio ambiente. También la “inteligencia” está en la persona que observa el fenómeno y no es una propiedad aislada del sistema.

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Arquitectura “Subsumption”

Control por capas



From Brooks, “A Robust Layered Control System for a Mobile Robot”, 1985

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Explorador marciano de Steels

- Este explorador marciano usa la arquitectura reactiva de Brooks.
- Alcanza un rendimiento casi óptimo cooperativo en esta tarea.
- El objetivo es explorar un planeta distante y en particular coleccionar muestras de un cierto tipo de roca.
- La localización de las rocas no se conoce, pero sí se sabe que ellas tienden a estar agrupadas en regiones cercanas.

Las reglas de Steels

- Para agentes individuales (no cooperativos), el nivel más bajo de conducta (y por lo tanto la de más alta prioridad) es el evitar obstáculos:
if detect an obstacle then change direction (1)
- Cualquier muestra que es cargada por los agentes es depositada en la nave nodriza:
*if carrying samples and at the base
then drop samples* (2)

Las reglas de Steels

- Los agentes cargando muestras regresan a la nave nodriza:
if carrying samples and not at the base
then travel up gradient (3)
- Los agentes coleccionarán las muestras que encuentren:
if detect a sample then pick sample up (4)
- Un agente con “nada mejor que hacer” explorará aleatoriamente el medio ambiente:
if true then move randomly (5)

Ventajas del enfoque reactivo

- Simplicidad
- Economía
- Factibilidad
- Robustez
- Elegancia

Limitaciones del enfoque reactivo

- Los agentes sin modelos del ambiente deben tener suficiente de la información disponible del ambiente local
- Si las decisiones son tomadas en el ambiente local, cómo se puede tomar en cuenta información no-local?
- Es difícil hacer agentes reactivos que aprendan.
- Dado que la conducta emerge de interacciones con el ambiente, es difícil ver como *diseñar por adelantado* agentes específicos.
- Es difícil diseñar también agentes con grandes números de conductas (la dinámica de interacciones se convierte demasiado compleja para entenderla).

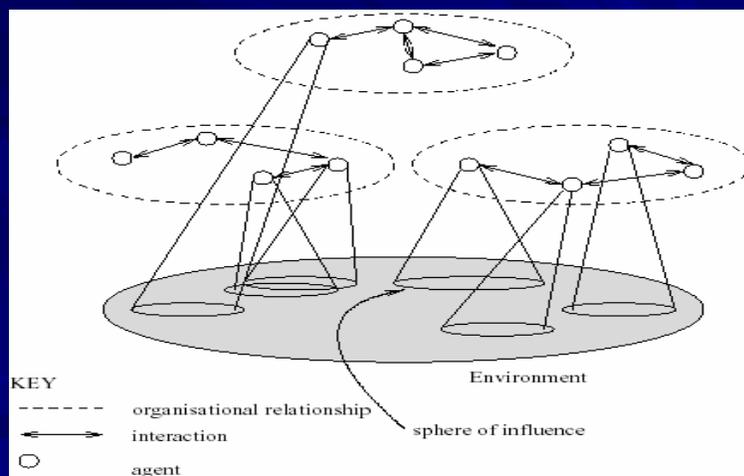
Arquitecturas híbridas

- Muchos investigadores han discutido que para construir agentes no es viable ni un enfoque puramente deliberativo ni un enfoque puramente reactivo.
- Entonces se ha sugerido un enfoque híbrido, en el cual se intenta conjuntar enfoques clásicos y alternativos.
- Un enfoque obvio es construir un agente a partir de dos (o más) subsistemas:
 - Uno deliberativo conteniendo un modelo del mundo simbólico que sea capaz de desarrollar planes y tomar decisiones complejas.
 - Otro reactivo que sea capaz de reaccionar a eventos más rápidamente sin razonamiento simbólico.

