AGILE SMALLSAT OPERATION TOOL-CHAIN DEVELOPMENT: HYPSO-1 HYPERSPECTRAL EARTH OBSERVATION EXPERIENCES

Dennis D. Langer[1], *Simen Berg*[2], *Joseph L. Garrett*[2], *Roger Birkeland*[3],
Sivert Bakken[2], *Tor A. Johansen*[2], *and Asgeir J. Sørensen*[1]

Centre for Autonomous Marine Operations and Systems
[1]Department of Marine Technology
[2]Department of Engineering Cybernetics
[3]Department of Electronic Systems
Norwegian University of Science and Technology (NTNU)
O.S. Bragstads Plass, 7034 Trondheim, Norway

ABSTRACT

Satellite operations is important for a successful mission, but can be time consuming for inexperienced operators in new small satellite projects. We present a method for the continuous development of an earth observation satellite operations system based on agile principles, specific mission objectives and lessons learned. Results are presented using data from the HYPSO-1 satellite. Downloaded data amount and quality with regards to image cloud content is maximized using a single ground station while drastically reducing operator time investment.

***Index Terms*—** cubesat, operations, hyperspectral remote sensing, HYPSO-1, software development

1. INTRODUCTION

In concert with improvement, higher availability and cost reduction of satellite system technology, e.g. cubesat components the number of satellites in orbit has increased [1]. Consequently, more services and data products are available from space-borne platforms. However, if a satellite is not able to fulfil mission objectives, large amounts of effort, time and money would be lost. The satellite will instead contribute to the increasing amount of space debris, which over a long period can threaten other functioning satellites in orbit. It is therefore important that satellites are not destroyed during launch or by the environment in orbit, and that they are operated to maximize benefit.

Many commercial entities have entered the small satellite market [2]. This includes for example companies specialized in designing, manufacturing and operating satellites. The expansion of the satellite market improves the expertise for specialized parts of space technology, and there is therefore higher likelihood for successful satellite missions. An increased market for space technology lowers the bar to start smaller and less costly projects that can use Commertial Off-The-Shelf (COTS) components. Instead of launching several satellites to gain the experience and knowledge needed to build a successful satellite system, it is reasonable to assume instant success when relying on state-of-the-art products in the market. In addition, making a space system with the ability to receive software updates, there is a possibility of iterative improvements as seen fit, which is more adaptable than a static configuration that is locked at launch.

Satellites used to require hundreds of involved personnel to operate and maintain [3]. Today's trend towards COTS small satellites is decreasing the cost, time and manpower requirements towards building, launching and operating a satellite. When launching a satellite system largely based on COTS componentsit is now possible to let the project team operate the satellite themselves.

While there are many papers on the design of satellite subsystems, the literature is sparse on how to operate a satellite. Research into ground based satellite operations has been down-prioritized in favor of on-board autonomy [4].

The first HYPerspectral Smallsat for Ocean observation satellite (HYPSO-1) is an example of a low-budget small satellite based on COTS components. Specifically, it consists of a 6U satellite bus from Kongsberg NanoAvionics with an in-house developed hyperspectral push-broom camera as payload. HYPSO-1 [5, 6] is a science-driven mission collecting ocean color data with its hyperspectral imager, as well as a test platform for novel operational concepts, operational procedures and planning methods. This includes the use of onboard data processing with low-latency delivery of data for real-use, e.g. guiding an autonomous surface vehicle [7], thus HYPSO-1 can be a part of a system-of-systems that involve

This work was supported by the Research Council of Norway through the Centre of Excellence funding scheme NTNU AMOS (grant no. 223254), MASSIVE (grant no. 270959), HYPSCI (grant no. 325961), Green-Platform (grant no. 328724), EEA NO Grants 2014 - 2021 under Project ELO-Hyp (contract no. 24/2020).

in-situ marine robots and unmanned aerial systems [8].

HYPSO-1 is operated by a team with little experience of satellite operations. To implement improvements from continuous lessons learned, the working methodology of the operators and ground software developers is based on agile principles.

There are many criteria for characterizing an agile software development team. Some of them are about whether the team is self-organizing, continuously striving for improvement and work closely with their stakeholders [9]. The Agile Alliance defines agile software development to be "[...] an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it", [10]. Some argue that agile development is required in a university context to support concurrent satellite development under presence of time and cost constraints [11]. Others argue for that agile means to continuously improve one's methods and tailor them to the team's specific situation, as opposed to sticking to a single framework [10, 12, 13]. In this sense, agile is a meta-method, sitting on top of and influencing the specific methods chosen.

Planet Labs PBC [14] is an example of a successful commercial new-space entity employing agile methods. They report that taking an agile approach helped with iterating and improving the design, collaboration with the end users and bringing in prospective customers early. Downsides are that fast iteration requires inexpensive components and launch costs, which is enabled by technological progress and miniaturization of electronics.

In this paper, we propose a method for the continuous development of a earth observation satellite operations system based on agile principles and specific mission objectives. The low-cost COTS aspects of HYPSO-1 are taken into account during development of the ground system and operation.

