Sesión 6 - Creación de un Sistema Multi Agente Competitivo

Sistemas de ayuda a la toma de decisiones

Alejandro Piedrafita Barrantes
Alejandro Ruiz Sumelzo
José Ignacio Hernández Gracia
Alberto Martínez Rodríguez
Alex Daniel Costa
Víctor Hernández Fernández
Jorge Fernández Muñoz

Universidad Zaragoza

Enero de 2021

Índice

Página

1. Descripción del problema

El problema que se plantea es realizar una sociedad de múltiples agentes que trabaje de forma competitiva. La idea se basa en la historia de la saga de películas de *Matrix*, describiendo un sistema en el que diferentes agentes (resistencia o sistema) con unos determinados pesos que influyen en las posibilidades de conseguir su objetivo interactúan con el resto de agentes en esa "lucha" por conseguirlo".

En cada enfrentamiento entre los agentes que persiguen el objetivo de ser vencedores se sumarán +-1 en su peso de posibilidades en función de si vencen o pierden el enfrentamiento. Adicionalmente, existe un agente especial (*Oráculo*), el cual cuando es encontrado por uno de los dos tipos de agente que quieren lograr vencer, les proporciona un bonus a su peso y finaliza su ejecución.

Finalmente el agente *Arquitecto* es el encargado de atender las peticiones de los dos tipos de agentes, tanto para coordinar enfrentamientos como para sus peticiones de reclutar nuevos agentes de la población (agentes *JoePublic*) y mantiene un log de la simulación de la ejecución del sistema.

2. Arquitectura del sistema

2.1. Propuesta de solución

La propuesta consiste en introducir dentro del sistema de Agentes utilizados, un Agente que realice las labores de inicialización del trabajo, haciendo uso de las funciones que permiten a un agente crear dinámicamente nuevos agentes, con el fin de que, dado que los agentes (resistencia y sistema) iniciales que buscan el objetivo de vencer en el sistema tienen comportamientos de tipo *CyclicBehaviour*, sea capaz de generar nuevos agentes a los que reclutar (*JoePublic*) y los otros agentes necesarios para procesar y coordinar la ejecución de la simulación.

Adicionalmente puede crearse otros agentes (o el mismo arquitecto) para finalizar el sistema una vez haya un vencedor y que elimine al resto de agentes.

2.2. Sociedad de agentes

2.2.1. *Agente Init*

Agente que se encarga de crear dinámicamente todos los agentes necesarios para el flujo de trabajo, especificará el número de ejecuciones para cada modelo de entrenamiento y el porcentaje de datos que utilizará para entrenar el modelo.

2.2.2. *Agentes De La Resistencia*

Inicialmente tres: *Neo*, *Morfeo* y *Triniti*. Su objetivo es doble. Por un lado, tendrán como cometido reclutar a los miembros de *Matrix* (*AgenteJoePublic*) que quieran salirse de la simulación y convertirse en agentes de la resistencia y, por otro, combatirán a los *AgenteSistema* hasta su eliminación. También tendrán como objetivo encontrar al *AgenteOraculo*, para conseguir que *Neo* aumente su bonus y tenga más posibilidades de cara al combate final.

2.2.3. *Agentes Del Sistema*

Inicialmente tres: *Smith*, *Torrente* y *Terminator*. Sus objetivos son múltiples. Por un lado, tendrán como objetivo convertir a los miembros de *Matrix* (*AgenteJoePublic*) en agentes del sistema o si no eliminarlos. Por otro, combatirán a los *AgenteResistencia*, para tratar de eliminarlos de la simulación.
Adicionalmente, tratarán de encontrar al *AgenteOraculo para eliminarlo*, quedándose su conocimiento y que así *Neo* no consiga ese bonus de probabilidad.

2.2.4. *Agentes JoePublic*

Agentes genéricos que pueden (definir probabilidad de disposición por su parte) ser convertidos a *AgenteResistencia* o *AgenteSistema* en función de quien les *reclute*. A excepción del *AgenteOráculo* que se camufla en el groso de *AgentesJoePublic*.

2.2.5. *El Oráculo*

Se oculta como un *AgenteJoePublic*, pero con un perfil especial. Proporcionará una bonificación especial a los agentes de la *Resistencia* o del *Sistema* cuando lo encuentren. En el momento en que esto ocurra, el agente se desvanecerá.

