



Práctica de Inteligencia Artificial.

Sistemas de Producción en CLIPS

Implementación de un Juego de Simulación Social

Curso 2009-10 (Febrero)

1. Descripción

El objetivo de esta práctica es la implementación en CLIPS de un sistema de producción para simular un juego de sociedad virtual tipo los SIMS.

El juego consiste en la simulación del comportamiento de dos o más personajes virtuales que actúan de forma autónoma en un mismo escenario. En cada turno, uno de los personajes elegirá y ejecutará una sola acción de entre todas las posibles acciones que puede realizar.

Los personajes vienen caracterizados por unos *indicadores* y un estado emocional. Los *indicadores* representan las necesidades básicas y pueden tomar valores entre 0, cuando la necesidad está completamente satisfecha, y 100 en caso contrario. Se consideran los siguientes *indicadores*: hambre, sed, suciedad, cansancio y aburrimiento.

Para poder rebajar cada uno de los *indicadores* el personaje puede realizar las siguientes acciones:

- Comer: decrementa el *indicador* de hambre en 1 unidad.
- Beber: decrementa el *indicador* de sed en 1 unidad.
- Lavarse: decrementa el *indicador* de suciedad en 1 unidad.
- Descansar: decrementa el *indicador* de cansancio en 1 unidad.
- Jugar: decrementa el *indicador* de aburrimiento en 1 unidad.

Cada acción de las anteriores además de rebajar en una unidad el *indicador* correspondiente, incrementa en 0.1 el resto de *indicadores*.

Para poder realizar las acciones anteriores el personaje requiere tener el objeto adecuado, por ejemplo, para comer debe tener comida, para descansar debe estar en una cama o un sofá, para lavarse necesita una ducha, una bañera o un lavabo, etc. Se supone que cada objeto está en una posición y el personaje tendrá que estar situado en el mismo lugar que el objeto para poder realizar la acción (puede haber varios objetos en la misma posición).

Los personajes tendrán distintas preferencias por cada uno de los objetos, que influirán en el estado emocional del personaje. Las preferencias toman valores enteros entre 0 y 10, 0 representa que el personaje detesta dicho objeto y 10 que le gusta mucho. El estado emocional se representa como el grado de infelicidad y toma valores entre 0 e infinito, 0 representa el mayor grado de felicidad y cuanto mayor sea su valor mayor será la infelicidad del personaje. Cuando un personaje realiza una acción con un objeto cuya preferencia vale p se incrementa su infelicidad en $(1 - \frac{p}{10})$.

Además de esas acciones, los personajes pueden realizar las siguientes acciones:

- Trabajar: para trabajar el personaje tiene que estar situado junto a un ordenador (tele-trabajo). El trabajo incrementa la riqueza del personaje en 10 unidades e incrementa todos los *indicadores* en 0.1.

- **Comprar:** para comprar el personaje tiene que estar situado junto al ordenador (compra por internet). Los únicos objetos que necesita comprar son la comida y la bebida puesto que el resto de objetos asumimos que están en el escenario inicialmente y no se deterioran. La comida y bebida que compran debe situarse en la nevera o despensa de la cocina y sólo puede ser consumida por el personaje que la ha comprado. Decrementa en 1 unidad la riqueza por cada objeto que compra e incrementa todos los *indicadores* en 0.1.
- **Moverse:** con esta acción los personajes pueden moverse de la posición en la que se encuentran a otra cualquiera dentro del escenario.

Se puede asumir que el estado emocional de los personajes varía de forma similar tanto al trabajar como al comprar ya que utilizan el mismo objeto, el ordenador, para realizar ambas acciones. El desplazamiento no afecta el estado emocional.

Cada vez que los personajes realizan una acción de comer o beber el objeto que utilizan desaparece del escenario. Con el resto de acciones el objeto involucrado permanece igual. Todos los objetos que hay en el escenario pueden ser usados por cualquier personaje, excepto la comida y bebida que sólo la puede consumir el personaje que la ha comprado.

