Sistema Multi-Agente para la Generación de Movimientos de Ajedrez

Juan Fco. Rodríguez Hervella

Departamento de INGENIERIA TELEMATICA
www.it.uc3m.es

Indice

- Introducción.
- MAS para la Generación de Movimientos.
- Arquitectura FIPA-OS.
- Trabajos Futuros.
- Referencias.

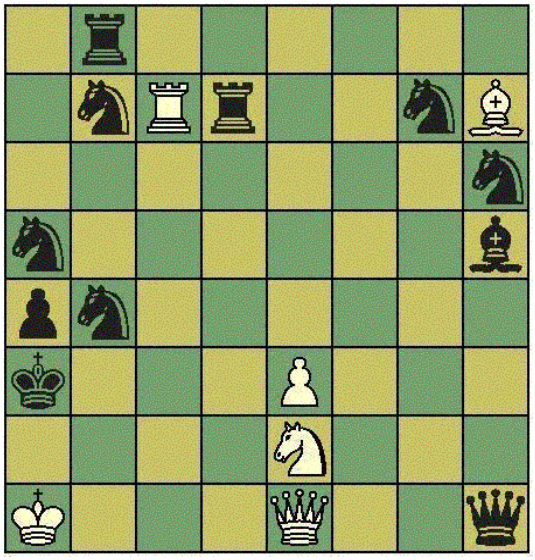
Introducción

- **Juegos de “ perfect information”**

  Ajedrez

  Damas

  Tres en Raya...



- **Ajedrez: métodos utilizados**

  Fuerza bruta (http://www.research.ibm.com/deepblue)

  Redes neuronales (http://neural-chess.netfirms.com)

  Búsqueda distribuida (http://chessbrain.net)

  Agentes ?

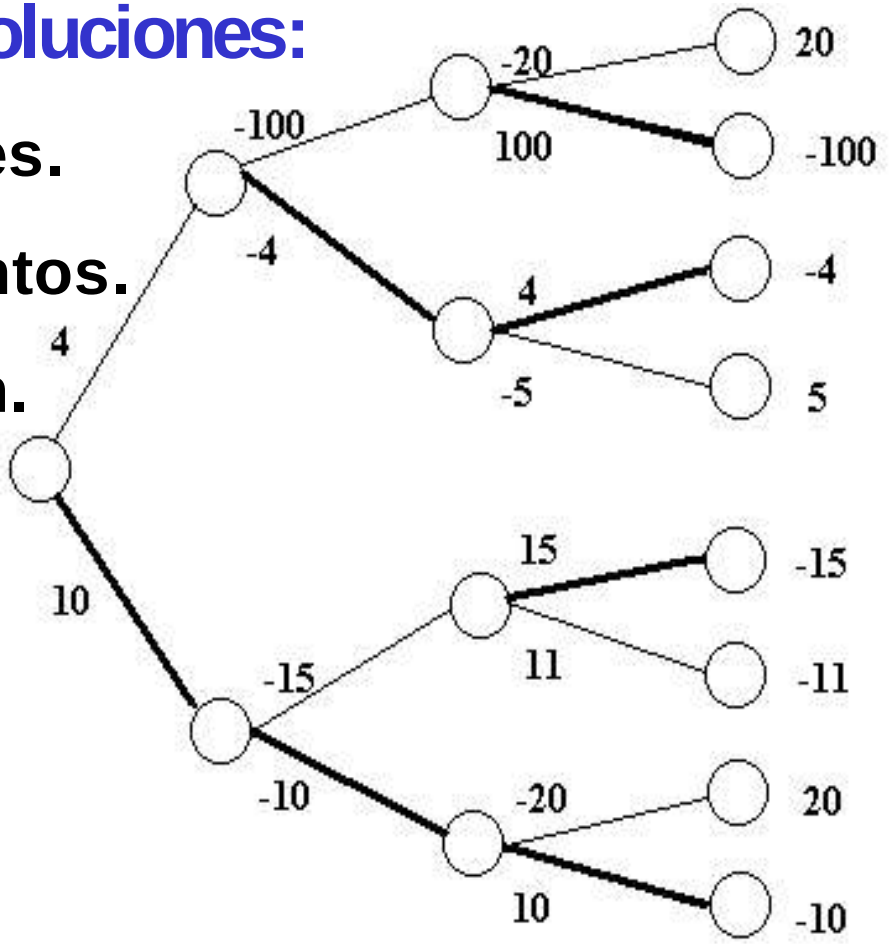
Introducción

- **Fuerza bruta no es factible.**
- **Búsqueda del espacio de soluciones:**

  Minimax + optimizaciones.

