Certification of a Civil UAS: A Virtual Engineering Approach

Neil Cameron[1] and Matt Webster[2]
Virtual Engineering Centre - University of Liverpool, Warrington, Cheshire, UK, WA4 4AD

Mike Jump[3] and Michael Fisher[4]
University of Liverpool, City, Liverpool,Merseyside, UK, L69 3GH

The use of Unmanned Autonomous Systems (UAS) is becoming an increasingly routine activity in military theatres of operation, particularly for the oft-cited 'dull, dangerous and dirty' missions. There is growing acceptance that UAS will find similar utility within the corresponding civilian missions and beyond. UAS technologies are maturing rapidly but the associated regulations to allow open access to civilian airspace are yet to be fully formulated. Current UK practice is therefore to allow UAS operation only in segregated airspace (airspace denied to all other potential users) or in non-segregated airspace but restricted to line-of-sight operations, below 400ft only. There is therefore a growing need to develop a means by which UAS can operate alongside existing airspace users, in all classes of non-segregated UK airspace. The University of Liverpool's Virtual engineering Centre, is developing tools and techniques that will allow both industry and regulators to establish a 'design for certification' ethos within the supply chain where safety-critical software and hardware is required. The processes will include requirements capture and validation phases, as well as a means of testing and evaluating whole UAS/sub-system virtual prototypes, with a view to being able to demonstrate compliance with the relevant airworthiness codes as early as possible in the design cycle.

Nomenclature

r	=	turn rate (deg/s)
ϕ	=	roll angle (deg)
$\Delta\psi$	=	change in heading (deg)
ψ	=	aircraft heading (deg)

I. Introduction

IN order to be able to operate an aircraft type, the manufacturer of that type must establish, to the satisfaction of the relevant regulatory bodies, that the vehicle conforms to the applicable airworthiness code (for example, for Large Transport aircraft in Europe, Ref. 1 applies). At the end of this process, the prototype aircraft is awarded a Type Certificate and hence the process is known generally as 'certification'. For manned aircraft, there is a well understood route, developed over the last 100+ years of maned flight, for manufacturers to demonstrate such compliance. One aspect of the airworthiness regulatory environment is to show that the probability of a failure or combination of failures, which could cause a significant hazard, is acceptably low. However, the manufacturer does not have to concern itself with certification of the pilot: it is assumed that a suitably qualified crew will operate the aircraft. Even though automatic systems for civil aircraft are becoming ever more capable and demonstrate increasing reliability, at least one pilot is still on board and is tasked with remaining situationally aware during the flight. Flights by Unmanned Autonomous Systems (UAS) on the other hand, are conducted without a pilot on board the aircraft, the human in the loop remaining only as a ground-based operator. The resulting remoteness of the

[1] Post Doctoral Research Associate, VEC - University of Liverpool, ncameron@liverpool.ac.uk
[2] Post Doctoral Research Associate, VEC - University of Liverpool, matt@liverpool.ac.uk
[3] Lecturer, School of Engineering, mjump1@liverpool.ac.uk
[4] Professor, Department of Computer Science, mfisher@liverpool.ac.uk

operator potentially deprives him/her of the sensory stimuli necessary to maintain situational awareness and deprives the UAS of a complex and adaptable sensor suite (the human senses). In addition, the command and control of the vehicle is reliant upon a data link between the UAS and its operator(s). In the event that this data link is lost, even temporarily, to maintain a safe flight, the UAS will have to be able to make the same rational decisions as an onboard human pilot would make in a given situation to maintain the same level of safety. Throughout this paper, this rational autonomous decision-making component of the vehicle will be called the Complex Flight Control System (CFCS). It is therefore clear, that the addition of a CFCS will add an additional element to the certification process. Evidence will have to be provided that, in lieu of an on-board human pilot, the CFCS plus the systems directly associated with it (e.g. power supplies[2]) will not compromise the safety of the aircraft or other airspace users to a level beyond that provided by a manned aircraft.

An acceptable set of airwirthiness regulations that will allow UAS to be incorporated into managed airspace on a routine basis has yet to be formalised by the regulatory authorities. However, guidance on how it might eventually be achieved in the United Kingdom (UK) is provided by the UK CAA in Ref. 3. The over-arching principle within Ref. 3 is that '*UAS operating in the UK must meet at least the same safety and operational standards as manned aircraft*'. In practice, this means that at present, UAS will typically fly in segregated UK airspace (airspace that is denied to other users), reducing the flexibility, efficiency and desirability of their operation. For UAS to be incorporated into non-segregated UK airspace in a routine manner, Ref. 3 specifies a number of requirements for UAS operations:

- 'Safety': The UAS must be no less safe than manned aircraft, as noted above.
- 'Equivalence': The UAS must be able to comply with existing aviation rules and regulation
- 'Transparency': The UAS must not require any special or additional services to allow it to operate without any adverse effects on other airspace users or to communicate and comply with existing air traffic control infrastructure

The UK ASTRAEA 2 project[4] is now working, in conjunction with the regulatory authorities, towards demonstrating the technologies that will be required to achieve both the guidelines and ultimately, the certification requirements for UAS. The likely certification solution for UAS will be that the elements of the aircraft that would form part of a piloted aircraft will be certified under the existing regulations (e.g. Ref. 1) whilst those elements of the aircraft designed to replace the pilot would be certified under, as-yet unpublished, UAS-specific regulations. Until such regulations exist, however, the UAS community finds itself with something of a paradox. Manufacturers require a set of certification standards against which they can design a UAS, whilst regulators would ideally like to be able to appraise an already extant prototype UAS. Virtual Engineering (VE), which, for the purposes of this paper is defined as '*the integration of product modelling with process modelling*', provides a potential means to help solve this impasse.

The use of VE to model the integration of civil UAS into non-segregated UK airspace is the subject of a case study at the University of Liverpool's Virtual Engineering Centre (VEC)[5]. One part of the case study is aimed at assisting the development of a certification process by investigating techniques that will help to demonstrate the equivalence of a CFCS to a human pilot i.e. show that the decisions made by the CFCS of a UAS are consistent with those that would be made by a human pilot. The initial focus here is compliance with the Rules of the Air laid out in the UK Air Navigation Order[6]. This effort will result in the development of processes, tools and techniques that will allow both industry and regulators to establish a 'design for certification' ethos where safety-critical software and hardware is required within the Product Life Cycle (PLC). The processes will include requirements capture and validation phases, as well as a means of testing and evaluating whole UAS/sub-system virtual prototypes. The aim will be to demonstrate compliance with the relevant UAS airworthiness codes (i.e. safety, transparency and equivalence requirements) as early in the design cycle as possible, through the use of VE.

The remainder of the paper reports on the progress made to date to achieve the aim of the civil UAS case study. Section II briefly introduces the VEC project as a whole. Section III reports on the specifics of the case study in more detail. Section IV reports on the real-time networked simulation facilities developed at the VEC to date and Section V reports on how the facilities are being used to develop processes, tools and techniques to provide UAS certification evidence to the regulatory authorities. Finally, Section VI concludes the paper.

II. The Virtual Engineering Centre

The Virtual Engineering Centre (VEC)[5] is a University of Liverpool (UoL) initiative, in partnership with the Science and Technology Facilities Council (STFC) Daresbury Laboratory, North West Aerospace Alliance (NWAA), BAE Systems (BAES), Morson Projects and Airbus UK. The main objective of the VEC is to provide a centre of excellence hub in VE in the North West (NW) of England. The intention is that by developing integrated VE processes, tools and techniques across the Product Life Cycle (PLC), illustrated in Figure 1, the overall business performance of the aerospace sector in the NW of England and beyond will be significantly improved.