Contenido

- **Introducción**
 - Introducción a los Agentes
 - Agentes Inteligentes y Sistemas Multiagentes
- **Arquitecturas de Agentes**
 - Agentes deliberativos con razonamiento práctico
 - Agentes con arquitecturas reactivas e híbridas
- **Sistemas Multiagentes**
 - **Interacción y negociación entre agentes**
 - **Comunicación y coordinación entre agentes**
- **Aplicaciones de Sistemas Multiagentes**
 - Pasado, presente y futuro de los sistemas de agentes
- **Ejemplos y ejercicios en NetLogo**

Dónde están los Sistemas Multiagentes?

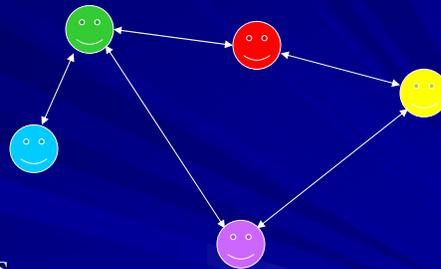


Los Sistemas Multiagentes

- Así, un sistema multiagente contiene un número de agentes que:
 - Interactúan con comunicación
 - Son capaces de actuar en el ambiente.
 - Tienen diferentes “esferas de influencia” (que pueden o no coincidir).
 - Están ligados unos a otros (organizacionalmente).

Sistemas Multiagentes

- Interacción de varios agentes similares o heterogéneos



- Sin control global
- Agentes descentralizados
- Agentes con conocimiento individual y limitado

Coordinación entre agentes



Acuerdos entre agentes

- Cómo los agentes pueden lograr acuerdos cuando ellos son egoístas?
- En el caso extremo (juegos suma-cero) los acuerdos no son posibles.
- Pero en la mayoría de los escenarios existe un *acuerdo mutuamente benéfico* en asuntos de intereses comunes.
- Las capacidades de negociación y argumentación son centrales para alcanzar tales acuerdos.

Coordinación entre agentes egoístas

- Cada agente busca maximizar su propio interés
- Cada agente supone que los demás también son interesados y egoístas
- Existe entonces la necesidad de protocolos de negociación



Mecanismos y protocolos

- La negociación es gobernada por un *mecanismo* o *protocolo* en particular.
- El mecanismo define las “reglas de encuentro” entre los agentes.
- El diseño del mecanismo es diseñar mecanismos de tal forma que se den ciertas propiedades deseadas.
- Dado un cierto protocolo, cómo puede una estrategia particular ser diseñada para que ciertos agentes la puedan usar?

Subastas

- Una subasta toma lugar entre agentes conocidos como el subastador y una colección de agentes conocidos como los participantes en la subasta.
- El objetivo de la subasta para el subastador es colocar el “bien” en uno de los participantes.
- En la mayoría de las subastas el subastador desea maximizar el precio y los participantes desean minimizar el precio.

Subasta inglesa

- Es el tipo de subasta más común:
 - “*First price*”
 - “*Open cry*”
 - “*Ascending*”
- La estrategia dominante para el agente es ir ofreciendo una pequeña cantidad más alta a la oferta actual hasta que alcanza su valor (para el agente) y entonces renuncia a la subasta.

Subasta Holandesa

- Este tipo de subastas son ejemplos de subastas
 - *“Open-cry”*
 - *“Descending”*
- El subastador empieza ofreciendo el “bien” a un valor artificialmente alto
- Entonces el subastador empieza a bajar el precio poco a poco hasta que algún agente hace una oferta igual al precio actual.
- El “bien” entonces es colocado al agente que ofertó.

Subastas de “Primer-Precio Cerrado”

- Este tipo de subastas son:
 - *“One-shot auctions”*
- Existe una sola ronda.
- Los agentes ofrecen una oferta cerrada por el “bien”.
- El “bien” es colocado al agente que hizo la oferta más grande.
- El ganador paga el precio más alto.

Subastas Vickrey

- Estas subastas son:
 - “*Second-price*”
 - “*Sealed-bid*”
- El “bien” es colocado al agente que hace la oferta más alta, pero al precio de la *segunda más alta* oferta.