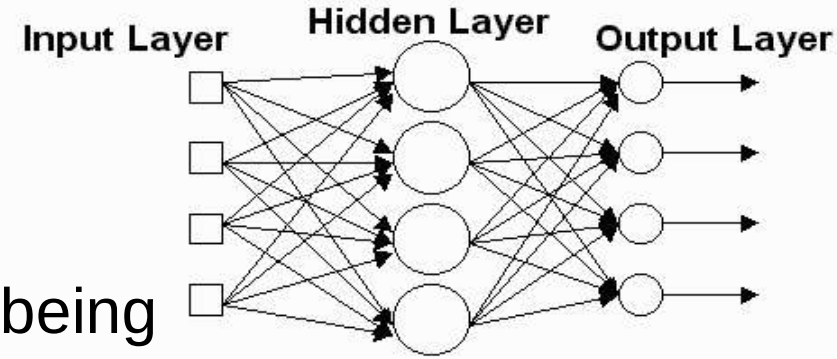
  Generación de movimientos.

  Funciones de evaluación.



Introducción

- **Redes neuronales:**



Basados en la evolución:

" Evolutionary algorithms being applied to a population of potential candidate evaluation functions" (genetic population)

Reproducción de la población:

" The likelihood of an individual to pass on its genetic information to the next generation is proportional to its performance in finding the correct moves for known given problems"

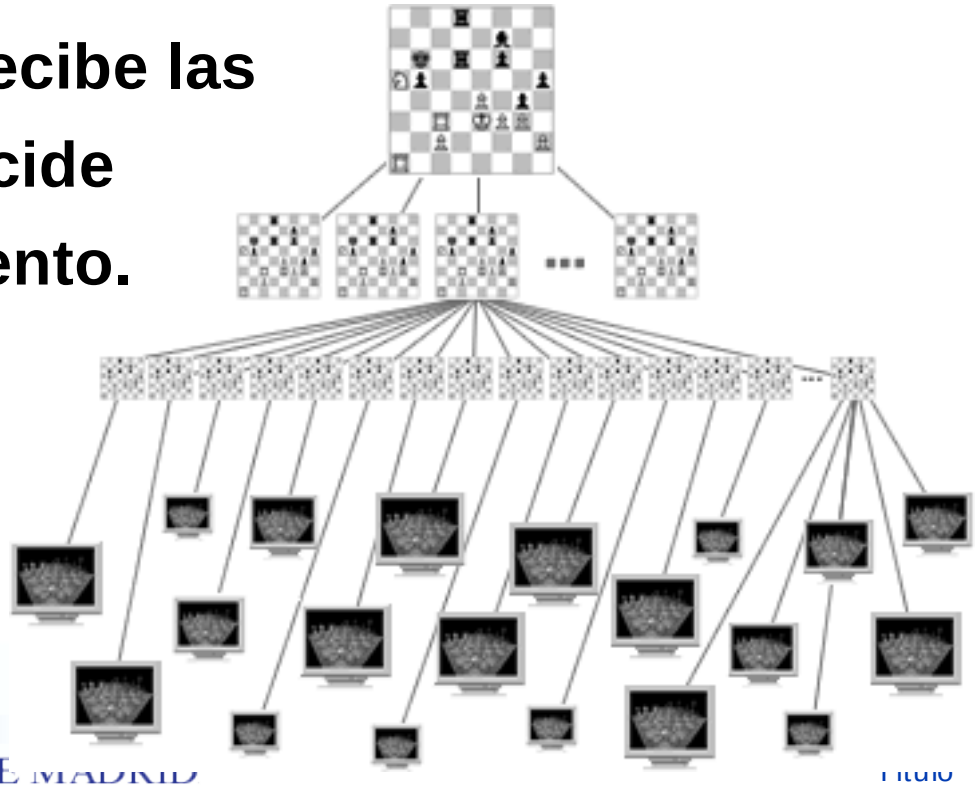
Introducción

- **Búsqueda distribuida:**

  Servidor central distribuye los sub-árboles.

  Clientes distribuidos evalúan la posicion.

  El servidor central recibe las contribuciones y decide el siguiente movimiento.



Introducción

- **Agentes que juegan al ajedrez:**

  Basado en el estudio:

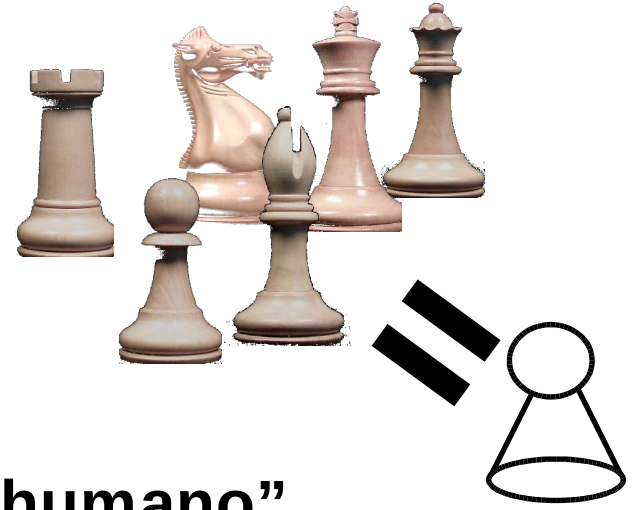
  "When Ants Play Chess", A. Drogoul, 1995

  Cada pieza es un agente.

  Objetivo codificado en reglas.