Se pueden establecer dos modalidades de juego que han de definirse al principio:

- **Por tiempo:** se asume que en cada turno se incrementa el tiempo total de juego en una unidad. Cuando se agote el tiempo establecido gana el personaje cuyo estado emocional tenga el valor más bajo.
- **Por indicadores:** el juego finalizará en el momento que alguno de los *indicadores* de un personaje llega a 100. Ese personaje pierde el juego y gana el que tenga el valor del estado emocional mas bajo.

2. Se pide

1. Construir la ontología, con las clases y plantillas necesarias para modelar el conocimiento del sistema en el fichero **ontologia.clp**.
2. Contrastar la jerarquía definida con otro grupo y hacer una única jerarquía común.
3. Definir las reglas, las instancias y hechos necesarios para poder realizar el control del juego en el fichero **servidor.clp**. Las reglas que se deben incluir son aquellas que permitan establecer los distintos turnos entre los jugadores y las que determinen cuándo finaliza el juego según las dos modalidades. De forma automática el sistema debe detectar el número de instancias de personajes distintos y asignar los turnos. Los valores de las variaciones de los *indicadores*, estado emocional y riqueza debe representarse en el servidor de forma que pudieran cambiarse fácilmente.
4. Definir las reglas, instancias y hechos necesarios para simular el comportamiento de dos personajes virtuales con el mismo comportamiento (pertenecen al mismo grupo), en el fichero **cliente.clp**. La estrategia de control debe ser *random* y habrá que modelar mediante reglas el comportamiento de los personajes para que intenten ganar el juego. Por medio de reglas habrá que implementar el tipo de acción que más le conviene realizar al personaje, según el valor de los *indicadores*. Una vez que se establece el tipo de acción, habrá que diseñar las reglas para que utilicen el objeto que más felicidad les proporciona. También, habrá que diseñar algún mecanismo para que las acciones implementadas sólo las puedan ejecutar los personajes del mismo grupo para cuando se juegue con el cliente.clp implementado por otro grupo.
5. Simular 5 partidas, como mínimo, en que juegan los dos personajes variando las preferencias por los distintos objetos, el nº de objetos iniciales en el escenario y la modalidad de juego (por tiempo o por *indicadores*). Para cada jugada definir el fichero **jugada1.clp**, **jugada2.clp**, ... con dichos valores. La traza de las partidas realizadas hay que entregarla y en la memoria hacer un pequeño resumen de las pruebas y resultados obtenidos. Con el comando *dribble-on "fichero"* se consigue que todos los *printout* del programa se salven en un fichero.

6. Simular 5 partidas, como mínimo, jugando contra el *cliente.clp* del otro grupo y comprobar qué personaje de los dos grupos gana. Es decir, cargar uno de los dos servidores.clp, los dos clientes y los valores iniciales del fichero **jugadaN.clp**. Para identificar los personajes de cada grupo denominarles *sim1-A* y *sim2-A* para los del primer grupo y *sim1-B* y *sim2-B*. Además, habrá que implementar algún mecanismo para evitar que los personajes de un grupo ejecuten las acciones diseñadas por el otro grupo. La traza de las partidas realizadas hay que entregarla y en la memoria hacer un pequeño resumen de las pruebas y resultados obtenidos.

3. Recomendaciones

- No se pueden usar **ifs** ni cualquier otra sentencia de programación estructurada en las reglas.
- No se deben usar **tests** muy cargados en la parte izquierda de las reglas.
- Se debe evitar el uso de funciones y variables globales. En caso de estimar necesario su uso consultad con alguno de los profesores de prácticas.
- Procurad diseñar el sistema basado en reglas de forma que las reglas sean escuetas con un objetivo sencillo y claro, y fáciles de comprender, evitando al mismo tiempo la proliferación de reglas demasiado simples. Se deben utilizar plantillas que representen conocimiento, de forma que las reglas sean lo más generales posibles
- El agrupamiento de varias reglas en una sola utilizando OR no significa que la regla obtenida esté simplificada. Es preferible construir varias reglas porque facilita la comprensión.
- Evitad los test de igualdad que comparan dos variables. No son necesarios, basta con poner el mismo nombre de variable.
- El código deberá ser lo más legible posible, incluyendo la indentación y los comentarios adecuados.
- La documentación sobre CLIPS se puede encontrar en:
[http : //scalab.uc3m.es/ docweb/ia/transparencias.html](http://scalab.uc3m.es/docweb/ia/transparencias.html)
- El manual de referencia de CLIPS, en html:
[http : //scalab.uc3m.es/ docweb/ia/CLIPS_REFERENCIA](http://scalab.uc3m.es/docweb/ia/CLIPS_REFERENCIA)
- Las diferentes versiones de CLIPS se pueden encontrar en:
[http : //scalab.uc3m.es/ docweb/ia/software.html](http://scalab.uc3m.es/docweb/ia/software.html)