2.2.6. *El Arquitecto*

Controla la simulación de *Matrix*. Atenderá peticiones de los agentes de la *Resistencia* y del *Sistema* para encontrar nuevos agentes que podrán ser de tipo *JoePublic* o contrarios, de acuerdo a la estrategia que decidan seguir. Llevará un log de la simulación que deberá permitir elaborar resultados al final del todo.

2.2.7. *Agente Resultado*

Agente que recibe el resultado proporcionado por *AgenteArquitecto* y lo hace visible en el área de texto indicada por el frame (*JframAgenteResultado*).

2.3. Protocolo de Comunicación

El *AgenteInit* crea dinámicamente un *número indicado* de agentes *AgenteJoePublic*, los cuales se introducirán mediante parámetros; un *AgenteArquitecto*; un *AgenteResultado*; un *AgenteOraculo*; tres *AgenteResistencia* con peso 50 (*Neo*, *Morfeo* y *Triniti*) y tres *AgenteSistema* (*Smith*, *Torrente* y *Terminator*). Una vez creado el sistema inicial se quedará bloqueado esperando a crear nuevos *AgenteResistencia* o *AgenteSistema* o a un mensaje que indique que ha finalizado la simulación.

En un principio se creará el *AgenteArquitecto* que se quedará bloqueado en un estado en el que espere a mensajes de *incialización* del conjunto de *AgenteJoePublic* para mantener un registro de los que son. A continuación pasará a un estado en el que se quedará bloqueado esperando las peticiones de los agentes de la *Resistencia* y *Sistema*, tanto para emparejarlos en combate y anotar los resultados del enfrentamiento, como para reclutar a nuevos agentes de tipo *JoePublic*.

Inicialmente los *AgenteJoePublic* se quedan bloqueados esperando una petición para convertirse en *AgenteResistencia* o *AgenteSistema*. Cuando esto ocurra hay un % de probabilidad de que esto ocurra (se manda una petición al arquitecto para que cree un nuevo agente del tipo que haya sido reclutado y se elimina este agente *JoePublic*) , si no ocurre se vuelve a quedar bloqueado en ese estado de espera.

De igual manera que los agentes *JoePublic*, los *agenteSistema* y *agenteResistencia* pueden recibir en cualquier momento mensajes de combate por parte de otros agentes del tipo contrario, empezando así el protocolo de comunicación de combate entre los agentes.

El *AgenteOraculo* se *registra* como *AgenteJoePublic*, lo que implica que existe una pequeña probabilidad de que cuando un *AgenteResistencia* o *AgenteSistema* intente reclutar un *AgenteJoePublic* se encuentre con el oráculo, el cual informará a ese agente de que es el oráculo para que salga del estado de reclutamiento, y al arquitecto para que modifique el peso de Neo o Smith con ese bonus por haber sido encontrado.

El *AgenteArquitecto* puede recibir varios tipos de mensaje. Inicialmente seerán de tipo *registro* para definir el estado de quienes son *AgenteJoePublic*, *AgenteResistencia* y *AgenteSistema*. Después las peticiones que puede recibir son de emparejamiento para combate, reclutamiento de *AgenteJoePublic* e información del resultado (tanto del combate, que puede influir en si elimina algún agente; así como en el reclutamiento, que puede suponer mandar peticiones para crear nuevos agentes de tipo *Resistencia* o *Sistema*) y adicionalmente mantendrá un log con las acciones que se produzcan en la ejecución y que mandará al finalizar a un *AgenteResultado*. Adicionalmente, el Agente Arquitecto puede recibir mensajes de petición del estado del sistema por parte de otros agentes que quieran

más información para su toma de decisiones.

AgenteResultado inicialmente se quedará bloqueado a la espera de recibir los resultados pertinentes al desarrollo de la ejecución y los mostrará mediante una interfaz gráfica.
El protocolo de comunicación utilizado para los mensajes intercambiados entre los agentes en esta arquitectura será el protocolo *FIPA Request Interaction Protocol* y *FIPA Query Interaction Protocol (Ref)*.