  Visión local del entorno.

  Comportamiento autónomo.



Idea: "jugar como lo hace un ser humano"

Problema: juega "mal" (los humanos también !).

MAS para la Generación de Movimientos

- **¿ Por qué el nivel es bajo ?**

  El mejor movimiento es determinado por el entorno y conocimientos locales del agente.

- **Idea: utilizar agentes para:**

  Sugerir una ordenacion de los movimientos posibles al "jefe" (el Rey o un nuevo agente)

- **Objetivo: "Taking the best of both worlds"**

  Táctica definida por los agentes.

  Estrategia basada en la decisión del "jefe".

MAS para la Generación de Movimientos

- **¿ Que mejoramos respecto a la fuerza bruta ?**

  Partidas rápidas

  los agentes deciden.

  Partidas lentas:

  El rendimiento mejora pues los mejores movimientos (según los agentes) se analizan antes.

- **Prueba:**

  Implementar, jugar....y ver que pasa.

MAS para la Generación de Movimientos

- **Algoritmo Base:**

  Valoración material de cada agente:

  Peón = 1, Caballo/Alfil = 3, Torre = 4, Dama = 10

  Cada posición del tablero sabe:

  La pieza que contiene.

  WhiteStrength (WS), BlackStrength (BS).

  WhiteDiff (WD = WS – BS).

  BlackDiff (BD = BS - WS).

MAS para la Generación de Movimientos

Sumar (+10) en los cuadros que se amenazan.

Informar a las piezas amenazadas.

Evaluación de cada casilla que puede ser ocupada:

(+ 2 * valor material de pieza comida)

(+ valor material de piezas amezadas)

Las piezas amenazas informan de este valor.

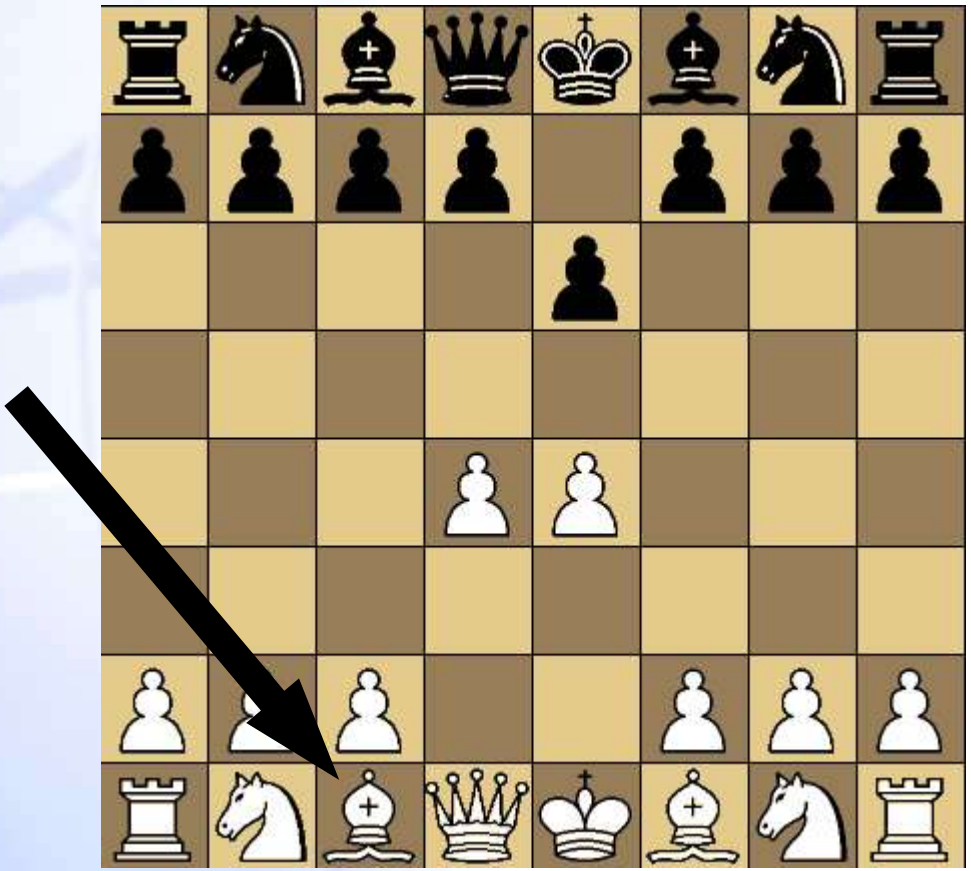
(+ valor material de piezas protegidas)

(- valor material del agente)

(- WD/BD del lugar que ocupa el agente).