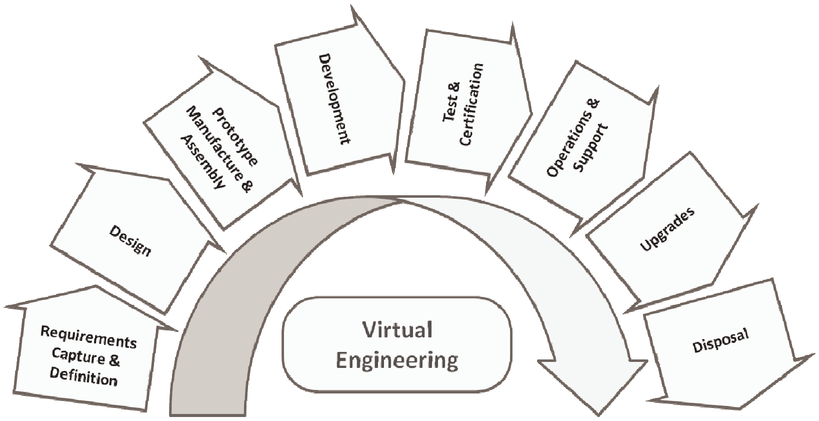

Figure 1. The Product Life Cycle (PLC, Ref. 7)

The NW of England has a high concentration of aerospace businesses serving both civil and military customers across the world and, despite the recent economic downturn, the long-term business prospects for the aerospace sector are very encouraging and offer these businesses excellent opportunities for growth[7]. However, aerospace product development is an increasingly more complex and globalised activity involving a world-wide supply chain. If the challenging performance goals for all new aircraft platforms and the engineering systems on which they are built are to be met, it is imperative that developers adopt effective system engineering processes to create the innovative solutions expected. In an effort to assist the North West Aerospace industry in rising to these challenges, the VEC aims to:

- Provide integrated product/process models, virtual prototyping capabilities and facilities for the benefit of industrial organisations of all sizes throughout the supply chain to design and rapidly evaluate new products, production facilities or services in virtual form;
- Provide a VE research focus through the creation of multidisciplinary teams working collaboratively and concurrently across industry and academia. This multidisciplinary approach will push the boundaries of existing capabilities resulting in high fidelity simulation for scenarios not currently possible;
- Create demonstrations and case studies from real-world product and process model data that demonstrate the business benefits of VE to the aerospace supply chain

To develop the state of the art tools, techniques and processes necessary to demonstrate VE practice across the PLC, the VEC project has been configured to contain a set of Work Packages (WP) that cover the relevant phases of the PLC, as illustrated in Figure 2. The technical work packages (WP2, WP3 and WP4) contain the tasks that will develop state of the art VE practice for use in the relevant phases of the PLC. WP5 will develop a framework for verification and validation (V&V) of VE and the associated Virtual Prototypes (VPs), drawing upon results from additional funded activities at the University of Liverpool. WP6 is a skills development work package to provide both industry and academia with graduates that are knowledgeable about VE practices with WP7 providing a business development and knowledge exchange element to the project. Finally, WP1 will establish an integrated product and process modelling framework, drawing on

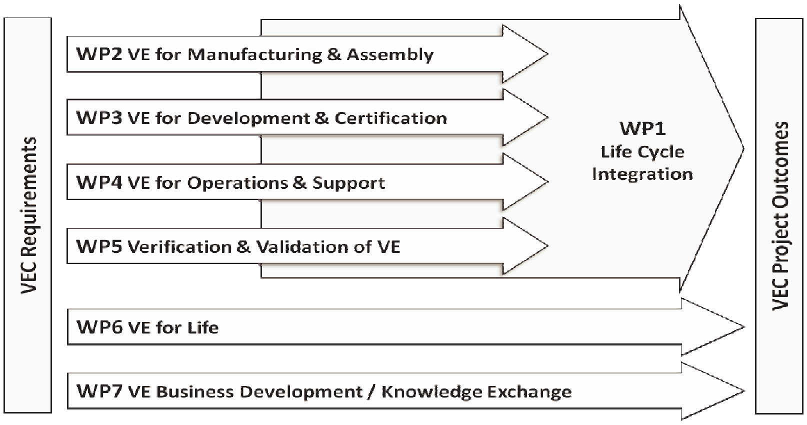
VEC Requirements
WP2 VE for Manufacturing & Assembly
WP3 VE for Development & Certification
WP4 VE for Operations & Support
WP5 Verification & Validation of VE
WP1 Life Cycle Integration
WP6 VE for Life
WP7 VE Business Development / Knowledge Exchange
VEC Project Outcomes

Figure 2. VEC Project Delivery Structure

the results of each of the technical WPs, to support VE practice at all levels of skills and competencies throughout the PLC.

III. VEC WP3 Development & Certification Case Study

For each VEC WP, the VE tools, processes and techniques relevant to its theme are being developed and exercised with reference to a WP-specific case study. For WP3, of particular interest are the issues surrounding the testing and certification of UAS into non-segregated UK airspace. These form the basis for the WP3 cases study, which will be described in more detail in this Section.

Currenly UAS operations in the UK are restricted to operations which are segregated from other air traffic in Class G airspace (no Air Traffic Control services or information on seperation from other aircraft are provided). Therefore, if UAS are to operate alongside existing air traffic in controlled airspace (Class A - E), they must meet certification requirements pertaining to the airframe and the CFCS. The VEC WP3 case study is considering what processes, tools and techniques are required to provide certification evidence for a UAS CFCS/sub-system within a UAS, to navigate autonomously between two airfields across a range of classes of UK airspace. Figure A1 in Appendix A shows a schematic of the UK's controlled airspace. For the case study, the UAS will be tasked with navigating from Aberporth aerodrome in south west Wales, to Sumburgh in the Shetland Islands, Scotland. Enroute, the UAS will pass through some uncontrolled airspace (Class G), but the majority of the flight will be along air corridors (Class A and D airspace) marked in blue in the Figure.

Before embarking on the flight, Air Traffic Control must be provided with a flight plan showing the intended route to destination, filed as a series of navigation reporting points that will be passed enroute. Each blue dot in Figure A1 represents one such possible reporting point that can be selected to form part of the route. However, large tracts of airspace, marked in pink and red in the Figure, are reserved for military or other purposes and as such, should not be entered. The flight plan must therefore take these, and any other temporary airspace restrictions, into account. Beyond this, the UAS and its operator(s) must be able to continually assess and provide contingency plans in the event of an unplanned event, such as a system failure, the identification that another airspace user is on a collision course or weather conditions that are worse than forecast, requiring, for example, a diversion or a return to base. These scenarios and their variants are to be included in the case study.

The UAS that will make this flight can be considered to consist of two components:

- The platform - aircraft structure and systems that would exist on the aircraft whether it is manned or unmanned. These elements will not feature to any great degree in this case study
- The 'Complex Flight Control System' (CFCS) - hardware and software which must replicate the functions of the pilot

During the flight, the CFCS must be able to comply with the Rules of the Air as set out in the Air Navigation Order[6], as well as meeting the high level functional requirements listed in CAP722[3], such as having an 'equivalent level of safety' as a manned aircraft. The approach adopted to ensure that the CFCS meets the 'equivalent level of safety' as a human pilot is as follows. The actions/reactions of a human pilot before, during and after a particular flight will be mapped and a representative CFCS will be developed. If it can then be shown that the CFCS behaves in the same manner as a human pilot is expected to, then this would be a step towards demonstrating human equivalence. Of itself, this task is too large for the VEC to tackle within the project timescales and with the resources available to it. The development of this validation process at the VEC has therefore taken the form of a 'proof of concept'. Components of the CFCS that relate to the sense and avoid algorithm have been selected. The intention is then to capture the engineering requirements for these components with reference to the functions of a human pilot, then demonstrate how and to what level the component can meet them. In this way, methods for generating, collecting and analysing evidence that the relevant certification authorities will consider acceptable to demonstrate that the safety of other airspace users is not compromised by the presence and/or proximity of a UAS will be developed.

