

MODELADO DE EVACUACIÓN DE MULTITUDES MEDIANTE AGENTES Y TRANSCRIPCIÓN DE COMPORTAMIENTOS

ELENA NÚÑEZ GONZÁLEZ

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA. FACULTAD DE
INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin Máster en Ingeniería del software e inteligencia artificial

8 de Septiembre de 2011

Director y colaborador:

Pablo Gervás
Gonzálo Méndez

Convocatoria: SEPTIEMBRE
Calificación: SOBRESALIENTE

Autorización de difusión

Elena Núñez González

8 de Septiembre de 2011

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: MODELADO DE EVACUACIÓN DE MULTITUDES MEDIANTE AGENTES Y TRANSCRIPCIÓN DE COMPORTAMIENTOS, realizado durante el curso académico 2010-2011 bajo la dirección de Pablo Gervás [y con la colaboración externa de dirección de Gonzalo Méndez] en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Abstract

A multi-agent system that simulates the behavior of the pedestrians involved in a fire evacuation of a building has been designed and analyzed. Three types of behavior have been simulated: The leaders, who help other pedestrians to leave the stage, dependent, unable to leave on their own, and independents, able to evacuate themselves, but they sometimes decide to put their lives at risk breaking the rules of evacuation. The simulation produces a large log with the events occurred. A system has been created, FromLog2StoryTelling, which take the log, summarizes and modifies it to generate a report with the major events organized, and ready to be translated into natural language.

Keywords

Agent, behaviour, content determination, crowd, evacuation, fact, interest, log, multi-agent system, pathfinding, perception.

Resumen

Se ha diseñado y analizado un sistema multi-agente que simula el comportamiento de las personas involucradas en una evacuación de incendios en un edificio. Se han simulado tres tipos de comportamiento: El líder, que ayuda a los demás a salir del escenario, el dependiente, sin capacidad para salir por cuenta propia, y que necesita que un líder le ayude, y el independiente, con capacidad para evacuar por sí mismo, pero que en ocasiones decide poner su vida en peligro saltándose las normas de evacuación. La simulación produce un extenso log con los eventos sucedidos. Se ha creado un sistema, FromLog2StoryTelling, que trata el log, lo resume y modifica para generar un informe con los sucesos más importantes ordenados, y preparado para traducirse a lenguaje natural.

Palabras clave

Agente, comportamiento, determinación de contenido, multitud, evacuación, evento, interés, log, sistema multi-agente, búsqueda de camino, percepción.

Índice general

Índice	4
Agradecimientos	7
1. Introducción	9
2. Trabajo relacionado	13
2.1. Simulación	13
2.1.1. Simulación con agentes	15
2.1.2. Simulación de evacuaciones	17
2.1.3. Crowd Simulation	24
2.2. Generación de Lenguaje Natural	26
2.2.1. TAP, un Pipeline para Aplicaciones de Generación de Lenguaje Natural	28
3. Objetivos	31
3.1. Objetivos del sistema multi-agente	32
3.1.1. Escenario configurable	32
3.1.2. Búsqueda de caminos: pathfinding	33
3.1.3. Detección de colisiones	33
3.1.4. Roles y comportamientos	34
3.1.5. Creación de grupos y Crowd Simulation	34
3.1.6. Desobediencia	35
3.2. Objetivos del generador de informes FromLog2StoryTelling	35
4. El Simulador Multi-Agente	37
4.1. Propuesta de diseño	38
4.1.1. Modelo de Comportamiento	38
4.1.2. Interacción de los agentes con el entorno	39
4.2. Diseño del SMA	41
4.2.1. Escenario	41
4.2.2. Pathfinding	43
4.2.3. Descripción de los agentes	47
4.2.4. Detección de colisiones y Crowd Simulation	51

5. Generación de informes: La Herramienta FromLog2StoryTelling	53
5.1. Contexto del sistema y estructura del log	53
5.2. Determinador de Contenido	55
5.2.1. El problema de las agrupaciones	56
5.2.2. Interés	57
5.3. Planificador de Discurso	59
5.4. Ejemplo	60
6. Resultados	63
6.1. Varios líderes y el fuego	64
6.2. Agentes dependientes	67
6.3. Cambio de líder	68
6.4. Desobediencia	70
6.5. Muerte de un agente	72
7. Conclusiones y trabajo futuro	77
7.1. Simulador Multi Agente	77
7.1.1. Escenario configurable	77
7.1.2. Búsqueda de caminos: pathfinding	78
7.1.3. Detección de colisiones y creación de grupos	79
7.1.4. Roles y comportamientos	79
7.2. From Log To Story Telling	80
A. JADE: Java Agent DEvelopment Framework	83
A.1. Comportamientos en agentes	85
A.1.1. Tipos de comportamientos	85
A.2. Comunicación entre agentes	87
A.3. El servicio de Páginas Amarillas	88
B. Algoritmos de búsqueda. Pathfinding	89
B.1. Algoritmo de Dijkstra	89
B.2. Algoritmo A*	90
C. Intenciones y acciones	91
C.1. Intenciones y acciones	91
D. Listado de acciones y parámetros de interés	93
D.1. Parámetros de interés	95
E. Ejemplos de logs	97
E.1. Varios líderes y el fuego	97
E.2. Muerte de un agente	99

Agradecimientos

Este proyecto no habría podido ser llevado a cabo ni no fuera, en primer lugar, por el apoyo de mi familia, sobre todo de mi madre, que siempre me ha insistido para que no abandonara, y me ha ayudado en todo lo que podido. Agradezco también a mis profesores los conocimientos que me han sabido transmitir, principalmente a mi director y codirector. Pero, sobre todo, agradezco a Juanen toda su paciencia y ayuda. Sin él no habría podido conseguirlo. Gracias.

Capítulo 1

Introducción

Una de las medidas de seguridad más importantes en cualquier edificio público debe ser la gestión de las situaciones de emergencia y las medidas de evacuación. Se considera necesario proporcionar una información clara sobre las salidas de emergencia más convenientes según la situación de los individuos, y asegurarse de la eficacia de las medidas de desalojo, entendiendo si estas ayudas son eficaces para evitar, por ejemplo, cuellos de botella en las salidas.

La gestión eficaz del orden público requiere precisión en la predicción de los efectos de incidentes que se pueden dar durante una evacuación; además, la participación de vidas humanas demanda un especial cuidado [Shendarkar et al., 2006]. Resulta por ello de gran importancia asegurar una comprensión completa por parte de los encargados de evacuación de los riesgos y peligros con que se pueden enfrentar en el caso del incendio de un edificio público.

Se entiende que en este tipo de situaciones de riesgo los simulacros no son suficientemente realistas ya que no se simula la psicología humana, aunque suponga un factor importante en el desarrollo de una evacuación. Generalmente no consiguen prepararnos de forma precisa para evacuaciones en las que el peligro de vida es inmediato [Fahy and Proulx, 1997]. En artículos y estudios como los de [Kelley et al., 1965], [Keating, 1982], [Johnson, 1987], [Elliott and Smith, 1993] se trata cómo el miedo, por sí solo, puede provocar catástrofes en multitudes. Por otro lado, este tipo de simulacros tiene un alto coste económico y de tiempo.

En los edificios públicos el papel de responsable de evacuación no se lleva a cabo por profesionales en este aspecto sino por trabajadores en otras áreas que, dada una emergencia, modifican su rol habitual (celadores, secretaría, profesorado, etc) y se convierten en evacuadores. Esto quiere decir que ge-

neralmente estas personas no están devidamente entrenadas y pueden no saber cómo comportarse bajo algunas circunstancias.

Por todo esto se considera que el entrenamiento por parte de los evacuadores del comportamiento social en emergencias puede ayudar a salvar vidas humanas. Se propone la creación de:

1. Un simulador del comportamiento de multitudes involucradas en una evacuación por emergencia en un edificio. El edificio es configurable. Como ejemplo se utiliza como modelo la Facultad de Informática de la Universidad Complutense de Madrid. En la evacuación se simulan varios tipos de agentes con perfiles y roles diferentes, que dotan a la simulación de más realismo.
2. El comportamiento de la simulación se describe en un log, en el que aparece representada cada acción o cambio de estado de los agentes. A partir de él, un segundo módulo genera un informe en lenguaje natural con la información relevante de la simulación. Este módulo es de gran importancia, puesto que proporcionará al evacuador la información más interesante sobre lo sucedido en la evacuación, ayudándole a entender los resultados de la misma.

Uno de los mayores riesgos durante una evacuación es el comportamiento de los implicados: ataques de pánico, desmayos, prioridades de los individuos diferentes al de salir del edificio (por ejemplo acudir a rescatar a una persona o incluso algún bien material). También, como se describe en [Li et al., 2004], la edad y el sexo influyen en el comportamiento de un individuo para escapar del fuego, y en sus reacciones ante él. Se estudia el comportamiento social de grupos en situaciones de emergencia para dotar de un mayor realismo a la simulación.

El simulador consiste en un sistema multi-agente en el que cada individuo es un agente, con características propias, las cuales, junto al entorno, determinan su comportamiento. Se tienen en cuenta diversos factores que distinguirán un agente de otro, principalmente el rol, evacuador y evacuado, la independencia, que distingue a dependientes de independientes, y el liderazgo, que distingue a líderes de seguidores.

El sistema contará con solución gráfica, por lo que es imprescindible que el coste computacional no sea demasiado alto. Se considera, por tanto, el estudio del acuerdo entre el coste y el nivel de detalle al que se quiere llegar en el comportamiento individual. Se propone la creación de tres tipos de agentes. Los primeros, que representan a la mayoría de los implicados en la evacuación, son agentes simples, con comportamientos similares entre ellos, y que cumplen las normas de evacuación. El segundo y tercer tipo de agentes

son mucho más complejos, y toman decisiones más complicadas. El segundo corresponde a los líderes, responsables de la evacuación. El tercer grupo lo formarían los independientes, con comportamientos complejos que pueden llegar a desobedecer explícitamente las normas de evacuación

Cada acción individual de cada personaje de la simulación da lugar a una entrada de un log en el que explica sus movimientos o reacciones (*dirigirse-a*, *seguir-a*, *sufrir-accidente*). Además, el fuego también genera un texto sobre sus propiedades (situación del foco, áreas afectadas). Toda esta información genera unos registros con un elevado número de datos.

El segundo módulo del sistema recibe todos estos datos, los trata y genera resúmenes con la información relevante sobre los acontecimientos acaecidos durante la simulación. Para ello, primero filtra la información que puede ser interesante y la que no. De estos datos, agrupa aquellos que pueden ser importantes sólo por ser repetidos por muchas personas (por ejemplo, *17 personas siguen al agente Ana Martín*) y se decide el orden en que se querrá contar la historia para preservar la coherencia y que resulte interesante.

Una vez seleccionados los logs relevantes y definido el orden en que se van a contar las acciones, el sistema genera la entrada de un generador de lenguaje natural, el *Text Arranging Pipeline* (TAP). Se ha elegido este generador por su gran versatilidad y flexibilidad a la hora de definir una entrada, puesto que acepta diversos formatos de la misma. Además, es capaz de generar textos tanto en inglés como en español.

Los agentes pueden realizar distintas *Acciones*. Estas acciones pueden ser intenciones (como *Obedecer-a*, que implican cambios de estado) o acciones propiamente dichas (*Subir*, *recoger [Objeto]*). Algunas pueden tener información implícita, como el grupo de acciones del tipo *Desobedecer-acción*, que indican que realizan la acción actuando contra las normas indicadas por los evacuadores, o que, conociendo el camino de salida, el agente se aleja de él.

No sólo las personas realizan acciones, sino que también el fuego tiene comportamientos que luego se tendrán en cuenta en el relato final. Sus acciones son del tipo *Llegar-a*, *Invadir-area* etc. Todos estos eventos, y algunos más, se encuentran también reflejados en el log inicial.

Para conseguir el informe final con un resumen de las acciones realizadas durante toda la simulación se realizan varios procesamientos sobre el log creado por el sistema multi-agente.

En primer lugar, se definen los grupos. Hay acciones que si las realiza una persona de forma individual resultan irrelevantes, pero al ser ejecutadas por grupos cobran importancia. Además, el liderazgo es una cuestión a estudiar en una evacuación, puesto que divide a las personas en líderes y seguidores, cuyo criterio queda delegado en los primeros. También es interesante el estudio de la creación, disolución o fusión de grupos para comprobar cómo emerge el comportamiento grupal a partir de los comportamientos individuales.

Después se determina el interés de cada agente individual y de cada grupo en función de su interés inicial (tipo de agente, rol, liderazgo, etc.) y el interés de las acciones que haya realizado durante toda la simulación. En función del interés se eliminan las personas y grupos no relevantes. El hecho de que un agente sea interesante no quiere decir que toda su historia vaya a ser interesante. Se eliminan las acciones no relevantes de agentes relevantes. Cabe destacar que, aunque el fuego se considere un agente más a lo largo de los módulos de *Determinación de Contenido*, su interés asignado será siempre muy alto, como se verá más adelante; por lo que nunca será eliminado.

Posteriormente, una vez seleccionadas las acciones que se contarán en el informe, estas son ordenadas en el espacio y en el tiempo para, por un lado, mantener la coherencia del relato y por otra, conseguir que sea entretenido.

Una vez ordenados los eventos, se transforman en una entrada válida del TAP, la herramienta de generación de lenguaje natural, donde se genera una narración.

Capítulo 2

Trabajo relacionado

2.1. Simulación

La simulación con computador es una útil alternativa que proporciona información para evaluar el diseño, ayudar al proceso de planificación y tratar con las emergencias [Pan et al., 2006]. La simulación es un tipo particular de modelización [Gilbert and Troitzsch, 2006]. Según [Shannon, 1998] “La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema y/o evaluar nuevas estrategias — dentro de los límites impuestos por un cierto criterio o un conjunto de ellos — para el funcionamiento del sistema”. Un modelo es una simplificación (más pequeña, menos detallada, menos compleja, o todo ello al mismo tiempo) de alguna otra estructura o sistema, y por sistema nos referimos a un grupo de elementos que interactúan entre sí para lograr algún objetivo determinado [Agre, 2003].

En [Shannon, 1998] se señalan algunas ventajas de la simulación sobre los modelos analíticos y matemáticos a la hora de analizar sistemas, entre las que destacamos las siguientes. En primer lugar, el concepto básico de simulación es fácil de comprender y por lo tanto bastante fácil de justificar ante terceras personas que no estén familiarizadas con métodos de análisis. Además, un modelo de simulación puede ser más creíble porque su comportamiento ha sido comparado con el del sistema real o porque necesita menos hipótesis simplificadas y por lo tanto captura más características ciertas del sistema bajo estudio. Otro de sus puntos fuertes es su habilidad para permitirnos experimentar con situaciones no familiares, y responder a cuestiones del tipo “¿Qué pasa si...?”

Además, la simulación introduce la posibilidad de una nueva forma de pensar acerca de los procesos sociales y económicos, basada en algunas ideas acerca de la aparición del comportamiento complejo a partir de actividades

relativamente simples [Simon, 1996]. Uno de los desafíos más importantes para poder entender los fenómenos sociales reside en el estudio de la emergencia de comportamientos en las sociedades resultado de la colaboración entre sus componentes individuales. La acción colectiva de individuos sociales produce fenómenos que no se pueden explicar como la suma de los comportamientos individuales. Las propiedades del todo generado a partir de acciones simples es mayor que la suma de las propiedades individuales de dichos elementos que conforman el sistema.

Uno de los propósitos principales de la simulación es la mejor comprensión de algunas características del mundo social [Gilbert and Troitzsch, 2006]. La simulación puede aportar luz sobre algunos aspectos del comportamiento de las sociedades, como por ejemplo cómo los comportamientos individuales pueden afectar al conjunto, y cada conjunto, en último grado, a la sociedad completa.

Se ha intentado comprender la complejidad de las acciones colectivas de individuos sociales desde distintas perspectivas. Una de ellas ha sido la aplicación de diferentes metodologías basadas en ecuaciones matemáticas, pero han fallado en capturar los comportamientos emergentes [Moretti, 2002]. El principal problema de la modelización matemática es que sólo permite jugar con tendencias (probabilísticas) y no considera el comportamiento original del sujeto, pieza básica de cualquier sistema social [Pavón et al., 2006].

Sin embargo, las teorías de sistemas complejos auto-organizados ofrecen un importante marco teórico y metodológico para el estudio de sistemas dinámicos no-lineales [Sansores-Pérez, 2007]. Uno de estos enfoques que se ha adoptado para el estudio de la dinámica social son las Sociedades Artificiales (SA). Una SA es una representación sintética de una sociedad [Steels and Brooks, 1995] que puede ser modelada como un sistema multi-agente (SMA), donde agentes autónomos imitan individuos reales de la sociedad así como sus interacciones.

Un SMA se define como una red de entidades software autónomas (los agentes) que interactúan entre ellas y con su entorno. El hecho de ser autónomos significa que los agentes son entidades activas que pueden tomar sus propias decisiones. Un agente decide si realiza o no una operación solicitada, para lo cual tendrá en cuenta sus objetivos y prioridades, así como el contexto en que crea encontrarse.

Puede entenderse, por tanto, que el paradigma de agentes se asimila bastante bien a lo que es un sistema social [Pavón et al., 2006] y facilita el modelado de los aspectos organizativos y de comportamiento de los individuos de una sociedad. Un agente puede representar un individuo en una sociedad, que percibe y reacciona ante los eventos de su entorno de acuerdo

a su estado mental [Russell and Norvig, 2004].

2.1.1. Simulación con agentes

El concepto de agente

Revisando la documentación existente se puede ver que no existe una definición precisa de lo que es un agente, aunque pueden establecerse criterios que permitan distinguir lo que es un agente de lo que no.

El término agente viene del latín “agere” que significa hacer. Agente deriva del participio “agens”. Expresa la capacidad de acción o de actuación de una entidad. [Mas, 2005]

Según [Huhns and Singh, 1998] el término agente se usa habitualmente para describir aquellos programas autónomos que pueden controlar las propias acciones basándose en sus percepciones de su entorno operativo.

Según [Russell and Norvig, 2004] un agente es cualquier cosa capaz de percibir el entorno con ayuda de sensores y actuar en ese medio con ayuda de actuadores. La secuencia de percepciones de un agente refleja el historial de percepciones que ha recibido. Un agente tomará una decisión en un momento dado dependiendo de la secuencia completa de percepciones hasta ese instante.

Una de las definiciones más citadas es la de [Wooldridge, 1997]: “Un agente es un sistema informático situado en un entorno y que es capaz de realizar acciones de forma autónoma para conseguir sus objetivos de diseño”. Aunque esta definición identifica algunas de las características de un agente, como la autonomía, no permite distinguirlo claramente de un sistema distribuido convencional. [Mas, 2005]

[Wooldridge and Jennings, 1995] destacan que los agentes informáticos típicamente tienen las siguientes propiedades:

- *Autonomía*. Los agentes operan sin que otros tengan control directo de sus acciones y de su estado interno.
- *Habilidad social*. Los agentes interactúan con otros agentes.
- *Reactividad*. Los agentes pueden percibir su entorno y pueden responder al mismo.

- *Proactividad*. Los agentes también son capaces de tomar la iniciativa, involucrándose en un comportamiento dirigido a un objetivo propio.

Aunque no necesariamente deben tenerlas todas. Además, a los agentes se les suele atribuir un cierto grado de intencionalidad.

Sin embargo, estas características no son muy útiles para diseñar un agente, y además un agente no tiene por qué poseer necesariamente todas estas características. Una forma más útil de describir agentes es decir que tienen las siguientes características: [Gilbert, 2008]

1. *Percepción*. Pueden percibir el entorno, posiblemente incluyendo la presencia de otros agentes de su alrededor. En términos de programación, esto significa que los agentes tienen los medios para determinar qué objetos y agentes están localizados en su vecindad.
2. *Representación*. Tienen un conjunto de comportamientos que son capaces de representar. Con frecuencia, dentro se incluyen:
 - a) *Movimiento*. Se pueden mover dentro de un espacio (ambiente).
 - b) *Comunicación*. Pueden enviar mensajes a y recibir mensajes de otros agentes.
 - c) *Acción*. Pueden interactuar con el entorno, por ejemplo, recogiendo “comida”.
3. *Memoria*. Tienen una memoria, que almacena sus percepciones de anteriores estados y acciones.
4. *Políticas*. Tienen un conjunto de reglas, heurísticas, o estrategias que determinan, dada su situación actual y su historia, qué comportamientos no deben llevar a cabo.

Hoy en día existen muchas organizaciones que se encuentran definiendo los estándares para el desarrollo de agentes inteligentes tales como SMA (Sistemas Multi Agentes), OMG (Object Management Group), KSE (Knowledge Sharing Effort) y FIPA (Foundation for Intelligent Physical Agents). De éstas, la que más aceptación ha tenido por parte de la comunidad investigadora y desarrolladora es FIPA.

Agentes como sistemas intencionales

Cuando explicamos el comportamiento humano, solemos realizar afirmaciones del tipo “Ana cogió su paraguas porque *creía* que iba a llover”, o “Javier se levantó muy temprano porque *quería* llegar pronto al trabajo”. Estas afirmaciones utilizan una *filosofía popular* [Wooldridge and Jennings, 1995],

en la que el comportamiento humano se predice y explica mediante la atribución de *actitudes*, tales como deseos, creencias, esperanzas, miedos, etc.

El filósofo Daniel Dennett en [Dennett, 1987] ha acuñado el término *sistema intencional* para describir entidades “cuyo comportamiento puede ser predicho-predecido atribuyéndoles creencias, deseos y perspicacia racional”.

Mc Arthy, entre otros, ha argumentado las ocasiones en que una *postura intencional* (*intentional state*) es apropiada. Según [McCarthy, 1978]:

“La atribución de creencias, libre albedrío, intenciones, conciencia, habilidad o necesidades a una máquina es *legítimo* cuando la adscripción expresa la misma información sobre la máquina que la que expresa sobre una persona. Es *útil* cuando la adscripción nos ayuda a entender la estructura de la máquina, su comportamiento pasado o futuro, o cómo repararlo o mejorarlo. Quizás nunca sea *lógicamente necesario* ni siquiera en humanos, pero expresar de forma razonablemente breve lo que se sabe en realidad sobre el estado de la máquina en una situación particular puede requerir cualidades mentales o cualidades isomórficas a ellas. Teorías de creencias, conocimiento y necesidades pueden ser construidas en un escenario más simple que para humanos, y luego aplicarlo a humanos. La adscripción de cualidades mentales es *más sencillo* para máquinas de estructura conocida tales como un termostato o un sistema operativo, pero es *más útil* cuando se aplica a entidades cuya estructura es conocida incompletamente.”

2.1.2. Simulación de evacuaciones

En esta sección se analizan sistemas de evacuación existentes en la actualidad así como estudios sobre el riesgo debido a ataques de pánico y otras situaciones provocadas en multitudes cuando estalla una emergencia de cualquier tipo. A continuación se detallan algunos sistemas que se han considerado representativos. Los dos primeros se basan principalmente en la importancia de tener en cuenta ciertos comportamientos en una simulación de evacuaciones (pánico y liderazgo), y los siguientes consisten en sistemas de simulación con sistemas multi-agente, con diferentes enfoques respecto al comportamiento de los agentes.

En [Helbing et al., 2000] se estudian los efectos del pánico en una evacuación, factor de gran importancia en el sistema que se pretende desarrollar; y señalan las siguientes características en escapadas de pánico:

1. La gente se mueve o intenta moverse más rápido de lo normal.
2. Los individuos empiezan a empujarse. La interacción entre la gente es física.
3. Los desplazamientos y, en particular, los pasos por cuellos de botella se vuelven descoordinados.
4. A la salida del edificio se suelen observar arcos y obstrucciones.
5. Se crean atascos.
6. Las interacciones físicas en multitudes atascadas aumentan causando presiones peligrosas.
7. La evacuación es más lenta por caídos o heridos que se convierten en ‘obstáculos’.
8. Se comprueba entre la gente una tendencia al comportamiento de masa, es decir, a hacer lo que hagan los demás.
9. Las salidas alternativas frecuentemente se pasan por alto, o no son utilizadas eficientemente en las evacuaciones.

Estos datos son muy interesantes para la simulación, puesto que determina algunos de los resultados esperados en la misma.

Los autores destacan el peligro de los comportamientos de pánico en concentraciones de personas. En el estudio se pueden encontrar simulaciones de peatones en diferentes escenarios (habitaciones o pasillos) en los que los individuos se comportan de diferentes formas: ordenados, con pánico y estampidas¹. También se estudia el comportamiento de manada, en el que la masa imita lo que hacen unos pocos individuos (líderes).

Aunque su objeto de estudio resulte interesante para este trabajo, luego su desarrollo se aleja de los objetivos del mismo. En primer lugar, utilizan un modelo físico para la simulación, no con agentes, lo que limita la definición de los comportamientos. Por ejemplo, un individuo no entra en pánico porque sea asustadizo, o sea sometido a mucha presión durante un tiempo, ni siquiera porque perciba una situación de emergencia, sino que desde el inicio hasta el fin de la simulación se encontrará con pánico. Por otro lado, los escenarios se limitan a una habitación o, al máximo, a dos habitaciones comunicadas. Por lo tanto el problema de la búsqueda del camino de salida no se trata. Tampoco existen líderes, sólo evacuados. Esto es debido a la limitación de objetivos del sistema de simulación a la demostración de que el pánico es peligroso a la hora de realizar una evacuación.

¹En <http://angel.elte.hu/~panic> se pueden ver varios videos de estas simulaciones.

H. Li, W. Tang y D. Simpson en [Li et al., 2004] describen un sistema de simulación del procedimiento de evacuación de edificios en llamas. Introdúcen la posibilidad de la indiferencia por parte de los agentes teniendo en cuenta dos tipos de comportamiento diferente en los individuos: comportamiento general, en el que se siguen las normativas de evacuación, y los comportamientos que ignoran estas normativas. Para ello se han realizado estudios en los que se distinguen los procedimientos de evacuación en función de aspectos físicos y psicológicos, como el consumo de tabaco, la situación en la ruta de evacuación o el miedo. También se ha tenido en cuenta la edad y el sexo de los individuos. El resultado indica, por ejemplo, que el género y la edad de los individuos tienen efectos en las decisiones de ignorar el fuego, recuperar objetos personales y enfrentarse al fuego. Añadiendo estos factores aumentan el realismo del modelo.

