WEBPLAN - WEB-BASED SENSOR SCHEDULING TOOL

Agnieszka Sybilska[(1)*], Mariusz Słonina[(1)], Paweł Lejba[(2)], Grzegorz Lech[(1)], Johannes Karder[(3)], Tomasz Suchodolski[(2)], Stefan Wagner[(3)], Piotr Sybilski[(1)], Alexandru Mancas[(4)]

[(1)] Sybilla Technologies, ul. Toruńska 59, 85-023 Bydgoszcz, Poland, agnieszka.sybilska@sybillatechnologies.com
[(2)]Centrum Badań Kosmicznych Polskiej Akademii Nauk, ul. Bartycka 18A, 00-716 Warszawa, Poland
[(3)] University of Applied Sciences Upper Austria, Roseggerstraße 15, 4600 Wels, Austria
[(4)] ESA/ESOC Space Debris Office (OPS-GR), Robert-Bosch-Str. 5, 64293 Darmstadt, Germany
** Corresponding Author*

ABSTRACT

WebPlan, a web-based sensor scheduling tool developed for the European Space Agency (ESA), is a marketplace and a meeting platform for SST data consumers (users) and data producers (sensors), with a capacity for up to 100-1000 sensors provisioning data every day and capable of <30 minutes request-to-delivery times, limited only by the provisioning sensors capability for reaction and data processing.

WebPlan is a large (over 50 services in the production environment) and complex (5 entities involved in the longest chain of tasking) distributed service based on Docker virtualization. It incorporates an advanced scheduling heuristic algorithm which performs around 30% better than other compared algorithms. It includes multiple custom scheduling agents executing specialized scheduling tasks for, e.g. GNSS calibrations, GEO tracking, high interest events, re-entry or collision avoidance data provisioning. Support for multiple observing campaigns and programs is possible with easy book-keeping and results delivery.

The system has been operational at Sybilla Technologies (ST) since Dec 2019, with 43 sensors connected at peak usage and currently on average 16 sensors fully booked at any given time. As of March 2021, over 178 000 timeslots have been booked through WebPlan which translates into ca. 45 000 observing hours. Nearly 2.5 million FITS files have been gathered and over 140 thousand Tracklet Data Message (TDM) files produced (including only data that have gone through the WebPlan system once acquired).

1 INTRODUCTION

Scheduling of astrophysical observations from the ground and from space is a challenging task, with SST observations typically being even more demanding due to unusual (when compared to typical stellar or galactic type objects) tracking or visibility requirements (speed, illumination conditions, presence of retroreflectors etc.).

Scheduling is partly automated at most larger observatories but nevertheless schedulers still often act only as assistants for human decisions, predominantly in the short-term planning and last-minute changes/decisions and dynamical adjustment of priorities (see e.g. [1]). For a detailed presentation of a scheduling problem and its constraints as well as different tools and algorithms currently used in scheduling at major observatories, please see e.g. [2]. None of the currently available schedulers is fully independent of human intervention nor is able to handle multiple sensors and dynamically adjust priorities.

Through ESA's Space Situational Awareness (SSA) Programme technologies are developed to maintain catalogued information of the population of man-made space objects. To achieve this task, up-to date data on these objects is crucial. Planning of new observations in short and long term and monitoring their execution is essential for most SST-related services and functionalities.

Scheduling of a distributed system with limited resources and dynamically changing environment is a well-known issue. SST observation planning is particularly challenging due to the relatively quick changes in the position of objects in orbit, coupled with a limited number of observation stations and their locations available. In addition, certain limitations need to be taken into account depending on the chosen observation technique, for example:

- observations possible only at night (passive optics),
- avoidance regions (Sun or Moon position, aircraft),
- geographical distribution of the stations,
- station hardware restrictions (limited field of view),
- optical range (passive optics) or range (SLR and radar),
- weather characteristics / station availability.

The planning tool must ensure that the given sensor parameters vary depending on the technique. For example, for the SLR it will be important to determine power restrictions as well as account for the availability of day/night observations. For optical observations this

Proc. 8th European Conference on Space Debris (virtual), Darmstadt, Germany, 20–23 April 2021, published by the ESA Space Debris Office
Ed. T. Flohrer, S. Lemmens & F. Schmitz, (http://conference.sdo.esoc.esa.int, May 2021)

will be e.g. the exposure time or the need to use the selected filter.

In addition, an appropriate observation planning strategy should be adopted, among which we can highlight: planning based on time, observation priorities or based on deterministic or stochastic planning (i.e. static scheduling in which all data is known in advance, or scheduling taking into account changing conditions where estimates are available).

An efficient, easy-to-use and automated process of tasking various sensors allows one to tap into a potentially very large source of existing observatories, which could possibly be available for SST observations, yet at the same time presenting a challenge of integrating various sensor types. Such sensors will have different levels of automation and autonomy executing their existing programs, which will have higher or lower priorities than the priorities of currently needed observations.

The WebPlan sensor scheduling tool comes as a response to the above challenges. The following sections provide details on the scheduling algorithms behind WebPlan, the system's architecture, web user interface as well as its current applications.

2 SCHEDULING

The WebPlan system successfully addresses the following scheduling challenges:

- available computation time and resources (the algorithm has to perform on "a modern laptop"),
- multiple objectives (minimizing error envelope, minimizing cost),
- dynamic changes (weather, failures, other players),
- benchmark data to evaluate scheduling algorithms (a set of rules prepared by experts).

The scheduling algorithm is tasked with the following:

- minimization of idle time and time overheads (cost increases when the booking is made late),
- maximization of goals (for example keeping the error envelope for all objects as low as possible within the budget limits),
- maximization of observation quality (signal-to-noise ratio, astrometric, photometric, and range precision),
- minimization of cost (observing time),
- comparison problem of radar, laser, optical data, quality, cost, creation of example cost functions,
- clustering effect and single or multiple observations impact on error envelope.

2.1 Testbench

The purpose of the Testbench is to provide an algorithm simulation environment resembling a real-life environment but simplified in order to speed up calculations and allow for easier modifications. The Testbench contains the following parameters:

- weather information (rain, low cloud base, relative humidity, wind speed, temperature),
- sensor data (location, type, limits, etc.),
- information about objects with orbital propagation,
- allowable error envelope values and alert times per object or orbital regime,
- probabilistic description of random events based on Mean Time Between Failures (MTBF) and Mean Time To Recovery (MTTR),
- goal function,
- scheduling strategy,
- cost, wallet, cashier functionality.

During the simulation setup, a sensor takes as input historical weather data from a 1-year span, either taken directly from ST-operated observatories or obtained from commercial providers.

Each sensor is described by a configurable list of failures and their corresponding MTBF and MTTR. Based on those statistics, the simulation decides randomly which of the potential failures are active in a given time slot. If at least one failure is active, the sensor cannot observe. System availability is then calculated as the percentage of time the system is operational, i.e. able to perform science observations.

