Article

FUSE: Future Urban Simulation Environment Towards Integration of Uncrewed Aerial Vehicles in Low-level Airspace

Toufik Souanef [1]*, Dimitrios Panagiotakoupoulos [1], Antonios Tsoudors [1], Chris Crockford [2], Chris Cheetham [3] and Chris Mewse [4]

[1] Cranfield University, Cranfield, UK ; toufik.souanef@cranfield.ac.uk; d.panagiotakopoulos@cranfield.ac.uk; a.tsourdos@cranfield.ac.uk
[2] Electric Aviation Ltd, Colchester, UK; Chris@ChrisCrockford.com
[3] Remex Software Ltd, Guildford UK; chris@remexsoftware.com
[4] Geoxphere Ltd, Basingstoke UK; chris.mewse@geoxphere.com
* Correspondence: toufik.souanef@cranfield.ac.uk

Abstract: This paper presents the design of a digital twin that blends aviation, gaming, simulation, and Geographic Information Systems (GIS) to create a synthetic environment within which strategies, laws and platforms for electric aviation may be tested out. This digital twin has been called Future Urban Synthetic Environment (FUSE). FUSE includes an in-built Unmanned Traffic Management (UTM) that can be used to run simulations to test the coordination of all urban traffic. It uses real GIS tagged imagery data and implements it at runtime into the game engine and thus link the optimised imagery loading to the visual performance of the simulation engine. FUSE provides a 3D digital twin of specified areas designed to simulate the effects (in terms of noise, visual impact, privacy) of drones and electric air taxis operating under various operational scenarios (such as number of deliveries allowed per day, maximum payload weight, no-fly areas, position of depots, vertiports, etc.). With so much high-fidelity data it is difficult for any game system to effectively render the environment and do justice to the detail whilst enabling enough of the landscape to be rendered to keep in focus the detail when looking out at the horizon.

Keywords: Digital Twin; Unmanned Traffic management; Geographic Information Systems; Immersive Simulator; Unmanned Aerial Systems

1. Introduction

Uncrewed Air Vehicles (UAVs or drones) offer a huge opportunity to add a new layer to every country's established transport infrastructure by providing delivery and transport services directly to the customer, amongst many other applications. This relieves pressure on road networks, which are already at full capacity in most urban areas.

The implementation of such drone services in any town will require the active cooperation and involvement of both the Local Council and the residents. However, both these groups are likely to be highly resistant to the introduction of drone services with regards safety, privacy and noise (as well as the natural resistance to change). Furthermore, the thinking behind UAV Traffic Management (UTM) so far has come from Air Traffic Control (ATC) specialists, taking a "one size fits all" approach, which ignores the needs, laws and requirements of the local communities. Leicester city council have for example invoked a by-law based on the road traffic Act, prohibiting overflight of any council-owned road by a drone without their prior permission [1].

The objective of this paper is to present an immersive (i.e. based on Virtual Reality) UAV simulator: Future Urban Synthetic Environment (FUSE). FUSE is a digital twin built using the latest technologies, by blending aviation, gaming, simulation, and Geographic Information Systems to create a synthetic environment within which strategies, laws and platforms for electric aviation may be tested out. Such simulations can then be run using FUSE's inbuilt UTM to test the coordination of all urban air traffic. Being built out of an

 doi:10.20944/preprints202205.0354.v1

already published gaming flight simulator, FUSE allows to design from the ground up, the first 400 feet(ft) of urban airspace, tackling the concerns of Air-Space Regulators, Local Government and the public alike.

Immersive technologies search to introduce the operators in robots' environment to deal with problems such as lack of situational awareness and alleviating workload peaks [2]. There are three main immersive technologies: Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR); although some authors merge the latter two into a unique category. Virtual and Augmented reality headsets allow for more immersive interaction with UAVs [3]. Most commercial VR devices take advantage of powerful gaming engines such as Unreal [4,5] and Unity [6,7] which allow users to develop a wide range of realistic scenarios, intelligent characters and objects with realistic dynamics and kinematics [8]. In [9] two methods were presented to genera virtual reality content based on UAV systems. The first one is using a head tracking system and the second is using a panoramic 360 field of view camera array. The former can be employed to achieve real-time virtual reality experiences for users. The latter can be used to generate offline VR content. With the head-camera synchronisation and camera array techniques, the real-time and offline VR methods have been developed to provide users with two kinds of VR experiences [9]. A "man-in-the-onboard-loop" UAV telepresence and simulation platform based on virtual reality technology was proposed and validated in prototype [10]. The objective was to improve the stability and reliability of UAV telepresence, reduce the difficulties of controlling UAVs from a long distance, whilst providing the training of UAV pilot with effective support.

In [11] a flexible approach was presented to easily simulate UAVs based on Unreal Engine and Visual Studio. The system, combined with VR technology, helped the trainees to immerse themselves into the virtual environment so that they could be trained more effectively and attentively without being affected by a wide range of environmental factors. In [12] the implementation of AVR was reviewed in Quadcopter drones for entertainment. The authors in [13] proposed the integration of a VR system with a low-cost semi-autonomous UAV, with the objective of immersing the user in the environment of the aircraft, providing visual content to people with limited mobility. This study combines the robot operating system (ROS) and the AR drone to provide the required features, such as the transmission of control data and video feed. The work in [14] used web technologies to create a VR environment reconstructed with real-world fly data and simulate the control of a UAV to present a higher perspective of 3D path planning to pilots. Besides the manual navigation of a UAV, the simulator includes functions such as automated routing to find the optimal path among the mission's locations, a safe return to the home point, and object detection and tracking.

