MajorTOM: Developing a tool suite to support regulation and monitoring of UK orbital risk

Samuel Diserens[(1)], Álvaro Menduiña Fernández [(1)], Samuel Frampton [(1)] , Charles Perera[(1)]

[(1)] UK Civil Aviation Authority, Aviation House, Beehive Ring Road, Crawley, West Sussex, RH6 0YR, UK
Email: Sam.Diserens@caa.co.uk

ABSTRACT

In July 2021 the UK Civil Aviation Authority (CAA) was appointed as the regulator for UK space activities with responsibility for monitoring spaceflight activities to protect public safety and national security and securing compliance with legislation, the conditions of issued licences and the UK's international obligations.

While the risks of missions are assessed during the licensing process, the dynamic nature of the orbital environment means that these risks are constantly evolving. To support in tracking and analysing the evolving risk the CAA's Space Engineering Team has developed MajorTOM (Tools for Orbital Monitoring).

This paper explores the development and use of MajorTOM, looking at the core infrastructure segment and the three currently supported three analysis tools: Orbital Manoeuvre Monitoring Suite (OMMS); Orbital Lifetime Analysis Suite (OLAS); and Conjunction Research & AnalysiS Helper (CRASH).

1 INTRODUCTION

Recent years have seen big changes in the orbital environment with massive growth in the space object population and the number of intact satellites in orbit increasing from 3,993 at the end of 2016 to 12,092 by the end of 2023 [1,2]. This rate of change has resulted in an increasingly complex environment for spacecraft to navigate leading to an increased need for co-ordination and oversight of orbital operations to manage the associated risks.

The UK's commercial space activity is primarily governed by the Space Industry Act 2018 (SIA) [3] and the Space Industry Regulations 2021 (SIR) [4], as well as the Outer Space Act 1986 (OSA) [5] for operations conducted by UK entities outside of the UK. The SIR came into force on 29th July 2021 and empowered the UK Civil Aviation Authority (CAA) as the regulator for space activities, independent of the UK government, with authority over the licensing and monitoring of spaceport, launch and return, range and orbital operations.

Since being established as the UK's space regulator the CAA has issued licences for the operation of 385 spacecraft and currently oversees the operation of 751 active spacecraft across LEO, MEO and GEO, including the OneWeb constellation of 654 satellites.

Section 2 of the SIA defines the primary duty of the appointed regulator to:

> *"exercise the regulator's functions with regard to spaceflight activities with a view to securing public safety"*

And to take into account a variety of other interests, including national security, the international obligations of the UK, and any space debris mitigation guidelines issued by an international organisation in which the government of the United Kingdom is represented.

Section 26 of the SIA defines that the regulator (now the CAA) is responsible for monitoring spaceflight activities, including:

> *"(a) securing compliance with the provisions contained in and made under this Act, the conditions of licences under this Act and the international obligations of the United Kingdom;*
>
> *(b) protecting public safety and the national security of the United Kingdom."*

To fulfil these responsibilities as the regulator the CAA's Space Regulation Team has a need for a range of Space Situational Awareness (SSA) products. These products can be used to develop insights into the behaviour and regulatory compliance of UK spacecraft in particular and the evolution of safety, sustainability, and security risks in the orbital environment in general.

The approach that has been adopted by the Space Engineering Team to meet this need is to develop a suite of Tools for Orbital Monitoring (MajorTOM), to analyse and report on the activities of UK missions and the environments they operate in.

2 GENERAL STRUCTURE

MajorTOM is deployed onto virtualised Microsoft Azure cloud infrastructure, allowing it to be scaled at need and enabling the automation of workflows through pipelines.

The majority of the development of MajorTOM has been using Python. This is the primary language of the CAA's Space Engineering Team due to the readability and ease of learning, the flexibility, and the extensibility offered by the many libraries.

One of the requirements for MajorTOM was for it to be easily maintainable and extensible so that it could be

Proc. 9th European Conference on Space Debris, Bonn, Germany, 1–4 April 2025, published by the ESA Space Debris Office
Editors: S. Lemmens, T. Flohrer & F. Schmitz, (http://conference.sdo.esoc.esa.int, April 2025)

adapted to meet the future needs of the UK CAA. Consequently, MajorTOM was designed to follow a modular pattern, with services divided between infrastructure and application layers.

