Simulación basada en agentes de tráFIco aéreo

ORIÓN GARCÍA GALLARDO

Máster en Investigación en Informática, Facultad de Informática,

Universidad Complutense de Madrid



Trabajo Fin Máster en Sistemas Inteligentes

Fecha

Director: RUBEN FUENTES FERNANDEZ

Autorización de Difusión

Autor

Fecha

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: “Simulación basada en agentes de tráfico aéreo”, realizado durante el curso académico 2010-2011 bajo la dirección del Dr. Rubén Fuentes Fernández 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.

Resumen en castellano

[Enter abstract here, no longer than 350 words. Be sure to retain the Section Break below.]

Palabras clave

Resumen en inglés

[Enter abstract here, no longer than 350 words. Be sure to retain the Section Break below.]

Keywords

Índice de contenidos

[Autorización de Difusión iii](#_Toc296083811)

[Resumen en castellano v](#_Toc296083812)

[Palabras clave v](#_Toc296083813)

[Resumen en inglés vi](#_Toc296083814)

[Keywords vi](#_Toc296083815)

[Índice de contenidos 7](#_Toc296083816)

[Tabla de Figuras 8](#_Toc296083817)

[Agradecimientos 9](#_Toc296083818)

[Introducción 11](#_Toc296083819)

[Conceptos básicos de navegación aérea 13](#_Toc296083820)

[Estado del arte 15](#_Toc296083821)

[Factores humanos en aviación 15](#_Toc296083822)

[Simulación y Gestión de Tráfico Aéreo 17](#_Toc296083823)

[Trabajos de investigación sobre UAVs 20](#_Toc296083824)

[Requisitos del Sistema 21](#_Toc296083825)

[Infraestructura y desarrollo 23](#_Toc296083826)

[Plataforma 24](#_Toc296083827)

[Especificación 24](#_Toc296083828)

[Módulo de lógica 34](#_Toc296083829)

[Visualización 34](#_Toc296083830)

[Controlador 35](#_Toc296083831)

[Casos de estudio 35](#_Toc296083832)

[Discusión 35](#_Toc296083833)

[Conclusión y trabajo futuro 35](#_Toc296083834)

[Bibliografía 36](#_Toc296083835)

[Glosario 39](#_Toc296083836)

Tabla de Figuras

[Figura 1: Casos de Uso](#_Toc296441829)

[Figura 2: Diagrama de Objetivos](#_Toc296441830)

[Figura 3: Estado Mental "FlightPlanner"](#_Toc296441831)

[Figura 4: Tarea de Crear Planes de Vuelo](#_Toc296441832)

[Figura 5: Diagrama de Interacción Para Recibir el Plan de Vuelo Inicial](#_Toc296441833)

[Figura 6: Creación del Estado Mental Inicial del Piloto](#_Toc296441834)

[Figura 7: Diagrama de Actividad General](#_Toc296441835)

[Figura 8: Comprobación de la Hora de Salida del Vuelo](#_Toc296441836)

[Figura 9: Interacción para Arrancar el Avión](#_Toc296441837)

[Figura 10: Descomposición del Plan de Vuelo](#_Toc296441838)

[Figura 11: Tiempo de Creación de las Decisiones](#_Toc296441839)

[Figura 12: Creación de las Decisiones](#_Toc296441840)

[Figura 13: Descomposición de Decisiones](#_Toc296441841)

[Figura 14: Interacción Piloto-Avión: Ejecución Maniobra](#_Toc296441842)

[Figura 15: Aplicación que Comprueba la Finalización del Tramo](#_Toc296441843)

[Figura 16: Diagrama para Ir al Siguiente Tramo](#_Toc296441844)

[Figura 17: Diagrama de Actividad Evitar Conflicto](#_Toc296441845)

[Figura 18: Comprobación de Nuevo Conflicto](#_Toc296441846)

[Figura 19: Creación de las Ordenes](#_Toc296441847)

[Figura 20: Interacción Controlador-Piloto traspaso de Ordenes](#_Toc296441848)

[Figura 21: Piloto Obedeciendo una Orden](#_Toc296441849)

[Figura 22: Conflicto Finalizado por Lejanía entre Aviones](#_Toc296441850)

Agradecimientos

The Acknowledgement page is optional. If you include it, retain the Acknowledgements heading and enter your text here. If you do not include it, delete the entire page. Be sure to retain the Page Break that occurs after the List of Table page above.

# 

Introducción

La navegación aérea es un conjunto de técnicas y procedimientos destinados a conducir eficientemente una aeronave a su destino. Esta conducción eficiente implica aspectos de rutas predefinidas, seguridad y costes que involucran campos muy diversos, desde la física del vuelo, hasta las características del propio avión, pasando por sus tripulantes o las diversas normativas y leyes aplicables. Además, se trata de un campo que se encuentra en continua evolución debido fundamentalmente al continuo incremento en el tráfico aéreo y a los cambios en la tecnología.

Diversas fuentes apuntan el constante incremento del tráfico aéreo durante las últimas décadas. En algunos países como EE.UU., el avión es ya el vehículo más usado para algunos ciudadanos, y Eurocontrol, el organismo a cargo del tráfico aéreo en Europa, estima que éste continuará creciendo aproximadamente al 3,7 por ciento anual en Europa entre 2005 y 2011 (1). En este contexto comienza a surgir la preocupación sobre la posible saturación futura tanto del espacio aéreo como de los aeropuertos (2). Si bien es difícil limitar la demanda de vuelos, existen otros factores que influyen en esta situación y que sí es posible abordar. Entre ellos se cuentan la inadecuación de los métodos depara su control del tráfico aéreo a la situación actual (2). Esta inadecuación se debe principalmente a:

1. Ineficiente utilización del espacio aéreo basada en pasillos predefinidos. El espacio aéreo se encuentra organizado en torno a puntos de paso fijos y vías predefinidas entre ellos, lo que impide aprovechar el resto del espacio aéreo.
2. El uso de tecnologías obsoletas, que en algunos casos tienen más de 30 años. La principal preocupación en este ámbito es la seguridad, por lo que sólo se adoptan tecnologías muy maduras, por ejemplo para gestionar las distancias entre aeronaves o la notificación de información. Ello dificulta la adopción de aproximaciones más recientes que sin embargo pueden mejorar la seguridad y reducir los costes.
3. Saturación de las comunicaciones entre los pilotos y los controladores aéreos. Relacionado con el punto anterior, existe una alta dependencia de las comunicaciones verbales entre los actores del tráfico aéreo. Ello pone fuertes límites al número de eventos y acciones que se pueden tratar por unidad de tiempo. Este es el caso, por ejemplo, de la gestión de los pasillos aéreos por los controladores.

El segundo aspecto que marca la continua evolución del tráfico aéreo es la introducción de cambios tecnológicos. Si bien existen reticencias para cambiar algunos procedimientos clave, como la gestión del tráfico aéreo, se introducen continuamente mejoras de menos calado. Por ejemplo, algunos cambios en los últimos años han sido los sistemas de gestión de vuelo, las aeronaves comerciales de mayor tamaño, o la introducción de aeronaves no tripuladas (UAV, *Unmanned Aerial Vehicle*).

El estudio de un fenómeno de esta complejidad hace uso de multitud de aproximaciones. Los estudios sobre entornos reales son necesarios para recopilar datos e identificar las características de los elementos participantes. Sin embargo, no son adecuados frecuentemente para la evaluación de hipótesis (por ejemplo por motivos de seguridad y tiempo), además de tener unos costes elevados. Para solventar estas limitaciones es frecuente recurrir a herramientas de simulación. Las simulaciones son un proceso relativamente económico para obtener una representación de la realidad sin interferir en el sistema real.

Actualmente hay numerosas aplicaciones que se dedican a la simulación aérea. Una clasificación básica de las mismas es según el foco de su estudio. Por un lado hay una serie de aplicaciones centradas en los aspectos físicos del vuelo. Dentro de estas, uno de los grupos más populares son las aplicaciones basadas en JSBSim. JSBSim (3) es una librería software de código abierto que modela la dinámica de vuelo de aeronaves, el denominado modelo físico de los movimientos de los aviones. Basándose en el entorno JSBSim se encuentran una serie de herramientas que simulan vuelos centrándose en el manejo y control de los aviones, estas son FlightGear (4) y el OpenEaagles (5). Otro trabajo que se centra también en los aspectos físicos pero que está algo más orientado a la investigación, y donde prima un poco más la ligereza y rapidez del programa que la fidelidad a la realidad, es una herramienta desarrollada bajo el entorno Matlab denominada *Aviator Visual Design Simulator* (6). Por otro lado, hay una serie de herramientas que se centran en la navegación aérea. Estas se caracterizan por simular trayectorias de múltiples aeronaves simultáneamente, modelar las distintas regiones espaciales de gestión de vuelos con el objetivo de evitar colisiones y modelar el efecto en el tráfico aéreo de algunos factores externos como puede ser el clima. Entre las más destacadas tenemos que por ejemplo la NASA ha desarrollado una herramienta denominada FACET (Herramienta para evaluar futuros conceptos del tráfico aéreo) (7) (8) mientras que, por otra parte, la FAA (Administración Federal de Aviación de EE.UU.) suele hacer uso de una herramienta denominada IMPACT (Modelo Inteligente basado en agentes para el Análisis de Políticas de Colaboración de la gestión del Tráfico aéreo)(9).

A pesar de su variedad y funcionalidad, las herramientas de simulación también presentan problemas. El principal es que los modelos siempre suponen una simplificación de la realidad, centrada en los aspectos que se consideran más relevantes. En un problema de la complejidad del tráfico aéreo, existen muchos elementos a estudiar, desde la física de los aviones al control del tráfico o las acciones de los actores, que están mutuamente relacionados entre sí. Obviar o simplificar algunos de estos elementos puede llevar a perder información relevante. Además, permiten escasa flexibilidad a la hora de querer introducir nuevos módulos para abordar necesidades diferentes, ya que suelen ser productos cerrados en buena medida) y hacen poca abstracción de la información a tratar. En atención a este último punto, se trata de herramientas donde el conocimiento se encuentra embebido en el código, por lo que resulta difícil de discutir, elaborar y actualizar.. Destacar también que en la mayoría de los casos estos sistemas no modelan ni comportamientos ni conductas de los humanos que intervienen en ellas (ej. pilotos y controladores), ni las interrelaciones entre ellos o con otros elementos externos. Históricamente las tecnologías basadas en simulación de aviación se han enfocado más en los aspectos funcionales o técnicos de la simulación pero fijándonos en que un tercio de los accidentes aéreos se deben a errores de los pilotos (10) parece interesante tener en cuenta el comportamiento, actitud o estado de un piloto en un determinado momento.

Con vistas a abordar los problemas anteriores, este trabajo se plantea la necesidad de desarrollar plataformas de simulación que hagan explícita la información manejada al nivel de abstracción de los expertos en el dominio, y cuya traslación al código sea precisa. Para ello se consideran propuestas de ingeniería dirigida por modelos (MDE, *Model-Driven Engineering*). La MDE (11) es un paradigma de desarrollo en el que los sistemas se abordan en base a una serie de modelos que describen las entidades del sistema desde diferentes perspectivas. Estos modelos son después sometidos a transformaciones semi-automatizadas para obtener el resto de productos del desarrollo, como código, documentación o pruebas.

Con vistas a facilitar el modelado a alto nivel de sistemas relacionados con el tráfico aéreo, y prestando especial atención a los aspectos humanos del mismo, se considera especialmente apropiadas las aproximaciones basadas en agentes. El Modelado Basado en Agentes (ABM, *Agent-Based Modelling*) usa como su principal abstracción los *agentes*, que son abstracciones computacionales intencionales y sociales. Esto quiere decir, que persiguen una serie de objetivos y dependen e interactúan con los otros agentes que hay en el sistema.. Por estos motivos guardan una cierta similitud con las personas que participan en la gestión o toma de decisiones de los vuelos (12). Es por tanto razonable establecer una correspondencia entre los conceptos de agente y persona, que si bien no son equiparables resultan más próximos que en el caso de otras abstracciones computacionales.

Este trabajo constituye un primer paso en la línea apuntada de simulaciones de tráfico aéreo, con especial atención a los elementos humanos, basadas en el ABM y MDE. La memoria presenta un modelo básico donde se consideran las interacciones entre controladores y pilotos, de estos con los aviones y las automatizadas entre aviones (ej. detección de proximidad o de velocidad). Para comprobar la extensibilidad y flexibilidad del modelo propuesto se incorporan UAVs en este espacio aéreo civil. Estos modelos son usados para generar automáticamente simulaciones usando el entorno del IDK (*INGENIAS* Development *Kit*) (13), un editor y generador de código para Sistemas Multi-Agente (SMAs). Los experimentos llevados a cabo con estas simulaciones han permitidos observar fenómenos de ¿?? similares a los existentes en entornos reales, al tiempo que han mostrado la flexibilidad de la propuesta para atender nuevas necesidades de estudio.