However, the previous works did not include multi-UAV systems and UTM interface in the developed solutions. Furthermore, the modelling of GIS information in simulators and game engine was not addressed explicitly. This is a challenging task for the following reasons:

- There are many GIS formats unsupported that need to be manually implemented / deserialised.
- Data volume: generally dealing with very large data sets which will not fit into RAM. Therefore, a streaming solution requires to be devised to load / unload at runtime.
- Geometry: Generally not optimised for real-time rendering, runtime level of detail solutions need to be implemented to run at sensible frame rates.
- Numerical precision issues: Game engines generally run on floating-point numbers which do not have the required precision when traversing large areas of the world / terrain without a world rebasing solution.
- Visualisation: All GIS layers / methods of visualisation will need to be manually implemented.

The FUSE simulator is innovative, as not only does it develop an immersive simulator, allowing for scenario planning and load testing of UTM systems, it also brings the local authority and the residents into the picture, allowing them to help shape this new transport

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

revolution that will be implemented in their area. This is likely to lead to a much faster acceptance and adoption. Another innovation is that the adopted approach focuses on blending existing technologies to create a UTM visualization experience providing insight into simulated business models. The power of GIS and gaming was harnessed to advance the development of UTM and empower:

- The commercial operators to simulate and find their optimal strategy.
- The people who live underneath such operations to realise what it will mean practically, without prejudice or bias.

Furthermore, this is the first time that a real GIS tagged imagery data has been loaded into a game engine at runtime and the optimization of the imagery loading engine is directly linked to the visual performance of the simulation engine.

The input of the local communities will add a vital layer of design features and constraints (such as operational hours and no-fly zones, etc.) which will need to be encoded into the UTM systems eventually. We speed up both sides of the coin, UTM development and public acceptance.

Potential use of the FUSE simulator might include:

- The UTM vendors and delivery companies could establish and test the implementation of the local authority's preferred schemes of operation. This will require for instance the establishment of local depots (mobile, or on the edge of town) from which the drones can pick up their loads for the last-mile delivery service.
- Air Traffic Management (ATM) Authorities could test the safety of the UTM services when considered in combination with other airspace users (police and ambulance helicopter services as well as normal aircraft operations).
- Drone developers, ATM and UTM vendors could establish and test drone air lanes as well as autonomous collision avoidance rules-of-the-road to allow drones to safely occupy the same airspace.
- Local Authorities being able to develop schemes and supportive legislation such that drones may be limited to operations below 400ft, and with limits on their missions according to their local needs (e.g. with no more than 200 sorties per day in a specific area and within the operating hours of 8 am to 8 pm). Drones may not operate below 300ft over parks or gardens and wherever possible drones must follow the road network. Medical deliveries and blue light services will always take priority.
- Local Authorities being able to conduct public consultations to demonstrate the effect of the proposed service from any geographical point in terms of noise and visual impact.

The main contribution of this paper is the presentation of an immersive flight simulator environment capable of:

- Modelling GIS information to create an informed urban environment.
- Scheduling drone flights around this environment such that the user might move around the environment and experience the drones flying.
- Modelling multiple drone flights in a VR simulator.
- Providing UTM interfacing information.

2. FUSE Simulator Design

The FUSE simulator takes a system-of-systems approach to enable us to test UTM systems and run simulations that:

- Set flight levels and limits on operations.
- Balance commercial potential with environmental impact.
- Engage and take the public with us on this journey.
- Help develop legal model articles to regulate urban airspace.
- Publish operational routes from the Local Authority into the UTM system.

The research improves on the current state-of-the-art of air traffic simulator by "gamifying" drones within an urban environment allowing UTM systems to be tested by an independent tool and the effects of different strategies to be realized, both environmentally and commercially by the operators, the Local Authorities and the public alike.

2.1. REMEX immersive flight simulator

Deadstick is a first-person high fidelity flight simulator built from the ground up to solve the problem of simulating low and slow bush flying [15]. Rather than trying to simulate the whole world, it instead tackles a series of fixed locations that, with the use of developed bespoke tools could be procedurally generated and populated to a very high fidelity where all objects / obstacles within the world are modelled and simulated (grass, rocks, foliage, etc.). This is something that has not been traditionally possible in flight simulators. A similar approach has been taken with all aspects of the simulator from modelling wind and other weather effects, and their interactions with the trees / terrain as well as the airframe itself to allow players to perform full walk-around and preflight inspections of the aircraft.

2.2. Airborne

The used airborne platform is the Mugin 3.6 SLT (Separate Lift and Thrust) Uncrewed Aerial System (UAS) [16]. Separate Lift and Thrust UAV is an integration of fixed-wing and multirotor to get vertical take-off and landing capability with high range and long endurance [17].

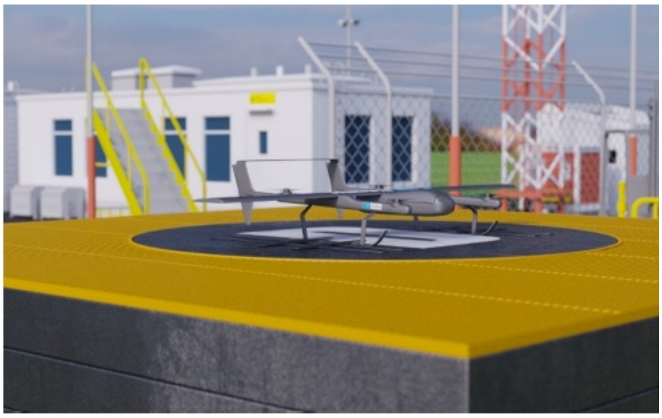


Figure 1. The 3D Studio max model of the 3.6m MUGIN SLT UAS.

