TASKING SOFTWARE OF THE ESA FLYEYE TELESCOPE

M. Hübner[(1)], K. Mokos[(1)], M. Rasotto[(1)], K. Davies[(1)], P. Bohn [(1)]

[(1)] Terma GmbH, Bratustraße 7, 64293 Darmstadt, Germany, Email: {mahr, knm, mras, ked, prb}@terma.com

ABSTRACT

The ESA Flyeye telescope aims at continuously scanning the sky for Near Earth Objects (NEOs), making it an important part of the ESA's Space Situational Awareness programme. Important goals include maximizing the area that can be covered within a nightly observation period and to minimize the number of consecutive observation periods for achieving maximum possible sky coverage. Thus, the optimization of the telescope observation schedule is critical.

The Flyeye telescope Tasking Software is responsible for planning and optimizing the Flyeye schedule as a sequence of nightly observations to be made by the telescope. The fundamental logic behind the Tasking Software, based on the concept of opportunity modelling, is described in [1]. The concept was used previously in the development by Terma of a similar system for a space-borne telescope of the Canadian Department of National Defence's Sapphire mission.

The Tasking Software is an example of a planning application whose driving requirement is to optimise the efficiency of use of the Flyeye telescope in surveying the night sky by maximising the area of the sky observed each night. On top of this driving requirement, the Tasking Software takes into account other factors including areas of the sky observed by other telescopes and the Flyeye telescope itself in the past days, Moon brightness at the observed location, Sun angle and galactic (Milky Way) brightness.

Besides its primary purpose for surveying the night sky, the Tasking Software also allows the operator to include follow-up observations at the expected locations of specific NEOs of interest, specific locations in the sky, and for performing telescope calibration observations. Lastly, the Tasking Software can handle physical restrictions that inhibit observations at certain locations ("exclusion zones").

The Tasking Software is currently in the operations preparation phase and has benefitted from feedback from Flyeye integration and test engineers, as well as the end users, the astronomers who will be using the Flyeye telescope in the future.

This paper presents how the Tasking Software meets the diverse needs, describing the software techniques, mechanisms and third-party open-source libraries that have been used to provide astronomers with a highly configurable and adaptable tool, which remains straightforward to use for routine survey operations through its web-based GUI.

1. INTRODUCTION

The currently in-development ESA Flyeye telescope will continuously take observations of the night sky, producing hundreds of images each night that will be automatically processed with the purpose of detecting Near Earth Objects (NEOs).

The Tasking Software is used to produce the schedule of observations in the form of commands that the Flyeye telescope control system executes each night. From its early conception, driving requirements on the Tasking Software were defined which characterise the capabilities of the Tasking Software today. Key requirements include:

- Maximizing the area that can be covered within a given time-period in order to minimize the time for achieving a full sky coverage. The target is to be able to make a complete survey of the visible sky every 3 to 5 days.
- Taking into account observations made by the Flyeye telescope itself and other telescopes in the preceding days, to ensure priority would be given to observing locations that had not recently be observed.
- Taking into account the Moon and Milky Way brightness at observation locations.
- Taking into account exclusion zones, where the telescope must not be commanded to point to or slew through.
- Ensuring a high level of configurability and flexibility in order that the Tasking Software can be easily adapted for other telescopes.

Although the priority for the Flyeye telescope is for routinely surveying the sky, the Tasking Software needed to have to capability for including follow-up observations of NEOs, user-specified observation locations, and calibration images.

An informal guide to the design and implementation approach of the Tasking Software was also to "keep it simple", in order to avoid overt complexity that might be difficult to maintain over the expected long lifetime of the Flyeye telescope.

In this paper some of the novel techniques that were utilised to meet the diverse needs are described.

The term 'merit figure' or 'figure of merit' in this paper is used in relation to the weighting factor that is given to each parameter used by the Tasking Software in determining the optimum observation to be made at any time. More information on the schedule generation and optimisation concept and how it is implemented is given in [1].