Competencia por llamadas telefónicas

- El cliente desea hacer llamadas de larga distancia
- Los operadores de larga distancia hacen ofertas simultáneas, enviando sus precios propuestos.
- El teléfono automáticamente escoge el operador (dinámicamente).



La mejor oferta gana

- El teléfono escoge el operador con el precio más pequeño.
- El operador obtiene esa cantidad.



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Atributos de este mecanismo

- El operador tiene la iniciativa de invertir su esfuerzo en una conducta estratégica!



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Con una subasta Vickrey

- El teléfono escoge el operador con el precio más bajo, pero éste obtiene el segundo mejor precio!



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Atributos de la subasta Vickrey

- Los operadores no tienen incentivos en gastar su esfuerzo en conductas estratégicas!



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Mentiras y colisiones

- En los distintos tipos de subastas, el subastador podría mentir y además los participantes podrían colisionar.
- Los cuatro tipos de subastas pueden ser manipuladas por colisiones.
- Un subastador deshonesto puede explotar la situación.
- En la subasta inglesa los precios pueden ser inflados.

Negociación

- Las subastas son sólo para colocar “bienes”: para lograr acuerdos es necesario tener técnicas más elaboradas.
- *Negociación* es el proceso de lograr acuerdos en asuntos de interés común.
- Cualquier negociación tiene cuatro componentes:
 - Un *conjunto de negociación*: Las propuestas posibles por los agentes.
 - Un *protocolo*.
 - *Estrategias*, una para cada agente, las cuales son privadas.
 - Una *regla de acuerdo* que determina cuando un acuerdo ha sido alcanzado por todos los participantes.
- Las negociaciones usualmente proceden en una serie de rondas, donde cada agente hace una propuesta por ronda.

Agentes egoístas y heterogéneos

- En este tipo de sistemas:
 - No hay un diseño central.
 - No se tiene la noción de utilidad global.
 - Son sistemas dinámicos (por ejemplo nuevos tipos de agentes pueden surgir).
 - No hay “benevolencia” por parte de los agentes, al menos que esa sea precisamente su intención.

El objetivo de la investigación

- La ingeniería social de comunidades de máquinas:
 - La creación de ambientes de interacción para provocar ciertos tipos de conductas sociales

El uso de “Teoría de Juegos” para el diseño de protocolos de alto nivel

Suposición general

- Los diseñadores (de diferentes compañías, países, etc.) se juntan para acordar *estándares* para sus agentes automáticos (en un dominio dado).
- Discuten varias posibilidades, así como sus pros y contras.
- Entonces acuerdan en ciertos protocolos, estrategias y leyes sociales para ser implementadas en sus máquinas.

Atributos de los Estándares

- ✓ *Eficientes*: Óptimos (Pareto).
- ✓ *Estables*: Sin incentivos de desviarse.
- ✓ *Simples*: Costos de comunicación y computacionales bajos
- ✓ *Distribuidos*: Sin decisiones centrales
- ✓ *Simétricos*: Agentes juegan roles equivalentes

El diseño de protocolos para clases específicas de dominios que satisfagan algunos o todos estos atributos

Inteligencia Artificial Distribuida

- Solución de problemas distribuidos:
 - Sistemas diseñados centralmente, con la cooperación pre-construida y un problema global a resolver.
- Sistemas multiagentes:
 - ✓ Grupo de agentes heterogéneos que maximizan sus utilidades y en un mismo ambiente, incluso posiblemente competitivo.

Concesiones monotónicas

- En este protocolo la negociación procede en rondas.
- En la ronda 1, los agentes proponen simultáneamente una oferta de su *conjunto de negociación*.
- Un acuerdo es alcanzado si un agente encuentra que un acuerdo propuesto por otro es al menos tan bueno o mejor que el suyo.
- Si no hay acuerdo, entonces la negociación procede con otra ronda de ofertas simultáneas.
- En la ronda $u + 1$, a ningún agente se le permite hacer ofertas menos preferidas por el otro en el tiempo u
- Si ningún agente hace la concesión en alguna ronda $u > 0$, entonces la negociación termina con un acuerdo “conflicto”.