The Mugin UAV has been evolving during the project development. More specifically, its flight and payload systems have included the option of operating the aircraft on floats. Future advances in the Mugin airframe structure may be imported from CAD into 3DStudioMax, animated and then exported directly into FUSE. This permits to rapidly engineer modifications to the airframe and then import them directly into the FUSE game engine.

The exact dynamic model of the Skylift Mugin was not implemented as the accurate information was not accessible. The flight model is just an object that will interpolate itself towards points on a spline with a sensible representative velocity.

 doi:10.20944/preprints202205.0354.v1

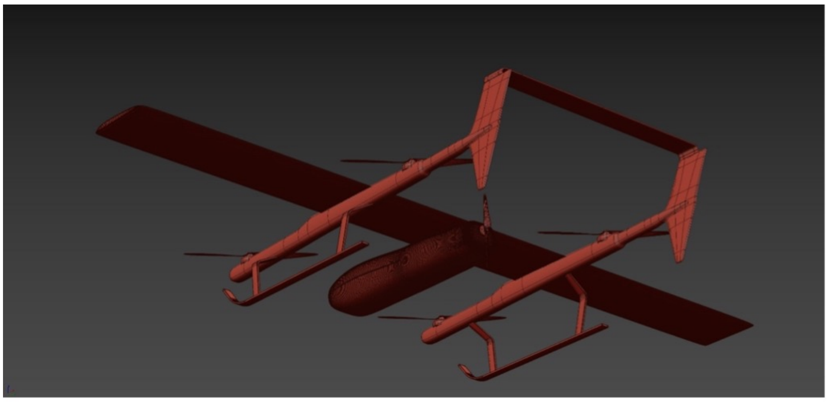


Figure 2. MUGIN SLT UAS Fixed-wing Configuration

2.3. GIS location data

The objective is to detail the implementation process of real GIS tagged imagery data into a game engine at runtime and the optimisation of the imagery loading engine that is directly linked to the visual performance of the simulation engine.

We started by modelling Cranfield airport and then continued to model the city of Bath. Effectively, we can model any other area as required by customers. Initial testing of high-fidelity data was in a smaller portion of airspace.

2.3.1. Airport Modelling using GIS

An area has been created of 50 miles square centered on Cranfield Airfield that has been initially modelled. The model included a terrain horizon of 400ft and 25 miles either side of Cranfield Airfield, which was shown to provide a representative perspective when viewing the FUSE environment landscape from the point of view of a frustum located on the SLT UAV that it initially modelled. This area of 50 miles square was modelled in higher detail, with the rest of the country outside of the 50 miles square box utilizing generic Google Earth data.

The specification of the modelled area (Figure 3.) are:

- A geographic area of 50 miles square.
- Centered upon Cranfield airport.
- With high-resolution ground sensing distance of 4 cm for the area bonded in red.
- Medium resolution ground sensing distance of 25 miles for the remainder of the 50 miles square.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

Figure 3. The area agreed for aerial survey

Furthermore, other GIS data that was imported into FUSE simulator to further enhance the urban environment includes:

- Public rights of way to show outdoor recreation routes.
- Settlement Boundaries to show where councils define urban areas.
- Ordnance Survey base mapping.
- Sites Locations (such as schools, churches, cemeteries, etc.) to show potential limiting factors for flights.
- Local Authority boundaries, including parishes and wards to show responsibility areas

2.3.2. Flight Survey of Cranfield Airport

A flight was conducted to capture imagery of Cranfield airport, using the Cesium ION platform (a robust, scalable, and secure high fidelity imagery platform for 3D geospatial data [18]). The Cranfield survey data has been ingested into FUSE visualisation engine (the Unreal engine version of the Deadstick Gaming Model). to provide the high-resolution imagery of the red bonded area in Figure 3. This is shown in Figure 4.

Figure 4. Cranfield Airport Survey Image

 doi:10.20944/preprints202205.0354.v1

2.3.3. GIS Landscape Optimization

This work item provided a means to load all the city of Bath's geographic information and imagery at run time into the Unity-based FUSE environment. With so much high-fidelity data it is difficult for any game system to effectively render the environment and do justice to the detail whilst enabling enough of the landscape to be rendered to keep in focus on the detail when looking out at the horizon.

Whilst it is trivial to get the simulated drones to fly over interfaces such as Google Earth which presents reasonable detail when viewed from high above, when you lower the viewport down to street level the aerial google earth imagery shows itself to lack in fidelity, which is why Google supplements it with "street view" imagery.

2.3.4. UTM Interface

To integrate UAS into the finite volume of airspace safely and efficiently, UAS must be able to detect and be detected using available and recognised electronic conspicuity technology, particularly if operating Beyond Visual Line of Sight (BVLOS) in non-segregated airspace [19].

A key component of electronic conspicuity systems is the Automatic Dependent Surveillance Broadcast (ADS-B) system. ADS-B provides continuous broadcasts of aircraft position, identity, velocity and other information over unencrypted data links [20].

The FUSE simulator can schedule synthetic flights within the environment and output simulated ADS-B streams such that a UTM system may be tested with regards to its ability to process mass simulated UAS. The challenge was on how to facilitate the data exchange between the synthetic environment and the real world, i.e., how the data from the simulated UAS's may be inputted into a UTM system and how potentially the simulator might receive https://global.gotowebinar.com/join/1556260175638299152/169443037 data from UTM system and import this into the synthetic environment in real-time. Since there are many UTM systems with heterogeneous interfaces the objective is ensuring that FUSE is interoperable with the other systems, and this can be achieved by ensuring that its data output follows a harmonised standard.