El resto de la memoria se organiza en una serie de secciones donde: primeramente se definirán algunos conceptos de la navegación aérea (*Conceptos de la navegación aérea*) para seguidamente entrar en profundidad en el estado del arte de los actuales sistemas de simulación (*Estado del Arte*). Posteriormente se definirán una serie de requisitos que algunas de las entidades del sistema deben de cumplir (*Requisitos del Sistema*). A continuación se presentará el objetivo y diseño que se ha elegido para afrontar este problema, donde se mostrarán y explicarán los distintos diagramas desarrollados con el IDK que modelan el problema a tratar (*Plataforma*). Seguidamente se presentarán una serie de casos de estudio y pruebas realizadas sobre este entorno de simulación desarrollado (*Casos de Estudio*) para, posteriormente, en el siguiente capítulo, discutir y analizar estos datos obtenidos (*Discusión*). Y por último se aportarán unas conclusiones ante las simulaciones realizadas, y se presentarán las áreas de trabajo futuro que se pueden realizar sobre el trabajo actual (*Conclusiones y Trabajos Futuros*).

Conceptos básicos de navegación aérea

La navegación aérea engloba el conjunto de técnicas y procedimientos que permiten conducir eficientemente una aeronave a su destino. Los aspectos clave de este proceso son la planificación del vuelo, su ejecución por la tripulación de la aeronave, y el seguimiento y ajuste del mismo por los controladores. Antes de discutirlos, hay que considerar cómo se organiza el espacio aéreo.

El espacio aéreo se divide en Regiones de Información de Vuelo (FIR, *Flight Information Region*). Cada país tiene un “área de responsabilidad” dentro de la cual existen varias FIRs. Cada una de estas regiones se puede dividir en espacio aéreo controlado y no controlado. El espacio aéreo controlado es aquel en el que se presta el servicio de control. Sin embardo, en el espacio aéreo no controlado los Servicios de Tráfico Aéreo (ATS, *Air Traffic Service*) prestan servicios de información y alerta, pero no existe ningún tipo de control. La Unidad encargada de entregar el servicio de control al tráfico aéreo en estas áreas recibe el nombre de Centro de Control de Área (ACC, Area Control Centre). Debido al amplio espacio aéreo que manejan, están divididos en Sectores de Control, cada uno responsable de una parte del espacio total a su cargo. Cuando un avión está a punto de salir de un sector es traspasado al siguiente sector de forma sucesiva, hasta el aterrizaje en su destino. Actualmente, la mayor parte de las rutas aéreas están cubiertas por radares, lo que permite hacer un seguimiento permanente de los vuelos.

En las regiones de información de vuelo se encuentran las áreas terminales de los aeropuertos importantes y entre ellas discurren las aerovías, pasillos por los que circulan las aeronaves. Otros elementos son las áreas prohibidas, restringidas o peligrosas que son zonas donde el vuelo de aeronaves se ve restringido en diferentes medidas y por causas diversas.

Todos los vuelos, tanto del espacio aéreo controlado como del no controlado, deben realizarse bajo unas normas que regulan la circulación aérea que se recogen en el Reglamento de Circulación Aérea.

Siguiendo con la organización del espacio aéreo, el máximo el responsable del control de tránsito aéreo es el controlador de tráfico aéreo. Esta es la persona encargada de dirigir el tránsito de aeronaves en el espacio aéreo y en los aeropuertos, de modo seguro, ordenado y rápido, autorizando a los pilotos con instrucciones e información necesarias, dentro del espacio aéreo de su jurisdicción, con el objeto de prevenir colisiones, principalmente entre aeronaves y obstáculos en el área de maniobras.

El controlador es responsable de las aeronaves que vuelan en un área tridimensional del espacio aéreo conocido como área de control, área de control terminal, aerovía, etc. Cada controlador ha de coordinarse con los controladores de sectores adyacentes para planificar las condiciones en que una aeronave ingresará en su área de responsabilidad, entregando dicho vuelo sin ningún tipo de conflicto respecto de otro tránsito, condición meteorológica, posición geográfica o de altitud (nivel de vuelo), siendo esto válido, tanto para vuelos nacionales como internacionales.

Los controladores trabajan en los Centros de Control de Área (ACC, Area Control Centre), en la Torre de Control o la Oficina de Control de Aproximación, donde disponen de varios sistemas electrónicos y de computación, que les ayudan en el control y gestión del tráfico, como el Radar (RDR, radio detection and ranging), que es un instrumento emisor/receptor de ondas de altísima frecuencia, el cual detecta los objetos que vuelan dentro de su espacio aéreo y, a través de programas computacionales, los presenta en las Pantallas Radar, que les facilitan la gestión y progreso de los vuelos en sus posiciones de control. Existen otros programas de asistencia, como los que ajustan las pistas disponibles, tanto para despegue como aterrizaje de aviones y el orden en que los vuelos han de despegar y aterrizar para optimizar el número de vuelos controlables.

Para volar en el espacio aéreo, cualquier aeronave tiene que contar con un *plan de vuelo*. Se trata de un informe donde se indican todos los datos referentes al vuelo, incluyendo el lugar y hora de salida y los de destino, la altitud y velocidad de crucero durante el vuelo, los puntos intermedios por los que ha de pasar el vuelo, el tipo de vuelo a realizar, y la información referente al avión. cuando alguien quiere referirse a estos puntos intermedios que hay en el plan de vuelo, que son puntos de referencia tridimensionales utilizados en la navegación por los que pasará la aeronave antes de llegar al destino, generalmente, la terminología más utilizada es referirse a ellos como *waypoints*. Antes de iniciarse el trayecto el plan de vuelo ha de ser notificado a la autoridad ATS competente para que sea aprobado.

La forma de ejecutar el plan de vuelo no es única, sino que depende del tipo de infraestructura disponible (ej. radiobalizas e indicadores de aproximación), del tipo de aeronave (ej. su capacidad de vuelo y pasajeros desplazados) y de la zona (ej. aeropuertos y baja altitud). Según estos factores es posible realizar navegación con Reglas de Vuelo Visual (VFR, *Visual Flight Rules*) o con Reglas de Vuelo Instrumental (IFR, *Instrument Flight Rules*).

Las VFR se pueden aplicar con una visibilidad mayor de 5 millas náuticas y techo de nubes por encima de los 1500m). En este caso los pilotos suelen usar la técnica de “navegación por estima”. Este procedimiento infiere la ubicación actual haciendo cálculos basados en el rumbo y la velocidad de navegación a lo largo de un período, combinados con observaciones visuales. Estos datos y cálculos se contrastan con cartas o mapas de navegación aérea. También emplea radio para hacer validaciones adicionales.

Las IFR son más utilizadas en aviones de línea puesto que ofrecen mayor seguridad. Aquí los pilotos navegan usando exclusivamente instrumentos y ayudas de navegación por radio, o directamente bajo las órdenes de controladores aéreos. El responsable en tierra de la navegación aérea es el Control de Tráfico Aéreo (ATC, *Air Traffic Control*), que gestiona el tráfico aéreo haciendo uso de la información suministrada por los pilotos y por los sistemas de radar.

Estado del arte

El presente trabajo se relaciona con varias líneas de investigación. Por una parte, su interés en el modelado de los aspectos humanos relacionados con el tráfico aéreo lo conectan con la investigación acerca de los Factores Humanos (HF, *Human Factors*) que intervienen en los vuelos, en particular aquellos relacionados con las tripulaciones, los pilotos o los controladores. Por otro lado, es preciso analizar también las simulaciones existentes sobre el espacio aéreo. Aunque en la mayor parte de los casos los actores participantes serán personas, se ha señalado en la introducción como este trabajo considera la introducción de UAVs en el espacio aéreo civil. Por ello se dedicará un apartado para analizar los trabajos que se centran en estas aeronaves. Este análisis se realizará sobre trabajos que se dediquen tanto a la programación de estos vehículos, como a los protocolos y la reglamentación que han de cumplir para poder ser integrados en el espacio aéreo civil.

Factores humanos en aviación

Como se ha indicado anteriormente, hay numerosos procedimientos que se centran en mejorar la eficiencia de los factores humanos que intervienen en un vuelo. Estos factores son muy importantes ya que se considera que más de un tercio de los accidentes aéreos son debidos a errores humanos, y dentro de estos la mayoría son debidos a fallos en la comunicación, toma de decisiones o liderazgo (14). Por lo tanto, es razonable tener en cuenta estos factores a la hora de realizar una simulación del espacio aéreo civil.

El estudio de “*los Factores Humanos (o ergonomía) puede definirse como la tecnología orientada a optimizar las relaciones entre personas y sus actividades haciendo uso de la aplicación sistemática de las ciencias humanas, integrada en el marco de la ingeniería de sistemas*” (15). De esta definición habría que hacer una serie de distinciones:

1. Al describir HF como una tecnología, se hace hincapié en su carácter práctico; se orienta a problemas más que centrarse en la disciplina. La relación entre HF y las ciencias humanas podría compararse con la relación entre ingeniería y ciencias físicas.
2. La ergonomía implica un interés por la comunicación entre individuos y en el comportamiento de grupos de personas.
3. La ergonomía se ha extendido desde ámbito laboral al del hogar, hospitales, escuelas, e incluso a las actividades de ocio.
4. Las ciencias humanas comprenden los estudios de la estructura y naturaleza del ser humano, sus capacidades y limitaciones, y sus comportamientos ya sea en solitario o en grupo. En común con todas las tecnologías, HF se preocupa de la búsqueda y uso de conceptos y datos seleccionados según su importancia en un problema práctico.
5. Con el fin de contribuir eficazmente en el diseño y las operaciones del sistema, es necesario que los ergónomos integren sus contribuciones con las de otros expertos.
6. La optimización en las relaciones personales viene dada por dos conjuntos de criterios: el bienestar humano y la eficacia del rendimiento del sistema.

El anterior foco hace especialmente apropiada la investigación en HF para el propósito del presente trabajo. Presta especial atención a las interacciones entre personas y a los factores individuales que afectan a su desempeño.

Dentro de los estudios en HF, este análisis se centra en dos ampliamente usados en la aviación comercial: la gestión de recursos por la tripulación y el entrenamiento de vuelo orientado.

La industria aeronáutica incorporó hace décadas (16) el procedimiento de gestión de los recursos por la tripulación (CRM, *Crew resource management*) como parte fundamental del entrenamiento y formación de las tripulaciones de aviones. El CRM se puede definir como “el uso eficiente de todos los recursos disponibles por la tripulación del avión, incluyendo el equipamiento, las habilidades tanto técnicas como procedimentales, y las contribuciones del resto de la tripulación del vuelo y demás personas” (17). A lo largo de los años, el CRM ha ido evolucionando (14) pasando a ser básico en cuanto a la coordinación de la tripulación y la toma de decisiones. Entre los principales objetivos del CRM se encuentran evaluar, desarrollar y mejorar el trabajo en equipo, la habilidad en la toma de decisiones, la conciencia situacional, el pensamiento crítico y las comunicaciones interpersonales. De los artículos (18) (19) (20) se puede resumir que las temáticas principales del CRM incluyen:

* Procesos de comunicación y toma de decisiones. Comunicación entre los distintos miembros de la tripulación y con el exterior, y las distintas acciones a tomar dependiendo de lo que se transmita en estas interacciones.
* Trabajo en equipo y dinámicas de grupo. Para poder abordar de forma conjunta y lo más rápido posible los problemas en situaciones críticas.
* Liderazgo y coordinación de actividades. El capitán de la aeronave es en última instancia el responsable dentro del vuelo, por lo que ha de saber dirigir y coordinar a su equipo.
* Carga de trabajo y conciencia situacional. Cada miembro del equipo debe tratar de dedicarse exclusivamente y lo más eficientemente posible a sus competencias.
* Rendimiento humano y psicológico. Las capacidades de atención, estrés y fatiga se irán deteriorando a lo largo del vuelo.
* Planificación. Capacidades de planificación de la ruta y tiempos de esta.

Una forma fácil y efectiva de comprobar la eficiencia de los procedimientos CRM es a través de las prácticas en simuladores. El uso de simuladores de vuelo proporciona a los estudiantes el conocimiento, habilidad y experiencia necesarios para operar eficientemente como miembro de una tripulación. En el artículo (21) además de proporcionarnos una herramienta educacional económica con la que se pueden enseñar los fundamentos del CRM, se enumeran los beneficios que nos aportan los simuladores de vuelo a la hora de instruir a los pilotos, éstos son:

1. Un entorno físico controlado.
2. Control de la carga de trabajo de la tripulación por parte de un instructor.
3. Control de los sistemas de la aeronave por parte del instructor.
4. El simulador puede facilitar escenarios de vuelo que de otra manera serían imposibles o muy arriesgados.
5. Los simuladores tienen la opción de congelar situaciones y repetirlas permitiendo opiniones y correcciones inmediatas cuando se necesitan.
6. Muchos de los simuladores pueden proveer opciones de grabación lo que permite el posterior análisis crítico de las decisiones tomadas.
7. Los simuladores proporcionan la posición instantánea de las aeronaves.
8. Un instructor puede monitorizar fácilmente las conversaciones y acciones de la tripulación sin distraerles.
9. Los simuladores pueden suministrar distintos tipos de aviones, sistemas de control de las aeronaves y otro equipamiento que no esté disponible, sea económicamente fácil de adquirir o apropiado usarse en entrenamientos.
10. La personalización de controles y cabinas se puede reconfigurar fácilmente.
11. La configuración de las aeronaves y las características de los vuelos es programable.
12. Una variedad de ayudas electrónicas a la navegación pueden ser simuladas.