Identifican también tres etapas distintivas de la actividad humana en situaciones de emergencia por fuego: reconocimiento, cuando alguien encuentra el fuego o suena la alarma; acción, cuando ese agente contacta con otros; y evacuación, en el que se realizan los procedimientos de evacuación. Estos estados permiten diferenciar claramente el tipo de comportamiento que tendrá un agente en cada instante de la simulación; y al definirse mediante diagramas de estado, su implementación es muy sencilla. Es por tanto un buen recurso a incluir en el sistema propuesto.

Sin embargo estas diferencias de comportamiento se quedaron en el estudio previo, ya que, a la hora de construir un prototipo, la posibilidad de ignorar las normas de seguridad fue obviada, y sólo se definió un tipo de comportamiento. Básicamente, los agentes finalmente han sido definidos para seguir las normas generales de evacuación. Aunque justifican esta falta aduciendo la limitada información psicológica sobre el comportamiento humano en situaciones de emergencia, esto no es del todo cierto, como se puede ver en [Helbing et al., 2000].

Respecto a los escenarios, son relativamente complejos, como se puede ver en la Figura 2.1. Para la búsqueda de camino, aunque se habla de la posibilidad de un algoritmo dinámico, se utiliza el algoritmo A* estático.

Ubik [Serrano et al., 2009] es una infraestructura creada para estudiar aplicaciones de inteligencia ambiental (AmI) en las que participen un gran número de usuarios. Se trata de una simulación multi-agente (MABS, *Multi-Agent Based Simulation*) programada en MASON². Ha sido desarrollado en la Universidad de Murcia.

Estudia la compatibilidad de los agentes y el espacio en la asistencia de emergencias. El edificio es configurable, por lo que el usuario puede simular virtualmente muchos escenarios, definiendo el número de pisos, las escaleras, ascensores, pasillos, habitaciones, etc.

²Website de MASON: <http://cs.gmu.edu/eclab/projects/mason/>

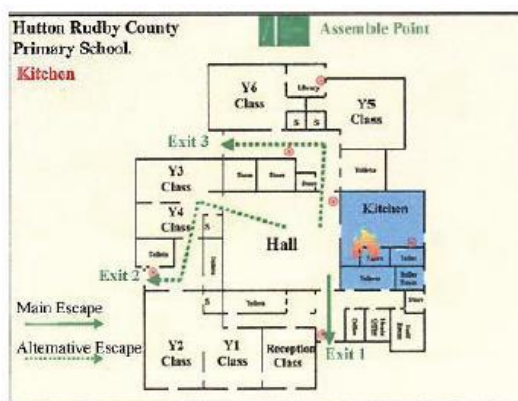


Figura 2.1: Entorno Virtual utilizado en el prototipo de H. Li.

El espacio modelado en Ubik es tratado como un entorno compuesto por edificios de oficinas donde la actividad principal es laboral. Cada trabajador es un agente, y antes de que empiece la emergencia cada agente se comporta en función de objetivos (permanecer en su lugar de trabajo, visitar a otro trabajador, salir a la calle, citarse entre ellos, etc.). Además, hay una forma física (*fitness*) asociada a cada trabajador, que determina la velocidad de movimiento. Pero a la hora de empezar la evacuación, todos los agentes tienen el mismo comportamiento, ya que la simulación no modela humanos con un punto de vista cognitivo, sino reactivo; es decir, se considera cada comportamiento humano como el de miembros de un grupo de otros humanos que tienen los mismos comportamientos frente a los eventos de interés. Además no se considera la posibilidad de que una parte de los trabajadores tengan intereses que superen el miedo al peligro, como, por ejemplo, la intención de rescatar a un amigo que se encuentre cercano al fuego, o de recuperar el computador. Por otra parte, como se ha explicado en secciones anteriores, el miedo implica un factor importante en la evacuación, puesto que es un motivo de fracaso en la seguridad de la misma. La persona que sufre un ataque de pánico se bloquea, quedando impedido para reaccionar con la lógica que es necesaria para dirigirse a la salida de forma calmada y segura.

Sin embargo, esta infraestructura es muy eficiente desde el punto de vista computacional, dada su habilidad para operar con un gran número de agentes en un espacio muy amplio. Ubik ha apoyado estudios con simulaciones de más de 200.000 agentes y 200 pisos eficientemente. Es evidente que con el sistema planteado en este artículo, al aumentar la complejidad de cada agente no se podrán realizar simulaciones a ese nivel, pero al destinarse a edificios más pequeños con no tantos agentes esta eficiencia no será necesaria.

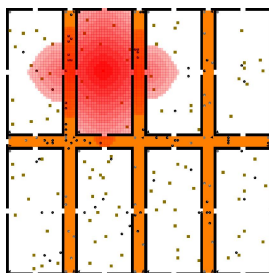


Figura 2.2: Captura de una simulación con Ubik en un escenario con ocho habitaciones.

En [Murakami et al., 2002] el estudio se basa en la demostración de que un SMA puede reproducir la interacción entre líderes y evacuados en una emergencia. Para ello realizan un SMA utilizando FlatWalk³, y lo comparan con un experimento real de control de una situación simple creada por [Sugiman and Misumi, 1988]. En el experimento de Sugiman se establece un entorno simple con pocos sujetos para determinar la efectividad de 2 métodos: “*Follow-Direction Method*” y “*Follow-Me Method*”. En el primero, el líder grita las instrucciones de evacuación y se dirige hacia la salida. En el segundo, el líder le dice a los evacuados más cercanos que le sigan, y se dirige a la salida sin verbalizar la dirección. Sugiman utilizó a estudiantes como evacuados y monitorizó el progreso de la evacuación con diferentes números de líderes. Con sus resultados justificaba la importancia del papel de líderes en evacuaciones, y determinaba la proporción óptima de los mismos en función del número de evacuados y del método de evacuación.

Para la simulación describen los dos tipos de comportamientos, el de los líderes y el de los evacuados, y realizan la simulación en un escenario igual al del experimento, comprobando los resultados mediante comparación con los de Sugiman. Tras esta comparación, realizan los ajustes que ven necesarios después de una observación más detallada del experimento para acercar los resultados.

El comportamiento de los agentes se define mediante reglas, que cambian en función del rol del agente, y si es líder, del tipo de líder que sea dicho agente.

La simulación obtenida es demasiado simple, con un escenario básico (ver Figura 2.1.2). A partir de la simulación no hay posibilidad de predicción, uno de los objetivos del proyecto por el que se realiza el estudio, puesto que se ha adaptado para conseguir unos resultados específicos. Además, sólo incluye dos roles, y dentro de ellos el comportamiento de los individuos es el mismo.

³FlatWalk es un simulador Multi-Agente basado en Scheme, un dialecto de Lisp inventado por Guy Lewis Steele Jr. y Gerald Jay Sussman.

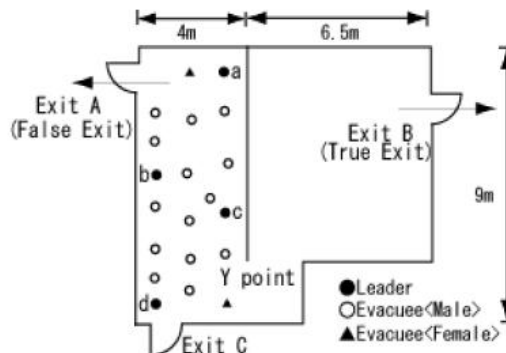


Figura 2.3: Plano del suelo del lugar del experimento de Sugiman y posición inicial de los sujetos.

En [Shendarkar et al., 2006] se propone el uso de Realidad Virtual mediante agentes capacitados con BDI (*Belief, Desire, Intention*) extendido para simular una evacuación por un ataque terrorista (el estallido de una bomba) en la Zona Centro de Washington DC. Antes de la explosión la gente está distribuída discretamente en diferentes áreas de la ciudad en función de sus objetivos (compras, trabajo, etc.). Después de la explosión la gente intenta escapar tan rápidamente como puede mediante las salidas disponibles (cada agente cambia el objetivo inicial). Algunas personas se convierten en líderes de un grupo y guían al resto del grupo por el camino de salida más corto. Otros se mueven de intersección en intersección hasta que encuentran un policía que les señala el camino de salida. Las características de cada agente son sexo, edad, conocimiento de la zona, escala de pánico, liderazgo, independencia, escala de lesiones (cuando es 1 es muerte, 0 es saludable), posición actual, etc.

El comportamiento de los agentes se representa con un diagrama de estados, donde los objetivos serían los estados de aceptación. El escenario se representa internamente como un grafo. El camino de salida más cercano se calcula con el algoritmo estático de Dijkstra. Ésto es mejorable con un algoritmo dinámico.

Dotan a cada agente de características individuales para construir la multitud. Por ejemplo, a la hora de evacuar un hombre fuerte puede elegir un camino más corto pero más peligroso, mientras que un hombre viejo elige uno con menos tráfico, aunque sea más largo. Al dotar a los agentes del marco BDI su complejidad interna crece, aumentando asimismo el coste computacional del sistema, tanto más cuantos más agentes se encuentren en la simulación. Los resultados son estadísticos y no se representa la simulación gráficamente, lo que aumentaría considerablemente el coste computacional. Aunque en el sistema propuesto se pretenda dotar a los agentes de carac-

terísticas individuales, se debe alcanzar un acuerdo entre coste y nivel de complejidad de cada agente. Con la arquitectura BDI el coste supera lo admisible para un sistema que se muestre gráficamente.

En [Tsai et al., 2011] se describe el sistema ESCAPES, un simulador multi-agente de evacuación del Aeropuerto Internacional de los Ángeles. Dichos agentes se basan en la arquitectura BDI (*Believes, Desires, Intentions*) en el que cada persona es un agente. Cada agente tiene acceso a un subconjunto de los 14 comportamientos disponibles. El comportamiento se realiza en función de un esquema de probabilidades. Tsai simula el comportamiento de las unidades familiares, en el que la actitud general de salvarse ante todo se sustituye por la de salvar a la familia. Al empezar la emergencia el comportamiento de cada miembro de una unidad familiar es encontrar a los otros miembros, y después, salir del edificio. Los otros dos roles son los viajeros independientes y los encargados de la seguridad. Los agentes tienen una serie de características propias tales como la salud, el conocimiento del entorno y el miedo, que determina la velocidad.

El escenario es fijo, no configurable. No se simula la posibilidad de desobediencia, ni el comportamiento de grupo. Tampoco trata la búsqueda de camino (*pathfinding*). Además, en el tipo de simulaciones que queremos realizar al no existir familias, no resulta útil esta aproximación, que basa su complejidad principalmente en los comportamientos asociados con las unidades familiares.

En [Barros et al., 2004] se describe PetroSim, un sistema para la simulación de evacuaciones en zonas urbanas exteriores. El modelo de comportamiento se basa en máquinas de estados finitos, en el que se definen las intenciones, que a su vez se traducen en varias acciones de bajo nivel. Simulan dos tipos de agentes, dependientes y líderes. Cada agente simulado es una entidad autónoma descrita como un triplete de información: el conocimiento del agente del entorno, el estado actual del agente y sus intenciones. Es un sistema muy complejo que no cuenta con simular más de 100 individuos. La implementación del movimiento de grupos se realiza siguiendo el concepto de manada descrito por Reynolds en [Reynolds, 1999], como se ve más adelante en la sección 2.1.3. No trata la opción de que los agentes no obedezcan las normas de evacuación. Además, el código es cerrado, y no hay artículos en los que se explique algo más detalladamente el diseño.

En resumen, en este apartado se han analizado sistemas en los que:

- En los dos primeros sistemas, aunque se trata la importancia de los comportamientos individuales, a la hora de la construcción de prototipos estos comportamientos no son tomados en cuenta, por lo que en el sistema final todos los individuos se comportan de la misma forma.

- En **Ubik** tampoco se tienen en cuenta los distintos roles que puede haber entre los implicados en una evacuación, pero el escenario ya es mucho más complejo, con cientos de pisos y habitaciones.
- En el trabajo de Murakami se tienen en cuenta dos roles diferentes y por lo tanto dos tipos de comportamiento, pero el nivel de individualidad es muy bajo. Por otro lado, el escenario es demasiado simple, y los agentes muy pocos.
- El sistema de Shendarkar se basa en los comportamientos individuales, diferentes unos de otros, y con líderes y evacuados. El escenario es relativamente complejo, y la búsqueda de la salida estática. El mayor problema encontrado en este sistema es la alta complejidad computacional, debida al nivel de inteligencia con que se ha dotado a los agentes.
- Tsai describe un SMA simulador de evacuaciones con arquitectura BDI y varios comportamientos, pero los principales no son aprovechables para el tipo de simulación que se desea (sitios tales como una universidad). No se simulan los comportamientos de grupo, ni los comportamientos desobedientes. La búsqueda de camino no se tiene en cuenta.
- PetroSim es un sistema bastante completo con dos tipos de comportamientos, de líderes y de dependientes. Se ha tomado en cuenta para el diseño del sistema, como se ve en la sección 4.1. No tiene búsqueda dinámica de camino, y el código y las especificaciones de diseño son cerradas.

En la Tabla 2.1 se muestra un esquema con las características encontradas. Cabe destacar que en ningún caso se ha desarrollado la posibilidad de que algunos individuos no obedezcan a las normas de evacuación, aunque se haya contemplado a nivel de estudio en [Li et al., 2004].

2.1.3. Crowd Simulation

Idealmente, un modelo de comportamiento para multitudes debería ser tanto realista como eficiente, esto es, debería ser posible simular multitudes con un alto número de agentes de forma lo suficientemente eficiente como para causar el mínimo impacto en el *framerate* [Thalmann and Musse, 2007]. Sin embargo, en la práctica debe haber un compromiso entre realismo y eficiencia.

La planificación de movimiento para entidades múltiples o una multitud es un desafío en los entornos virtuales actuales [Foudil et al., 2009].

Sistema	Es SMA	Inteligencia	Escenario	Coste
[Helbing et al., 2000]	NO	↑ Estudio, ↓ Desarrollo. No hay roles	Simple ⇒ Pathfinding no tratado	Baja complejidad, Bajo coste
[Li et al., 2004]	SI	Estudio: 2 comportam. Desarrollo: 1 comp.	Complejo Pathf: A*	Agentes simples, Bajo coste
[Serrano et al., 2009]	SI	1 Comport.	Muy complejo Path. estático	Agentes simples, Bajo Coste
[Murakami et al., 2002]	SI	2 Roles: líderes y evacuados	Líderes conocen el camino Evacuados siguen al líder Path. estático	Simul. muy simple Bajo coste
[Shendarkar et al., 2006]	SI (BDI)	SI: Comp. individuales	Path. estático: Dijkstra	Alta complejidad, Alto coste
[Tsai et al., 2011]	SI (BDI)	SI: Comp. individuales	No bien descrito. Pathfinding no tratado	Alta complejidad, Alto coste
[Barros et al., 2004]	SI	SI: Comp. individuales	Pathfinding no tratado	Alta complejidad, alto coste

Cuadro 2.1: Características principales de los sistemas de simulación de evacuaciones analizados.

Una multitud no es sólo un grupo de muchos individuos. En el modelado de multitudes participar problemas que surgen sólo al centrarse en grupos. Por ejemplo, los problemas de evitar colisiones cuando hay un alto número de individuos en el mismo sitio necesita distintas estrategias, en comparación con los métodos utilizados para evitar las colisiones entre individuos.[Musse and Thalmann, 2000] [Pelechano and Badler, 2006]. En esta sección se estudian algunos métodos propuestos anteriormente para manejar la simulación de multitudes (en inglés, *crowd simulation*).

La simulación del comportamiento ha sido estudiado desde los comienzos de la investigación en *Computer Graphics*. Los primeros trabajos se concentraban en comportamientos animales, con los pájaros como opción más popular, aunque actualmente hay mucho trabajo sobre el comportamiento humano. En la comunidad de entornos virtuales el enfoque más común para simular el movimiento de grupos es el concepto de **manada** (del inglés, *flocking*). Este concepto fue introducido por Reynolds en [Reynolds, 1987]. Describe el comportamiento de las unidades de un grupo utilizando reglas locales para dichas unidades. Después, Reynolds extendió la técnica para incluir comportamiento reactivo autónomo, como se describe en [Reynolds, 1999]. La idea consiste en que cada unidad se dirige a sí misma de forma que evita colisiones con otras unidades en el entorno, mientras que al mismo tiempo intenta alinearse con otras unidades y estar cerca de ellas⁴. Un sistema que utiliza una implementación del movimiento basado en comportamiento de manada de Reynolds es Petrosim [Barros et al., 2004], un sistema para la simulación de evacuaciones en zonas urbanas.

Otra técnica utilizada es la de **Grid Searching**. Se divide el entorno en un grid y se busca con un enfoque A* el camino libre mientras que evitan

⁴<http://www.red3d.com/cwr/steer/>.

colisiones. El problema es que lleva a las unidades a atascos que sólo se resuelven con movimientos antinaturales. [Russell and Norvig, 2004].

En [Reif and Wang, 1995] se define la técnica llamada *Social Potential Field Technique*, en la que se definen campos de fuerzas potenciales entre las unidades del grupo. El comportamiento deseado se consigue asignando los campos de fuerza correctos. Como sólo se tiene en cuenta la información local, se repite el problema de la creación de atascos.

Utilizando la técnica anterior, en [Kamphuis and Overmars, 2004] se define un algoritmo multifase, para un grupo de unidades coherente. En una primera fase se calcula el camino para un rectángulo deformable, que representa la forma del grupo. En la segunda fase se calcula mediante *Social Potential Field Technique* el movimiento interno, y en una tercera fase los caminos locales y globales se combinan para dar lugar a un movimiento completo. El problema principal de esta técnica es el comportamiento antinatural cuando un grupo entra o sale de un pasillo o área estrecha.

Lo que propone [Foudil et al., 2009] es un algoritmo en tres fases: en la primera, se discretiza el entorno. En la segunda, cada agente realiza una búsqueda de camino (*pathfinding*) estático A* hasta el límite de lo que ve el agente. Cuando llegue a este límite, vuelve a realizar la búsqueda del camino. En la tercera fase, que consiste en el movimiento dentro del área discretizada, el agente comprueba la posibilidad de colisión, y la evita. Si para evitar la colisión el agente se aleja demasiado del camino, con A* dinámico se recalcula el camino. Los problemas de esta técnica es que no se crean grupos, por lo que no se evitan las colisiones entre grupos.

En [Helbing et al., 2001] se muestra un estudio realizado en base a estudios anteriores y evaluando vídeos de multitudes de personas y se ha encontrado que, en la apariencia caótica del movimiento individual de una persona en la multitud, se pueden encontrar regularidades, entre las que se destacan: Las personas muestran una fuerte aversión a tomar desvíos o a moverse desviándose de la dirección deseada, incluso si la ruta elegida está llena de gente; y las personas mantienen una cierta distancia de otras personas y bordes (de calles, muros y obstáculos). Esta distancia es más corta cuanto más prisa tenga la persona, y también disminuye con la densidad creciente de la gente.

2.2. Generación de Lenguaje Natural

La GENERACIÓN DE LENGUAJE NATURAL (GLN) es un área de investigación proveniente de la Inteligencia Artificial y de la Lingüística Compu-

tacional, y está orientada al estudio de sistemas capaces de generar textos de forma automática en un idioma humano, como el español o el inglés. Típicamente, la GLN parte de una representación estructurada no lingüística, y construye textos como informes, historias, partes de diálogos, mensajes de ayuda, etc.

Los primeros trabajos relacionados con la GLN comenzaron entre los años cincuenta y sesenta del siglo XX. Básicamente eran primeras investigaciones dedicadas al estudio de la traducción automática. No fueron hasta los años setenta las primeras aportaciones que realmente separaron la GLN de la CLN (COMPRESIÓN DEL LENGUAJE NATURAL, rama orientada a recibir mensajes en un idioma natural, de parte de un humano, y traducirlos a estructuras que pueden ser interpretadas o “entendidas” por una máquina), con las investigaciones de [Goldman, 1975, Davey, 1978]. En estos trabajos se puso de manifiesto que, en contra de lo que algunas corrientes científicas postulaban, la GLN no podía ser tratada como el proceso inverso de la CLN, puesto que estas dos disciplinas tienen problemas y características no complementarias o inversas.

Ya en los años ochenta, la investigación sobre la GLN experimentó un avance notable. Los trabajos doctorales de [McKeown, 1985, Ramsay, 1988] por ejemplo, ejercieron una influencia notable en las técnicas e ideas que después iban a ser desarrolladas. Aparecieron los primeros congresos sobre GLN (International Workshop on Natural Language Generation, 1983), y cobró fuerza la tendencia de crear sistemas particulares sobre áreas más específicas de la GLN, dejando atrás los intentos de desarrollo de grandes sistemas de programación monolíticos que pretendían resolver muchos problemas de una manera acoplada.

Reiter, en [Reiter, 1994], investigando algunas aplicaciones orientadas a sistemas de generación de Lenguaje Natural, a pesar de los distintos trasfondos teóricos, encontró e identificó una estructura muy similar con relación a los módulos en que se divide el proceso de generación, el desarrollo computacional de estos módulos y la fórmula en que éstos interactúan. Estos sistemas se dividen en módulos que se conectan con una arquitectura en *pipeline* y sin *feedback* (marcha atrás). Estos módulos son:

Planificador de Contenido, que mapea el contenido inicial a una forma semántica, anotado la mayoría de las veces con relaciones teóricas, y determina qué información se debería comunicar al oyente o receptor y ordena la información de una manera coherente retóricamente;

Sentence Planning, cuyo objetivo es la conversión de la representación semántica en una representación lingüística abstracta que especifica *content words* (palabras o grupos de palabras que se agrupan en un lexicón del lenguaje, es decir, con un significado) y relaciones gramaticales. Mapea los

conceptos del dominio y las relaciones entre las *content words* y relaciones gramaticales.

Realizador Superficial, que posee el conocimiento sobre la forma de expresar en inglés (o en el idioma de salida) las relaciones gramaticales. Recibe como entrada la especificación abstracta de la información y produce una forma de comunicar esta información.

Efectivamente éste es el esquema que siguen la mayoría de generadores de lenguaje natural, como [León et al., 2007] o TAP⁵.

En [León et al., 2007] y [Hassan et al., 2007] se describe un sistema que opera sobre logs generados artificialmente por un simulador social generado automáticamente por un sistema multi-agente. El simulador emula la vida real de un gran número de agentes durante un cierto periodo de tiempo, y proporciona un log de eventos de dichos agentes. La herramienta de generación de lenguaje natural se ha diseñado siguiendo un enfoque basado en reglas. Se ha implementado el módulo de *Planificador de Contenido*, que cuenta con el submódulo de *Determinador de Contenido*, que filtra los hechos contando sólo los que se puedan considerar interesantes para el lector, y *Planificador de Oraciones*, que ordena los hechos filtrados de forma que el receptor perciba una historia coherente. También cuenta con un módulo de *Realizador Superficial* sencillo basado en plantillas para la creación del tiempo final, para mostrar de forma legible para los humanos el contenido final. A continuación se describe más detalladamente la estructura de su sistema:

El submódulo *Determinador de Contenido* toma cada individuo como un todo, y analiza el conjunto de agentes dependiendo de sus reglas configuradas. Estas reglas, dependientes del contexto, son definidas para elegir correctamente los individuos representativos.

Planificador de Contenido utiliza plantillas (*templates*) para darle orden lógico y coherente al texto que ha sido filtrado en la etapa anterior. Las plantillas tienen secuencias simples como orden temporal de los hechos y descripciones de las relaciones más importantes.

2.2.1. TAP, un Pipeline para Aplicaciones de Generación de Lenguaje Natural

TAP (Text Arranging Pipeline) [Gervás, 2007] es una arquitectura software para la generación automática de texto. TAP está formado por una serie de interfaces que definen una funcionalidad genérica para un pipeline de tareas orientadas a la generación de lenguaje natural, desde una entrada

⁵TAP: Text Arranging Pipeline, proyecto en desarrollo por el grupo NIL de la Universidad Complutense de Madrid, explicado en la sección 2.2.1.

conceptual inicial hasta la realización superficial final en forma de texto, con etapas intermedias de planificación de contenido y planificación de oraciones. A partir de la entrada conceptual que se va a transformar en texto, los diversos módulos de TAP trabajan sobre representaciones intermedias para almacenar resultados parciales de manera que progresivamente se filtra, agrupa y enriquece la entrada hasta conseguir estructuras cada vez más cercanas al lenguaje natural tanto en estructura como en contenido.

La filosofía seguida en el diseño de TAP es identificar la naturaleza estructural genérica de la información que requiere de procesamiento en las diferentes fases de la generación de lenguaje natural, y proporcionar interfaces para acceder a esta estructura. La arquitectura de TAP ha sido específicamente diseñada para permitir el desarrollo de un conjunto reutilizable de componentes software capaces de resolver tareas básicas de GLN, trabajando con información codificada en formatos estándar, y proporcionando textos adecuados a diferentes tareas en diferentes dominios.

La arquitectura del TAP se compone de tres módulos básicos: el *Planificador de Contenido*, el *Planificador de Oraciones*, y un *Realizador Superficial*. Estos módulos están organizados como un pipeline donde la información pasa secuencialmente por los módulos que se encargan de las distintas tareas del proceso.

Planificador de Contenido

La implementación del módulo de Determinación de Contenido de TAP recibe como entrada un conjunto de referentes, un conjunto de propiedades, un conjunto de relaciones y un conjunto de eventos. Las propiedades pueden aplicarse tanto a referentes (en cuyo caso suelen acabar representadas como adjetivos) como a eventos (en cuyo caso acaban representadas como adverbios). Las relaciones pueden establecerse indistintamente entre cualesquiera de los cuatro elementos posibles (referentes, propiedades, relaciones o eventos), de manera que permiten tanto la representación de modificadores de frases nominales, complementos de oraciones, o de relaciones de causalidad.

Dentro del módulo de Planificador de Contenido, la tarea básica que se lleva a cabo es agrupar los eventos en secuencias de eventos contiguos en el tiempo y que ocurren en el mismo lugar. Estos bloques se consideran como una primera aproximación al concepto de escena, y se utilizan para establecer una planificación inicial del discurso. Las escenas se ordenan según el orden cronológico de los primeros eventos de cada una. Esta primera aproximación del discurso a generar debe enriquecerse con información adicional relativa a las descripciones de los referentes que participan en cada evento.

La información que se introduzca en este punto aparecerá en el texto final mencionada explícitamente como parte del discurso en oraciones independientes como *Ana se dirige hacia el pasillo*.