Proc. 2nd NEO and Debris Detection Conference, Darmstadt, Germany, 24-26 January 2023, published by the ESA Space Safety Programme Office
Ed. T. Flohrer, R. Moissl, F. Schmitz (http://conference.sdo.esoc.esa.int, February 2023)

2. OBSERVATION LOCATIONS

The primary purpose of the Tasking Software is to produce an optimised plan of observations for ESA's Flyeye telescope, such that a complete survey of the observable night sky can be made in the shortest time. However, it is also a goal that the software is capable of being easily adapted for other telescopes. This is achieved by parameterising the variables that define the characteristics of a telescope from the perspective of the observations that it is capable of making. The tessellation grid is one such input to the Tasking Software that defines every potential observation location in terms of pointing location and field of view of the telescope.

The tessellation grid can be adapted without requiring changes to the Tasking Software, allowing users of the Tasking Software to explore different observation geometries. At the start of the development of the Tasking Software, it was assumed that the telescope camera would be mounted square to the horizon, and a tessellation grid was defined as shown in figure 1, which shows the Mollweide projection of the celestial sphere overlaid with the tessellation grid.

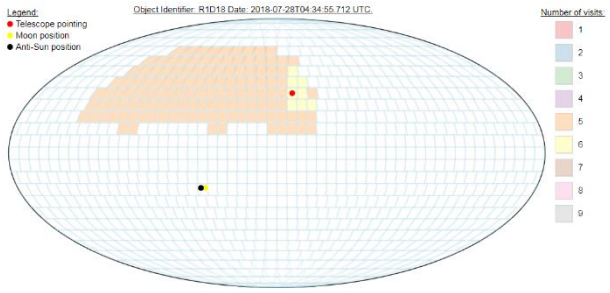

Figure 1. Tessellation Grid Version 1

Later, it became known that the Flyeye camera is mounted at 45° to the horizontal. Analyses were made by the Tasking Software developers and astronomers at ESA to find suitable tessellation grids that traded off high levels of sky coverage with few gaps (which requires a high degree of overlapping of tessellation cells) against grids with more gaps but less overlapping of cells. A tessellation grid with a high degree of overlapping cells will result in less efficient use of the telescope since it will be observing many overlapping locations. This would result in a longer time needed to survey the complete night sky. An example of the overlap of observation areas is shown in Figure 2, which shows a close-up of the tessellation cells over the North Pole.

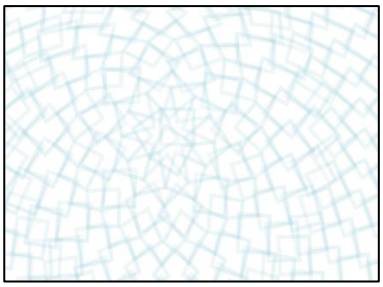

Figure 2. Overlapping Observation Cells at North Pole

Figures 3 and 4 show examples of the evolution of the tessellation grid designs, where different approaches were tried to find the optimum trade-off between the amount of overlapping and the total area of the regions that would never be observed (gaps).

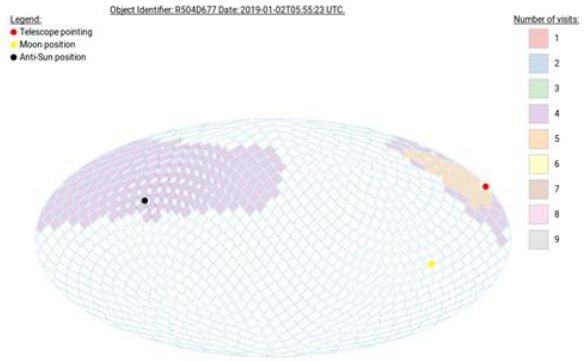

Figure 3. Tessellation Grid Version 2

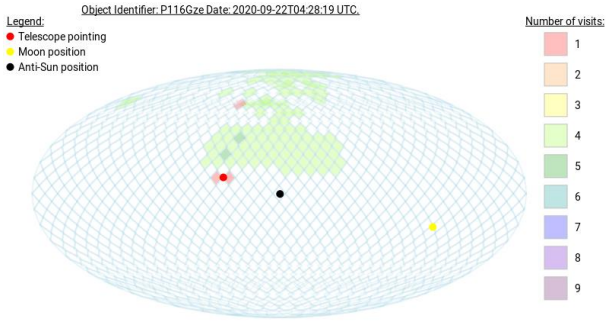

Figure 4. Tessellation Grid Version 3

By having the user-defined tessellation grid as the starting point for the planning the observations, the Tasking Software is able to support telescopes with different optical characteristics, such as field of view, without modification.