Each target in the simulation has an associated accuracy envelope. Each envelope is described by its volume (size) – the lower the volume, the better the accuracy. Each envelope has its alert time and alert size, which is the size the envelope would have after alert time with no tracklets (assuming an initial zero error).

To properly estimate the increase of position errors in time we used historical TLE data obtained from the Celestrak database[1]. TLEs presented daily by JSpOC generally follow the real evolution of orbital parameters of the catalogued objects in time. Large statistics allows for removal of observational effects and close-to-real-life estimation of orbital parameters' changes.

In order to assess the influence of added observations of various types (i.e. optical, laser, radar) on orbital position uncertainty, we performed simulations using a subset of objects from the TLE catalogue.

The testbench was used in scheduling algorithms testing as described in more detail in the following subsection.

[1] https://www.celestrak.com/NORAD/elements/

For this purpose a set of 100 objects was selected for monitoring with 22 sensors (including passive and active optical and radars) located worldwide. The targets were selected such that their visibility from the sensor sites was maximized and all orbital regimes (LEO/MEO/GEO) were covered.

2.2 Catalogue maintenance algorithms

Algorithm testing and selection was performed with the use of HeuristicLab[2] [3], a paradigm-independent and extensible software environment for heuristic optimisation. HeuristicLab features various machine learning algorithms, as well as numerous population-based and trajectory-based (meta-) heuristic optimization algorithms for single- and multi-objective (SO/MO) optimization.

To gain first results, the testbench described in Section 2.1 is used in a simulation-based optimization run. Since schedules are evaluated according to two objectives, i.e. the number of used credits and the envelope accuracy, the first algorithms to be evaluated will be multi-objective ones, namely NSGA-II and G-SEMO. Whereas NSGA-II is a population-based algorithm, G-SEMO is configured as a trajectory-based algorithm which reduces the number of simulations runs that are called throughout the search process. Due to the execution time of a single simulation run and the fact that simulation-based optimization requires a larger number of simulations which therefore lengthens the optimization process, we also implemented a construction heuristic as described later in the section, which should yield good solutions in a shorter amount of time. Since during the initial experiments G-SEMO was outperformed by NSGA-II, further experiments were conducted by applying the latter algorithm. The subsequent results compare NSGA-II and the implemented construction heuristic.

2.2.1 NSGA-II

The multi-objective non-dominated sorting genetic algorithm II (NSGA-II) extends the classic GA by ranking solutions according to their objective values as described by [4]. In each generation, the Pareto front, which is the set of solutions that is not dominated by any other solution that was found by the algorithm, is calculated. A solution dominates another one if it is better in at least one objective value and better or equal in all others. The algorithm maintains a set of Pareto solutions, which, compared to other approaches, enables the user to choose from a variety of optimized solutions after the optimization run. Conventional GAs only return a single best found solution in each run.

[2] https://dev.heuristiclab.com

2.2.2 Construction heuristic

Another approach to optimize schedules for SST is to implement construction heuristics. Construction heuristics create a solution from the ground up step by step by applying a set of rules, usually in a way that yields promising initial solutions. Compared to evolutionary approaches, construction heuristics are advantageous in the sense that there is no need to conduct multiple simulation runs in order to evaluate and improve a set of solutions over multiple generations. One can use such constructed schedules directly or use them to seed other optimization algorithms which then try to further improve the solutions.

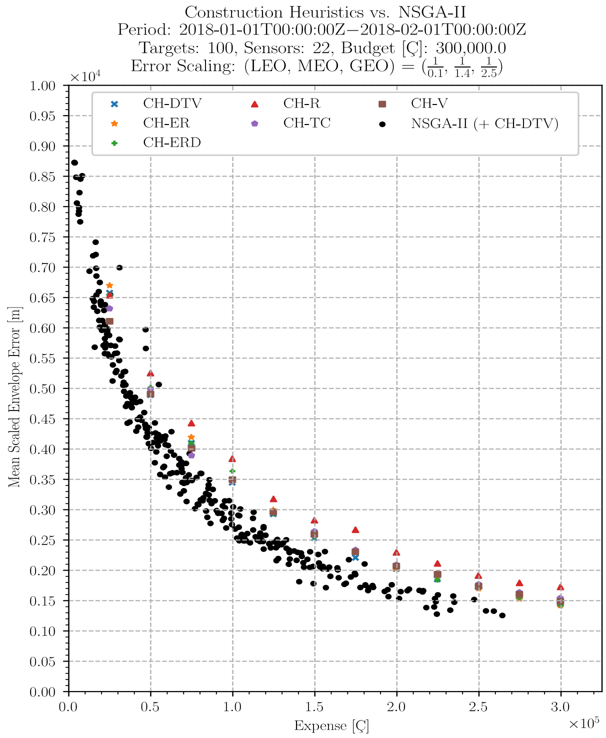

Figure 1. Comparison of NSGA-II and construction heuristic. Two objectives are minimized: envelope error and expense/credits (Ç). See section 2.2.3 for the interpretation of the results.

Table 1. Mean durations of solution creation (CH), evaluation (simulation) and overall algorithm execution (NSGA-II) calculated from 10 runs for 1-day and 1-month schedules.

Mean Duration	**1-Day Schedule [hh:mm:ss.sss]**	**1-Month Schedule [hh:mm:ss.sss]**
Solution Creation (CH)	00:00:03.143	00:00:07.235
Solution Evaluation (Simulation)	00:00:05.902	00:00:07.661
Overall (NSGA-II)	00:33:03.978	00:42:52.638

The following six target selection schemes were implemented as part of the algorithm testing:

1. Random (R): Targets are ordered randomly.
2. Total Count (TC): Sort targets according to their booked/observed count. Targets with fewer bookings/observations are preferred.
3. Violation (V): Shuffle and then sort targets according to the amount of time they have been violating the alert threshold in descending order.
4. Envelope Error (ER): Shuffle and then sort targets according to their total envelope error in descending order.
5. Envelope Error Delta (ERD): Shuffle and then sort targets according to their envelope error delta, i.e. the error that has been added to their minimum error, in descending order.
6. Distance To Violation (DTV): Shuffle and then sort targets according to the ratio between their envelope error and their violating error in descending order.

2.2.3 Results

Figure 1 shows the qualities of the schedules created by the construction heuristic and those found by NSGA-II (using the construction heuristic to seed the initial population). The two objectives, i.e. credits spent and mean envelope volume (per object per tick), are both minimized. Therefore, better solutions are located in the lower left corner of the objective space. The average execution times of the chosen NSGA-II configuration, as well as the average durations of CH and solution evaluation (i.e. the simulation) are shown in Table 1.