Capítulo 3

Objetivos

En este capítulo se analizan los objetivos que se pretenden alcanzar con este proyecto, desglosándolos en dos conjuntos principales. El primero consiste en la creación de un sistema multi-agente (SMA) de una evacuación de emergencia. Se desea que el sistema sea lo suficientemente realista para ayudar a los responsables de una evacuación a prevenir y controlar los principales riesgos de una evacuación. Se pretende que los evacuadores se sientan familiarizados con la mayoría de los riesgos que se puedan dar en la evacuación y así puedan prepararse debidamente para minimizar dichos riesgos. Entre estos se pueden destacar:

- El pánico, que hace perder el control de aquél que lo sufre y que, además, es contagioso.
- Personas irresponsables, que no cumplen las normas de evacuación y no siguen a los responsables de evacuación hacia la salida, poniendo en peligro su salud y en ocasiones, la de otras personas.
- Cuellos de botella y otros problemas del comportamiento social, tal como la tendencia de las personas a dirigirse hacia la salida por el camino más corto, aunque éste sea más peligroso o esté más abarrotado.

El segundo conjunto de objetivos son abarcados con la generación de un sistema capaz de recibir un informe detallado de la simulación en forma de log y transformarlo en un informe redactado en lenguaje natural y cuyo contenido sea fácilmente comprensible, y que explique los hechos más importantes de la simulación evitando las acciones irrelevantes. De esta forma los eventos, acciones, deseos etc. que hayan tenido lugar durante la simulación se pueden consultar de forma fácil en un informe, y los evacuadores podrían conocer mejor los peligros de las evacuaciones aunque no hayan observado la simulación mientras se realizaba.

A continuación se comentan más profundamente los objetivos que se quieren alcanzar.

3.1. Objetivos del sistema multi-agente

Como argumenta Shannon en [Shannon, 1998], los simuladores proporcionan la capacidad de experimentar situaciones no conocidas, por lo que resulta un tipo de sistema idóneo para aprender sobre los riesgos de las evacuaciones.

Para simular de forma realista cómo interactúan los humanos en un contexto social específico, consideramos necesario modelar el tipo de relación que tienen y especificar cómo afecta a su comportamiento interpersonal. Los actores responden tanto a los impulsos externos (fuego, órdenes de los responsables de evacuación, etc.) como a las características internas de los individuos (miedo, capacidad de liderazgo, independencia). El paradigma de los agentes se asimila bastante bien a lo que es un sistema social [Pavón et al., 2006], por lo que se creará un sistema multi-agente que modele este tipo de relaciones.

En un SMA en donde agentes autónomos imitan a individuos reales se puede simular la **emergencia** de comportamientos, donde el comportamiento de un grupo no se puede describir mediante el comportamiento de sus miembros, sino que se transforma en una entidad con características y comportamientos propios.

Resulta evidente que la simulación será más realista cuantos más detalles tengan los agentes, más diferenciados estén, y más compleja sea la búsqueda de caminos. Pero para ser más amigable se ha decidido que tenga una salida gráfica. Además, para que la simulación se pueda realizar en cualquier entorno se necesita que el sistema sea medianamente ligero. Por tanto, para que la simulación resulte de utilidad se quiere conseguir el máximo realismo con el mínimo coste computacional.

Los objetivos concretos que se quieren obtener con el simulador son:

3.1.1. Escenario configurable

Para que la simulación proporcione más información sobre las evacuaciones se quiere que el escenario no sea único, sino que se pueda configurar. El sistema aceptará cualquier mapa en blanco y negro o color, y a partir de ahí generará el escenario sobre el que se moverán los agentes y se realizará toda la evacuación. Y no solamente los escenarios deben ser configurables, también las salidas, de forma que se pudiera estudiar el resultado de las evacuaciones cuando, por ejemplo, una salida no existe, o si se habilitara una salida particular, y los responsables de la evacuación la conocieran. Para ello se creará un mapeo de la imagen de entrada, que se debe trans-

formar en un objeto que entienda el entorno, de forma que se conozcan las coordenadas con obstáculos, libres, con salidas, con otros agentes, etc.

De esta forma el sistema será útil para estudiar las evacuaciones de centros de estudios, empresas, bloques de oficinas, y en general de cualquier edificio público.

3.1.2. Búsqueda de caminos: pathfinding

Para la búsqueda de caminos hay que enfrentarse a dos problemas. El primero de ellos es la búsqueda del camino a la salida en rasgos generales, que equivaldría, en una situación real, a pensar “Para salir a la calle primero tengo que atravesar esta habitación, salir al pasillo, girar a la derecha, continuar hasta encontrarme una puerta a mano izquierda, y salir por allí”. Es decir, el agente debe conocer el camino de sectores a la salida, pero sin saber todavía las coordenadas exactas por las que se va a mover, o si hay una mesa que le haga desviarse y girar hacia la derecha dentro de una habitación. Esta búsqueda se adapta bastante bien a lo que obrece un algoritmo estático.

En segundo lugar, el agente debe dirigirse al final de cada sector, entrar en el siguiente y así, poco a poco, llegar a la salida. A este nivel ya debe moverse, y por tanto, conocer las coordenadas exactas hacia las que se va a dirigir. Debe evitar las colisiones, tanto con obstáculos permanentes, (paredes, mesas, etc.) como móviles (otros agentes). El mayor problema al que nos enfrentamos aquí son los obstáculos que se mueven, de los cuales podemos saber en qué punto se encuentran en un momento dado, pero no dónde estarán después de x instantes. Se quiere encontrar un método de búsqueda de camino que esté preparado para enfrentarse a estos problemas, y que, junto con los métodos que se utilicen para evitar las colisiones, genere un movimiento natural en los agentes.

3.1.3. Detección de colisiones

La detección de colisiones es un problema íntimamente ligado con la búsqueda de camino entre sectores, como se explica anteriormente en el apartado 3.1.2.

Las colisiones más simples de evitar son las de obstáculos estáticos, como las paredes, puesto que se dispondrán en unas determinadas coordenadas al principio de la simulación, cuando se dibuja el mapa, y continuarán allí mientras dure la simulación. Por tanto se puede calcular cómo evitar las colisiones con estos objetos en cualquier momento de la búsqueda de camino. Sin embargo, con los obstáculos móviles el problema aumenta en complejidad,

puesto que un agente dado puede saber en el momento t qué coordenadas de su sector están ocupadas por otros agentes, pero no cuáles lo estarán en el instante $t+1$. Se requiere, por tanto, algo de predicción. Así ocurre en la vida real, si una persona está en movimiento delante nuestro y mirando hacia nosotros, suponemos que podríamos colisionar, así que nos movemos hacia la izquierda o hacia la derecha, pero, muchas veces, esa persona piensa lo mismo, de forma que moviéndonos los dos hacia el mismo lado, volvemos a estar en riesgo de colisión.

Se quiere implementar un algoritmo de detección de colisiones que evite siempre las colisiones con objetos estáticos, y que posea una predicción lo bastante buena como para evitar en la mayoría de los casos las colisiones con obstáculos móviles. Además de predecirlos, se deben generar nuevos movimientos que resulten bastante realistas.

3.1.4. Roles y comportamientos

Como se ha visto en el capítulo 2, en una evacuación por emergencia nos podemos encontrar con muchas situaciones, y muchas de ellas están relacionadas con el comportamiento que puedan tener las personas implicadas en dicha evacuación. Los líderes de la evacuación cumplen un papel fundamental, puesto que ayudan al resto de los implicados a llegar a la salida de una forma segura. La gente que no es capaz de llegar a la salida por sí misma es la más abundante, y en ocasiones, pueden provocar accidentes. También hay en las evacuaciones gente que por un motivo u otro no sigue las normas de evacuación y en un momento dado se puede dirigir a zonas peligrosas, poniendo en riesgo su vida y, a veces, la de más gente. Y por último, algunas personas, sin necesidad de ayuda, pero también sin ayudar a los demás, son capaces de dirigirse a la salida. Se quiere simular todo este tipo de gente en el sistema, para conseguir resultados más exactos de las simulaciones.

Por otro lado no todas las personas que pertenezcan a un cierto rol se comportan igual en una simulación sino que muchos factores determinan el comportamiento [Li et al., 2004], tales como la edad, la salud, la valentía... Se quiere tener en cuenta estos factores para tomar distintos comportamientos durante la simulación en función de las características propias de cada individuo, aunque los eventos externos sean los mismos.

3.1.5. Creación de grupos y Crowd Simulation

Cuando un líder guía a los dependientes hacia la salida, éstos le siguen, formando un grupo. El grupo va aumentando y disminuyendo, y todos sus componentes se mueven en la misma dirección. Al estar rodeados unos agentes de otros, la detección de colisiones es diferente. Cada agente debe tener

cuidado para, por un lado, no chocar demasiado con los demás agentes y, por otro, para no alejarse demasiado del grupo.

Los grupos resultan difícilmente gestionables cuando deben pasar por puertas, o entradas pequeñas, puesto que se forman cuellos de botella y es posible que los agentes quedaran bloqueados. Con las técnicas habituales de detección de colisiones es posible que no se pueda sortear el problema.

Se quiere, por tanto, diseñar un mecanismo que trate el movimiento de las personas (agentes) dentro de un grupo teniendo en cuenta las características del mismo, y que trate sus particularidades.

3.1.6. Desobediencia

Como hemos visto en la sección 2.1.2, no se ha encontrado ningún SMA que simule a la gente que ignora las normas de evacuación, y que no sigue a los responsables de la misma. Se ha considerado este tipo de gente importante para aumentar la similitud entre una evacuación real y una simulada mediante un SMA. Se ha considerado que, además, los sistemas multi agente son ideales para simular este tipo de comportamientos, puesto que los agentes toman decisiones sobre sus acciones en función de sus intereses y objetivos.

Se quiere por tanto simular agentes “independientes”, que realicen acciones dentro de la simulación al margen de los que se le ordene. El tipo de acciones que puede realizar son: dirigirse a una zona que se salga de la ruta de evacuación para recoger algún objeto, quedarse donde se encontraba al comenzar la emergencia negándose a salir, acercarse al fuego, etc.

3.2. Objetivos del generador de informes From-Log2StoryTelling

El sistema multi-agente genera una gran cantidad de información, que se guarda en un fichero de log. En él, principalmente se guardan todas las acciones e intenciones de cada agente en un momento dado (*dirigirse-a, entrar-en, buscar-dependientes*, etc.). Un log de este tipo puede resultar tedioso, e incluso complejo, de descifrar. Constará de mucha información inútil que puede eclipsar la información realmente importante. Además, como se verá más adelante, puede haber información que sea útil por realizarse por muchos agentes a la vez, y no por la acción en sí.

Teniendo en cuenta estas cuestiones, y algunas más, se quiere, en primer

lugar, lograr un resumen del log que contenga todos las acciones y hechos en general que puedan resultar significativos. Se trata de quedarse con los datos que una persona “contaría”. Después todos esos hechos se deben ordenar para que la historia sea coherente, puesto que las acciones tienen lugares en muchos lugares a la vez.

Esa salida se tratará para adaptarla a la entrada de la herramienta TAP. Esta herramienta se explica en la sección 2.2.1. TAP producirá un texto en lenguaje natural que cuenta de forma ordenada y amena cómo se ha desarrollado la simulación.

Capítulo 4

El Simulador Multi-Agente

La mayoría de simulaciones de emergencia tipo modelan las multitudes como grupos de gente con características y/o objetivos comunes, como se estudia en [Shendarkar et al., 2006]. Como ejemplo se puede tomar [Serrano et al., 2009], que dota a todos los agentes de características comunes. Sin embargo, las características propias de cada individuo son determinantes para el desenlace de una evacuación. La primera característica importante es la que define a los evacuadores o encargados de evacuación, que toman el rol de líderes. Pero un líder no se determina sólo por esto, sino que alguien con un mayor conocimiento del escenario, o simplemente más seguro de sí mismo puede adoptar ese papel en un momento dado. Aunque algunas simulaciones sí diferencian estos roles, como en [Shendarkar et al., 2006], aún son pocos los que lo hacen.

Por otro lado, unos factores de riesgo a tener en cuenta en una evacuación son los creados por factores psicológicos de los implicados en la misma. Generalmente en las simulaciones de situaciones de emergencia los individuos siguen las normas de desalojo estrictamente ([Shendarkar et al., 2006], [Murakami et al., 2002],[Serrano et al., 2009]), pero en el mundo real esto no sucede así: ataques de pánico, empujones, gente que sigue el camino contrario al indicado. . .son peligros añadidos que pueden determinar el resultado de la evacuación; y que, por desgracia, se encuentran presentes en muchas de las evacuaciones de emergencia que suceden en la vida real¹.

Tanto para diseñar un plan de evacuación, mejorarlo, o entrenar a los evacuadores, resulta de gran utilidad el conocimiento de ciertas situaciones que se pueden crear debido a estos comportamientos. Por otro lado, es interesante estudiar el impacto de las reacciones personales de los individuos pertenecientes a una multitud ante el incendio del edificio en que se encuentra como sistema social con propiedades emergentes.

¹Ver <http://angel.elte.hu/~panic> principales desastres hasta 2003.

Se propone un sistema multi-agente que simula una evacuación por emergencia de un edificio público. Para el desarrollo del sistema se estudia el trabajo relacionado con la simulación del comportamiento humano. Se considera el comportamiento de la multitud como la suma de los comportamientos individuales, más comportamientos definibles como “de grupo”, por lo que se estudia el trabajo relacionado con la simulación del movimiento de multitudes (crowd simulation).

Un segundo punto a tratar es la búsqueda de camino. Al tratarse la simulación de una evacuación, los agentes deben encontrar el camino de salida. Esto se realiza con algoritmos de búsqueda de camino o *pathfinding*. Se estudia el tipo de algoritmo, estático o dinámico, y su efectividad.

Por último, se pretende llegar a un compromiso entre complejidad y coste. Por un lado, se quiere dotar a los agentes de bastante inteligencia, diferenciándolos unos de otros según características personales; pero por otro, al pretender incluir en el sistema una solución gráfica y tratarse de una simulación en tiempo real, el coste no debe ser excesivo. Se analiza por tanto el compromiso al que se ha llegado en los sistemas investigados.

En este capítulo se explica el funcionamiento de un sistema de simulación de evacuaciones adaptable a distintos escenarios para facilitar la comprensión, prevención y minimización de riesgos creados durante una emergencia; en el que se contemplan los protocolos de evacuación del edificio y las reacciones personales provocadas en los agentes involucrados tanto a dichos protocolos como al riesgo en sí mismo. Se ha desarrollado como sistema multi-agente (SMA), donde agentes autónomos imitan individuos reales de la sociedad, así como sus interacciones.

4.1. Propuesta de diseño

4.1.1. Modelo de Comportamiento

El modelo de comportamiento de los agentes se basa parcialmente en el modelo descrito en [Barros et al., 2004].

Cada persona simulada corresponde a un agente. Cada agente esta determinado por un triplo de información: CONOCIMIENTO, ESTADO e INTENCIONES.

El CONOCIMIENTO se compone de la información computada en tiempo real u obtenida del entorno:

- **Información ambiental** obtenida del entorno. Incluye la localización de zonas importantes y posibles caminos.
- **Percepciones.** Se computan en tiempo real y consiste en la posición y estado de los agentes cercanos y de las regiones afectadas por el incendio que, por tanto, deben ser evitadas. Todo lo que se encuentre dentro de un *radio de percepción* determinado puede ser percibido.

El ESTADO se compone por la información predefinida sobre el agente que puede ir cambiando durante el desarrollo de la simulación. Será parte del estado:

- **Perfil del agente.** Puede ser:
 1. *Independiente*, capaz de salir del edificio por su cuenta
 2. *Líderes*, que intentan ayudar a otros agentes. Son los responsables de la evacuación, más algunos evacuados.
 3. *Dependientes*, que no pueden caminar distancias largas sin la ayuda de un líder.
- **Salud.** Indica el daño sufrido por el agente hasta el momento. 1 simboliza salud completa, 0 la muerte.
- **Situación.** Puede ser *Vida-normal*, antes del inicio de la emergencia, *Riesgo* durante la emergencia y *Seguridad* una vez terminado el desalojo.

Una INTENCIÓN es un objetivo de alto nivel que un agente intenta alcanzar. Se genera mediante una Máquina de Estados Finitos (en inglés, *Finite State Machine*, FSM) en función del conocimiento y del estatus. Las intenciones tienen su correspondencia en acciones de más bajo nivel: **movimientos** e **interacciones**. Los movimientos especifican la locomoción desde la localización actual del agente a otra. Las interacciones especifican cualquier tipo de comportamiento que involucre a otros agentes (o a ciertos objetos). Cada intención puede corresponderse con una o más tuplas `<movimiento, interaccion>`. En el caso de corresponderse con varias tuplas, se seleccionará una de manera aleatoria para que las simulaciones varíen. Un ejemplo de intención es `Dejar dependientes en lugar seguro`, que se correspondería con la acción `<ir-a-salida,dejar>`.

4.1.2. Interacción de los agentes con el entorno

El escenario inicial de la evacuación es la Facultad de Informática de la Universidad Complutense de Madrid. Inicialmente se simula la evacuación

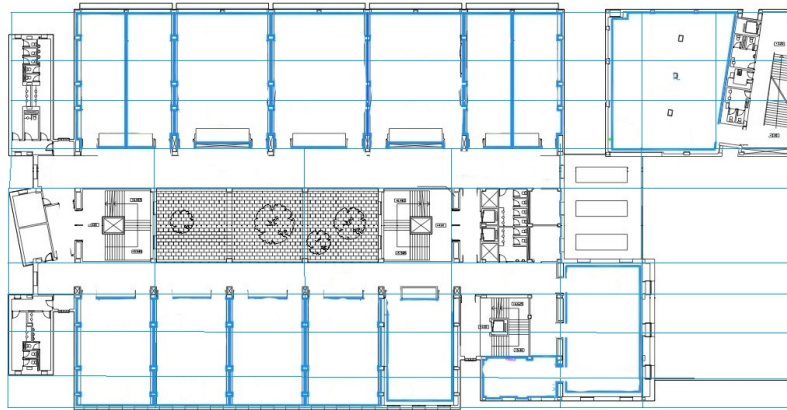


Figura 4.1: Mapa dividido por sectores

de la primera planta de dicha facultad. El escenario está dividido en 61 sectores (Fig. 4.1), de forma que cada personaje tenga acceso a la información de 2 de ellos, aquél en el que se encuentra y al que se dirige. La información completa del mapa la tiene el ENTORNO, y le pasa mediante mensajes la información requerida a los agentes.

Cada sector recibido por un agente contiene la siguiente información:

- Obstáculos fijos, tales como paredes, mesas, sillas, etc.
- Resto de agentes que se encuentran en el mismo sector. De cada agente se obtiene su posición (que puede ir variando), estado e identificador (para permitir la comunicación).

Cuando un agente *Líder* o *Independiente* entra en un estado de riesgo, le pide al ENTORNO el camino más rápido de salida (o de peligro, en caso de que el agente vaya a intentar salvar a otros agentes). El ENTORNO lo calcula mediante el algoritmo de Dijkstra, y le devuelve al agente los identificadores de los sectores por los que debe ir. Dado que el ENTORNO conoce las secciones afectadas por el accidente, éstas se omitirán en el cálculo de recorrido. Una vez obtenido el camino de salida, el agente pide la información del sector en que se encuentra actualmente y del siguiente al que debe acudir. Una vez obtenido, se construye un submapa en el que sólo se encuentren los cuadrantes pertenecientes a la ruta, y mediante A* se calcula el recorrido exacto hasta la salida. Está ruta se modifica únicamente para evitar las colisiones con otros agentes.

Cuando un agente *Dependiente* se encuentra en una situación de peligro, busca en el sector en que se encuentre algún *Líder* o algún grupo, y se une a aquél que posea un mayor liderazgo. Si en el sector no hay líderes ni

grupos, pide al ENTORNO la información del siguiente cuadrante y se dirige a él, repitiendo la operación. Cuando un *Dependiente* que pertenezca a un grupo y esté siguiendo a un *Líder* se encuentre con otro cuyo liderazgo sea mucho mayor, se cambiará de líder, siguiendo al nuevo y dejando de seguir al anterior. Esto modificará los niveles de liderazgo de los líderes.

4.2. Diseño del SMA

Esta sección describe el funcionamiento del SMA, estudiando algunos de los problemas que se han encontrado, y cómo han sido resueltos.

4.2.1. Escenario

El escenario donde se produce la simulación es configurable. El sistema recibe como entrada un fichero JPG con el mapa en color o blanco y negro, y a partir de él genera un escenario virtual en el que se desarrolla la simulación. Para generar el escenario se realiza un mapeo píxel a píxel del fichero en el que se va reconociendo el color de cada uno. Se consideran los colores claros como posiciones libres, y los colores oscuros como obstáculos permanentes (paredes, mobiliario, etc.). Para conseguir modelados realistas se asume un grado de relajación sobre los colores asumiendo que la zona libre de los mapas no tiene por qué ser completamente blanca, sobre todo cuando se trata de imágenes de baja calidad. Por ello se marcan como libres las posiciones del mapa con colores muy claros.

A continuación se crean las **celdas**, que constan de varios píxeles. El tamaño de cada celda es configurable, y correspondería con el tamaño aproximado de una persona. Si alguno de los píxeles de la celda se corresponde con un obstáculo, la celda se considera ocupada por un *Obstáculo Permanente*. Si no, se considera *Libre*.

Una ventaja importante de este sistema sobre la mayoría de los estudiados ([Tsai et al., 2011], [Barros et al., 2004], [Murakami et al., 2002], [Shendarkar et al., 2006], etc.) es que no es necesario el modelado manual del escenario, sino que se realiza un tratamiento automático del mismo. Para la evaluación se ha añadido el plano de la primera planta de la Facultad de Informática de la Universidad Complutense de Madrid. El plano se puede observar en 4.2.

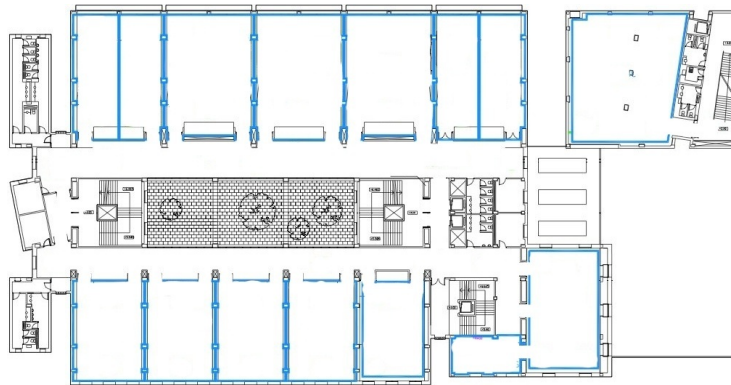


Figura 4.2: Mapa utilizado en la simulación

Sectores

Una vez que se ha creado el mapa virtual, éste se divide en sectores o **cuadrantes**, como se ve en 4.3. La división en cuadrantes se especifica mediante un fichero con las coordenadas de cada uno, junto con las adyacencias. Los agentes tienen acceso al entorno situado en el que cuadrante en el que se encuentren en cada momento. Consiste en su “radio de percepción” para la búsqueda de líderes, la detección de colisiones, etc. Las adyacencias se utilizan para los algoritmos de búsqueda de camino, como se verá más adelante en 4.2.2

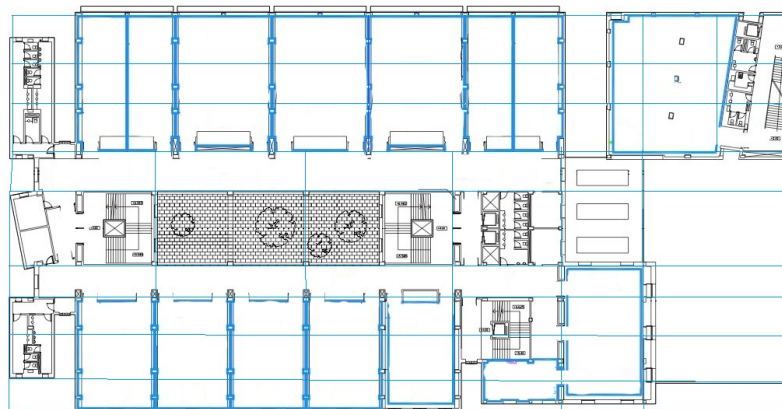


Figura 4.3: Mapa con los sectores divididos por líneas azules

Como se ve en la imagen 4.1, cada aula consta de 3 sectores, y los pasillos han sido divididos en 4. Estas opciones pueden ser configurables mediante

fichero.

4.2.2. Pathfinding

Durante la simulación se realizan principalmente dos tipos de búsquedas de camino. La primera de ellas devuelve el camino de sectores desde la posición del agente que lo solicita hasta la salida que se considere más cercana. La búsqueda se realiza sobre el mapa completo, por lo que se ha decidido realizar una búsqueda por cuadrantes siguiendo el algoritmo de Dijkstra, puesto que el coste de una búsqueda sobre mapa completo a nivel de píxeles es demasiado elevado. Una vez que se obtiene el camino de cuadrante se realiza la Búsqueda en Tiempo Real para moverse de un cuadrante a otro.

Algoritmo de Dijkstra

El algoritmo de Dijkstra es un algoritmo eficiente de complejidad $O(n^2)$ donde n es el numero de vértices, que encuentra el camino con coste mínimo desde un nodo origen a todos los demás del grafo. Se basa en el principio de optimalidad: si el camino más corto entre los vértices u y v pasa por el vértice w , entonces la parte del camino que va de w a v debe ser el camino más corto entre todos los caminos que van de w a v [Sánchez-Torrubia and Lozano-Terrazas, 2001] (ver figura 4.4) De esta ma-

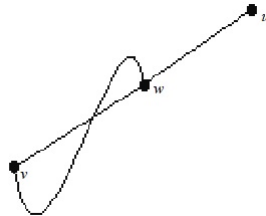


Figura 4.4: Camino mínimo

nera se van construyendo los caminos de coste mínimo desde un vértice inicial hasta cada uno de los vértices del grafo, y se utilizan los caminos conseguidos como parte de los nuevos caminos. En el apéndice B.1 se muestra el pseudocódigo del algoritmo.

Para adoptar el algoritmo al problema particular del escenario, se ha procedido de la siguiente forma. Los sectores se consideran los vértices del grafo. Las aristas son definidas por las adyacencias que se han leído previamente del fichero de configuración nombrado en la sección 4.2.1. Estas

adyacencias son en ambos sentidos: si el sector u es adyacente al sector v , entonces el sector v es adyacente al sector u . El coste de cada arista será de 1.