3. SKY COVERAGE

In order for the generation of an observation schedule to take into account observations made by the Flyeye and other telescopes in the previous days, the Tasking Software imports sky coverage information once per day. For the Flyeye telescope, the coverage information is provided by the Flyeye Data Processing Chain (DPC),

which determines whether each observation location has been successfully observed (a minimum of four adequate images of a location during the night are needed for the location to be considered to have been observed). There is a one-to-one match between the coverage areas reported by the DPC and the observation areas defined by the tessellation grid, therefore checking whether a location has been observed previously is trivial.

In the case of sky coverage data from other telescopes, the problem is more complex. Reported sky coverage information from other telescopes is retrieved daily from the IAU Minor Planet Center [3]. An example of sky coverage data available from the MPC is shown graphically in figure 5. It can be seen in figure that external observations vary in size, shape as well as coordinates. It is difficult to match the tessellation cells of the Flyeye telescope with the external observation locations. This difficulty is further exacerbated by the fact that the Flyeye tessellation grid cells are tilted by 45°.

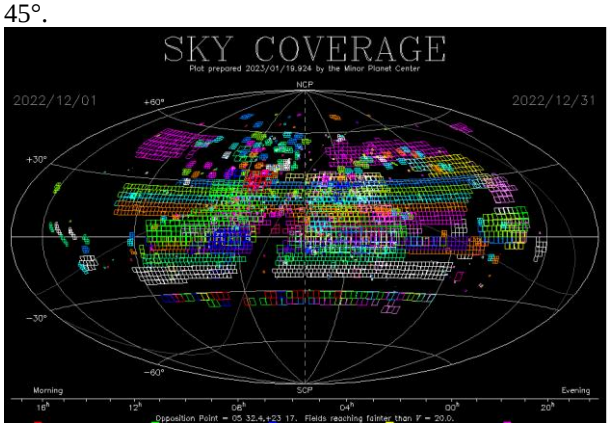

Figure 5. Sky Coverage Data from Minor Planet Center

The locations observed by other telescopes can be at any location and with varying (and narrower) fields of view. To recognize which tessellation cells of the grid have been previously observed, the Tasking Software has to organise and process the imported sky coverage data, which it does in two steps.

In a first step, all imported external observations are grouped by date. The individual observations for each date are then merged to create contiguous areas for each night in the specified time frame, using a built-in Java package, java.awt.geom [4]. This package provides the Java classes for defining and performing operations on objects related to two-dimensional geometry.

In a second step, for each tessellation cell that lies fully within one of the merged areas, the corresponding survey object's figure of merit is adjusted relative to the time since the date the external observations. More recent external observations result in a lower figure of merit, resulting in the area being less likely to be included in the observation schedule.

4. MOON BRIGHTNESS

The Tasking Software takes into account the Moon brightness at each location in order to select the locations to be observed. An initial, overly simple, approach was to apply a merit figure based on the angular distance between each observation location and the Moon. This was subsequently improved by implementing an algorithm which also takes into account the phase of the Moon, as described in the master thesis paper of P. Kollo [2]. Figure 6 shows the parameters involved in the calculation. All indicated distances are angular distances.

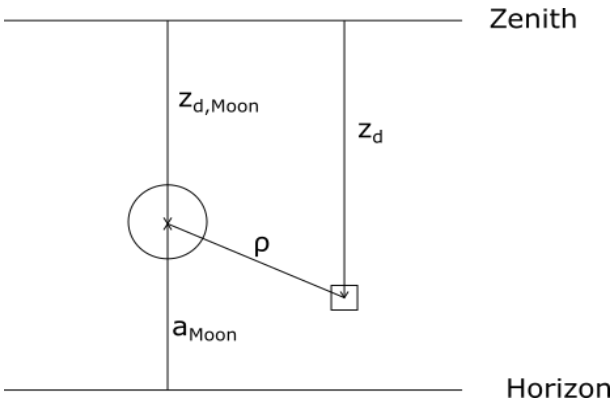

Figure 6. Moon Brightness Calculation Parameters

The brightness impact of the Moon on a location defined by its tessellation cell is a function of the angular distance between the Moon and the centre of the tessellation sky cell (ρ), the phase angle of the Moon (α), the zenith distances of the Moon and of the tessellation sky cell and the extinction coefficient.