As expected, spending more credits leads to lower envelope volumes. Figure 1 shows that the construction heuristic already yields good results. NSGA-II, with its population seeded by the construction heuristic with target selection DTV, is able to find even better solutions. It is important not to restrict the initial set of solutions to a certain amount of expense, otherwise the algorithm will struggle to improve solutions located at this credit limit. As visible in the plot, solutions found by NSGA-II that require around 200 000 credits are comparable in terms of envelope error with solutions constructed by the construction heuristic with a budget of 275 000 credits. Vice versa, solutions created by construction heuristic with a budget of 200 000 credits are equivalent in terms of envelope error with NSGA-II solutions that require around 150 000 credits. Overall, one can observe that both approaches (CH and NSGA-II) yield results that are similar in terms of envelope error, however, those found by NSGA-II are better in terms of credits spent.

3 SYSTEM ARCHITECTURE

A high-level design overview of WebPlan is shown in Figure 2. The SAGE blocks represent 0-N scheduling agents which can be connected to the Central Hub and 0-N sensors. NON-SAGE Client block represents an external client which consumes interfaces exposed by the Central Hub such as credits status, adding, removal or retrieving observational data. The “Catalogue” block shows the auxiliary functionality of the Central Hub to provide up-to-date or historical ephemeris data for objects orbiting the Earth (in the current implementation based on the SpaceTrack catalogue). The “Data storage” block depicts the general persistence capability of the Central Hub to store the commands going to, from or through the system, user data, sensor data, as well as observation data.

3.1 System components

WebPlan is deployed with GitLab CI/CD with Docker images, divided into the following major software components:

Component	**Description**
Central Hub (CH)	Set of services deployed in one server room, serving as a persistence, caching layer, broker between Scheduling Agents users and Sensors, assuring fair and safe communication between them, time allocation functionality; bookkeeping of the resources and requests; monitoring, filtering and archiving data going through (time slots, observing programs, credits).
Scheduling Agent – CAT (SAGE-CAT)	Scheduling agent tasked with catalogue maintenance, with the goal of keeping the position error below a specific threshold; decay of the precision provided on the basis of historical data with NSGA-II algorithm implementation.
Sensor (SENSOR)	Receives schedules, provides information on itself (sensor), reports scheduling progress, uploads data

	(TDM and/or raw, additional data) to CH.
Catalogue (CAT)	Service enabling querying and/or requesting ephemeris dataform. Part of the CH.
AstroDrive (AD)	Stores observational data. Based on the existing solution from Sybilla Technologies – AstroDrive. Allows for data queries, data download and upload. Part of the CH.

3.2 Auxiliary components

All auxiliary services are part of the CH deployment:

SST Utils Services	Provide the following functionalities: • Conversion between formats TLE/OEM/CPF
Authorization and authentication service (AAS)	Authorization (compatible with OAuth 2.0) and authentication (OpenID) for users and agents. Basic functionality for security of data, credits, sharing and limiting access, visibility of features, data. Based on Sybilla Identity Server.
Central Logging Service	Entity that keeps and allows for browsing, search of the logs from critical system components.
Central Configuration Repository	Entity responsible for versioning and central point for configuration files used by the system, not obligatory but optional for services willing to track their configuration.
Mock sensors	Entities (radar, laser, optical) providing a way for quick testing without the need for real sensors usage
GitLab	Configuration files storage system, with create, delete, change and versioning capability and Web UI for viewing accessing files.

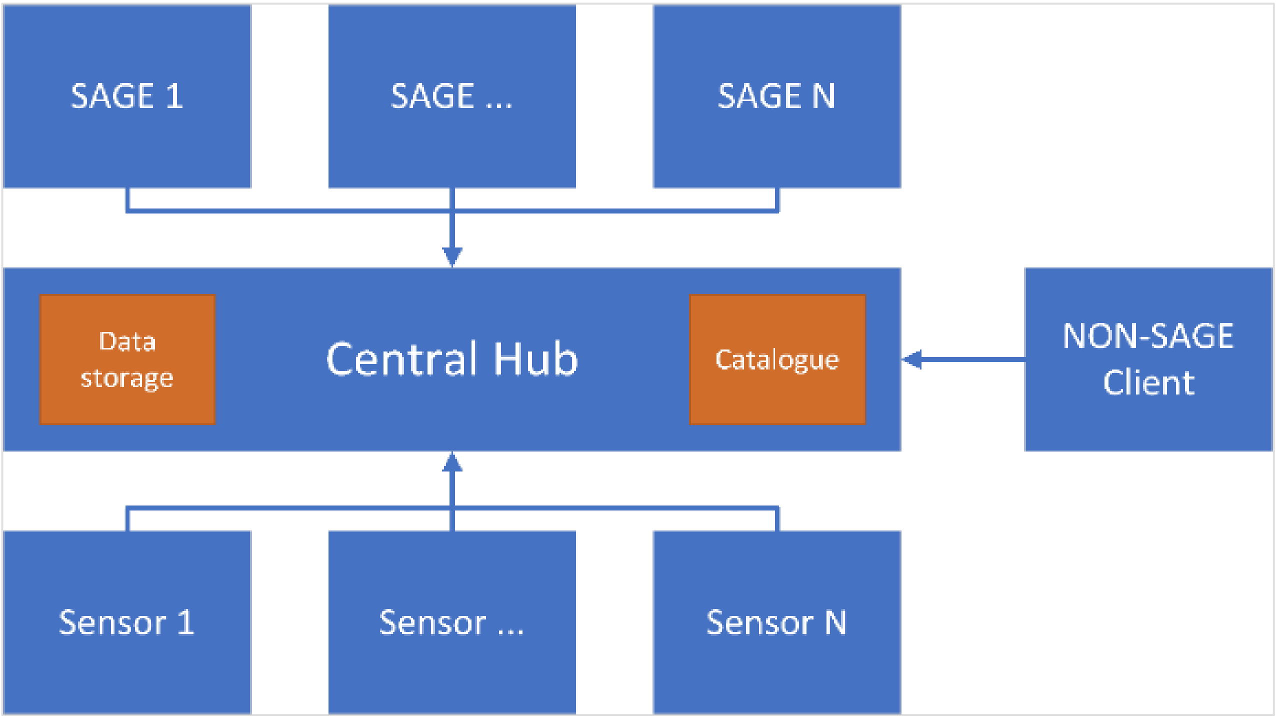

Figure 2. WebPlan: a high-level design overview with the main system components depicted. SAGE stands for Scheduling Agent. For a detailed description of the blocks please see Section 3.