Argumentación

- Argumentación es el proceso de tratar de convencer a otros de algo.
- Gilbert (1994) identifica cuatro modos de argumentación:
 1. “*Logic*”: “Si tu aceptas A y A implica B, entonces tu debes aceptar B”
 2. “*Emotional*”: “Cómo te sentirías si esto te pasara a ti?”
 3. “*Visceral*”: “Cretino!”
 4. “*Kisceral*”: “Esto en contra de las enseñanzas cristianas!”

Comunicación entre agentes

- Necesidad de lenguajes estándares de comunicación
- Desde mediados de los 90's empezaron a desarrollarse los primeros: KIF, KQML
- En la actualidad: FIPA-ACL
- Basados en explicitar la intención del mensaje
- Ejemplo:



```
(inform
  :sender agent1
  :receiver hp1-auction-server
  :content (price (bid good02) 150)
  :in-reply-to round-4
  :ontology hp1-auction)
```

KQML y KIF

- El lenguaje de comunicación entre agentes mejor conocido es el KQML.
- Fue desarrollado por la ARPA en una iniciativa para compartir conocimiento.
- Este lenguaje esta compuesto de dos partes:
 - El lenguaje de manipulación y consulta de conocimiento (KQML)
 - Y el formato de intercambio de conocimiento (KIF)

KQML y KIF

- KQML define varios “verbos comunicativos” o *performativas*
- Ejemplos de performativas:
 - `ask-if` (‘is it true that. . .’)
 - `perform` (‘please perform the following action. . .’)
 - `tell` (‘it is true that. . .’)
 - `reply` (‘the answer is . . .’)
- KIF es un lenguaje para expresar el contenido del mensaje.

KIF

- Es usado para describir:
 - Propiedades de cosas en un dominio (“Fulano de Tal es el jefe”).
 - Relaciones entre cosas en un dominio (“Fulano es el jefe de Sutano”).
 - Propiedades generales de un dominio (“Todos los estudiantes están registrados en al menos un curso”).

KIF

- “La temperatura de m1 es 83 grado Celsius”:
`(= (temperature m1) (scalar 83 Celsius))`
- “Algún x es soltero si ese x es hombre y no está casado”:
`(defrelation bachelor (?x) :=
 (and (man ?x) (not (married ?x))))`
- “Cualquier individuo con la propiedad de ser una persona también tiene la propiedad de ser un mamífero”:
`(defrelation person (?x) :=> (mammal ?x))`

KQML y KIF

- Para poderse comunicar, los agentes deben estar de acuerdo en un conjunto de términos.
- La especificación formal de un conjunto de terminos es conocido como una *ontología*.
- Ejemplo de un diálogo KQML/KIF:

```
A to B: (ask-if (> (size chip1) (size
                  chip2)))
B to A: (reply true)
B to A: (inform (= (size chip1) 20))
B to A: (inform (= (size chip2) 18))
```

FIPA ACL

- Más recientemente, la FIPA (Foundation for Intelligent Physical Agents) inició un trabajo en la creación de estándares para agentes y la pieza central es el lenguaje ACL
- La estructura básica es muy similar a KQML:
 - performativas*
 - 20 performativas en FIPA
 - “*Housekeeping*”:
Por ejemplo, el emisor, receptor, etc.
 - “*Content*”:
El contenido en sí del mensaje.

FIPA ACL

■ Un ejemplo:

```
(inform
  :sender      agent1
  :receiver   agent5
  :content (price good200 150)
  :language   sl
  :ontology   hpl-auction
)
```

Las performativas del FIPA ACL

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

“Inform” y “Request”

- Las performativas “Inform” y “Request” son las dos performativas básicas en el ACL de FIPA. Todas las demás son definiciones *macro* en términos de éstas.
- El significado de “inform” y “request” es definido en dos partes:
 - La pre-condición
 - El efecto racional.