Algoritmo de búsqueda en celdas

El agente *líder* recibe un camino de cuadrantes hacia la salida más cercana. Por ejemplo, imaginemos que el Agente Líder de la figura 4.5 representado con un círculo rojo debe ir a la salida, representada por la estrella verde. El camino de cuadrantes será: Cuadrante 1 (actual), 2, 3, 4, 5, 6 (destino).



Figura 4.5: Ejemplo de camino de cuadrantes

Inicialmente se han estudiado los algoritmos de búsqueda estáticos, principalmente el algoritmo A^* , para resolver el camino exacto, ya sea a la salida o al cuadrante siguiente, pero se ha llegado a la conclusión de que no resultan adecuados para el problema concreto al que nos enfrentamos. Esto se debe a varios motivos, que se detallan a continuación.

En el caso de la búsqueda de camino al siguiente cuadrante, se decidió crear un submapa cuyos únicos sectores activos fueran aquellos que estuvieran incluidos en la ruta, y se calcularía el camino más corto a la salida con el algoritmo A^* . Continuando con el ejemplo anterior, la ruta se calcularía sobre el sub-mapa de la figura 4.6, con los cuadrantes pertenecientes al camino activos, y el resto marcados como ocupados.

Aún así, el algoritmo de A^* no resulta adecuado por dos motivos. El primero es que resulta un algoritmo muy complejo y costoso, sobre todo teniendo en cuenta los pocos obstáculos permanentes de la mayor parte del escenario. Por tanto un algoritmo tan complejo en la mayoría de los casos

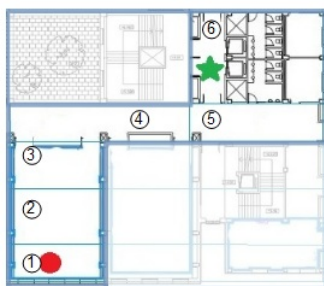


Figura 4.6: Ejemplo de submapa para el Algoritmo A*, como se estudió inicialmente resolver la búsqueda.

no se considera necesario. Además la ruta resultante se espera que sea frecuentemente modificada para evitar las colisiones con los demás agentes, por lo que en varias ocasiones habrá que recalcularse la ruta, lo que supone una desventaja de la utilización de este algoritmo, debido a su coste computacional. Por otro lado, cuando el camino se realiza de un cuadrante a otro, no hay un único píxel de llegada, sino que todo un lado del cuadrante podría ser salida. Y, como se puede ver en el apéndice B.2, el Algoritmo A* no encuentra la salida más cercana, sino una salida cualquiera, por lo que se obtendrían rutas poco “naturales”.

En el caso de la búsqueda del camino completo, el algoritmo de A* podría considerarse adecuado, como se decidió al principio, puesto que siempre encuentra el camino a la salida, si éste existe, pero se trata de un algoritmo muy costoso, y cuando hay muchos agentes que calculan la ruta al mismo tiempo el sistema pierde dinamismo. Una vez calculada la ruta, puede suceder que el agente, al evitar un obstáculo móvil, tal como un grupo de personas, se aleje tanto de su ruta original que no sepa volver. En ese caso tendría que recalcularse la ruta completa. Se espera que esto suceda en incontables ocasiones, por lo que el coste se multiplicaría. Un segundo motivo es la falta de cohesión entre el conocimiento teórico del agente y el real. En teoría el agente debe conocer solamente el cuadrante en el que se encuentra y el camino aproximado a la salida, es decir, el camino de cuadrantes. Pero en la práctica conocería también el camino exacto, píxel a píxel, en cuadrantes que no podría ver. Esto se aleja del realismo de que se quiere dotar a la simulación.

Un Algoritmo en Tiempo Real es un algoritmo que genera una serie de movimientos con un tiempo de computación requerida de $O(1)$ antes del primer movimiento y en cada movimiento siguiente [Parberry, 1995]. No calcula el camino completo hasta el objetivo, sino que a partir de una heurística

calcula uno o varios movimientos, con los que se acerca a la salida.

El problema principal de este sistema es la falta de información. En el caso en el que el agente se encuentre un obstáculo, se pueden realizar muchos movimientos en falso hasta que se consigue bordear. Por ejemplo, en el caso de una mesa pegada a la pared, el agente puede dirigirse a la pared, puesto que no sabe que es el sentido equivocado. Evidentemente, este problema se puede superar añadiéndole una perspectiva de una cierta distancia l , pero aparece el problema de qué distancia es la apropiada, puesto que el sistema debe ser válido para cualquier escenario. Y debido a esto mismo, se debe tener en cuenta el caso en el que el escenario no sea tan simple. Puede pasar que las aulas, por ejemplo, estuvieran amuebladas. En estos casos la mejor alternativa sería un algoritmo estático con la misma visión que se supone que tiene el agente: un cuadrante.

Ante esta disyuntiva se ha decidido llegar a un acuerdo. En primer lugar el agente calcula el punto más cercano a sí mismo del siguiente cuadrante. Se calcula la línea recta discretizada hacia el objetivo desde la posición actual hasta un obstáculo, el objetivo, o un máximo de posiciones (de forma que el coste computacional es de $O(1)$), lo que ocurra antes. En el caso de que se halla encontrado el objetivo, se toma esta línea como el camino provisional. En caso contrario, se calcula el camino mediante el algoritmo de A*.

Así conseguimos que en escenarios simples el cálculo sea rápido, y en escenarios complejos, correcto.

A grandes rasgos, la lógica del *pathfinding* es la siguiente:

1. El agente recibe el camino de cuadrantes del ENTORNO. El camino se compone de los identificadores de los cuadrantes, empieza en el sector actual y termina en el que se encuentra la salida que se considera más cercana.
2. El agente tiene acceso al cuadrante actual a través del ENTORNO, por lo que lo recupera. Recibe también la **adyacencia** (*Norte*, *Sur*, *Este* u *Oeste*) del cuadrante actual con el siguiente (por ejemplo, si el cuadrante siguiente se encuentra encima del actual, la adyacencia será *Norte*).
3. Cálculo del objetivo y generación de camino:
 - a) Se marcan como objetivos toda la línea de coordenadas que sirve de límite del cuadrante por el lado de la adyacencia. En el caso anterior de adyacencia *Norte*, se marcará la línea de coordenadas superiores del cuadrante. Si el agente se encuentra en el último

cuadrante, se marcan como posibles metas todos los puntos que se consideren seguros, es decir, los objetivos finales que corresponden con las salidas de emergencia.

- b) Se calcula el objetivo más cercano según una aproximación: Se elige aquella celda de toda la línea de objetivos que fuera más cercano en línea recta al agente en el caso de que no hubiera ningún obstáculo. Si esa coordenada no está libre, se elige la más cercana a la misma que lo estuviera.
- c) Selección del método de pathfinding y generación de coordenadas:
 - 1) Se calcula la línea recta discretizada de la posición actual hacia el objetivo. Se para cuando: (1) Se llega al objetivo, (2) Se encuentra un obstáculo, o (3) Se llega a un número máximo de pasos; lo que ocurra antes.
 - 2) Si se ha encontrado el objetivo, se salta a 3c4.
 - 3) Si se ha encontrado un obstáculo o se ha cumplido el número máximo de pasos sin encontrar el objetivo, se calcula el camino mínimo mediante el algoritmo A^* .
 - 4) Se calcula la coordenada siguiente: se comprueba si en la celda siguiente de la lista hubiera algún obstáculo, o si lo hay a una cierta distancia. Si los hay, como se explica en 4.2.4. Si el camino se aleja más de un número de pasos de la posición deseada, se vuelve a 3b. Si no hay obstáculos, envía la coordenada y se marca la nueva coordenada como coordenada actual.
 - 5) Se comprueba si la coordenada es un objetivo final. Si lo es, termina. Si no, se comprueba si es meta, es decir, el final del cuadrante. Si lo es, se vuelve a 2. Si no, se vuelve a 3b.

4.2.3. Descripción de los agentes

En esta sección se explica el diseño de los agentes que componen el sistema, tanto los agentes persona como el resto. El sistema cuenta con los siguientes tipos de agentes:

- Agentes Persona. Son los agentes que simulan el comportamiento humano.
- El Agente Fuego (*Fire Agent*). Su comportamiento se podría considerar “invasivo”, puesto que alcanza nuevas posiciones pero sin dejar de estar situado en las posiciones anteriores.

- Agente Mundo (*Environment Agent*). Representa el ENTORNO. Es el único agente con el conocimiento completo del entorno, y, mediante mensajes, le pasa a los agentes apropiados parte de su conocimiento.
- Agente de Comunicación (*Communication Agent*). Funciona como un intermediario entre el recurso gráfico, la GUI, y los agentes con representación gráfica (los agentes persona y el fuego). Cada cierto tiempo le pide a los agentes persona su posición actual, y se la envía a la GUI.
- Agente creador (Creator Agent). Es el agente inicial, encargado de crear al resto de los agentes. Los agentes y su parámetros de configuración se se pasan al sistema mediante un fichero. A partir de él el agente Creador es capaz de crear a los agentes indicados en el mismo.

A continuación se estudia en mayor profundidad algunos tipos de agentes.

Los agentes persona

Los agentes persona simulan a las personas involucradas en la evacuación. Cada persona es representada por un agente. Estos agentes se mueven a través del entorno, toman decisiones, y cambian sus comportamientos según sus características personales y estímulos externos, provenientes del entorno o del resto de agentes persona.

Los agentes persona representan al personal del centro de estudio, celadores, secretarios, camareros, profesores e investigadores, y a los alumnos. Su comportamiento es relativo a su ocupación. Habrá un profesor en cada aula, con varios alumnos, algún celador se pasea por el pasillo. . . Hasta que empieza la alarma de evacuación. En ese momento los roles serán diferentes. Hay tres roles principales en la evacuación: el rol *líder*, el *independiente* y el *dependiente*.

Los líderes son aquellos agentes persona que se encargan de ayudar a los demás a realizar la evacuación en condiciones de seguridad. Los dependientes no son capaces de tomar las suficientes decisiones por cuenta propia como para realizar la evacuación, por lo que siguen a los líderes. Los independientes, aunque en ciertos momentos puedan pertenecer a un grupo, realizan acciones por su cuenta, y en ciertas ocasiones pueden ocasionar problemas, al contradecir las normas de evacuación. A continuación se describe más detalladamente cada tipo de agente persona en función de su rol durante la evacuación.

Agentes Líderes Los agentes líderes pueden ser los los responsables de la evacuación asignados, y también otras personas con un grado de certeza del

incidente, conocimiento del entorno y determinación suficiente para actuar como un líder en la evacuación. Son agentes con autonomía para tomar decisiones, tienen la capacidad de conocer el camino de salida, y se preocupan por el resto de los agentes, buscándolos y llamándolos para guiarlos a la salida.

Cada agente *Líder* particular tiene una serie de características propias que lo identifican, como su nombre, y que pueden llegar a determinar su comportamiento a lo largo de la simulación. Cuentan con un cierto grado de liderazgo, definida mediante el grado de conocimiento del entorno que posee el agente y su carisma y su cercanía al foco del incidente.

Cuando empieza una evacuación los agentes de este tipo le piden al Agente Mundo el camino de sectores desde su posición actual hasta la salida más cercana. Esta salida recibida no siempre será la correcta para modelar las situaciones en las que una persona, aunque conozca bien el entorno por el que se mueve, debido al miedo o a los nervios pierde en ciertas situaciones la orientación. Una vez que obtienen la ruta en sectores, calculan el camino exacto a seguir dentro de esos sectores. Durante la evacuación, los agentes buscan dependientes. Cada vez que llegan a un nuevo sector llaman a todos los agentes persona que se encuentren a su alcance (los que estén en el mismo sector) pidiéndoles que se unan a ellos. Su liderazgo, que inicialmente depende de variables de configuración, va variando en función de los agentes que empiezan a seguirle, aumentando, o que dejan de hacerlo, disminuyendo.

Como se comenta en la sección 4.1, los agentes tienen un triplo de información a partir del cual se decide el comportamiento del agente en cada momento.

- El CONOCIMIENTO. La información del agente obtenida del ENTORNO de los agentes Líderes consiste en:
 - **Las salidas de la evacuación.** Los lugares seguros hacia los que hay que dirigirse. Son proporcionadas por el ENTORNO. En ciertas ocasiones en las que el agente se encuentre bajo demasiada presión, puede elegir la salida equivocada. En ese caso el sistema logará el error.
 - **El camino hacia la salida.** Es capaz de obtener en camino en sectores desde la posición inicial hasta la salida que considere más cercana. Estos dos tipos de información constituyen la **información ambiental**.
 - **Percepciones.** Todo aquello que se encuentre dentro de los límites de la *Zona de percepción* puede ser percibido por el agente.

La zona es el sector en el que se encuentra. El agente tiene acceso a la información de todo lo que necesite que se desarrolle o se encuentre en su sector. Pero además, dispone de información básica de los sectores adyacentes: Si el fuego se encuentra en ellos.

- El ESTADO. La información predefinida del agente, que puede ir cambiando a lo largo de la simulación, es:
 - El Perfil, que será el de Líder.
 - La Salud, que empieza con valor '1', y puede ir disminuyendo.
 - La Situación: *Vida-normal*, *Riesgo* o *Seguridad*.
 - El Grado de liderazgo. Es un atributo que depende de varios. Depende del grado de conocimiento del entorno, la certeza del incidente, su rol (encargado de la simulación, o no), y su carisma. Todos estos atributos se han agrupado en el *Grado de liderazgo*. Cuanto más alto sea el liderazgo de un agente, más gente le seguirá y obedecerá sus órdenes. El grado de liderazgo cambia durante la simulación y depende también de circunstancias externas. Cuanta más gente siga a un determinado agente, mayor será su liderazgo. Si la gente que sigue a un agente deja de hacerlo para, por ejemplo, seguir a otro agente, el liderazgo del primero disminuye. Si un agente líder con un liderazgo medio se encuentra en una zona con muchos agentes dependientes, éstos empezarán a seguirle, haciendo que su liderazgo aumente.
- Las INTENCIONES. Consisten en objetivos a alto nivel que el agente pretende alcanzar. No tienen por qué llegar a alcanzarse, y no son permanentes, es decir, el agente puede cambiar de intención antes de conseguir la intención en curso. En el apéndice C se explican la mayoría de las intenciones que puede tener un agente.

Cuando un agente líder entra en un cuadrante en el que haya fuego, modifica sus INTENCIONES y pasa a dirigirse rápidamente hacia otra salida, buscando un camino que esté libre. Sin embargo, se puede dar la circunstancia de que el fuego le rodee, y en ese caso el agente morirá.

Agentes independientes Los Agentes Independientes son quizá los agentes a tener más en cuenta en la simulación. Su triplo de información es el mismo que el de los Agentes Líderes, pero sus INTENCIONES cambian. No ayudan a los dependientes, pero no sólo eso. Normalmente se dirigen a la salida por sí mismos, pero en ocasiones deciden ignorar las normas de evacuación, y se dirigen hacia zonas peligrosas, o alejadas de la salida. Sus INTENCIONES son, en primer lugar, egoístas, puesto que nunca ayudan a nadie, y, en segundo lugar, pueden resultar peligrosas. Puede ir a buscar a un

amigo, una pertenencia (portátil, etc.) a cualquier zona, sin tener en cuenta las normas de evacuación, que exigen dirigirse directamente hacia la salida, sin recoger ningún objeto. Como consecuencia, es más fácil que sufran un accidente. Se considera que el comportamiento de este tipo de agentes debe ser atentamente tenido en cuenta, puesto que supone el mayor riesgo de la evacuación. En el apéndice C se explican la mayoría de las intenciones.

Agentes dependientes Los Agentes Dependientes son los más numerosos de la simulación. Representan a todas las personas que necesitan ayuda para salir del edificio. Son casi todos los estudiantes y parte de los trabajadores del centro. Mientras la situación es normal, sin riesgos, los dependientes vagan por el escenario siguiendo su rol (las actividades son diferentes según su rol profesional). Pero cuando estalla la alarma y debe comenzar la evacuación, los agentes dependientes no saben hacia dónde dirigirse. Por lo tanto, se mueven aleatoriamente por el escenario buscando un líder que les guíe. Si en un cierto tiempo no han encontrado uno, consiguen del entorno el camino a un sector adyacente. Cuando reciben ofertas de ayuda, los agentes se unen al líder con mayor liderazgo. Esto hace que el liderazgo del líder elegido aumente. El Agente Dependiente pasa a formar parte de un grupo, y pertenecerá a él hasta que encuentre un líder que le proporcione más *confianza*.

Al igual que los Agentes Líderes, los Agentes Dependientes tienen un tripo de información CONOCIMIENTO, ESTADO e INTENCIONES.

- El CONOCIMIENTO de los Agentes Dependientes se limita a su **zona de percepción**, que consiste en el sector en el que se encuentra, y el camino a un cuadrante adyacente. Del sector en el que se encuentra tiene conocimiento total. Todo el conocimiento es obtenido del ENTORNO.
- El ESTADO. Los atributos del estado son los mismos que los de los Agentes Líderes, con la diferencia de que su Perfil será el de Dependiente, y su Grado de liderazgo nulo.
- Las INTENCIONES. Las intenciones se describen en el apéndice C.

Una vez que un agente dependiente comienza a seguir a un líder, va recibiendo su posición y se intenta aproximar a él todo lo posible. De esta forma se va acercando a la salida.

4.2.4. Detección de colisiones y Crowd Simulation

Cuando un Agente Dependiente comienza a seguir a un líder, se forma un grupo. El movimiento lo decide el líder del grupo, y se lo envía a los demás. Cuando el grupo es pequeño, de tres o cuatro personas, esta lógica es

suficiente, pero cuando se hacen muy numerosos es necesaria la simulación de multitudes (*Crowd Simulation*). En este apartado se estudia cómo se afrontan estas dos problemáticas.

Detección de Colisiones

Los Agentes Líderes tienen un camino concreto para llegar a su destino, donde cada paso corresponde a un píxel, que se ha hallado con el Algoritmo A*. Al hallar el recorrido sólo se han tenido en cuenta los obstáculos permanentes, como paredes o mobiliario, pero los agentes tienen que esquivar también a otros agentes que se van cruzando por el camino. Como los agentes persona tienen conocimiento de todo su cuadrante, saben dónde se encuentran el resto de agentes, y pueden esquivarlos.

El mapa, además de en Sectores, está dividido en celdas. Una celda tiene el tamaño que se supone que tiene un agente. Cada agente sabe a dónde se quiere dirigir en un momento dado, considerando ésta su *celda ideal*, e indica que el agente quiere moverse en una dirección, *N*, *NE*, *E*, *SE*, *S*, *SW*, *W* o *NW*. Lo que hace entonces es comprobar si ese punto está libre u ocupado. Si está libre, lo ocupa. Si está ocupado, trata de rodearlo dirigiéndose hacia el mismo destino haciendo el menor rodeo posible. Por ejemplo, si quiere dirigirse hacia el *Norte*, y esa celda está ocupada, prueba con la casilla hacia el *Noreste* y *Noroeste*. Si las dos están libres, elige una de ellas aleatoriamente, si no, prueba a quedarse quieto. Si el agente pasa un límite de tiempo quieto y sigue sin poder moverse, entonces recalcula la ruta hacia su objetivo.

Capítulo 5

Generación de informes: La Herramienta FromLog2StoryTelling

El SMA con el que se ha trabajado es el sistema de prevención de evacuación de incendios explicado en el capítulo 4. Se tratan los problemas que se han resuelto para la implementación del sistema así como las diferentes representaciones semánticas intermedias que son necesarias para transformar un archivo de log de la salida de un sistema multi-agente con la entrada del generador de lenguaje. Se busca solución, entre otros, al problema de las agrupaciones y de la elección de la representatividad de la actividad de los agentes a la hora de la evacuación de un edificio.

5.1. Contexto del sistema y estructura del log

Como se explica en el Capítulo 1 el sistema que genera el log es un simulador multi-agente de evacuaciones en incendios. Se estudia el comportamiento de las personas involucradas, y las consecuencias que estos comportamientos pueden provocar en los resultados de la evacuación.

El escenario de la evacuación es la Facultad de Informática de la Universidad Complutense de Madrid. Para gestionar el espacio se ha utilizado un mapa de la Facultad dividido por plantas (véase Figura 5.1). El escenario está dividido por sectores (se puede ver un plano con los sectores en la Figura 4.1). Por ejemplo cada pasillo se divide en 4 sectores y las aulas cada una en 3 partes.

Para que el sistema genere apropiadamente el informe necesita alguna información adicional. El SMA genera un fichero con la información de cada agente que simbolice una persona. Este informe se utiliza en el módulo de

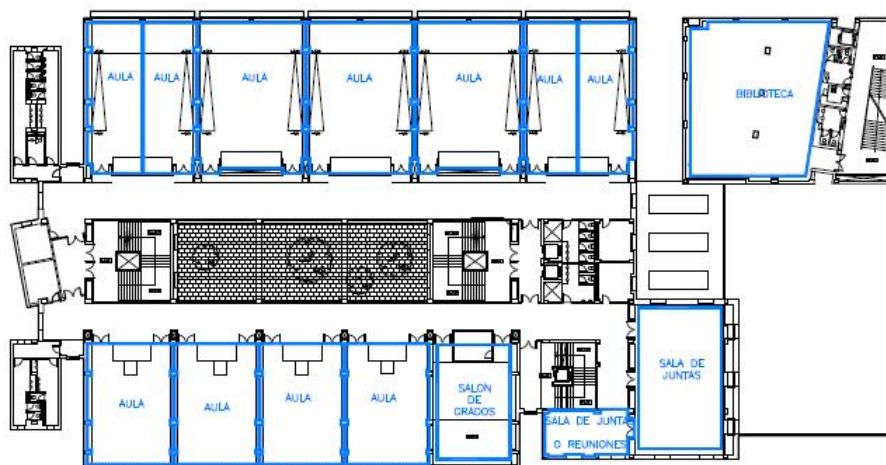


Figura 5.1: Parte del escenario de entrada: planta primera de la Facultad de Informática.

Determinación de Contenido, como se ve la Sección 5.2.2.

El SMA genera un log, en el que están reflejadas las acciones o *facts* que han realizado los agentes durante la simulación. Nuestro sistema recibe el log, y a partir de él, genera el informe de lo sucedido en la simulación. Cada línea del log tendrá la forma:

[NombreAgente] Fecha y Hora IdAc Place Action Parameter

- **NombreAgente:** El nombre personal del agente, por ejemplo, Juan Martín. Un agente es una persona implicada en la evacuación, o bien el fuego.
- **Fecha y Hora:** Instante en que se realizó la acción.
- **IdAc:** Número que identifica cada acción de forma inequívoca.
- **Place:** Lugar en que se realiza la acción. Por ejemplo, *Aula16-zona1*.
- **Action:** Acción que se realiza. Puede referirse también a un cambio de estado (*Dirigirse-a*, *Accidente*, etc).
- **Parameter:** Parámetro que complementa a la acción. Por ejemplo, el parámetro de la acción llamada *Andar-hacia* será un lugar, como *puerta Aula 14*, *Escaleras segunda planta*. En algunos casos el parámetro será el nombre de otro agente, como por ejemplo cuando la acción es

Seguir-a. Este campo puede estar vacío, puesto que no todas las acciones lo necesitan. Por ejemplo, en la acción *Llegar-a*, el complemento queda implícito en el campo *Place*.

Los agentes pueden realizar distintas *Acciones*. Estas acciones pueden ser intenciones (como *Obedecer-a*, que implica más un cambio de estado) o acciones propiamente dichas (*Subir*, *Llegar-a*). Algunas pueden tener información implícita, como el grupo de acciones del tipo *Desobedecer-acción*, que indican que realizan la acción actuando contra las normas indicadas por los evacuadores, o que, conociendo el camino de salida, el agente se aleja de él.

No sólo las personas realizan acciones, sino que también el fuego tiene comportamientos que luego se tendrán en cuenta en el relato final. Sus acciones son del tipo *Llegar-a*, *Invadir-area* etc. Todos estos eventos, y algunos más, se encuentran también reflejados en el log inicial. La lista completa de las acciones que pueden aparecer en el log inicial se puede consultar en el Apéndice D.

5.2. Determinador de Contenido

Como explicamos en la Sección 2.2 el módulo de *Determinación de Contenido* se encarga de seleccionar la información que resulte interesante incluir en el texto final, eliminando aquella que sea poco relevante para el discurso, o que no aporte información útil para la descripción de la evacuación. Por ejemplo, no es necesario explicar uno por uno las 50 personas que escuchan a un responsable de evacuación; además, creemos importante que la narración sea medianamente dinámica y entretenida.

Para lograr estos objetivos, el módulo realiza las funciones que se mencionan a continuación:

1. Creación de los grupos. Hay acciones que si las realiza una persona de forma individual resultan irrelevantes, pero al ser ejecutadas por grupos cobran importancia. Además, el liderazgo es una cuestión a estudiar en una evacuación, puesto que divide a las personas en líderes y seguidores, cuyo criterio queda delegado en los primeros. También es interesante el estudio de la creación, disolución o fusión de grupos para comprobar cómo emerge el comportamiento grupal a partir de los comportamientos individuales.
2. Se determina el interés de cada agente individual y de cada grupo.

3. En función del interés se eliminan las personas y grupos no relevantes. El hecho de que un agente sea interesante no quiere decir que toda su historia vaya a ser interesante. Se eliminan las acciones no relevantes de agentes relevantes. Cabe destacar que, aunque el fuego se considere un agente más a lo largo de los módulos de *Determinación de Contenido*, su interés asignado será siempre muy alto, como se verá más adelante; por lo que nunca será eliminado.

5.2.1. El problema de las agrupaciones

A la hora de definir los hechos de la simulación se debe tener en cuenta la creación de los grupos, dado que una acción es más interesante cuando la realiza un número elevado de agentes, que por otro lado no la realizan por decisión propia, sino porque siguen a una persona denominada líder. En definitiva, son conjuntos de personas que delegan sus decisiones en otra cuyo liderazgo es mayor y simplemente imitan las acciones que este último realiza mientras le siguen. A estos conjuntos de personas es a lo que llamamos un grupo, y viene determinado por su líder.

Para gestionar agrupaciones se han creado una nueva entidad *grupo* que tiene la información del líder, de la gente que pertenece al grupo (con tiempo de inicio y fin de pertenencia) y de las acciones realizadas por el grupo.