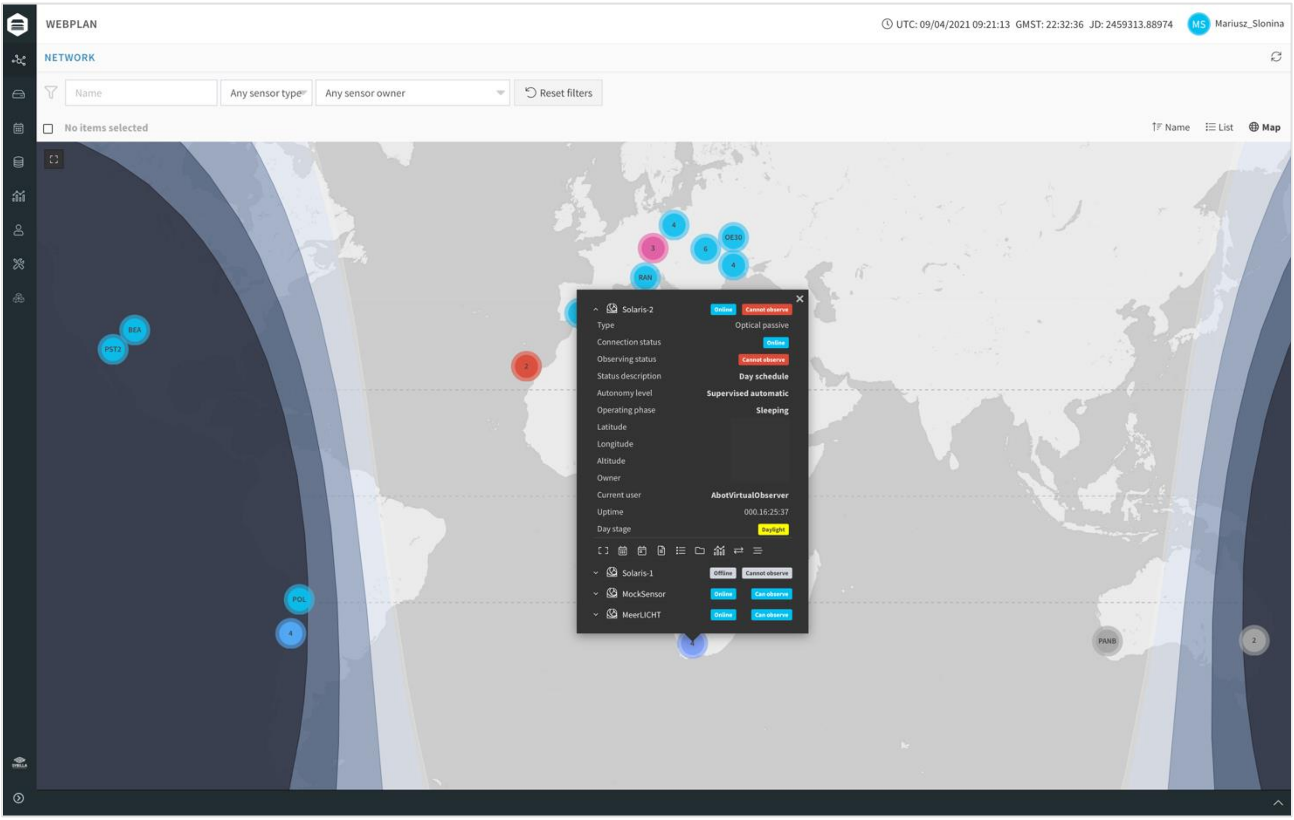

Figure 3. Central Hub: example map view with twilight overlays; sensor properties and telemetry can be displayed. See section 4.2.1 for more details.

4 WEB USER INTERFACE

WebPlan web user interface is built on top of Sybilla's AstroDrive UI and integrates all of the system components into one web application (Chrome, Firefox, Safari, Edge are the supported browsers).

4.1 Overview

The AstroDrive UI is a single page web application. It is decomposed into several modules:

- **Network** – a top-level Central Hub view of all connected sensors that are available to the user, including the sensor's dashboard,
- **AstroDrive** – a dedicated storage service for managing, displaying and manipulating astronomical data,
- **Scheduling** – combines sensor's timebooking, observation planning and scheduling agents (Timeslots, Programs, SAGE),
- **Catalog** – space object catalogs: spacecrafts, Solar System Objects, Near Earth Asteroids and Stellar objects (SpaceTrack/Celestrak, Simbad, SSO/NEO),
- **Banking** – bank accounts management.

The following sections provide a more detailed description of the modules.

4.2 Network (Central Hub)

4.2.1 Network state view

The Network page (*Figure 3*) provides an overview of current sensor network state. The network shows sensors that are available to the current user; these are available as a (world) map or a list view. The system supports passive optical, active optical, passive RF and radar sensors.

The following information for each sensors is provided to the user:

- *Name,*
- *Location (WGS84),*
- *Type,*
- *Owner,*
- *Connection status (to CH),*
- *Observing status,*
- *Uptime.*

The map view provides a world-map view of the sensor network, overlayed with twilight (civil, nautical, astronomical) data, updated in nearly real-time.

If real-time, advanced telemetry is provided by the sensor (i.e. providing state of particular sensor's components), additional properties might be shown:

- *Autonomy level* – i.e. Manual, Automatic,
- *Operating phase* – e.g. Waiting, Observing, Taking Bias,
- *Current observer* – the current user performing the sensor's operations.

It is possible to export data on selected sensors as a JSON, GeoJSON or CSV file.

4.2.2 Sensor dashboard

Sensor dashboard provides an overview of a given sensor's activity in a single place (see *Figure 4*, *Figure 5*). Additionally, the user can register or unregister the sensor from the network and pass can/cannot observe state over the network.

The current sensor state is split into the following:

- *Overview* – provides sensor's telemetry,
- *Schedule* – sensor's time allocation,
- *Timeslots* – timeslots allocated on the particular sensor,
- *Programs* – the list of observing programs scheduled on the particular sensor,
- *Queue* – the observing queue of the sensor for the next 24 hours,
- *Operations* – any operations performed by and over the particular sensor,
- *Logs* – logs provided by the sensor.

The *Overview* page provides an overview of the particular sensor state. The *Schedule* tab provides a calendar view of timeslots over a specified period of time on a particular sensor. Day, Week (default) or Month calendar views are available.

Timeslots are color-coded according to their status (inbound, accepted, rejected, pending, maintenance). To view the allocated timeslot's details, the user clicks on the specific timeslot and a dialogue window will appear.

To reserve a timeslot on the specified sensor, the user may use either the *Reserve a timeslot* action or click on a calendar single or multiple cells (datetime range) in the *Schedule* page.

4.3 Drive

The Drive module provides access to user's data volumes available within the AstroDrive. Two data volumes are provided:

- *Files* – user's files and folders stored within the AstroDrive,
- *Shared* – files and folders shared by users within the AstroDrive.

The *Files* volume page shows the folders and files of the current user, either as tiles or as a list (see *Figure 6*).

The following actions are possible for the Drive:

- drag'n'drop file upload,
- folder management: create, rename, move, share, delete, archive,
- file management: download, share, delete, archive
- basic file and folder filtering,
- file and folder sorting (stored in browser's local storage),
- shared view.

Built-in FITS support is offered (importing, managing and displaying) through the SlimFits[3] and ThreeJS[4] libraries (see *Figure 7*):

- image statistics (configurable annulus, frame, projections, streaks),
- zoom, pan, rotate,
- WCS coordinates,
- FITS header browser,
- basic image manipulation: histogram, scaling, color maps (settings stored in the database).