El segundo procedimiento típico de HF considerado aquí es el LOFT (Entrenamiento de Vuelo Orientado) (16). Una aproximación enfocada en LOFT es un tipo de entrenamiento donde se simula un vuelo con la tripulación completa del aparato. La simulación incluye planificación de vuelos orientados, donde las compañías diseñaran las rutas con los posibles incidentes que puedan surgir durante su realización, documentación completa de los vuelos, actividades previas a los vuelos, procesos y actividades de distribución de tareas, operaciones en ruta e información post-vuelo (22). En todos los casos se consideran procedimientos normales, anormales y de emergencia típicos en vuelos de ruta. Un instructor monitoriza el rendimiento y el resultado de toda la tripulación, tanto en el simulacro actual como en su histórico, para mejorar rendimiento o arreglar fallos de procedimientos en la toma de decisiones o en la manera de actuar.

Este tipo de estudios, pruebas y entrenamientos HF han permitido establecer que saber controlar factores como el estrés y la fatiga es esencial para poder llevar a cabo un vuelo sin ningún incidente significativo. En el caso concreto de los pilotos, es importante que estos controlen el tipo de pilotaje que realizan, aspecto relacionado con su personalidad (23). También es clave su experiencia, determinada por las horas de vuelo que han acumulado. Ésta. Esta le permitirá resolver más rápido y de manera más eficaz situaciones comprometidas que se haya podido encontrar en el pasado. Ejemplo de ello es que todos los pilotos, después de adquirir el título de pilotaje, deben acumular una serie de horas de entrenamiento en simuladores antes de empezar en los vuelos reales.

Simulación y Gestión de Tráfico Aéreo

Como se ha comentado en la introducción actualmente hay numerosas aplicaciones que se dedican a la simulación aérea. Estas simulaciones se pueden clasificar atendiendo a varios criterios. En la introducción ya se apuntó el foco de su estudio: interesa saber cómo se están simulando a nivel individual cada una de las aeronaves y que ocurre cuando interaccionan varias de ellas y además lo hace también con el control.

Las simulaciones de aeronaves individuales se centran en los aspectos físicos de su vuelo, su manejabilidad y sus sistemas de control.

Por el lado de las aplicaciones centradas en los aspectos físicos y más concretamente en los aspectos característicos del vuelo, se puede destacar una serie de herramientas desarrolladas sobre JSBSim. JSBSim(3) que es una librería software de código abierto que permite modelar esta dinámica de vuelo de las aeronaves, el denominado modelo físico de los movimientos de los aviones. Entre las herramientas construidas sobre JSBSim se encuentran FlightGear (4) que simula el pilotaje de un avión, y OpenEaagles (5), que es un marco de aplicaciones diseñado para facilitar la construcción de aplicaciones de simulación en tiempo real usando software orientado a objetos.

Entre las características más destacadas del FlightGear se encuentran:

* Información extensa y realista de elementos físicos como aeropuertos, ciudades, montañas, ríos, etc.
* Amplia variedad de aeronaves.
* Modela de una forma realista el comportamiento de los instrumentos del mundo real. Incluso reproduce de forma precisa los fallos de muchos sistemas e instrumentos.
* Opciones de simulación en tiempo real.

FlightGear se ha utilizado en una serie de proyectos en el mundo académico y la industria (incluida la NASA). Esta herramienta ha sido empleada en numerosos proyectos de universidades de todo el mundo, podemos destacar:

* *Universidad Tecnológica de Delf (Delf, Holanda)*, donde usan el FlightGear con el objetivo de diseñar, probar y evaluar las distintas técnicas de computación que se pueden usar para el desarrollo de los sistemas inteligentes de asistencia a la tripulación en situaciones de máxima atención (24).
* *Universidad de Gales (Aberystwyth, UK)*, el Grupo de Inteligencia Robótica está usando el FlightGear como parte de la investigación en robots aéreos para diseñar vehículos aéreos que puedan realizar operaciones en atmósferas de otros planetas (25).

En cuanto al ámbito empresarial, los proyectos que se pueden destacar son:

* *NASA/Ames Human Centered System Lab*, donde se han desarrollado simulaciones a escala real de cabinas de aeronaves.
* *Aeronautical Development Agency*, (Bangalore, India). FlightGear se usa como generador de imágenes de simulaciones de vuelo para la evaluación experimental de recuperación de aviones de combates de portaviones arrestados.

Por otro lado la herramienta OpenEaagles contiene un extenso conjunto de objetos y clases para crear rápidamente instrumentos de aeronaves. Aprovechando esta capacidad se pueden construir simulaciones complejas, muy interactivas y que implementan la funcionalidad completa de los instrumentos.

El hecho de que las librerías gráficas y de instrumentos están desarrolladas por separado permite que ambas capacidades puedan ser mezcladas con otras soluciones. Sin embargo, se obtiene un mayor rendimiento cuando simulación y gráficos se usan conjuntamente. Por lo tanto, OpenEaagles no se queda sólo en proporcionar librerías de clases para dibujar vuelos en OpenGL y DirectX, que son las plataformas más usadas para producir gráficos en 2D y 3D, sino que también proporciona herramientas para diseñar simulaciones complejas.

Otra de las características importantes de OpenEaagles es la interoperabilidad. La interoperabilidad es la capacidad de trabajar conjuntamente con distintos sistemas y organizaciones. Cuando hablamos de interoperabilidad en el ámbito del software nos referimos a la capacidad entre los distintos programas de intercambiar datos a través de un conjunto común de formatos, para poder leer y escribir sobre los mismos formatos de archivo y usar los mismos protocolos.

Otra característica importante de la que dispone OpenEaagles es que soporta el estándar DIS (Distributed Interactive Simulation), que es un estándar abierto, definido por la IEEE (Institute of Electrical and Electronics Engineers), para llevar a cabo, en tiempo real, juegos de guerra a nivel de plataforma a través de múltiples computadoras centrales.

Algunos ejemplos de aplicaciones donde se usa OpenEaagles son:

* OpenSpace - An Apollo Command Service Module and Lunar Module Simulation. Fue diseñada para mostrar el uso de OpenEaagles en un entorno algo distinto a la simulación de vuelos. La simulación contiene el módulo aeroespacial Apollo CSM (Command Service Module) y un módulo lunar LM (Lunar Module) en órbita con la tierra (al estilo Apollo 9). El objetivo es que el operador, que tiene el control del LM, realice un acoplamiento con el CSM.
* LifeForce - A First Person Shooter. Esta aplicación se diseñó para ser la primera aplicación donde se puede controlar una persona que puede andar y correr sobre un terreno a la vez que hay otros elementos en el aire y en el terreno. Durante la simulación los aviones, que estará manejados por pilotos, podrán despegar, aterrizar o interactuar con los caminantes. Usando esta aplicación un caminante puede, por ejemplo, mirar al cielo y ver en vivo combates entre aviones.

Otro trabajo importante en el ámbito de la investigación, desarrollado por la universidad de Nápoles y centrado también en los aspectos físicos, es el AVDS (Simulador de Diseño Visual de la Aviación) (6). AVDS es un conjunto de herramientas de simulación y visualización de vuelo diseñadas para ser usadas por ingenieros, estudiantes, educadores y otros investigadores interesados en la investigación y desarrollo aeroespacial. Están desarrolladas en Simulink (26), que es un entorno sobre MATLAB para la simulación y diseño basado en modelos de sistemas dinámicos y embebidos. Con estos modelos, los creadores del AVDS comentan que esta herramienta facilita una mejor comprensión de la dinámica de los sistemas y permite el uso de simulaciones realistas en ordenadores de bajo coste (6). Entre otras cosas el AVDS se caracteriza porque la visualización combina los parámetros del estado de los aviones como un flujo de imágenes del movimiento de las maniobras de estos. A su vez, permite rápidas iteraciones de los nuevos diseños durante entornos de simulación en tiempo real gracias a la interacción con paquetes de vuelo. Las características estándar incluyen cartas de navegación definibles en pantalla, superficies articuladas, datos del de terreno y tres cámaras virtuales. Por último mencionar que la arquitectura de red para desarrollad aplicaciones distribuidas está construida basándose en los estándares de los protocolos de Internet y permite un fácil integración con herramientas desarrollas con esta aplicación.

Por otro lado, también es importante saber cómo se han representado entornos complejos donde toman parte multitud de diferentes tipos de elementos relacionados con la navegación. Es por ello, por lo que existen una serie de herramientas dedicadas a la simulación del tráfico aéreo y su gestión.

En cuanto a otras herramientas de simulación en primer lugar se enumerarán una serie de herramientas que tiene en común que comparten un enfoque ABMS (Simulación y Modelado Basado en Agentes) (32). De este modo se tiene que en el documento (27) se hace un resumen de los distintos entornos de simulación en el ámbito de la investigación, y cuya finalidad es presentar el diseño y la metodología de una simulación multi-agente de la ATFM (gestión del flujo del tráfico aéreo) usando diferentes estrategias de selección de rutas simples. En dicho trabajo se presenta el concepto CATFM (Gestión del Tráfico Aéreo Colaborativo) que supone la existencia de múltiples entidades independientes con sus propias creencias y deseos. Este concepto aumenta el intercambio de información y distribuye alguno de los elementos de decisión.

Dentro del enfoque ABMS también se tiene la herramienta IMPACT (9), donde se modelan los agentes de la FAA y de las aerolíneas. Dicho modelado se hace usando un diseño basado en enjambre y tiene como finalidad evaluar el impacto de tres factores: la planificación anticipada, los programas de retrasos en tierra (GDPs, *Ground Delay Program*) sin intercambio de información y los horarios de vuelo compartido. Los agentes de las aerolíneas toman sus decisiones basándose en el cálculo del coste de cada tarea realizada. Mediante la imposición de determinados acontecimientos al azar en el comienzo, el sistema devuelve como salida de la simulación una serie de estadísticas basadas en el comportamiento emergente de los agentes en el sistema. El enfoque de IMPACT es modelar las aerolíneas y la FAA basándose en el uso de un algoritmo simple de toma de decisiones, y centrarse principalmente en la gestión de retrasos.

Otros ejemplos de herramientas ABMS son ACES (Sistema de Evaluación de los Conceptos Aéreos)(28) y MIDAS (Sistema de análisis y diseño integrado de la interacción hombre-máquina).

ACES es una herramienta de simulación basada en agentes distribuidos del NAS (Sistema Nacional Aeroespacial de EE.UU.), que incluye pero no se restringe al ATFM. El núcleo de ACES se basa en la arquitectura HLA (High-Level Architecture), que es una aproximación para implementar sistemas de simulaciones distribuidos. En dicho estándar, desarrollado por el Departamento de Defensa de EE.UU., se definen un conjunto de herramientas, reglas y protocolos para implementar estos sistemas de simulación distribuidos. Por otro lado, el MIDAS es un modelo basado en agentes que se centra en la eficiencia humana cuando interacciona con interfaces máquina. Esta herramienta ha sido usada con ATFM y enfatiza principalmente las capacidades y limitaciones de las facultades cognitivas de los humanos en vez de la complejidad de la toma de decisiones.

Por último destacar dentro de este tipo de simulaciones, que la NASA ha desarrollado una herramienta denominada FACET (30). Un sistema que sirve para simular el flujo del tráfico aéreo y que contiene herramientas que se centran en el modelado de la trayectoria y del tiempo, y también contiene un modelo de la estructura del espacio aéreo, incluidas las regiones ARTCC (Centro de control del tráfico de las rutas aéreas), sectores y rutas aéreas. FACET puede actuar como un simulador o como un mecanismo de reproducción, ya sea a partir de datos históricos o de una fuente de datos reales de la FAA. Esta herramienta ha sido integrada en un producto comercial, Flight Explorer (31), que se usa en la mayoría de las principales compañías aéreas de EE.UU. No es sólo una simulación basada en agentes, que se concentra en los aspectos físicos del flujo de tráfico aéreo, sino que incluye también otros aspectos, tales como la carga de trabajo del controlador y las iniciativas de gestión del tráfico aéreo.

Por otro lado, tenemos una serie de sistemas que se centran en el estudio de los procesos colaborativos y de coordinación entre los distintos elementos que intervienen en la gestión del tráfico aéreo, Ejemplo de ello es el sistema STEAM (29), que es una herramienta que permite la representación explícita de los objetivos, planes y compromisos de conjuntos de equipos. Se usa para evaluar un sistema colaborativo al sincronizar el tráfico en tiempo real. Dicha sincronización es aquella a la que se dedican los controladores de sectores individuales cuando gestionan vuelos que pasan por múltiples sectores.

Otro ejemplo de este tipo de sistemas es el mostrado en el documento (30). Donde se muestra un algoritmo adaptativo y distribuido del ATFM que puede implementarse y probarse de manera sencilla con la herramienta FACET, el método está basado en una serie de entidades que representan ciertas regiones y donde cada agente determina la separación entre los aviones que se aproximan a la región. Estas entidades usan un aprendizaje por refuerzo para aprender políticas de control. El artículo (30) se centra en estudiar las distintas funciones que premian o asignan pesos al algoritmo de aprendizaje y las diferentes formas de estimar estas funciones.