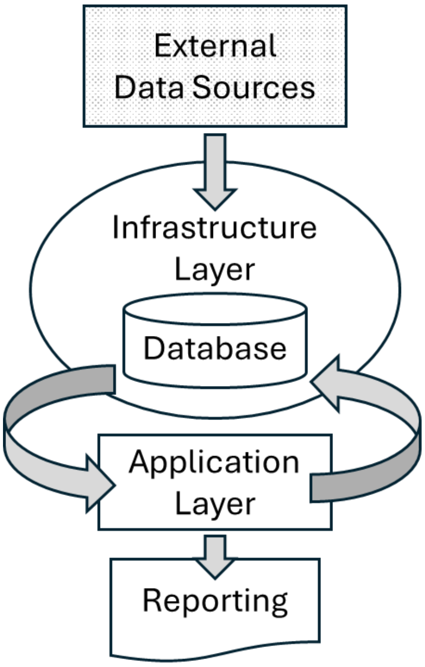

Figure 1: Top level structure of MajorTOM

The infrastructure layer is a key element to the modular approach and has three main components:

1) A database to store and organise data that has been retrieved for utilisation or produced by analysis services.
2) An interface library consisting of a package of functions to standardise and streamline interactions with the database by abstracting different basic and commonly performed tasks.
3) A tool to automate the retrieval of data and ingestion into the database from a range of sources, both internal and external.

The back end of the infrastructure layer is a SQL database and access and retrieval from the different data sources is achieved through their respective REST APIs.

The application layer currently consists of three independent services aimed at providing the CAA Space Team with different areas of situational intelligence about UK space missions and the evolving orbital risk environment:

- The Orbital Manoeuvre Monitoring Suite (OMMS) to track spacecraft manoeuvres and pattern of life.
- The Orbital Lifetime Analysis Suite (OLAS) to calculate and monitor the residual orbital lifetime of space objects for which the UK has liability.
- The Conjunction Research & AnalysiS Helper (CRASH) to analyse and map conjunction events and track changes to relative frequency and the locations they occur.

Reporting the outputs of these applications is currently a manual process. The development team is exploring options for automating reporting and contextualising the resulting knowledge through dashboards.

3 INFRASTRUCTURE

MajorTOM's infrastructure layer consists of the three previously identified components: the database, the interface library and the data retrieval tool.

By abstracting this functionality into the infrastructure layer MajorTOM enables the rapid prototyping and testing of new analysis services without the need to develop and maintain multiple solutions to the same problem, or for each engineer to acquire specific skills and knowledge around databases and APIs.

3.1 Database

MajorTOM's backend is a relational database using Microsoft Azure SQL Server. Monitoring data is stored in a number of linked tables. Fig. 2 shows how the core tables in the database are linked on different indexes to allow different records to be combined for later analysis.

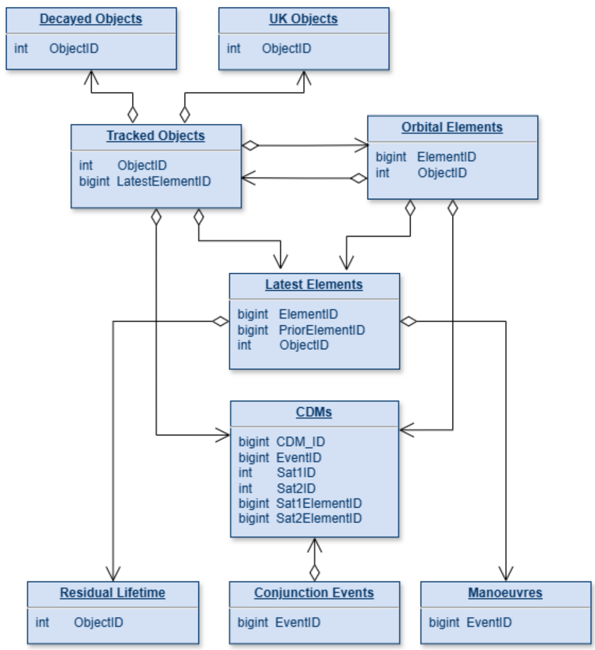

Figure 2: A graphical representation of the core tables and indexes in the MajorTOM database.