The system supports the following Space Safety data formats: TDM (XML, KVN), CRD/FRD/NPT, JSON, XML, plain text.

[3] https://github.com/sybilla/slimfits

[4] https://threejs.org/

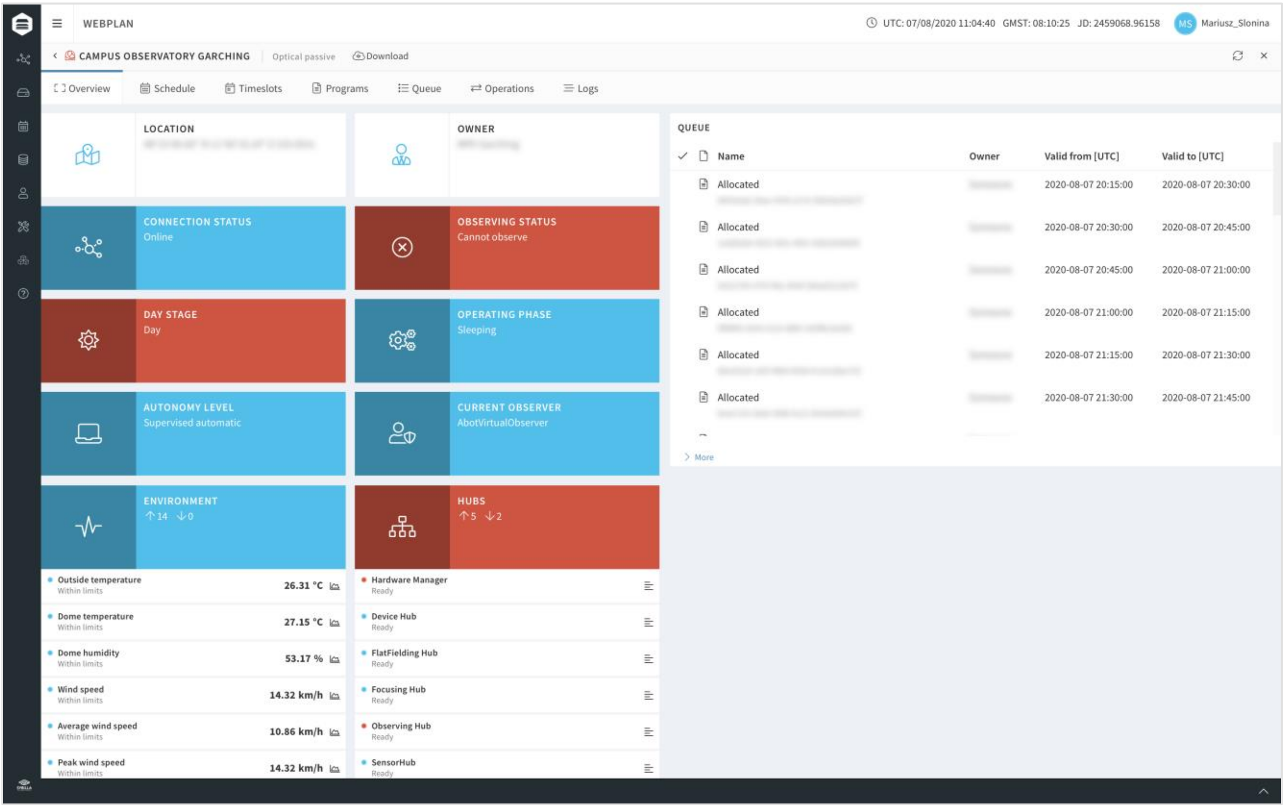

Figure 4. Single entity view: Sensor UI with full component overview (displayed if provided by the sensor configuration and telemetry); see section 4.2 for more details.

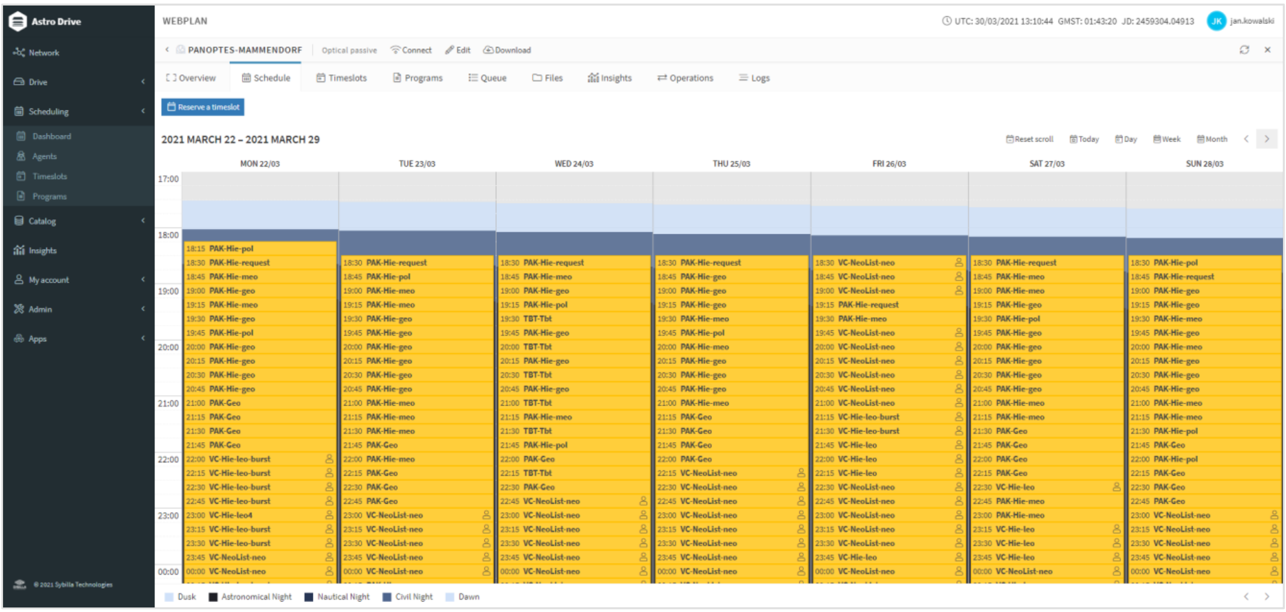

Figure 5. Sensor UI: calendar view; see section 4.2 for more details.