Por último, el trabajo (34) describe la experiencia de usar una arquitectura de agente BDI (modelo de Creencia-Deseo-Intención) (35) para desarrollar una simulación de un ATFM colaborativo y los problemas de eficiencia que encuentran. Como conclusión propone una aproximación híbrida donde se combinan elementos del sistema implementados con agentes BDI y componentes implementados sin ellos, dependiendo de una serie de recomendaciones:

1. Usar BDI para procesos cognitivos explícitos. El paradigma BDI se adapta perfectamente cuando se quiere modelar toma de decisiones que están bien definidas.
2. Importancia del nivel de granularidad. No hay necesidad de modelar detalles de toma de decisiones cuando uno está más interesado en el resultado que en el proceso mismo.
3. Considerar las propiedades de ejecución del lenguaje. Toda implementación BDI se convierte en código ejecutado en un ordenador, y entender las propiedades del algoritmo a veces es necesario. Por otro lado, ya que la eficiencia se degrada con el número de agentes introducidos, a veces es conveniente combinar procesos de múltiples agentes en un sólo agente.

Trabajos de investigación sobre UAVs

El uso cada vez mayor de UAVs en diferentes tareas, tanto militares como civiles, así como el reducido coste de un UAV con respecto a los vehículos tripulados, ha llevado a que la producción de UAVs haya alcanzado unos niveles fiables y rentables. Esto ha producido que una serie de organizaciones tales como la FAA o Eurocontrol (Organización Europea de la Seguridad de la Navegación Aérea) se hayan comprometido a hacer uso del potencial de estas nuevas tecnologías para solucionar parte de los problemas de saturación aérea. A raíz de esto se han propuesto como objetivo la introducción, regulación (36) y mantenimiento de vuelos coordinados de UAVs dentro del espacio aéreo civil. Esto conlleva inevitablemente que se tengan que hacer pruebas y simulaciones para comprobar que dicha regulación y mejora tecnológica se adaptan adecuadamente a la situación actual de tráfico aéreo antes de su implantación real.

Hay numerosos trabajos que se centran en la investigación y mejora del manejo de UAVs. En primer lugar hay que destacar que para la introducción de UAVs en el espacio civil actual es necesario establecer una serie de procedimientos y regulaciones que dichos vehículos deben cumplir. De hecho, si la introducción de este tipo de vehículos en el espacio aéreo se ha visto retrasada no ha sido porque existan limitaciones tecnológicas que aún no han sido resueltas, sino porque había un déficit a la hora de definir los requisitos para que pudieran volar junto con tráfico tripulado y a la hora certificar estos. Para suplir estas deficiencias se ha dedicado el documento (37), que es un documento oficial donde se define un sistema ATM (Gestión del Tráfico Aéreo) que estaría debidamente preparado para cumplir las siguientes tareas:

* Establecimiento de los requisitos de aeronavegabilidad y reglamentos comunes para UAVs.
* Establecimiento de la regulación y estandarización del tráfico aéreo y los procedimientos derivados para la integración adecuada de UAVs en el Sistema de Gestión del Tráfico Aéreo.

Con vistas a analizar el comportamiento de los UAVs, tanto en vuelo con tráfico tripulado como en otras situaciones, se ha recurrido a simulaciones. Al igual que en los casos de vuelos tripulados existen ejemplos centrados en el vuelo de UAVs individuales y en la gestión de grupos de estos.

En cuando al estudio de UAVs como elementos individuales destacar una serie de trabajos que se centran en modelar las características físicas de estos. Así por ejemplo es importante destacar que los sistemas de percepción de los UAVs pueden sufrir algún tipo de limitación debida a posibles oclusiones, es por ello, por lo que en (38) se propone un algoritmo para solucionar este problema. Dicho algoritmo modela las limitaciones de los vuelos de estos vehículos de una manera más realista a lo existente anteriormente, y nos confirma la importancia de la oclusión a la hora de la planificación de rutas en el vuelo.

Otro modelo de UAV que sigue este tipo de enfoque físico puede verse en (39) donde se presenta un modelo genérico de UAV a escala reducida enfocado en las claves de los esfuerzos físicos que actúan en la dinámica del avión (física del avión) con el fin de ser lo suficientemente simple como para diseñar un controlador del vehículo.

Por otra lado, el enfoque basado en agentes también se ha intentado aplicar en el tecnología de lo UAVs. Así se puede ver como el trabajo (40) presenta un framework basado en agentes para modelar UAVs, en donde cada vehículo de este tipo se descompone en una serie de niveles controlados por agentes. Los principales agentes serán el físico y el lógico. El agente físico será el principal responsable de las interacciones físicas del UAV, como pueden ser maniobras hacia el siguiente waypoint (punto intermedio), comprobaciones para asegurarse que no va a colisionar con ningún objeto, actualización del estado del combustible, comprobación de los sensores, transmisión de mensajes pendientes, etc. Por otro lado, el agente lógico es el responsable de generar la lista de waypoints y mandársela al agente físico, y tiene una serie de tareas como decirle a un agente físico transmisor que mande un determinado mensaje.

Requisitos del Sistema

En este apartado se van a especificar los requisitos básicos que el sistema debe cumplir. El objetivo es obtener una simulación del espacio aéreo suficientemente fidedigna para modelar fenómenos que involucran a varios actores y flexible para adaptarse a nuevos o modificados elementos en el dominio. Para este fin, y en función del estudio del estado del arte de la anterior sección se plantean los requisitos discutidos en esta sección. Estos se organizan en las siguientes secciones: el vuelo, las aeronaves, los pilotos y los controladores.

El vuelo:

Los planes de vuelos son los informes que contienen toda la información referente a un vuelo y que son suministrados por las compañías a los pilotos antes de iniciar el vuelo. De este modo habrá unas entidades encargadas de crear los planes de vuelo que enviarán estos planes a los pilotos correspondientes para que inicien el vuelo.

Los planes de vuelo oficiales suelen contener numerosa información pero se va a restringir a los datos que son imprescindibles para realizar el vuelo. Estos datos son:

* El aeropuerto de origen.
* El aeropuerto de destino.
* La hora de despegue.
* La velocidad de crucero del avión. (determinado por el modelo del avión)
* La altura de crucero del avión.
* La lista de waypoints por los que tiene que pasar el avión antes de llegar al destino.
* Aeropuerto alternativo de destino.
* El piloto al que se le asigna el plan de vuelo.
* El avión al que va dirigido este plan de vuelo.

La creación y coordinación de planes de vuelos es un proceso bastante costoso que no es objeto de estudio en este trabajo. Es por ello, por lo que, para simplificar este proceso, la asignación tanto del origen como del destino de un plan de vuelo se hará de manera automática y aleatoria, sólo restringiéndose a que no coincidan. Del mismo modo los waypoints también serán elegidos de manera aleatoria pero con alguna restricción, con el objetivo de que un avión no se desvié mucho de la ruta más corta y sea un poco más realista. Dicha restricción es que, a la hora de elegir un nuevo waypoint, la distancia recorrida por el avión pasando el nuevo waypoint no será mayor de 1.5 veces a la distancia que se recorrería si no se tuviera que pasar por ese waypoint.

En cuanto a la velocidad y altura de crucero estarán establecidas siempre a un valor por defecto que vendrá determinados por las especificaciones técnicas de los aviones. Así por ejemplo, cómo podemos ver en los documentos (41) (42), para los Airbus A-320 dichos valores se corresponderían con:

* Velocidad de crucero: Mach 0.76 (930 Km/h)
* Altura de crucero: 36000 feet (11000 Km)

Por último, tanto el piloto como el avión asignados a este plan se corresponderán con los primeros que el sistema vea disponibles, es decir, que no tengan otro plan asignado.

Una vez los pilotos dispongan de su plan de vuelo y se dispongan a realizar sus respectivos trayectos no se van a considerar ni el despegue ni el aterrizaje de los aviones, ya que no son muy relevantes para este estudio. Por lo tanto, la simulación se centrará más en el vuelo y las características de los aviones durante las rutas de vuelo. Éstas consistirán en realizar el trayecto desde el aeropuerto de inicio al de destino pasando por los waypoints que vengan determinados en el plan de vuelo, sin que se produzca ningún incidente grave durante el trayecto como pueden ser colisiones con otras aeronaves.

En cuanto a los datos que se usarán en este ámbito, y teniendo en cuenta que el modelo es fácilmente extensible a todo el mundo, se va a restringir el entorno a los principales aeropuertos y waypoints españoles de la península ibérica (43).

Las aeronaves:

Dentro de los factores a tener en cuenta de un avión las características más relevantes son aquellas que describen su vuelo. Se puede hacer una diferenciación entre aquellas ligadas al tipo de avión y las que lo están al vuelo.

En cuanto a las características ligadas al tipo de avión se puede decir que son aquellas que limitan las capacidades físicas del vuelo. Dentro de este tipo de características, para esta aproximación se han tenido en cuenta la velocidad y la altitud. El tipo de avión limita estas características con valores máximos y mínimos, aunque en algunos casos uno de estos límites puede ser prescindible, y establece un valor de crucero recomendado. Un avión concreto tendrá un valor actual de cada una de estas características entre los límites fijados por su tipo de avión. A modo de ejemplo de tipo de avión se incluye el Airbus A-320 cuyas características se pueden ver en (41) (42). Dicho modelo de avión es el más utilizado por Iberia, la mayor compañía aérea española (44).

Las características ligadas al vuelo identifican aspectos que no están limitados por el tipo de avión. Ejemplos de estas características, que se tendrán en cuenta en este sistema, serán el rumbo y la posición actual.

Las posibles maniobras se pueden realizar sobre los aviones se abstraen como modificaciones de los valores de algunos de los atributos previos, en concreto del rumbo, la altura y la velocidad. El resto de los valores de atributos se actualizarán automáticamente a partir de los ya mencionados.

Requisitos de los pilotos:

Los pilotos son las entidades encargadas de manejar y, por tanto, dar instrucciones a los aviones que estén pilotando. Los pilotos deben ser conscientes de su plan de vuelo durante todo el trayecto, pero deberán acatar las órdenes les puedan dar los controladores que en un momento dado. Por otro lado, se tendrán en cuenta algunas características de su comportamiento como pueden ser la experiencia, el estrés, la fatiga o el tipo de agresividad a la hora de ejecutar las maniobras.

Los UAV serán representados como pilotos donde, por ejemplo, tanto la fatiga como el estrés serán siempre cero y no se incrementará por muchas horas de vuelo que vayan acumulando. Lo cual no será así para los pilotos que son representados por personas, donde estos valores, asignados aleatoriamente en un principio, irán incrementando durante el tiempo e influirán negativamente en la toma de decisiones o a la hora de mandar ciertas instrucciones al avión.

Requisitos de los controladores:

Los controladores se limitarán, en esta primera aproximación, a gestionar las posibles colisiones que puedan surgir entre aviones. Dichas gestiones las harán basándose en las regulaciones y procedimientos descritos en el documento (45), donde se establecen los siguientes valores mínimos de separación entre aviones:

* Separación vertical:
  + 1000 pies (305 metros) cuando se vuela por debajo de los 29.000 pies (8.839 metros).
  + 2000 pies (610 metros) cuando se vuela por encima de los 29.000 pies (8.839.metros)
* Separación longitudinal: 5-10 millas náuticas (9.260 - 18520 metros).

No considerándose muy relevantes para el estudio de este trabajo otras posibles tareas que los controladores suelen tener en la realidad, como pueden ser la gestión de las aproximaciones a los aeropuertos o la gestión de aterrizajes y despegues.

Infraestructura y desarrollo

La aplicación ha sido desarrollada siguiendo la metodología INGENIAS (46) (47) y usando JAVA como lenguaje de programación. INGENIAS proporciona un lenguaje de modelado y un proceso software para la especificación de los agentes, su entorno, y el código asociado (48). Además dispone de la herramienta IDK, la cual soporta tanto el modelado como la generación de código de manera automática además de la creación de documentación HTML.

Las últimas versiones de INGENIAS se basan en la aproximación MDE, que es un paradigma de desarrollo de software que se centra en la creación y explotación de modelos de dominio (es decir, representaciones abstractas de los conocimientos y actividades que rigen un dominio de aplicación particular), más que en conceptos informáticos (o algoritmos) (11). Por lo tanto, INGENIAS se basa en la programación por modelos como principal forma de expresión. Estos modelos son creados incluyendo acciones de ejecución y a partir de ellos se genera el código, pudiéndose modificar o añadir código a mano a posteriori para especificar acciones concretas.

De esta forma INGENIAS permite desarrollar sistemas distribuidos complejos y dinámicos usando el paradigma de agentes. Un agente es una entidad software que contiene su propio hilo de control y esta modelada en cuanto a su comportamiento y conceptos sociales como objetivos, intenciones, tareas, roles, etc. Todos estos conceptos serán representados como ciertos componentes que se pueden añadir a los modelos. El hecho de usar los conceptos de modelado de agentes hace que el diseño y el análisis sean más fáciles de entender ya que estos conceptos son más cercanos a los comportamientos humanos (49). De esta manera en INGENIAS cuando un determinado agente puede satisfacer un objetivo, es decir, se dan todas las condiciones para que un objetivo pueda ser cumplido, entonces la correspondiente tarea, que tiene asociada el agente en el modelo y puede satisfacer ese objetivo potencialmente, es ejecutada.

Plataforma

Dada la aproximación MDE y ABM adoptada, el trabajo propuesto tiene como elemento clave los modelos basados en agentes de estos sistemas. A partir de estas especificaciones se genera la simulación en sí, cuyo diseño se basa en una arquitectura Modelo-Vista-Controlador (MVC, Model–view–controller) (50) de tres capas: lógica, directamente relacionada con los anteriores modelos; visualización, dependiente de la plataforma de destino; controlador, que sincroniza a los anteriores. Las siguientes secciones detallan estos componentes.

