University of Glasgow | School of Computing Science

A Prototype System for Simulating the Risks of Sub-Orbital Space Flight for Commercial Aviation

Marco Sarconi

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 22, 2013

Abstract

Over the coming years there is expected to be an increase in the number of sub-orbital space flights for various purposes such as space tourism and scientific research. It is therefore advisable to analyse the potential risks of sub-orbital space travel on commercial aviation. This report presents research into the risks of sub-orbital debris on commercial aviation as well as the development of a simulation system which integrates live flight data through Automatic Dependent Surveillance-Broadcast and meteorological data. It then uses a mathematical model to calculate a debris field for a given suborbital vehicle and displays a model of the potential impact on aircraft within the vicinity.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: ____________________ Signature: ______________________

Acknowledgments

I would like to thank Prof. Chris Johnson for his continued support and guidance throughout the year. I would also like to thank the participants who took part in the evaluation.

Contents

1 Introduction and Background Research **1**

1.1 Aims and Motivation 1

1.2 Sub-Orbital Space Flight 2

1.2.1 What is Sub-Orbital Space Flight 2

1.2.2 Space Tourism 3

1.2.3 Point to Point Travel 4

1.2.4 Research Payloads 4

1.3 Safety Issues 5

1.4 Case Study: Space Shuttle Columbia Disaster 7

1.4.1 The Disaster 7

1.4.2 The Investigation 8

1.5 Debris 9

1.5.1 Orbital Debris 9

1.5.2 Sub-Orbital Debris 10

1.5.3 Weight 10

1.5.4 Aerodynamic Drag 10

1.5.5 Lift 11

1.5.6 Newton's second law of motion 11

1.5.7 Trajectory 11

1.5.8 Wind Force 12

1.6 Debris Modelling 13

1.6.1 DEBRA 13

1.6.2 TAP Debris Model 14
1.6.3 CRTF 15
1.7 ADS-B 16
1.7.1 What is ADS-B 16
1.7.2 How does ADS-B work 16
1.7.3 Advantages of ADS-B 17
1.7.4 Relevance 17
1.8 Related Work 18
1.8.1 Shuttle Hazard Area to Aircraft Calculator 18
1.8.2 Future Air Traffic Management Concepts Evaluation Tool 19
1.8.3 Debris Dispersion Model Using Java 3D 20
1.9 Risk 20
1.9.1 What is Risk 20
1.10 Risk Perception 21
1.10.1 Psychometric Paradigm 21
1.10.2 Cultural Theory 21
1.10.3 Dread Events 22

2 System Proposal **24**
2.0.4 High Level Concept 24
2.0.5 Stakeholders 24
2.0.6 Functional Requirements 25
2.0.7 Non-Functional Requirements 25
2.0.8 System Scope 26
2.0.9 System Actors and Use Cases 27

3 Design **28**
3.0.10 Coding Tools 28
3.1 Web Services 30
3.1.1 Airspace Data 30

3.1.2 Map Data . . . 31
3.1.3 Meteorologic Data . . . 32
3.2 Debris Model . . . 32
3.3 Visualisation and GUI . . . 33
3.3.1 Visualisation . . . 33
3.3.2 GUI . . . 33
3.4 System Architecture . . . 34
3.4.1 Debris Model Class . . . 34
3.4.2 Aircraft Class . . . 35

4 Implementation **36**
4.1 Scene Graph . . . 36
4.2 States . . . 37
4.3 Debris Calculations . . . 37
4.4 Map Tiles . . . 37
4.5 Map projection . . . 38
4.6 Collision Detection . . . 39

5 Evaluation **40**
5.1 Experiment Design . . . 40
5.1.1 High Level Aim . . . 40
5.1.2 Hypothesis . . . 40
5.1.3 Experimental Procedure . . . 41
5.2 Analysis Method . . . 42
5.2.1 Parametric Tests and Non-Parametric Tests . . . 42
5.2.2 Conclusion . . . 43
5.2.3 Wilcoxon signed-rank test . . . 43
5.2.4 Results . . . 44
5.3 Overall Conclusion . . . 45
5.4 Other Findings . . . 45

5.5 Validation 46
5.5.1 Debris Field Validation 46
5.5.2 System validation 46
5.6 Usability Evaluation 46
5.6.1 Retrospective think aloud 46
5.6.2 Neilsens Heuristics 47

6 Conclusion **49**
6.1 Future Work 49
6.2 Conclusion 50

A Evaluation Participant Pack **51**

B Data used in evaluation and verification **52**

C User Guide **53**

List of Figures

1.1 A sub-orbital trajectory is shown in red [40] 3

1.2 Technical Snapshot of Virgin Galactic's SpaceShipTwo[25] 3

1.3 A model of the SpaceLiner[4] 4

1.4 X-34 re-entry Fault Tree[39] 5

1.5 Penetration Consequence Analysis Results and Logic.[51] 6

1.6 The Columbia debris field in East Texas spread over 2,000 square miles[11] 7

1.7 Recovered debris in the Kennedy Space Center[11] 8

1.8 Comparison between space debris 1960 and 2000[2] 9

1.9 Drag and weight influencing a piece of debris[12] 11

1.10 The Influence of the Ballistic Coefficient, β, and Wind upon Debris Impact Points[12] 12

1.11 Contributions to debris dispersion models[12] 13

1.12 An example of the graphical output from DeBRA showing the debris and population data[3] . . 14

1.13 ADS-B data flow[23] 16

1.14 SHAAC[36] 18

1.15 FACET visualisation of the US airspace[7] 19

1.16 The Debris Dispersion Model in Java3dD[12] 20

1.17 A diagram showing group grid theory [35] 22

1.18 Relations between judged frequency and the actual number of deaths per year for 41 causes of death.[43] 23

2.1 A use case diagram showing the interactions between the system and actors 27

3.1 A QOC diagram for programming language choice 29

3.2 NASA WorldWind Imagery 31

3.3 Planned location of the map tiles 33
3.4 The system sceneGraph 34
3.5 The Debris Model Class and related classes 34
3.6 Aircraft Class 35

4.1 The system sceneGraph 36

5.1 Bell curve of standard normal distribution 42
5.2 Rating of elements affecting risk perception pre and post system use 45

6.1 An example of the system's visual output 50

List of Tables

2.1 Functional Requirements 26

Chapter 1

Introduction and Background Research

This chapter presents an introduction to the project and a background review which includes: A summary of sub-orbital space travel and its potential risks. It also gives a summary of debris mechanics, an understanding of which is necessary for the design and implementation. The chapter also presents an outline of Automatic Dependant Surveillance-Broadcast(ADS-B) standard. A review of various debris models and similar systems are also presented. Finally as the system is aimed to be a tool for risk analysis, the concepts of risk and risk perception are discussed.

1.1 Aims and Motivation

Aims

The aim of this project is to create a system which can be used to simulate the impact an uncontrolled sub-orbital space vehicle re-entry could have on conventional aviation.

The system can aid regulating bodies, air traffic controllers, commercial space companies, and researchers in understanding the risk an uncontrolled re-entry could have on the airspace. The system also aims to be a re-usable application which is not tied to one particular Reusable Launch Vehicle (RLV) or a specific spaceport.

Research has been conducted in order to better understand the risks RLVs pose to the airspace as well as current contingency measures. Research has also been carried out to better understand the mechanics of debris dispersal and air traffic control. This was necessary in order to model these elements within the system.

Motivations

In the past fifty years there has been around 4,500 space flights, when contrasted to the one million aircraft built in the first fifty years of conventional aviation it becomes clear that the number of space launches has been limited.[8] Launch levels remained low for a number of reasons; spaceflight is riskier than conventional aviation as it is not possible to gradually increase and test performance levels as can be done with traditional aviation, the materials required to build space rockets were expensive and the expertise required was vast.[8] Therefore in the past space flight was limited to world superpowers such as the United States and the Soviet Union.

However space flight has now been achieved by many commercial companies, particularly sub-orbital space flight, and is no longer solely the domain of Government agencies. These companies plan to conduct numerous launches over the coming years.[6] As the industry develops it is extremely likely that a marked rise in space launches worldwide will occur.

This increase in space launches will increase the potential for a space related disaster. Of particular concern after the Columbia Disaster[8] is the risk falling debris could pose to traditional aviation. The increase in space launches could mean that existing safety measures and contingency plans are no longer viable.

It is therefore important that the risks sub-orbital travel places on aviation can be assessed; as such a system which can simulate the potential impact would be greatly desirable. Further detail on the motivations is provided in the next chapter.

1.2 Sub-Orbital Space Flight

This section defines sub-orbital space flight and outlines some sub-orbital space flight proposals.

1.2.1 What is Sub-Orbital Space Flight

In order define sub-orbital space flight it is helpful to first define orbital space flight.

Orbital Space Flight

An orbital space flight is one in which the space vehicle reaches an altitude of above 100 kilometers above sea level- known as the Karman line - and is traveling at a sufficient velocity such that it reaches “orbital speed”. If both these conditions are met and the vehicle maintains its orbital speed then it will begin to orbit the Earth.[38]

Sub-Orbital Space Flight

Sub-orbital flights have the same altitude requirements as orbital flights as such they cross the karman line and are officially deemed to be in space. However they do not reach a high enough velocity to enter orbit. As such after engine shutdown the sub-orbital vehicle falls back to Earth and begins its re-entry.[40]

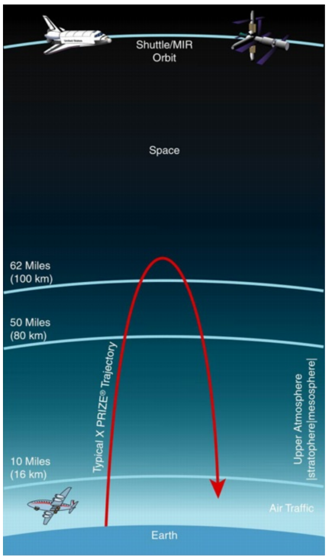

Figure 1.1: A sub-orbital trajectory is shown in red [40]

1.2.2 Space Tourism

A sub-orbital space operation is significantly cheaper than an orbital space flight. This has resulted in numerous commercial companies laying out plans for taking civilians into space as part of a commercial venture. These launches generally involve ’space tourists’ spending a few minutes in space and experiencing zero gravity before returning back to Earth. At the forefront of space tourism, and perhaps the most public, is Virgin Galactic.[25] Virgin Galactic’s SpaceShipTwo is based on the XPrize winning SpaceShipOne design and can carry six passengers into space. At the time of writing no sub-orbital space tourism flight has been launched, however Virgin Galactic is expected to launch its first official flight this year (2013).

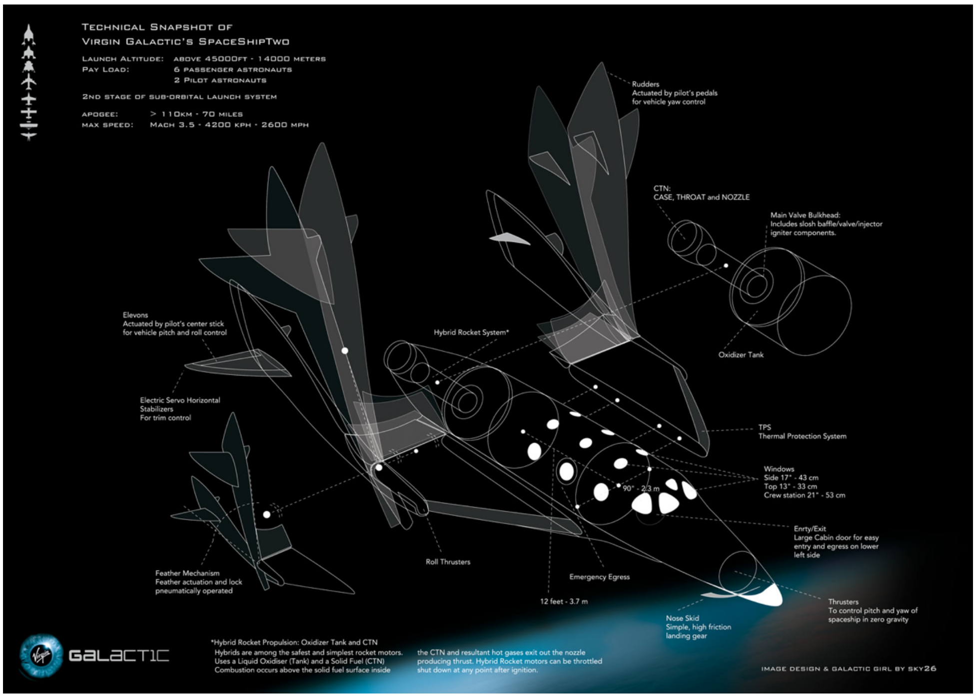

Figure 1.2: Technical Snapshot of Virgin Galactic’s SpaceShipTwo[25]

1.2.3 Point to Point Travel

There are proposals for utilising sub-orbital vehicles for passenger transportation, similar to conventional aviation. One such vehicle proposed by DLR is the SpaceLiner, the SpaceLiner is an:

> ”ultrafast intercontinental passenger transport based on a rocket powered two stage vehicle.”[42]

The concept is being researched by DLR as part of The European Space Agency’s FAST20XX (Future high-Altitude high-Speed Transport 20XX) research project. The SpaceLiner would be able to travel from the UK to Australia in approximately 90 minutes by completing part of its trajectory in space. Research began in 2005 and the SpaceLiner could be completed by 2030. Currently there are difficulties in reducing the risk of engine failure to acceptable levels for mass transportation.[49]

Figure 1.3: A model of the SpaceLiner[4]

Point to point sub-orbital vehicles are expected to launch from large international airports or at dedicated nearby spaceports in order to be commercially viable and convenient for passengers. Therefore they will be subject to rigours safety standards inline with conventional aircraft.

1.2.4 Research Payloads

Sub-Orbital RLVs are already being utilised for research and scientific purposes. RLVs are capable of carrying a research payload, essentially cargo in the form of an experiment. This allows researchers to subject experiments to the conditions found during the launch and re-entry of a sub-orbital flight, as well as the harsh conditions of space. For example XCOR’s lynx RLV can support:

- Cockpit experiments
- Externally mounted experiments
- Test pilot/astronaut training
- Upper atmospheric sampling
- Microsatellite launch / ballistic trajectory research[5]

1.3 Safety Issues

Space flight is inherently risky. Space vehicles are complex systems which are subject to extremely harsh conditions as a result they currently fail between 2-5 percent of the time.[8] The technology and experience can still be described as in its infancy when compared to conventional aviation.

Until recent years spaceflight has been the domain of publicly funded bodies such as NASA and the ESA. This limited the number of space launches, however recent developments have lead to an increase in commercial space flight companies. Many of these companies have released plans for sub-orbital space flight at various space ports throughout the world including the commercial US spaceports, spaceport Sweden, and proposed spaceports in the United Kingdom and Malaysia.[53]

The increase in commercial space launches, particularly sub-orbital space launches, intensifies a number of potential risks. Of particular interest is the risk to the airspace as a result of the uncontrolled re-entry of a RLV.