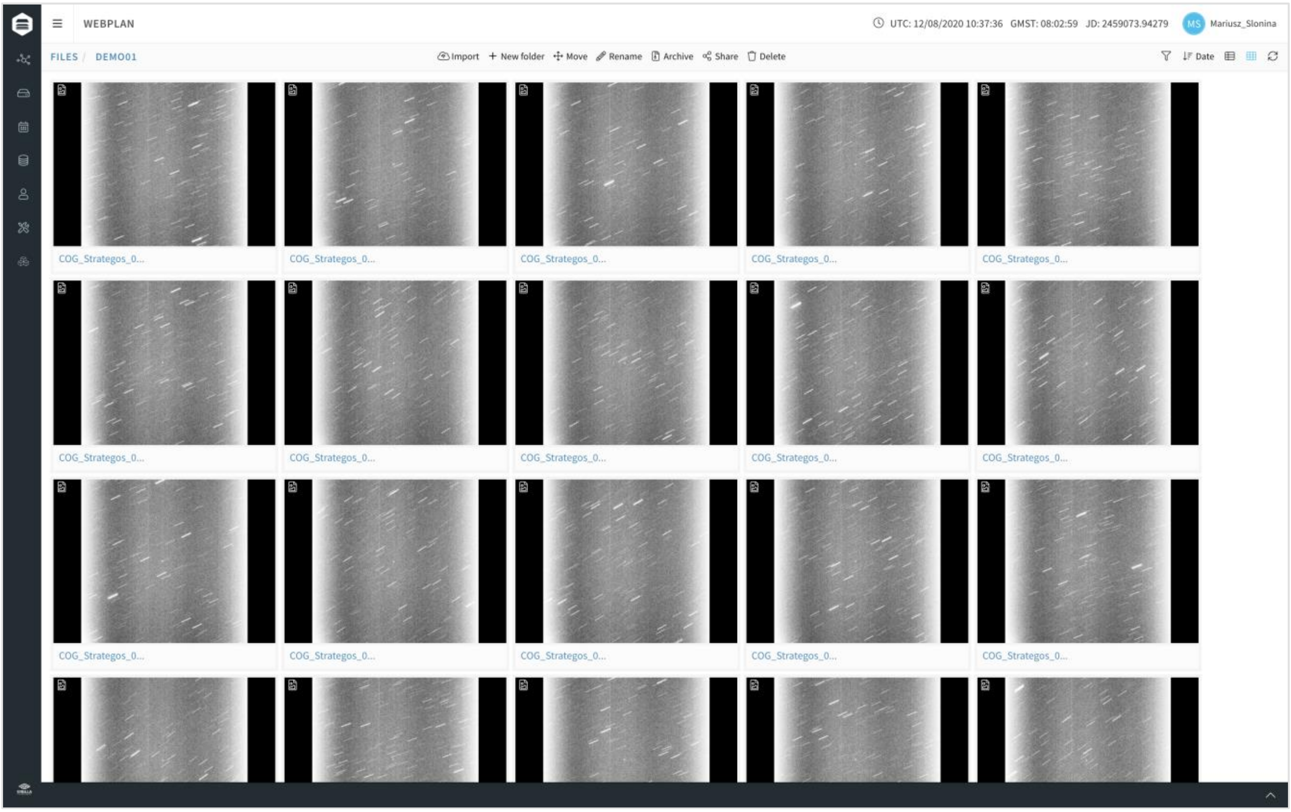

Figure 6. Drive: tile view. See section 4.3 for more details.

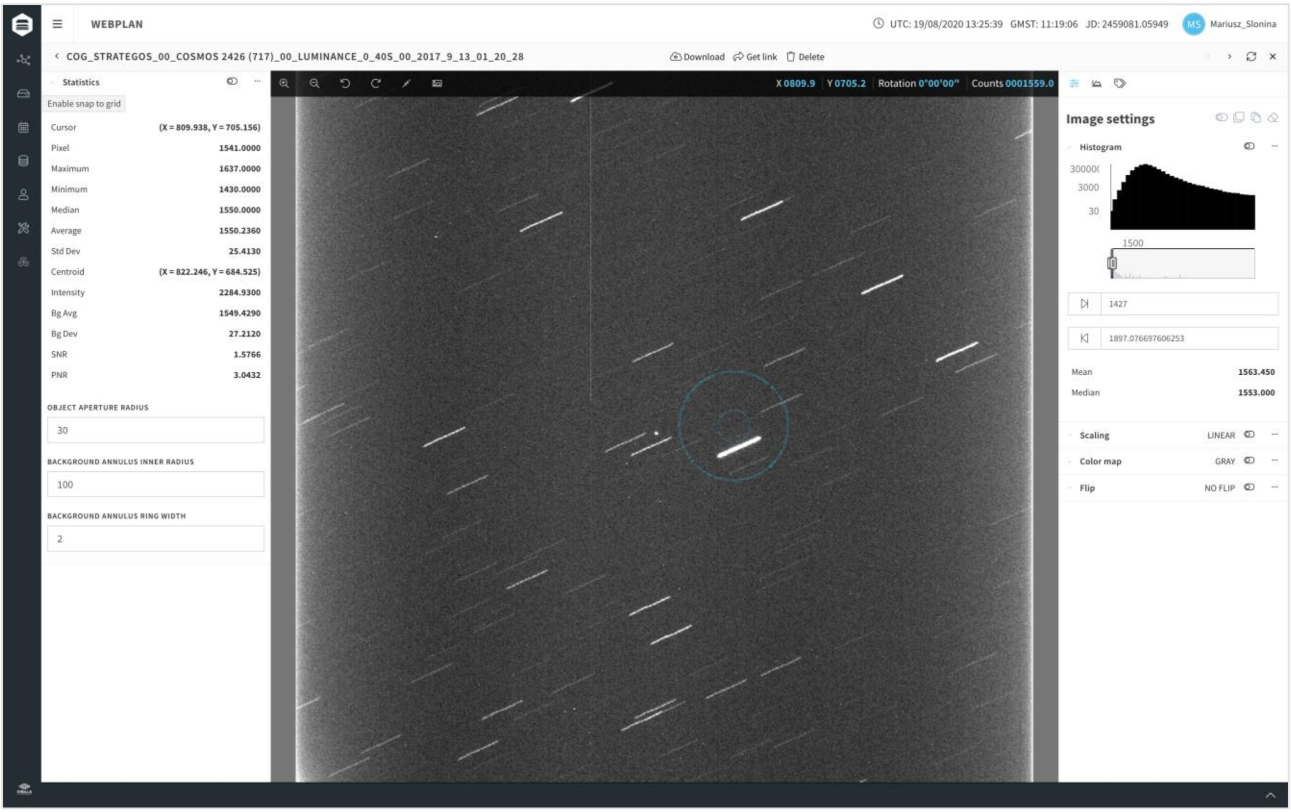

Figure 7. FITS viewer. See section 4.3 for more details.

4.4 Scheduling

The Scheduling module adds the capability of scheduling timeslots, observing programs, as well as managing scheduling agents.

The scheduling agents page (see *Figure 9*) gives users the view of agents associated with the current user along with the agents' properties such as type, status or budget. The agent's dashboard then provides more detailed information such as the available and used sensors, monitored objects, collected raw and TDM files, the agent's observing campaigns, as well as available budget.

There are several types of scheduling agents that can be created, including generic, SST, NEO, stellar and ad-hoc agents. For each agent, the user can configure targets, sensors and budget. The scheduling process can be automated then: from computing object passes to allocating time and preparing observation programs.