To enable the proof of concept process, the VEC project has constructed a new VE networked simulation facility with the assistance of one the project's industrial partners. Within this facility, the Virtual Engineering Simulation Laboratory (VESL), a virtual prototype of a UAS can be inserted into a virtual environment. This has required the creation of both online (real-time) and offline (non real-time) simulation models of the sense and avoid components, the UAS and the environment (atmosphere, air traffic control, conflicting air traffic and a UAS operator ground station). Beyond the proof of concept itself, the capabilities of this facility will allow:

1. UAS developers and their suppliers to capture design and certification requirements as early as possible and in a collaborative manner;
2. System suppliers access to development tools that would otherwise be unaffordable and
3. UAS developers to manage the risks associated with the certification process when integrating third-party software and hardware into a UAS.

With the completion of the construction of the basic elements of the VESL, a methodology is being developed to provide evidence of how a CFCS meets/does not meet the relevant certification requirements. Two methods currently being implemented (and are discussed in more detail in Section V) are:

- How the decision making processes within the CFCS can impact upon the frequency of an identified risk occurring. Agent-based programming and the associated formal methods[8] (in the form of an "intelligent" agent) are being used to examine this issue and
- How sensor models with various resolutions will impact upon the UAS decision making process.

IV. VESL Facilities

The Flight Science and Technology Research Group (FS&T) at the University of Liverpool, now part of the Centre for Engineering Dynamics, has developed over the last decade, a world-class motion flight simulation research facility[9-10]. Recent research has expanded the virtual engineering capabilities in the area of distributed networked simulations[11-12] and the knowledge gained in these projects forms the cornerstone of the development of the VEC's new Virtual Engineering Simulation Laboratory (VESL).

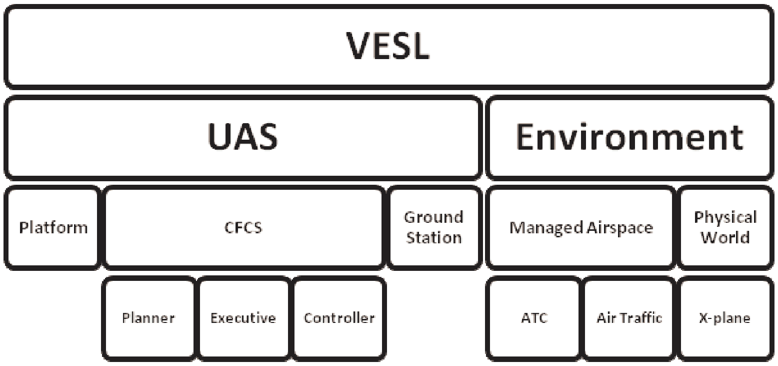

Figure 3. Hierarchical structure of the VEC VESL

The architecture of the real-time networked simulation facilities that have been developed in conjunction with one of the project's industrial partners, is illustrated in Figure 3. This currently consists of a UAS and it's associated subsystem(s) such as the decision making "*agent*" and ground control station. An agent is defined in Ref. 13 as "*a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives*". The UAS is immersed in a simulation environment which has a geodetic coordinate system and real world elevation data. The UAS shares the managed UK airspace with other air traffic, which is generated and regulated via an Air Traffic Control (ATC) simulation. The disparate elements of the UAS and the synthetic environment are connected via a central software hub. In addition to the hub being a core element linking the VESL components, the hub has two further applications. The first is that it can link with external sites such as the flight simulation facilities at the University of Liverpool (UoL)[10], so that if required, several real-time piloted simulation, high-fidelity vehicle models can be injected into the airspace local to the UAS to assess, for example, the performance of its decision making algorithms. The second is that all entity information in the simulation is sent to the hub, from where it is broadcast to a visualisation tool, where all players in a simulation (UAS, air traffic and external entities etc.) are displayed. Each of the lower level simulation components are discussed in more detail in the remainder of this Section.

A. UAS

The UAS in its current form consists of a simulation model of the vehicle platform, a traditional flight control system consisting primarily of automatic pilot functions and an agent-based CFCS. A ground control station (GCS) is currently under development to add to this suite of tools. More details of those elements that are complete are described below.

1. Vehicle Platform and BasicControllers

For this project, an existing FLIGHTLAB[14] model of a light aircraft, (Class 1 aircraft performing Category B/C manoeuvres[15]) developed at the University of Liverpool, has been used as the basis for the unmanned air vehicle. The UAS airframe with no augmentation, as illustrated in Figure 4, is a stable platform. The real part of the oscillatory mode is negative (spiral and roll modes are also stable for this platform). Therefore, no additional stability augmentation system has been added to the aircraft. The bare airframe is not discussed in detail here as it is not the primary focus of this paper.

In order that the CFCS can guide and navigate the bare airframe, a series of autopilot functions have been added to the vehicle flight control system. The UAS can accurately navigate to any waypoint on the ground or in the air using a combination of the simple heading, altitude and speed controllers that have been implemented.

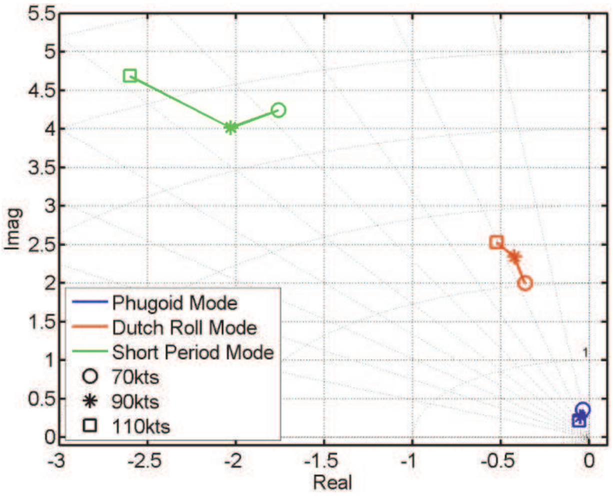

Figure 4. UAS platform stability analysis

If the heading controller is considered first, the UAS's current position is compared to the desired position (i.e. the next waypoint's latitude and longitude) and a heading calculated to reach the desired waypoint. The commanded heading is then compared with the current heading and the error output, $\Delta\psi$, passed to a limiter. The limiter is in place because gross heading-change manoeuvring of the aircraft is achieved by rolling the aircraft about its longitudinal axis. The secondary effect of this is to induce a turn rate. The output of the controller will therefore be applied to the aileron actuator to control roll rate. If a large turn is required, comparing $\Delta\psi$ to the aircraft bank angle ϕ will result in the aircraft potentially performing an aggressive turn or attempting to roll beyond its design limits. The range of allowable $\Delta\psi$ values was selected such that the bank angle results in a desired turn rate, in this case selected to be a Rate 1 turn (3°/s). Figure 5 shows the response of the system to a 90° commanded heading change. The aircraft responds to the step input command with an increase in turn rate that does not quite reach the desired Rate 1 level (indicating that some further controller tuning is required) but achieves a smooth and linear heading change nonetheless. Should a more aggressive rate of turn be required, the $\Delta\psi$ limiter can be adjusted to provide it. This heading controller is of course only applicable when the aircraft is in the air. When on the ground, $\Delta\psi$ is passed to the nose wheel steering system. This is used to provide a taxi to/from the runway to/from the airfield apron. It is anticipated that as the project progresses, more sophisticated controllers will be implemented. A heading-acquire controller will be of less use than a planned ground track follower, for example, when environmental variables such as wind are introduced.