“Inform” y “Request”

- Para la performativa “inform” :
 - El contenido es un *enunciado*.
 - La pre-condición quiere decir que el emisor:
 - Crea que el contenido es verdadero.
 - Tenga la intención de que el receptor crea el contenido.
 - No crea ya que el receptor está consciente de si el contenido es verdadero o falso.

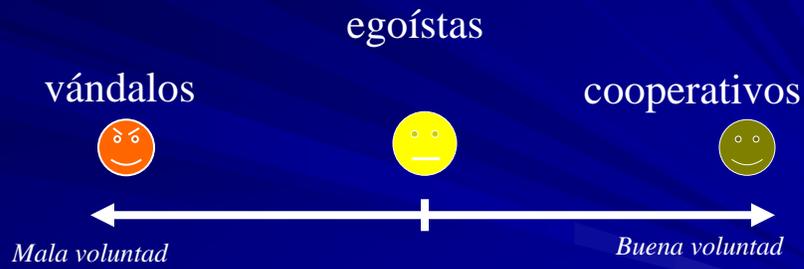
“Inform” y “Request”

- Para la performativa “request”:
 - El contenido es una *acción*.
 - La pre-condición quiere decir que el emisor:
 - Tenga la intención de que el receptor ejecute la acción .
 - Cree que el receptor es capaz de ejecutar la acción
 - No crea que el receptor ya tiene la intención de ejecutar la acción.

Agentes trabajando juntos!

- Por qué y cómo los agentes pueden trabajar juntos?
- Es importante hacer la distinción entre:
 - *Agentes benevolentes*
 - *Agentes egoístas*

Cooperación / competencia



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Agentes benevolentes

- Si nosotros somos “dueños” de todo el sistema, entonces podemos diseñar agentes que se ayuden unos a otros.
- En este acaso, se asume que los agentes son benevolentes: nuestro interés es su mejor interés.
- La solución de problemas en sistemas benevolentes es *solución cooperativa de problemas distribuidos* (CDPS)
- *La benevolencia simplifica enormemente la tarea del diseño del sistema!*

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Agente egoístas

- Si los agentes representan intereses individuales (o de organización en casos más generales), entonces no podemos asumir la suposición de benevolencia.
- Entonces es mejor asumir que los agentes van a perseguir sus propios intereses, quizá incluso a expensas de los otros.
- Existe entonces una situación potencial de conflicto.
- Todo esto puede complicar enormemente la tarea de diseño del sistema.

Coordinación entre agentes cooperativos

- Distribución de tareas, recolección de resultados
- Solución Distribuida de Problemas:
 - Subdivisión de una tarea en varias
 - Coordinación de soluciones
 - Solución de conflictos
 - Recolección e integración de resultados
- Soluciones muy robustas, eficientes y escalables



Solución Cooperativa de Problemas Distribuidos (CDPS)

- No hay control global ni almacenamiento global.
- Ningún agente tiene información suficiente para resolver el problema entero.
- El control y los datos están distribuidos.

Compartiendo tareas y resultados

- Existen dos modos principales para la solución cooperativa de problemas:
 - “*Task sharing*”:
Los componentes de la tarea son distribuidos entre los agentes.
 - “*Result sharing*”:
La información (resultados parciales, etc.) es distribuida entre los agentes.

El protocolo “Contract Net”

- Un protocolo muy bien conocido de “task sharing” es el “Contract Net”.
- Este protocolo consta de las siguiente fases:
 1. Reconocimiento
 2. Anuncios
 3. Ofertas
 4. Contratación
 5. Ejecución

Reconocimiento

- En esta fase, un agente reconoce que tiene un problema que quiere resolver.
- Y el agente entonces tiene un objetivo, pero:
 - Se da cuenta de que no puede alcanzar dicho objetivo por sí solo (no tiene la capacidad).
 - Se da cuenta que no sería preferible alcanzar dicho objetivo por sí mismo (por la calidad de la solución alcanzada, el tiempo límite para realizarlo, etc.).