FUSE works by having a flight scheduler determine the take-off, flight path and landing points of multiple UAS's in simulated real-time. The data that these UAS's export as they are flying needs to be achieved in an internationally recognised format.

In line with the Civil Aviation Authority's (CAA) guidance on electronic conspicuity for UAS CAP722 [19] and CAA CAP1391 for electronic conspicuity devices [21], the FUSE simulator will output ADS-B signals for the simulated UAS movements within the virtual environment. The simulated ADS-B signals will take the form shown in Figure 5 and Figure 6.

```
+----------+----------+-------------+------------------------+-----------+
|  DF (5)  |  CA (3)  |  ICAO (24)  |         ME (56)        |  PI (24)  |
+----------+----------+-------------+------------------------+-----------+
```

Figure 5. ADS-B Information

Bit	No. bits	Abbreviation	Information
1-5	5	DF	Downlink Format
6-8	3	CA	Transponder capability
9-32	24	ICAO	ICAO aircraft address
33-88	56	ME	Message, extended squitter
(33-37)	(5)	(TC)	(Type code)
89-112	24	PI	Parity/Interrogator ID

Figure 6. Structure of the ADS-B frame

A review was conducted of other potential future systems in order to ensure that the FUSE simulator has the ability to also simulate UAS movement and export the Eurocontrol Specification for Surveillance Data Exchange – ASTERIX Part 29 Category 129 UAS Identification Reports [22]. ASTERIX UAS Identification messages are used to transmit an identification code for the UAS together with its position and will take the form in Figure 7.

Data Item Reference Number	Description	Resolution
I129/010	Data Source Identification	N.A.
I129/015	Data Destination Identification	N.A.
I129/020	UAS Manufacturer Identifier	N.A.
I129/030	UAS Model Identifier	N.A.
I129/040	UAS Serial Number	N.A.
I129/050	UAS Office Registration Country	N.A.
I129/070	Time of Day	1/128 s
I129/080	Position in WGS-84 Coordinates	2 cm
I129/090	Altitude above Mean Sea Level	0.1 m
I129/100	Altitude above Ground Level	0.1 m
I129/110	GNSS Signal Accuracy	1m
I129/120	Operational Risk Levels	N.A.

Figure 7. ASTERIX UAS Identification messages

Whilst the above two forms will form the outputs of the FUSE system in terms of UAS reporting data, the system will also has the ability to receive the live ADS-B stream data relevant to the geographic location that the FUSE operator is working with.

Figure 8. FUSE simulator showing Mugin UAS flying over Google Earth terrain with ADS-B Information in White

2.4. Weather Services Liaison

The FUSE simulator can represent all classes of weather and now can access the Copernicus dataset [23], such that FUSE has the potential to access any UK weather for any location, and represent with a good level of accuracy based on the historical records of the last twenty years.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

3. Software Architecture

As it was already mentioned, the FUSE environment is based on the deadstick immersive flight simulator. The following modules comprise the software system.

Module Name	Code item
Drone	Drone performance and dynamics Drone flight controller
Routing and Navigation	Route creation and loading Route visualization Drone navigation controller
Terrain Synthesis	Elevation data + aerial imagery Scanned data + 3d building data
Environment	Atmospheric rendering Time of day Volumetric clouds Weather effects
Data Acquisition	Real-time weather - METAR's Traffic - ADSB NOTAMS Airspace Headings Drone Controller Observer VR

Table 1. Software System Modules.

3.1. Backend development

The Backend development builds on the route planning algorithms and builds on the development of the Electric Airspace GIS portal [24]. The details of that environment are described in the following sections.

3.1.1. Building the drone operating environment

The flight engine has been designed to allow users to build up a detailed representation of the environment in which drone flights will be taking place. Before scheduling a flight, the following information must be loaded into the engine:

- Details of the sites a drone will be visiting along its route, including geographical location of any landing pads on site.
- Information on the types of drones in operation, including the flight performance and handling characteristics of each, and their battery capacity and range on a full charge.
- A register of drones that will be flying, linked back to their type of information, which helps define the types of the journey they can undertake, and weather conditions they can fly in.
- A map of UK airspace restrictions, both permanent and temporary via e.g., NOTAMs, which are used by the routing algorithm to ensure drones do not infringe on controlled airspace [25].

All information about the operating environment is stored in the flight engine's Structured Query Language (SQL) server database and is automatically used by the routing algorithm when scheduling new flights. At present, the database includes a map of UK airspace restrictions from the OpenAirspace data set [26]. This will be upgraded to the full ICAO airspace dataset [27].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

3.1.2. Flight scheduling

Once the information above has been loaded into the flight engine, new drone flights can be planned and scheduled by providing a few basic pieces of information to the Application Programming Interface (API) schedule endpoint, as follows:

- A list of the site landing pads that must be visited along a route.
- The ID of the drone, which will be flying the route.
- The anticipated take-off time of the drone when leaving its starting site.
- Optionally, information on the drone endurance (if it is partially charged) and required layover time at any stops en-route to the final destination.

The flight engine will then generate an optimised route for the drone, avoiding controlled airspace and including stops at each site en-route. If required, it can also provide a comparative road route, and generate statistics on carbon and time savings of using a drone versus an internal combustion engines or electric road vehicle. This routing information can be accessed from the API route endpoints.

3.1.3. Weather module

Incorporating weather information into the routing and scheduling algorithm is important to understand how trends in local and synoptic weather conditions will affect drone flight performance and their ability to operate on given days at different times of the year. Our initial study in this area has consisted of profiling the weather conditions over the Morecambe Bay region to verify how often the conditions are favourable for drone flights to take place. This has been done to lend support to provide FUSE with yet another real-world use case. This demonstrates FUSE's capabilities to model the historic weather and when linked to the FUSE visualisation engine, will allow the synthetic environment to model the historical average weather, be that on a per month request, per date request or average weather across the year.