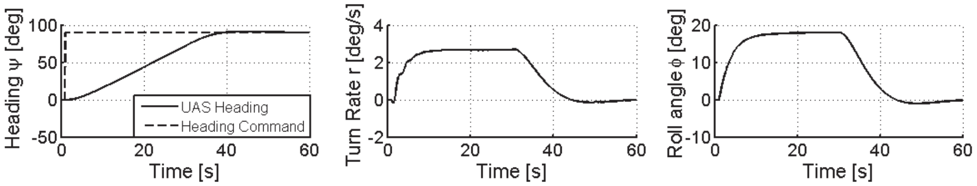

Figure 5. Heading capture controller example – response to a 90° heading change step input

The speed-acquire and altitude-acquire controllers work on the same error-based principles as the heading controller. However, these controllers perform different tasks depending upon the flight phase in question. When on the ground ('ground mode'), the throttle is used to control speed and the altitude controller is switched off. During take-off, the altitude controller is automatically switched on and the aircraft rotates by applying a step input in elevator. When in the the climb, the throttle is set to maximum power and the elevator used to trim the UAS to a 70 knots climb speed, resulting in a climb rate of approximately 700ft/min. In the cruise, the throttle is used to control speed, the elevator being used to control the aircraft's altitude. For the descent phase of flight, the system then switches to use elevator to control speed, and throttle to control descent rate. Once landed, the controllers revert back to 'ground mode' where for landing, throttle is removed and brakes applied to slow the aircraft.

2. *CFCS*

Due to the anticipated remoteness of the UAS from its operator(s) and the GCS, there are likely to be situations where the ground-based pilot could lose operational situational awareness or lose contact with the vehicle entirely (e.g., due to a data link drop-out). It is therefore considered a sensible approach to ensure that the UAS therefore be capable of making its own decisions. This being the case, it must be ensured that the outcomes of those decisions do not present any loss of safety to other airspace users. One means of achieving this goal is being investigated as part of this study. The method being used is to develop an agent-based system that forms the core of the CFCS. The agent is charged with being able to make the same informed and rational decisions that a human pilot would in a given situation[13]. For this project, this agent is called the 'Executive' and its role is to make high-level decisions about the progress of the mission based upon: (i) its *beliefs* about the vehicle status and external environment, and, (ii) its *desires* (e.g., successful completion of the mission). The results of these decisions are *intentions* to act to achieve the Executive desires. The Executive is supported by a ‘Planner’. The Planner’s role is to reduce the computational burden on the Executive by generating plans which can be either rejected or approved (and then executed) by the Executive. For example, route planning is (in general) a complex computational problem, and if implemented within the Executive would reduce its responsiveness. Therefore the generation of route plans (as well as other plans, such as evasive manoeuvres) is assigned to the Planner.

The Executive is an independent application written using the autonomous agent language Gwendolen[16]. Gwendolen itself is written in Java, a popular general-purpose high-level language. An advantage of using Gwendolen is that agent programs can be verified formally in order to determine their correctness, e.g., relative to the Rules of the Air. Formal verification of Gwendolen programs is made possible using the Agent JPF model checker, described in more detail in Section V.0.

Typically, the Executive is programmed with a mission goal (in this example, the UAS will fly from Aberporth to Sumburgh), defined as a set of plans which relate to the UAS flight phase. The flight phases in this example are:

1. Waiting at ramp.
2. Taxi to runway hold position.
3. Taxi to runway line-up position.
4. Takeoff and climb.
5. Cruise (navigation).
6. Sense-and-Avoid manoeuvre.
7. Descend and runway approach.
8. Landing.
9. Taxi to ramp.

The flight of the autonomous UAS proceeds as follows. The Executive sends the desired destination to the Planner. The Planner, which possesses a geographical map of the UK with aeronautical navigation fixes, calculates a route to the desired destination using a shortest-path graph search algorithm such as Dijsktra’s Algorithm[17]. The route that the aircraft should follow, consisting of a list of waypoint coordinates (latitude and longitude) is sent back to the Executive. For the terminal phases of operation (in and around aerodromes), the Planner stores additional waypoints that are required for taxiing to and from the runways and for the approach and landing profiles (aircraft can land from either direction on a runway depending on wind conditions). Therefore the Planner must be equipped to select the appropriate plan based on current airfield operations.

When the Executive receives the route from the Planner, it must determine if the plan is acceptable. Typically, the Executive will base this decision on its current beliefs, desires and intentions. For example, the Executive may present two navigation plans to the Executive for review. The Executive may determine that one of the plans will reach the destination in a shorter period of time but may result in a loss of safe separation from another aircraft. The other plan will take longer but maintains safe separation. If the Executive believes that safe separation is a safety-critical issue but time is not, then it will select the plan to maintain safe separation.The Executive actions the plan by forwarding it to the flight control system. Once the plan is being executed, the flight control system takes the approved route and determines the correct heading to follow from the aircraft's current position.

To date only one Planner is used; however, it is possible to use multiple Planners based on different architectures. For instance one planner could use Dijkstra’s algorithm for route planning, whereas another may use a genetic algorithm. The Executive can then determine the most appropriate plan to pass to the flight controller. Additionally, planners may take the same approach to route planning (i.e., use the same algorithm), but are implemented by different software developers in order to eliminate the possibility of a single point failure.

B. Environment

The modeling of the vitual environment extends to a representation of the physical geography of the UK as well as a model of its associated airspace system.

1. *Physical World Model*

Simulation of and mitigation against hazards can really only be achieved if a modelled system is immersed in it's (modelled) environment. In this case, a UAS in UK airspace. Although FLIGHTLAB[14] provides a tool to develop a high fidelity flight mechanics model of the aircraft platform with CFCS and simulation environment, it does not include a physical world database based on UK real-world coordinates and terrain elevation. This is an important aspect of the simulation as the UAS must be able to take-off/land at designated airfields using a predefined flight plan. If the UAS does not have the correct altitude information, it cannot perform the flight phases involving terrain.

This has been addressed by including X-Plane® (http://www.x-plane.com/) software into the VESL architecture to provide coordinates and terrain data for the UK, as well as being used as a visualisation tool. UAS position and orientation data is sent via the communications hub to X-Plane as illustrated in Figure 6, where the data is incorporated by way of an X-Plane plug-in, allowing the UAS to be visualised in the X-Plane environment on a computer/TV monitor or on the VEC's 6.0m x 2.1m visualisation suite, Figure 7. Using this architecture, X-Plane is provided with the latitude, longitude and barometric altitude of the UAS. Using this information, the height above terrain can be determined using the X-Plane database and radar altimeter function. These data are broadcast back to the UAS FLIGHTLAB model, again via the hub. Using this height above terrain parameter, a ground contact model can be triggered within the FLIGHTLAB model when a 'detection' between the aircraft tyres and the ground is flagged.