Anuncios

- En esta fase, el agente con la tarea envía anuncios de la tarea incluyendo una especificación de la tarea a ser alcanzada.
- La especificación debe incluir:
 - Una descripción de la tarea
 - Las restricciones a cumplir (fechas límites, de calidad)
 - Información de alto nivel de la tarea (por ejemplo, las ofertas deben ser alcanzadas para tal fecha ...)
- Los anuncios son enviados en modo “broadcast”.

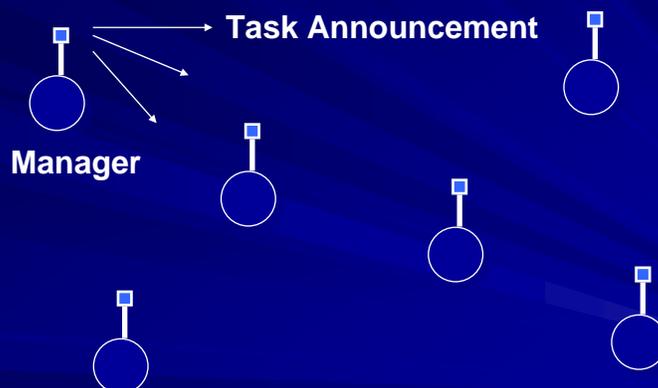
Ofertas

- Los otros agentes reciben los anuncios y deciden por sí mismos sus ofertas para la tarea anunciada.
- Factores:
 - El agente debe decidir si es capaz de ejecutar la tarea.
 - El agente debe determinar las restricciones de calidad y precio (si esto es relevante)
- Si ellos deciden ofertar, entonces envían la oferta al anunciante seleccionado.

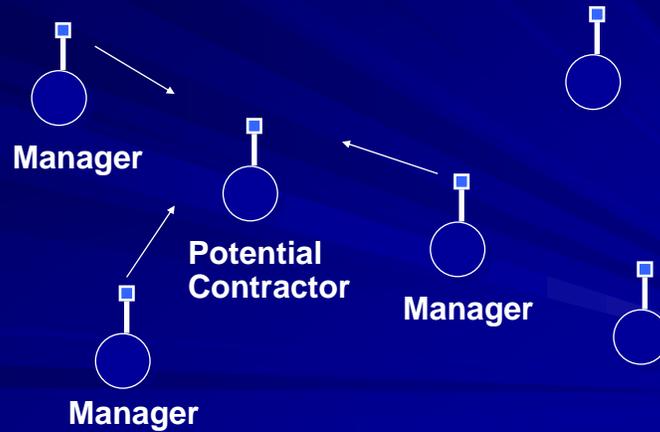
Contratación y Ejecución

- El agente anunciador debe escoger entre las ofertas y decidir con quién hacer compromiso y contratar a dicho agente.
- El resultado del proceso es comunicado a los agentes que enviaron la oferta.
- Los agentes que ofertan entonces ejecutan las tareas encomendadas.
- Este protocolo puede generar a su vez más relaciones de contratación en sub-contrataciones de otros agentes.