Se ha utilizado el protocolo *FIPA Query Interaction Protocol* para realizar los emparejamientos, bien de combate o bien de reclutamiento. El agente que inicia la comunicación (Sistema o Resistencia) hace una consulta (query) al Agente Arquitecto para que este le devuelva un agente con el que posteriormente comunicarse para llevar a cabo la acción deseada. Se ha empleado este protocolo en esta acción en lugar del *FIPA Request Interaction Protocol* debido a que este protocolo ya devuelve el resultado concreto como respuesta final del intercambio de mensajes (en caso de que la comunicación haya sido correcta).

Una vez establecida la relación ("matchmaking") entre los agentes del sistema (Resistencia con Sistema, JoePublic u Oráculo , y viceversa) la comunicación se realizará entre ellos mediante mensajes de tipo REQUEST de acciones. En los cuales aceptarán o rechazarán la petición e informarán del resultado de la misma al Arquitecto.

2.4. Dinámica de funcionamiento

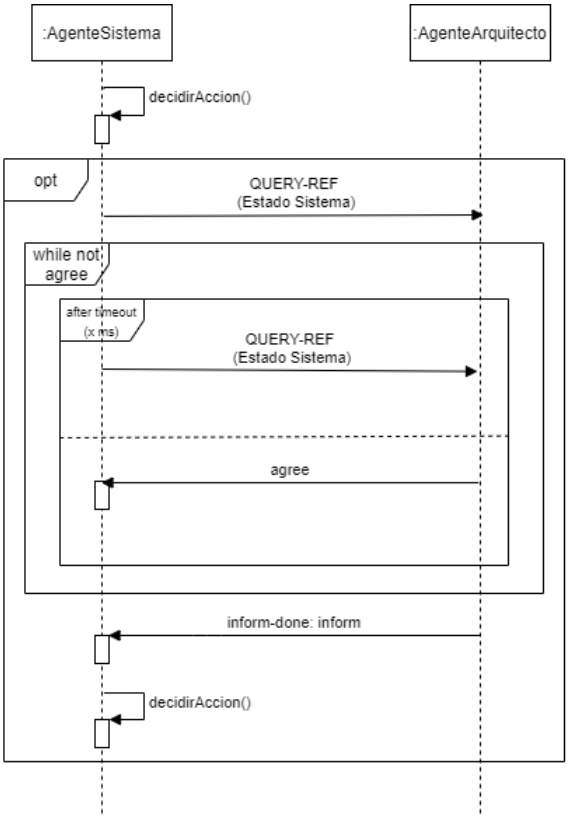

Figura 1: Diagrama de secuencia del *AgenteArquitecto*, primera parte.

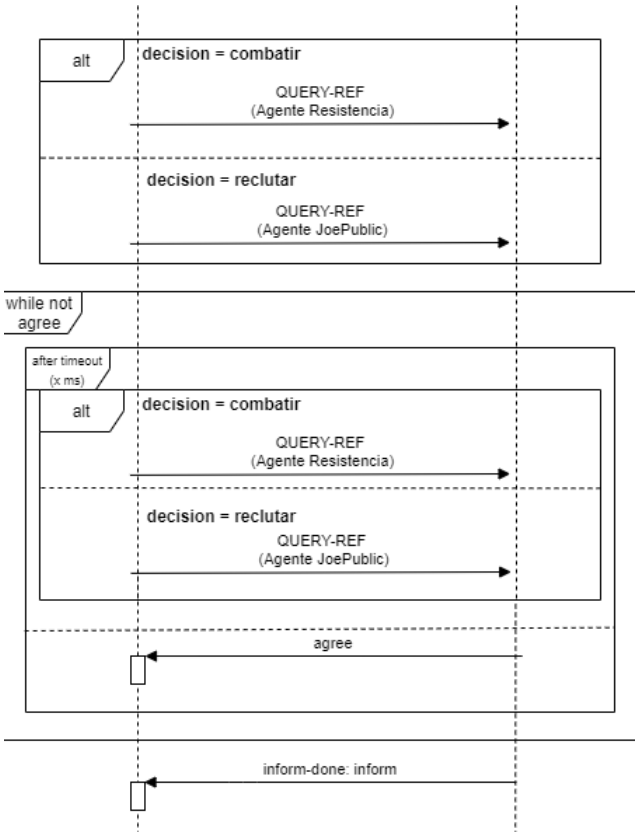

Figura 2: Diagrama de secuencia del *AgenteArquitecto*, segunda parte.