En esta sección se presentan los distintos diagramas del modelo diseñado en el IDK y la arquitectura del programa global y su interacción con la simulación. En primer lugar indicar que el sistema desarrollado es un MAS, por lo que los diagramas se centran principalmente en como los agentes realizan las distintas operaciones. Con la herramienta de IDK se ha generado una especificación donde se pueden ver y modificar todos estos diagramas. En el siguiente apartado de esta sección se entrará más en detalle en la especificación y en la explicación de cada uno de los diagramas.

Especificación

En esta sección se presentarán los flujos y diagramas más importantes, con el objetivo de que se pueda tener una idea bastante detallada del funcionamiento del sistema.

Casos de uso

En primer lugar se van a presentar los casos de uso principales para el agente piloto, que es el agente principal de este estudio. Como se puede observar en la Figura 1, el objetivo principal del piloto será llegar al destino sin sufrir ningún incidente grave, es decir, que la tripulación llegue al destino sana y salvo (“Arrive Safe And Sound”). Para ello el piloto participará en tres casos de uso principales, que se corresponden con tres partes claramente diferenciadas del vuelo. Dichos casos de uso serán:

* El despegue del avión. Donde se englobarán todas las tareas previas al arranque del avión y despegue del mismo. En esta apartado se incluiría, por ejemplo, el suministro de los planes de vuelo a cada piloto.
* La interacción con el entorno. Proceso que podría traducirse como pilotar el avión durante toda la ruta del vuelo, recibiendo señales externas y emitiendo señales al exterior.
* El aterrizaje del avión. Que en este trabajo tan sólo va a consistir en parar el avión.

Como se ha especificado en los requisitos y como se verá a continuación, este trabajo se ha centrado principalmente en el estudio del pilotaje del avión durante la realización del trayecto y las interacciones de los pilotos con el exterior que durante el vuelo puedan surgir.

Figura 1: Casos de Uso

Objetivos principales del sistema

La participación del piloto en los caso de uso anteriores viene dirigida por su persecución de varios objetivos De este modo en la Figura 2 se puede ver que lo primero que un piloto tendría que hacer antes de poder satisfacer el objetivo “Arrive Safe And Sound” es obtener el plan de vuelo (“Flight Plan Take”). Esto se debe a que ningún piloto (y por consiguiente avión) puede iniciar su vuelo sin que antes la empresa (en nuestro caso será un agente que explicaremos más adelante) haya suministrado un plan de vuelo a este piloto. Una vez que el piloto dispone de su plan de vuelo su próximo objetivo sería llegar al destino, siempre y cuando se eviten las colisiones con otros aviones y los conflictos que se puedan tener con obstáculos que se interpongan en la trayectoria de avión, como podrían ser montañas elevadas o zonas restringidas. Esto último se ve representado en el diagrama por los objetivos “Collision Avoided” y “Conflict Flag Generated”, y la satisfacción de estos es prioritaria al de la satisfacción del objetivo de llegar al destino “Arrive to Destination”.Este último objetivo se viene a descomponer a su vez en otro tres objetivos. Dichos objetivos son:

* Descomponer el plan de vuelo, “Flight Plan Descomposed”. Donde el plan de vuelo se descompondrá en una serie de tramos. A su vez se puede ver que, una vez se ha descompuesto el plan de vuelo en tramos, el piloto intentará satisfacer el objetivo “DecisionsForLegMade”. Dicho objetivo consistirá en crear las decisiones pertinentes para completar el tramo.
* Actualizar la posición del avión, “Plane Position Updated”. Que consistirá en una serie de tareas donde se tendrá en cuenta la velocidad, altura y orientación del avión para calcular la siguiente posición de la aeronave.
* Comprobar que se ha llegado al destino, “Flight Plan Completed”. Donde, cada vez que se complete uno de los tramos del vuelo, se comprobará si ese era el último tramo y se ha llegado ya al destino. Es por ello, por lo que este objetivo depende de que se haya cumplido el objetivo de tramo completado “Flight Leg Completed”.



Figura 2: Diagrama de Objetivos

Generación y obtención del plan de vuelo

En primer lugar, se explicará todos los procesos que se van a llevar a cabo hasta que el piloto pueda satisfacer el objetivo “Flight Plan Taken”. Para ello se ha de explicar que, como se introdujo en los requisitos, habrá un agente encargado de suministrar los planes de vuelo a cada uno de los pilotos. Este agente se denominará “FlightPlanner” y, como se muestra en la Figura 3, inicialmente tendrá en su estado mental crear los planes de vuelo.



Figura 3: Estado Mental "FlightPlanner"

A partir de este hecho, y como se muestra en la Figura 4, el agente procederá a crear los planes de vuelo para los distintos pilotos-avión que existan. Será dentro de estos planes de vuelo donde se almacenará esta información, es decir, a que piloto-avión se ha de enviar el mencionado plan de vuelo.



Figura 4: Tarea de Crear Planes de Vuelo

De este modo, una vez se tengan todos los planes de vuelo de los pilotos en activo, se procederá al envió de cada uno de ellos. Para ello se creará una interacción, como muestra la Figura 5, donde el “FlightPlanner” tomará el rol de iniciador de la comunicación y el piloto será el colaborador de esta. De esta forma el “FlightPlanner” pasará el plan de vuelo al piloto encapsulándolo en un campo del hecho denominado “PlanAnswer”. El piloto, accediendo a la interacción, consumirá este hecho y creará dos hechos en su estado mental:

1. Un hecho que representará el plan de vuelo (“Flight Plan”) que ha de seguir el piloto durante todo el trayecto, salvo que alguna circunstancia especial lo impida. Este plan de vuelo el piloto lo encapsulará, como puede verse en Figura 6, como un campo de su estado mental “Pilot Mind” para poder acceder a él en todo momento. Conviene decir que la tarea de “CreatePilotMind”, mostrada en la Figura 6, también tendrá la finalidad de suministrar los valores iniciales de experiencia, estrés y fatiga del piloto dependiendo de su perfil inicial.
2. Un hecho para arrancar el avión (“CanInitiateStartPlane”). Hecho que el piloto tendrá presente hasta que la hora de salida del vuelo que suministra en el plan coincida con la hora actual.

Por otro lado, el piloto creará otro hecho (“PlanReceived”) en la interacción, cuya finalidad será indicar que ha recibido el plan de vuelo y lo ha procesado satisfactoriamente. Una vez el “FlightPlanner” haya recibido el mencionado hecho, este procederá a finalizar la comunicación. En este momento se podría decir que el proceso de suministrar el plan de vuelo habría concluido y el objetivo de “Flight Plan Taken” se habría satisfecho.



Figura 5: Diagrama de Interacción Para Recibir el Plan de Vuelo Inicial



Figura 6: Creación del Estado Mental Inicial del Piloto

Seguimiento del plan de vuelo

Una vez el piloto dispone de su plan de vuelo su siguiente prioridad sería la de llegar al destino. Para explicar este proceso se ha hecho uso del siguiente diagrama de actividad (Figura 7), donde se muestra el desarrollo general del sistema en caso de que el trayecto no sufra ningún improvisto. De esta manera lo primero que hará un piloto sería recibir el plan de vuelo (“Take Initial Plan”), y que sería el resultado final de lo que se ha explicado anteriormente. A continuación el piloto arrancará el avión a la hora que el plan de vuelo indique debe partir (“Start Plane”). Una vez arrancado el avión se entra en un proceso iterativo hasta que se llegue al destino. Dicho proceso empieza con obtener el siguiente tramo a partir del plan de vuelo (“Flight Plan Monitoring”). Posteriormente se actualizará la posición del avión (“Update Plane Status”) y consultará si el tramo se ha completado (“Check Leg Completed”). En caso de que el tramo se haya completado pueden pasar dos cosas:

1. El tramo que se completó es el último del plan de vuelo, por lo que se satisface el objetivo de plan de vuelo completado y se procederá a parar el avión (“Stop Plane”).
2. En caso contrario, se intentará generar y pasar al siguiente tramo.

En otro caso, es decir, cuando el tramo todavía no se ha completado y después de actualizar la posición de la aeronave, el piloto creará las decisiones correspondientes para completar con éxito dicho tramo. Dichas decisiones posteriormente serán descompuestas (“Make and Descompose Decisions”) en una serie de instrucciones que el piloto ejecutará. Dicha ejecución consistirá en iniciar una interacción con el avión, donde el piloto le mandará las maniobras a ejecutar y chequeará que se hayan ejecutado con éxito (“Interaction Pilot-Plane”). Seguidamente se procederá a la actualización de la posición de la aeronave de nuevo. Hay que decir que, aunque el diagrama pueda llevar a pensar lo contrario, la actualización de la posición del avión no es dependiente de la ejecución de las instrucciones. Es decir, que esta posición será actualizada tanto si el piloto tiene que ejecutar instrucciones como si no tiene ninguna para ejecutar.

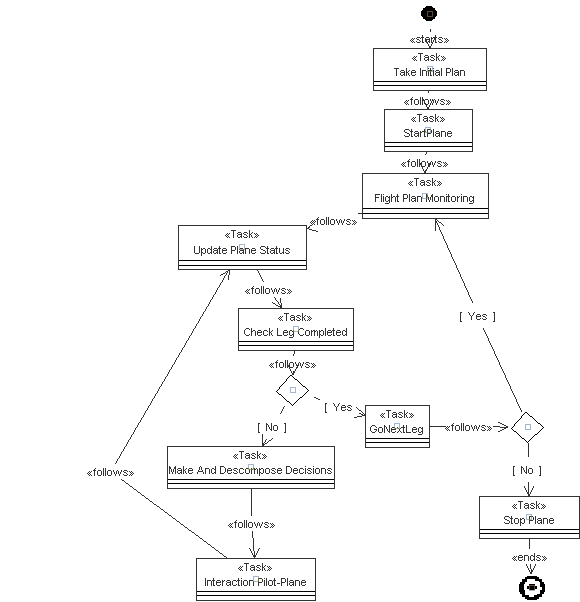


Figura 7: Diagrama de Actividad General

Conviene entrar más en profundidad en algunas de las tareas mostradas anteriormente para poder saber en qué consiste realmente cada una de estas acciones. Así, por ejemplo, se empezará por explicar cómo se realiza el arranque del avión. Como se comentó hace unos párrafos, el piloto, a la vez que recibe el plan de vuelo y, posteriormente, lo encapsula en su estado mental, crea un hecho denominado “CanInitiateStartPlane”. Dicho hecho le servirá al piloto, ayudándose de su estado mental para acceder al plan de vuelo, para iniciar una tarea que consistirá en comprobar si se ha cumplido la hora a la que el vuelo tenía programada su salida y así proceder a arrancar el avión. Este proceso se lleva a cabo en el diagrama mostrado en la Figura 8.



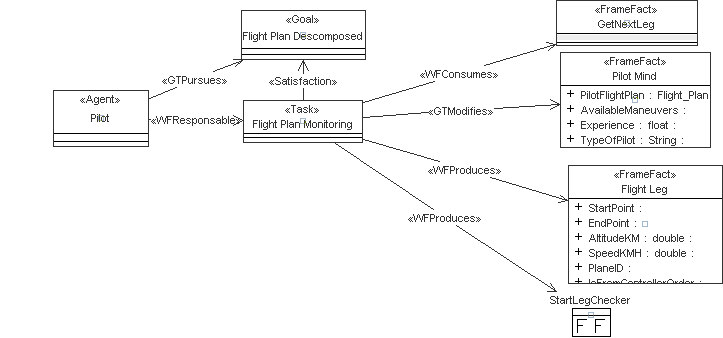
Figura 8: Comprobación de la Hora de Salida del Vuelo

Una vez que el piloto vea que la hora para iniciar el vuelo se haya cumplido, iniciará el arranque de los motores del avión y su puesta en marcha. Esto se traducirá en el diseño como la inicialización de la interacción entre el piloto y el avión que se muestra en la Figura 9. En esta interacción el piloto, que tomará el rol de iniciador de la conversación, primeramente accederá al plan de vuelo que dispone en su estado mental para saber que avión se le ha asignado para pilotar o, lo que es lo mismo, saber con qué avión tiene que comunicarse. A continuación creará la conversación con dicho avión y le pasará, a través de ella, el hecho (“TurningOnPlane”) que indica que el avión debe arrancarse y las coordenadas del aeropuerto de salida. Una vez el avión ha recibido este hecho actualizará su posición inicial e indicará con el hecho (“IniciateUpdateStatus”) que los valores de su estado mental se pueden empezar a actualizar teniendo en cuenta la velocidad y orientación que tiene éste en cada momento. Como última acción, el avión comunicará, a través de la interacción, que la realización de las tareas correspondientes a la inicialización del avión ha finalizado (“PlaneOn”). Una vez que el piloto ha recibido esta información éste podrá iniciar la descomposición del plan de vuelo y obtener el primer tramo que ha de realizar para completar el trayecto (“GetNextLeg”).



Figura 9: Interacción para Arrancar el Avión

El siguiente proceso que se va a explicar es el de la descomposición en tramos del plan de vuelo (Flight Plan Monitoring”). En dicho proceso realmente el piloto procederá a extraer el siguiente tramo a realizar del plan de vuelo, como se ve en la Figura 10. Para ello el piloto, una vez recibe el hecho que indica que puede proceder a obtener el próximo tramo (“GetNextLeg”), accederá a su estado mental para obtener el plan de vuelo y saber cuál ha sido el último tramo completado. A partir de esta información el piloto construirá el siguiente tramo que tiene que cubrir y producirá un hecho que iniciará el proceso que comprueba si el tramo se ha completado con éxito (“StartLegCheck”).