A manual scheduling process is also available via the sensor's Scheduling tab. Once a timeslot is reserved, the user can create a program using built-in visual editor (though XML/ASCII forms are also provided). Multiple targets of different types can be specified, with configurable parameters, such as exposure time or filters, depending on the sensor type. The object visibility chart is available to help choose the best opportunity window (*Figure 8*). Once ready, the program is automatically converted into the format that is recognized by the sensor and is scheduled.

4.5 Catalogue

The Catalogue module offers integration with astronomical and spacecraft databases (satellites and debris, Solar System objects, Near-Earth Objects, stellar objects). Each of the integrated databases offers filtering objects by name and type, where applicable also by owner, launch date, etc. The catalogues can be accessed and searched during observation planning, either manually or via scheduling agents. Both latest or historical object's ephemeris can be retrieved.

4.6 Banking

Through the Banking module the user can manage multiple sub-accounts, assign accounts to timeslots and scheduling agents, transfer credits between accounts as well as top up their primary account.

5 APPLICATIONS

5.1 Test campaigns

The first real-life test campaigns took place in Dec 2019 and March 2020 as part of the Factory Acceptance Testing for ESA. Six optical sensors (locations: Spain, Germany, Australia) together with the CBK PAN SLR station and a mock radar setup. A total of 40 objects from the LEO/EMO/GEO regimes were followed during the campaign. AstroDrive and Astrometry24.NET (see [5]) were used during the observation campaigns as optical data storage and processing tools. For the SLR data the post-processing software of CBK PAN was used and the results of the observations (TDM KVN files) were sent automatically to the WebPlan server using the AdSync tool and a dedicated sensor account on the server.

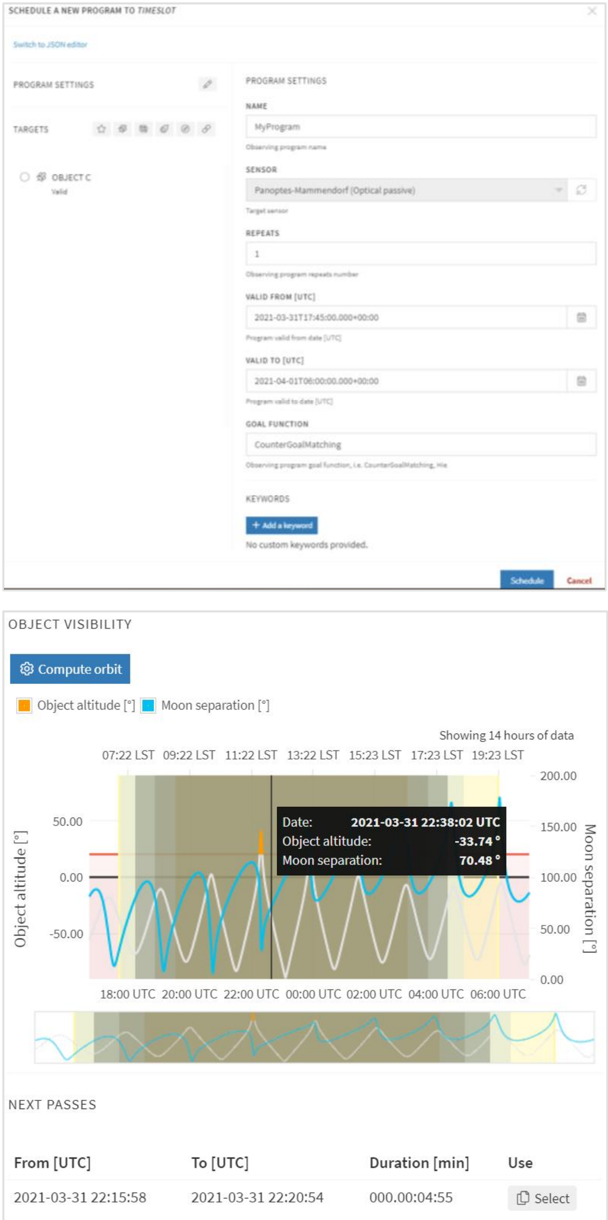

Figure 8. Manual observation planning: target selection in a program scheduling form (top) and a computed visibility chart (bottom). See section 4.4 for more details.

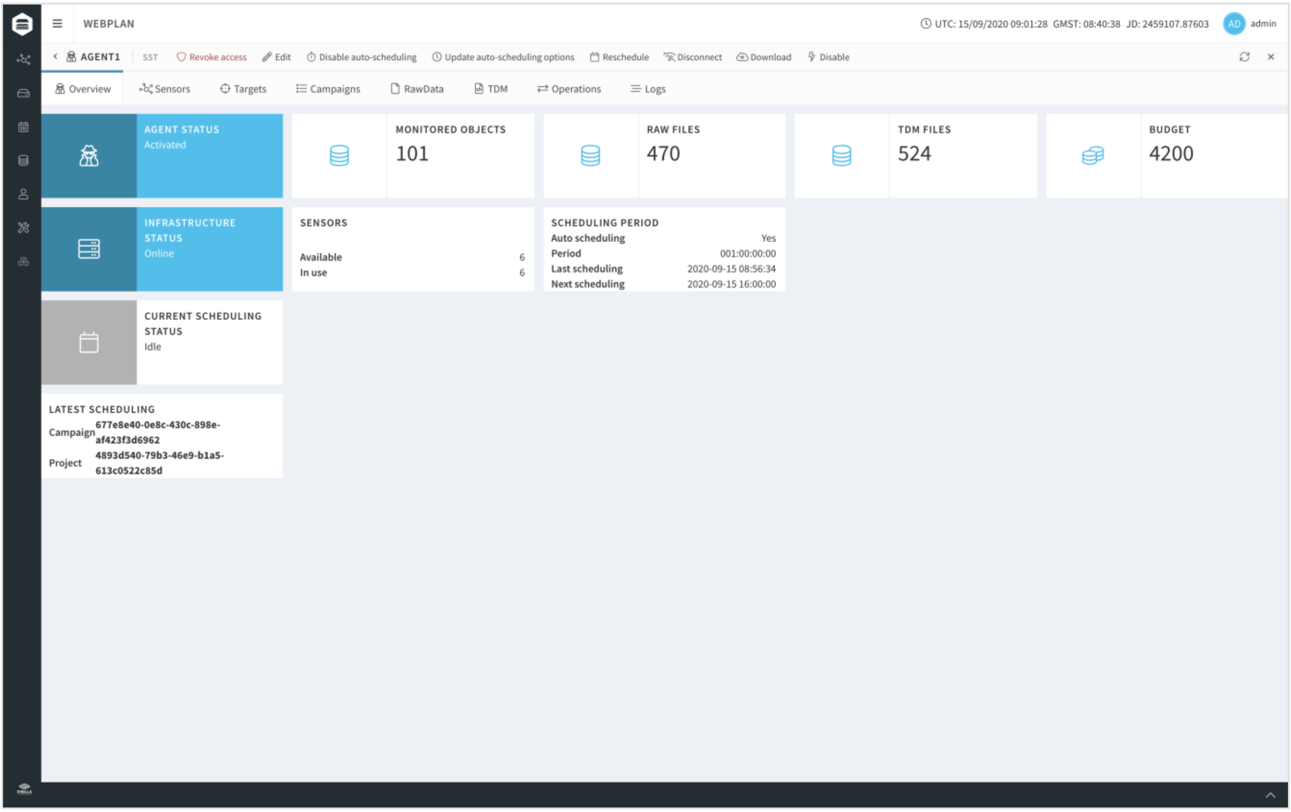

Figure 9. Scheduling Agent dashboard. See section 4.4 for more details.