Using a cloud-based SQL database provides a scalable solution to storing growing volumes of data along with the flexibility of being able to efficiently sort, combine and extract data as required.

3.2 Interface Library

The interface library of Majoron utilises the PYODBC [6] Python library to initiate a connection to the SQL database. The package then uses either PYODBC or SQLAlchemy [7] to execute SQL queries and perform pre-defined actions, including bulk inserts, updates of data and other frequently used interactions.

This abstraction removes the need for individual engineers to develop specific skills with SQL or to have a detailed understanding of the structure of the database.

The library is then packaged into a Python wheel and published to an internal feed from which it can be downloaded and installed using Python package-management solutions. This approach allows the library to be imported into other applications and for new versions of the library to be easily published and rolled out to existing applications.

3.3 Data Retrieval

To populate the database MajorTOM includes a Python data retrieval application which interfaces with a range of different data sources, primarily through REST APIs, to retrieve SSA data. The interface library is then used to create or update records in the database as required.

The main data sources from which this data is drawn are:

- Space-Track.org [8] is a source of catalogued objects, orbital elements and Conjunction Data Messages (CDMs).
- ESA's DISCOS [9] provides more detailed characteristics of objects, including estimated dimensions and mass.
- UK Resident Space Object Database (RSO DB) is a separate CAA maintained database which provides a list of objects where the UK has liability, along with detailed characteristics of the spacecraft where available.

This utility is automated to run on a daily basis to reconcile the MajorTOM database with these external data sources in preparation for any required analysis.

4 OMMS

The Orbital Manoeuvre Monitoring Suite is a Python tool allowing the automatic detection and categorisation of spacecraft manoeuvres. This tool supports the CAA's monitoring of licensed spacecraft operators by providing the capability to detect, follow and understand manoeuvres as they happen and over time to analyse pattern of life.

Fig. 3 provides an overview of how OMMS progresses through the following process:

1) Retrieve orbital elements.
2) Calculate differences between orbital elements.
3) Detect manoeuvre using predefined thresholds.
4) Classify the manoeuvre (see Fig. 4).
5) Update satellite status.
6) Upload manoeuvre to database.

The key inputs to the OMMS process are orbital elements and a specified spacecraft fleet. The outputs are updates to the 'Manoeuvres' table in the database. The 'Latest Elements' table within the database provides a pair-wise set of orbital elements records for an object which are compared to determine if a manoeuvre has taken place.