3.1.4. Flight Engine Architecture and Hosting

The drone flight engine is written in .NET5 and presented as a secure cloud-hosted REST API, which is currently accessible at [28]. The system is connected to a SQL Server database, which stores information about the drone operating environment as well as a complete record of scheduled flights and their associated routing information. API endpoints are being developed to integrate the flight engine with the XMap system from GeoXphere [29] and the VR modelling environment provided by REMEX [30].

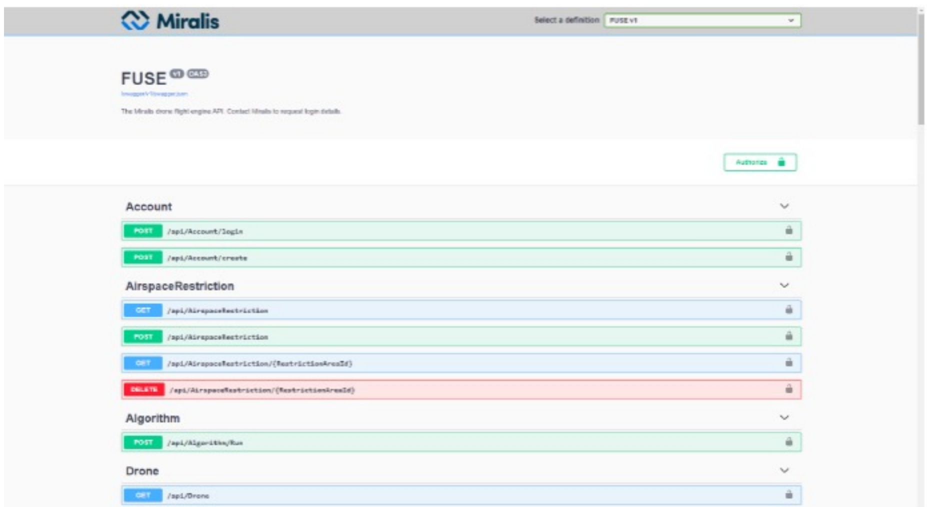


Figure 9. Flight Engine REST API

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

The system is currently hosted in the Microsoft Azure and supported by a Continuous Integration / Continuous Delivery (CI/CD) pipeline hosted on Azure DevOps, which manages automated code testing, deployment configuration, and publishing of new code as it becomes ready for release.

However, complexities in publishing from .NET to Amazon Web Services (AWS) have so far resulted in difficulties around securing the API (particularly HTTPS encryption of data in transit), and reliably connecting the flight engine to its SQL database. We are working with Amazon to understand these issues and develop a solution that will integrate with the used CI/CD pipelines and allow reliable updates to the API as development continues.

3.1.5. Flight Engine Database

The flight engine database (Figure 10) contains the following components:

- Site: Premises that contain one or more landing pads. Includes mean historical wind data. Lat/Long for site entrance on e.g., SatNav. Windspeed in knots and direction in degrees.
- Pad: Where the drones actually land. Lat/Long is for the centre of the landing pad itself. Altitude is in metres Above Mean Sea Level (AMSL).
- WeatherForecast: Up-to-date weather forecast containing parameters which may affect drone flight performance on a given route. Can be used to flag problems with a scheduled take-off of a given drone. Windspeed in knots, direction in degrees, temperature in Celsius, pressure in h Pa, visibility in m.
- DroneModel: The type of drone that may be flying a route. The model includes flight parameters that will affect a drone's ability to carry out a scheduled flight under different atmospheric conditions.
- Drone: The ID details of the individual drone of a given type, which will fly a route.
- Waypoint and GroundWaypoint: Waypoints on a route. Seconds from start is in seconds from take-off time in Schedule. Height in metres AMSL.
- Schedule: A scheduled drone take-off on a given route. This uses multiple records to build a timetable an includes take-off parameters that may affect flight performance (needs review). Endurance in minutes. Weight in kilograms (kg).
- AirspaceRestriction and AirspaceRestrictionVertex: Store the airspace restriction geometry used by the routing algorithm to avoid airspace that a drone cannot fly into.
- DeconflictionLeg: Stores pairs of legs that come within 50 m of each other in latitude, longitude and altitude that are e used for route deconfliction logic. The WayPointindex for each route is the start index of the leg. The end index will be this $+1$.
- RouteConflicts: Full X, Y, Z, T coordinates of conflict start and end, where drone separation is < 50 m between two scheduled fights. Stored explicitly for rapid visualisation of the conflict in VR. MinSeparation specifies how close the two drones will come at closest approach.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