Figura 10: Descomposición del Plan de Vuelo

En caso de que el tramo no se haya completado todavía, el piloto pasará a tomar las decisiones oportunas para llegar al final del tramo. Pensar las decisiones correctas supone un tiempo para estos pilotos. Tiempo que dependerá de característicos como el estrés, la fatiga, la experiencia y el tipo de personalidad de dicho piloto. Es por ello, por lo que, se ha implementado un tarea que representa este tiempo como puede verse en la Figura 11. Para ello el piloto en primer lugar consumirá un hecho que dispara la tarea (“StartThinkNewDecision”), accederá al tramo en cuestión (“Flight Leg”) y a su estado mental, para saber los valores de sus características, características de las que hemos hablado anteriormente. Posteriormente, tras un tiempo que representará el tiempo que el piloto se ha tomado para procesar la información y producir la decisión, se producirá el hecho que indicará que ya se pueden crear las decisiones (“CanCreateDecision”).

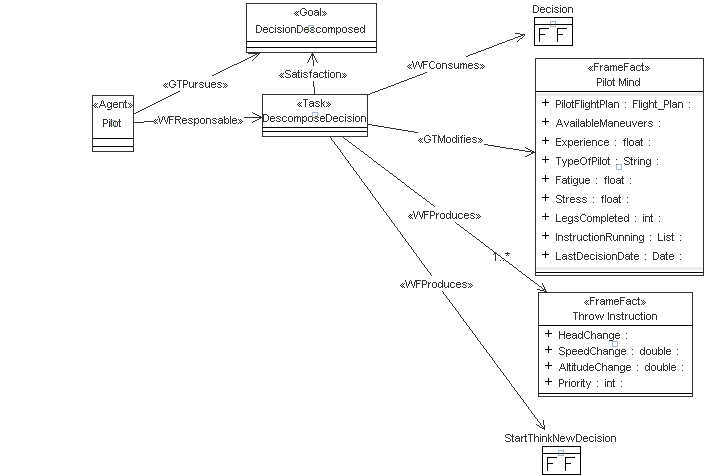
Figura 11: Tiempo de Creación de las Decisiones

Una vez el piloto se haya tomado su tiempo para decidir qué hacer para poder completar el tramo, pasará a crear dichas decisiones. De este modo, como podemos observar en la siguiente figura (Figura 12), el piloto creará las decisiones valiéndose del tramo a completar (“Flight Leg”) y accediendo a la aplicación del avión (“Plane Position Service”) donde puede obtener los valores de su estado actual, estado que vendría representado por la posición, la orientación y la velocidad del avión.

Figura 12: Creación de las Decisiones

Una vez el piloto ha tomado las decisiones para llevar el avión a su objetivo temporal y completar el tramo correspondiente en un determinado momento, pasará a descomponer estas decisiones en instrucciones a dar al avión. De esto se encarga la siguiente tarea (Figura 13), donde podemos observar que se hace uso de la decisión tomada y se produce una seria de instruccones (“Throw Instruction”), que contienen la información para generar la maniobra y a su vez iniciarán la interacción entre piloto y avión. Por otro lado, también se crea un hecho que indica que se puede proceder a pensar las siguientes decisiones que se van a llevar a cabo (“StartThinkNewDecision”). Esta tarea hace uso a su vez, del estado mental del piloto para saber si hay alguna instrucción que se esté ejecutando al mismo tiempo (campo “InstructionRunning”). En caso afirmativo se comprobará la prioridad de las instrucciones que se están ejecutando, y se detendrán todas aquellas instrucciones que tengan una prioridad menor que las instrucciones que genera esta tarea. En principio sólo hay tres tipos de prioridades en nuestro sistema. Estas son:

* Prioridad con valor 0: decisiones y/o instrucciones que provienen de los pilotos.
* Prioridad con valor 9: órdenes que vienen de los controladores.
* Prioridad con valor 10: instrucción de parar o aterrizar el avión.

Figura 13: Descomposición de Decisiones

La siguiente figura (Figura 14) viene a representar la ejecución por parte del avión de la instrucción que el piloto ha producido. Para ello se creará una interacción entre piloto y avión, donde el piloto interpretará el rol de iniciador de esta interacción mientras que el avión tendrá el rol de colaborador. De este modo haciendo uso de la tarea “Execute Instrucction” el piloto creará una interacción cuyo colaborador será el avión que está pilotando. Para averiguar este avión el piloto accederá al plan de vuelo de su estado mental “Pilot\_Mind.getPilotFlightPlan()” y dentro del plan de vuelo accederá a la identificación de la aeronave. El estado mental del piloto servirá también para indicar que está instrucción está siendo ejecutada actualmente. Por último, haciendo uso de la instrucción producida por el piloto, esta tarea generará la maniobra que el avión tiene que llevar a cabo “Manoeuvre”.

Una vez la maniobra llega al avión a través de la interacción este procede a actualizar los valores de los atributos de su estado mental “Plane Mind” de la aeronave. Dichos atributos serán, como se comentó en los requisitos, la velocidad, altitud u orientación del avión.

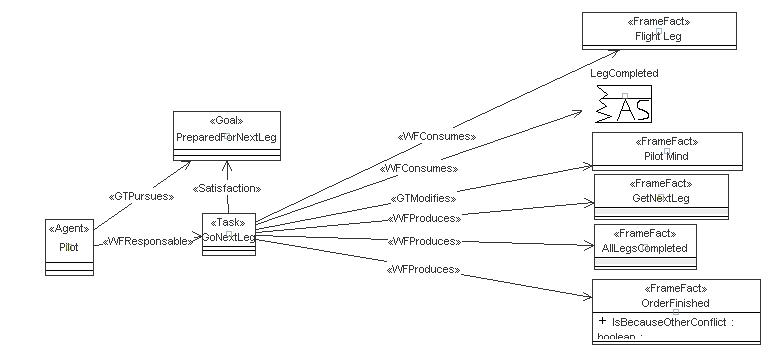
Por último, y después de comprobar que la maniobra ha finalizado, el piloto procederá a eliminar la instrucción de aquellas marcadas como que se estaban ejecutando dentro de su estado mental.

Figura 14: Interacción Piloto-Avión: Ejecución Maniobra

En caso de que se haya completado el tramo sobre el que se está volando, la aplicación interna “Leg Checker”, que se inició cuando se ejecutó la tarea “Check Leg Completed” la primera vez (Figura 15), lanzará un evento “LegCompleted”. Haciendo uso de este evento, del tramo que se acaba de finalizar y del estado mental del piloto se procederá a ejecutar la tarea “GoNextLeg” (Figura 16). Esta tarea tendrá la finalidad de confirmar que el tramo se ha finalizado y generar un determinado hecho dependiendo de las condiciones en que se ha acabado dicho tramo. Así por ejemplo, si el tramo fue generado como resultado de un conflicto entre aviones, o lo que es lo mismo a partir de una orden de un controlador, posteriormente se comentará como se generan estos tipos de tramos, la tarea generará el hecho de “OrderFinished” para poder seguir con las operaciones pertinentes a la finalización de estas órdenes. Por otro lado, y como se comentó anteriormente, si se trata de un tramo descompuesto del plan de vuelo entonces pueden ocurrir dos cosas:

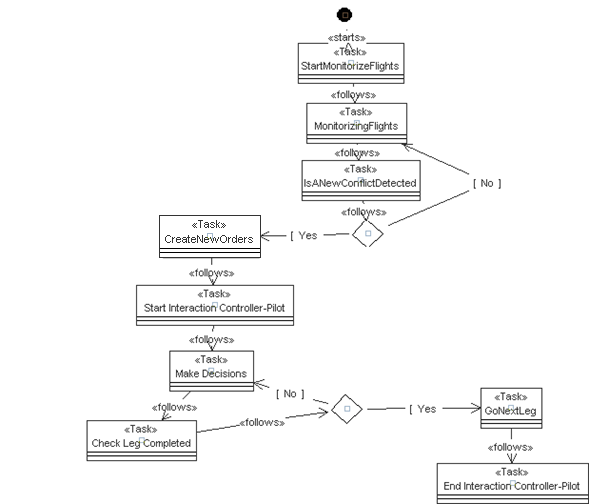
1. El tramo que se completó es el último del plan de vuelo, por lo que se satisface el objetivo de plan de vuelo completado y se procederá generar el hecho “AllLegsCompleted” que posteriormente iniciará la tarea de parar el avión (“Stop Plane”).
2. En caso contrario, se generará el hecho “GetNextLeg” que permitirá que se obtenga el siguiente tramo del plan de vuelo.

Figura 15: Aplicación que Comprueba la Finalización del Tramo

Figura 16: Diagrama para Ir al Siguiente Tramo

Gestión de conflictos

A continuación se pasará a hablar de la actividad de tareas que se ejecutan cuando se lleva a cabo una detección de posible conflicto entre dos o más aeronaves. Como muestra la Figura 17, en primer lugar los controladores aéreos encenderán los monitores que monitorizan los aviones y sus posiciones en cada momento, lo cual en nuestro sistema se traducirá en iniciar la aplicación que detectará estos posibles conflictos (“StartMonitorizeFlights”). Cuando la aplicación detecte una posible colisión le informará al controlador. Dichos conflictos serán detectados cuando no se cumpla la separación mínima entre aviones que se describió anteriormente en los requisitos. Seguidamente el controlador comprobará que este conflicto no está siendo gestionado actualmente (“IsANewConflictDetected”). Si la posible colisión ya fue gestionada entonces el controlador seguirá monitorizando los vuelos como estaba haciendo antes. Mientras que, si por el contrario, es un nuevo conflicto que tiene que empezar a gestionar, en primer lugar creará la lista de órdenes correspondientes para cada uno de los pilotos (“CreateNewOrders”) y seguidamente se iniciará una comunicación con estos pilotos para pasarles las ordenes que tienen que llevar a cabo (“Start Interaction Controller-Pilot”). Dichas órdenes normalmente se corresponden con cambios de altitud, ir a una posición determinada y/o reducir velocidad. Seguidamente el piloto, tras recibir la orden, y, con el objetivo de reaprovechar parte del desarrollo existente a la hora de pilotar los aviones, creará un nuevo tramo “ficticio” con los datos de la posición, velocidad y altura que el controlador le haya indicado. Posteriormente, y parecido a como se indicó en la Figura 12, los pilotos irán tomando diferentes decisiones (“Make Decisions”) hasta que este nuevo tramo haya finalizado por completo (“Check Leg Completed”). Cuando esto suceda, el piloto intentará ir al siguiente tramo (“GoNextLeg”), el cual coincidirá con el tramo que se estaba realizando antes de detectarse el conflicto. Una vez se haya restablecido este tramo como el actual, el piloto comunicará al controlador que la orden que este le mandó se ha llevado a cabo con éxito. Por último, el controlador marcará que acatamiento de las órdenes se ha llevado a cabo y que la gestión para evitar que este conflicto ha concluido (“End Interaction Controller-Pilot”).

Figura 17: Diagrama de Actividad Evitar Conflicto

Como se comentó previamente la tarea “IsANewConflictDetected” comprobará que el nuevo conflicto detectado (evento “PlanesInConflict”) no ha sido gestionado anteriormente (Figura 18). Para ello accederá a su estado mental “ControllerMind”, donde se encuentra la lista de todos los conflictos gestionados hasta la fecha. A parte de producir el hecho que transmite que se puede iniciar la creación de las órdenes, donde se traspasan a su vez los aviones en conflicto, esta tarea también tiene la función de iniciar una aplicación interna de comprobación de distancias entre aviones en conflicto “CheckDistanceBetweenPlanesInConflict”. Esta aplicación tendrá por objeto determinar si el riesgo de posibilidad de conflicto entre los aviones ha desaparecido antes de que estos cumplan o completen la orden transmitida por el controlador. En tal caso se procederá como en el diagrama de la Figura 22 y se marcará la orden como finalizada.

Figura 18: Comprobación de Nuevo Conflicto

Una vez el conflicto se ha detectado y confirmado como no existente anteriormente se procede a crear cada una de las órdenes que se transmitirán a cada uno de los pilotos de estos aviones en conflicto. Esto se realizará con la tarea “CreateNewOrders” (Figura 19), que encapsulará en los hechos “StartAvoidCollision” por un lado el id del piloto al que va destinada la orden (se generará un hecho de este tipo por cada avión en conflicto) y por otro lado, el nuevo tramo. Este tramo, como se ha comentado anteriormente, será un tramo adicional “ficticio” que los aviones deberán completar antes de seguir con el procedimiento normal del plan de vuelo. Se han traducido las órdenes a nuevos tramos para que los pilotos puedan así seguir, en cierto modo, la estructura de generación de decisiones e instrucciones que los pilotos llevan a cabo en el cuerpo central del sistema. Para la generación de estas nuevas órdenes se tienen en cuenta las restricciones de los controladores de las que se habló en la sección de Requisitos. Estas órdenes se limitarán, en un principio, a transmitir cambios de altura y velocidad aleatorias dentro de los valores permitidos y que eviten la colisión en caso de cruce cercano.