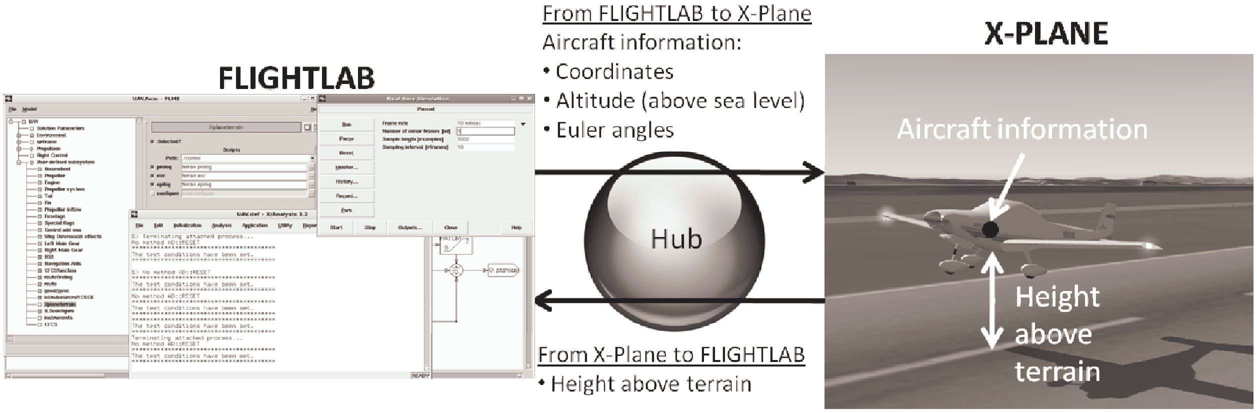

Figure 6. Data Flow between FLIGHTLAB and X-Plane

Figure 7. VEC 6m x 2.1m visualization screen showing outside world and air traffic situation

2. *Managed Airspace*

Through the use of X-Plane, the UAS has been incorporated into a geographical model of the UK. However, if the VEC is to demonstrate that the UAS can be integrated into non-segregated airspace without impacting on the safety of other airspace users, then clearly a representative model of UK airspace and its associated air traffic must also be incorporated into the simulation.

In this study, managed airspace and air traffic simulation is achieved using a modified version of air traffic simulation game 'London Control'. This software provides a high fidelity representation of UK air traffic operations, complete with representation of airways, terminal areas and air traffic control sectors. Developed by DM aviation (http://www.londoncontrol.com/), it has been further developed by its creator, under the guidance of one of the project's industrial partners (independent of the VEC project) to allow its functions to be used independently of the game play. This functionality is now being used at the VEC. The flight plans which aircraft follow in London Control are created from recorded samples of air traffic movements. However, the user can inject additional flights which will, for example, influence the decisions made by the UAS Executive. Figure 8 shows screenshots of the managed airspace environment. The FLIGHTLAB UAS (callsign VEC123) is passing an aircraft injected from the ATC simulation (call sign INT001). The Figure shows both the ATC and X-Plane components of the simulation which are connected via the central communications hub.

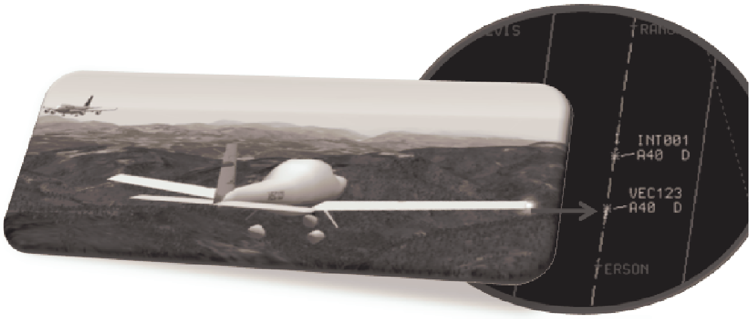

Figure 8. VESL-introduced entities to the virtual environment

C. Sense and Avoid Functionality

The initial scenario implemented in VESL is based upon a flight between two airfields across a range of UK airspace classifications, with the flight phases defined in Section IV. This basic scenario was chosen as it encapsulates the core flight phases that must be performed in every flight, manned or unmanned. In addition to these core flight phases, mission-specific flight phases will need to be built into the simulation. Such topical missions include gathering information on volcanic ash clouds or monitoring radiation levels at a nuclear power plant. In addition to the mission specific capabilities, if fully autonomous UAS are to be seamlessly integrated into managed airspace, their capabilites must extend to functions such as sense and avoid, as specified in Ref. 3.

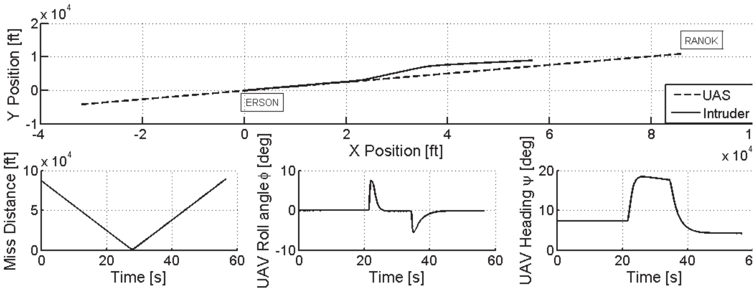

Figure 9. Example collision avoidance

A simple non-physics based sense and avoid function has been implemented in the VESL UAS. This is illustrated in Figure 8 and Figure 9. In this scenario, the UAS (VEC123) is en route to it's destination and passes waypoint ERSON on a heading of approximately 007^o towards the next reporting point, RANOK. Meanwhile, another so-called 'intruder' aircraft (INT001) has passed RANOK on a heading 187^o towards ERSON at the same altitude as VEC123. The UAS, equipped with a simple sense and avoid algorithm, detects that it is on a direct collision course with the intruder aircraft. The sense and avoid function, which is currently receiving data on other aircraft in a non-physical way from the central hub, calculates the distance between the two aircraft and the relative heading. If seperation becomes less than an arbitrarily defined value, in this case 20,000 ft, the UAS FCS, which is in the 'Cruise' flightphase, notifies the Executive. The Executive determines that evasive action is required and changes the aircraft flight phase to 'Sense-and-Avoid manoeuvre'. In this situation, a new heading, based upon the UAS adhering to the Rules of the Air (i.e. turn right), is passed directly to the heading controller. This will ensure the 1000ft seperation distance (an arbitrary figure) is adhered to as illustrated in Figure 9. Figure 8 shows screen capture images of the evasive manoeuvre as visualised in X-Plane where the UAS has turned right to pass the intruder. Once the intruder aircraft has passed, the Executive returns to flightphase 'Cruise' and the UAS continues to its original waypoint.

This is a rudimentary collision avoidance algorithm in two respects. Firstly it is a very simple algorithm activated only if another aircraft is within a given distance and on a collision course. The VEC aims to build upon this ground work by including into the facility industry experimental algorithms such as that described by Gates[18], as well as routines for returning to the original flight plan after such excursions[19].   Secondly, the simulation as described above is based solely upon truth data, i.e. the sense and avoid function always has access to the exact position and heading of all other air traffic. In practice, data pertaining to other airspace users would be obtained intermittently (but regulary) via radar, air traffic control or by other sensors on board the aircraft such as electrooptic/infra-red (EO/IR) sensors. Furthermore, this information still does not necessarily supply the precise location of other aircraft as the data will be subject to inherent measurement errors. Therefore the example discussed, could have many potentially different outcomes if the errors inherent to system components were incorporated into the simulation. It is intended that data that has realistic variability in it will be introduced to the simulation scenarios. This will initially by applying mathematical variability to a number of the sensor inputs to the model but later, physics-based sensor models will be introduced to the VESL.