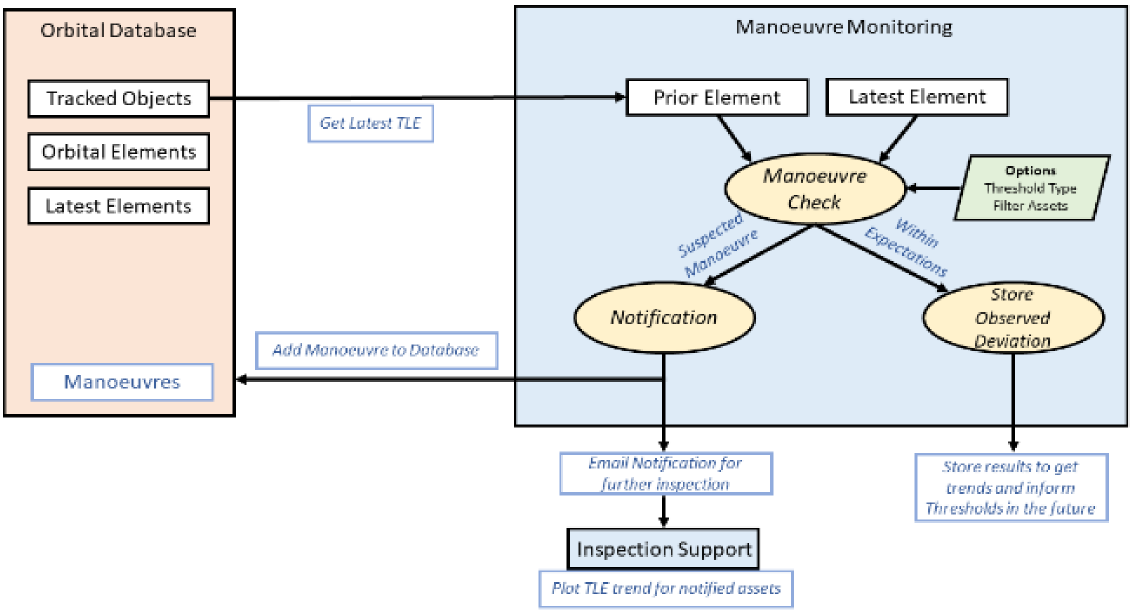

Figure 3: A process diagram of manoeuvre checking in OMMS

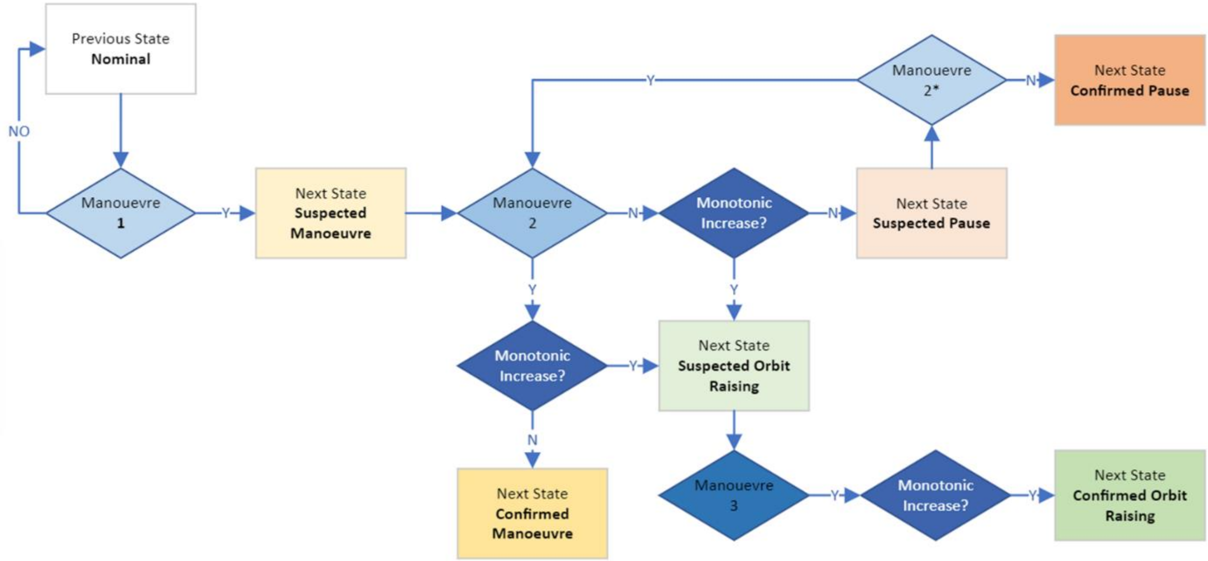

Figure 4: Decision logic for manoeuvre classification in OMMS

When comparing the orbital elements to identify a potential manoeuvre it is necessary to account for the natural orbital perturbations of the orbit and the uncertainty inherent in a measured position. These were accounted for by defining a set of thresholds above which a deviation in the orbit might be considered a manoeuvre:

Table 1: Manoeuvre detection thresholds

Parameter	Threshold
Inclination	0.05°
Semi-major Axis	2.0 km
Perigee	1.0 km
Apogee	1.0 km
Orbital Period	10 %

If a sufficiently large deviation in the orbit is detected, then the OMMS logic will look for any recent manoeuvre records for the object to attempt to classify the deviation as either: a Suspected Manoeuvre; a Confirmed Manoeuvre; a Suspected Orbit Raising; a Confirmed Orbit Raising; a Suspected Pause; or a Confirmed Pause.

Where a Pause is a break in an ongoing orbit raise. Fig. 4 outlines the decision tree used in this classification.