Figura 19: Creación de las Ordenes

A continuación se procede a explicar cómo los controladores transmitirán estas órdenes a los pilotos. Par ello primeramente, como podemos ver en Figura 20, los controladores crearán una nueva interacción cuyo colaborador será el piloto identificado dentro del hecho “StartAvoidCollision”. Por otro lado se generará un hecho que representamos como la orden que se le va a trasmitir al piloto y que contendrá el nuevo tramo generado en la tarea anterior “CreateNewsOrders”. De esta forma el piloto, en el rol de colaborador, accederá a la interacción y a la información relacionada con ella, como puede ser la orden del controlador. En esta tarea, “TakeOrder”, el piloto cogerá esta orden y extraerá el tramo que le manda el controlador para encapsularlo en un nuevo hecho “OrderNewLeg” que luego será tratado en otra tarea de otro diagrama (Figura 21), que se pasará a explicar más adelante. El piloto también se encargará de consumir el tramo “Flight Leg” que estaba realizando y almacenarlo en el hecho “OrderOldLeg” para así poder retomarlo cuando finalice de completar la orden. La siguiente tarea que se explicará, y que se encuentra en el mismo diagrama de la interacción, es “ObeyOrderCheck”. Dicha tarea se ejecutará cuando se entienda que el cumplimiento de la orden ha finalizado y por tanto se ha generado el hecho “OrderFinished”. Esto puede suceder por tres motivos:

1. Que el piloto haya completado entero el tramo solicitado por el controlador. En este caso se procederá a establecer como siguiente tramo, “Flight Leg”, el que se estaba llevando a cabo antes de que se detectara el conflicto “OrderOldLeg”. A su vez el piloto se encargará de iniciar la aplicación para que se vaya comprobando cuando se ha completado el nuevo tramo “StartLegChecker”.
2. Que el conflicto haya desaparecido, es decir, que el avión ya no esté en riesgo con ningún otro avión. El trato de este caso será como en el caso anterior, donde se restablecerá el tramo que llevado a cabo antes del conflicto.
3. Que se haya detectado un nuevo conflicto. En este caso no se generará el siguiente tramo, “Flight Leg”, puesto que este habría sido establecido por el nuevo conflicto.

Seguidamente esta tarea se encargará de comunicar al controlador que las tareas que tenía que realizar el piloto para acatar la orden han finalizado (“OrderDone”).

Como última tarea de la interacción “ObeyOrderCompleted”, el controlador se encargará de eliminar este conflicto del listado de conflictos actuales que gestiona dentro de su estado mental.

Figura 20: Interacción Controlador-Piloto traspaso de Ordenes

Como comentamos anteriormente, una de las partes importantes en la gestión de conflictos es como el piloto pasa a realizar el nuevo tramo que el controlador le ordena. Lo que el piloto hará en esta tarea (“CreateNewLegForOrder”) es tan simple como crear como tramo actual el que el controlador le mando en el hecho “OrderNewLeg” (Figura 21). A su vez el piloto tendrá que iniciar también la aplicación de finalización de tramo con el hecho “StartLegChecker”. A partir de entonces este nuevo tramo se tratara como cualquier otro tramo del plan de vuelo.



Figura 21: Piloto Obedeciendo una Orden

También es posible que el conflicto desaparezca antes de que se complete la orden que el controlador mando al piloto cuando este se detectó (Figura 22). Para detectar estos casos se inició la aplicación “CheckDistanceBetweenPlanesInConflict” de la Figura 18. Esta aplicación lanzará un evento cuando detecte que para un determinado avión ya no hay riesgo de conflicto. Dicho evento producirá que el controlador inicie una interacción con el piloto del avión para comunicarle el suceso “ConflictFinished”. Será entonces cuando el piloto creará un hecho interno para interpretar que esta situación equivale a haber finalizado la orden “OrderFinished” y consumirá el tramo que correspondía a la orden mandada por el controlador “Flight Leg”. A partir de este momento se procederá como en la Figura 20 donde se realizarán todas las tareas para la confirmación de la finalización de la tarea.

Figura 22: Conflicto Finalizado por Lejanía entre Aviones

Despliegues

Módulo de lógica

Es la capa formada por todas las clases generadas a partir del modelo desarrollado en el IDK. Aquí se incluirá la mayor parte del código autogenerado, esto quiere decir, que se incluirá tanto las clases que se basan estrictamente en la lógica del sistema (normalmente representadas por tareas) como las clases que representan entidades o agentes. Estas últimas clases se incluyen en esta capa porque el IDK al transformar agentes a código genera algo de lógica en estas clases (como puede ser que tareas se han de ejecutar cuando un objetivo se puede satisfacer). Aunque es cierto que realmente la lógica del programa está en cada una de las tareas que pueden ejecutar los agentes. La herramienta IDK permite que se pueda añadir código propio para moldear estas tareas a las necesidades del programador.

Dentro de esta capa también se incluyen otros procesos como por ejemplo los encargados de monitorizar aviones o actualizar el estado de estos en todo momento. Dichos procesos son iniciados por agentes a través de una aplicación pero, al tratarse de procesos iterativos muy largos, serán ejecutados en nuevos hilos del sistema para no interferir a la hora de los posibles objetivos que dichos agentes puedan satisfacer simultáneamente.

Visualización

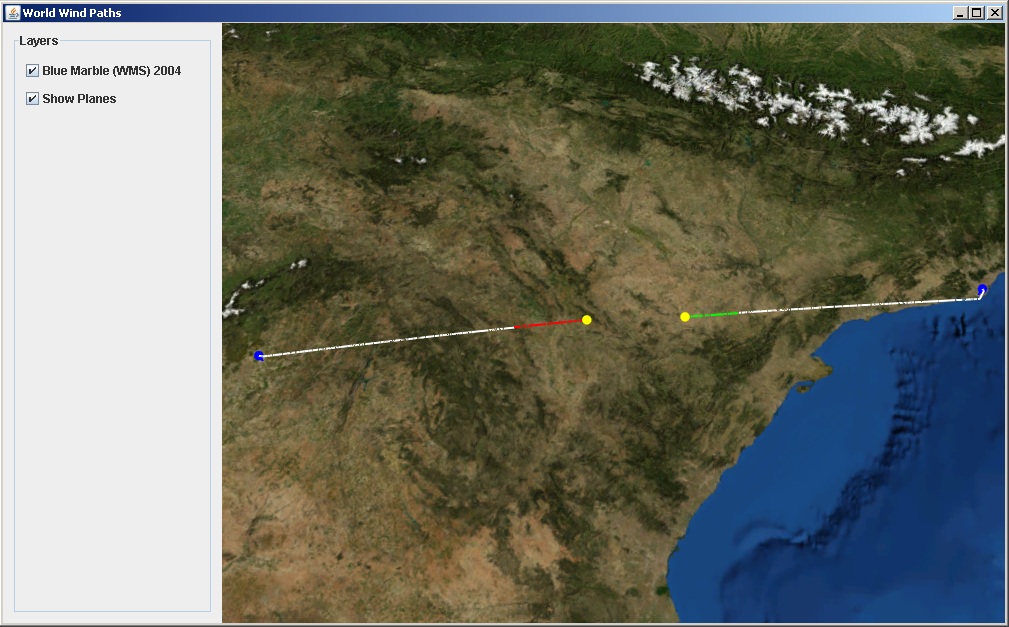
Para poder tener una representación realista de lo que se va a simular y con el objetivo de poder detectar posible errores o defectos del sistema, se han representado los datos producidos por el sistema en una serie de visualizaciones. El sistema dispone de dos posibles tipos de visualización. Por un lado está la visualización que nos aporta el IAF (INGENIAS Agent Framework), que es la que por defecto nos aporta el código autogenerado. En este tipo de visualización podremos ver los estados mentales de cada uno de los agentes, así como las interacciones y las tareas que van ejecutando en cada momento dichos agentes. Esta visualización también aporta una serie de *logs* que mostrarán un histórico de cada una de estas interacciones y tareas que se han ejecutado.

(Imágenes,: interacciones, agente…)

Por otro lado se ha implementado una visualización gráfica más realista donde se hace uso de una herramienta llamada World Wind de la NASA (51). World Wind es un programa de código abierto, desarrollado por la NASA, que actúa como un globo terráqueo virtual. Superpone imágenes de satélites de la NASA y fotografías aéreas del United States Geological Survey (USGS) sobre modelos tridimensionales de la Tierra, y en las últimas versiones, Marte y la Luna. A través de esta herramienta el usuario puede interactuar con el planeta seleccionado, rotándolo y ampliando zonas. Además se pueden superponer topónimos y fronteras, entre otros datos, a las imágenes. El programa también contiene un módulo para visualizar imágenes de otras fuentes en Internet que usen el protocolo del Open Geaospatial Consortium Web Map Service. Adicionalmente existen multitudes de ampliaciones para World Wind que aumentan su funcionalidad, como por ejemplo, poder medir distancias u obtener datos de posición desde un GPS. Algunos ejemplos de aplicaciones desarrolladas con esta herramienta son:

* *Terrain Profiler*: Aplicación que muestra la capa de perfil del terreno junto con una serie de controles. Permite visualizar en tiempo real un gráfico de la sección del perfil de cualquier parte del planeta, a cualquier escala. Resulta particularmente útil para explorar los fondos marinos, donde los datos de batimetría revelan importantes características geológicas.
* *Air Search And Rescue*: Es una aplicación que ayuda a los planificadores de búsqueda y rescate a determinar cuáles son los mejores lugares por donde buscar después de la pérdida de un avión. Usando la última posición conocida desde un radar terrestre y otros recursos, los planificadores pueden reconstruir la ruta del avión y estudiar su posición con respecto a tierra desde cualquier punto de vista, incluido el del piloto.

Haciendo uso de esta herramienta, World Wind,  se mostrarán los mapas, waypoints y aeropuertos, y como se van desplazando los aviones para llegar a su destino junto con la ruta que van recorriendo estos. Como podemos ver en la () los aeropuertos vienen marcados por puntos de color azul, los waypoints vendrán determinados por puntos en color magenta, mientras que los aviones en principio serán de color verde. Sin embargo el color de los aviones puede cambiar a ser amarillo en el caso de que los controladores establezcan que han entrado en una zona de conflicto (Ilustración 1). También es posible que este color pase a ser rojo, esto sucederá cuando el avión deje de desplazarse. Esto puede ocurrir o bien porque la aeronave ha llegado a su destino o bien porque ha colisionado con otro avión. Por otro lado, algo similar pasará con la ruta, cuyo color inicial será blanco, lo que indicará que el avión está volando a la altura establecida como altura estándar de crucero para esa aeronave. Cuando el avión empiece a volar por encima de esta altura de crucero el color de la ruta pasará a ser rojo, mientras que cuando vuele por debajo este color será verde.

Ilustración 1

Para poder visualizar cada una de las clases de la capa de la lógica se ha creado una capa de visualización, que adapta las entidades del sistema (aviones, rutas, aeropuertos, etc) a clases de la herramienta World Wind para que estas puedan ser gráficamente mostradas en los distintos mapas.

En cuanto al diseño de la visualización gráfica viene determinada por un agente (Simulation Creator) que se encarga de arrancar el entorno para mostrar la simulación. Este entorno esta implementado por una aplicación de INGENIAS que, dentro de otro hilo de ejecución, cargará las distintas capas y paneles, donde estarán incluidos los mapas, los puntos sobre los mapas y los controles que permiten personalizar la visualización.

Controlador

Esta capa estará encargada de sincronizará el estado de las entidades de la lógica con el representado por la visualización, para que en todo momento tengamos una situación actualizada de las posiciones de los aviones. También gestionará que al iniciar la visualización se muestren los aeropuertos y waypoints por los que los aviones pasarán. En resumen tendrán una lista de los elementos que visualizan cada uno de las entidades (aviones, rutas,…) y siempre que desde la lógica o modelo se actualice el estado de alguno de estos elementos este controlador lo transmitirá a cada uno de estos elementos de visualización. Para que esto sea posible además de estos elementos de visualización el controlador tendrá acceso a la capa con los mapas, para poder refrescar estos gráficos y poder ver en todo momento el estado más actual de la simulación.

Casos de estudio

Caso básico dos aviones (conflicto, funcionamiento básico,…)

Como primer caso de estudio se observará el comportamiento de dos aeronaves que se encuentran en un punto volando en sentidos contrarios. Dicho caso servirá para tener un ejemplo básico del comportamiento del sistema ante los conflictos. Para ellos diseñaremos un despliegue (Figura 23) en el habrá dos pilotos, dos aviones, un planificador de vuelos (que suministra el plan de vuelos a los pilotos), un controlador aéreo que gestiona los posibles conflictos, y un creador de la simulación que se encarga de iniciar la aplicación que muestra la simulación.



Figura 23: Despliegue dos aviones

Para evitar complejidad se limitará el número de aeropuertos a dos y los planes de vuelo no tendrán planificados ningún waypoint por los que tengan que pasar los aviones durante el trayecto. De esta forma cada aeronave tendrá como aeropuerto de origen y de destino el contrario al del otro avión. Así, aproximadamente en el centro de la trayectoria de ambas aeronaves, el controlador detectará que los aviones entran en zona de riesgo de conflicto y mandará operaciones de cambio de altura y velocidad a ambas aeronaves. Esto puede observarse porque las aeronaves pasarán a tener un color amarillo y sus rutas tomaran el color verde o rojo dependiendo si la variación de su altura supone un descenso o aumento con respecto a la altura de crucero. Una vez pasado el riesgo de conflicto, y acorde al diseño del sistema, los aviones volverán a sus respectivas alturas y velocidades de crucero.