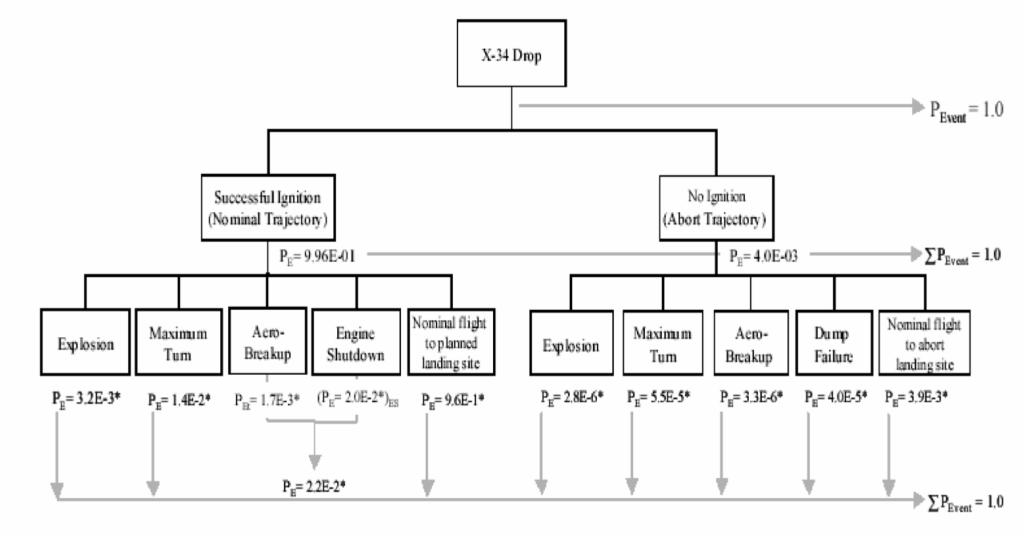

Figure 1.4: X-34 re-entry Fault Tree[39]

The figure above shows the fault tree and risk analysis for the X-34 RLV. The X-34 is a sub-orbital RLV designed for experimentation and test flights. The Fault tree displays the ten possible failure modes of the X-34 during re-entry.

If a RLV breaks-up on re-entry the resulting debris has the potential to impact aircraft within the airspace, which could result in a catastrophic accident. The Federal Aviation Administration(FAA) has stated that an impact with a debris fragment of 300 grams or above is expected to cause the destruction of a commercial aircraft. [51]

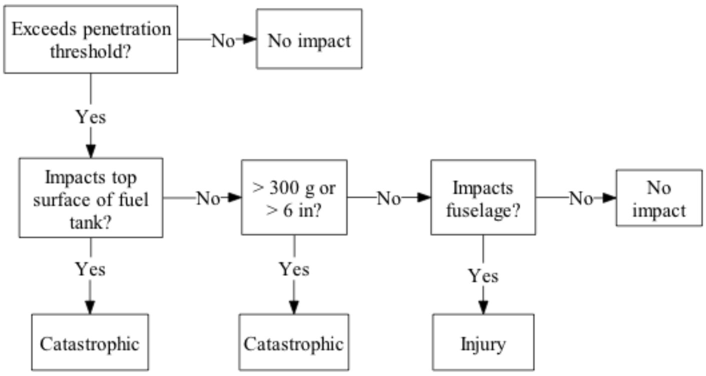

Figure 1.5: Penetration Consequence Analysis Results and Logic.[51]

The figure above is obtained from the FAA analysis. It shows that an impact piece of debris greater than 300 grams will result in a catastrophic disaster irrelevant of the impact location. It also shows that a smaller debris fragment may also cause a catastrophic disaster if it impacts a particularly vulnerable area of the aircraft, such as the top of the fuel tank.

As is expected there are measures in place to protect aircraft during the launch and re-entry of space vehicles[36]:

- Special Use Airspace(SUA): Many RLV launch sites operate within protected airspace. This means aircraft are permanently prohibited from entering this area of airspace.
- Temporary Flight Restrictions(TFR): If the launch takes place in an area of non-restricted airspace then a temporary flight restriction can be imposed during the launch and re-entry of the RLV.
- NOTAM: During the launch or re-entry of a RLV, notice to airmen (NOTAM) these alert pilots and air traffic controllers of the operation being conducted and the boundaries of the required airspace. On receipt of the notice aircraft whose flight path would normally intersect the SUA or TFR must take an alternate route.

These methods have proven effective for ensuring safe separation distance of a rocket and other aircraft in past shuttle launches. However as the number of RLV launches increases over the coming years restricting large areas of airspace for each launch and re-entry may become unfeasible. Furthermore, the safe separation distances and buffer zone are calculated based on the potential debris field for each vehicle. As commercial companies design and manufacture various different RLV, it will become more difficult to manage the airspace.[29]

1.4 Case Study: Space Shuttle Columbia Disaster

1.4.1 The Disaster

On the 1st of February 2003 the NASA space shuttle Columbia disintegrated as it began its re-entry into the Earth's atmosphere. As Columbia executed its re-entry the wing leading edge temperature rapidly increased to abnormal levels. At approximately 231,600 feet and traveling at Mach 23 the Columbia shuttle started to break-up. Spectators on the ground were able to identify signs of falling debris however at this point Mission Control believed the re-entry phase to be normal. Columbia began to fully disintegrate as it crossed Texas, breaking into more than 84,000 pieces which were dispersed across 2000 square miles within Western Louisiana and Texas.[10]

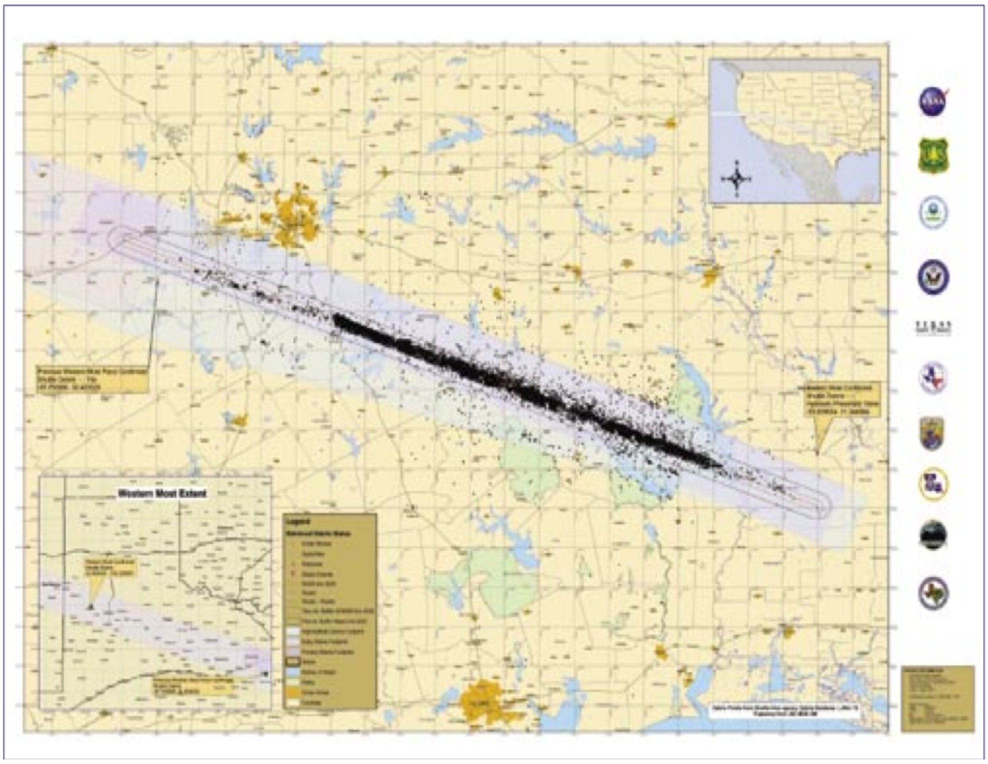

Figure 1.6: The Columbia debris field in East Texas spread over 2,000 square miles[11]

The figure above shows a map of the shuttle's debris field. The debris is sparse just after the shuttle enters Texas, as the shuttle continues to disintegrate rapidly significantly more debris falls. The debris field created closely emulates the shuttle's final flight path, creating an elongated ellipse. The resulting total debris weighed over 84,900 pounds, 38 percent of Columbia's dry mass, the remaining 62 percent likely burned up during re-entry or was not found.[12] Given the total weight and number of pieces the average weight of a fragment was 1 pound or 450 grams. As discussed in the previous section, this is significant because a debris fragment of 300 grams or above is expected to cause the destruction of a commercial aircraft.[51]

Figure 1.7: Recovered debris in the Kennedy Space Center[11]

1.4.2 The Investigation

The Columbia disaster was the first time in which a NASA space shuttle had disintegrated during the re-entry phase of the mission. The only previous disaster, the challenger disaster, occurred during the launch phase. [1]As such, the shuttle disintegration prompted an investigation into several safety and management issues that were raised as a result of the disaster. A major concern outlined in the Columbia Accident Investigation Board(CAIB)report is the potential risk falling debris poses to the population on the ground and to other aircraft in the airspace. [9]Although the Columbia shuttle debris did not impact any person on the ground or any aircraft within the airspace it was unknown if this result would be typical. As such The CAIB carried out an investigation with the aim :

> "To confirm whether the lack of casualties is the expected consequence, or whether this happened to be good fortune."[12]

The analysis concluded that the lack of ground population casualties was the expected result of the Columbia disaster but, there was a reasonable probability (less than 0.5 but greater than 0.05) that casualties could potentially have occurred. The investigation also carried out a preliminary analysis on the risk of the Columbia debris striking traditional aviation. It found the worst case probability of debris impact to be 0.08. The report concluded that the probability of debris-aircraft impact were higher than would be allowed for unrestricted aircraft operations. The conclusion also noted that:

> "A more detailed aircraft risk analysis should be performed using the actual records of aircraft activity at the time of the accident."[12]

The Columbia disaster highlights the potential risk of space vehicle re-entry to air traffic. As noted in the CAIB report this risk is not fully explored in risk analysis. This suggest that further research is required.

1.5 Debris

In the context of this paper debris is used to refer to space debris. Space debris are discarded or destroyed man made objects or, fragments of objects which are in Earth Orbit or re-entering into the Earth's atmosphere.

1.5.1 Orbital Debris

There are known to be around 319,000 debris fragments in orbit around Earth.[2] These fragments are made up discarded satellites, debris caused by space collisions and spent rocket bodies. The quantity of debris in orbit around earth has increased significantly since the early days of space flight. The figure below shows tracked orbital debris in 1950 and tracked orbital debris in 2000. This dramatic increase in debris presents some potential problems to current space operations:

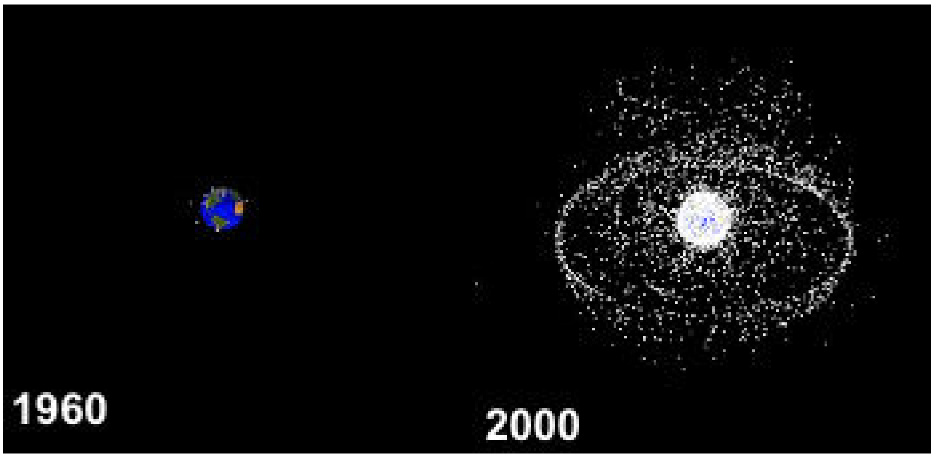

Figure 1.8: Comparison between space debris 1960 and 2000[2]

Risk to space vehicles

Space vehicles which are in Earth orbit are at risk of colliding with the aforementioned space debris. The result of such a collision is dependent on the size of the debris. unmanned space vehicles such may be subject to constant collision with extremely small debris fragments, these collisions have little affect as sensitive areas of the vehicle are strategically positioned away from the probable impact direction. However a collision with a larger debris fragment, larger than 10 cm is predicted render the vehicle in-operational. Furthermore a collision involving a space vehicle and a debris fragment of larger than 10 cm is expected to cause a catastrophic break-up.[33]

This occurred in 2009 when a non-operational satellite- Cosmos 2251(debris)- collided with the Iridium 33. This resulted in the destruction of both vehicles. As the destruction occurred in orbit, this created more orbital debris which further intensifies the risk to operational space vehicles.[31]

Risk of re-entry

Debris fragments may re-enter the Earth's atmosphere through orbital decay or a controlled re-entry. Debris with a low melting point will burn-up during re-entry. However if the debris fragment has a high melting point-for example titanium- it is likely to survive re-entry and impact the Earth's surface.[37] This presents a potential risk to both the ground population and aviation.

This risk is intensified if the debris is large. One extremely large debris fragment (100m squared) re-enter every several years.[32] These are seen as high risk as they produce many fragments during break-up which can survive re-entry. This occurred in 2001 when the Russian space station Mir performed a controlled re-entry to Earth. Fragments which did not burn up fell into the Pacific Ocean.[32]

As a result of the potential risks, space debris is closely Monitored by NASA's Orbital Debris Program Office and the European Space Agency's space debris office. Tools have also been developed to analyse the risk and trajectory of re-entering space debris, such as NASA's Object Re-entry Survival Analysis Tool.[34]

1.5.2 Sub-Orbital Debris

The re-entry of orbital debris can be considered sub-orbital debris once the debris fragment is no longer in orbit. Sub-orbital debris can also refer to debris which is created in sub-orbit (i.e the Earth's atmosphere) for example through the destruction of an aircraft. This paper is mainly concerned with sub-orbital debris. As such it necessary to gain an understanding of the mechanics which affect falling debris. A piece of falling debris is subject to aerodynamic and gravitational forces .

1.5.3 Weight

Weight is a force created by gravitational attraction of the Earth on any object. It can be defined as the object's mass times the gravitational acceleration.[28]

$$w = m * g \tag{1.1}$$

As the objects distance from Earth's surface increases the gravitational acceleration g decreases. Therefore the weight of the object decreases with respect to altitude.