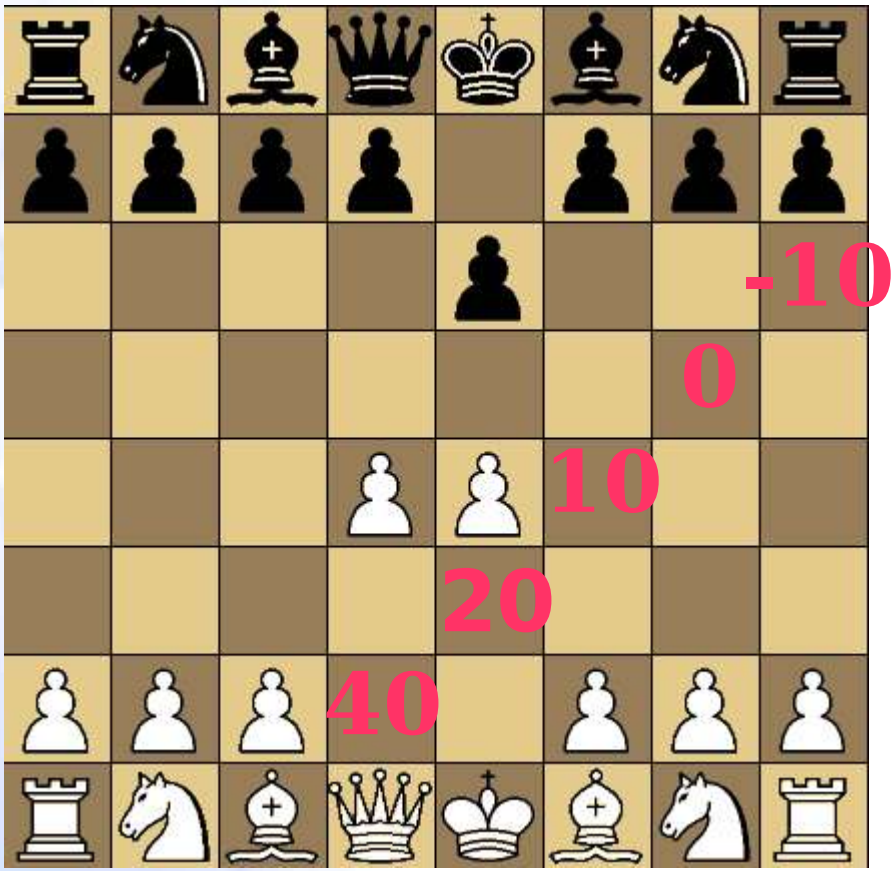
MAS para la Generación de Movimientos

- **Ejemplo:**



MAS para la Generación de Movimientos

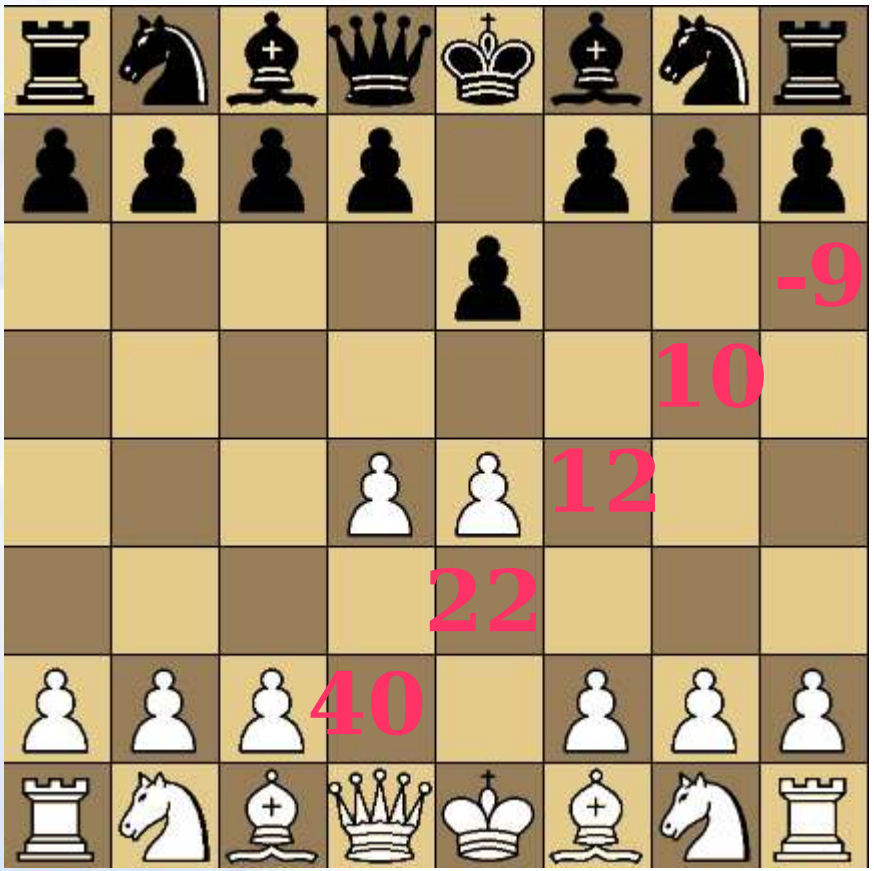
- **Ejemplo:**



1. Evaluación

MAS para la Generación de Movimientos

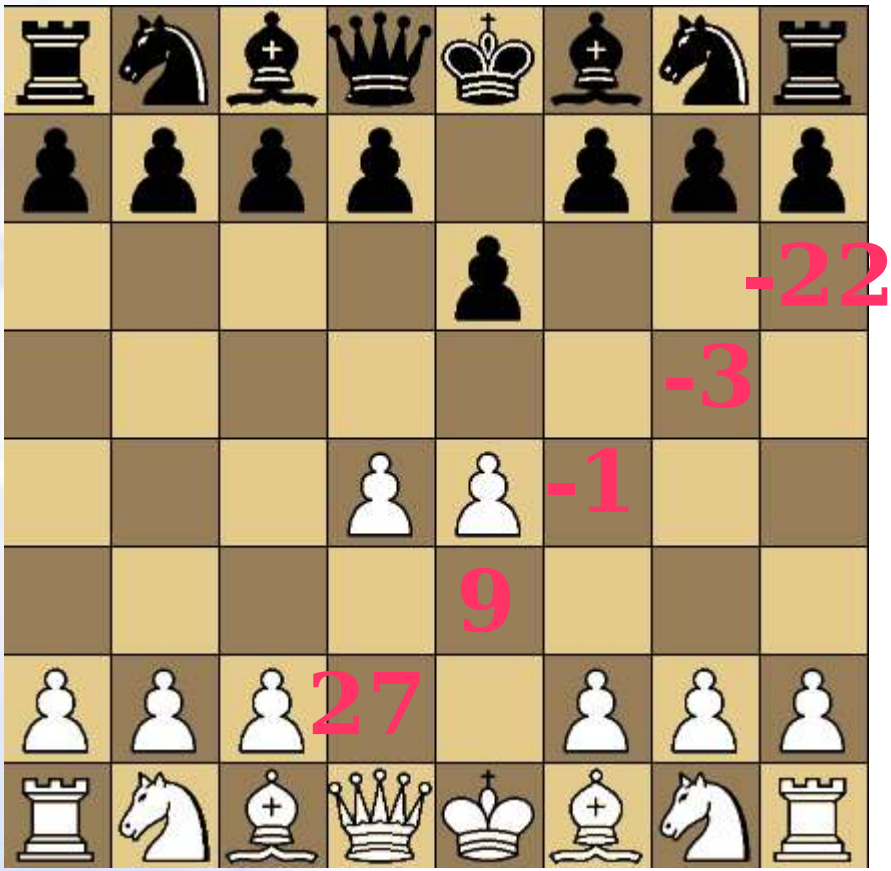
- **Ejemplo:**



1. Evaluación
2. Ataque y defensa