Ultimately, the data supplied by the sensor suite to the sense and avoid function will be used by the onboard Executive in determining if and when avoiding or evasive manoeuvring would be necessary. If the Executive is presented with inaccurate information, a wrong decision could be taken resulting in a catastrophic incident. Therefore, it is safety-critical to ensure that, in any situation, the Executive makes a safe decision. VE can be used here to generate evidence that could be used to inform a safety assessment. One means of testing this would be to run a particular scenario many times and record the outcome. An alternative approach would be to prove that the Executive will make decisions to demonstrate that safety regulations are not breached or that when they are, it is to preserve the required level of safety. These safety assessment methods are the focus of the next section.

V. Generating Evidence for Certification

With the development of the VESL synthetic environment, the VEC has demonstrated a tool which it is currently using as a test-bed to develop a methodology for demonstrating evidence that a component of, or a subsystem of a UAS meets certification requirements. As stated in Sections 2 and 3, the VEC has selected a test-case based upon the sense-and-avoid function for the development of it's data collection techniques. The first step in this process is to identify failures and perform a Functional Hazard Assessment (FHA) to determine the perceived level of severity of a failure or combination of failures within a function. The overall safety requirement for UAS systems is that of equivalence to the current level of “safety” of manned aircraft[3]. Therefore, to define what UAS safety objectives should be, it is necessary to determine what the current level of safety of human-piloted aviation is. In manned aircraft, the severity of failure(s) is defined in five levels (catastrophic, hazardous, major, minor, and no effect[3]). The prescribed level of severity for a failure then confers a probability of failure per flight hour, to which the function must be designed against. In the current case study for the sense and avoid function, the perceived worst case is that a failure could result in a loss of safe seperation, ultimately resulting in a collision with another aircraft and loss of life. Therefore, a failure in the sense and avoid function is classed as catastrophic (Ref. 2 argues that this failure should ne classed as major as air traffic control will provide seperation in Class A airspace) and must be designed such that the probability of the failure occuring is no more than once every 10^{-9} flying hours[3].

Once the failure rate has been determined, the function has a design requirement to which the UAS manufacturer must provide evidence to the regulatory authorities that all components used by a function, work together to meet the safety requirement and conform with the rules of the air. Component manufacturers will supply evidence to the UAS manufacturer of the reliability of the component. The UAS manufacturer can then utilise this reliability figure in a fault tree, to determine if the new solution meets the design goal. However, this still provides little solid evidence that the system works correctly and meets the operational requirements. Thus, the greater the amount of evidence that can be gathered to present to the certification authorities, the greater the confidence will be that the system has met the prescribed requirements. Two further data gathering techniques to support the evidence supplied through fault tree analysis are considered at the VEC.

A. Hazard Based Analysis

The first is the industry standard 'hazard based' Monte Carlo simulation method whereby a scenario is run many times to gather evidence for certification, building a picture of how a component will impact upon a UAS decision making process. The VEC is well placed to generate this type of evidence as it has access to, through the STFC Daresbury Laboratory, computational power at a level able to tackle the massive complexities of VE required for aerospace systems. The VEC will build models of sensors[20], ranging in complexity and tolerances into the existing facilities described in this paper, which are used to drive the input to the sense and avoid function. Then, using the high Powered Computer (HPC) facilities at Daresbury[21], perform the simulation many times to demonstrate and build if the system does/does not meet certification requirements. This can be a costly and timely exercise as potentially billions of scenarios need to be investigated. Ultimately however, the process is not exhaustive and critical failure conditions, particularly in a UAS CFCS could be missed. the VEC is also investigating another methodology, using Formal Methods[22] to demonstrate 'human equivalence' of a CFCS by exhaustively assessing how the decision making processes can impact upon the frequency of an identified risk occurring, to deliver evidence of certification.

A. Demonstrating Human Equivalence

The VEC is investigating the use of *formal verification*[13] to demonstrate "human equivalence" of an agent-based autonomous UAS control system, Figure 10. Formal verification is an example of the use of formal methods, i.e., mathematics-based approaches to the specification, development and verification of computer systems. Model checking is a popular and widespread approach to formal verification, and involves creating a model of a program or process which is then analysed exhaustively to determine whether a given property holds for that program/process[16]. For example, a sense-and-avoid system for an autonomous UAS could be modelled, and a determination made that the following property holds: "It is always the case that if a sensor has detected a risk of collision, then the UAS is taking evasive action." Properties are typically specified using *temporal logic*, which provides a rigorous way to describe events which take place over time. For example, "in the next moment", "later" and "always" are concepts expressible within temporal logic. *Modal logic* can be used to specify formally concepts surrounding knowledge, intention and motivation.

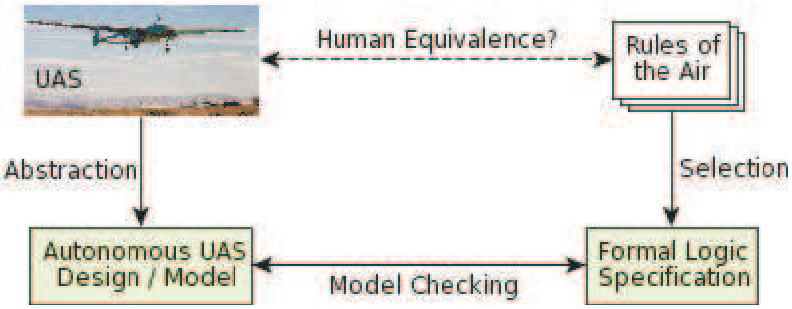

Figure 10. Assessing human equivalence of an autonomous UAS using formal methods and the Rules of the Air (Ref.6)

An *agent* is a computer system capable of autonomous action within an environment towards achieving its goals[13]. A *rational agent* is an agent that must have explicit reasons for making decisions. The rational agent concept therefore provides a natural and flexible way to describe autonomous systems. An autonomous UAS control system has been implemented using *Gwendolen*, an agent programming language in the BDI paradigm[16] (BDI stands for beliefs, desires and intentions and describes an approach to reasoning[13]). Gwendolen agent models are executable, meaning that they are already implemented through a Java interface and therefore can be integrated into other software and hardware systems. The Executive described in Section IV was implemented as a Gwendolen agent for use within the VESL.

Agents programmed in Gwendolen can be verified formally using the AIL (Agent Infrastructure Layer) and the Agent JPF model checker developed as part of the MCAPL project[23]. It is possible to verify, for example, properties concerning an agent's beliefs, desires and intentions. Programming a rational agent using Gwendolen and verifying key properties relating to the Rules of the Air[6], for instance, may be useful for demonstrating the airworthiness of the rational agent's decision making by demonstrating equivalence with human decision making. Clearly, the closer the autonomous UAS model is to the actual UAS control system implementation and the closer the properties checked are to the actual meaning of the Rules of the Air, the more useful model checking will be in generating analytical evidence for certification. Ideally, the autonomous UAS model should specify all of the decisions/choices the UAS can possibly make. As a feasibility study, a small subset of the Rules of the Air was examined:

1. **Sense and Avoid**: "... when two aircraft are approaching head-on, or approximately so, in the air and there is danger of collision, each shall alter its course to the right." (Section 2.4.10, Ref. 6)
2. **Navigation in Aerodrome Airspace**: "[An aircraft in the vicinity of an aerodrome must] make all turns to the left unless [told otherwise]." (Section 2.4.12(1)(b), Ref. 6)
3. **Air Traffic Control Clearance**: "An aircraft shall not taxi on the apron or the manoeuvring area of an aerodrome without [permission]." (Section 2.7.40, Ref. 6)