Los agentes anuncian



Los agentes escuchan los anuncios



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

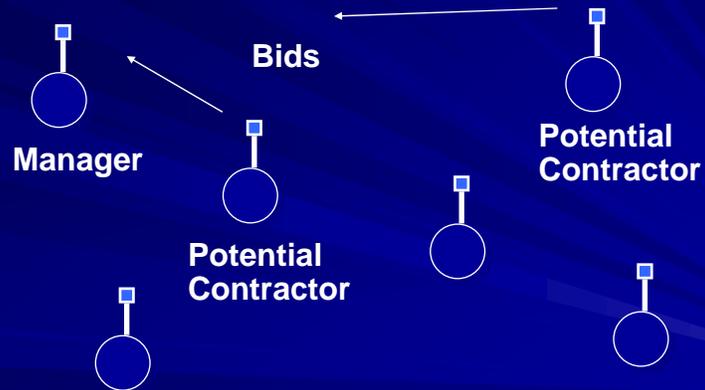
Los agentes envían sus ofertas



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

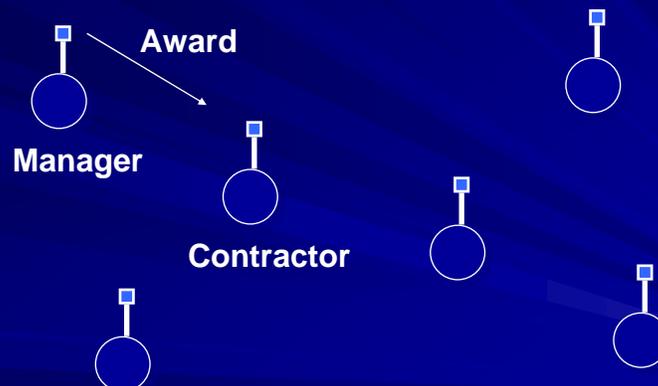
Los iniciadores escuchan las ofertas



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

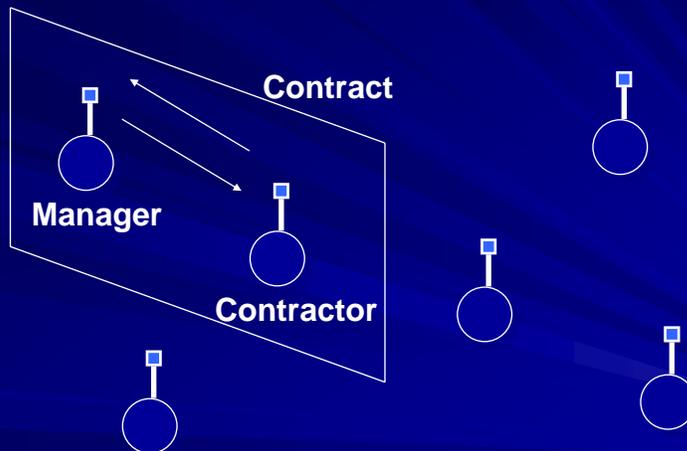
Los iniciadores hacen compromisos



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Los contratos son establecidos



Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Atributos de este protocolo

- Existe una transferencia de información en dos vías.
- La evaluación es local.
- Selección mutua (los agentes seleccionan entre diferentes anunciadores y los anunciadores entre diferentes agentes).

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Limitaciones de este protocolo

- Algunas partes del problema no son triviales:
 - La descomposición del problema en tareas.
 - La síntesis de la solución.
- Existe un “overhead” en el sistema.
- Existen también varios métodos alternativos para llevar a cabo los anuncios, la evaluación de tareas y de ofertas.

Contenido

- **Introducción**
 - Introducción a los Agentes
 - Agentes Inteligentes y Sistemas Multiagentes
- **Arquitecturas de Agentes**
 - Agentes deliberativos con razonamiento práctico
 - Agentes con arquitecturas reactivas e híbridas
- **Sistemas Multiagentes:**
 - Interacción y negociación entre agentes
 - Comunicación y coordinación entre agentes
- **Aplicaciones de Sistemas Multiagentes**
 - **Pasado, presente y futuro de los sistemas de agentes**
- **Ejemplos y ejercicios en NetLogo**

Cuándo un sistema basado en agentes es apropiado?

- El ambiente es abierto o al menos es altamente dinámico, incierto o complejo.
- Los agentes es la metáfora natural.
- Los datos o el control está naturalmente distribuido.
- Existen ya en uso “legacy systems”.

Desarrollo de sistemas de software basados en agentes

- Existen ya *muchos* proyectos de sistemas de agentes individuales o sistemas de múltiples agentes ...
- Sin embargo, el *desarrollo* orientado a agentes a penas se está consolidando.

Áreas de aplicación

- Los agentes son útiles en dominios donde se requieren acciones autónomas.
- Agentes inteligentes son útiles en dominios donde la acción debe ser flexible.
- Las principales áreas de aplicación en sistemas de software son:
 - *Sistemas distribuidos/concurrentes.*
 - *Redes de computadoras*
 - *Interfaces humano-computadora.*
 - *Aplicaciones en Internet.*
- En sistemas de hardware también hay muchas áreas de aplicación para sistemas robóticos multiagentes!