1.5.4 Aerodynamic Drag

The motion of the falling debris fragment is opposed by aerodynamic drag. Drag is a mechanical force which is generated by the difference in velocity between the solid debris fragment and the gaseous atmosphere. The equation below can be used to calculate drag.[46]

$$F_D = \frac{pv^2 C_D A}{2} \tag{1.2}$$

where
F_D is the drag force.
p is the mass density of the gas or fluid.
v is the velocity of the object relative to the gas or fluid.
A is the reference area.
C_D is the drag coefficient.

The reference area is generally taken as the frontal area of the fragment, the area which is perpendicular to the flow direction.[26] As demonstrated in the formula the drag is dependent on the size of this area. The larger the reference area, the grater the drag. Therefore a small fragment of debris will be subject to less drag than a larger piece.

The drag coefficient is a variable used to characterise the the dependencies of shape, flow conditions, and inclination that affect drag.[28]

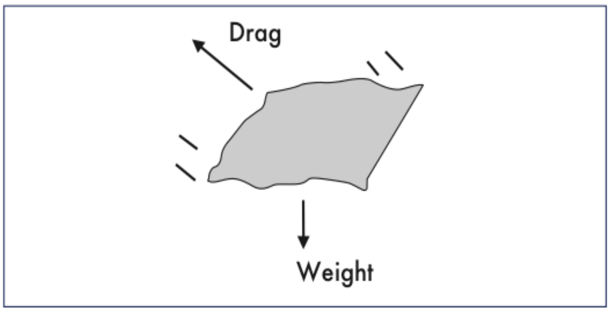

Figure 1.9: Drag and weight influencing a piece of debris[12]

1.5.5 Lift

Falling debris are also subject to lift. Lift is a force which occurs when a moving flow of gas is turned by a solid object. Following Newton's third law of motion the flow is turned in one direction and lift is generated in the opposite direction. Lift is perpendicular to the debris' direction of motion. The equation for calculating liftF_L is shown below.[46]

$$F_L = \frac{C_L A p v^2}{2} \tag{1.3}$$

where
C_L is the lift Coefficient.
A is the surface area of the debris
p is the air density.
v is the velocity of the debris.

The lift force generated by a piece of debris is directly dependent on the debris' shape. For example; a flat piece of debris, such as a space vehicle wing will generate a larger lift force when compared to a cube shaped piece of debris. A larger lift force will result in the debris having a slower velocity. Therefore the wing will fall slower than the cube.

1.5.6 Newton's second law of motion

$$f = m * a \tag{1.4}$$

Newton's second law of motion states that the force f of a moving object with a constant mass is equal to the object's mass m times the object's acceleration a.[28] As such the acceleration of a debris fragment can be calculated as the net external force over the fragment's mass. By calculating values for the forces outlined above Newton's second law can be used to calculate the debris' acceleration.

$$a = f/m \tag{1.5}$$

1.5.7 Trajectory

A a debris fragment has an initial state vector which is represented by a position and velocity vector. In the context of debris resulting from an explosion this initial state vector can be altered by the explosion which imparts a velocity and modifies the velocity vector.

"There is no adjustment to the initial position because the velocity is added impulsively."[12]

Ballistic Coefficient

The trajectory for a piece of falling debris is mainly defined by its Ballistic Coefficient(β). The equation for calculating β can be expressed as:

$$\beta = \frac{W}{C_D A} \tag{1.6}$$

where W is the weight of the debris fragment, C_D is the drag coefficient and A is the representative area used to calculate the drag coefficient(see 1.2)[28]

The ballistic coefficient represents the weight to drag ratio. Objects with a low β fall slowly where as objects with a high β will fall faster. Furthermore, the trajectory of objects with a low β will be affected to a far greater extent by wind velocity.[13, **?**]

1.5.8 Wind Force

Wind is another force which affects falling debris. As discussed above wind velocity can potentially alter the debris' trajectory. The affect of wind velocity on debris is proportional to the debris ballistic coefficient. As such wind velocity will not result in substantial displacement of debris with a large β. Conversely debris with a low β will be moved more as a result of the wind velocity.[**?**] The figure below illustrates the effect of wind and $/beta$

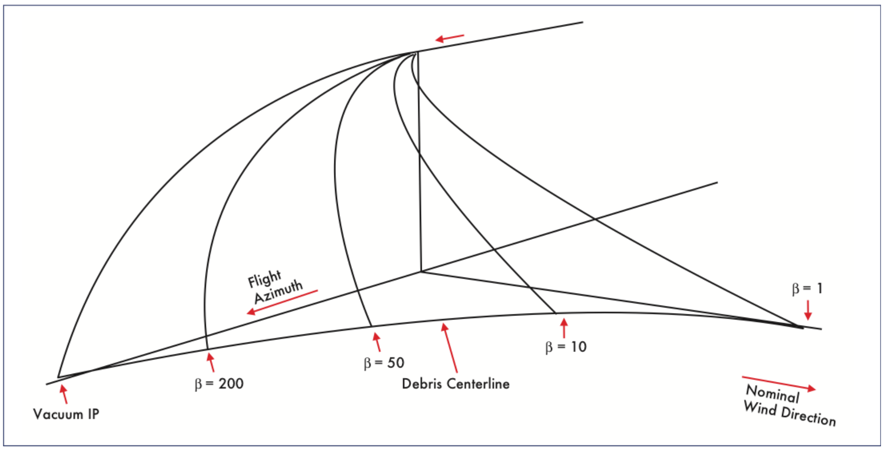

Figure 1.10: The Influence of the Ballistic Coefficient, β, and Wind upon Debris Impact Points[12]

The label "Vacuum IP" in the figure describes the impact point of the debris when there is no atmosphere.

The figure compares the impact points of debris with various β falling in a vacuum, to the same debris falling in a more realistic atmosphere with a nominal wind speed and direction. It clearly shows that debris with high β under the wind force fall close to their impact point in the absence of wind and debris with a low β can be seen to fall much further from their impact point in a vacuum.

1.6 Debris Modelling

> ”A debris model is a theoretical model that calculates the motion, impact locations and areas, and the probabilities and risks associated with debris falling within a finite area” (Van Suetendael, 2003)[50].

A debris model is a mathematical model which attempts to model the uncertainties and underlying mechanics which act on falling debris, outlined in the previous section).

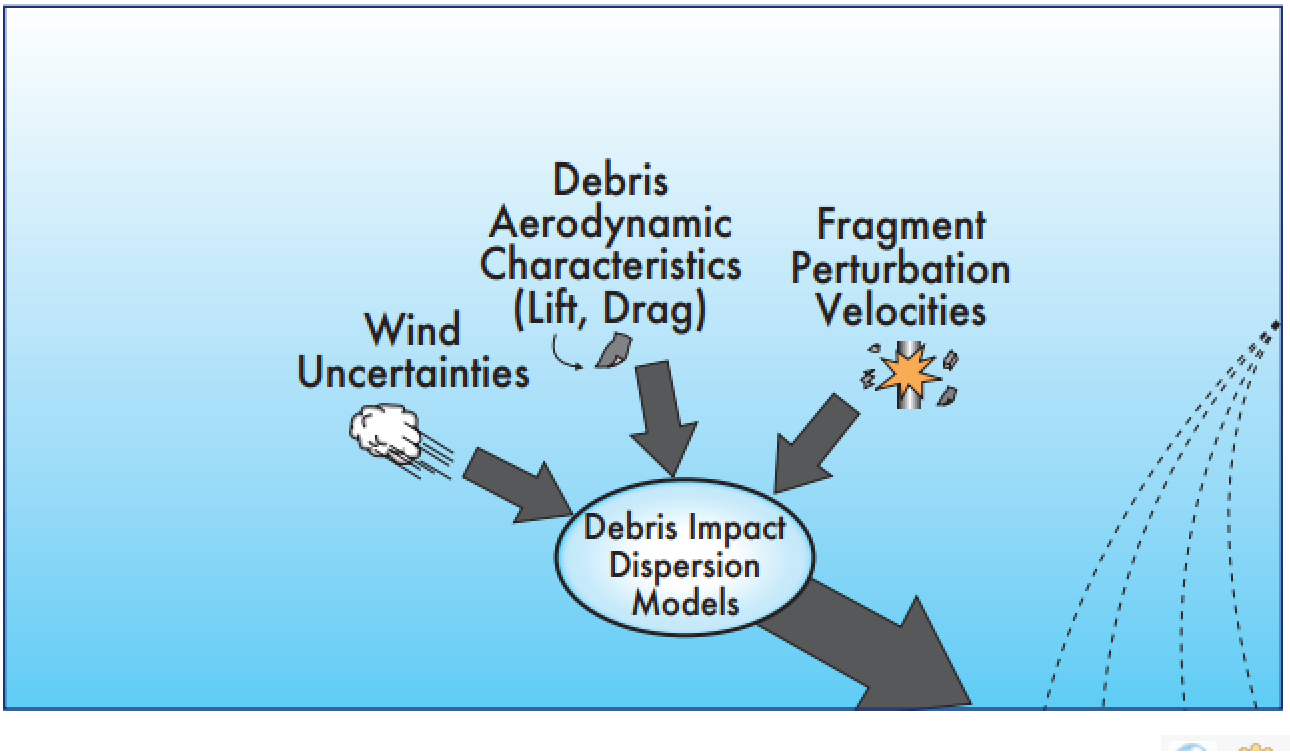

Figure 1.11: Contributions to debris dispersion models[12]

The figure above illustrates the different forces simulated within a debris model. Some debris models that were considered for integration into the system are outlined in the following sections.

1.6.1 DEBRA

Debris Risk Assessment(DeBRA) is a debris mode and risk assessment tool developed by APT Research.[3] It asses the risk of RLV failure modes. It uses user input defining a nominal RLV trajectory and the failure mode information to calculate the debris footprint. The model can calculate the debris footprint based on multiple RLV failure modes including: explosions/breakups, engine shutdown failures and malfunction turns. DeBRA then overlays the calculated footprint on a population map in order to calculate the risks to the ground population.

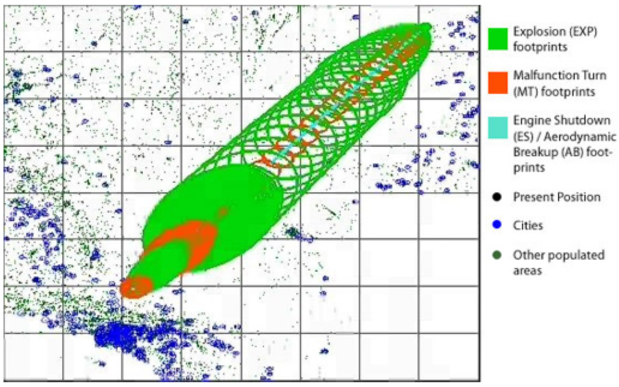

Figure 1.12: An example of the graphical output from DeBRA showing the debris and population data[3]

The potential impact points for each failure mode of the RLV are shown along with the corresponding debris footprint. Unfortunately as DeBRA is a piece of commercial software its source code and debris model is unavailable for use. Furthermore the algorithms used to create the debris model are closely guarded so are unavailable for reference or implementation.

1.6.2 TAP Debris Model

TAP(Trajectory Analysis Program) is a freely available mathematical model- created by H. Oldham for:

> ”accurately predicting the debris scatter distance of an in-flight airframe separation”[30]

The TAP model requires the following inputs:

- Initial altitude of disintegration
- Initial density altitude
- Altitude of impact at ground level
- Wind velocity and direction
- Horizontal true airspeed at disintegration
- Rate of climb or sink at disintegration
- Weight of projectile
- Projectile drag coefficient
- Projectile frontal area

The model then applies the gravitational and aerodynamic forces affecting the debris to output the following:

- Horizontal distance from disintegration at impact
- Horizontal, vertical, and total velocities

- Terminal velocity
- Flight-path angle at impact
- Ground speed of projectile at impact and x and z components of that velocity

The debris model was created for air show environments, as such it will not be entirely accurate for modelling sub-orbital debris however, it does model the mechanics of debris outlined in the previous section. Furthermore, it was successfully used to model aircraft ground impact situations (Jazen N, 2010). As such the TAP model could provide a reasonable guide to debris dispersion if a more accurate model is unavailable.

1.6.3 CRTF

CRTF(Common-Real-Time Footprint) is a probabilistic debris model, It is designed to calculate the debris dispersion caused by an instantaneous vehicle breakup. It can calculate the debris dispersion instantaneously in real-time using the initial state vector or alternatively can be integrated into a risk assessment program which inputs "A large number of state vectors describing all of the potential accident/failure conditions along with their corresponding probabilities".[12]

CRTF is able to model various uncertainties such as the shape and mass of the various debris fragments, the atmospheric conditions such as wind speed, and direction and the point of vehicle break-up. Monte Carlo is used to model these uncertainties and to create the impact distributions.[12] The CRTF is currently utilised by NASA co-ordinated with the FAA(Federal Aviation Administration to calculate safe flying distances during the launch and re-entry of RLVs. As the CRTF model is designed specifically for simulating the debris field of a RLV it would be an excellent candidate for integration into the system however, as is typical of propriety software the algorithms are not released publicly therefore, the CRTF model can not be integrated into the system.

1.7 ADS-B

1.7.1 What is ADS-B

ADS-B(Automatic Dependent Surveillance Broadcast) is a data communications protocol used to obtain an aircraft's position within the airspace.