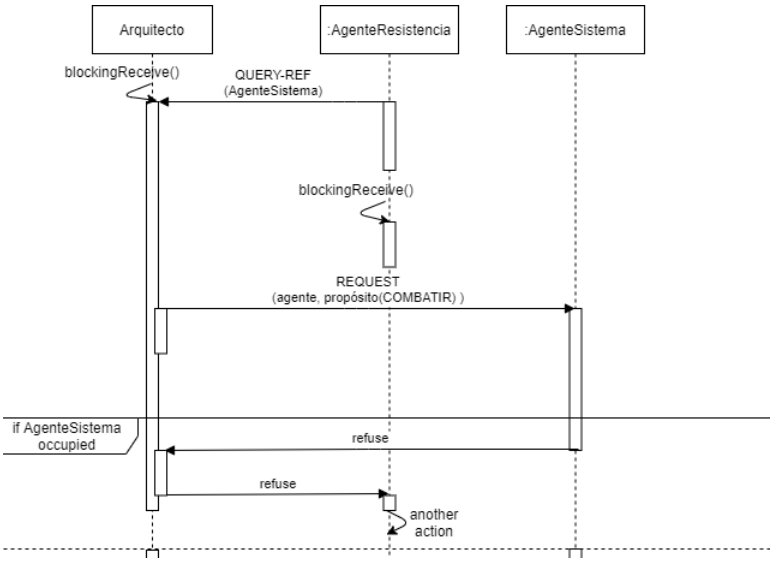

Figura 3: Diagrama de secuencia del *AgenteResistencia* en el combate.

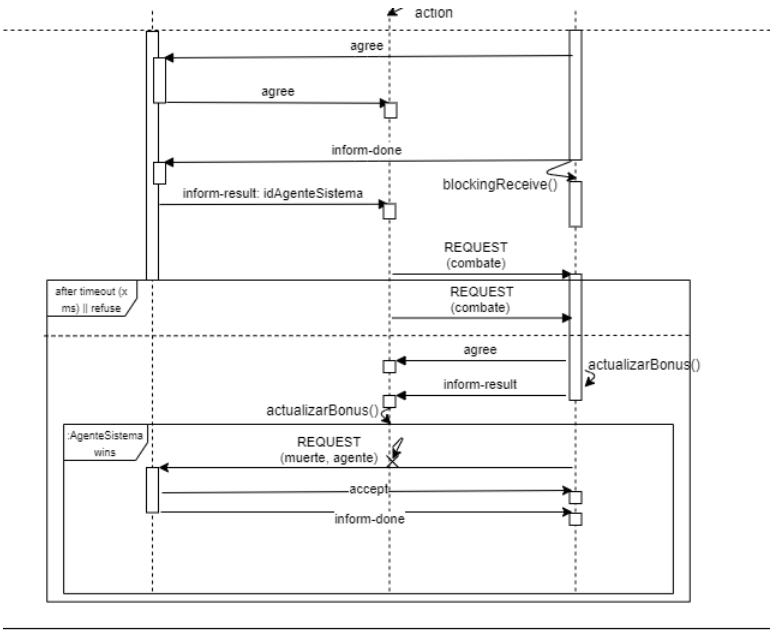

Figura 4: Diagrama de secuencia del *AgenteResistencia* en el combate, *AgenteSistema* gana el mismo.