Si un agente por sí solo o un grupo pequeño de agentes se une o separa del grupo, ¿se considera un grupo diferente?. Para solucionar esto consideramos que cada grupo está determinado por un representante, su líder. Se empezará a hablar del grupo cuando su número supere un porcentaje sobre el número total de individuos del SMA, y hasta que disminuya a otro porcentaje.

Otra consideración a tener en cuenta es en qué momento se forman los grupos, es decir, si dos personas se consideran grupo o si debe estar formado por más gente. Finalmente se ha decidido que un grupo existe desde el momento en que una sola persona sigue a otra, aunque el grupo no será considerado relevante a la hora del discurso hasta que no sea más grande. Para ello se crea un valor umbral que determina el tamaño mínimo del grupo para que se considere interesante contar cosas sobre él, y por tanto, susceptible de ser comentado en el informe final (en lenguaje natural). De la misma forma, un grupo dejará de ser interesante cuando su número de miembros disminuya por debajo del umbral. Los grupos se crean virtualmente cuando se ejecuta la acción *Crear-grupo*. Esta acción surge cuando una primera persona realiza la acción *Seguir-a* sobre otra que aún no es seguida por nadie. Una persona abandona su grupo cuando realiza la acción *Dejar-de-seguir-a*. El grupo se disuelve cuando se realiza la acción *Deshacer-grupo*, que se ejecuta cuando

la última persona que seguía al líder X ejecuta *Dejar-de-seguir-a X*, y no hay ninguna otra persona que esté siguiendo a X .

5.2.2. Interés

Para determinar los personajes a los que se considera relevantes y eliminar al resto, para después elegir las acciones importantes de cada uno de ellos, entendiendo como personaje tanto a una persona como un grupo o el fuego, se asigna un *factor de interés* siguiendo el planteamiento de [León et al., 2007]. Tanto el número de agentes que se mencionarán en el informe como el nivel de detalle con que se describirá la evacuación es configurable.

La selección de contenido se realiza en dos fases. En la primera, *Selección de Personajes*, se seleccionan los agentes que tengan un mayor interés personal, ya sea por sus características personales o por las acciones en que haya participado. Para ello se utilizan dos heurísticas. La primera asocia un valor base a cada característica del agente (como por ejemplo el perfil), y la segunda define la importancia que tiene para cualquier agente el hecho de haber participado en cada acción, ya sea como sujeto o como complemento. Por tanto, cada agente será seleccionado por su interés personal y por su interés acumulado, interés que ha ido adquiriendo a lo largo de la simulación por sus acciones tanto realizadas como recibidas. En la segunda fase, llamada *Selección de Eventos*, se seleccionan los eventos en los que hayan participado los agentes seleccionados que puedan ser interesantes para incluir en el informe. Esto se decide, por una parte, por el interés personal del agente que participe en ella (o los agentes, en caso de que sea un agente el parámetro de la acción) y por otra por la heurística asociada a esa acción.

Selección de Personajes

La *Importancia Personal* de cada agente (I_p) determinará de quién se habla y de quién no. En este punto, se consideran tres tipos de agentes: el fuego, los grupos y las personas. El fuego será siempre considerado como de mucho interés. Respecto a los grupos y las personas, su selección dependerá de sus características personales y de las acciones de las que formen parte durante toda la simulación. Esta medida se toma con los siguientes factores:

- **Interés primario.** Es un valor basado en una heurística en función de las características personales. Para el fuego su tipo de personaje es condición suficiente para que sea considerado importante, mientras que para el resto (grupos y personas) se miden más atributos. De las

personas se mide su rol (evacuador o evacuado) y otras características como nivel de liderazgo o independencia, y de los grupos características como número medio y máximo de miembros. A este valor lo llamamos $I_c(X)$, siendo X un personaje.

- **Interés derivado de realizar una acción $h_s(F)$.** Es una heurística asociada a cada tipo de acción (*andar-hacia*, *dirigirse-a*, *accidente*) que valora el interés que tiene que alguien la realice. Por ejemplo, la acción *Dirigirse-a* tiene asociado un valor de $h_s(F)$ menor que *Accidente*.
- **Interés derivado de participar como objeto en una acción $h_c(F)$.** Existen algunos casos en los que un personaje aparece no como sujeto de la acción, sino como complemento; como por ejemplo en *Obedecer-a*, *Seguir-a*, etc. (*Juan obedece a Pedro*, *Paco sigue a Mariana*). Esta heurística valora la importancia de recibir una acción F .

De esta forma, queda definida la función de importancia personal para cada personaje (X) como la suma de las importancias de todas y cada una de las acciones que ha realizado y que ha recibido, teniendo en cuenta su interés primario:

$$I_p(X) = \sum_{i=1}^n I_c(X) \cdot h_s(i) + \sum_{j=1}^m I_c(X) \cdot h_c(i)$$

o, lo que es lo mismo

$$I_p(X) = I_c(X) \cdot [\sum_{i=1}^n h_s(i) + \sum_{j=1}^m h_c(j)]$$

donde n es el número de acciones realizadas por el personaje X , m el número de acciones cuyo complemento es X , $I_c(X)$ el interés primario de X , y $h_s(i)$ y $h_c(i)$ los intereses de realizar y de recibir, respectivamente, la acción i .

En función de la cantidad de personajes que hayan participado en la simulación se elige un número k de los mismos de los que se hablará. De todos los personajes se eligen aquellos con mayor valor en I_b , y se eliminan todas las acciones del resto no seleccionado.

Selección de Eventos

En este submódulo se seleccionan los eventos que se incluirán en el informe. En una primera fase se eligen aquellos cuya importancia depende exclusivamente del tipo de acción y de los agentes que participen. Después, en la segunda fase, se tiene en cuenta el contexto para seleccionar también los eventos irrelevantes que dieron lugar a eventos muy relevantes, como se explica más adelante.

Llegados a este nivel, se tiene una lista con todas las acciones que realizan los individuos importantes. Pero por una lado, no nos interesa contar todo lo que hace cada individuo (En vez de decir “*Pedro se dirige a la puerta. Pedro llega a la puerta. Pedro se dirige al Pasillo 2, Pedro sale de la puerta, Pedro llega al Pasillo 2*”, el concepto queda igual de claro diciendo tan sólo “*Pedro está en el Aula 16 y llega al Pasillo 2*”). Además, de los personajes más importantes se cuentan más hechos que de los menos importantes. Para esto asignamos a cada evento un valor de interés $I_{ev}(event)$ que define lo interesante que se considera el evento $event$. Si el evento tiene como sujeto al agente X y no tiene agente como complemento, entonces

$$I_{ev}(event) = I_b(X) \cdot h_s(F)$$

sin embargo, si el complemento del evento es también un agente (el agente Y), entonces el interés del evento queda determinado por

$$I_{ev}(event) = \text{máx}\{I_b(X) \cdot h_s(F), I_b(Y) \cdot h_c(F)\}$$

Utilizando esta función, se mide el interés de cada evento. De forma análoga al sub-módulo *Selección de Personajes*, en función del número de eventos totales se elige el porcentaje que se incluirá en el resumen final, definiéndose un valor umbral, de forma que todos los eventos cuyo interés supere dicho umbral serán marcados como “interesantes”, y por tanto se contarán en el relato final. Como se puede ver en la fórmula, la importancia de cada evento es directamente proporcional al interés del sujeto que la ejecute. Así, un evento realizado por el personaje *fuego* tendrá siempre asociado un alto valor de interés.

Sin embargo, los eventos con un bajo valor de I_{ev} aún no son eliminados, puesto que en este punto aparece otro problema a tratar. Se considera que un evento puede tener mayor importancia si precede o sucede a un evento relevante, siempre y cuando el sujeto de ambos eventos sea el mismo. Por ejemplo, si un personaje muere, cualquier usuario presumiblemente querrá saber qué cadena de sucesos le han llevado a esa situación, aunque dichos sucesos por sí mismos no tengan una gran importancia como eventos aislados. Para este propósito hemos incluido un nuevo valor umbral, al que hemos llamado *top*, y un nuevo concepto, el de “alto interés”. Un evento v tendrá un alto “interés” si $I_{ev}(v) \geq top$. Entonces, aquellos n eventos anteriores y posteriores (si existieran) al evento elegido como de *alto interés* cuyo sujeto sea igual a este último serán marcados como “interesantes” y por lo tanto no serán eliminados.

5.3. Planificador de Discurso

Los principales objetivos del *Planificador de Discurso* son:

1. Ordenar en el espacio y tiempo las acciones. La función principal de este módulo es la ordenación de los hechos y acciones de una forma coherente.
2. Adaptar la salida a la entrada del TAP.

Para ordenar las acciones seguimos dos criterios:

- **Orden posicional y temporal.** En primer lugar se extrae de la entrada cada evento ordenándolo por el lugar en que se lleva a cabo (el lugar no será un aula, sino que se seccionará en escenarios más pequeños). Este orden se lleva a cabo de forma que primero están los escenarios más alejados de las salidas según la ruta de evacuación, y en último lugar los más cercanos. Esto es así para facilitar la coherencia, puesto que la mayoría de los personajes irán desde las zonas alejadas de las salidas hasta las más cercanas. Los casos en que no ocurra así, son marcados de forma que sus eventos que no mantengan esa coherencia son ordenados sólo por el orden temporal en que ocurren, sin pertenecer a grupos ni de escenario ni de tiempo. Una vez que los eventos se ordenan por escenario, para cada lugar se ordenan estos hechos por el instante de tiempo en que se llevan a cabo; de forma que se obtiene una relación de los eventos que tienen lugar en cada escenario y ordenados en el tiempo.
- **Bloques de tiempo.** Si como texto final de salida se contara en orden todo lo que sucede en cada escenario de principio a fin habría conjuntos de acciones que quedarían inconexos, como aquellos en los que los agentes cambian de un escenario a otro, y además el texto resultaría tedioso para el lector. Debido a ello, hemos decidido unir las acciones en bloques temporales de 5 segundos. De esta forma, se va contando lo que pasa en cada bloque temporal en cada escenario. Sin embargo, estos bloques podrían cambiar su duración excepcionalmente en los casos en los que ciertos conjuntos de acciones deban contarse unidos para preservar la coherencia. Por ejemplo, en el caso en que en la última acción del bloque n un responsable de evacuación se dirija a las personas que se encuentren en un escenario, se deberá añadir a este bloque la acción respuesta de los agentes (que le escuchen o no), aunque en principio fueran asignadas al bloque $n+1$.

5.4. Ejemplo

En la Figura 5.2 podemos ver un ejemplo de un esquema de evacuación. Los círculos de color violeta representan a los evacuados que siguen las normas de desalojo, el rojo es el evacuador y el de color negro es un evacuado que ignora las normas, y acaba llegando al fuego. Las flechas indican los

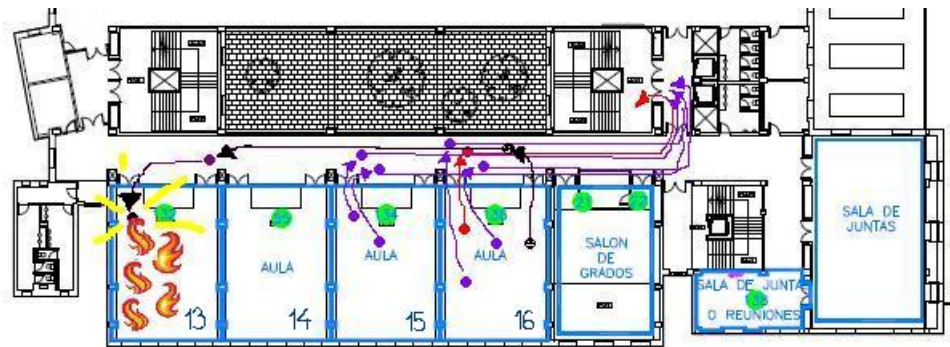


Figura 5.2: Planta primera de la Facultad de Informática en un ejemplo de evacuación

recorridos que realizan. En el Aula 16 el líder ordena a todos los evacuados de la sala que le sigan hacia la salida. Todos le obedecen excepto el agente negro. En el Aula 15 hay 2 evacuados más que se dirigen al pasillo. Todos los agentes del Aula 16 salen al pasillo, donde el evacuador vuelve a pedir que le sigan. Los dos agentes provenientes del Aula 15 se unen al líder, mientras que el agente de negro sigue ignorándole. El grupo del evacuador se dirige y llega a las escaleras, mientras que el agente de negro se dirige al Aula 13, donde se encuentra el fuego. Al entrar sufre un accidente y muere.

- En el módulo de **Determinación de Contenido** se creará el grupo liderado por el responsable de evacuación (rojo). No se considera relevante el hecho de que las dos personas del *Aula 15* salgan y vayan al pasillo. La persona que no quiere seguir al líder (negro) acaba presa de las llamas, por lo que será de *alto interés*.
- En el **Discourse Planning** se define el orden de los eventos.
 - Primero, se ordenan por escenarios de más alejados a cercanos a la salida).
 - Después, dentro de este orden, por tiempo.
 - En último lugar se dividen los eventos en bloques.
 - Debido al agente que no cumple las normas no se puede mantener en ese orden por motivos de coherencia.

Capítulo 6

Resultados

Se ha desarrollado un sistema multi-agente que simula el comportamiento de las personas involucradas en una evacuación. Se han creado tres roles de comportamiento. En primer lugar, los líderes. Son aquellas personas, que o bien por formar parte del personal encargado de las evacuaciones, o por ciertas características personales, se encargan de ayudar a la gente a salir. Los dependientes necesitan ayuda para evacuar el edificio, y la reciben de los líderes. Por último se simulan los comportamientos de independientes, agentes que actúan por su cuenta y en ocasiones desobedecen las normas de evacuación, poniendo en peligro su vida e incluso la de los demás. Esto es un gran avance puesto que no se ha encontrado ningún simulador de evacuaciones con agentes en el que se tuviera en cuenta ningún comportamiento contrario a las normas de evacuación, como se puede ver en el capítulo 2.

Pero no se considera suficiente la observación de la salida gráfica de la simulación. En muchas ocasiones es necesario acceder a los logs de para tratar los resultados, hacer estadísticas, etc. Además, puede ser que algún evento importante se haya escapado, o que por el motivo que sea no se esté presente cuando se ejecuta la simulación. Además, hay algunos tipos de eventos, tales como las intenciones, que no se pueden ver en la simulación, hay que revisar el log para reconocerlas. Pero un log contiene demasiada información, y la mayor parte de ella no suele resultar relevante. Se pueden ver ejemplos de logs generados en el apéndice E.

La herramienta From Log To Story Telling trata y resume los logs para que, al meterlos como entrada de la herramienta TAP, salga un resumen en lenguaje natural con los eventos importantes. Queda como trabajo futuro adaptar el resultado de FL2ST como entrada de TAP.

En este capítulo se muestran y analizan los resultados de los dos sistemas diseñados e implementados. Se muestran varios ejemplos de simulación en

el que se muestran los tres perfiles diferentes. A partir de cada simulación, se muestra cómo se tratan los logs de resultado con la herramienta *From-Log2StoryTelling* para conseguir un resumen con los sucesos más relevantes de la simulación.

6.1. Varios líderes y el fuego



Figura 6.1: Varios líderes

En el primer caso se muestran varios líderes que se dirigen a la salida, cada uno a la que considera más cercana. La posición inicial de cada agente, se puede observar en la figura 6.1. Los líderes están representados por los círculos azules. Hemos marcado los líderes más importantes, los que se encontrarán con el fuego, rodeándolos con una circunferencia roja. Como



Figura 6.2: Varios líderes

se ve en la figura 6.2, el fuego se ha iniciado en el camino hacia la salida de la izquierda, y los agentes que se dirigían hacia esa salida se encuentran con él. Cuando los agentes están llegando se cruzan con el foco del fuego, y por tanto, tienen que cambiar el destino, y, sin atravesar el sector con fuego, dirigirse hacia la otra salida. Cambian su dirección para llegar a la salida de la derecha. Los agentes ven el fuego cuando éste alcanza su radio de percepción, es decir, cuando ambos se encuentran en el mismo cuadrante. En la figura 6.3 se puede ver cómo los dos agentes que se encontraron con



Figura 6.3: Varios líderes

el fuego, y que siguen rodeados por la circunferencia roja, han cambiado la dirección, y se dirigen ahora hacia el nuevo destino, la salida de la derecha, llegando hasta ella sin más complicaciones.

El log obtenido ocupa más de mil líneas, por lo que deducir qué ha ocurrido en la simulación resulta impracticable. Se muestra una parte en el apéndice E.1.

Como en este caso no hay dependientes, no se forman grupos, por lo que no se trata esta parte. Como entrada del FL2ST introducimos el log, el archivo de propiedades con las características de los agentes que se utilizó para la simulación con los que recoger algunos valores importantes, como el número de agentes y el liderazgo de los líderes. Otro archivo de propiedades muy importante es aquel en el que añadimos los valores de *Interés* por sujeto y complemento. Además, añadimos el número máximo de agentes de los que queremos hablar. Por defecto el valor es 10. Dejando los valores por defecto la salida que obtenemos es:

```
Fire1 en el cuadrante 4 acceso 4
Fire1 en el cuadrante 12 inicio-foco (10,30)
Leader3 en el cuadrante 28 empezar
```

```

Leader3 en el cuadrante 28 andar-hacia 41
Leader1 en el cuadrante 14 empezar
Leader1 en el cuadrante 14 andar-hacia 12
Leader6 en el cuadrante 39 empezar
Leader6 en el cuadrante 39 andar-hacia 41
Leader2 en el cuadrante 19 empezar
Leader2 en el cuadrante 19 andar-hacia 12
Leader4 en el cuadrante 36 empezar
Leader4 en el cuadrante 36 andar-hacia 41
Leader5 en el cuadrante 43 empezar
Leader5 en el cuadrante 43 andar-hacia 41
Fire1 en el cuadrante 3 acceso 3
Leader1 en el cuadrante 5 visto-fuego
Leader1 en el cuadrante 5 andar-hacia 41
Leader5 en el cuadrante 41 llega-a-meta
Leader5 en el cuadrante 41 permanecer-lugar-seguro
Leader2 en el cuadrante 22 visto-fuego
Leader2 en el cuadrante 22 andar-hacia 41
Leader6 en el cuadrante 41 llega-a-meta
Leader6 en el cuadrante 41 permanecer-lugar-seguro
Leader4 en el cuadrante 41 llega-a-meta
Leader4 en el cuadrante 41 permanecer-lugar-seguro
Leader3 en el cuadrante 41 llega-a-meta
Leader3 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 11 acceso 11
Fire1 en el cuadrante 5 acceso 5
Leader2 en el cuadrante 41 llega-a-meta
Leader2 en el cuadrante 41 permanecer-lugar-seguro
Leader1 en el cuadrante 41 llega-a-meta
Leader1 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 6 acceso 6

```

Si queremos reducir la salida de forma que sólo se hable de los 3 principales agentes, que en esta caso serían el fuego y los dos agentes que se encuentran con él, introducimos 3 como número máximo de agentes de los que queremos hablar. La salida es:

```

Fire1 en el cuadrante 12 inicio-foco (10,30)
Fire1 en el cuadrante 4 acceso 4
Leader1 en el cuadrante 14 empezar-en
Leader1 en el cuadrante 14 andar-hacia 12
Leader2 en el cuadrante 19 empezar-en
Leader2 en el cuadrante 19 andar-hacia 12
Fire1 en el cuadrante 3 acceso 3
Leader1 en el cuadrante 5 visto-fuego
Leader1 en el cuadrante 5 andar-hacia 41
Leader2 en el cuadrante 22 visto-fuego
Leader2 en el cuadrante 22 andar-hacia 41
Fire1 en el cuadrante 11 acceso 11
Fire1 en el cuadrante 5 acceso 5
Leader2 en el cuadrante 41 llega-a-meta
Leader2 en el cuadrante 41 permanecer-lugar-seguro
Leader1 en el cuadrante 41 llega-a-meta
Leader1 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 6 acceso 6
Fire1 en el cuadrante 10 acceso 10

```

Así queda mucho más claro. Se pueden ver los valores de las acciones que se han tomado para tener estos resultados en el apéndice D.

En este caso, como no hay dependientes, no se forman grupos. En el siguiente ejemplo se muestra cómo los agentes líderes buscan dependientes, y cómo éstos se van uniendo al grupo y siguiendo a su líder.



Figura 6.4: Varios agentes dependientes

6.2. Agentes dependientes

En este caso se introduce la presencia de agentes dependientes. Se ha incluido la presencia de varios dependientes (ver figura 6.4, de amarillo) por el escenario. Los agentes se quedan quietos, sin saber qué hacer, hasta que un líder (de color azul) les llama. En ese momento deciden unirse a él, y por tanto comienzan a seguirle, como se intuye en la figura 6.5.



Figura 6.5: El líder forma un grupo con los agentes dependientes a los que se aproxima

Finalmente todos los agentes llegan a la salida sanos y salvos.

En este caso, el log resulta aún más largo que en el ejemplo anterior. Como hay grupos, se hace el tratamiento del log creando el agente *Group-nombreSeguidor*, tal y como se explica en el capítulo 5. De esta forma, se

genera el siguiente informe final:

```
Fire1 en el cuadrante 25 inicio-foco (34,57)
Leader1 en el cuadrante 30 buscar-dependientes
Leader1 en el cuadrante 30 empezar-en
Leader1 en el cuadrante 30 andar-hacia 41
Group-Leader1 en el cuadrante 30 creacion-grupo
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 6
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 8
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 10
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 12
Group-Leader1 en el cuadrante 30 andar-hacia 41
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 13
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 14
Fire1 en el cuadrante 26 acceso 26
Fire1 en el cuadrante 24 acceso 24
Group-Leader1 en el cuadrante 31 aumentar-liderazgo 16
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 18
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 20
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 22
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 24
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 26
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 28
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 30
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 32
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 34
Group-Leader1 en el cuadrante 48 aumentar-liderazgo 36
Group-Leader1 en el cuadrante 41 llega-a-meta
Group-Leader1 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 27 acceso 27
```

Modificando los valores umbral, se obtiene el informe sin datos sobre el aumento del liderazgo del grupo:

```
Fire1 en el cuadrante 25 inicio-foco (34,57)
Leader1 en el cuadrante 30 buscar-dependientes
Leader1 en el cuadrante 30 empezar-en
Leader1 en el cuadrante 30 andar-hacia 41
Group-Leader1 en el cuadrante 30 creacion-grupo
Group-Leader1 en el cuadrante 30 andar-hacia 41
Fire1 en el cuadrante 26 acceso 26
Fire1 en el cuadrante 24 acceso 24
Group-Leader1 en el cuadrante 41 llega-a-meta
Group-Leader1 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 27 acceso 27
```

6.3. Cambio de líder

Como se ha explicado en el capítulo 4, los agentes dependientes seguirán a un líder hasta que encuentren a otro que supere el liderazgo del anterior. Este liderazgo no depende sólo de un valor inicial, sino que el liderazgo aumenta al aumentar el número de seguidores. En este ejemplo se ilustra el caso de un agente que cambia de líder. Como se ve en la figura 6.6, en la simulación hay 2 líderes, uno cerca del otro, pero en distintas aulas. El líder señalado con la letra *A* tiene un grado de liderazgo inicial igual a 4, mientras que el líder marcado con la letra *B* tiene un liderazgo igual a 20. Como se ve en la figura 6.7, el líder *A* ha formado un grupo con 14 seguidores, mientras que el líder *B* solamente es seguido por un agente dependiente, marcado en verde por la letra *C*. Debido a ello, en ese instante el liderazgo del agente *A* habrá aumentado proporcionalmente a su número de seguidores, y, dado que a *B* sólo le sigue un agente, ahora el liderazgo de *A* es mayor que el de *B*. Por eso en el momento en que ambos líderes se encuentren en el mismo



Figura 6.6: Situación inicial de los agentes



Figura 6.7: Los líderes conducen a los seguidores a la salida

cuadrante, el agente dependiente *C* deja de seguir a *B*, aunque ya está al lado de la salida, y pasa a seguir a *A*, como se ve en la figura 6.8.

Este caso se entiende más fácilmente en el informe generado. El informe indica el cambio de liderazgo y la creación y disolución de los grupos. El informe queda como se muestra a continuación:

```

Leader1 en el cuadrante 30 empezar-en
Leader1 en el cuadrante 30 andar-hacia 41
Leader2 en el cuadrante 38 empezar-en
Leader2 en el cuadrante 38 andar-hacia 41
Group-Leader1 en el cuadrante 30 creacion-grupo
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 6
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 8
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 10
Group-Leader1 en el cuadrante 30 aumentar-liderazgo 12
Fire1 en el cuadrante 26 acceso 26
Group-Leader1 en el cuadrante 31 aumentar-liderazgo 14

```



Figura 6.8: El agente dependiente C cambia de líder

```

Fire1 en el cuadrante 24 acceso 24
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 16
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 18
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 20
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 22
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 24
Group-Leader1 en el cuadrante 33 buscar-dependientes
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 26
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 28
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 30
Group-Leader1 en el cuadrante 33 aumentar-liderazgo 32
Group-Leader2 en el cuadrante 48 creacion-grupo
Group-Leader2 en el cuadrante 48 aumentar-liderazgo 22
Group-Leader2 en el cuadrante 49 disminuir-liderazgo 22
Leader2 en el cuadrante 49 deshacer-grupo 22
Group-Leader1 en el cuadrante 49 aumentar-liderazgo 34
Leader2 en el cuadrante 41 llega-a-meta
Leader2 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 23 acceso 23
Group-Leader1 en el cuadrante 41 llega-a-meta
Group-Leader1 en el cuadrante 41 permanecer-lugar-seguro

```

6.4. Desobediencia

En esta ocasión a un caso parecido al de la simulación anterior le hemos añadido un agente independiente, caracterizado por un punto gris, que aparece en la figura 6.9 junto a un líder, en el aula inferior izquierda. Además en el escenario aparecen otros agentes, líderes y dependientes. Los agentes independientes son capaces de llegar a la salida sin necesidad de líderes que les guíen, pero también tienen la capacidad de decidir en un momento dado faltar a las normas de evacuación, alejándose de la salida. El agente independiente se dirige junto al líder a la salida más cercana a su posición, la salida izquierda, pero cuando está a punto de llegar (figura 6.10), se acuerda de que se ha dejado algo en un aula, y se dirige a por ello, desoyendo las órdenes del líder (figura 6.11). Una vez que ha recogido lo que quería, vuelve a dirigirse a la salida. Como ahora la salida más cercana es la derecha, es hacia allí hacia donde se dirige (figura 6.12), y consigue salvarse a tiempo,



Figura 6.9: Simulación con agente independiente

aunque esta vez por poco. Mientras, el resto de los agentes hace tiempo que están a salvo.