These rules were chosen as cases of particular relevance to UAS autonomy: "Sense and Avoid" and "human in the loop" cases (rules 1 and 3 respectively) are essential for UAS engineering[24]. In addition, Rules 1 and 2 are interesting because they are potentially conflicting – Rule 1 dictates that aircraft should turn right, and Rule 2 states that turns should be made to the left in aerodrome airspace unless told otherwise – and thus presents an interesting challenge for engineering and formal verification. For example, an autonomous UAS may be flying through aerodrome airspace when it encounters an intruder aircraft approaching head-on. At the point Rules 1 and 2 are in conflict, as Rule 1 specifies that the UAS should turn right, and Rule 2 specifies that the UAS should turn left. Ideally, the autonomous UAS should demonstrate airmanship when deciding between conflicting alternatives: a key challenge in UAS engineering. Airmanship also presents a challenge for formal verification: when we verify the autonomous UAS's behaviour, it may be the case that Rule 1 or Rule 2 is violated (or that both are violated) as a result of design decisions made during the UAS's development. Therefore, do we verify Rule 1, Rule 2, both, neither, or an alternative rule?: Using the Agent JPF model checker it was shown that the three properties corresponding to the above scenarios were satisfied by the agent-based autonomous UAS control system programmed in Gwendolen. A more detailed exposition of the work on the use of formal verification to demonstrate human equivalence of autonomous UAS can be found in Ref. 25.

Fault trees are routinely used in providing evidence for certification for manned aircraft, primarily for determining hardware reliability[26]. In addition, manned aircraft are heavily dependent on pilot-in-the-loop for control and guidance, therefore, humans will feature heavily in a fault tree and nodes that feature a human will require a failure rate. However, analysing human behaviour is in general very difficult, and the failure rates may be unreliable[27]. It may be possible however, to produce more reliable failure rates for autonomous UAS. It should be noted that fault trees are not usually used to represent software systems, however, it is considered that there might be some merit in the approach, described below, with regards to model checking certain conditions/scenarios.

As described in Section V.0, model checking is used for automatic, exhaustive exploration of a program or process model with the aim of proving that a given property or condition holds for every possible execution of the program or process. Typically, model checking provides a yes/no answer to a given property: either the property holds for a program/model, or not. Probabilistic model checking extends "traditional" model checking to include numeric results for a given property[27]. For example, it may be the case that a particular property holds in a given number of cases. Here, a probabilistic model checker enables that quantity to be determined. For example, in the autonomous UAS Sense-and-Avoid example, it may be the case that the sensor the UAS uses has a 1% failure rate. This failure rate can be integrated into the model of the autonomous UAS to determine the results of this inaccuracy on the Sense-and-Avoid functionality. The property, "It is always the case that if a sensor has detected a risk of collision, then the UAS is taking evasive action," would have a numeric answer when analysed with a probabilistic model checker. This would help to provide a statistical measure of the autonomous UAS's adherence to the property.

Similarly, probabilistic models of autonomous control systems for UAS can be constructed. Probabilistic aspects of the environment can be included, e.g., sensor failure rates, and the resulting effects on the model analysed. To demonstrate this, a model of an autonomous control system was developed using the PRISM language[28], and sensor errors were incorporated for the sense-and-avoid sensor. The PRISM model checker[28] was then used to calculate the resulting adherence to one of the Rules of the Air corresponding to Sense-and-Avoid (Rule 1 above). For instance, the sensor was given an error rate of 0.01%, and it was found that the property corresponding to Rule 1 was satisfied 99.98% of the time.

It is possible that this approach could be used to determine the effects of failure rates for a given component/sub-system on the behaviour of an autonomous UAS. For example, a given hardware sensor could be analysed and its failure rate determined. This failure rate could be integrated into a probabilistic model of an autonomous UAS and the probabilistic model checker could be used to determine the effects on the overall behaviour of the autonomous UAS and quantify failure rates of higher level functions or hazards within a fault tree. Of course, whenever using models or simulations the results obtained are constrained by the accuracy of the model; however, assuming that a probabilistic model could be validated then the use of probabilistic models to determine failure rates for autonomous UAS would perhaps be useful for safety engineering of autonomous UAS.

VI. Conclusion

This paper has reported on the Virtual Engineering Centre (VEC), a University of Liverpool initiative, which aims to develop a centre of excellence in Virtual Engineering by creating integrated Virtual Engineering processes, tools and techniques across the Product Life Cycle (PLC). The VEC project has broken the PLC into a number of WPs and each WP has an associated case study. The case studies are intended to provide realistic examples through which the integrated tools, techniques and processes will be generated. The focus for this paper has been WP3, the Test and Certification phase of the PLC. The associated case study is based upon the problem of the engineering and certification requirements that will allow routine autonomous UAS operations in non-segregated airspace.

The Virtual Engineering Simulation Laboratory has been described. This newly created facility allows a virtual UAS to be operated in a realistic synthetic environment. It's initial use has been as a test-bed for a proof-of-concept exercise to demonstrate the efficacy of rational autonomous agents being used to replace the pilot as the decision-maker on-baord an aerial vehicle. An example of the autonomous decision-making agent working in conjunction with the UAS in it's operational environment has been presented. It is believed that this is the first time that such a combination of systems has been demonstrated.

In addition, the VESL facility is being used to develop techniques aimed at building confidence in the ability to gather virtual engineering evidence that a component of, or a subsystem of a UAS, will meet its certification requirements. The processes being developed make use of the high powered computing facilities available to the VEC to perform Monte Carlo analysis of system performance and the use of Formal Methods to verify that a Complex Flight Control System will consistently make the same correct decisions a human pilot would make in order to follow the Rules of the Air.

Appendix A

Figure A1 shows a schematic of UK controlled airspace. The route to be taken by the autonomous UAS developed at the VEC is highlighted. The departure and arrival aerodromes, and reporting point names, are also shown.