Sistemas distribuidos

- En esta área, la idea de agentes se ve como una metáfora natural, así como el desarrollo de la idea programación de objetos concurrentes.
- Ejemplos de dominios:
 - Control de tráfico aéreo.
 - Administración de procesos de negocios.
 - Administración de sistemas de energía.
 - Sensado distribuido.
 - Control de procesos en fábricas.

Redes de computadoras

- Actualmente existe mucho interés en los agentes móviles que pueden moverse a través de la red (o por Internet) por sí mismos.
- También últimamente el interés en “grid computing”, “embedded systems” y “ubiquitous systems” está creciendo enormemente, dando grandes oportunidades a los sistemas basados en agentes.
- Las aplicaciones se pueden incluir:
 - En Hand-held PDAs con ancho de banda limitados.
 - La Recolección de información y procesamiento paralelo, entre otras.

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Interfaces humano-computadora

- Otra área de mucho interés actual es el uso de agentes para interfaces humano-computadora.
- La idea es avanzar y dejar atrás el paradigma actual de *manipulación directa* que domina en estos días.
- La idea es que los agentes estén detrás de las aplicaciones observando, aprendiendo y eventualmente haciendo cosas sin que se les diga directamente que las tienen que hacer (que sean pro-activos y tomen la iniciativa).
- Algunas aplicaciones pueden ser en navegadores de Internet, lectores de emails y noticias por ejemplo.

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Agentes en Internet

- El *potencial* de Internet es enorme.
- La *realidad* es sin embargo a veces no es prometedora:
 - Internet es enorme y encontrar información no siempre es fácil (incluso con motores de búsqueda como Google!).

Agentes en Internet

- *Búsquedas sistemáticas son difíciles:*
 - *Factores humanos:* Nosotros a menudo nos aburrirnos por tiempos de respuesta lentos, encontramos difícil de leer el *www*, nos cansamos, nos distraemos, etc.
 - *Factores organizacionales:* La infraestructura de la red es sólo superficial (faltan todavía muchos estándares para *home pages*, marcas semánticas que digan el significado de los contenidos, etc.).
- La cantidad de información que nos es presentada nos lleva a la “sobrecarga de información”

Agentes en comercio y negocios

- Otra motivación para los agentes en Internet es su potencial como agentes en comercio electrónico y negocios.
- La mayoría del comercio en la actualidad es hecho manualmente. Sin embargo, esto está empezando a cambiar y las tareas empiezan a ser delegadas a agentes.

Agentes en comercio y negocios

- Un ejemplo simple: Encontrar la copia de Office 2003 de las tiendas en línea.
- Un ejemplo un poco más complejo: volar de México a Europa con comida vegetariana, asiento en la ventanilla y reservaciones de hotel y auto por adelantado.

Contenido

- **Introducción**
 - Introducción a los Agentes
 - Agentes Inteligentes y Sistemas Multiagentes
- **Arquitecturas de Agentes**
 - Agentes deliberativos con razonamiento práctico
 - Agentes con arquitecturas reactivas e híbridas
- **Sistemas Multiagentes:**
 - Interacción y negociación entre agentes
 - Comunicación y coordinación entre agentes
- **Aplicaciones de Sistemas Multiagentes**
 - Pasado, presente y futuro de los sistemas de agentes
- **Ejemplos y ejercicios en NetLogo**

Dr. Leonardo Garrido

Agentes Inteligentes y Sistemas Multiagentes

Agentes Inteligentes y Sistemas Multiagentes

Tutorial
MICAÍ 2006

Dr. Leonardo Garrido
Centro de Sistemas Inteligentes
Tecnológico de Monterrey
E-mail: leonardo.garrido@itesm.mx
Web page: <http://www-csi.mtv.itesm.mx/~lgarrido>