- Automatic: Velocity and position data are automatically transmitted a minimum of once every second.
- Dependent: The transmission is reliant on the on-board equipment GPS receiver and transmitter functioning correctly.
- Surveillance: surveillance data- position, velocity and other flight data are surveillance data.
- Broadcast: The data is broadcast to any ADS-B receiver.[52]

1.7.2 How does ADS-B work

Aircraft can obtain their latitude. longitude and altitude via standard GPS(Global Positioning System) technology. The aircraft's position and altitude can then be transmitted using ADS-B. Collectively this data can be used to uniquely locate the aircraft at any point on the Earth. The aircraft is also able to transmit its velocity and heading. Given the above data the air traffic controllers can safely navigate the planes in the airspace with far more accuracy and reliability than radar.[47]

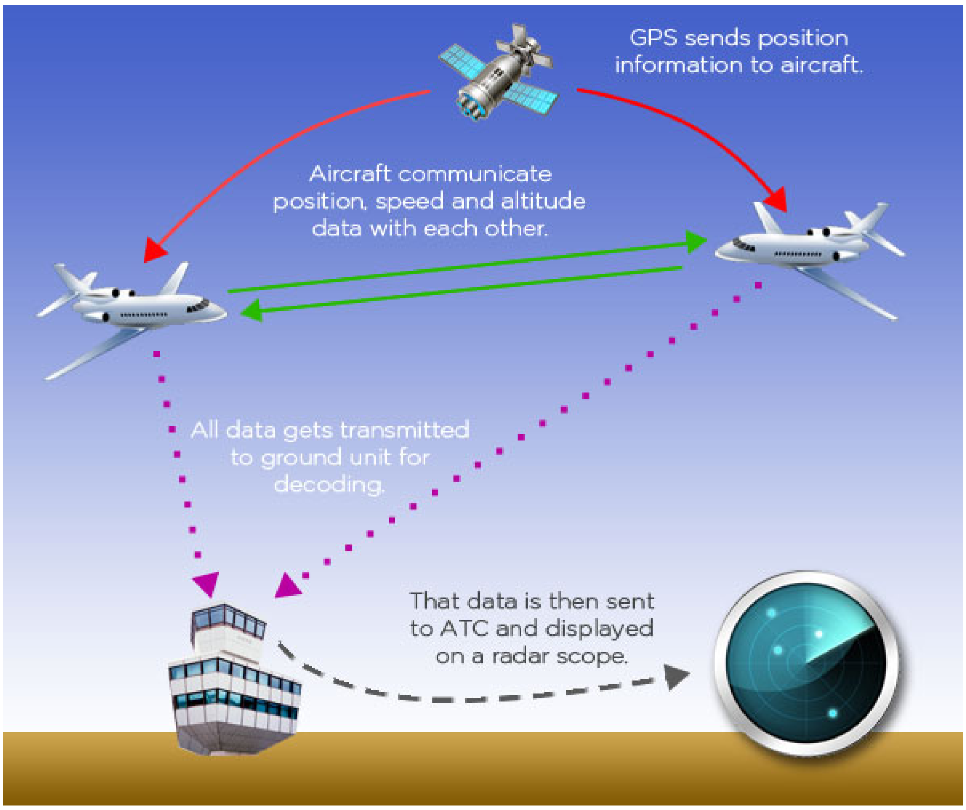

Figure 1.13: ADS-B data flow[23]

The figure above shows the transmission of ADS-B data between aircraft and base stations. ADS-B equipped aircraft broadcasts the data which can be received by any entity that has a receiver. The base station receives these signals and transmits them to air traffic control. An ADS-B equipped aircraft can also receive transmissions from other aircraft, this allows the aircraft's collision detection system to alert the pilot in the event that another aircraft enters its alert zone.

ADS-B allows for a far more detailed picture of the airspace for both the aircraft and the ATC(air traffic controller).

1.7.3 Advantages of ADS-B

It has a number of advantages over the traditional Radar system:

- ADS-B signals do no degrade as range increases. The signal would simply become unavailable if the range was too great.
- ADS-B equipped aircraft are able to view the same compete image of the airspace as the ATC.
- ADS-B signals are not as sensitive as radar to atmospheric and weather conditions[19]

As such ADS-B is an integral part of the United States NextGen airspace transformation and the Single European Sky, its European counter Part. [22]

1.7.4 Relevance

Access to ADS-B data would allow the system to accurately represent the real-time positions of aircraft within a specified area of airspace. With ADS-B data the system could feasibly represent the aircraft within any specified area of the globe. Therefor the system could run simulations on real-time airspace data, as opposed to statistical data such as average airspace density, which has been used in previous simulations[12]

1.8 Related Work

The following section examines existing software systems which are used for the risk analysis of debris dispersal and/or simulating the airspace.

1.8.1 Shuttle Hazard Area to Aircraft Calculator

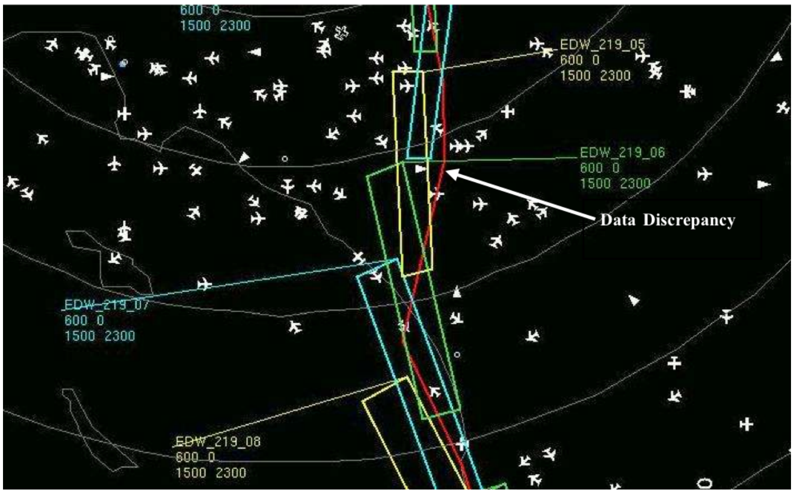

Figure 1.14: SHAAC[36]

The Shuttle Hazard Area to Aircraft Calculator(SHAAC) is a system created by the FAA to simulate a Shuttle reentry accident and analyse the impact this would have on the airspace.[36] The system is able to model the debris footprint of the NASA orbiter shuttle, for each state vector it is given as input. This is then displayed as the hazard area. The hazard area is the area which shows the extent of airspace that may be impacted by falling debris. This is then overlaid on the airspace control displays and can clearly show which aircraft may be affected by falling debris.

The system is capable of running in two modes; A real time mode and a planning mode. Real time mode is designed to be used in the event of an uncontrolled re-entry (similar to the Columbia Disaster) to attempt to asses the level of impact and rapidly re-direct aircraft from the out of the hazard area.

In planning mode the system can take multiple potential shuttle state vectors as input and producing a file of multiple potential hazard areas as the output. This method is currently used by ATC centres to gain an initial idea of the potentially impacted airspace during shuttle re-entries.

Although the software is extremely capable of predicting the debris field of the NASA shuttle, given that it utilises NASA's Common Real Time Footprint, it does not predict debris dispersal for other RLVs. Additionally it outputs a projected hazard area however it does not offer any kind of non visual output such as: potential impacts.[36]

1.8.2 Future Air Traffic Management Concepts Evaluation Tool

the Future Air Traffic Management Concepts Evaluation Tool (FACET) is a software system created at NASA's aviation systems division. It is capable of visualising thousands of aircraft trajectories across the United States.[7]

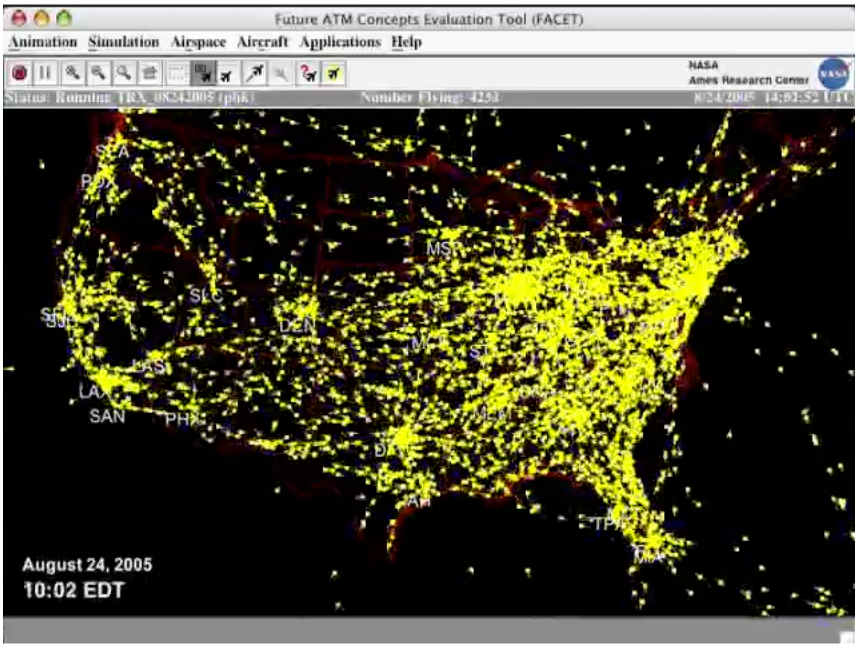

Figure 1.15: FACET visualisation of the US airspace[7]

It is designed to provide an environment for researchers to test air traffic management concepts and provides advance support for:

- Airspace models
- Weather data
- Flight schedules
- Climb,cruise and descent trajectories
- Different aircraft types

The FACET system is incredibly powerful and is capable of simulating a full day's dynamic airspace operations. It can model uncertantiy and measure the results of different air traffic management decisions. Furthermore, it is capable of providing its results in graphical form.[7]

FACET is capable of simulating an extremely accurate model of the airspace. However it does not model debris, hazard areas or impacts. If access to the system is possible FACET could provide the ideal airspace model which could be integrated with a debris model.

1.8.3 Debris Dispersion Model Using Java 3D

The Debris Dispersion Model Using Java 3d is a visualisation tool created by Thirumalainambi and Bardina, NASA.[48] The tool is designed to model the dispersion of debris created during a space shuttle if there is a failure resulting in an explosion. The predicted dispersion is then used to evaluate safe range distances for launch.

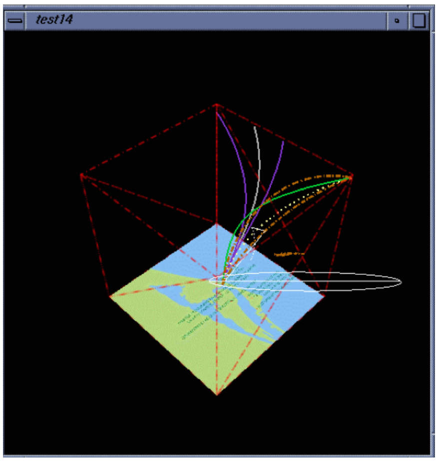

Figure 1.16: The Debris Dispersion Model in Java3dD[12]

The debris model has an integrated weather model which uses real time gathered data and decision making capabilities to estimate the weather conditions and their effects at the launch site.

The tool is designed to provide Range Safety Officers with a system for simulating failure and debris dispersion at shuttle launch time. This information can then be used to ensure a safe operating difference for the shuttle.

1.9 Risk

1.9.1 What is Risk

Risk is a concept which is used to characterise the probability of a negative event such as an injury or loss. Risk is the subject of much research as it is not a one dimensional concept. Kaplan and Garrick(1981) defined risk as three questions:

- What can go wrong? This Explores the different possible failures or undesirable outcomes of an activity for example crashing a car
- How likely is it to go wrong? This attempts to define the probability of the undesirable activity being realised
- What are the consequences?If the undesirable outcome is realised, what will the effects be and how sever will they be? This can also be described as the expected loss. This is not always a loss of life, it could be a loss of finances for example.

By answering these three questions an overall idea of the risk can be considered. Considering only one of these questions will give only a single dimension of the risk. In order to fully comprehend risk all three should be considered.

Objective Risk

If the probability of a negative event is defined by scientific means the measure is known as the objective risk. For example: The objective risk of vaccinations causing illness could be calculated in a sample set, by recording the number of illnesses as a result of the vaccine and the number of people who were unaffected. [14]

Subjective Risk

If the probability is defined in a non-scientific manner it is known as the subjective risk. Using the same example as above: The subjective risk of vaccinations causing illness could be the patients estimation of how likely the vaccine is to cause illness.[14]

1.10 Risk Perception

Risk perception is the subjective risk people associate with a negative event.[45] Research has produced numerous ways of measuring risk perception.

1.10.1 Psychometric Paradigm

The psychometric paradigm attempts to determine the factors which influence risk perception. It divides risk perception into nine elements such as, severity of consequences, control over risk, and newness of the risk. These elements can then be measured using psychometric tests. Participants can rate each of these elements for a particular hazard. The results can the be analysed to give an idea of the test group's risk perception of the hazard.[24]

Fischhoff et al 1978 conducted a study on the use of psychometrics for risk perception of different activities, to elicit quantitative judgments of perceived risk, acceptable risk and perceived benefit.

The study asked participants to rate each activity against a scale for each of the nine identified elements which affect risk perception. The study then uses multivariate analysis and 'psychophysical scaling, analysis techniques to produce quantitative risk perception results. This ability to gather quantitative results is one of the advantages of the psychometric paradigm. It is a well recognised method for quantitatively judging risk perception.[41][44]

1.10.2 Cultural Theory

Cultural Theory is a sociological approach to risk perception as opposed to the psychological approach of the 'Psychometric paradigm'. the Cultural Theory of risk perception presents the theory that risk perception is tied to social learning. The theory proposes that risk perception is 'selected' by a person to reinforce and maintain social links.[21] People can then be catagorised by their social group and the extent to which they are governed by rules. This is known as Group Grid.[35]

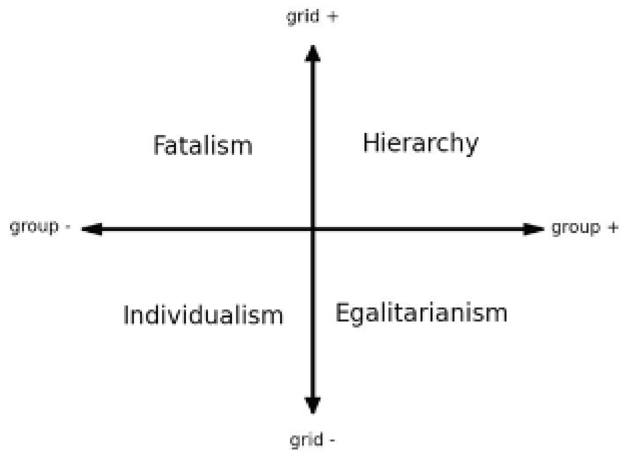

Figure 1.17: A diagram showing group grid theory [35]

The diagram above shows the different categories of the group-grid. Group is the extent to which an individual is affected by their social group. Grid is the extend to which they are governed by rules. These can be rated from high to low. By combining the grid and group scales Douglas identified four separate social outlooks: Fatalistic, Hierarchic, Individualistic and Egalitarian.

Cultural Theory concludes that individuals within each social outlook group perceive events which threaten their outlook as risky. This can cause individuals within each group to have different risk perceptions for the same hazard.[21]

1.10.3 Dread Events

An event which has a low probability but a high consequence if it is realised is known as a dread event. Research has shown that people find it hard to imagine dread events happening to themselves.[15] Furthermore, people find it extremely difficult to estimate low probabilities such as: 0.000001. This is because people have been shown to perceive risks based on past experience, this results in a" bias of information availability". [20] Studies which asked participants to estimate the frequency of different causes of death-for example accidents and diseases- show that people overestimate the risks of low probability causes of death, such as a plane crash and, underestimate the risk of higher probability causes of death such as, heart disease.