Caso más complejo: muchos aviones, diferentes comportamientos de pilotos, UAVs….

Discusión

Discusión de los casos de uso. Como se habría abordado el estudio con otras herramientas, ….

Conclusión y trabajo futuro

Similitud con la introducción

Más general que la discusión

Agentes

Basado en modelos

Separación con el modelaje del avión

Separación Pilotos/Sistemas

Comparativas de simulación a nivel general

Ventajas:

flexibilidad a la hora de introducir tiempos de ejecución de maniobras

Facilidad a la hora de hacer/ofrecer la simulación

Trabajos futuros y mejoras

# Bibliografía

1. *A vision for European Aviation.* **Eurocontrol.** 2005. Eurocontrol and ACI Europe Press Conference.

2. *Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems.* **Claire Tomlin, George J. Pappas, and Shankar Sastry.** April de 1998, IEEE Transactions on Automatic Control, Vol. 43, págs. 509--521. 4.

3. **JSBSim.** JSBSim flight dynamics model. [En línea] http://jsbsim.sourceforge.net/ .

4. **FlightGear Flight Simulator.** FlightGear Flight Simulator. [En línea] http://www.flightgear.org/.

5. **OpenEaagles.** OpenEaagles. [En línea] http://www.openeaagles.org/.

6. **AVIATOR VISUAL DESIGN SIMULATOR.** *Manual, AVIATOR VISUAL DESIGN SIMULATOR (AVDS) User.* http://www.rassimtech.com/documentation/AVDSManual.pdf.

7. *FACET: Future ATM Concepts Evaluation Tool.* **Bilimoria, K.** Napoli, Italy : s.n., 2000. 3rd USA/Europe ATM 2001 R&D Seminar.

8. *The design of FACET to support use by airline operations centers.* **Smith, P.** 2004 : s.n., IEEE.

9. *A Multiagent Simulation of Collaborative Air Traffic Flow Management.* **Shawn R. Wolfe, Peter A. Jarvis, Francis Y. Enomoto, Maarten Sierhuis,Bart-Jan van Putte.** [ed.] Franziska Klügl Ana L. C. Bazzan. s.l. : Information Science Reference, 2009, Multi-Agent Systems for Traffic and Transportation Engineering, págs. 357-381. NASA Ames Research Center.

10. *Human performance models of pilot behaviour.* **Foyle, D. C., Hooey, B. L., Byrne, M. D., Corker, K. M., Deutsch, S., Lebiere, C.** Santa Monica, CA : s.n., 2005. The Human Factors and Ergonomics Society 49th Annual Meeting. págs. 1109-1113.

11. **Frankel, David.** *Model Driven Architecture: Applying MDA to Enterprise Computing.* s.l. : John Wiley & Sons, 2003.

12. **Gilbert, N., Troitzsch, K. G.** Simulation for the Social Scientist. *Open University Press.* 2005.

13. *INGENIAS development kit: a visual multi-agent system development environment.* **Jorge J. Gómez-Sanz, Rubén Fuentes, Juan Pavón, Iván García-Magariño.** 2008. AAMAS (Demos)'2008. págs. 1675-1676.

14. **Helmreich RL, Merritt AC, Wilhelm JA.** The evolution of crew resource management training in commercial aviation. *Int J Aviation Psychology.* 1999, Vol. 9, págs. 19–32.

15. **Earl L. Wiener, David C. Nagel.** *Human Factors in Aviation.* s.l. : Academic Press, Inc, 1988.

16. *Resource Management on the Flightdeck: Proceedings of a NASA/Industry Workshop.* **Cooper, G. E., White, M. D., & Lauber, J. K.** Moffett Field, CA : NASAAmes Research Center, 1980.

17. *Crew resource management: Achieving enhanced flight operations.* **Taggart, W. R.** Aldershot, UK : Aviation Psychology in Practice, 1994. Avebury Technical. págs. 309-339.

18. **FAA.** *Crew resource management training.* Departament of Transportation, FAA. Washington, DC : Advisory Circular No AC 123-51B, 1995. págs. 10-12, Appendix 3, 1-2.

19. **Mellor, A.** Design, development and implementation of a CRM program. *Aviation instruction and Training.* 1993, págs. 368-384.

20. **T., Kern.** Redefining airmanship. 1997.

21. *The use of personal computer-based aviation training devices to teach aircrew decision-making, teamwork, and resource management.* **Duncan, J.C. and Feterle, L.C.** Dayton, OH : s.n., 2000. Proceedings of IEEE 2000 National Aerospace and Electronics Conference. págs. 421–426.

22. **Lauber J.K, Foushee.** Guidelines for the development of line oriented flight training. *NASA Conference Publication 2184.* 1981.

23. *Pilot Personality and Crew Coordination: Implications for Training and Selection.* **Helmreich, Robert L., Chidester, Thomas R. y Gregorich, Steven E.** 1, 1991, The International Journal of Aviation Psychology, Vol. 1, págs. 25-44.

24. **Ehlert, P.A.M., Rothkrantz, L.J.M.** *The Intelligent Cockpit Environment (ICE) Project.* Data and Knowledge Systems Group, Delft University of Technology, The Netherlands. 2003.

25. *Aerobot airdata measurement for planetary exploration.* **Geneste, E., Barnes, D.P.** Manchester, UK : s.n., Ablir, 2001. TIMR 01 - Towards Intelligent Mobile Robots, 3rd British Conference on Autonomous Mobile Robots and Autonomous Systems.

26. Simulink. [En línea] http://www.mathworks.com/products/simulink/.

27. *An Agent Based Framework for Modeling UAVs.* **N. Huff, A. Kamel, and K. Nygard.** 2003. The 16th International Conference on Computer Applications in Industry and Engineering (CAINE03).

28. *Fast-Time Simulation System for Analysis of Advanced Air Transportation Concepts.* **Sweet, D. N., Manikonda, V., Aronson, J. S., Roth, K., & Blake, M.** Monterey, California : s.n., 2002. American Institute of Aeronautics and Astronautics (AIAA) Modeling and Simulation Technologies Conference and Exhibit.

29. *Agent architectures for flexible, practical teamwork.* **Tambe, M.** Providence, Rhode Island : s.n., July, 1997. American Association for Artificial Intelligence Conference (AAAI-2007).

30. *Distributed agent-based air traffic flow management.* **Adrian K. Agogino, Kagan Tumer.** Honolulu, Hawaii : s.n., May, 2007. Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems.

31. **Flight Explorer Inc.** Flight Explorer. [En línea] http://www.flightexplorer.com/.

32. *Modeling Distributed Human Decision-Making in Traffic Flow Management Operations.* **Keith C, C.** Napoli, Italy : s.n., 2000. 3rd USA/Europe Air Traffic Management R&D Seminar.

33. *AGENTFLY: A multi-agent airspace test-bed.* **D. Sislak, P. Volf, S. Kopriva, and M. Pechoucek.** May, 2008. 7th Intl. Conf. on Autonomous Agents and MultiAgent Systems.

34. *To bdi or not to bdi.* **S. Wolfe, M. Sierhuis, and P. Jarvis.** 2008. Design choices in an agent-based Traffic Flow Management Simulation Multiconference.

35. *BDI-agents: From Theory to Practice.* **Georgeff, A. S. Rao and M. P.** San Francisco : s.n., 1995. Proceedings of the First International Conference on Multiagent Systems.

36. **EUROCONTROL.** *EUROCONTROL SPECIFICATIONS FOR THE USE OF MILITARY UNMANNED AERIAL VEHICLES AS OPERATIONAL AIR TRAFFIC OUTSIDE SEGREGATED AIRSPACE.* 2007. http://www.barnardmicrosystems.com/download/EUROCONTROL\_MIL\_UAV\_ATM\_SPEC\_2007.pdf.

37. *Improving air traffic management through agent suggestions.* **Adrian K. Agogino, Kagan Tumer.** 2009. AAMAS (2) 2009. págs. 1271-1272.

38. **IABG Dept. Airborne Air Defence.** *CARE Innovative Action, Preliminary Study on Integration of Unmanned Aerial Vehicles into Future Air Traffic Management.* 2001.

39. *Autonomous UAV Surveillance in Complex Urban Environments.* **Eduard Semsch, Michal Jakob, Dusan Pavlícek, Michal Pechoucek.** 2009. IAT 2009. págs. 82-85.

40. *Generic Nonlinear model of reduced scale UAV.* **Cheviron, T., Chriette, A., & Plestan, F.** Kobe, Japan : s.n., 2009. IEEE International Conference on Robotics & Automation. págs. 3271-3276.

41. Dimensions & key data for A-320. [En línea] http://www.airbus.com/aircraftfamilies/passengeraircraft/a320family/a320/specifications/#details.

42. **Airbus.** *A320 AIRPLANE CHARACTERISTICS FOR AIRPORT PLANNING.* 2005.

43. Directory of Waypoints in Spain. [En línea] http://www.fallingrain.com/world/SP/waypoints.html.

44. Flota de Iberia en el Grupo Iberia. [En línea] http://grupo.iberia.es/portal/site/grupoiberia/menuitem.8e32b03d06b292dda0d4a195d21061ca/?id\_avion=57d89a1b25862010VgnVCM100000950216ac\_\_\_\_&origen=datos&vengode=Iberia.

45. **Federal Aviation Administration.** *Air Traffic Control FAA Order 7110.65T.* Washington, DC. : s.n., Feb. 11, 2010.

46. *The INGENIAS Methodology and Tools.* **Pavón, J., Gómez-Sanz, J. J., y Fuentes, R.** [ed.] B., y Giorgini, P. Henderson-Sellers. 2005. Idea Group Publishing. págs. 236-276.

47. *The Ingenias Project: Methods and Tool For Developing Multiagent Systems.* **Botia, J. A., Gonzalez, J. C., Gomez, J., y Pavon, J.** 2008, Latin America Transactions, IEEE (Revista IEEE America Latina), págs. 529-534.

48. *Modelling a Knowledge Management System Architecture with INGENIAS Methodology.* **Soto, J. P., Vizcaino, A., Portillo, J., y Piattini, M.** Washington, DC : s.n., 2006. En Proceedings of the 15th international Conference on Computing. págs. 167-173.

49. *Model Driven Development of Multi-Agent Systems.* **Pavón, J., Gómez-Sanz, J.J., Fuentes, R.** [ed.] A., Warmer, J. Rensink. Springer, Heidelberg : s.n., 2006. ECMDA-FA 2006. Vol. 4066, págs. 284- 298.

50. **Reenskau, T. M. H.** MVC XEROX PARC 1978-79. [En línea] http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

51. World Wind Java SDK. [En línea] http://worldwind.arc.nasa.gov/java/.

52. **Daniel J. Garland, John A. Wise and V. David Hopkin.** *Handbook of Aviation Human Factors.* [ed.] Lawrence Erlbaum Associates. 1999. ISBN 0-8058-1680-1.

Glosario

ABMS Agent-Based Modeling and Simulation

ACES Airspace Concept Evaluation System

AOCC Airport Operations Control Center

ARTCC Air Route Traffic Control Center

ATM Air Traffic Management

ATFM Air Traffic Flow Management

ATS Air Traffic Service

AVDS Aviator Visual Design Simulator

BDI Belief-Desire-Intention software mode

CATFM Collaborative Air Traffic Flow Management

Eurocontrol European Organisation for the Safety of Air Navigation

CRM Crew Resource Management

FAA Federal Aviation Administration of USA

FACET Future ATM Concepts Evaluation Tool

GDP Ground Delay Program

GPL GNU Public License

HF Human Factors

IAF INGENIAS Agent Framework

IDK INGENIAS Development Kit

IMPACT Intelligent agent-based Model for Policy Analysis of Collaborative Traffic flow management

LOFT Line Oriented Flight Training

MAS Multi-Agent System

MDA Model-Driven Arquitecture

MDE Model-Driven Engineering

MIDAS The Man-Machine Integrated Design and Analysis System

NAS National Aispace System

OMG Object Management Group

STEAM Shell for TEAMwork

UAV Unmanned aerial vehicle

VFR Visual Flight Rules

IFR Instrumental Flight Rules

###### Enter Your Appendix Title Here

Appendices must be identified by letters (A, B, etc.) rather than by numbers. For this reason, different style headings are used with appendices. (The style at the top of this page is “Appendix A - Heading 6.”)

First-level Subhead (Heading 7 style)

Within an appendix, Heading 7 is the style to use for all first-level subheads. If you need to add another subhead level within Heading 7, use Heading 8 as shown below.

Second-level Subhead (Heading 8 style)

Use Heading 8 for all second-level subheads within an appendix. If you need to add another subhead level within Heading 8, use Heading 9 as shown below.

Third-level Subhead (Heading 9 style)

If you need a third-level subhead in an appendix, use Heading 4.

Figures and Tables Within Appendices

When you first add a figure or table to an appendix, it will be numbered as though it were in a regular chapter. For example, when the figure below was first inserted, it became “Figure 4.1”. As a figure within Appendix A, it should be “Figure A.1”

To make this change, the codes in the caption labels must be modified, and it’s best to wait until all figures and tables have been added to appendices.

Figure A.1 First Figure in Appendix A

###### Enter Your Appendix Title Here

If you need additional appendices, use style “Appendix A – Heading 6” for the appendix heading. This will label appendices in alphabetical order (A, B, C, etc.).