Currently the OMMS application is automated to run daily for a restricted fleet of spacecraft and can be launched on an as-needed basis for an expanded set of spacecraft.

4.1 Validation Results

The process and logic for OMMS has been tested against several UK objects known to be manoeuvring. Fig. 5 and Fig. 6 show the changes in apogee of two different OneWeb satellites where the OMMS logic has correctly detected a manoeuvre.

In the case of the manoeuvre shown in Fig. 5 a significant change in the apogee was detected as the satellite manoeuvred into its operation plane. For the scenario show in Fig. 6 the OMMS logic identified that there was an on-going change in the apogee across multiple data points and correctly classified it as the orbit raising phase.

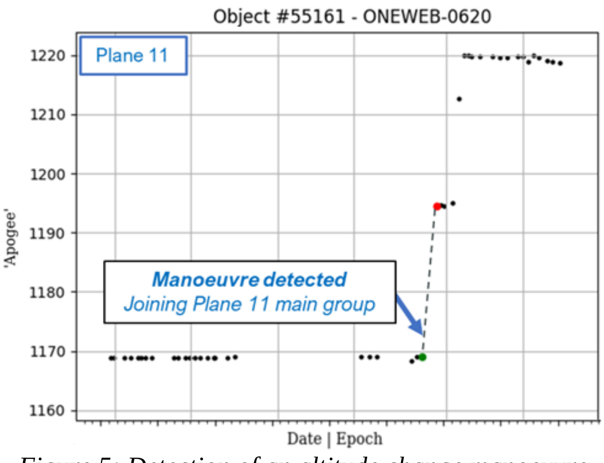

Figure 5: Detection of an altitude change manoeuvre

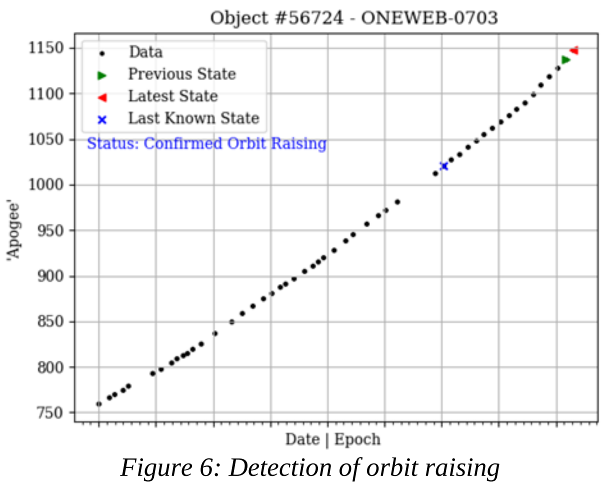

Figure 6: Detection of orbit raising

5 OLAS

OLAS, the Orbital Lifetime Analysis Suite, is a Python tool to assist the CAA Space Team in predicting and monitoring the remaining orbital lifetime of space objects.

This tool allows the CAA to track on-going compliance with space debris mitigation guidelines and offers insights into decisions made regarding past applications, enhancing the CAA's decision-making processes.

The procedure followed by the application is:

1) Take the catalogue of UK spacecraft from the UK RSO Database and filter out the objects in orbits which are too high to decay in a meaningful period.
2) Retrieve the latest orbital elements for the considered objects.
3) Retrieve latest solar and geomagnetic data and models and, configure the propagator accordingly.
4) Propagate the orbits to re-entry and compute the residual lifetime.
5) Update the database with he calculated lifetime and the relevant models used.

The propagation of the atmospheric decay and the calculation of the orbital lifetime is currently performed through integration with the OSCAR module of ESA's DRAMA software [10]. However, the propagation interface is designed to be modular to allow future integration with other tools, such as NASA's DAS [11], CNES's STELLA [12] or an internal tool.

To perform the propagation, it is necessary to pass the propagator a set of input parameters, including the initial spacecraft orbital parameters, the mass and cross-sectional area of the spacecraft, and the solar & geomagnetic data to be used. As such, one of the capabilities of the tool is to utilise the multiple different options for solar activity that OSCAR provides as well as retrieving the most up-to-date ESA predictions. The model used is then stored within the lifetime record in the database.