MAS para la Generación de Movimientos

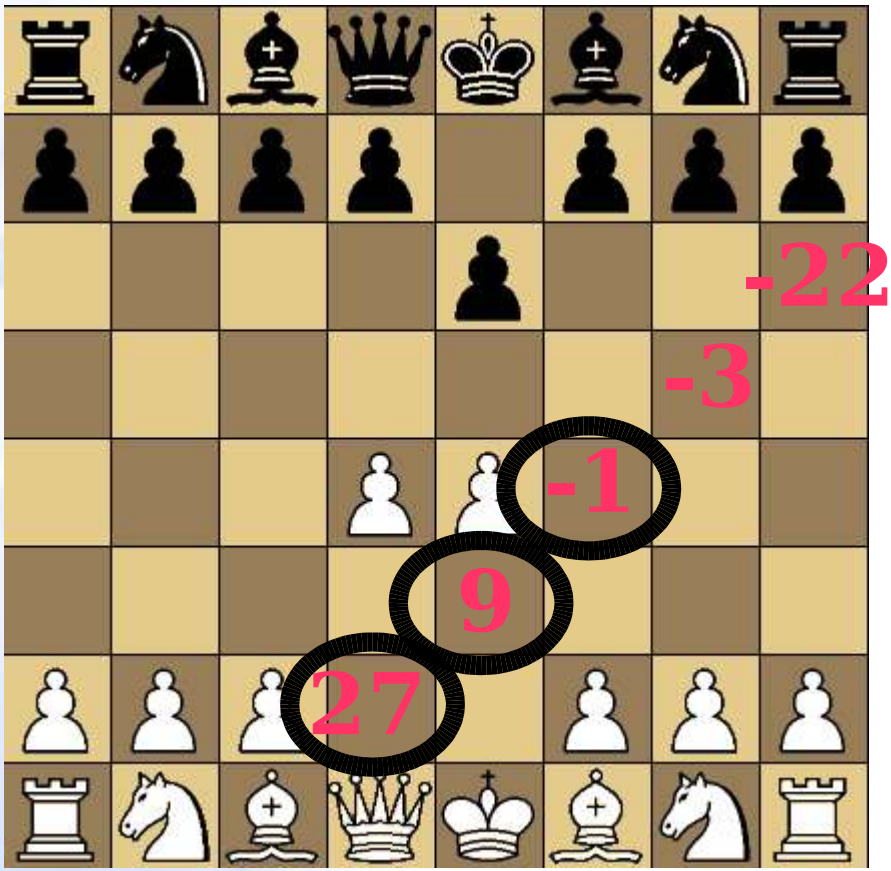
- **Ejemplo:**



1. Evaluación
2. Ataque y defensa
3. Desplazamiento

MAS para la Generación de Movimientos

- **Ejemplo:**



1. Evaluación
2. Ataque y defensa
3. Desplazamiento
4. Selección aleatoria

MAS para la Generacion de Movimientos

- **Resultados obtenidos (1995):**

  Contra humanos:

  Victorias: 57, Derrotas: 83, Tablas: 60 (28.5%)

  Contra GNU Chess:

  Victorias: 0, Derroras: 50, Tablas: 0

- **Conclusión:**

  “El programa es MUY MALO”.

MAS para la Generacion de Movimientos

- **Algoritmo Modificado:**

  Comunicar los movimientos evaluados a un agente central (“jefe”).

  El agente central ordena los movimientos, y selecciona los mejores:

  Descartando el resto (“Human Behaviour”)

  No descartando ninguno.

- **Esto sólo tiene sentido si los agentes son “rapidos” !**
- **Implementación en FIPA-OS !**

FIPA-OS

- **¿Por qué usar FIPA-OS ?**

  Evaluación de la plataforma :-)

  Maximizar la "interoperabilidad".

- **Instalación de la plataforma (~ 5 min.):**

  Requisitos:

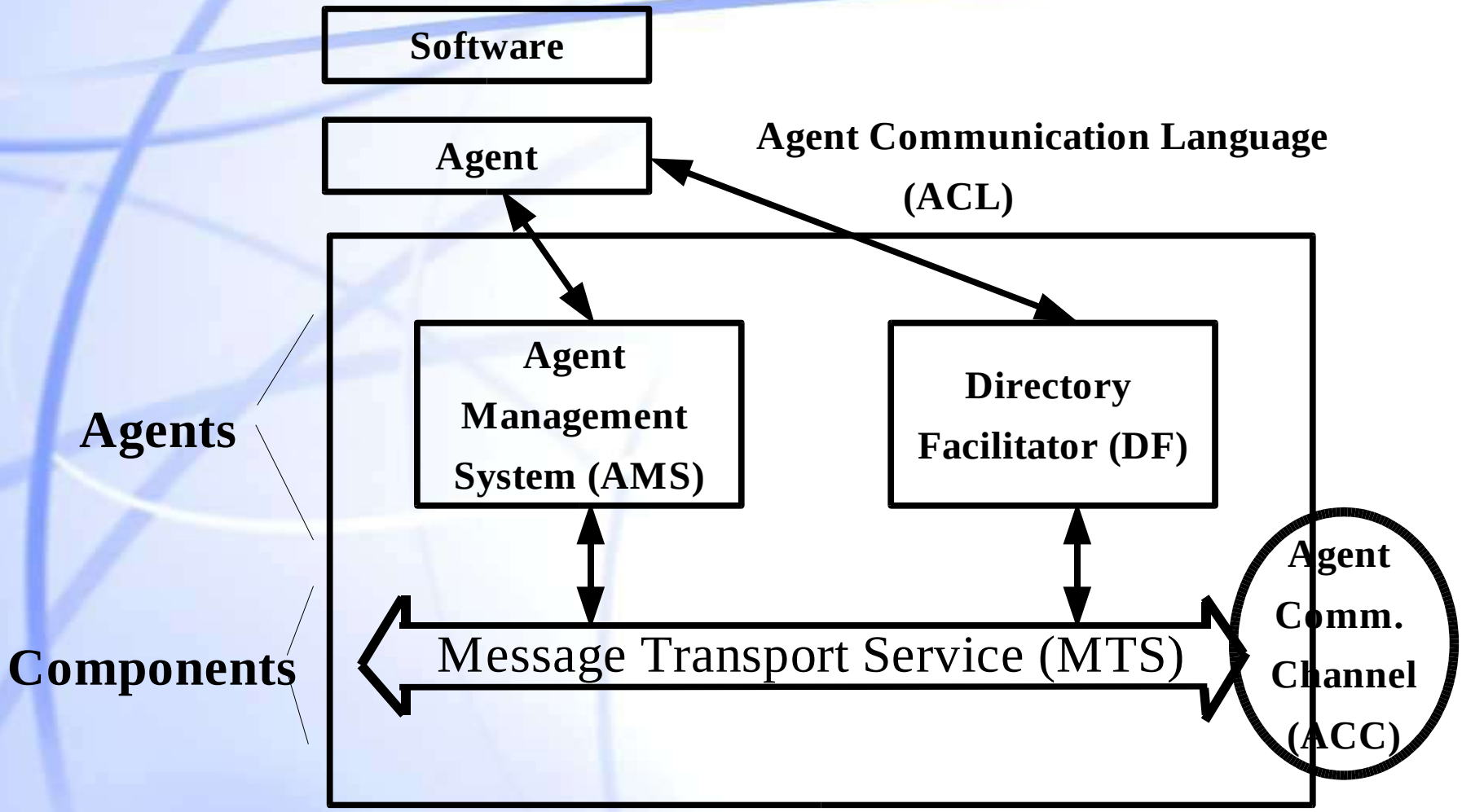
  Pentium 166, 64 Mbytes RAM, 4 Mbytes disco.