Testing of the algorithm subsequently revealed that it was effective until the angular distance to the Moon became small, at which point it was sufficient to simply avoid observations for locations that were less than a certain (configurable) angular distance from the Moon.

5. GALACTIC BRIGHTNESS

The Milky Way brightness is defined as a fixed 2-demonsional (2D) model, with the brightness value calculated for each tessellation cell. The calculation of the Milky Way brightness is based on the master thesis paper of P. Kollo [2]. Figure 7 shows a representation of the Milky Way brightness model, where the brightness value at a location is represented by a colour scale.

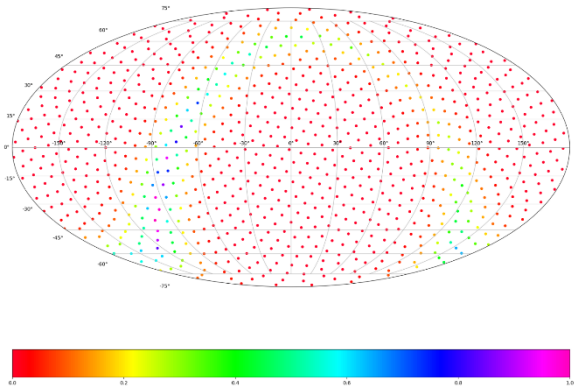

Figure 7. Milky Way Brightness 2D Model

Since the 2D model is fixed, the brightness value for each observation location can simply be added to each observation location in the tessellation grid.
For additional flexibility, the user is provided with a configurable merit factor to adjust the impact of galactic brightness values on observation location selection.

6. EXCLUSION ZONES

The Tasking Software provides for the definition of custom exclusion zones to prohibit the telescope from pointing to specific locations (no-go zones), or to constrain the telescope's motion when moving from one area to another (no-slew zones).
Exclusion zones are defined by the user as a set of coordinates forming a polygon, either using equatorial coordinates (RA-Dec) and horizontal coordinates (Az-El). However, that since all the underlying computations in the Tasking Software's scheduling algorithm are performed using equatorial coordinates, a conversion might be needed in case horizontal coordinates are used for defining an exclusion zone. In detail, the process of converting a sky region from horizontal to equatorial coordinates requires careful handling. Converting the coordinates of each polygon vertex might lead to incorrectly converted areas, despite the correctness of the transformation for an individual point. This is due to the limited domain of trigonometric functions involved in the transformation, which might cause an angle to be wrapped around the admissible domain and therefore alter the distance between one vertex and the others, that is, the sides of the polygon and consequently the polygon itself.
A more robust approach has therefore been adopted, based on converting the centre of each cell of the tessellation grid into horizontal coordinates (by default provided in equatorial coordinates) and building an area around it which approximates (in horizontal coordinates) the original sky area. The level of approximation depends on the size of the cells used in the tessellation grid. Larger cells lead to a larger approximation error. Each area is then directly compared with the polygons defining the exclusion zones using the Java library java.awt.geom [4], which conveniently provides Java 2D classes for defining and performing operations on objects related to two-dimensional geometry. If complete or partial overlappings are found, then the corresponding original cell is identified as belonging to an exclusion zone and the information is stored in a map. It is worth highlighting that due to the time dependency of the coordinates' transformation, these computations are performed at each time step.
During the scheduling computations, for each candidate object and for each time step, the cell in which the object is belonging to is first identified and then compared with the previously generated map of no-go zones. Similarly, for no-slew zones, the algorithm first computes the telescope trajectory between one point in the sky and another, and then checks the no-slew zone map for possible intersections. In case of positive matches, the corresponding merit figures are set to zero, thus ensuring these locations are not selected for inclusion in the observation schedule.

7. NEO FOLLOW-UP OBSERVATIONS

Although primarily designed for planning optimised sky surveying, the Tasking Software also allows the operator to include specific observations of NEO follow-up objects, or a specific user-defined location. Lists of follow-up objects are imported from the NASA JPL Scout NEOP Hazard Assessment web server [4] and from the ESA near-earth objects coordination centre (NEOCP) "Risk List".
Once imported, the operator can adjust priorities of follow-up objects and other merit factors and generate the schedule until satisfied with the time of inclusion of the follow-up observation in the schedule. Inclusion of a follow-up observation is not guaranteed since the Tasking Software is always trying to optimise the use of the telescope, taking into account all factors such as the time to slew the telescope to the observation location.
Examples of merit factors that the operator can adjust include a factor for the apparent magnitude of the object to be observed.