Con la herramienta FL2ST, lo que se espera en este caso es, en primer lugar, que tenga en cuenta el grupo que se crea (arriba), y, sobre todo, que se describan atentamente las acciones del agente que desobedece las normas de seguridad, y a mitad de la evacuación, cambia de objetivo.

El log obtenido es el siguiente:

```

Fire1 en el cuadrante 25 inicio-foco (34,57)
Fire1 en el cuadrante 25 acceso 25
Leader1 en el cuadrante 30 empezar-en
Leader1 en el cuadrante 30 andar-hacia 41
Leader3 en el cuadrante 14 empezar-en
Leader3 en el cuadrante 14 andar-hacia 12
Leader2 en el cuadrante 38 empezar-en
Leader2 en el cuadrante 38 andar-hacia 41
Group-Leader1 en el cuadrante 30 creacion-grupo
Fire1 en el cuadrante 26 acceso 26
Independent1 en el cuadrante 14 empezar-en
Independent1 en el cuadrante 14 andar-hacia 12
Group-Leader1 en el cuadrante 31 aumentar-liderazgo 8
Independent1 en el cuadrante 14 salir-de
Independent1 en el cuadrante 13 entrar-en
Group-Leader1 en el cuadrante 32 aumentar-liderazgo 10
Independent1 en el cuadrante 13 desobedecer-salir-de
Independent1 en el cuadrante 5 desobedecer-entrar-en
Independent1 en el cuadrante 5 entrar-en
Fire1 en el cuadrante 24 acceso 24
Independent1 en el cuadrante 4 desobedecer
Independent1 en el cuadrante 5 desobedecer-entrar-en
Leader3 en el cuadrante 12 llega-a-meta
Leader3 en el cuadrante 12 permanecer-lugar-seguro
Independent1 en el cuadrante 5 desobedecer-salir-de
Independent1 en el cuadrante 23 desobedecer-entrar-en
Leader2 en el cuadrante 41 llega-a-meta
Leader2 en el cuadrante 41 permanecer-lugar-seguro
Group-Leader1 en el cuadrante 41 llega-a-meta
Group-Leader1 en el cuadrante 41 permanecer-lugar-seguro
Independent1 en el cuadrante 23 desobedecer-salir-de
Independent1 en el cuadrante 34 desobedecer-entrar-en
Independent1 en el cuadrante 34 desobedecer-salir-de
Independent1 en el cuadrante 35 desobedecer-entrar-en
Independent1 en el cuadrante 35 desobedecer-salir-de
Independent1 en el cuadrante 36 desobedecer-entrar-en
Independent1 en el cuadrante 36 entrar-en
Independent1 en el cuadrante 36 meta
Independent1 en el cuadrante 37 salir-de

```



Figura 6.10: Líder e independiente se dirigen hacia la salida, pero independiente se da la vuelta

```

Independent1 en el cuadrante 36 entrar-en
Independent1 en el cuadrante 36 entrar-en
Independent1 en el cuadrante 36 salir-de
Independent1 en el cuadrante 35 entrar-en
Independent1 en el cuadrante 35 colision
Independent1 en el cuadrante 35 salir-de
Independent1 en el cuadrante 34 entrar-en
Independent1 en el cuadrante 34 colision
Independent1 en el cuadrante 34 salir-de
Independent1 en el cuadrante 50 entrar-en
Independent1 en el cuadrante 50 salir-de
Independent1 en el cuadrante 49 entrar-en
Fire1 en el cuadrante 27 acceso 27
Independent1 en el cuadrante 49 salir-de
Independent1 en el cuadrante 41 entrar-en
Independent1 en el cuadrante 41 llega-a-meta
Independent1 en el cuadrante 41 permanecer-lugar-seguro
Fire1 en el cuadrante 28 acceso 28
Fire1 en el cuadrante 5 acceso 5

```

6.5. Muerte de un agente

En este caso se muestra un ejemplo sencillo de un agente que es alcanzado por el fuego. En este caso la muerte era prácticamente ineludible, como se ve en la figura 6.13, pero sirve para estudiar el caso genérico de la muerte de un agente. En la figura 6.14 se observa cómo el agente intenta escapar del fuego, pero finalmente, éste le alcanza y el agente muere (figura 6.15). En este caso, dado que el agente acaba muriendo, se desea que el informe final muestre más información sobre el agente del que sería aconsejable en otro caso. El informe obtenido es:

```

Fire1 en el cuadrante 23 inicio-foco (36,44)
Independent1 en el cuadrante 26 empezar-en
Independent1 en el cuadrante 26 andar-hacia 41
Independent1 en el cuadrante 26 empezar-en
Independent1 en el cuadrante 26 andar-hacia 41
Independent1 en el cuadrante 26 salir-de
Independent1 en el cuadrante 25 entrar-en
Independent1 en el cuadrante 25 entrar-en

```



Figura 6.11: El agente independiente se ha dado la vuelta y se ha dirigido a otro aula

```

Independent1 en el cuadrante 25 desobedecer-salir-de
Independent1 en el cuadrante 24 desobedecer-entrar-en
Independent1 en el cuadrante 24 entrar-en
Fire1 en el cuadrante 27 acceso 27
Independent1 en el cuadrante 24 visto-fuego
Independent1 en el cuadrante 24 andar-hacia 12
Independent1 en el cuadrante 24 visto-fuego
Independent1 en el cuadrante 24 andar-hacia 41
Fire1 en el cuadrante 24 acceso 24
Independent1 en el cuadrante 24 visto-fuego
Independent1 en el cuadrante 24 andar-hacia 12
Independent1 en el cuadrante 24 visto-fuego
Independent1 en el cuadrante 24 andar-hacia 41
Fire1 en el cuadrante 28 acceso 28
Independent1 en el cuadrante 24 visto-fuego
Independent1 en el cuadrante 24 andar-hacia 12
Independent1 en el cuadrante 24 visto-fuego
Independent1 en el cuadrante 24 andar-hacia 41
Fire1 en el cuadrante 25 acceso 25
Fire1 en el cuadrante 34 acceso 34
Fire1 en el cuadrante 5 acceso 5
Independent1 en el cuadrante 24 accidente muerte
Fire1 en el cuadrante 35 acceso 35
Fire1 en el cuadrante 13 acceso 13
  
```

En el informe se ve cómo el agente, viendo el fuego, intenta dirigirse una y otra vez hacia otro lado para huir de él, pero no es posible, y al final es alcanzado por él.



Figura 6.12: El agente independiente se dirige a la nueva salida



Figura 6.13: Posición inicial del agente y del fuego



Figura 6.14: El agente intenta huir del fuego, pero está muy cerca



Figura 6.15: El agente es alcanzado por el fuego y muere

Capítulo 7

Conclusiones y trabajo futuro

En este capítulo se analizan los resultados obtenidos una vez finalizado el estudio y los sistemas. Por cada sistema y objetivo, se propone el trabajo futuro.

7.1. Simulador Multi Agente

Se ha diseñado y analizado un sistema multi-agente que simula el comportamiento de las personas involucradas en una evacuación de incendios en un edificio. Se han simulado tres tipos de comportamiento: El líder, que ayuda a los demás a salir del escenario, el dependiente, sin capacidad para salir por cuenta propia, y que necesita que un líder le ayude, y el independiente, con capacidad de evacuar por sí mismo, pero que en ocasiones decide poner su vida en peligro saltándose las normas de evacuación. Este último tipo de agente es el que se considera más importante, puesto que, como se ve en el capítulo de trabajo relacionado 2, es un tipo de comportamiento que no se tiene en cuenta en otras simulaciones con SMA.

7.1.1. Escenario configurable

Para aumentar la escalabilidad del sistema, el escenario en el que se desarrolla la simulación completa es configurable. Para cambiar el mapa del escenario en el que se desarrolle la simulación, se debe modificar un fichero de configuración que toma el sistema como entrada. En él se especifican, tal y como se esperaba, el nombre del fichero con el dibujo del mapa, las coordenadas de las salidas, el número de agentes y las posiciones iniciales de los mismos. Esto último no es necesario, en el caso de no especificar las posiciones, éstas se generarán aleatoriamente a lo largo de todo el mapa. Además de todo esto, hace falta también configurar las coordenadas de los

cuadrantes. Como se ha visto en el capítulo 4, la percepción de cada agente se limita al cuadrante o sector en el que se encuentre. Por tanto, no es posible generar automáticamente la malla en que se divide el escenario, puesto que hay muchos factores que pueden hacer variar los sectores deseados, tales como el tamaño de las habitaciones, la visibilidad que se quiera proporcionar a los agentes, los pasillos, etc. Las coordenadas se añaden en el mismo fichero de configuración.

Como trabajo futuro se propone la creación de una interfaz amigable para la configuración completa del escenario, los cuadrantes, el tamaño de las celdas y los agentes de cada tipo. Hoy por hoy toda esta información se saca de ficheros de configuración, que, si bien son modificables, un usuario no habituado no sabrá automáticamente como modificarlo, y además, no se puede cambiar en tiempo de ejecución.

7.1.2. Búsqueda de caminos: pathfinding

Para el SMA se han desarrollado tres tipos distintos de búsquedas de caminos. Cada uno se utiliza en una situación para garantizar conseguir los mejores resultados en el menor tiempo. Cuando el agente necesita saber por qué zonas pasar utiliza el algoritmo de Dijkstra, que calcula el camino de cuadrantes. Posteriormente, para saber qué celdas debe recorrer el agente en un cuadrante para llegar al siguiente de su lista de cuadrantes (es decir, de su camino a la salida) puede dirigirse directamente hacia la salida, si fuera posible, o utilizar el algoritmo A* en el caso de encontrarse con obstáculos estáticos. Para dirigirse directamente a la salida se ha utilizado un algoritmo dinámico que calcula x pasos hacia el destino, y se mueve en función de la detección de colisiones.

Gracias a la creación de las celdas las búsquedas resultan bastante rápidas, alcanzando así el objetivo propuesto. Sin embargo, este método tiene algunos ajustes que realizar. A la hora de elegir el objetivo para pasar de un cuadrante a otro el agente no sabe hacia dónde se dirigirá en el siguiente cuadrante, por lo que no lo tiene en cuenta. Como resultado, se pueden dar situaciones de este tipo: un agente que al salir de un aula se dirigirá hacia la derecha (para llegar a la salida de la derecha) o, por estar situado ligeramente más cerca de la puerta del aula de la izquierda, sale por la izquierda, teniendo que andar más después para acercarse a la salida que le interesa. Este problema, sin embargo, es fácilmente solucionable, y no daña los resultados de las simulaciones.

7.1.3. Detección de colisiones y creación de grupos

Los grupos se crean cuando varios agentes dependientes se unen al mismo líder, y lo siguen. El agente líder le envía las coordenadas actuales cada vez que se mueve a sus seguidores, y éstos se intentan acercar. La detección de colisiones se hace de la siguiente manera: Suponiendo que el agente seguidor se quisiera mover hacia el Suroeste para acercarse al líder, si el agente no puede dirigirse hacia allí porque la posición inferior izquierda adyacente al agente estuviera ocupada, prueba lo siguiente. Intenta moverse bien al Este, bien al Sur (las siguientes posiciones en orden de cercanía a la deseada). Si alguna de las posiciones está libre, la ocupa, y si no, se queda quieto, esperando a que se libere.

Es posible que en el momento de tomar la decisión una cierta posición esté libre, pero que al ir a moverse no lo esté. En ese caso se produce una colisión, y el agente no llega a moverse.

Este sistema de detección de colisiones podría sofisticarse, añadiendo predicción. Para ello sería necesario conocer la dirección de cada agente, tal y como ocurre en [Foudil et al., 2009].

7.1.4. Roles y comportamientos

Como se ha visto en el capítulo anterior, los tres roles existentes están claramente definidos, y han sido ampliamente explicados, por lo que se comentan los resultados en dos secciones, por un lado líderes y dependientes, ya que están íntimamente relacionados, y por otro lado independientes.

Líderes y dependientes

Los líderes, cuando empieza la simulación, se dirigen hacia la salida, mientras que por el camino buscan a los dependientes. Les ofrecen ayuda a todos los que se encuentren en el mismo cuadrante que ellos. Los dependientes se unirán al líder que más liderazgo posea. En el caso de seguir a un líder y cruzarse con otro, si el nuevo líder tiene más liderazgo que el primero, los dependientes se unirán a éste.

Cabe destacar que el liderazgo no es sólo un valor inicial introducido como variable de configuración, sino que varía a lo largo de la simulación. Si un líder consigue un nuevo seguidor, su liderazgo aumenta, mientras que si lo pierde disminuye. Así se puede observar en el ejemplo de simulación comentado en la sección 6.3. El líder con mayor liderazgo inicial, 20, pierde a su seguidor cuando se cruza con el otro líder, puesto que, a pesar de tener un liderazgo inicial bajo, 4, a lo largo del trayecto hasta cruzarse con el otro

líder ha conseguido superar el grado del primero.

Por otra parte, el comportamiento de los líderes cuando no encuentran ningún líder cerca es muy limitado, puesto que se quedan esperando a que aparezca un líder, y piden ayuda, sin llegar a moverse. Esto se podría mejorar, añadiendo algún comportamiento en el que se movieran.

Los líderes, por su parte, podrían ir a buscar dependientes a otras áreas del escenario.

Independientes

Los agentes independientes se consideran un logro bastante importante del SMA. Sus características son similares a las de los líderes, aunque sin buscar dependientes. Pero, en ocasiones, estos agentes deciden cambiar de rumbo y dirigirse hacia otro punto del mapa, para ir a buscar algún objeto, o porque, debido al pánico, pierde el sentido de la orientación y se equivoca de destino. Una vez haya llegado hasta allí, ya sea porque ha recogido el objeto o porque se ha dado cuenta de dónde está, vuelve a dirigirse a la salida más cercana.

Este tipo de comportamiento se ha considerado bastante habitual en las evacuaciones, y es muy arriesgado, por lo que hay que tenerlo en cuenta a la hora de diseñar un plan de evacuación, y de educar a los evacuadores.

7.2. From Log To Story Telling

Se ha propuesto un sistema que genera un resumen de un conjunto de eventos acaecidos en una simulación de evacuaciones. En una primera fase se eligen los personajes más relevantes de la simulación, y de ellos a su vez, se cuentan sus acciones que resulten más interesantes, teniendo en cuenta también el contexto en suceden. Después, una vez seleccionados los eventos que serán narrados se elige el orden en que se contarán. Para ello se ordenan por escenarios en que ocurren y tiempo. Se tienen en cuenta a los independientes, que desobedecen las normas de evacuación.

En primer lugar, y más importante, se deja como trabajo futuro la adaptación de la salida como entrada de TAP. Hasta ahora no se ha convertido el log en texto de lenguaje natural, sino que se muestra un informe con el formato mostrado en el capítulo anterior. Este formato se puede modificar añadiéndole más información.

Como trabajo futuro se plantea el ajuste de los parámetros del factor de interés. Por otro lado, se puede hacer un refinamiento de la fase de *Pla-*

nificador de Discurso para mejorar el orden en que se cuentan las acciones sucedidas durante la simulación, principalmente de aquellos personajes que no cumplen los protocolos de evacuación y se alejan de la salida. Un posible enfoque es de las escenas, en el que se empieza o termina de hablar de un escenario cuando entra o sale de él algún individuo.

Otra posible mejora sería la adaptación del sistema a cualquier tipo de log. Es una tarea sencilla ya que todo el preprocesamiento del log se realiza en la misma clase, y sólo habría que cambiar esa clase. Las acciones y sus intereses intrínsecos se encuentran en una clase estática, por lo que cambiarlas resultaría sencillo y no alteraría el sistema. Sin embargo, el nuevo log debería contar al menos con los campos correspondientes a sujeto, tiempo, acción y complemento. En otro caso el sistema no se adaptaría fácilmente.

Apéndice A

JADE: Java Agent DEvelopment Framework

JADE (Java Agent DEvelopment Framework)¹ es un framework completamente implementado en Java. Simplifica la implementación de Sistemas Multi-Agente a través de un programa intermedio que cumple las especificaciones FIPA² y proporciona un conjunto de herramientas gráficas que permiten la depuración y fases de despliegue.

JADE proporciona:

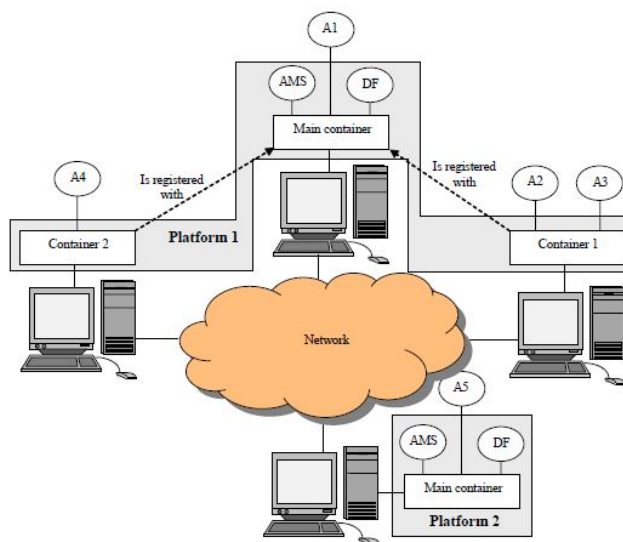


Figura A.1: “Containers” y “Platforms”

¹JADE está disponible en <http://jade.tilab.com/>

²FIPA, foundation fo Intelligent Physical Agents es una organización de estándares de IEEE Computer Society que promueve la tecnología basada en agentes y la interoperabilidad de sus estándares con otras tecnologías. Mas información en <http://www.fipa.org/>.

- Un **Runtime Environment** (entorno en tiempo real) donde los agentes JADE pueden “vivir” y que debe estar activo en un host antes de que uno o más agentes sean ejecutados en ese host.
- Una **librería** de clases que los programadores pueden/deben usar para desarrollar sus agentes.
- Un conjunto de **herramientas gráficas** que permiten la administración y monitorización de actividad de los agentes en ejecución.

Cada instancia en ejecución del JADE runtime environment se llama **Contained** (contenedor) y puede contener muchos agentes (ver Fig. A.1). El grupo de contenedores activos se llama **Platform** (plataforma). Un único y especial **Main container** (contenedor principal) debe estar siempre activo en una plataforma y el resto de contenedores se registran en él cuando se inician.

El modelo estándar de una plataforma, como se define por FIPA, se representa en la Fig. A.2:

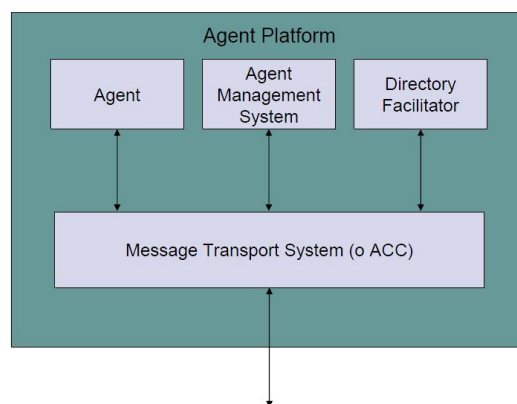


Figura A.2: Arquitectura de una plataforma de agentes según FIPA

Agent Management System (AMS) proporciona el servicio de nombres (se asegura de que cada agente en la plataforma tiene un nombre único) y representa la autoridad en la plataforma (control de supervisión sobre el acceso y uso de la plataforma de agentes).

Directory Facilitator (DF) proporciona el servicio de PÁGINAS AMARILLAS, por medio del cual un agente puede encontrar otros agentes que presten los servicios que necesite para lograr sus objetivos.

A.1. Comportamientos en agentes

Un **comportamiento** representa una tarea que un agente puede llevar a cabo. Para hacer que un agente ejecute la tarea implementada por un objeto comportamiento es suficiente añadir el comportamiento al agente.

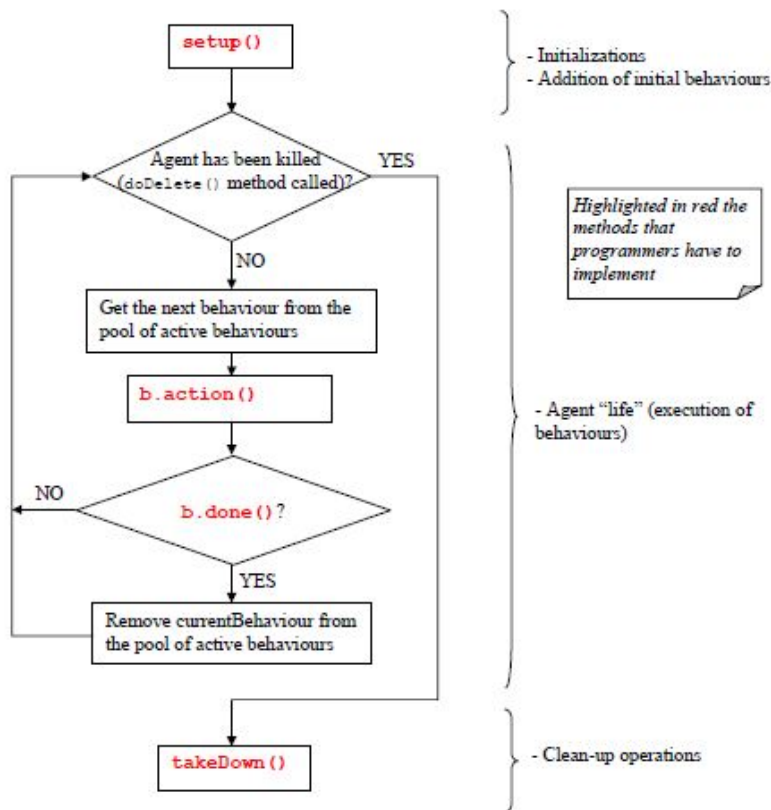


Figura A.3: Hilo de ejecución del agente

La programación de los comportamientos no es preventiva sino cooperativa. Esto significa que cuando un comportamiento está previsto para ejecutarse, se llama al método que lo invoca, que se ejecuta hasta que dicho método termina. En la Fig. A.3 se muestra el hilo de ejecución de un agente. Cuando no hay comportamientos accesibles para un agente, su hilo de ejecución se duerme hasta que vuelva a haber algún comportamiento ejecutable.

A.1.1. Tipos de comportamientos

Básicamente hay dos tipos de clases de comportamiento: primitivos, como el *Simple* o *Cyclic*, y compuestos, que se componen de varios comporta-

mientos para que sean ejecutados secuencialmente o en paralelo.

Comportamientos primitivos

Jade tiene tres clases abstractas de comportamientos primitivos:

- **SimpleBehaviour**. La clase básica de comportamiento de la que se puede extender para crear comportamientos más complejos.
- **One Shot Behaviour**. La clase *OneShotBehaviour* modela comportamientos atómicos simples, que sólo se ejecutan una vez. El método *done()* (ver figura A.3) está ya implementado, devolviendo siempre *true*.
 - *WakerBehaviour*: Es invocado en un momento determinado
 - *ReceiverBehaviour*: Se ejecuta cuando se recibe algún tipo específico de mensaje, o cuando expira un *timeout*.
- **Cyclic Behaviour**. Nunca termina, el comportamiento se mantiene mientras viva el agente, y es invocado cada cierto tiempo.
 - *TickerBehaviour* Un comportamiento cíclico que ejecuta las mismas operaciones cada vez que se le llama. La clase implementa *done()* devolviendo siempre *false*.

Comportamientos compuestos

Estos tipos de comportamientos están compuestos por un número de otros comportamientos “hijo”. La clase *CompositeBehaviour* proporciona una interfaz común, pero no define la política de organización entre comportamientos. Esta política se define por las subclases:

SequentialBehaviour Esta clase es un comportamiento compuesto que que ejecuta sus comportamientos hijo secuencialmente y termina cuando todos los subcomportamientos se terminan. Son apropiados para modelar comportamientos complejos, subdividiéndolos en varios comportamientos más simples.

ParallelBehaviour Esta clase ejecuta sus sub-comportamientos concurrentemente, y termina cuando se cumple alguna condición particular de éstos comportamientos hijo. La clase proporciona las constantes adecuadas para que el comportamiento compuesto termine cuando todos sus sub-comportamientos finalizan, cuando termine uno en especial, o cuando un número N definido por el usuario de sub-comportamientos finalizan.

FSMBehaviour Esta clase ejecuta a sus comportamientos hijo de acuerdo con una Máquina de Estados Finito (en inglés, *Finite Static Machine, FSM*), definido por el usuario. Cada sub-comportamiento representa la actividad que se debe ejecutar dentro de cada estado del FSM, y el usuario define las transiciones entre los estados del FSM. Cuando un comportamiento hijo que se corresponde con un estado S_j termina, su valor de terminación (devuelto por el método *onEnd()*) se utiliza para seleccionar la transición que se escoge y el nuevo estado S_j que se alcanza. En la siguiente vuelta se ejecutará el sub-comportamiento correspondiente al método S_j . Algunos sub-comportamientos se registran como estados finales, así, cuando alguno de ellos termina, el *FSMBehaviour* finaliza.

A.2. Comunicación entre agentes

Los agentes en JADE se comunican mediante **paso asíncrono de mensajes**. Cuando un agente recibe un mensaje, éste se coloca al final de la cola de mensajes (en la Fig. A.4, “A2’s message queue”), y se le notifica.

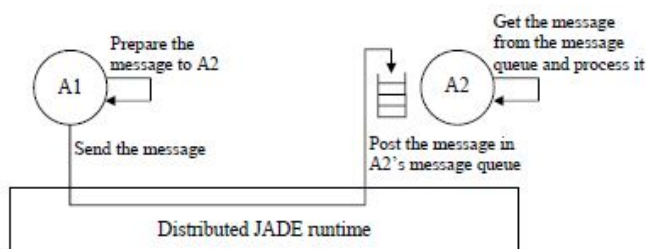


Figura A.4: Paso de mensajes entre agentes

Los mensajes en JADE tienen un formato especificado por el lenguaje ACL (Agent Communication Language), definido por la FIPA. Este formato tiene varios campos:

- El emisor del mensaje.
- La lista de receptores.
- La intención de la comunicación. Puede ser:
 - REQUEST: Si el emisor quiere que el receptor realice una acción.
 - INFORM: Si el emisor quiere advertir de algo al receptor.
 - QUERY_IF: Si el emisor quiere saber si se cumple una condición.
 - CFP: Implica una convocatoria de propuesta.
 - PROPOSE,

- ACCEPT_PROPOSAL,
- REJECT_PROPOSAL: Si emisor y receptor están comprometidos en una negociación.
- El contenido.
- El lenguaje del contenido: la sintaxis utilizada.
- La ontología con el vocabulario de los símbolos utilizados en el contenido y su significado.
- Campos de control para recurrencia en conversaciones y para especificar *timeouts*, como *conversation-id*, *reply-with*, *in-replay-to*, *replay-by*.

especificando una determinada propiedad (*template*), un agente puede seleccionar el primer mensaje que cumpla ciertas características, ignorando el resto.