4. Entregas de la práctica

Habrà 2 entregas de esta práctica:

- Primera entrega (día **20 de noviembre**): ontología y alguna regla. Se realizará en los casilleros de los profesores de prácticas de la asignatura o en clase. Se podrán entregar un máximo de 5 páginas explicando las plantillas, clases y cualquier consideración de diseño que os parezca relevante. Se debe incluir alguna regla que haga algo necesario del comienzo de la práctica. Por ejemplo, la asignación de los distintos turnos. Haced un pequeño programa con 4 instancias de personajes donde 2 de ellos pertenezcan a un grupo y los otros 2 al otro, y las reglas que permitan de forma iterativa dar opción a cada personaje a elegir una acción para ejecutar.
- Segunda entrega (día **10 de enero**): práctica completa.

La primera entrega servirá para que los profesores os confirmen que la práctica está bien orientada.

Para la última entrega habrá que entregar una memoria en papel y los ficheros correspondientes a la práctica cuya entrega se realizará por Aula Global. **Es muy importante seguir estrictamente las normas de entrega aquí descritas.**

La memoria se entregará directamente a los profesores de prácticas o se dejará en uno de sus casilleros.

■ La memoria debe tener la siguiente estructura:

1. Portada con los nombres y direcciones de correo de los autores.
2. Introducción. Una hoja de descripción de la práctica desde vuestro punto de vista, no desde la copia del enunciado.
3. **Código y explicación de:**
 - Estructura estática de la memoria de trabajo (plantillas y jerarquía de marcos).
 - Las reglas que representan las inferencias del sistema de producción. Se debe incluir una explicación general sobre cómo están organizadas estas reglas (por ejemplo, si hay varias reglas que realizan una determinada función).
4. Experimentación: se debe incluir la explicación de los experimentos realizados tanto con los dos personajes propios como con los personajes del otro grupo y un breve resumen de los resultados obtenidos.
5. Conclusiones: resumen de lo realizado y conclusiones técnicas acerca de la práctica y el uso de sistemas de producción (ventajas e inconvenientes).
6. Comentarios personales: opinión personal.

■ Los ficheros se entregarán por Aula Global. Se entregará un único fichero comprimido (.tgz) cuyo nombre será el NIA de uno de los autores precedido por la letra *a* . Al descomprimir este fichero se deberá generar un directorio cuyo nombre debe corresponder con el NIA de uno de los autores precedido por la letra *a*. Este directorio debe contener exclusivamente los siguientes ficheros con los **nombres que se especifican**:

- Fichero con el nombre y la dirección de e-mail de los autores, **autores.txt**
- El fichero con la ontología, **ontologia.clp**
- El fichero con las reglas para controlar el juego, **servidor.clp**
- El fichero con las reglas que modelan el comportamiento de los personajes, **cliente.clp**
- Los ficheros de traza correspondientes a las 5 evaluación del sistema con los dos personajes propios, **traza1.txt, traza2.txt, ... traza5.txt** y los valores utilizados **jugada1.clp, ... jugada5.clp**
- Los ficheros de traza correspondientes a las 5 evaluación del sistema con el cliente del otro grupo, **trazaG1.txt, trazaG2.txt, ... trazaG5.txt** y los valores utilizados **jugadaG1.clp, ... jugadaG5.clp**

Ejemplo: supongamos que el NIA de uno de los autores es 100000000. Por lo tanto, los ficheros a entregar estarán en un directorio con nombre a100000000. Para generar el fichero comprimido que se debe entregar, haremos:

```
tar -czvf a100000000.tgz a100000000/
```

La práctica se debe hacer en grupos de dos personas. En ningún caso se admitirán prácticas de grupos con más de dos alumnos