8. SCM STANDARD

The schedule of commands generated by the Tasking Software and sent to the Flyeye Telescope Control System for execution conforms to the Scheduling and Commanding Message (SCM) Standard [5]. The standard is a new CEN/CENELEC standard for tasking telescopes for NEO and SST purposes. It defines the data as an XML document compliant to the schema defined by the standard. The Tasking Software can in principle support any telescope which utilises the SCM standard.

9. PROGRAMMATIC INTERFACE

The Tasking Software is implemented as a server application with a web-based user interface (UI), which allows it to be operated using a standard web browser.
The web user interface is implemented using Representational State Transfer (REST), which allows the exposed interfaces to be used by any application that supports such an interface. Most modern programming languages have libraries that provide support for REST interfaces, which means that Tasking Software can be controlled programmatically as well as through the web interface. Examples of how this approach is beneficial are:

- Testing the application can be automated by scripts using the same functions that the user interface uses. Testing is done using Python scripts, but any suitable scripting language could be used.
- Users are able to write scripts to drive the Tasking Software remotely, for example, using Python scripts in Jupyter Notebooks.

10. LESSONS LEARNED

The technologies used for the development of the Tasking Software (including Spring Boot [7] and Angular [7]) allowed for early prototyping and rapid development. Coupled with a continuous integration / continuous deployment (CI/CD) approach for delivering the software to ESA, this meant that users were given working versions of the Tasking Software from the early stages of its development, and iterative updates later as functions were added. This allowed feedback to be provided to the developers at the time of the actual development, particularly in the area of usability. As an example, changes were made in the layout of the web UI, which users requested in order to make their workflow using the Tasking Software more natural. The current web UI is shown in figure 8 below.

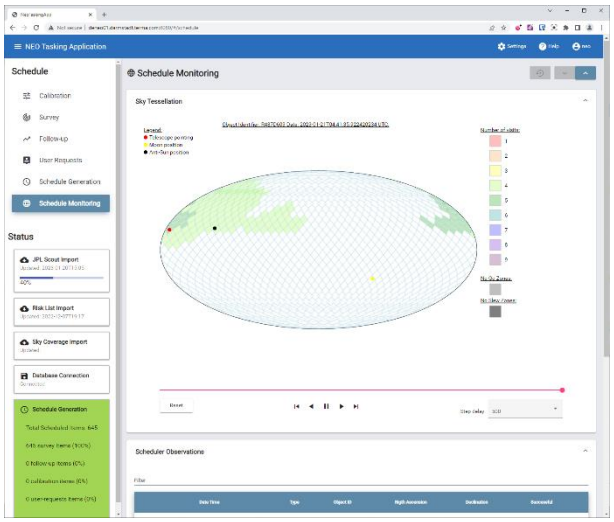

Figure 8. Tasking Software Web UI

With the technologies used, such changes could be made rapidly, and quickly deployed, keeping the feedback loop between the software developers and the users short.

11. CONCLUSIONS

The Tasking Software provides a highly flexible tool to astronomers for the scheduling of telescope observations that combines pragmatism of function and operation with the use of modern and straight-forward-to-use open-source software.

12. REFERENCES

1. Mokos, K., Davies, K., Pedersen, J.S., Bohn, P., Gad, R. (2019). An Opportunity Model Strategy for Scheduling NEO Observations.
2. Kollo, P. Investigation of Survey Strategies for the Observation of Near-Earth Asteroids with ESA's Fly-Eye Telescope Master Thesis, ESA-SSA-NEO-RP-0188.
3. IAU MPC Sky Coverage, online https://www.minorplanetcenter.net/iau/SkyCoverage.html
4. Java 2D classes package java.awt.geom, online https://download.java.net/java/early_access/loom/docs/api/java.desktop/java/awt/geom/package-summary.html
5. SCM - Scheduling and Commanding Message - Standard, CSN EN 17350
6. Spring Boot framework, online https://spring.io/projects/spring-boot.
7. Angular framework, online https://angular.io/.