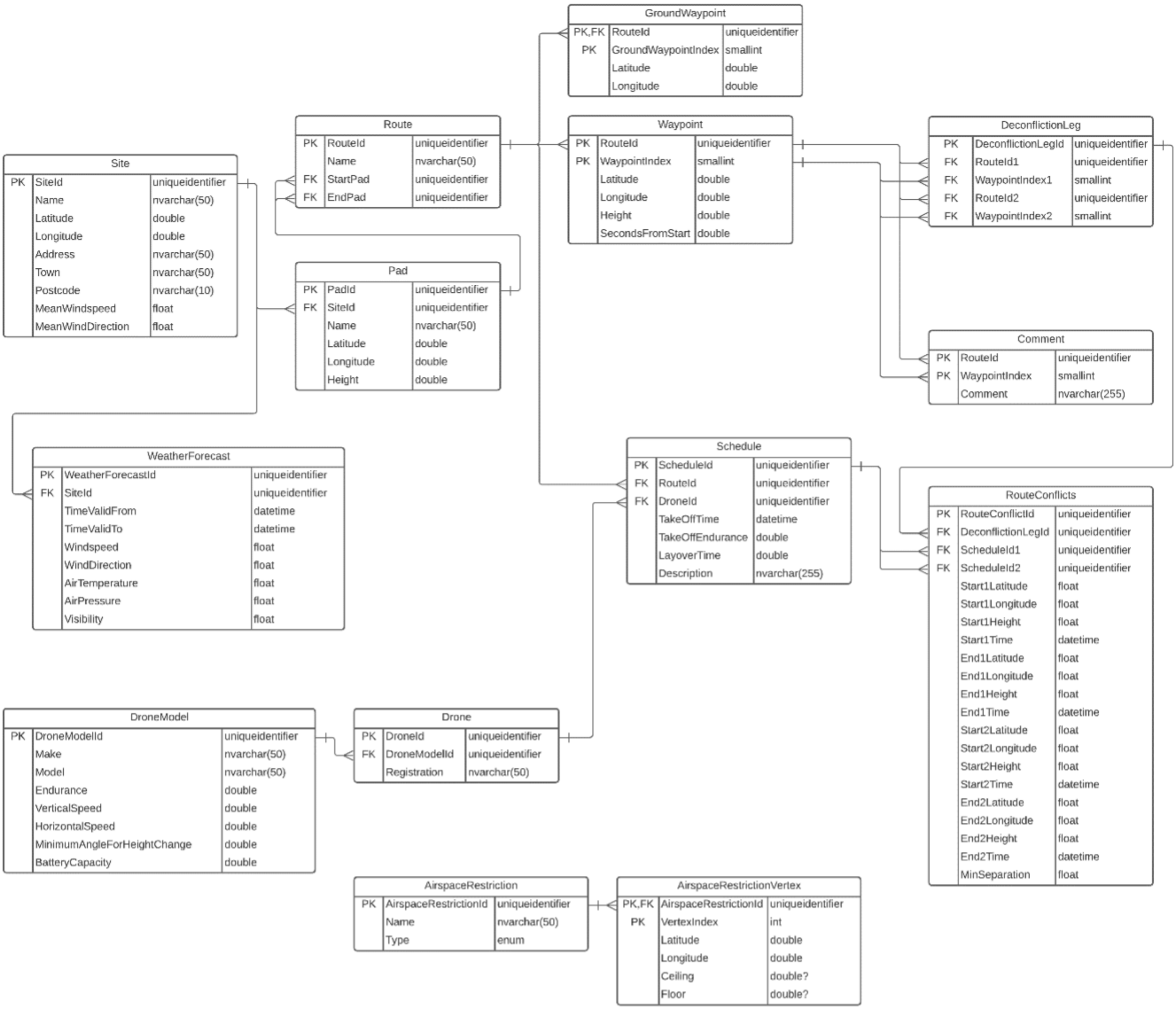


Figure 10. Flight Engine Database Schema

3.1.6. Redeveloping the FUSE Visualisation Engine

This was an issue on both a pipeline level (trying to identify what tools/formats could be used to process the data into a usable format) and at runtime level, where a custom streaming solution would be required to read the data, process it and deliver it at varying different Level of Detail (LOD) chunks so to be performant and fit in memory.

Upon further investigation, it became apparent that such a runtime solution for dealing with complex 3D geospatial content had already been implemented in the form of 3D tiles and in a manner that would allow GeoXphere to easily push and process data in a format that was runtime ready via the Cesium ION platform.

An implementation was provided in Unreal and after some analysis, it quickly became apparent that the cost to port the existing FUSE functionality was vastly outweighed by the benefit in risk removed by the need to generate complex bespoke tooling and streaming solutions. Subsequently, the decision was made to switch to Unreal engine and the existing functionality has been ported to ensure 1:1 functionality.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

3.1.7. The porting to Unreal from Unity

Unity and Unreal are very different with regard to their architecture. The former using the Mono libraries to take advantage of C# and .NET to lower the barrier for development and iteration, the latter, using C++ to provide lower-level access and, with it, performance. The port, therefore, presented many potential risks as there would be a need to rewrite much of the functionality required for FUSE. Those risks were outweighed, however, by the functionality that was provided by Cesium ION platform with regards to the terrain streaming, GIS support and the pipelines that would allow to seamlessly import scanned data for the city of Bath and Cranfield. Much time was dedicated in the development schedule to the creation of such functionality within Unity, it was quickly realised however that the same time could be used to port the functionality developed for FUSE, which due to its modular nature and known quantity would be a much simpler job and lower risk. This was further solidified after early tests with Cesium in Unreal demonstrated that it was possible to import the scanned data with promising results. The port, whilst not insignificant, went very smoothly.

3.1.8. Data export to Deadstick Gaming Model

Using the Cesium ION platform, the Cranfield survey data has been ingested into FUSE visualisation engine (the Unreal engine version of the Deadstick Gaming Model) to provide the high-fidelity terrain and building data for the area around Cranfield Airport as depicted in Figure 10.

4. Use Cases

FUSE has created a synthetic environment for both Bath and Cranfield. The Cranfield environment has been optimized with detailed modelling applied, whereas the Bath model has not yet but will be in the future. The Cranfield model has been used for test and validation and port to Bath as we were working with that local authority and with Bath being a national heritage site, it is a good test case for Local Authorities drone regulations.