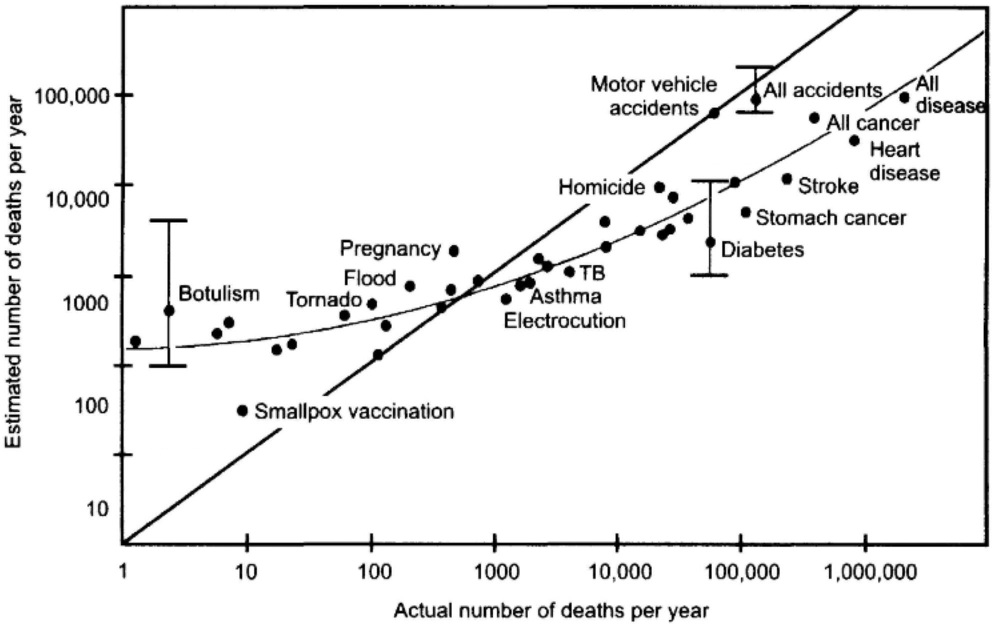

Figure 1.18: Relations between judged frequency and the actual number of deaths per year for 41 causes of death.[43]

The above figure shows people's estimates of death per year for different causes against the actual number of deaths per year for each. It clearly shows that people overestimate the deaths caused by low probability high consequence events, for example: people estimated that Tornado's kill approximtly 700 people per year in the United States however the actual figure is shown to be around 60. Additionally it shows people underestimate high probability causes of death, for example; Motor vehicle accidents were estimated at aproximtely 4000 per year. The actual figure was approximtly 80,000 per year.

Chapter 2

System Proposal

The system proposal chapter will outline the system concept, how the concept has been created as well as the system stakeholders and requirements.

2.0.4 High Level Concept

The system concept is based on the identified need for a system which is capable of simulating the risks of sub-orbital RLV re-entry on conventional aviation. This need has been identified through three main sources:

- Background Research: The background research explored in the previous chapter suggests that sub-orbital launches are set to increase in the coming years and, debris from a re-entry disaster presents a potential hazard to conventional aviation. In order to plan for future sub-orbital flights it is essential to analyse these risks.
- Analysis of other tools: There are a limited number of tools which analyse the impact of RLV debris on aviation. The tools which do exist are designed for calculating the debris of an orbital shuttle- specifically the NASA space shuttle. A system which can analyse the risk of multiple sub-orbital RLV would be desirable.
- Communication with Specialists: Based on the background research and other tools analysis a system proposal was shown to the European Space Agency's Re-entry and Space Debris Safety Manager, who expressed a keen interest in such a system.

2.0.5 Stakeholders

Potential stakeholders of the system have been identified and are listed below.

Aviation Authorities

Aviation authorities such as the FAA may use the system to simulate the potential risks posed by an accident during the re-entry of a sub-orbital flight. By analysing the risk of particular shuttles and space ports, the FFA could then take this information into consideration when issuing sub-orbital flight licences or spaceport operation licences as discussed in Commercial Space Port Licensing Review and Recommendations.[17]

Air Traffic Controllers

ATC(Air Traffic Controllers) may use the system to experiment with potential future airspace systems in order to analyse their impact.

Commercial Space Flight Companies

Sub-orbital spaceflight companies may use the software in a similar manner to aviation authorities to visualise the impact of the debris field created by their RLV. By doing so the company could evaluate the levels of risk associated with potential space port locations, the best time of day for the shuttle to re-enter, and other possible ways of minimising risk.

Researchers

The system may be used by academics wishing to research the risks sub-orbital space travel poses on air traffic.

2.0.6 Functional Requirements

To capture the system's functional requirements the MoSCoW method was used. The MoSCoW method is a simple scheme for prioritising requirements. The method separates system requirements into four different categories based on their priority. This allows developers to focus on high priority requirements first. The four categories of the MoSCoW method are shown below:

- **Must Have** Requirements prioritised as 'must have' are essential to the system's success and so must be met.
- **Should Have** Should have requirements are important to the system's success but are not critical.
- **Could Have** These requirements do not necessarily have to be met but the system would benefit from their inclusion.
- **Would like to Have** Requirements labeled as 'would like to have' are requirements which will not be met by the system at this time however they may be included in later revisions.

The MoSCoW analysis of the system's requirements is shown below.

Requirements were gathered iteratively over the course of the project. New requirements were added, removed and priorities changed as the research focus became more defined. Requirements validation can be found in the Evaluation chapter.

2.0.7 Non-Functional Requirements

Non functional requirements are used to describe constraints and qualities that are possessed by the system. Considering the Non Functional Requirements should improve the overall system. The Non Functional Requirements are detailed below.

Must Have	Should Have
Access to a form of airspace data	Live airspace data
Mode debris dispersion for an RLV	Model debris dispersion for a user entered RLV
Detect collision between aircraft and debris	Allow user to choose location
Location imagery	Allow user to control simulation (play/pause)
Statistical output	Output results to a file
Visualisation	

Could Have	Would Have
Live meteorological data	Air Traffic Control interactions
Different visualisation angles	Repeated simulation mode
Allow user to input airspace data	
Debris colour coded based on risk	

Table 2.1: Functional Requirements

- The system should be designed in such a way that it is easily extendible.
- The system is relatively graphically intense therefore it requires the machine has support for OpenGL.
- The system has been developed and tested on an intel based Mac. The System has also been tested on Windows 7.
- The system should be portable and maintainable.

2.0.8 System Scope

The system scope outlines what the system should and should not do. What the system should do has been covered in the functional requirements. Functonality the system will not support is listed below:

- The system should use live flight data to setup aircraft positions and flight paths. However it will not continuously access live data. Aircraft will remain at their original altitude and follow the retrieved flight path. Any changes in altitude or flight path will not be modelled. This was deemed suitable as aircraft are only likely to change altitude significantly during take off and landing. Furthermore the aircraft would still be in the debris hazard zone irrelevant of altitude.
- The system does not model any air traffic controller interaction. The user can not modify aircraft flight paths.

2.0.9 System Actors and Use Cases

Use case diagrams provide a high level overview of the system and allow the system actors to be easily identified. A use case diagram for the system is shown below.

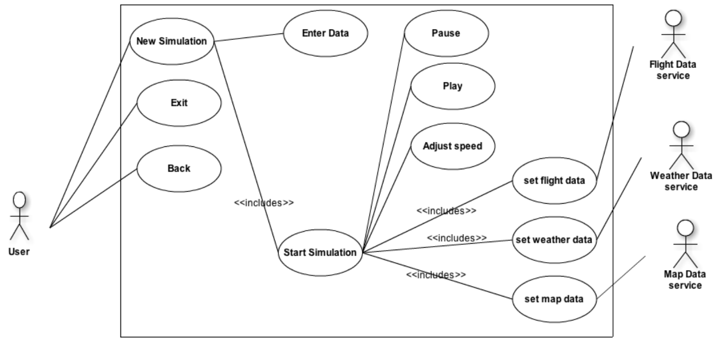

Figure 2.1: A use case diagram showing the interactions between the system and actors

Chapter 3

Design

3.0.10 Coding Tools

In order to implement the system a programming language must be selected.Some key requirements the language must meet were identified:

- **GUI Support** The system aims to be user friendly and easy to use therefor the chosen language must have support for creating a graphical user interface.
- **3D Support** The language must support 3D graphics and provide libraries which aid in the development of 3D systems.
- **Documentation** In order to aid development the chosen language must be well documented and provide support for developers.

C++

C++ is an object oriented extension of the C programming language. It is the industry standard for developing 3D games. As such there is a large amount of documentation available. C++ provides excellent speed optimisation potential and allows low level access to OpenGL which can be used to improve performance. However low level OpenGl usage can be complex and time consuming.

JME

JME(JMonkeyEngine) is a collection of java libraries which add 3D functionality to the Java Programming language. It provides advance support for: Lighting and shading, physics and networking. JME also provides a SDK which simplifies asset management and the importing of 3D models.

Additionally JME supports a 2D GUI through the Nifty GUI library. Nifty provides support for multiple GUI elements and the interface is clean and modern.

JME has been in development since 2003 and continues to be developed. There is a large amount of documentation and an active community.JME is currently in use by a number of commercial game studios and appears to be a robust piece of open source software.

Java3D

Java3D is an API for the java programming language which adds support for 3D. It provides constructs for creating a 3D geometry and building the structures used for rendering that geometry. As Java3D is simply a java library the extensive collection of standard java libraries can also be used. Java also brings with it support for the swing GUI library.

However Oracle(then sun) ceased development of Java3D in 2004, since then it has been open source and developed by a community. The last major release occurred in 2008.

Conclusion

Shown below is a Question Objective Criteria(digram) which shows the selection criteria used for choosing a programming environment.

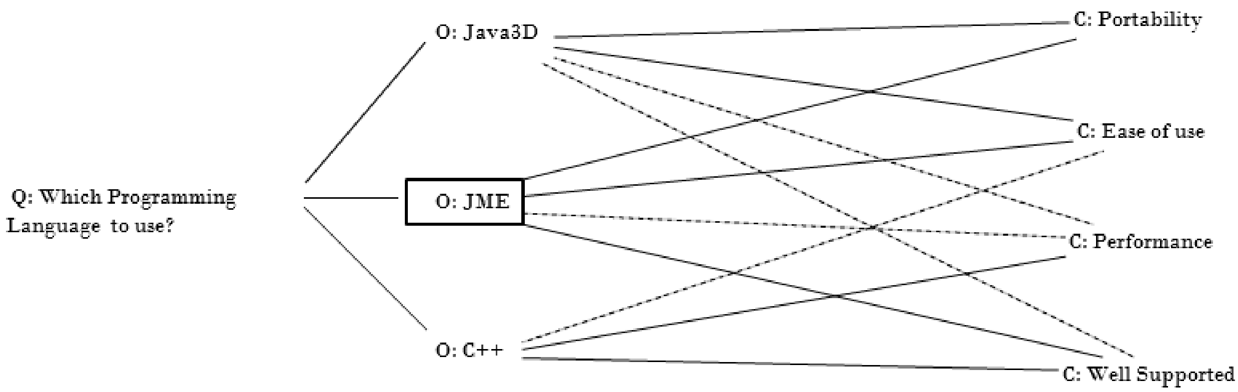

Figure 3.1: A QOC diagram for programming language choice

Although all of the above languages could be used to develop the system, JME was chosen for a number of reasons. C++ provided performance benefits that came at a trade off with complexity. These benefits were not deemed necessary for the graphical visualisation required by the system and the additional complexity could hinder development time. Java3D is a relatively old technology which is not in active development. JME can be seen as a natural successor to java3D providing all the benefits that java brings-such as cross platform deployment as well as the additional benefits of a modern system.

3.1 Web Services

The system required access to three different sets of data in order to create a visualisation, mainly:

- Airspace Data: Flight data is required to accurately model aircraft and their flight paths within the system.
- Map Data: Map Data is required so that the aircraft can be projected on to a representation of their real world location.
- Meteorologic Data: Wind speed and direction influence debris dispersion therefore access to current meteorologic data will increase the accuracy of the debris model's prediction.

3.1.1 Airspace Data

In the early stages of project design numerous potential options for accessing airspace data were considered:

ADS-B Receiver

Attempts were made to gain access to a network of ADS-B receivers or a single receiver. Accessing pure ADS-B data would reduce coupling as the system need not rely on a third-party web service. This would also result in extremely accurate data. However this method also has some disadvantages; de-coding ADS-B data is a non-trivial task and is not the focus of this research, if the ADS-B receiver network is small or only one device is used then only aircraft data within range of those receivers will be available. This would limit the system by restricting simulations to one location.

Web Scrapping

If access to real-time flight data was unavailable scrapping web data was considered as a backup plan. This would likely be undertaken manually by recording aircraft flight paths shown by sites such as PlaneFinder. This data could then be used to create a number of flight data samples which could be integrated into the system. This method was not ideal because the flight data is based on limited data samples and would be restricted to a limited number of locations.

Flight Data Server

There exists various web services which collect and gather live flight data. This data is gathered through a network of ADS-B receivers (often contributed by hobbyist) as well as government sources and registered flight plans. These services often provide an API which allows developers to access this data. This was chosen as the best way to access flight data as it allows access to ADS-B data in an easy to use processed form, although the data will be subject to slight delays this was deemed acceptable for the system's scope. FlightAware.com was chosen as the flight data server.

3.1.2 Map Data

Aircraft must be projected on some form of world visualisation. A number of different services were considered for integrating a location data and imagery into the system:

Google Maps

Goolge maps provides an extensive javascript API for retrieving and manipulating map date including ariel imagery. There are also unofficial libraries which wrap the javascript API in java including arial imagery. Google maps has a number of unique features which may be useful for the system:

- 3D buildings
- traffic information
- Three imagery types: Aerial, terrain and standard.
- High quality imagery

However it became apparent that using Google Maps outside of a web browser environment violated the terms of service. As the system is a desktop application Google Maps could not be used.

NASA Worldwind

World Wind is an open-source virtual globe created by NASA. WorldWind provides developers with access to a virtual globe of Earth to which they can add their own data. WorldWind includes high quality arial imagery.

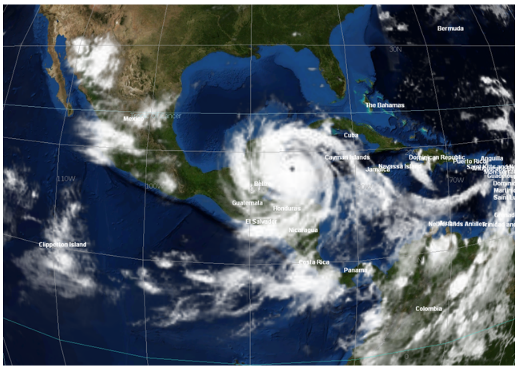

Figure 3.2: NASA WorldWind Imagery

Unfortunately the system is tied to a SDK which uses a conflicting technology to JME therefor the two cannot be easily integrated.

MapQuest Tiles

MapQuest is a free online web mapping service. It offers numerous APIs for accessing map data, most of which are focused on displaying map data in web browsers through javascript. However it also offers a Map Tile Service. This service can be used to download map imagery in smaller tiles which can then be pieced together. MapQuest has a number of advantages:

- The service is free and open.
- Offers two types of imagery: Aerial and standard map.
- Allows the imagery to be used in mobile,desktop and web applications

The disadvantage of using the MapQuest Tile server is it's lack of API. only images are obtained therefor any geo-processing functionality required needs to be coded into the system. This was not considered to make MapQuest unesable and so MapQuest was choosen for providing the application with map imagery.