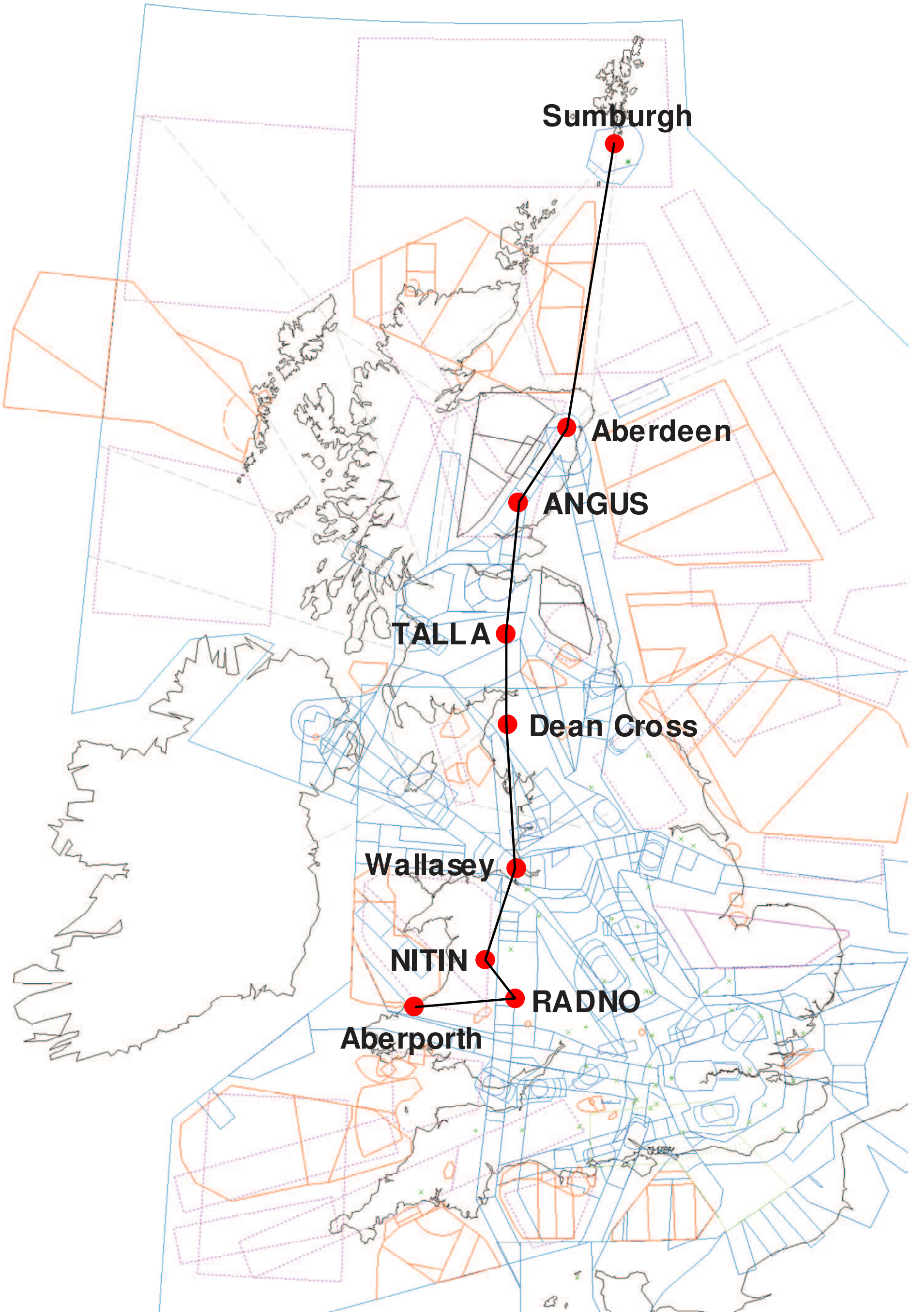

Figure A1. Schematic illustration of UK controlled airspace

Acknowledgments

This work is supported through the Virtual Engineering Centre (VEC), which is a University of Liverpool initiative in partnership with the Northwest Aerospace Alliance, the Science and Technology Facilities Council (Daresbury Laboratory), and a number of industrial partners. The VEC is funded by the Northwest Regional Development Agency (NWDA) and European Regional Development Fund (ERDF) to provide a focal point for virtual engineering research, education and skills development, best practice demonstration, and knowledge transfer to the aerospace sector.

References

[1]Anon., “Certification Specifications for Large Aeroplanes CS-25, Amendment 8,” European Aviation Safety Agency, 18th Dec. 2009.

[2]Kelly J., et. Al., “Preliminary Considerations for Classifying Hazards of Unmanned Aircraft Systems,” NASA Technical Report TM-2007-214539, Feb. 2007.

[3]Anon., “CAP722 - Unmanned Aircraft System Operations in UK Airspace – Guidance,” Directorate of Airspace Policy, Civil Aviation Authority, 6th Apr. 2010.

[4]Anon., “Astraea,” URL: http://www.uavs.org/astraea [cited 18 July 2011].

[5]Cooper, J. E., et al., “Virtual Engineering Centre – Examples of Virtual Prototyping and Multidisciplinary Design Optimization,” *Proceedings of the NATO RTO Workshop AVT-173 on Virtual Prototyping of Affordable Military Vehicles Using Advanced MDO*, Sophia, Bulgaria, May 2011.

[6]Anon., “CAP 393 Air Navigation: The Order and the Regulations,” Civil Aviation Authority, 14th April 2010.

[7]Anon., “Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities,” Version 3.2,

[8]Clarke, E.; M., Wing, J., M., et al., "Formal Methods: State of The Art and Future Directions," *ACM Computing Surveys*, Vol. 28, No. 4, December 1996.

[9]Padfield, G.D., White, M.D., “Flight Simulation in Academia; HELIFLIGHT in its first year of operation,” *The Aeronautical Journal of the Royal Aeronautical Society*, Sep. 2003.

[10]White, M. D., Perfect, P., Padfield, G. D., Gubbels, A. W., Berryman, A., “Progress in the Development of Unified Fidelity Metrics for Rotorcraft Flight Simulators,” *American Helicopter Society 66th Annual Forum Proceedings*, Phoenix, USA, 2010.

[11]Cameron, N., Haverdings, H., Bruno, G., Sandri, F., Sodano, M., Paul, H., “The Impact of Civil Tilt Rotor Aircraft on the Air Traffic Management Environment - A Virtual Engineering Assessment,” *Proceedings of the 36th European Rotorcraft Forum*, Paris, France, 2010.

[12]Cameron, N., White, M. D., Perfect, P., Jump, M., Padfield, G. D., “University of Liverpool Networked Simulations Case Studies into Recreating the Past and Shaping the Future,” *Heliworld 2010*, Frankfurt, 2010.

[13]Wooldridge M., *An Introduction to Multiagent Systems*, John Wiley & Sons, 2002.

[14]Duval, R.W., “A Real-Time Multi-Body Dynamics Architecture for Rotorcraft Simulation,” *The Challenge of Realistic Rotorcraft Simulation*, RAeS Conference, London, November 2001.

[15]Anon., “MIL-HDBK-1797 - Flying Qualities of Piloted Aircraft,” Air Force Flight Dynamics Laboratory, WPAFB, Dayton, USA, 19 Dec 1997.

[16]Dennis, L. A., Farwer, B., “Gwendolen: A BDI Language for Verifiable Agents,” *Logic and the Simulation of Interaction and Reasoning, AISB’08 Workshop*, 2008.

[17]Dijkstra, E. W., “A Note on Two Problems in Connexion with Graphs,” *Numerische Mathematik 1*, 1959, pp. 269–271.

[18]Gates, D. J., “Properties of a Real-Time Guidance Method for Preventing a Collision,” *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, May–June 2009.

[19]Gates, D. J., “Nonlinear Path Following Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, March–April 2010.

[20]Merhav, S., *Aerospace Sensor Systems and Applications*, Springer Press, 1996.

[21]Emerson, D.; Engineering at the Edge: Challenges for Petaflop Computing and Computational Fluid Dynamics, *First International Conference on Parallel, Distributed and Grid Computing for Engineering*, p1-14, 2009.

[22]R. Bordini, M. Dastani, J. Dix, *Multi-Agent Programming:Languages, Tools and Applications*. Springer, 2009.

[23]Dennis, L. A., Fisher, M., Webster, M. P., Bordini, R. H., “Model Checking Agent Programming Languages,” *Automated Software Engineering* (to be published).

[24] Patchett C., Ansell D., "The Development of an Advanced Autonomous Integrated Mission System for Uninhabited Air Systems to Meet UK Airspace Requirements," *Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation*, 2010.

[25] Webster, M., Fisher, M., Cameron, N., Jump, M., "Formal Methods for the Certification of Autonomous Unmanned Aircraft Systems," *Proceedings of the 30th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2011)*, Naples, Italy, Sep. 2011 (to be published).

[26] Anon., "ARP4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment," Society of Automotive Engineers, December 1996.

[27] Kirwan, B., "The Validation of Three Human Reliability Quantification Techniques - THERP, HEART and JHEDI: Part 1 - Technique Descriptions and Validation Issues," *Applied Ergonomics*, Vol. 27, No. 6, 1996, pp. 359-373.

[28] Kwiatkowska, M., Norman, G., Parker, D., "PRISM: Probabilistic Symbolic Model Checker," *Proceedings of the 12th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS)*, Springer LNCS, vol. 2324, pp. 200–204, 2002.