  Java 1.2.2

  Plataforma utilizada:

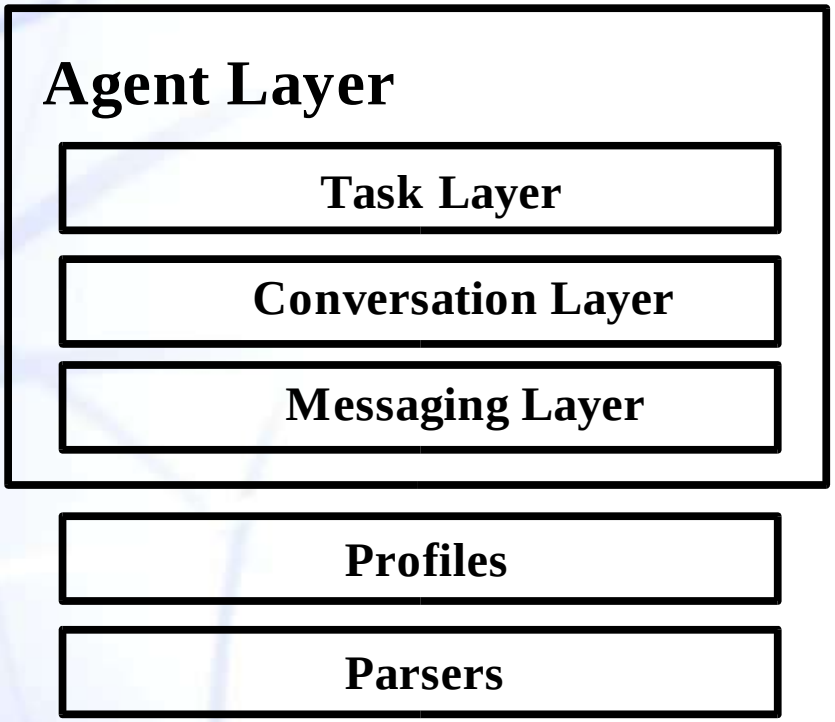
  FreeBSD-4.10 + Java 1.4.2

FIPA-OS



FIPA reference model

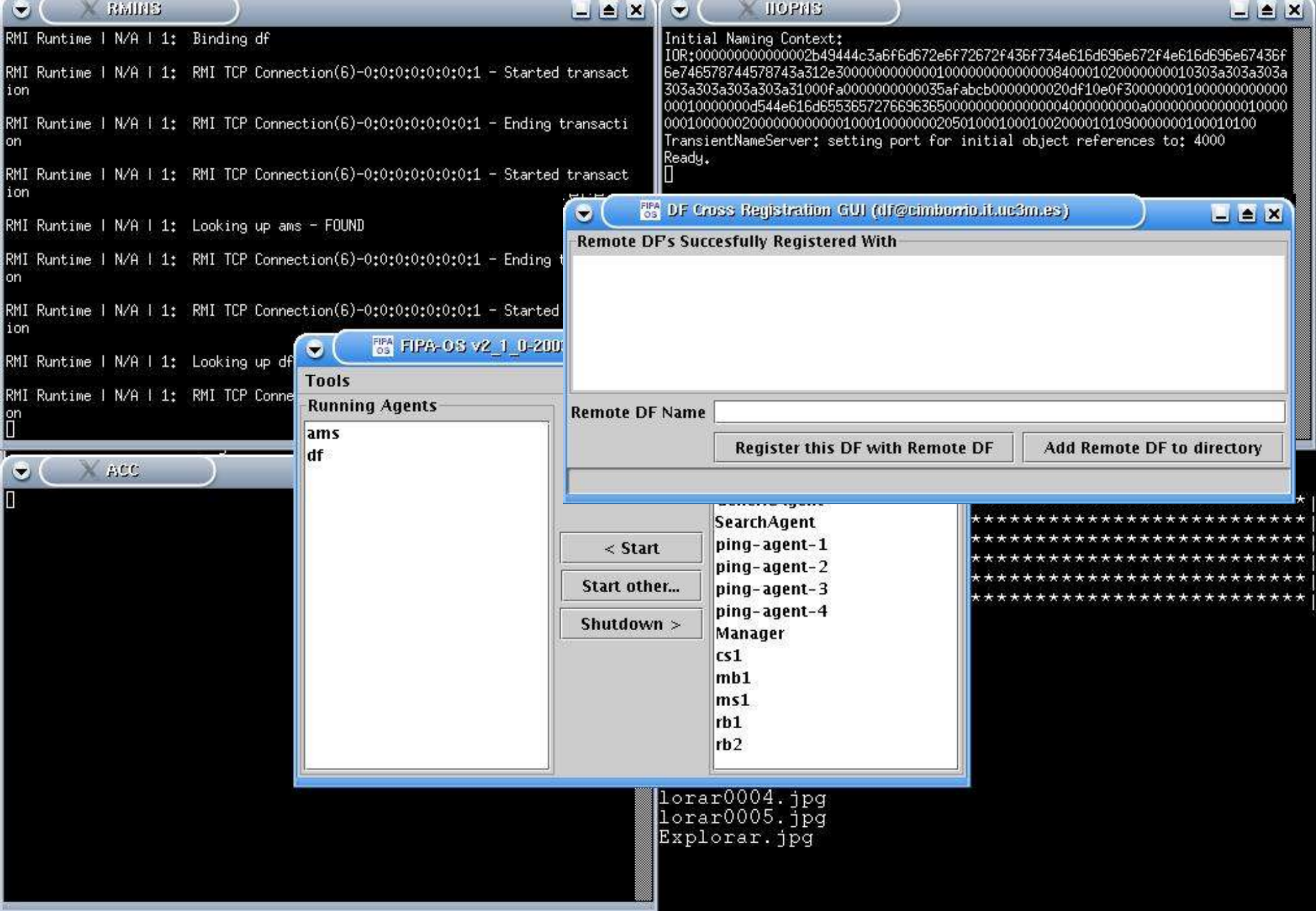
FIPA-OS



FIPA-OS architecture stack

FIPA-OS

Ruta de archivo	Descripción
\bat	scripts para ejecutar la plataforma
\certificates	claves y certificados de agentes. RMI + SSL
\classes\ FIPA_OSv2_1_0.jar	FIPA-OS core compilado sin opciones de depuración.
\databases	Lugar donde se almacenan datos persistentes
\imports	Componentes de terceras partes (e.g. Xerces)
\src	Código fuente
\javadocs	Documentacion de clases
\docs	Distribution notes.pdf (~ manual)
\docs\licenses	Licencia
\tools	Herramientas de ayuda
\profiles	Profiles de agentes suministrados
\examples	Mensajes ACL de ejemplo


RMINS
RMI Runtime | N/A | 1: Binding df
RMI Runtime | N/A | 1: RMI TCP Connection(6)-0:0:0:0:0:0:0:1 - Started transact
ion
RMI Runtime | N/A | 1: RMI TCP Connection(6)-0:0:0:0:0:0:0:1 - Ending transacti
on
RMI Runtime | N/A | 1: RMI TCP Connection(6)-0:0:0:0:0:0:0:1 - Started transact
ion
RMI Runtime | N/A | 1: Looking up ams - FOUND
RMI Runtime | N/A | 1: RMI TCP Connection(6)-0:0:0:0:0:0:0:1 - Ending
on
RMI Runtime | N/A | 1: RMI TCP Connection(6)-0:0:0:0:0:0:0:1 - Started
ion
RMI Runtime | N/A | 1: Looking up df
RMI Runtime | N/A | 1: RMI TCP Conne
on
IIOPNS
Initial Naming Context:
IOR:000000000000002b49444c3a6f6d672e6f72672f436f734e616d696e672f4e616d696e67436f
6e746578744578743a312e3000000000000100000000000000840001020000000010303a303a303a