A.3. El servicio de Páginas Amarillas

El servicio de Páginas Amarillas (Fig. A.5) permite a los agentes publicar uno o más servicios que proporcionan, y otros agentes pueden encontrar a los agentes que proporcionen los servicios que necesiten mediante plantillas. El servicio lo proporciona un agente llamado DF (*Directory Facilitator*).

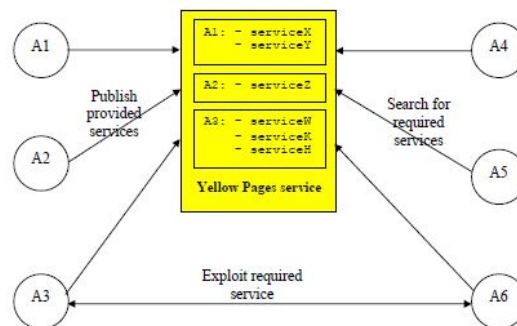


Figura A.5: Servicio de Páginas Amarillas

Cada plataforma FIPA aloja a un Agente DF por defecto, pero se pueden activar más. Para cada servicio publicado se proporciona una descripción que incluye el tipo de servicio, el nombre del servicio, el lenguaje y ontologías necesarias para utilizarlo, y un número de propiedades específicas del servicio.

Apéndice B

Algoritmos de búsqueda. Pathfinding

En este apéndice se muestran en pseudocódigo los algoritmos de búsqueda utilizados para la simulación de las evacuaciones. Cuando un Agente Líder o Independiente quiere conocer el camino de salida se lo pide al ENTORNO, y éste le devuelve el camino en sectores desde la posición actual del agente hasta la salida que se considere más cercana. Este camino se halla con el Algoritmo de Dijkstra. A partir de él se crea un submapa, y mediante el Algoritmo A* se halla el camino exacto, a nivel de celda (x, y) , hasta la salida.

B.1. Algoritmo de Dijkstra

A continuación se muestra el pseudocódigo utilizado para hallar el camino mínimo entre dos nodos.

```
//Función Dijkstra (Grafo G, nodo-salida s)
//Usaremos un vector para guardar las distancias del nodo salida al resto
int distancia[n] //Inicializamos el vector con distancias iniciales
booleano visto[n] //vector de booleanos para controlar los vertices de los que
//ya tenemos la distancia mínima

para cada w e V[G] hacer
  Si (no existe arista entre s y w) entonces
    distancia[w] = Infinito
  Sino
    distancia[w] = peso(s, w)
  fsi
fpara
distancia[s] = 0
visto[s] = cierto
//n es el número de vertices que tiene el Grafo
mientras (no_esten_vistos_todos) hacer
  vertice = coger_el_minimo_del_vector distancia y que no este visto;
  visto[vertice] = cierto;
  para cada w e sucesores (G, vertice) hacer
    si distancia[w]>distancia[vertice]+peso (vertice, w) entonces
      distancia[w] = distancia[vertice]+peso (vertice, w)
  fsi
fpara
fmientras
finfuncion
```

B.2. Algoritmo A*

El Algoritmo A* o A Estrella es un algoritmo de búsqueda para grafos que encuentra el camino de menor coste entre un nodo inicial y un nodo meta. Fue presentado por primera vez en 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael. Utiliza una función de evaluación $f(n) = g(n) + h'(n)$, donde $h'(n)$ representa el valor heurístico del nodo a evaluar desde el actual, n , hasta el final; y $g(n)$, el coste real del camino recorrido para llegar a dicho nodo n . A* mantiene dos estructuras de datos auxiliares, que podemos denominar *abiertos*, una cola ordenada por el valor $f(n)$ de cada nodo, y *cerrados*, donde se guarda la información de los nodos que ya han sido visitados. En cada paso del algoritmo, se expande el nodo que esté primero en *abiertos*, y en caso de que no sea un nodo objetivo, calcula la $f(n)$ de todos sus hijos, los inserta en *abiertos*, y pasa el nodo evaluado a *cerrados*. A continuación se presenta el pseudocódigo del algoritmo A*, que calcula el camino entre el nodo de salida, INICIAL, y el objetivo, FINAL:

```
//Función A*. Listas de nodos: ABIERTOS, CERRADOS
//Nodo inicial: INICIAL.
list ABIERTOS = [INICIAL] //inicialización de ABIERTOS con el nodo inicial
list CERRADOS = [] //vacío
f'(INICIAL) = h'(INICIAL)
mientras (no_fallo AND no_solucion_encontrada)
    si (ABIERTOS = [])
        entonces FALLO;
    si no // quedan nodos
        // cola de prioridad
        nodo MEJOR_NODO = extraer_nodo_de_ABIERTOS_con_f'_mínima
        mover_MEJORNODO_de_ABIERTOS_a_CERRADOS
        si MEJORNODO.es_FINAL entonces
            SOLUCION_ENCONTRADA = TRUE
        si no
            generar_SUCESTORES_de_MEJORNODO
            para_cada_SUCESOR_hacer_TRATAR_SUCESOR // detallado abajo
            hasta_SOLUCION_ENCONTRADA_o_FALLO
        fsi
    fsmientras

//Función TRATAR_SUCESOR
SUCESOR.ANTERIOR = VIEJO
caso SUCESOR = VIEJO perteneciente a CERRADOS
    si g(SUCESOR) < g(VIEJO) entonces // (no si monotonía)
        // nos quedamos con el camino de menor coste
        VIEJO.ANTERIOR = MEJORNODO
        actualizar g(VIEJO) y f'(VIEJO)
        propagar g a sucesores de VIEJO
        eliminar SUCESOR
        añadir VIEJO a SUCESTORES_MEJORNODO
caso SUCESOR = VIEJO perteneciente a ABIERTOS
    si g(SUCESOR) < g(VIEJO) entonces
        // nos quedamos con el camino de menor coste
        VIEJO.ANTERIOR = MEJORNODO
        actualizar g(VIEJO) y f'(VIEJO)
        eliminar_SUCESOR
        añadir_VIEJO_a_SUCESTORES_MEJORNODO
caso SUCESOR no estaba en ABIERTOS ni CERRADOS
    añadir_SUCESOR_a_ABIERTOS
    añadir_SUCESOR_a_SUCESTORES_MEJORNODO
f'(SUCESOR) = g(SUCESOR) + h'(SUCESOR)
```

Apéndice C

Intenciones y acciones

En este capítulo se describen todas las intenciones y acciones que pueden llevar a cabo los agentes, y la Máquina de Estados Finita que determina las intenciones que lleva a cabo cada agente.

C.1. Intenciones y acciones

Las intenciones, como se explica en la Sección 4.1.1, es un objetivo de alto nivel que un agente intenta alcanzar.

Las intenciones que puede intentar alcanzar un líder son: `Buscar gente que necesita ayuda`, `Dejar dependientes en lugar seguro` y `Permanecer en lugar seguro`.

Las intenciones de un dependiente pueden ser: `Buscar líderes y grupos`, `Unirse a grupo`, `Permanecer en lugar seguro`.

Las intenciones de un normal pueden ser: `Desalojo`, `Unirse a grupo`, `Desobedecer` y `Permanecer en lugar seguro`.

Por otra parte, las acciones son tuplas del tipo `<movimiento, interacción>`. A continuación se especifican los tipos de movimiento:

- `Ir-a-lugar-concurrido`
- `Ir-a-lugar-seguro`
- `Ir-a-lugar-peligroso`
- `Ir-a-lugar-aleatorio`
- `Ninguno`

Las definiciones de lugares concurridos, seguros y peligrosos se obtienen del entorno.

Las interacciones consisten en acciones de bajo nivel que lleva a cabo un agente y que afectan al conocimiento de los agentes que se encuentren dentro de un radio de interacción determinado. Normalmente afecta a aquellos que se encuentren dentro del mismo sector (ver Sección 4.1.2). Estas interacciones pueden tener en algunos casos un complemento, y son:

- Ordenar. Esta interacción no tiene complemento.
- Dejar. Indica cuando el líder de un grupo deja a su grupo en algún lugar.
- Buscar-grupo.
- Seguir-a + complemento.
- Dejar-de-seguir-a + complemento. El complemento indica en los dos casos al líder del grupo.
- Ignorar-a + complemento. El complemento es el identificador de quien anteriormente ha dado la orden.
- Ninguno. Interacción vacía.

En el Cuadro C.1 se indica la correspondencia entre intenciones y acciones. Cuando aparece más de una tupla, es que hay más de una correspondencia. Durante la ejecución se elegirá una de las tuplas aleatoriamente para evitar que todas las simulaciones sean iguales.

Intención	Acción
Buscar gente que necesite ayuda	<Ir-a-lugar-concurrido, Ordenar> <Ir-a-lugar-peligroso, Ordenar> <Ninguno, Ordenar>
Dejar dependientes en lugar seguro	<Ir-a-lugar-seguro, Dejar>
Permanecer en lugar seguro	<Ninguno, Ninguno >
Buscar líderes y grupos	<Ir-lugar-aleatorio, Buscar-grupo> <Ninguno, Buscar-grupo>
Unirse	<Ninguno, Seguir-a>
Desalojo	<Ir-a-lugar-seguro, Ninguno>
Desobedecer	<ir-a-lugar-seguro, Ignorar> <ir-a-lugar-peligroso, Ignorar>

Cuadro C.1: Correspondencia entre intenciones y acciones

Apéndice D

Listado de acciones y parámetros de interés

Con el objetivo de poder memorizar lo sucedido durante cada simulación el sistema genera un log con los eventos que tienen lugar en la misma. Como se indica en la sección 5.1, cada entrada del log tiene la forma:

[NombreAgente] Fecha y Hora IdAc Place Action Parameter

Las acciones que se pueden memorizar son de distintos tipos:

1. **Intenciones.** Objetivos que el agente quiere alcanzar:
 - Buscar gente que necesite ayuda (BGNA)
 - Dejar dependientes en lugar seguro (DDLs)
 - Permanecer en lugar seguro (PLS)
 - Buscar líderes y grupos (BLG)
 - Unirse a grupo (UG)
 - Permanecer en lugar seguro (PLS)
 - Desalojo
 - Desobedecer
2. **Movimientos e Interacciones.** Como se explica en la Sección 4.1.1, forman las intenciones.
 - Ir-a-lugar-concurrido
 - Ir-a-lugar-seguro
 - Ir-a-lugar-peligroso
 - Ir-a-lugar-aleatorio

- Ordenar. El agente, que será un responsable de evacuación, ordena a todos los agentes que se encuentren en el campo Lugar que le sigan. Los agentes pueden obedecer (comenzando a seguirle) o ignorar.
 - Dejar. El agente, que será el líder de un grupo, deja a los dependientes en un lugar seguro. La consecuencia será la disolución del grupo.
 - Buscar-grupo. El agente busca un líder o un grupo para unirse a él.
 - Seguir-a. Indica que el agente comienza a seguir a un líder, indicado en el Parámetro. A partir de este punto, el agente pasa a formar parte del grupo dirigido por dicho líder.
 - Dejar-de-seguir-a. El agente, que seguía al líder de parámetro, deja de hacerlo.
 - Ignorar. Ante una acción *Obedecer* del agente escrito en parámetro, el agente le ignora.
3. **Eventos de bajo nivel.** Son aquellas acciones que forman los movimientos, o que pueden ocurrir sin la intención explícita del agente.
- Nacimiento. Indica la hora de creación del agente, y por lo tanto, el inicio de la simulación. A la hora de escribir el informe, no es una acción de interés.
 - Colisión. Indica que el agente colisiona con un objeto o agente. El parámetro indica con qué colisiona el agente.
 - Accidente. El agente sufre un accidente. El parámetro tiene un valor numérico que indica la gravedad del accidente.
 - Dirigirse-a [complemento]. El complemento indica un sector.
 - Andar-hacia . Indica un cambio de estado interno del paciente. Demuestra la intención del agente de dirigirse a un sector del escenario, pero no asegura que vaya a llegar. El parámetro indica el lugar al que se dirige. Comienza a dirigirse andando, a una velocidad baja.
 - Correr-hacia. Igual que Andar-hacia, pero el agente se dirige con una velocidad alta.
 - Llegar-a. El agente llega a un sector nuevo. No tiene parámetro, puesto que el nuevo escenario queda reflejado en el campo Lugar.
 - Entrar-en
 - Creación-grupo. Cuando un agente comienza a seguir a otro que no era seguido por nadie, se ejecuta esta acción. El sujeto de la acción es el líder. No tiene parámetro.

- Disolución-grupo. Cuando el último agente que seguía a un líder deja de hacerlo, el grupo desaparece, y se añade al log esta acción. El sujeto es el líder, y no tiene líder.
- Desobedecer-andar-hacia. Todas las acciones que empiecen por desobedecer indican que el agente actúa en contra del protocolo de evacuación, dirigiéndose en sentido contrario a la ruta de salida más próxima.
- Desobedecer-correr-hacia
- Desobedecer-llegar-a
- Desobedecer-entrar-en

4. Acciones del fuego

- Inicio-foco. El fuego comienza en la zona indicada en el campo Lugar.
- Invasión. El fuego invade completamente un escenario, indicado por el campo Lugar.

D.1. Parámetros de interés

A continuación se muestra la lista final de acciones, con sus valores de interés tanto para el sujeto como para, en su caso, el complemento.

```
#InterestForSubject
subject. andar-hacia = 20
subject. correr-hacia = 2
subject. llegar-a = 5
subject. colision = 4
subject. salir-de = 5
subject. entrar-en = 5
subject. seguir-a = 10
subject. bajar = 6
subject. ordenar = 10
subject. obedecer = 10
subject. ignorar = 50
subject. dejar-de-seguir = 15
subject. accidente = 100
subject. recoger = 50
subject. creacion-grupo = 20
subject. disolucion-grupo = 20
subject. desobedecer-andar-hacia = 15
subject. desobedecer-llegar-a = 15
subject. desobedecer-salir-de = 15
subject. desobedecer-entrar-en = 15
subject. desobedecer-subir = 15
```

```
subject.desobedecer-bajar = 15
subject.aumentar-liderazgo = 50
subject.disminuir-liderazgo = 50
subject.buscar-dependientes = 8
#FUEGO
subject.inicio-foco = 100
subject.invasion = 50
subject.acceso = 30

subject.recibe-msj = 0
subject.empezar-en = 50
subject.envia-msj = 0
subject.termina = 10
subject.inicializado = 2
subject.objetivo = 2
subject.encuentra-el-camino = 10
subject.llega-a-meta = 50
subject.permanecer-lugar-seguro = 15
subject.cuadrante = 2
subject.busqueda-a-estrella = 2
subject.visto-fuego = 50

## Interés de un complemento
complement.seguir-a = 15
complement.obedecer = 15
complement.ignorar = 15
complement.dejar-de-seguir = 15
```

Apéndice E

Ejemplos de logs

En este apéndice se muestran muestras de los logs obtenidos en las simulaciones de ejemplo del capítulo 6. Con una simulación sencilla, de unos 10 agentes, hay unas mil líneas de log, por lo que se muestran sólo algunas de las líneas.

E.1. Varios líderes y el fuego

```
Fire1 2011-09-06 19:16:51,606 256 12 inicio-foco (10,30)
Fire1 2011-09-06 19:16:51,600 255 4 acceso 4
Fire1 2011-09-06 19:16:51,618 257 many envia-msj posiciones
Leader1 2011-09-06 19:16:51,760 258 (17,56) recibe-msj 'genera'
Leader1 2011-09-06 19:16:51,762 259 -1 envia-msj 17,56-camino
Mundo1 2011-09-06 19:16:51,763 260 STATIC recibe-msj 17,56-camino
Mundo1 2011-09-06 19:16:51,770 261 STATIC envia-msj id:camino
Leader5 2011-09-06 19:16:51,782 262 (67,52) recibe-msj 'genera'
Leader5 2011-09-06 19:16:51,782 263 -1 envia-msj 67,52-camino
Mundo1 2011-09-06 19:16:51,782 264 STATIC recibe-msj 67,52-camino
Leader3 2011-09-06 19:16:51,784 265 (42,53) recibe-msj 'genera'
Mundo1 2011-09-06 19:16:51,784 266 STATIC envia-msj id:camino
Leader3 2011-09-06 19:16:51,784 267 -1 envia-msj 42,53-camino
Mundo1 2011-09-06 19:16:51,789 268 STATIC recibe-msj 42,53-camino
Leader2 2011-09-06 19:16:51,793 269 (35,3) recibe-msj 'genera'
Mundo1 2011-09-06 19:16:51,794 270 STATIC envia-msj id:camino
Leader2 2011-09-06 19:16:51,794 271 -1 envia-msj 35,3-camino
Mundo1 2011-09-06 19:16:51,798 272 STATIC recibe-msj 35,3-camino
Mundo1 2011-09-06 19:16:51,799 273 STATIC envia-msj id:camino
Leader6 2011-09-06 19:16:51,801 274 (69,10) recibe-msj 'genera'
Leader6 2011-09-06 19:16:51,802 275 -1 envia-msj 69,10-camino
Mundo1 2011-09-06 19:16:51,802 277 STATIC recibe-msj 69,10-camino
Leader4 2011-09-06 19:16:51,804 278 (54,52) recibe-msj 'genera'
Mundo1 2011-09-06 19:16:51,804 279 STATIC envia-msj id:camino
Leader4 2011-09-06 19:16:51,804 280 -1 envia-msj 54,52-camino
Mundo1 2011-09-06 19:16:51,813 281 STATIC recibe-msj 54,52-camino
Mundo1 2011-09-06 19:16:51,814 282 STATIC envia-msj id:camino
Leader1 2011-09-06 19:16:51,970 283 (17,56) recibe-msj 'camino'
Leader1 2011-09-06 19:16:51,971 284 14 buscar-dependientes
Leader1 2011-09-06 19:16:51,971 285 14 empezar-en
Leader1 2011-09-06 19:16:51,971 286 14 andar-hacia 0
Leader1 2011-09-06 19:16:51,973 287 14 envia-msj encontrado
Leader1 2011-09-06 19:16:51,977 288 14 envia-msj ven-conmigo-14-1
Leader5 2011-09-06 19:16:51,990 289 (67,52) recibe-msj 'camino'
Leader5 2011-09-06 19:16:51,991 290 43 buscar-dependientes
Leader5 2011-09-06 19:16:51,991 291 43 empezar-en
Leader5 2011-09-06 19:16:51,991 292 43 andar-hacia 0
Leader5 2011-09-06 19:16:51,991 293 43 envia-msj encontrado
Leader5 2011-09-06 19:16:51,996 294 43 envia-msj ven-conmigo-43-5
Leader3 2011-09-06 19:16:52,001 295 (42,53) recibe-msj 'camino'
Leader3 2011-09-06 19:16:52,002 296 28 buscar-dependientes
Leader3 2011-09-06 19:16:52,002 297 28 empezar-en
Leader3 2011-09-06 19:16:52,002 298 28 andar-hacia 0
Leader3 2011-09-06 19:16:52,004 299 28 envia-msj encontrado
```

Leader2	2011-09-06	19:16:52,004	300	(35,3)	recibe-msj	'camino'
Leader2	2011-09-06	19:16:52,005	301	19	buscar-dependientes	
Leader2	2011-09-06	19:16:52,005	302	19	empezar-en	
Leader2	2011-09-06	19:16:52,005	303	19	andar-hacia	0
Leader2	2011-09-06	19:16:52,006	304	19	envia-msj	encontrado
Leader3	2011-09-06	19:16:52,014	306	28	envia-msj	ven-conmigo-28-3
Leader2	2011-09-06	19:16:52,017	307	19	envia-msj	ven-conmigo-19-2
Leader6	2011-09-06	19:16:52,022	308	(69,10)	recibe-msj	'camino'
Leader6	2011-09-06	19:16:52,023	309	39	buscar-dependientes	
Leader6	2011-09-06	19:16:52,023	310	39	empezar-en	
Leader6	2011-09-06	19:16:52,023	311	39	andar-hacia	0
Leader6	2011-09-06	19:16:52,024	312	39	envia-msj	encontrado
Leader4	2011-09-06	19:16:52,024	313	(54,52)	recibe-msj	'camino'
Leader4	2011-09-06	19:16:52,025	314	36	buscar-dependientes	
Leader4	2011-09-06	19:16:52,025	315	36	empezar-en	
Leader4	2011-09-06	19:16:52,025	316	36	andar-hacia	0
Leader6	2011-09-06	19:16:52,026	317	39	envia-msj	ven-conmigo-39-6
Leader4	2011-09-06	19:16:52,027	318	36	envia-msj	encontrado
Leader4	2011-09-06	19:16:52,030	319	36	envia-msj	ven-conmigo-36-4
Leader1	2011-09-06	19:16:52,181	320	(17,56)	recibe-msj	'camino'
Leader1	2011-09-06	19:16:52,182	321	14	buscar-dependientes	
Leader1	2011-09-06	19:16:52,182	322	14	empezar-en	
Leader1	2011-09-06	19:16:52,182	323	14	andar-hacia	0
Leader1	2011-09-06	19:16:52,182	324	14	envia-msj	encontrado
Leader1	2011-09-06	19:16:52,185	325	14	envia-msj	ven-conmigo-14-1
Leader5	2011-09-06	19:16:52,200	326	(67,52)	recibe-msj	'camino'
Leader5	2011-09-06	19:16:52,202	327	43	buscar-dependientes	
Leader5	2011-09-06	19:16:52,202	328	43	empezar-en	
Leader5	2011-09-06	19:16:52,202	329	43	andar-hacia	0
Leader5	2011-09-06	19:16:52,203	330	43	envia-msj	encontrado
Leader5	2011-09-06	19:16:52,205	331	43	envia-msj	ven-conmigo-43-5
Leader3	2011-09-06	19:16:52,213	333	(42,53)	recibe-msj	'camino'
Leader3	2011-09-06	19:16:52,214	334	28	buscar-dependientes	
Leader2	2011-09-06	19:16:52,215	335	(35,3)	recibe-msj	'camino'
Leader3	2011-09-06	19:16:52,215	336	28	empezar-en	
Leader3	2011-09-06	19:16:52,215	337	28	andar-hacia	0
Leader2	2011-09-06	19:16:52,215	338	19	buscar-dependientes	
Leader2	2011-09-06	19:16:52,216	339	19	empezar-en	
Leader2	2011-09-06	19:16:52,216	340	19	andar-hacia	0
Leader3	2011-09-06	19:16:52,216	341	28	envia-msj	encontrado
Leader2	2011-09-06	19:16:52,216	342	19	envia-msj	encontrado
Leader3	2011-09-06	19:16:52,221	343	28	envia-msj	ven-conmigo-28-3
Leader2	2011-09-06	19:16:52,227	344	19	envia-msj	ven-conmigo-19-2
Leader6	2011-09-06	19:16:52,235	345	(69,10)	recibe-msj	'camino'
Leader6	2011-09-06	19:16:52,236	346	39	buscar-dependientes	
Leader6	2011-09-06	19:16:52,236	347	39	empezar-en	
Leader6	2011-09-06	19:16:52,236	348	39	andar-hacia	0
Leader6	2011-09-06	19:16:52,237	350	39	envia-msj	encontrado
Leader4	2011-09-06	19:16:52,237	349	(54,52)	recibe-msj	'camino'
Leader4	2011-09-06	19:16:52,237	351	36	buscar-dependientes	
Leader4	2011-09-06	19:16:52,238	352	36	empezar-en	
Leader4	2011-09-06	19:16:52,238	353	36	andar-hacia	0
Leader4	2011-09-06	19:16:52,239	354	36	envia-msj	encontrado
Leader6	2011-09-06	19:16:52,239	355	39	envia-msj	ven-conmigo-39-6
Leader4	2011-09-06	19:16:52,243	356	36	envia-msj	ven-conmigo-36-4
Leader5	2011-09-06	19:16:53,038	363	(67,52)	recibe-msj	'genera'
Leader5	2011-09-06	19:16:53,039	364	43	envia-msj	67,52-coordenadas
Leader3	2011-09-06	19:16:53,057	366	(42,53)	recibe-msj	'genera'
Leader3	2011-09-06	19:16:53,059	367	28	envia-msj	42,52-coordenadas
Leader2	2011-09-06	19:16:53,060	368	(35,3)	recibe-msj	'genera'
Leader2	2011-09-06	19:16:53,061	369	19	envia-msj	35,4-coordenadas
Leader6	2011-09-06	19:16:53,084	371	(69,10)	recibe-msj	'genera'
Leader6	2011-09-06	19:16:53,094	372	39	envia-msj	69,11-coordenadas
Leader4	2011-09-06	19:16:53,112	374	(54,52)	recibe-msj	'genera'
Leader4	2011-09-06	19:16:53,115	375	36	envia-msj	54,52-coordenadas
Leader1	2011-09-06	19:16:53,235	385	(17,56)	recibe-msj	'genera'
Leader1	2011-09-06	19:16:53,238	386	14	envia-msj	17,55-coordenadas
Leader5	2011-09-06	19:16:53,264	390	(67,52)	recibe-msj	'genera'
Leader5	2011-09-06	19:16:53,267	391	43	envia-msj	67,51-coordenadas
Leader5	2011-09-06	19:16:53,267	392	43	salir-de	
Leader5	2011-09-06	19:16:53,268	393	42	entrar-en	
Leader5	2011-09-06	19:16:53,268	394	42	entrar-en	
Leader5	2011-09-06	19:16:53,268	395	42	buscar-dependientes	
Leader3	2011-09-06	19:16:53,284	396	(42,52)	recibe-msj	'genera'
Leader3	2011-09-06	19:16:53,290	397	28	colision	
Leader3	2011-09-06	19:16:53,298	398	28	envia-msj	42,52-coordenadas
Leader5	2011-09-06	19:16:53,300	399	43	envia-msj	ven-conmigo-42-5
Leader2	2011-09-06	19:16:53,307	400	(35,4)	recibe-msj	'genera'
Leader2	2011-09-06	19:16:53,309	401	19	envia-msj	35,5-coordenadas
Leader6	2011-09-06	19:16:53,327	405	(69,11)	recibe-msj	'genera'
Leader6	2011-09-06	19:16:53,329	406	39	envia-msj	69,12-coordenadas
Leader4	2011-09-06	19:16:53,351	408	(54,52)	recibe-msj	'genera'
Leader4	2011-09-06	19:16:53,360	411	36	envia-msj	54,51-coordenadas
Leader4	2011-09-06	19:16:53,360	412	36	salir-de	

```

Leader4 2011-09-06 19:16:53,360 413 35 entrar-en
Leader4 2011-09-06 19:16:53,360 414 35 entrar-en
Leader4 2011-09-06 19:16:53,360 415 35 buscar-dependientes