4.1. Design of Synthetic Version of the City of Bath

The city of Bath was chosen because of the accessibility of its high-resolution photogrammetry data through GeoXphere for the region and the Digital Elevation Model (DEM) model was populated with previously acquired photogrammetry data. Figure 11 details FUSE's ability to model drone operations over the urban landscape.

The deployment of FUSE working on a VR headset (Oculus Rift) was well-received at DroneX with significant interest regarding the ability of the synthetic environment to represent the effect that drones will have on the skies above Bath. This was the original concept behind the FUSE project.

4.2. Design of Synthetic Version of Cranfield Airport

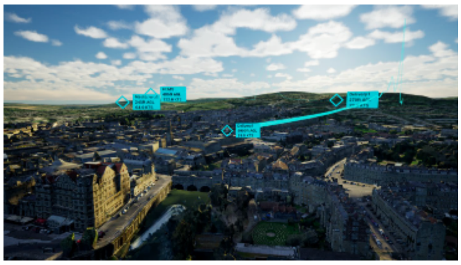
Using the photogrammetry data, a high-resolution model of the airfield has been created. It was also possible to fly drones synthetically around the airfield as well as being able to recreate historic flights from ADS-B data. As can be seen in Figure 12, a pretty accurate synthetic version of Cranfield airport and the old control tower was created. It has been able to deploy drones into the circuit synthetically and to realise the historic flight paths of aircraft within FUSE.

4.3. Review of Data from FUSE Simulator

The 12 months' worth of historic ADS-B data for the Cranfield area was provided by Selborne Technologies [31]. The data has been extracted and rendered into Google Earth historic ADS-B tracks. Working with a different dataset from the same historic ADS-B source, it has been able to visualise all aircraft (equipped with ADS-B or Secondary Surveillance Radar (SSR) Transponder) passing across a similar geographic area as it is shown in Figure 13.

 doi:10.20944/preprints202205.0354.v1

(**a**) Multiple drones overhead the city



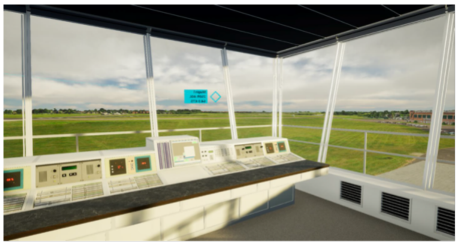
(**b**) Users position themselves anywhere in Bath and witness the effect of the drones overhead

(**c**) FUSE enables lateral separation distances to be established even allowing for deliveries

(**d**) Audio modelling of the drones so the user can hear the effects of the drone deliveries.

Figure 11. Synthetic Version of the City of Bath

(**a**) Approaching Cranfield towards one of the disused runways.



(**b**) The view from the tower.

(**c**) The Track of the RPAS overhead the main runway

(**d**) The RPAS holding at the entry to the NBEC with fading light

Figure 12. Synthetic Version of Cranfield Airport

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 May 2022 doi:10.20944/preprints202205.0354.v1

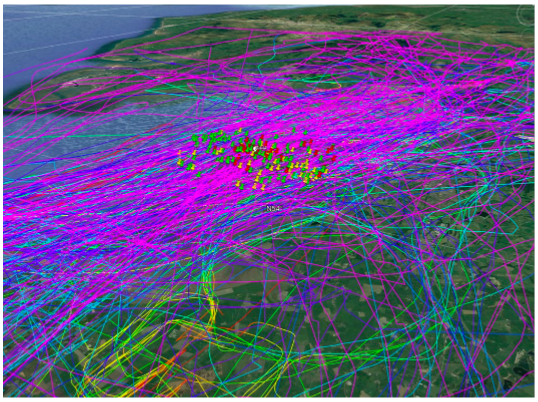


Figure 13. FUSE visualisation of historic ADS-B tracks

Being able to visualise past aviation traffic for any region gives FUSE huge potential with regards to the modelling of airspace for utilisation by drone or electric Air Taxi usage. The software developed to do this enables FUSE to provide historic usage analysis for airspace in support of Airspace Change Processes, which will have significant relevance for the planning of Urban Airspace.

The export of the processed historic ADS-B data was done in a manner where it could be imported into the FUSE synthetic environment in Unreal engine. This was a significant piece of work as it meant that FUSE needed to be able to read in a database of other flight information as opposed to reading in the live ADS-B stream that was previously presented.

Various historic flights were singled out for rendering and were implemented for testing into the FUSE engine, to validate that it can access the historic data. This was done through a separate hard coded interface, rather then try and modify the Miralis FUSE Rest API flight planning interface.

5. Conclusions

In this paper the design of the Future Urban Synthetic Environment (FUSE) was presented. This is the first time a gaming engine has used this level of photogrammetry data to provide an environment within which drone operations may be modelled. FUSE can be used anywhere where photogrammetry data of this fidelity exists to augment existing digital elevation and terrain models, which enables FUSE to be a significant tool for any Local Authorities looking to understand how urban drone operations may affect their local populations. FUSE has also the ability to recreate aircraft movements from historic data, so for drone operations near active airfields it was able to model the historic aircraft saturations of airspace and how commercial drone operations in the local vicinity may affect the airfield or maybe affected by the airfields' historic and planned operations.

However, FUSE needs to bring together more of the elements that have been created at runtime and do this in more elegant ways. It can recreate all manner of variables and combinations of variables within the synthetic environment, but this needs time to bug fix, and to make the data exchange more fluid between software elements. The 2D GIS flight planning interface, that calls the routing algorithm, needs a more eloquent way of communicating with the 3D FUSE interface. Opportunities exist within the gaming market to provide a gamified drone planning tool which could end up as a game title in its own right.