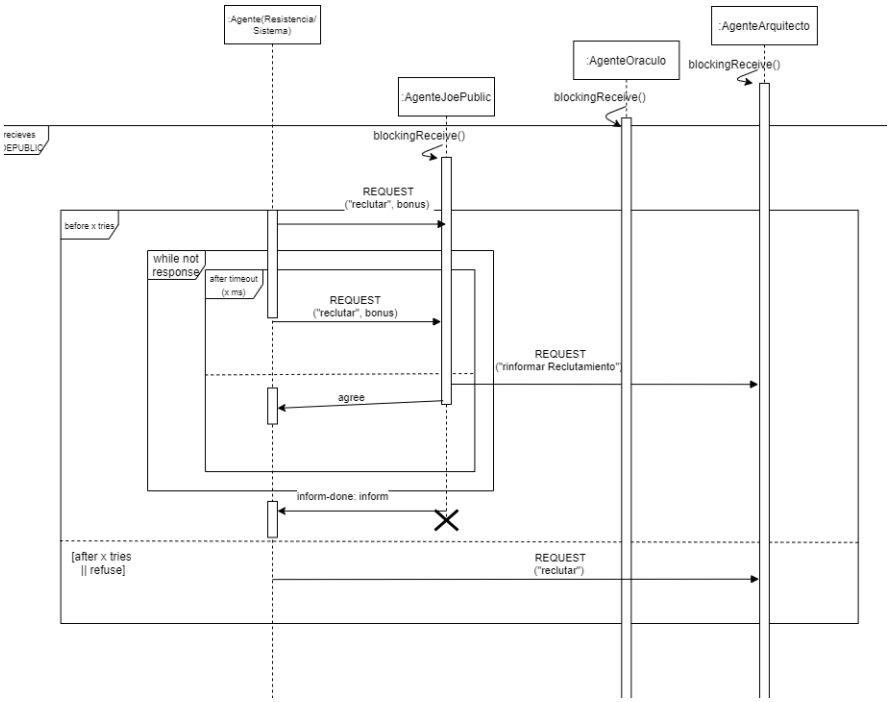

Figura 5: Diagrama de secuencia del *Reclutamiento*, primera parte.

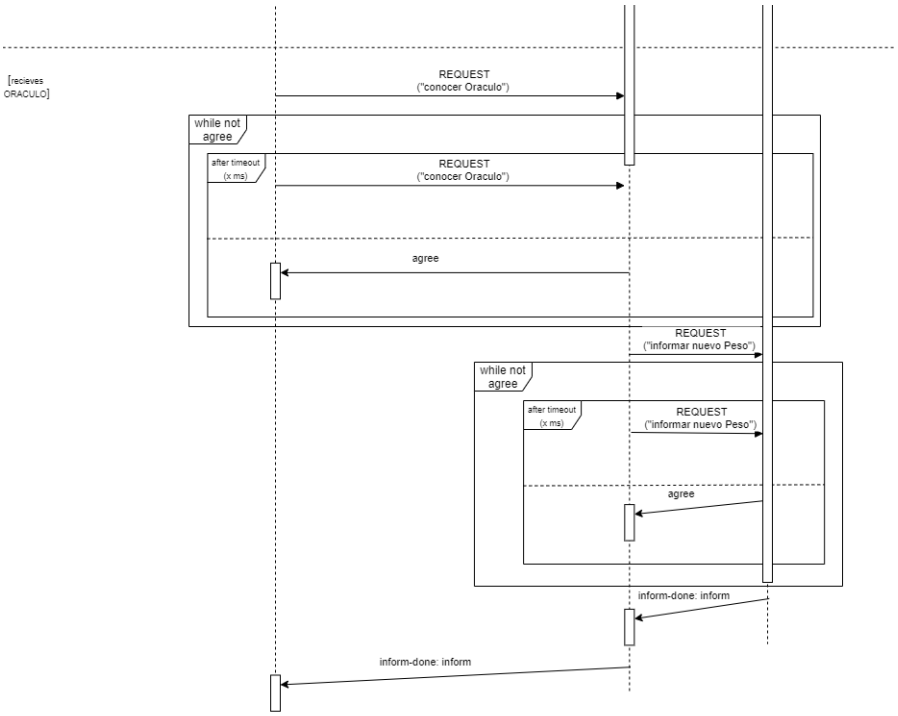

Figura 6: Diagrama de secuencia del *Reclutamiento*, segunda parte.

Anexo I. Reuniones

Fecha de reunión	Asistentes	Decisiones tomadas
16/12/2020 8:00 - 10:00 h	José Ignacio Hernández	Se ha comprendido el sistema que se propone en el trabajo. Se ha realizado una primera versión del informe y arquitectura del sistema. .
18/12/2020 9:30 - 12:30 h	José Ignacio Hernández Alberto Martínez Alex Daniel Costa Alejandro Piedrafita Alejandro Ruiz Jorge Fernández	Se especifica mediante diagramas de secuencia el protocolo de comunicación entre agentes en distintas situaciones.
21/12/2020 10:00 - 11:00 h	José Ignacio Hernández Alberto Martínez Alex Daniel Costa Alejandro Piedrafita Alejandro Ruiz Jorge Fernández Víctor Hernández	Se termina la memoria correspondiente a la segunda parte. Se actualizan los diagramas en función de lo comentando por el profesor.