```

E.2. Muerte de un agente

```

Independent1 2011-09-07 17:22:40,487 126 26 nacimiento \\
Fire1 2011-09-07 17:22:40,575 127 23 inicio-foco (36,44)\\
Fire1 2011-09-07 17:22:40,578 128 many envia-msj posiciones \\ \\ \\ \\
Independent1 2011-09-07 17:22:40,752 130 -1 envia-msj 33,64-camino \\
Mundo1 2011-09-07 17:22:40,754 131 STATIC recibe-msj 33,64-camino \\
Mundo1 2011-09-07 17:22:40,757 132 STATIC envia-msj id:camino \\
Independent1 2011-09-07 17:22:40,960 134 (33,64) recibe-msj 'camino' \\
Independent1 2011-09-07 17:22:40,961 135 26 empezar-en \\
Independent1 2011-09-07 17:22:40,961 136 26 andar-hacia 41 \\ \\ \\ \\
Independent1 2011-09-07 17:22:40,965 137 26 envia-msj encontrado \\
Independent1 2011-09-07 17:22:41,174 141 (33,64) recibe-msj 'camino' \\
Independent1 2011-09-07 17:22:41,385 146 26 envia-msj 33,63-coordenadas \\
Independent1 2011-09-07 17:22:41,594 150 26 envia-msj 33,62-coordenadas \\
Independent1 2011-09-07 17:22:41,801 154 26 envia-msj 33,61-coordenadas \\
Independent1 2011-09-07 17:22:42,009 158 26 envia-msj 33,60-coordenadas \\
Independent1 2011-09-07 17:22:42,217 162 26 colision \\
Independent1 2011-09-07 17:22:42,217 163 26 envia-msj 33,60-coordenadas \\
Independent1 2011-09-07 17:22:42,426 167 26 envia-msj 33,59-coordenadas \\
Independent1 2011-09-07 17:22:42,426 168 26 desobedecer-salir-de \\
Independent1 2011-09-07 17:22:42,426 169 25 desobedecer-entrar-en \\
Independent1 2011-09-07 17:22:42,426 170 25 entrar-en \\
Fire1 2011-09-07 17:22:42,578 174 many envia-msj posiciones \\
Independent1 2011-09-07 17:22:42,633 176 25 envia-msj 33,58-coordenadas \\
Independent1 2011-09-07 17:22:42,841 180 25 envia-msj 33,57-coordenadas \\
Independent1 2011-09-07 17:22:43,050 184 25 envia-msj 33,56-coordenadas \\
Independent1 2011-09-07 17:22:43,257 188 25 envia-msj 33,55-coordenadas \\
Independent1 2011-09-07 17:22:43,466 192 25 envia-msj 33,54-coordenadas \\
Independent1 2011-09-07 17:22:43,672 196 25 envia-msj 33,53-coordenadas \\
Independent1 2011-09-07 17:22:43,880 200 25 envia-msj 33,52-coordenadas \\
Independent1 2011-09-07 17:22:44,086 204 25 colision \\
Independent1 2011-09-07 17:22:44,087 205 25 envia-msj 33,52-coordenadas \\
Independent1 2011-09-07 17:22:44,294 209 25 envia-msj 33,51-coordenadas \\
Independent1 2011-09-07 17:22:44,294 210 25 desobedecer-salir-de \\
Independent1 2011-09-07 17:22:44,294 211 24 desobedecer-entrar-en \\
Independent1 2011-09-07 17:22:44,294 212 24 entrar-en \\
Independent1 2011-09-07 17:22:44,499 216 24 envia-msj 33,50-coordenadas \\
Fire1 2011-09-07 17:22:44,584 220 27 acceso 27 \\
Fire1 2011-09-07 17:22:44,585 221 many envia-msj posiciones \\
Independent1 2011-09-07 17:22:44,708 223 24 envia-msj 33,49-coordenadas \\
Independent1 2011-09-07 17:22:44,917 227 24 envia-msj 33,48-coordenadas \\
Independent1 2011-09-07 17:22:45,125 231 24 envia-msj 33,47-coordenadas \\
Independent1 2011-09-07 17:22:45,331 235 24 envia-msj 33,46-coordenadas \\
Independent1 2011-09-07 17:22:45,536 239 24 colision \\
Independent1 2011-09-07 17:22:45,537 240 24 envia-msj 33,46-coordenadas \\
Independent1 2011-09-07 17:22:45,744 244 24 visto-fuego \\
Independent1 2011-09-07 17:22:45,744 245 24 andar-hacia 12 \\ \\
Independent1 2011-09-07 17:22:45,949 246 24 envia-msj 32,46-coordenadas \\
Independent1 2011-09-07 17:22:46,158 250 24 colision \\
Independent1 2011-09-07 17:22:46,158 251 24 envia-msj 32,46-coordenadas \\
Independent1 2011-09-07 17:22:46,363 255 24 visto-fuego \\
Independent1 2011-09-07 17:22:46,364 256 24 andar-hacia 41 \\ \\
Independent1 2011-09-07 17:22:46,569 257 24 colision \\
Independent1 2011-09-07 17:22:46,570 258 24 envia-msj 32,46-coordenadas \\
Fire1 2011-09-07 17:22:46,589 259 24 acceso 24 \\ \\
Fire1 2011-09-07 17:22:46,590 260 many envia-msj posiciones \\
Independent1 2011-09-07 17:22:46,775 265 24 visto-fuego \\
Independent1 2011-09-07 17:22:46,776 266 24 andar-hacia 12 \\ \\
Independent1 2011-09-07 17:22:46,983 267 24 envia-msj 31,46-coordenadas \\
Independent1 2011-09-07 17:22:47,188 271 24 colision \\
Independent1 2011-09-07 17:22:47,189 272 24 envia-msj 31,46-coordenadas \\
Independent1 2011-09-07 17:22:47,394 276 24 visto-fuego \\
Independent1 2011-09-07 17:22:47,394 277 24 andar-hacia 41 \\ \\
Independent1 2011-09-07 17:22:47,600 278 24 colision \\
Independent1 2011-09-07 17:22:47,601 279 24 envia-msj 31,46-coordenadas \\
Independent1 2011-09-07 17:22:47,806 283 24 visto-fuego \\
Independent1 2011-09-07 17:22:47,807 284 24 andar-hacia 12 \\ \\
Independent1 2011-09-07 17:22:48,013 285 24 colision \\
Independent1 2011-09-07 17:22:48,014 286 24 envia-msj 31,46-coordenadas \\
Independent1 2011-09-07 17:22:48,220 290 24 visto-fuego \\
Independent1 2011-09-07 17:22:48,220 291 24 andar-hacia 41 \\ \\
Independent1 2011-09-07 17:22:48,427 292 24 colision \\
Independent1 2011-09-07 17:22:48,427 293 24 envia-msj 31,46-coordenadas \\
Fire1 2011-09-07 17:22:48,593 297 many envia-msj posiciones \\
Independent1 2011-09-07 17:22:48,634 299 24 visto-fuego \\
Independent1 2011-09-07 17:22:48,635 300 24 andar-hacia 12 \\ \\

```

Independent1 2011-09-07 17:22:48,841 301 24 envia-msj 30,46-coordenadas\\
 Independent1 2011-09-07 17:22:49,047 305 24 colision \\
 Independent1 2011-09-07 17:22:49,048 306 24 envia-msj 30,46-coordenadas\\
 Independent1 2011-09-07 17:22:49,255 310 24 visto-fuego\\
 Independent1 2011-09-07 17:22:49,256 311 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:49,463 312 24 colision \\
 Independent1 2011-09-07 17:22:49,463 313 24 envia-msj 30,46-coordenadas\\
 Independent1 2011-09-07 17:22:49,668 317 24 visto-fuego\\
 Independent1 2011-09-07 17:22:49,669 318 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:49,877 319 24 colision \\
 Independent1 2011-09-07 17:22:49,878 320 24 envia-msj 30,46-coordenadas\\
 Independent1 2011-09-07 17:22:50,084 324 24 visto-fuego\\
 Independent1 2011-09-07 17:22:50,084 325 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:50,292 326 24 colision \\
 Independent1 2011-09-07 17:22:50,292 327 24 envia-msj 30,46-coordenadas\\
 Independent1 2011-09-07 17:22:50,498 331 24 visto-fuego\\
 Independent1 2011-09-07 17:22:50,498 332 24 andar-hacia 12\\
 Fire1 2011-09-07 17:22:50,602 333 many envia-msj posiciones\\
 Independent1 2011-09-07 17:22:50,704 335 24 colision \\
 Independent1 2011-09-07 17:22:50,705 336 24 envia-msj 30,46-coordenadas\\
 Independent1 2011-09-07 17:22:50,911 340 24 visto-fuego\\
 Independent1 2011-09-07 17:22:50,911 341 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:51,116 342 24 envia-msj 29,46-coordenadas\\
 Independent1 2011-09-07 17:22:51,321 346 24 colision \\
 Independent1 2011-09-07 17:22:51,322 347 24 envia-msj 29,46-coordenadas\\
 Independent1 2011-09-07 17:22:51,527 351 24 visto-fuego\\
 Independent1 2011-09-07 17:22:51,528 352 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:51,733 353 24 colision \\
 Independent1 2011-09-07 17:22:51,733 354 24 envia-msj 29,46-coordenadas\\
 Independent1 2011-09-07 17:22:51,938 358 24 visto-fuego\\
 Independent1 2011-09-07 17:22:51,938 359 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:52,144 360 24 colision \\
 Independent1 2011-09-07 17:22:52,144 361 24 envia-msj 29,46-coordenadas\\
 Independent1 2011-09-07 17:22:52,351 365 24 visto-fuego\\
 Independent1 2011-09-07 17:22:52,351 366 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:52,560 367 24 colision \\
 Independent1 2011-09-07 17:22:52,560 368 24 envia-msj 29,46-coordenadas\\
 Fire1 2011-09-07 17:22:52,610 372 many envia-msj posiciones\\
 Independent1 2011-09-07 17:22:52,766 374 24 visto-fuego\\
 Independent1 2011-09-07 17:22:52,766 375 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:52,972 376 24 envia-msj 28,46-coordenadas\\
 Independent1 2011-09-07 17:22:53,180 380 24 colision \\
 Independent1 2011-09-07 17:22:53,180 381 24 envia-msj 28,46-coordenadas\\
 Independent1 2011-09-07 17:22:53,385 385 24 visto-fuego\\
 Independent1 2011-09-07 17:22:53,386 386 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:53,593 387 24 colision \\
 Independent1 2011-09-07 17:22:53,593 388 24 envia-msj 28,46-coordenadas\\
 Independent1 2011-09-07 17:22:53,798 392 24 visto-fuego\\
 Independent1 2011-09-07 17:22:53,798 393 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:54,004 394 24 colision \\
 Independent1 2011-09-07 17:22:54,005 395 24 envia-msj 28,46-coordenadas\\
 Independent1 2011-09-07 17:22:54,210 399 24 visto-fuego\\
 Independent1 2011-09-07 17:22:54,211 400 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:54,416 401 24 colision \\
 Independent1 2011-09-07 17:22:54,417 402 24 envia-msj 28,46-coordenadas\\
 Fire1 2011-09-07 17:22:54,618 406 many envia-msj posiciones\\
 Independent1 2011-09-07 17:22:54,622 407 24 visto-fuego\\
 Independent1 2011-09-07 17:22:54,623 408 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:54,828 410 24 envia-msj 27,46-coordenadas\\
 Independent1 2011-09-07 17:22:55,035 414 24 colision \\
 Independent1 2011-09-07 17:22:55,036 415 24 envia-msj 27,46-coordenadas\\
 Independent1 2011-09-07 17:22:55,241 419 24 visto-fuego\\
 Independent1 2011-09-07 17:22:55,241 420 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:55,447 421 24 colision \\
 Independent1 2011-09-07 17:22:55,447 422 24 envia-msj 27,46-coordenadas\\
 Independent1 2011-09-07 17:22:55,655 426 24 visto-fuego\\
 Independent1 2011-09-07 17:22:55,655 427 24 andar-hacia 41\\
 Independent1 2011-09-07 17:22:55,860 428 24 colision \\
 Independent1 2011-09-07 17:22:55,861 429 24 envia-msj 27,46-coordenadas\\
 Independent1 2011-09-07 17:22:56,067 433 24 visto-fuego\\
 Independent1 2011-09-07 17:22:56,068 434 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:56,273 435 24 colision \\
 Independent1 2011-09-07 17:22:56,274 436 24 envia-msj 27,46-coordenadas\\
 Independent1 2011-09-07 17:22:56,479 440 24 visto-fuego\\
 Independent1 2011-09-07 17:22:56,479 441 24 andar-hacia 41\\
 Fire1 2011-09-07 17:22:56,622 442 28 acceso 28\\
 Fire1 2011-09-07 17:22:56,625 443 many envia-msj posiciones\\
 Independent1 2011-09-07 17:22:56,685 445 24 colision \\
 Independent1 2011-09-07 17:22:56,685 446 24 envia-msj 27,46-coordenadas\\
 Independent1 2011-09-07 17:22:56,892 450 24 visto-fuego\\
 Independent1 2011-09-07 17:22:56,892 451 24 andar-hacia 12\\
 Independent1 2011-09-07 17:22:57,098 452 24 envia-msj 26,46-coordenadas\\
 Independent1 2011-09-07 17:22:57,307 456 24 colision \\
 Independent1 2011-09-07 17:22:57,307 457 24 envia-msj 26,46-coordenadas\\

```

Independent1 2011-09-07 17:22:57,514 461 24 visto-fuego\\
Independent1 2011-09-07 17:22:57,514 462 24 andar-hacia 41\\
Independent1 2011-09-07 17:22:57,721 463 24 colision \\
Independent1 2011-09-07 17:22:57,721 464 24 envia-msj 26,46-coordenadas\\
Independent1 2011-09-07 17:22:57,927 468 24 visto-fuego\\
Independent1 2011-09-07 17:22:57,928 469 24 andar-hacia 12\\
Independent1 2011-09-07 17:22:58,133 470 24 colision \\
Independent1 2011-09-07 17:22:58,134 471 24 envia-msj 26,46-coordenadas\\
Independent1 2011-09-07 17:22:58,339 475 24 visto-fuego\\
Independent1 2011-09-07 17:22:58,339 476 24 andar-hacia 41\\
Independent1 2011-09-07 17:22:58,546 477 24 colision \\
Independent1 2011-09-07 17:22:58,546 478 24 envia-msj 26,46-coordenadas\\
Fire1 2011-09-07 17:22:58,629 482 25 acceso 25\\
Fire1 2011-09-07 17:22:58,632 483 many envia-msj posiciones\\
Independent1 2011-09-07 17:22:58,751 485 24 visto-fuego\\
Independent1 2011-09-07 17:22:58,752 486 24 andar-hacia 12\\
Independent1 2011-09-07 17:22:58,958 487 24 envia-msj 25,46-coordenadas\\
Independent1 2011-09-07 17:22:59,165 491 24 colision \\
Independent1 2011-09-07 17:22:59,165 492 24 envia-msj 25,46-coordenadas\\
Independent1 2011-09-07 17:22:59,372 496 24 visto-fuego\\
Independent1 2011-09-07 17:22:59,374 497 24 andar-hacia 41\\
Independent1 2011-09-07 17:22:59,580 498 24 colision \\
Independent1 2011-09-07 17:22:59,581 499 24 envia-msj 25,46-coordenadas\\
Independent1 2011-09-07 17:22:59,787 503 24 visto-fuego\\
Independent1 2011-09-07 17:22:59,788 504 24 andar-hacia 12\\
Independent1 2011-09-07 17:22:59,994 505 24 colision \\
Independent1 2011-09-07 17:22:59,994 506 24 envia-msj 25,46-coordenadas\\
Independent1 2011-09-07 17:23:00,199 510 24 visto-fuego\\
Independent1 2011-09-07 17:23:00,199 511 24 andar-hacia 41\\
Independent1 2011-09-07 17:23:00,404 512 24 colision \\
Independent1 2011-09-07 17:23:00,405 513 24 envia-msj 25,46-coordenadas\\
Independent1 2011-09-07 17:23:00,614 517 24 visto-fuego\\
Independent1 2011-09-07 17:23:00,614 518 24 andar-hacia 12\\
Fire1 2011-09-07 17:23:00,641 519 many envia-msj posiciones\\
Independent1 2011-09-07 17:23:00,819 521 24 colision \\
Independent1 2011-09-07 17:23:00,820 522 24 envia-msj 25,46-coordenadas\\
Independent1 2011-09-07 17:23:01,027 526 24 visto-fuego\\
Independent1 2011-09-07 17:23:01,027 527 24 andar-hacia 41\\
Fire1 2011-09-07 17:23:02,647 528 34 acceso 34\\
Fire1 2011-09-07 17:23:02,650 529 many envia-msj posiciones\\
Fire1 2011-09-07 17:23:04,655 531 many envia-msj posiciones\\

```


Bibliografía

- [Agre, 2003] Agre, P. E. (2003). Architecture of complexity. *Journal of the Learning Sciences*, 12(3):413–426.
- [Barros et al., 2004] Barros, L. M., da Silva, A. T., and Musse, S. R. (2004). Petrosim: An architecture to manage virtual crowds in in panic situations. In Press, A., editor, *Proceedings of Computer Animation and Social Agents*, pages 111–120.
- [Davey, 1978] Davey, A. (1978). *Discourse production: A computer model of some aspects of a speaker*. Edinburgh University Press, Edinburgh, 1 edition.
- [Dennett, 1987] Dennett, D. C. (1987). *The Intentional Stance*. The MIT Press. Cambridge, MA.
- [Elliott and Smith, 1993] Elliott, D. and Smith, D. (1993). Football stadia disasters in the united kingdom: learning from tragedy? *Industrial and Environmental Crisis Quaterly*, 7(3):205–229.
- [Fahy and Prouxl, 1997] Fahy, R. F. and Prouxl, G. (1997). Human behavior in the wold trade center evacuation. In *International Association for Fire Safety Science, Fifth International Symposium*, pages 713–724.
- [Foudil et al., 2009] Foudil, C., Nouredine, D., Sanza, C., and Duthen, Y. (2009). Path finding and collision avoidance in crowd simulation. *Journal of Computing and Information Technology - CIT*, 17(3):217–228.
- [Gervás, 2007] Gervás, P. (2007). Tap: a text arranging pipeline. Technical report, Universidad Complutense de Madrid.
- [Gilbert, 2008] Gilbert, N. (2008). *Agent-Based Models*. SAGE Publications.
- [Gilbert and Troitzsch, 2006] Gilbert, N. and Troitzsch, K. G. (2006). *Simulación $\frac{1}{2}n$ para las Ciencias Sociales*. Mc. Graw Hill, second edition.

- [Goldman, 1975] Goldman, N. (1975). Conceptual generation. In Schank, R. C., editor, *Conceptual Information Processing*, pages 289–371. North-Holland and Elsevier, Amsterdam and New York.
- [Hassan et al., 2007] Hassan, S., León, C., Gervás, P., and Hervás, R. (2007). A computer model that generates biography-like narratives. In *4th International Joint Workshop on Computational Creativity*, Goldsmiths, University of London.
- [Helbing et al., 2000] Helbing, D., Farkas, I., and Viesek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407:487–490.
- [Helbing et al., 2001] Helbing, D., Molnár, P., Farkas, I. J., and Bolay, K. (2001). Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design*, 28(3):361–383.
- [Huhns and Singh, 1998] Huhns, M. and Singh, M. (1998). *Reading in Agents*. Morgan Kaufmann Publishers, Inc.
- [Johnson, 1987] Johnson, N. R. (1987). Panic at "the who concert stampede": An empirical assessment. *Social Problems*, 34(4):362–373.
- [Kamphuis and Overmars, 2004] Kamphuis, A. and Overmars, M. H. (2004). Motion planning for coherent groups of entities. In *ICRA*, pages 3815–3822.
- [Keating, 1982] Keating, J. P. (1982). The myth of panic. *Fire Journal*.
- [Kelley et al., 1965] Kelley, H. H., Condry, J. C. J., Dahlke, A. E., and Hill, A. H. (1965). Collective behaviour in a simulated panic situation. *Journal of Experimental Social Psychology*, 1:20–54.
- [León et al., 2007] León, C., Hassan, S., and Gervás, P. (2007). From the event log of a social simulation to narrative discourse: Content planning in story generation. In Olivier, P. and Kray, C., editors, *AISB'07 Symposium: AI and Narrative Games for Education*, pages 402–409, Culture Lab, Newcastle University, Newcastle upon Tyne, UK.
- [Li et al., 2004] Li, H., Tang, W., and Simpson, D. (2004). Behaviour based motion simulation for fire evacuation procedures. In *Theory and Practice of Computer Graphics (TPCG)*, pages 112–118.
- [Mas, 2005] Mas, A. (2005). *Agentes Software y Sistemas Multi-Agente. Conceptos, arquitecturas y aplicaciones*. Prentice Hall.
- [McCarthy, 1978] McCarthy, J. (1978). Ascribing mental qualities to machines. Technical Report, Stanford University AI Lab.

- [McKeown, 1985] McKeown, K. R. (1985). Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41.
- [Moretti, 2002] Moretti, S. (2002). Computer simulation in sociology: What contribution? *Social Science Computer Review*, 20(1):143–166.
- [Murakami et al., 2002] Murakami, Y., Minami, K., Kawasoe, T., and Ishida, T. (2002). Multi-agent simulation for crisis management. *Knowledge Media Networking, IEEE Workshop on*, 0:135.
- [Musse and Thalmann, 2000] Musse, S. R. and Thalmann, D. (2000). From one virtual actor to virtual crowds: requirements and constraints. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, pages 52–53, New York, NY, USA. ACM.
- [Pan et al., 2006] Pan, X., Han, C. S., and Law, K. H. (2006). A multi-agent based simulation framework for the study of human and social behavior in egress analysis. *Proceedings of International Conference in Civil Engineering*.
- [Parberry, 1995] Parberry, I. (1995). A real-time algorithm for the $(n^2 - 1)$ -puzzle. *Information Processing Letters*, 56:23–28.
- [Pavón et al., 2006] Pavón, J., Hassan, S. M., and Sansores-Pérez, C. (2006). Simulación de sistemas sociales con agentes software. *Campus Multidisciplinar en Percepción e Inteligencia*, 1:389–400.
- [Pelechano and Badler, 2006] Pelechano, N. and Badler, N. I. (2006). Modeling crowd and trained leader behavior during building evacuation. *IEEE Computer Graphics and Applications*, 26(6):80–86.
- [Ramsay, 1988] Ramsay, A. (1988). Appelt, d.e. planning english sentences. *Cognitive Science*, 12(3):467–477.
- [Reif and Wang, 1995] Reif, J. and Wang, H. (1995). Social potential fields: A distributed behavioral control for autonomous robots. In Goldberg, K., Halperin, D., Latombe, J. C., and Wilson, R., editors, *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 431–459.
- [Reiter, 1994] Reiter, E. (1994). Has a consensus nl generation architecture appeared, and is it psycholinguistically plausible? In *INLG '94: Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 163–170, Morristown, NJ, USA. Association for Computational Linguistics.
- [Reynolds, 1987] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th*

annual conference on Computer graphics and interactive techniques, pages 25–34, New York, NY, USA. ACM.

- [Reynolds, 1999] Reynolds, C. W. (1999). Steering behaviors for autonomous characters. In *Game Developers Conference*.
- [Russell and Norvig, 2004] Russell, S. and Norvig, P. (2004). *Inteligencia Artificial. Un enfoque moderno*. Prentice Hall, second edition.
- [Sansores-Pérez, 2007] Sansores-Pérez, C. (2007). *Metodología para el estudio de sociedades artificiales*. PhD thesis, Universidad Complutense de Madrid.
- [Serrano et al., 2009] Serrano, E., Botia, J. A., and Cadenas, J. M. (2009). Ubik: a multi-agent based simulator for ubiquitous computing applications. *Physical Agents*.
- [Shannon, 1998] Shannon, R. E. (1998). Introduction to the art and science of simulation. In *Winter Simulation Conference*, pages 7–14.
- [Shendarkar et al., 2006] Shendarkar, Ameya, Vasudevan, Karthik, Lee, Seunggho, and Son, Y.-J. (2006). Crowd simulation for emergency response using bdi agent based on virtual reality. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 545–553. Winter Simulation Conference.
- [Simon, 1996] Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press, 3 edition.
- [Sánchez-Torrubia and Lozano-Terrazas, 2001] Sánchez-Torrubia, M. G. and Lozano-Terrazas, V. (2001). Algoritmo de dijkstra: Un tutorial interactivo. In Miró, J., editor, *Proceedings of the VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001)*, pages 254–25, Palma de Mallorca.
- [Steels and Brooks, 1995] Steels, L. and Brooks, R. (1995). Building agents out of autonomous behavior systems. *The Artificial Life Route to Artificial Intelligence*.
- [Sugiman and Misumi, 1988] Sugiman, T. and Misumi, J. (1988). Development of a new evacuation method for emergencies: Control of collective behavior by emergent small groups. *Journal of Applied Psychology*, 73(1):3–10.
- [Thalmann and Musse, 2007] Thalmann, D. and Musse, S. R. (2007). *Crowd Simulation*. Springer, 1 edition.

- [Tsai et al., 2011] Tsai, J., Fridman, N., Bowring³, E., Brown, M., Epstein, S., Kaminka, G., Marsella, S., Ogden, A., Rika, I., Sheel, A., Taylor, M., Wang, X., Zilka, A., and Tambe¹, M. (2011). Escapes - evacuation simulation with children, authorities, parents, emotions, and social comparison. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Taipei, Taiwan.
- [Wooldridge, 1997] Wooldridge, M. (1997). Agent-based software engineering. *IEEE Proc. on Software Engineering*.
- [Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *Knowledge Engineering Review*, pages 115–152.