303a303a303a303a31000fa000000000035afabcb0000000020df10e0f3000000100000000000
0001000000d544e616d6553657276696365500000000000000040000000000a000000000000010000
0001000000200000000000010001000000020501000100010020000101090000000100010100
TransientNameServer: setting port for initial object references to: 4000
Ready.
DF Cross Registration GUI (df@cimbornio.it.uc3m.es)
Remote DF's Succesfully Registered With
Remote DF Name
Register this DF with Remote DF
Add Remote DF to directory
FIPA-OS v2_1_0-200
Tools
Running Agents
ams
df
< Start
Start other...
Shutdown >
SearchAgent
ping-agent-1
ping-agent-2
ping-agent-3
ping-agent-4
Manager
cs1
mb1
ms1
rb1
rb2
ACC
lorar0004.jpg
lorar0005.jpg
Explorar.jpg

Trabajos Futuros

- **1. Validar el sistema mediante experimentación:**

  a) Implementar los agentes en FIPA-OS.

- **2. Extraer conclusiones:**

  Ajedrez:

  Aprender estrategias nuevas => nuevas funciones de evaluacion....”feedback”

  Otros ámbitos:

  Simulacion de sociades complejas jerarquizadas (los agentes NO tienen la última palabra).

Referencias

- A. Drogoul, “When Ants Play Chess (Or Can Strategies Emerge From Tactical Behaviours?)”, 1995.
- R. Brooks, “Elephants Don't Play Chess”, USA Robotics and Autonomous Systems, 1990, MIT.
- P. Frey, “An Introduction to Computer Chess”, in “Chess Skills in Man and Machine”, Springer-Verlag, NY, 1977
- S. Poslad and P. Charlon, “Standardizing Agent Interoperability: The FIPA approach”, Multi Agent Systems and Applications, 9th ECCAI, Prague, Czech Republic, 2001.