As an outlook, it is aimed to take the developed system and build it specifically for Local Authorities enabling them to plan ahead to support drones in their local urban airspace. To do this there is a need to create a Bylaw that publishes its impacts through FUSE to UTM systems, such that operators may know the local desired flight rate over specific urban densities. Furthermore, it is planned to operate real flights at Cranfield airport to validate the FUSE system in real conditions. FUSE has also the potential to model UTM data in the future when more established datasets become publicly available. One of the simplest next steps would be also to export weather models that have been created into a UTM system such that historic analysis, plus live weather data may be analysed together for example for incident investigation purposes.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, C.M. and C.Ch; validation, T.S., D.P. and A.T.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, T.S.; writing—review and editing, T.S, D.P, C.Cr. and A.A. ; visualization, X.X.; supervision, X.X.; project administration, C.Cr.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not Applicable.

1. Leicester. https://www.leicester.gov.uk/leisure-and-culture/parks-and-open-spaces/using-our-parks/flying-drones-and-model-aircraft/usage-policy/. Accessed: 2021-12-14.
2. Chen, J.Y.; Haas, E.C.; Barnes, M.J. Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2007**, *37*, 1231–1245.
3. Vempati, A.S.; Khurana, H.; Kabelka, V.; Flueckiger, S.; Siegwart, R.; Beardsley, P. A virtual reality interface for an autonomous spray painting UAV. *IEEE Robotics and Automation Letters* **2019**, *4*, 2870–2877.
4. Sanders, A. *An introduction to Unreal engine 4*; CRC Press, 2016.
5. unreal. https://cesium.com/platform/cesium-ion//. Accessed: 2021-12-14.
6. unity engine. https://unity.com/. Accessed: 2021-12-14.
7. Haas, J. A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE* **2014**.
8. Roldán, J.J.; Peña-Tapia, E.; Garzón-Ramos, D.; de León, J.; Garzón, M.; del Cerro, J.; Barrientos, A. Multi-robot systems, virtual reality and ros: Developing a new generation of operator interfaces. In *Robot Operating System (ROS)*; Springer, 2019; pp. 29–64.
9. Lan, G.; Sun, J.; Li, C.; Ou, Z.; Luo, Z.; Liang, J.; Hao, Q. Development of UAV based virtual reality systems. 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). IEEE, 2016, pp. 481–486.
10. Zhi-Hua, Q.; Yi-Bo, L.; Shao-Peng, K.; Qiong, Z. Design of UAV telepresence and simulation platform based on VR. 2008 International Conference on Cyberworlds. IEEE, 2008, pp. 520–524.
11. Liu, H.; Bi, Z.; Dai, J.; Yu, Y.; Shi, Y. UAV simulation flight training system. 2018 International Conference on Virtual Reality and Visualization (ICVRV). IEEE, 2018, pp. 150–151.
12. Kim, S.J.; Jeong, Y.; Park, S.; Ryu, K.; Oh, G. A survey of drone use for entertainment and AVR (augmented and virtual reality). In *Augmented reality and virtual reality*; Springer, 2018; pp. 339–352.
13. Mangina, E.; O'Keeffe, E.; Eyerman, J.; Goodman, L. Drones for live streaming of visuals for people with limited mobility. 2016 22nd International Conference on Virtual System & Multimedia (VSMM). IEEE, 2016, pp. 1–6.
14. Nguyen, V.T.; Jung, K.; Dang, T. Dronevr: A web virtual reality simulator for drone operator. 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR). IEEE, 2019, pp. 257–2575.
15. Roadmap Sustainable Aviation. http://deadsticksimulator.com/press/sheet.php?p=deadstick_bush_flight_simulator. Accessed: 2021-12-14.
16. mugine. https://www.muginuav.com/. Accessed: 2021-12-14.

17. Dewi, P.T.; Hadi, G.S.; Kusnaedi, M.R.; Budiyarto, A.; Budiyono, A. Design of separate lift and thrust hybrid unmanned aerial vehicle. *Journal of Instrumentation, Automation, and Systems* **2015**, *2*, 45–51.
18. Cesium ion plateform. https://cesium.com/platform/cesium-ion/. Accessed: 2021-12-14.
19. Authority, U.C.A. Unmanned Aircraft System Operations in UKAirspace – Guidance CAP722. Technical report, CAA, 2021.
20. Part, C. Automatic Dependent Surveillance—Broadcast (ADS–B) Out Performance Requirements to Support Air Traffic Control (ATC) Service. *Final Rule* **91**, *91*.
21. Authority, U.C.A. Electronic conspicuity devices CAP 1391. Technical report, CAA, 2021.
22. EUROCONTROL Specification for Surveillance Data Exchange ASTERIX Part 29 Category 129: UAS Identification Reports. Technical report, Eurocontrol, 2018.
23. Copernicus dataset. https://cds.climate.copernicus.eu/#!/home. Accessed: 2021-12-14.
24. Alectric Airspace. http://www.electric-airspace.com/. Accessed: 2021-12-14.
25. NOTAM. https://www.notams.faa.gov/dinsQueryWeb/. Accessed: 2021-12-14.
26. OpenAirspace dataset. https://www.openaip.net/. Accessed: 2021-12-14.
27. ICAO dataset. https://applications.icao.int/dataservices/default.aspx. Accessed: 2021-12-14.
28. miralis engine. https://fusedrone.azurewebsites.net/index.html. Accessed: 2021-12-14.
29. XMap. https://xmap.geoxphere.com/cloud-gis. Accessed: 2021-12-14.
30. REMEX. http://deadsticksimulator.com/press/index.php. Accessed: 2021-12-14.
31. selbroneTech. https://find-and-update.company-information.service.gov.uk/company/08487793. Accessed: 2021-12-14.