One of the ways in which OLAS provides enhanced awareness is that, for recently licensed UK spacecraft, the CAA can make use of operator provided data. The UK RSO database contains details on the mass and cross-section of the spacecraft from the licensing phase. This enables more accurate predictions of the orbital lifetime.

OLAS is intended to be run on a monthly basis to generate a record of how the expected residual lifetime changes over time. This residual lifetime can be compared against expectations, such as re-entry within 25 years. For disposed spacecraft the estimate can be combined with the known disposal date to calculate total disposal time and monitor the effectiveness of attempts to comply with disposal guidelines.

It is hoped that the data generated by OLAS on the will provide insights into the sensitivity of lifetime predictions by analysing the changes in the predicted residual lifetime across a large sample size. These insights would help inform future licensing and monitoring decisions made by the CAA by allowing the uncertainty of lifetime predictions to be better understood.

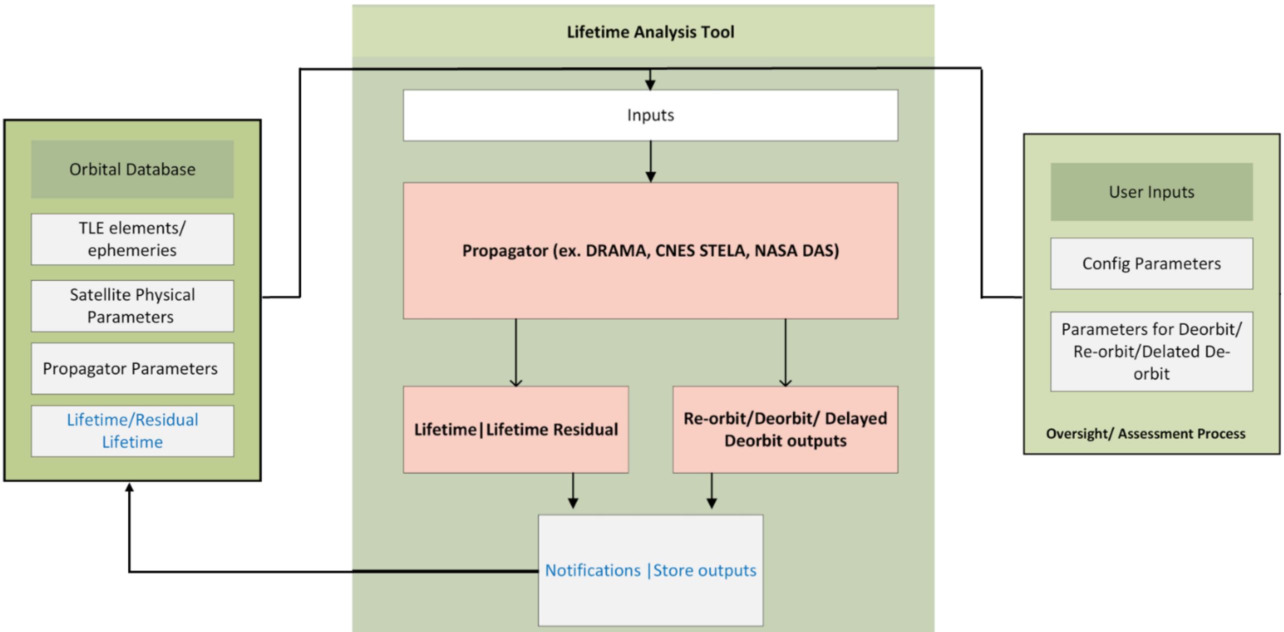

Figure 7: The process for monitoring residual orbital lifetime

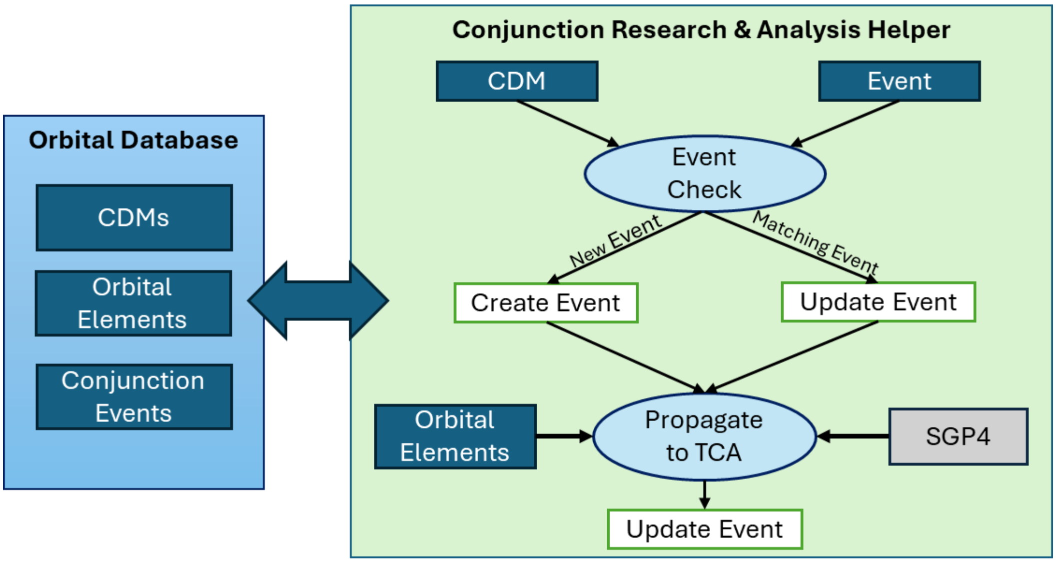

Figure 8: The process flow followed by CRASH to analyse conjunction events

6 CRASH

The Conjunction Research & AnalysiS Helper, CRASH, is a Python application for processing, analysing and mapping conjunction events.

The information generated by CRASH supports the CAA's understanding of the evolving orbital risk environment in order to improve licence assessment and licensee monitoring. The capability enables the tracking of changes to the relative frequency of conjunction events and the locations they occur.

The logic in CRASH follows the below process flow:

1) Retrieve unprocessed CDMs from the Orbital Database.
2) Cross-check CDMS against previously identified conjunction events by Object IDs and Time of Closest Approach (TCA) and either:
   a) Create new conjunction event, or
   b) Update existing conjunction event.
3) For each conjunction event determine whether position data is available for the conjunction.
4) Retrieve most recent orbital elements for both objects where conjunction location is not available.
5) Propagate both sets of orbital elements to TCA.
6) Update conjunction event records with the average position of the two objects and their relative velocity.