3.1.3 Meteorologic Data

The debris model will make use of windspeed and direction in order to predict debris dispersal. Two possible methods of gathering this data were considered. Wind speed and direction could be entered by the user or current weather data could be obtained from a web service. A web service was chosen as the preferred method for gathering the data. The reason for this is because gathering data from the user decreases the system's usability and may lead to input errors.Wunderground.com was selected as a source of meteorologic data.

3.2 Debris Model

After surveying existing debris models(Chapter 2, section)The Trajectory Analysis Program (TAP) debris model was chosen for integration with the system as it is freely available. The other models considered are commercial models and despite requests for access were not available for educational use. As the TAP model is designed to model debris dispersal of aircraft disintegrations, some potential caveats have been identified which may affect its application to sub-orbital RLVs:

- lateral corrections for wind shift are not made: This means the model assumes no sudden change in wind direction. This assumption may be reasonable for limited altitudes of aircraft which operate in the troposphere. However a RLV will reach at significantly higher altitude, approx. 100 km, and operate within three atmospheric zones:the troposphere, the stratosphere and the mesosphere. Therefore there is a greater potential for sudden wind shifts to occur.
- The model assumes that the aircraft suddenly disintegrates into a number of parts. It does not model multiple or progressive disintegration.

Despite these the model was deemed reasonable for a prototype system and was successfully verified against one known debris field created by an orbital shuttle(Chapter xyz, section xyz)

3.3 Visualisation and GUI

3.3.1 Visualisation

Map Visualisation

As map tiles were chosen to represent the location, the location would then be a flat two dimensional plane. It was decided that the plane would be placed across the x and z axis, forming the visualisation floor.

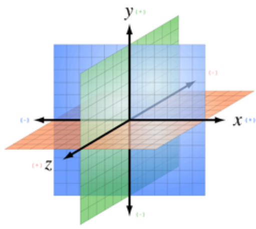

Figure 3.3: Planned location of the map tiles

The red plane in the figure above shows the planned location of the map tiles. This allows aircraft to be projected on top of this floor, forming a three dimensional visualisation. The visualisation can then show the aircraft's position and its altitude.

AirspaceVisualisation

Aircraft within the visualisation are represented by a 3D model of a Boeing-747 aircraft. This was chosen as it is instantly identifiable as an aircraft. Aircraft follow a flight path which is defined by waypoints.A red square was chosen to represent the waypoints within the system and a blue line the flight path. These were chosen as they are highly visible against the black background and do not obscure the white aircraft.

Debris Field Visualisation

Originally the debris field was planned to be visualised as an ellipse which would be colour coded to represent the severity of impact with certain areas of the debris field. However limitations on time and resources resulted in a simpler debris visualisation- a yellow rectangle.

3.3.2 GUI

Main Interface

The main aim of the GUI design is to be simple, easy to use and not to hinder the users interaction. Nifty was selected as the GUI framework as it has a modern feel and elements are defined in XML and functionality in java ensuring a separation of concerns.

Output

A graphical visualisation alone is not enough to asses risk therefore an important feature is to provide output statistics; such as the number of impacts and the total number of aircraft. This would aid in performing any kind of formal risk analysis.

3.4 System Architecture

The system is designed using the Model View Controller (MVC) design pattern. This provides a separation of concerns helping to ensure the data(Model)-such as the debris model is kept separate from the logic(Controller) and the user interface/Display(View)

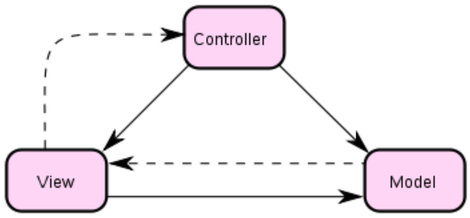

Figure 3.4: The system sceneGraph

The figure above shows the MVC design pattern.

3.4.1 Debris Model Class

The MVC design pattern will allow easy changes to the model data, such as the debris model, to be made in the future without major modification to the view or controller. A class diagram representing the debris model is shown below.

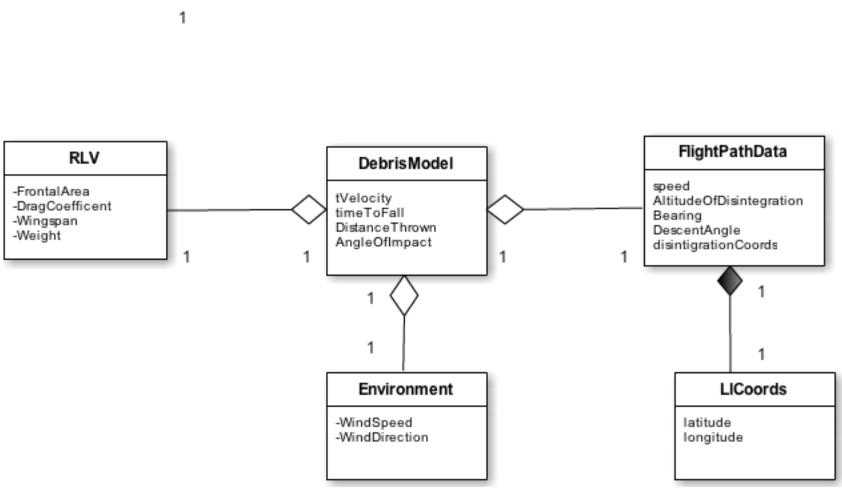

Figure 3.5: The Debris Model Class and related classes

3.4.2 Aircraft Class

How to model an aircraft within the system is another important design choice. The Aircraft class is shown below

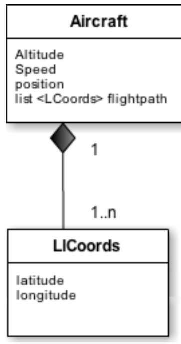

Figure 3.6: Aircraft Class

An Aircraft has data members to represent its current position,speed, heading and altitude. AS well as a list of waypoints represented using the LlCoord Class.

Chapter 4

Implementation

This chapter will briefly cover some key features of the System's implementation.

4.1 Scene Graph

The view is implemented using a scene graph, a scene graph is a tree data structure. Nodes of the scene graph are objects of abstract type `Spatial`. A `Spatial` can be instantiated to one of two possible sub classes: `Geometry` and `Node`. An object of type `Geometry` can not have child objects, it is analogous to a leaf node of a standard tree data structure. `Geometry` objects attached to the scene graph's `rootNode` are visible in the view. An object of type `Node` is not visible but can have children of type `Spatial`.

The system has a scene graph with three `Spatial` objects: `aircraft`, `map`, `debris` attached to the `rootNode`. `aircrafts` has a child `Geometry` object for each aircraft displayed. map has a child `Geometry` for each of the sixteen map tile which make up the map visualisation. `debris` has a `Geometry` object representing the debris visualisation.

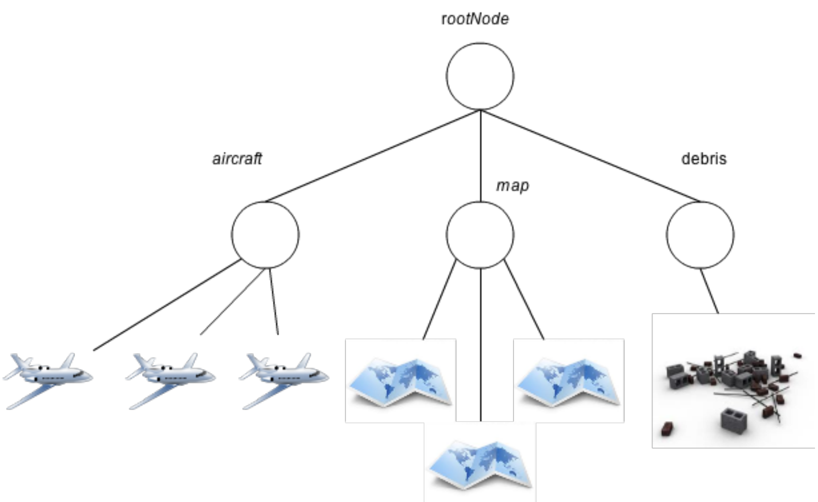

Figure 4.1: The system sceneGraph

The figure above shows an example state of the system's scene graph. Only three map tiles are shown above however the system uses sixteen map tiles. Three plane `Geometry` objects are also shown, `aircraft` may have 0 to n children, depending on the number of aircraft within the real world airspace

By structuring the scene graph in this way it helps separate concerns and reduces the complexity of traversing the scene graph when searching for a particular `Geometry` object. For example: when the system needs to detect which `Geometry` object representing aircraft is involved in a collision, only the children of `aircraft` needs to be traversed. Additionally transformations can easily be applied to all child `Spatial` objects by transforming their parent `Node`. This allows all the children of `aircraft` to be repositioned or scaled as a collection.

4.2 Map Tiles

The system obtains map imagery in the form of tiles from the MapQuest web server. the static class MapQuest-Client contains a method which sends REST requests(using the Jersey library) to the server. The request contains a zoom level-a zoom level of 1 will return tiles representing a map of the world, increasing the zoom level will return tiles which are "zoomed" closer to the central coordinate. The request also contains the map tile type-arial imagery or standard map imagery and a tile key, made from an x and y integer..

```
public class MapQuestClient {

    public static BufferedImage buildResponse(String zoom, String x, String y, boolean useSat) {
        Client client = Client.create();
        String base = "";
        if (useSat) {
            base = "http://otile1.mqcdn.com/tiles/1.0.0/sat/" + zoom + "/" + x + "/" + y + ".jpg";
        } else {
            base = "http://otile2.mqcdn.com/tiles/1.0.0/osm/" + zoom + "/" + x + "/" + y + ".jpg";
        }
        WebResource webResource = client.resource(base);
        return webResource.get(BufferedImage.class);

    }
}
```

Before a request can be sent it is necessary to calculate which tiles are needed. As part of the required flight data the user enters a latitude and longitude of the RLV disintegration, this is used as the central map point. The user also enters a zoom level. Combining the zoom level and central point the number of the central tile can be found as shown below:

```
protected Tile getTileNumber(final double lat, final double lon,
            final int zoom) {
        int xtile = (int) Math.floor((lon + 180) / 360 * (1 << zoom));
        int ytile = (int) Math
                .floor((1 - Math.log(Math.tan(Math.toRadians(lat)) + 1
                / Math.cos(Math.toRadians(lat)))
                / Math.PI)
                / 2 * (1 << zoom));
        return new Tile(xtile, ytile, zoom);
    }
```

Once the key of the central tile is known it is possible to calculate the keys of the remaining tiles:

```
    public void getTiles(Tile t, ArrayList<Tile> tiles) {
        int x = t.x;
        int y = t.y;
        int zoom = t.zoom;
        int oddEvenX = x % 2;
        int oddEvenY = y % 2;
        int maxX = x + 2 - oddEvenX;
        int minX = x - 1 - oddEvenX;
        int maxY = y + 2 - oddEvenY;
        int minY = y - 1 - oddEvenY;
        if (minX < 0) {
            maxX = maxX - minX;
            minX = 0;
        }
        if (minY < 0) {
            maxY = maxY - minY;
            minY = 0;
        }
        if (maxX > (Math.pow(2, zoom) - 1)) {
            minX = minX - (int) (maxX - (Math.pow(2, zoom) - 1));
            maxX = (int) (Math.pow(2, zoom) - 1);
        }
        if (maxY > (Math.pow(2, zoom) - 1)) {
            minY = minY - (int) (maxY - (Math.pow(2, zoom) - 1));
            maxY = (int) (Math.pow(2, zoom) - 1);
```

```
        }
        int minXcopy = minX;
        while (minY <= maxY) {

            minXcopy = minX;
            while (minXcopy <= maxX) {
                tiles.add(new Tile(minXcopy, minY, zoom));
                minXcopy++;
            }
            minY++;
}
}
```

The method above accepts the central tile t as a parameter and an Arraylist of Tiles. As the map is made from 16 tiles in a 4 by 4 formation the central tile is always off centre, the method calculates the actual position of the central tile and the maximum x and y tiles required. Once this has been calculated the remaining tile keys are all integers where $x >= minXandx =< maxXandx! = t.x$ The same follows for y.

Once the x and y key values of each required tile are known, the tile REST request can now be sent. The method which carries out all of the above is shown below:

```
  public void updateTiles() {

        double lat = flightData.coords.latitude;
        double lon = flightData.coords.longitude;
        ArrayList<Tile> tiles = new ArrayList<Tile>();
        getTiles(getTileNumber(lat, lon, zoomLevell), tiles);
        bb = new BoundingBox(tiles.get(0).x, tiles.get(0).y, tiles.get(15).x, tiles.get(15).y, tiles.get(0).zoom);
        LlCoords.bb = bb;
        ImageCache im = new ImageCache();
        int count = 0;
        for (Tile x : tiles) {
            BufferedImage bImage = im.getImage(x.x, x.y, x.zoom, useSat);
                updateMap(("" + count), bImage);
                count++;
        }
    }
```

After the tile numbers are calculated the requests are sent and the resulting map tiles are stored in an ImageCache, this means if the user decides to run the simulation again the images are already stored and requests do not need to be repeated. Using the ImageCache helps improve the systems overall efficiency. It should be noted that the optimum way of improving loading time would be to store the tiles on the hard disk, however this would quickly lead to saving thousands of images which would take up hard drive space perhaps unbeknown to the user. Additionally saving a permeant local copy of the map tile data is againstMapQuest's terms of service.

4.3 Map projection

The flight data server returns a list of waypoints(latitude and longitude pairs). These are parsed and stored in a list of LlCoords. An LlCoord is simply a class representing a latitude and longitude coordinate.

As the visualisation uses a two dimensional representation of a world location, however latitude and longitude coordinates are based on a spherical earth model. it is therefore necessary to perform a conversion in order to accurately represent the latitude and longitude as a cartesian(x,y) coordinate within the system's visualisation.

The conversion must match the projection type of the 2D map imagery otherwise the point will be inaccurately placed on the map. MapQuest tiles are based on Mercator Projection. The LlCoords class contains a method for converting the latitude and longitude to a Vector3f object: convert(int alt) the returned Vector3f object is the cartesian coordinates of the point on the map projection as well as the altitude. An extract of the code is shown below.

```
        double mapLonDelta = mapLonRight - mapLonLeft;
        double mapLatBottomDegree = mapLatBottom * Math.PI / 180;
        double worldMapWidth = ((mapWidth / mapLonDelta) * 360) / (2 * Math.PI);
        double mapOffsetY = (worldMapWidth / 2 * Math.log((1 + Math.sin(mapLatBottomDegree)) / (1 - Math.sin(mapLatBottomDegree))));
        double x = (lon - mapLonLeft) * (mapWidth / mapLonDelta);
        double y = 0.1;
        if (lat < 0) {
            lat = lat * Math.PI / 180;
```

```
        y = mapHeight - ((worldMapWidth / 2 * Math.log((1 + Math.sin(lat)) / (1 - Math.sin(lat)))) - mapOffsetY);
    } else if (lat > 0) {
        lat = lat * Math.PI / 180;
        y = mapHeight - ((worldMapWidth / 2 * Math.log((1 + Math.sin(lat)) / (1 - Math.sin(lat)))) - mapOffsetY);
    } else {
        y = mapHeight / 2;
    }
```

The mapLonRight, mapLonLeft,MapLatBottom are obtained from the boundingBox latitude and longitude values described earlier. The code then implements the standard formulas for the Mercator Projection of the current map being displayed.