5.2 Current status

WebPlan is currently used to operate the scheduling for the Panoptes-Solaris network of 14 telescopes (see e.g. [6], [7]). The scheduling module of WebPlan was also implemented at the Polish Space Agency (POLSA) at the end of 2020. As part of the cooperation with POLSA, WebPlan was used to prepare operational plans for the Polish contribution under EU SST[5]. The system was also used within the framework of the ESA-funded POSST project [8] to execute observing programmes.

Since its commissioning WebPlan has had 43 connected sensors at peak usage, with on average 16 sensors fully booked through WebPlan at any given time as of March 2021. Table 1 provides statistics on the time booked through WebPlan and the system's productivity.

It should be noted that the quoted FITS and TDM counts are not complete as they only include the products of observations that have gone through the WebPlan system. As some sensors deliver observations directly to their customer, the actual file/product numbers will be higher.

Table 2. WebPlan productivity in numbers as of March 29, 2021.

	Since commissioning	**Last six months**
Time slots booked	178 K	91 K
Time slots accepted	166 K	89 K
FITS gathered	2 479 K	1 220 K
TDMs produced	144 K	95 K

6 CONCLUSIONS

Efficient observation planning is required in the demanding SST environment. The WebPlan project's goal was to equip sensor owners, astronomers and SST operators with sophisticated scheduling tools that take into account sensor availability, visibility of objects and environmental statistics, and look for the best opportunity windows.

WebPlan is a web-based software suite for an easy schedule creation for a single sensor or a network of sensors, with the main focus being on the user's observing goals. The system provides simple and clear user interfaces, automation, streamlining of the

[5] https://www.eusst.eu/

scheduling process, monitoring its progress and retrieving results. It incorporates NSGA-II, an advanced scheduling heuristic algorithm which performs around 30% better than other compared algorithms when tasked with catalogue maintenance. It includes multiple custom scheduling agents executing specialized scheduling tasks for GNSS calibrations, GEO tracking, high interest events, re-entry or collision avoidance data provisioning.

WebPlan sends created observing programs to sensors and waits for the data to be received. Once observations have been performed, the sensors' processing pipelines can upload data to the AstroDrive. These can be either FITS or TDM files, or both. It is possible to visualize the results in the browser, and monitor the progress of the campaign.

The suite is a large (over 50 services on production environment) and complex (5 entities involved in the longest chain of tasking) distributed service based on Docker virtualization. WebPlan is currently used to operate a 14-telescope Solaris-Panoptes optical network and its components have also been adapted by the Polish Space Agency to operate the Agency's observatories and plan contributions to the EU SST consortium. The system has a capacity for up to 100-1000 sensors, depending on the system architecture and hardware used.

7 REFERENCES

[1] M. Mora i M. Solar, „A Survey on the Dynamic Scheduling Problem in Astronomical Observations," w *IFIP International Conference on Artificial Intelligence in Theory and Practice* , 2010.

[2] J. Colome, P. Colomer, J. Guàrdia, I. Ribas, J. Campreciós, T. Coiffard, L. Gesa, F. Martínez i F. Rodler, „Research on schedulers for astronomical observatories," w *Observatory Operations: Strategies, Processes, and Systems IV. Proceedings of the SPIE, Volume 8448*, 2012.

[3] S. Wagner, G. Kronberger, A. Beham, M. Kommenda, A. Scheibenpflug, E. Pitzer, S. Vonolfen, M. Kofler, S. Winkler, V. Dorfer i M. Affenzeller, „Architecture and Design of the HeuristicLab Optimization Environment," w *Advanced Methods and Applications in Computational Intelligence. Topics in Intelligent Engineering and Informatics*, Heidelberg, Springer, 2014, pp. 197-261.

[4] K. Deb, A. Pratap, S. Agarwal i T. Meyarivan, „A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, tom 6, nr 2, pp. 182-197, 2002.

[5] A. Sybilska, S. Kozłowski, P. Sybilski, R. Pawłaszek, M. Słonina, A. Gurgul, P. Konorski, M. Drzał, S. Hus, G. Lech, M. Litwicki, M. Pilichowski, R. Ślimak i U. Kolb, „Astrometry24.net – precise astrometry for SST and NEO," w *ESA NEO and debris detection conference - exploiting synergies*, Darmstadt, Germany, 2019.

[6] S. K. Kozłowski, P. W. Sybilski, M. Konacki, R. K. Pawłaszek, M. Ratajczak, K. G. Hełminiak i M. Litwicki, „Project Solaris, a Global Network of Autonomous Observatories: Design, Commissioning, and First Science Results," *Publications of the Astronomical Society of the Pacific*, tom 129, nr 980, October 2017.

[7] M. Konacki, P. Lejba, P. Sybilski, R. Pawłaszek, S. Kozłowski, T. Suchodolski, M. Słonina, M. Litwicki, A. Sybilska, B. Rogowska, U. Kolb, V. Burwitz, J. Baader, P. Groot, S. Bloemen, M. Ratajczak, K. Hełminiak, R. Borek i P. Chodosiewicz, „The Solaris-Panoptes Global Network of Robotic Telescopes and the Borowiec Satellite Laser Ranging System for SST: A Progress Report," w *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, Wailea, Maui, Hawaii, 2017.

[8] N. Siódmiak, J. Doubek, P. Sybilski, W. Bykowski, E. Kerr, N. Maric, A. Mancas i E. Cordelli, „Polish SST small telescope assessment and prototyped operations," w *8th European Conference on Space Debris*, Darmstadt, Germany, 2021.

[9] T. Sasaki, G. Kosugi, R. Hawkins, J. A. Kawai i T. Kusumoto, „Observation scheduling tools for Subaru Telescope," w *Optimizing Scientific Return for Astronomy through Information Technologies*, 2004.

8 ACKNOWLEDGEMENTS

The authors would like to thank Wieńczysław Bykowski, Michał Drzał, Agnieszka Gurgul, Paweł Nerka, Maciej Pilichowski. Rafał Ślimak, Alicja Wójcik (Sybilla Technologies), Piotr Michałek (CBK PAN), Noelia Sánchez Ortiz, Roxana Larisa Andrișan (Deimos Space) for their contributions at various stages of the project.

The project was funded under an ESA contract No. 4000125521/18/D/CT.