For a sequence of CDMs relating to a single conjunction event it is expected that the TCA fluctuate as knowledge on the position of the objects is updated and new TLEs are generated. To account for this fluctuation when attempting to match a CDM to an existing conjunction event record the approach in CRASH is to filter for events within five minutes either side of the CDMs TCA. This amounts to a segment roughly 10% of a circular orbit for a spacecraft in Low Earth Orbit (LEO) and should be sufficient to capture variations in TCA while preventing inclusion of multiple distinct events. The exception to this will be a very small sub-set of objects in close proximity coplanar orbits which may experience multiple conjunctions per orbit.

The propagation of the orbital elements sets to TCA is performed using an implementation of SGP4 contained in a Python package [13] which compiles David Vallado's official C++ code from CelesTrak [14de from CelesTrak [14]. SGP4 is the intended propagator for the GP data class of orbital elements provided by Space-Track.org so this method should provide the most comparable position data to that used in the conjunction screening that generated the original CDM.

CRASH is intended to be run on a daily basis in order to keep up with the volume of CDMs that are being generated.

7 FUTURE EXPANSION

The anticipated future expansion work on MajorTOM can be divided into three categories: enhancements to the existing applications; development of additional applications; and an overhaul of the reporting functionality.

A number of enhancements to OMMS, OLAS and CRASH have already been identified, some of which are mentioned within this paper. One of these is the planned expansion of the OLAS functionality to use a selection of different propagators for comparison. Another is to integrate the use of special ephemerides into the OMMS manoeuvre detection algorithm.

Some exploration has been made into what additional data is available that could be used to enhance the CAA's monitoring of operator behaviours. Some of the potential options are:

- Monitoring the active participation of operators in sharing data by analysing the frequency of ephemeris submissions
- Automatically checking the availability of contact details to enable conjunction co-ordination
- Tracking updates to the listed status and manoeuvring capability of spacecraft and cross-checking with the UK RSO database.