4.4 Collision Detection

Collision detection is implemented through the use of the JBullet physics library. A `GhostControl` is created for every plane `Geometry`, a `GhostControl` is an invisible mesh which outlines the `Geometry` and follows it. It is capable of registering interactions between other `GhostControl` objects.

The code below shows a `GhostControl` being instantiated for a `Geometry`(which represents an aircraft). The `GhostControl` must then be registered with the physics controller.

```
CollisionShape sceneShape = CollisionShapeFactory.createDynamicMeshShape((Node) plane);
                GhostControl ghost = new GhostControl(sceneShape);
                plane.addControl(ghost);
                bulletAppState.getPhysicsSpace().add(ghost);
```

Every aircraft visualised by the system has a `GhostControl`, as does the debris. This allows collisions between debris and aircraft to be registered.

In order to detect a collision a collision listener must be registered to the physics controller:

```
  bulletAppState.getPhysicsSpace().addCollisionListener(new runningState.DebrisCol());

 public void collision(PhysicsCollisionEvent event) {

                if (event.getNodeA().getParent().getName().equalsIgnoreCase("aircraft") && event.getNodeB().getParent().getName().equalsIgnoreCase("deb
                    Spatial nd = event.getNodeA();
                    nd.removeFromParent();
                    collisionCount++;
                } else if (event.getNodeB().getParent().getName().equalsIgnoreCase("aircraft") && event.getNodeA().getParent().getName().equalsIgnoreCa
                    Spatial nd = event.getNodeB();
                    nd.removeFromParent();
                    collisionCount++;
                }
            }
}
```

When a collision is detected by the listener the `collision` method is called. The listener gives the`collision` method a `PhysicsCollisonEvent` object which contains the two `Nodes` that collided. If a collision is found the `Geometry` which represents an aircraft is removed from the scene graph and a counter which tracks the number of collisions is incremented.

Chapter 5

Evaluation

5.1 Experiment Design

5.1.1 High Level Aim

As discussed in the previous section people have difficulty imaging and estimating the probability and impact of dread events. The probability of a failure occurring during the re-entry of a sub-orbital RLV resulting in it's disintegration is relatively low. Additionally current spaceports have been strategically located in areas where air traffic is low therefore the probability of this debris impacting an aircraft is also low. However the consequences of such an impact would be considered catastrophic.

Previous research(Pollatsek et al. 2006) has shown that simulation systems can have an affect on people's risk perception. Pollatsek conducted research into the effects of simulation systems on novice driver's risk perception.

The aim of this experiment is to test if the system has an effect on participant's risk perception of sub-orbital debris caused by a RLV re-entry disaster impacting aircraft.

5.1.2 Hypothesis

The following variables were identified for the experiment:

- Dependant variables: Risk perception
- Independent variables: Use of system

by varying he independent variable-use of system- its effects on the dependent variable can be tested. The independent variable has two possible states, true or false. Following from this the experiment's hypothesis is as follows:

H_1 Using the system will affect the user's risk perception of sub-orbital debris impacting aircraft.

In order to prove the hypothesis the following null hypothesis must be disproved.

H_0 Using the system will have no affect on the user's risk perception of sub-orbital debris impacting aircraft.

5.1.3 Experimental Procedure

The experiment was designed to compare participants baseline risk perception to their risk perception after using the system. A within subjects design was chosen as a comparison between the participant's baseline risk perception and the same participant's post-system use risk perception was required.

Participants were all members of the University of Glasgow level 4 Computing Science cohort therefore the assumption was made that all participant would have the same baseline knowledge of sub-orbital flight operations and their risk. To ensure that the system affected risk perception at both extremes (low risk levels and high risk levels) two trials were conducted. In one trial participants ran simulations at locations at which the risk of debris impacting aircraft was expected to be lower. In the second trial participants ran simulations at locations at which the risk of debris impacting aircraft was expected to be higher In order to measure risk perception a psychometric test outlined in Fischoff et al 1978 was used. As discussed in the previous section psychometric tests are an accepted model for obtaining quantitative data on risk perception. The test asked participants to consider the risk of a sub-orbital vehicle disintegrating into debris and this debris impacting an aircraft. Participants were asked to then rate the risk they would associate with this scenario. The psychometric also analysed if participants found the current level of risk acceptable as well as their perception of the risk based on the nine elements(control,consequence,knowledge etc) discussed in Fischoff 1978.

Trial One

All participants were shown a short video of debris falling to Earth(Space Shuttle Columbia re-entry footage). This ensured that all participants were given the same information on debris created by the re-entry of a space vehicle. Participants were then asked to complete the psychometric test. After completing the test participants were then asked to use the system to simulate debris dispersal at three different locations.

The RLV data used for the simulations was based on Virgin Galactic's SpaceShipTwo. The flightpath data used was based on nominal sub-orbital re-entry trajectory. This input data can be found in the appendices. This data was kept constant for all simulation runs.

For the first trial each participant was given the same three locations. The locations used were:

- Mojave Space port
- Space port America
- Corn Ranch Space port

These locations were selected because they are locations at which a RLV can reasonably be expected to be returning to. Furthermore the airspace at these locations is not densely populated with aircraft therefore it was expected that the chance of debris impacting an aircraft in these locations was slim. After completing the simulations participants were asked to complete the psychometric test once again, considering what they had seen and the output of their simulation runs.

Trial two

Trial two used a different set of participants and followed the same procedure as trial one with the exception that the locations used in the simulation runs were changed.The locations used were.

- JFK Airport, New York
- Los Angeles international Airport, Los Angeles
- Miami International Airport, Miami.

These locations were selected because the airspace at these locations is densely populated with aircraft. As such it was expected that the chance of debris impacting aircraft in these locations was high.

5.2 Analysis Method

In order to analyse the results obtained from the experiment it is necessary to consider different methods of statistical analysis.

5.2.1 Parametric Tests and Non-Parametric Tests

A parametric test is a a group of statistical tests which make assumptions about the data. A well known example is the T-test.One of the major assumptions is that the sample data has a normal distribution for the population being tested.[18]

Standard normal distribution means if the experiment was repeated an infinite number of times on different samples and the mean of the results taken and plotted as a line graph they would always form a perfect bell curve as shown below.[18]

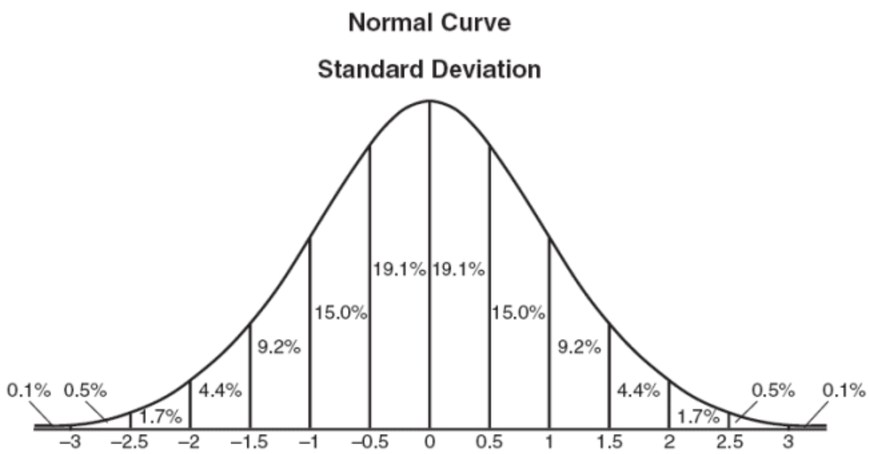

Figure 5.1: Bell curve of standard normal distribution

Parametric tests are capable of rejecting or confirming the null hypothesis with greater statistical power than their non-parametric counterpart. However if the assumption of normal distribution is false then false positives may be found and the results may be misleading.

Non-parametric tests do not make as many assumptions about the data as parametric tests. Importantly they do not assume that the sample data has a normal distribution. As such non-parametric tests can be considered more robust. Although non-parametric tests are more robust they have less statistical power than parametric tests.[18]

5.2.2 Conclusion

The experiment has two trials each of which use five participants. This is considered a very small sample set. The central limit theorem shows that a sample set of less than thirty is unlikely to yield a normal distribution of the population.[16] Given that the sample groups within the experiment are very unlikely to represent a normal distribution which would invalidate the assumptions made by a parametric test. A non-parametric test has been chosen for the analysis.

5.2.3 Wilcoxon signed-rank test

The Wilcoxon signed-rank test can be used when comparing two repeated measurements as such it is an appropriate test for analysing the results of the experiment.

Definition

H_0 : The median difference between the paris is 0

H_1 : The median difference is not 0 let N be the sample size, thus, there are a total of $2N$ data points.

For $i = 1, ..., N$, let $x_{1,i}$ and $x_{2,i}$ denote the measurements.

For $i = 1, ..., N$ calculate $|x_{2,i} - x_{1,i}|$ and $sgn(x_{2,i} - x_{1,i})$

Remove pairs where $|x_{2,i} - x_{1,i}| = 0$

let N_r be the reduced sample size.

Order the N_r pairs from the smallest absolute different to the largest absolute difference.

Rank pairs starting fro m1.

Rank Pairs which have an equal absolute difference as the average of the ranks they span.

Let R_i denote the rank.

Let W = $|\sum_{i=1}^{N} r = [sgn(x_{2,i} - x_{1,i}) \cdot R_i]|$

x_1	x_2	$sgn(x_{2,i} - x_{1,i})$	$\lvert x_{2,i} - x_{1,i} \rvert$	Ordered $\lvert x_{2,i} - x_{1,i} \rvert$	R_i	$sgn(x_{2,i} - x_{1,i}) \cdot R_i$
8	9	1	1	1	1	1
4	9	1	5	3	2	2
5	8	1	3	4	3	3
1	5	1	4	5	4	4
3	10	1	7	7	5	5

5.2.4 Results

Trial One Results Analysis

The table above shows participants rating of risk before(x_1) and after using the system(x_2). Other columns show the results at each step of Wilcoxon's signed rank test. W can be obtained by summing the final column $sgn(x_{2,i} - x_{1,i}) \cdot R_i$ therefore:

$$W = 15 \tag{5.1}$$

As W is not equal to 0 there exists a difference between the medians of the pre and post system test, as such the null hypothesis has been rejected.The results show that there was a statistically significant difference in participant's risk perception before and after using the system, this suggests that interacting with the system had an affect on the participant's risk perception.

Furthermore the results show that the participants perceived less risk from sub-orbital debris impacting aircraft after using the system. This was the expected result as described area the locations chosen for the simulations in trial one were unlikely to cause any fatalities.

Trial Two Results Analysis

x_1	x_2	$sgn(x_{2,i} - x_{1,i})$	$\lvert x_{2,i} - x_{1,i} \rvert$	Ordered $\lvert x_{2,i} - x_{1,i} \rvert$	R_i	$sgn(x_{2,i} - x_{1,i}) \cdot R_i$
2	1	-1	1	1	1.333	-1.333
4	1	-1	3	1	1.333	-1.333
10	4	-1	6	1	1.333	-1.333
2	1	-1	1	3	4	4
3	2	-1	1	6	5	-5

The table above shows participants rating of risk before(x_1) and after using the system(x_2). Other columns show the results at each step of Wilcoxon's signed rank test. W can be obtained by summing the final column $sgn(x_{2,i} - x_{1,i}) \cdot R_i$ therefore:

$$W = -12.9999 \tag{5.2}$$

As W is not equal to 0 there exists a difference between the medians of the pre and post system test, as such the null hypothesis has been rejected. The results show that there was a statistically significant difference in participant's risk perception before and after using the system. This suggests that interacting with the system had an affect on the participants risk perception.

Furthermore the results show that the participants perceived a greater risk from sub-orbital debris impacting aircraft after using the system. This was the expected result as described previously the locations chosen for the simulations in trial two were likely to cause fatalities.

5.3 Overall Conclusion

From the results outlined it can reasonably be concluded that the system is able to affect the users risk perception of suborbital debris impacting aircraft. The results also show that the system is capable of affecting the user's risk perception in two directions, allowing them to perceive more or less risk based on the results presented by the system.

One possible reason for this is people's perception of risk is based on their own past experience. Covello [20]suggests that people are unlikely to take any protective action against risk unless it has occurred to them before or they have been made graphically aware of it.

The system is a method of making people graphically aware of the risk by presenting both a visualisation and a statistical output.

Given the limitations on time and resources which were available for the evaluation it should be noted that further research is required in order to ensure experimental validity. One potential improvement would be increasing the sample size.

5.4 Other Findings

Participants were also asked to rate the risk on the nine elements which Starr and Fisschoff(1986) identified affect risk perception. In both trials the means of the ratings did not differ by more than one and so were merged.

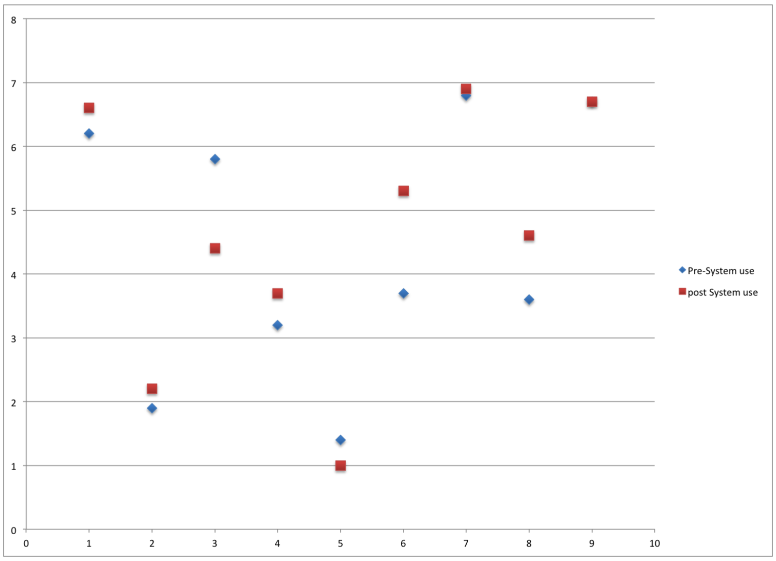

Figure 5.2: Rating of elements affecting risk perception pre and post system use

The scatter plot above shows the participants mean rating of each element pre and post system use. Overall there appears to be little correlation between system use and the ratings given. However some characteristics showed change. participants associated more dread with the risk after using the system than before. After using the system participants also felt that the risk was less of a voluntary one and believed people have less control than pre-system.

Fisschoff showed that dread is linked to the voluntariness of the risk and the level of control. People have a greater sense of dread for risks which they do not volunteer for and do not have control over. The results obtained would seem to mirror this.

5.5 Validation

5.5.1 Debris Field Validation

As discussed in the design section the TAP debris model was not designed specifically for the debris field created by sub-orbital RLVs. In order to validate the use of the TAP debris model within the system it was used to predict an already known debris field.

Space Shuttle Columbia Debris Field

The debris field created by Columbia is known to be around 350 miles. Approximate vehicle specifications and flight path data(obtained from the CAIB report) based on the point of breakup for the Columbia were entered into the TAP model. The TAP model estimated that the debris would disperse 336 from point of break up.

This suggest that the TAP debris model provides a reasonable estimation of debris dispersal. However it should be noted that the Columbia is an orbital space shuttle and not a sub-orbital RLV.

5.5.2 System validation

The system functionality and aims were described to the European Space Agency Debris Safety Manager, who showed a great interest in experimenting with the system.

5.6 Usability Evaluation

The following section documents the results of the different usability evaluations carried out. Usability evaluations are concerned with how easy the system is to use, generally focusing on the systems interface.

5.6.1 Retrospective Think Aloud

A retrospective think aloud involves the user carrying out a set of tasks. After they complete the tasks They can then verbalise any problems they had and explain choices they made while carrying out the tasks. A retrospective think aloud was chosen over a think aloud as speaking out loud while using the system is not natural user behaviour.[27] Furthermore speaking may have distracted the user from the simulation causing them to miss key points which in turn may have affected their risk perception.

In addition to the tasks outlined in the experimental evaluation, users were also asked to input their own RLV data and flight path. A number of different improvements were pointed out by users as well as positive aspects of the system.

- Loading Screen: Multiple users suggested a loading screen should appear after they click start simulation as the black screen provides no information and many users thought the system had crashed.

- Zoom Level: Users found the zoom level confusing and were not sure how it would affect the visualisation. Some users requested functionality for changing the zoom level from within the visualisation.
- Notifications: Some users suggested that it should be clearer when a plane is impacted by debris, perhaps through an alert, sound or 'explosion'.
- Debris visualisation: Users suggested the visualisation of the debris field could be improved. Additionally users commented that it was difficult to see if the debris had impacted an aircraft.
- GUI: Users commented that the GUI is simple, clean and looks professional.
- Controls: Some users said they found the controls intutive and easy to use
- Save: Users liked that they could save the output of the simulation.

Overall the think aloud provided valuable feedback on the usability of the system, confirming its good features and pointing out a number of potential features for future development.

5.6.2 Neilsens Heuristics

A heuristic based evaluation involves an expert judging the system's interface against a set of standard recommendations. Neilsen's heuristics[**?**] are considered a de-facto standard for performing a heuristic based interface evaluation. The system was evaluated against each of Neilsen's heuristics.

× Visibility of system status:

The system did not meet this heuristic as the user is presented with a black screen while the simulation loads. This should be a loading screen keeping the user informed of the system's state.

✓ Match between system and the real world:

The system was deemed to meet this heuristic as it uses words and phrases which would be familiar to its intended user.

✓ User control and freedom:

The system meet this heuristic as it all system screens presented to the user have the option to go back to the previous screen or to exit.

✓ Consistency and standards:

The system met this heuristic as the terminology and button placement etc and are consistent throughout.

× Error prevention:

The system did not meet this criteria as it has no support for error checking or error dialogs.

✓ Recognition rather than recall:

The system met this heuristic as objects and actions are visible to the user and do not have to be remembered. Furthermore the system saves the input data for later use.

✓ Flexibility and efficiency of use:

The system was deemed to meet this heuristic as it supports some standard key commands, such as tab to switch fields.

- ✓ Aesthetic and minimalist design:

  System dialogs and forms are kept simple. Few buttons are shown on each screen. Overall the design is minimalist.

- × Help users recognise, diagnose, and recover from errors:

  The system does not help users recognise or recover from their errors.

- ✓ Help and documentation:

  The system has no inbuilt help function. However documentation is provided.

Overall the system passed seven of Neilsen's heuristics out of ten. This is mainly a result of the system lacking error checking and alerts as well as a loading screen. It should be noted that in its current state the system is a prototype which focuses on the simulation,output and visualisation. Improving the user interface in order to meet the heuristics could be carried out in future revisions if the prototype is to be developed into a product.

Chapter 6

Conclusion

6.1 Future Work

The system offers different opportunities for future work

Debris Model

The TAP debris model currently integrated into the the system is not an ideal debris dispersion model for sub-orbital RLV disintegration. An area of future work would be to source a debris model design for sub-orbital RLV's or to create a new model for integration into the system.

Evaluation

The evaluation carried out in this paper into the system's affects on risk perception was carried out on a small sample size due to limitations on available resources. Extending this evaluation to a far greater sample size could form the basis of future research and would improve the validity of the findings.

Multiple ADS-B data sources.

FlightAware.com's network is currently US centric as such data coverage is greatest when US based locations are used. FlightAware is actively increasing it's European data coverage as such data coverage should improve in the future. Until data coverage improves the system could utilise multiple different sources of flight data to improve its world data coverage.

Non visual System

In addition to the visualisation which helps users see the potential risks and hazards. The system could also have a non visual mode that can be used for repeated simulation runs, outputting only the statistics. This would lend itself to formal risk analysis using Monte Carlo method.

6.2 Conclusion

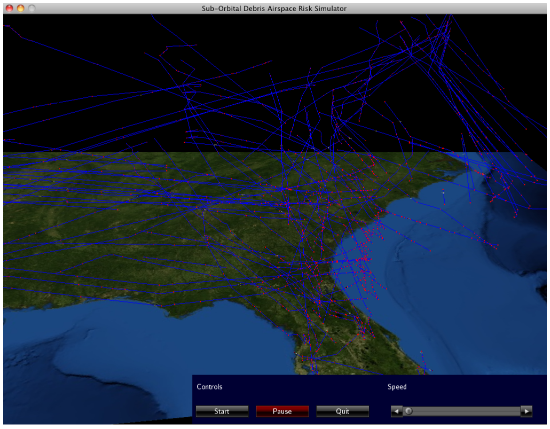

Figure 6.1: An example of the system's visual output

This project explored the the potential impact of sub-orbital RLVs on conventional aviation. This was conducted by reviewing relevant literature and research findings and facilitated by the creation of a simulation system. The system uses live flight data obtained from an Automatic Dependent Surveillance-Broadcast (ADS-B) server combined with up to date meteorological data. The system user enters the specifications of a sub-orbital RLV and the coordinates of disintegration. The system is then able to calculate the predicted debris field and display a model of the subsequent impact on aircraft within the vicinity. Finally an evaluation was conducted to test the affect of the system on people's risk perception, of the impact of sub-orbital RLVs on conventional aviation.

Bibliography

[1] Columbia and challenger: organizational failure at nasa. *Space Policy*, 19(4):239 – 247, 2003.

[2] The threat of orbital debris and protecting nasa space assets from satellite collisions. `http://images.spaceref.com/news/2009/ODMediaBriefing28Apr09-1.pdf`, 2009.

[3] Debra. apt research. `http://www.apt-research.com/products/models/DebRA.html`, 2013.

[4] Dlr. `http://www.dlr.de/`, 2013.

[5] Xcor lynx: Payload capacities. `http://www.xcor.com/products/vehicles/lynx_payload.html`, 2013.

[6] Federal Aviation Administration. The annual compendium of commercial space transportation.

[7] NASA Aviation Systems Divsion. Future atm concepts evaluation tool. `http://www.aviationsystemsdivision.arc.nasa.gov/research/modeling/facet.shtml`, note = " accessed 2013".

[8] Columbia Accident Investigation Board. Columbia accident investigation board report, volume i. part 1. chapter 1, 2003.

[9] Columbia Accident Investigation Board. Columbia accident investigation board report, volume i. part 1. chapter 10.7, 2003.

[10] Columbia Accident Investigation Board. Columbia accident investigation board report, volume i. part 1. chapter 2.6, 2003.

[11] Columbia Accident Investigation Board. Columbia accident investigation board report, volume i. part 1. chapter 2.7, 2003.

[12] Columbia Accident Investigation Board. Columbia accident investigation board report, volume ii. appendix d.16, 2003.

[13] National Transportation Safety Board. Aircraft accident report: In flight breakup over the atlantic ocean trans world airlines flight 800 g 747-131, n93119, 1996.

[14] Noel T Brewer and William K Hallman. Subjective and objective risk as predictors of influenza vaccination during the vaccine shortage of 2004–2005. *Clinical infectious diseases*, 43(11):1379–1386, 2006.

[15] Ian Burton and Robert W Kates. Perception of natural hazards in resource management, the. *Nat. Resources J.*, 3:412, 1963.

[16] Horng-Jinh Chang, Kuo-Chung Huang, and Chao-Hsien Wu. Determination of sample size in using central limit theorem for weibull distribution. *International journal of information and management sciences*, 17(3):31–46, 2006.

[17] Commercial Space Transportation Advisory Committee. Commercial spaceport licensing review and recommendations white paper. http://www.faa.gov/about/office_org/headquarters_offices/ast/advisory_committee/meeting_news/media/2012/oct/Spaceport_Licensing_White_Paper_Final.pdf, 2012.

[18] Hugh Coolican. *Research methods and statistics in psychology*. Hodder Education, 2009.

[19] James Coote. *Mapping planes using ADS-B*. PhD thesis, University of Sussex, 2012.

[20] Vincent T Covello. The perception of technological risks: A literature review. *Technological Forecasting and Social Change*, 23(4):285–297, 1983.

[21] Mary Douglas and Aaron Wildavsky. *Risk and culture: An essay on the selection of technological and environmental dangers*. University of California Press, 1983.

[22] FAA. Fact sheet: Next gen. http://www.faa.gov/news/fact_sheets/news_story.cfm?newsid=8145, 2007.

[23] Plane Finder. How plane finder works using ads-b. http://planefinder.net/about/ads-b-how-planefinder-works/, year = 2013.

[24] Baruch Fischhoff, Paul Slovic, Sarah Lichtenstein, Stephen Read, and Barbara Combs. How safe is safe enough? a psychometric study of attitudes towards technological risks and benefits. *Policy sciences*, 9(2):127–152, 1978.

[25] Virgin Galactic. Virgin galactic. http://www.virgingalactic.com/, 2013.

[26] NASA Glenn Research Center. The drag equation.

[27] Zhiwei Guan, Shirley Lee, Elisabeth Cuddihy, and Judith Ramey. The validity of the stimulated retrospective think-aloud method as measured by eye tracking. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1253–1262. ACM, 2006.

[28] R.A. Freedman H.D. Young and L. Ford. *University Physics with Modern Physics*. Pearson, 2007.

[29] C.W. Johnson. The future of sub-orbital and orbital space accident investigation.

[30] H. Oldham jr. Aircraft debris trajectory analysis. 1990.

[31] TS Kelso et al. Analysis of the iridium 33 cosmos 2251 collision. 2009.

[32] H Klinkrad and B Fritsche. Assessment and management of on-ground risk during re-entries. In *Joint ESA-NASA Space-Flight Safety Conference*, volume 486, page 325, 2002.

[33] Heiner Klinkrad, P Beltrami, S Hauptmann, C Martin, H Sdunnus, H Stokes, R Walker, and J Wilkinson. The esa space debris mitigation handbook 2002. *Advances in Space Research*, 34(5):1251–1259, 2004.

[34] Tobias Lips and Bent Fritsche. A comparison of commonly used re-entry analysis tools. *Acta Astronautica*, 57(2):312–323, 2005.

[35] Virginie Mamadouh. Grid-group cultural theory: an introduction. *GeoJournal*, 47(3):395–409, 1999.

[36] D Murray and Richard VanSuetendael. A tool for integrating commercial space operations into the national airspace system. In *In Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit.*, 2004.

[37] NASA. Nasa orbital debris program office. http://orbitaldebris.jsc.nasa.gov/reentry/reentry.html. Accessed March 2013.

[38] OrbSpace. Suborbital vs orbital space flight. http://www.orbspace.com/Background-Information/Suborbital-vs-Orbital.html. Accessed March 2013.

[39] LUIS FELIPE ROBLEDO. *Analysis and Integration of a Debris Model in the Virtual Range Project*. PhD thesis, University of Central Florida Orlando, Florida, 2004.

[40] M. Sarigul-Klijn and N. Sarigul-Klijn. Flight mechanics of manned sub-orbital reusable launch vehicles with recommendations for launch and recovery. http://www.spacefuture.com/archive/flight_mechanics_of_manned_suborbital_reusable_launch_vehicles_with_recommendations_for_launch_and_recovery.shtml. Accessed March 2013.

[41] Markus Schmidt. Investigating risk perception: a sh ort introduction, 2004.

[42] Martin Sippel. Introducing the spaceliner vision. In *7th International Symposium on Launcher Technologies, Barcelona, Spain*, 2007.

[43] Paul Slovic, Baruch Fischhoff, Sarah Lichtenstein, and FJC Roe. Perceived risk: psychological factors and social implications [and discussion]. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 376(1764):17–34, 1981.

[44] Paul Slovic, Donald MacGregor, and Nancy N Kraus. Perception of risk from automobile safety defects. *Accident Analysis & Prevention*, 19(5):359–373, 1987.

[45] Paul Slovic and Ellen Peters. Risk perception and affect. *Current Directions in Psychological Science*, 15(6):322–325, 2006.

[46] B. L Stevens and F.L Lewis. *Aircraft Control and Simulation*. John Wiley and sons, New jersey, 2003.

[47] ADS-B Technologies. Ads-b technologies - what is ads-b? http://www.ads-b.com, 2013.

[48] Rajkumar Thirumalainambi and Jorge Bardina. Debris dispersion model using java3d. 2004.

[49] Arnold Van Foreest, Martin Sippel, Josef Klevanski, Ali Guelhan, and Burkard Esser. Technical background and challenges of the spaceliner concept. In *7th International Symposium on Launcher Technologies, Barcelona, Spain*, 2007.

[50] Richard Lee VanSuetendael. *Applying Experimental Design in a Debris Dispersion Model Study*. PhD thesis, Florida Institute of Technology, 2003.

[51] P.D Wilde and C. Draper. Aircraft protection standards and implementation guidelines for range safety.

[52] K. O'Brien W.R Richards and D.C Miller. New air traffic surveillance technology. http://www.boeing.com/commercial/aeromagazine/articles/qtr_02_10/pdfs/AERO_Q2-10_article02.pdf.

[53] Norul Ridzuan Zakaria and Dato Nasri Nasron. The significance of a commercial suborbital spaceport to a modern small developing country and space city. 2010.