The final key area for expansion is the reporting functionality of MajorTOM. The intention is to simplify the process of accessing the data products of MajorTOM by developing an internal web-portal interface, including reporting dashboards. This would allow additional analysis runs to be triggered manually as well as scheduled and would lower the bar to access, allowing a greater number of users to access and use the data.

8 SUMMARY

This paper has addressed the motivation of the CAA in creating the MajorTOM suite of tools to help fulfil its duties as the UK Space regulator under the SIA and the SIR. The modular structure of the solution was discussed, with the division of services between infrastructure and applications layers, with the infrastructure layer comprising database, interface library, and data retrieval service.

Three core application were presented with an overview of the process flows used and the purpose of the tools. These applications were OMMS, for tracking spacecraft manoeuvring against expectations and understanding patterns of behaviour through life; OLAS, for automating the calculation and recording of orbital lifetime for monitoring compliance with disposal guidelines and the accuracy of lifetime predictions; and CRASH, for aggregating, monitoring and mapping conjunction events to understand the evolving collision risk environment.

MajorTOM is expected to continue to expand over future years to meet the needs of the CAA and the end of this paper put forward several different areas in which this development is likely to focus.

9 ACKNOWLEDGEMENTS

The space situational awareness data used in this work is freely provided by the U.S. Government through Space-Track.org and ESA through DISCOS.

10 REFERENCES

1. ESA Space Debris Office (2017). ESA's Annual Space Environment Report. Issue 1. ESA/ESOC. GEN-DB-LOG-00208-OPS-GR. 27 April 2017
2. ESA Space Debris Office (2024). ESA's Annual Space Environment Report. Issue 8. ESA/ESOC. GEN-DB-LOG-00288-OPS-SD. 19 July 2024
3. UK. Space Industry Act 2018. Chapter 5. London https://www.legislation.gov.uk/ukpga/2018/5
4. UK. The Space Industry Regulations 2021. 2021 No. 792. London. https://www.legislation.gov.uk/uksi/2021/792
5. UK. Outer Space Act 1986. Chapter 38. London https://www.legislation.gov.uk/ukpga/1986/38
6. Michael Kleehammer. Pyodbc (2024). GitHub. https://github.com/mkleehammer/pyodbc/wiki
7. Bayer, M. (2012). SQLAlchemy. In A. Brown & G. Wilson (Eds.), The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks. aosabook.org. Retrieved from http://aosabook.org/en/sqlalchemy.html
8. 18th Space Control Squadron: Space-Track. Online. (2025). https://www.space-track.org. Accessed 12 Feb 2025
9. McLean, F., Lemmens, S., Funke, Q., & Braun, V. (2017, April). DISCOS 3: An improved data model for ESA's database and information system characterising objects in space. In Proc. *7th European Conference on Space Debris* (Vol. 11, pp. 43-52).
10. Braun, V., Funke, Q., Lemmens, S., & Sanvido, S. (2020). DRAMA 3.0-Upgrade of ESA's debris risk assessment and mitigation analysis tool suite. *Journal of Space Safety Engineering*, *7*(3), 206-212.
11. Liou, J. C., Opiela, J. N., Vavrin, A. B., Draegar, B., Anz-Meador, P. D., Ostrom, C. L., & Sanchez, C. (2019). Debris Assessment Software User's Guide: Version 3.0 (No. JSC-E-DAA-TN70236).
12. Goester, J. F., Bellucci, A., & Fremeaux, C. (2021). CNES tools and methods for de-orbit and re-entry trajectories evaluation. In Proc. *9th European Conference For Aeronautics And Space Sciences (EUCASS)*
13. Brandon Rhodes. Python-SGP4 (2025). GitHub. https://github.com/brandon-rhodes/python-sgp4
14. Vallado, D., Crawford, P., Hujsak, R., & Kelso, T. S. (2006, August). Revisiting spacetrack report# 3. In Proc. *AIAA/AAS astrodynamics specialist conference and exhibit* (p. 6753). https://celestrak.org/publications/AIAA/2006-6753/