First, the proposed method is expanded in Section 2. Then Section 3 shows the results of continuous development to software and operational procedures. Finally, Section 4 concludes the research.

2. METHOD

Design principles and operational goals depend heavily on the specific mission. Hence, we some details in the principles and goals discussed here are specific to HYPSO-1.

2.1. Design Principles and Operational Goals

The design principles that are followed overlap with the principles attributed to agile software development [12]. Some of them are listed in the following: 1) The software should address specific user needs. Satellite operations have a time pressing aspect, as satellites have limited lifetimes in orbit that cannot be set on hold. There is often no time to focus on general features before addressing specific needs. 2) The software should be *modular*. The software tool-chain in a satellite operations software system should consist of components with few inter-dependencies and clearly defined inputs and outputs. Related to the first point: 3) the software should be *simple* and have *intuitive user interfaces*. It should not contain unnecessary components and it should be possible to figure out how to use a tool with minimal dependence on external documentation.

The main operational requirement for the HYPSO-1 satellite is to maximize the delivery of useful hyperspectral remote sensing data from the prioritized targets. The data is in the form of hyperspectral data cubes with corresponding metadata, see also [15]. The prioritized targets may change on a day-to-day basis. In addition, HYPSO-1 missions typically have a number of additional requirements for frequently revisit targets, for low latency data delivery, aiming to minimize cloud content in images, and to support experimental image acquisitions and extension of on-board processing via software updates.

The operational goal is to achieve these requirements under a set of constraints. The driving constraints for HYPSO-1 include factors like the power and energy budgets, ground station availability, downlink throughput and available on-board storage space, while also working with imposed software tools and infrastructure that are provided by the COTS satellite bus and components provider.

2.2. Development of Satellite Operations

As an inexperienced team begins to operate their recently launched small satellite, it is likely that the operational procedures that are used initially include more manual work than necessary. This manual work can be made more efficient and less time intensive via automation.

Modern open source software packages and tools are powerful while still being easy to use. This enables the development of custom software utilities to rapidly improve and automate satellite operations, according to the needs of the operators. Another aspect of the availability of these software tools, is that operators themselves can develop their own software to automate their tasks. In a sense, the operators may become their own stakeholder, defining their own requirements for software development. In another sense, the requirements are defined by operational goals and design principles.

2.3. Methodology

Figure 1 illustrates the proposed methodology, showing the flow from identification of a need for improvement until use of new software to address this need. The methodology can be divided into 5 steps:

1. The operators identify a need for improvement to their workflow.

2. New software is developed to satisfy the need.

3. The software is tested in simulation or on a hardware-in-the-loop setup.

4. The software is tested on the satellite.

5. The software is integrated into day-to-day operations.

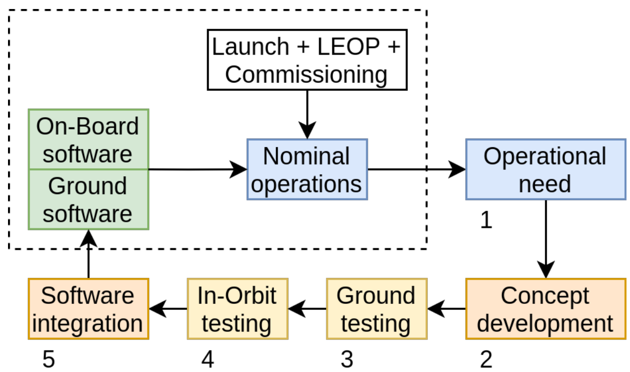

Fig. 1. Block diagram of the proposed workflow. This workflow was adopted during the operation of HYPSO-1.

The term *software* here can refer to both on-board software of the satellite and software used on ground to aid in satellite operations. As the ground software and operations team gains experience operating the small satellite, ideas for improvements become clear (1). These ideas inform the ongoing development of both the ground segment software and the on-board software (2). Testing is essential to make sure the software is working as intended without side effects of causing issues (3) & (4), see [16]. After successful initial deployment on the satellite during operations, the updates to software is integrated into the nominal operational procedures (5).

The methodology speeds up the software development cycle, with real-time feedback, more software iterations and faster deployment of new features that address specific user needs. There is little delay between development, testing and deployment if it is the operators themselves who are developing the software, and new functionality may be integrated in a matter of days to weeks. Software relating to satellite operations can be broadly categorized into three groups. I.e. software is used:

1. by the operators to plan actions with the satellite,

2. on the satellite, or

3. to processes data from the satellite.

1) is used to schedule automated communication with the satellite, and create and upload scripts to the satellite (ground software for planning). 2) is used to add new features to the on-board command and data processing of the satellite. 3) is used for ground processing and analysis. For 1) and 3), step 4 in Figure 1 may not be relevant or possible, in which case this step is skipped.

For HYPSO-1, there is a system architecture that provides mechanisms for fault tolerance and resilience, such that software can be tested in-orbit without risk for permanent damage to the satellite [17].

3. RESULTS

The methods described in Section 2.3 have been applied while operating the HYPSO-1. The operational goals (see Section 2.1) have remained the same for HYPSO-1 since launch. However, the tools used to achieve the goals have changed drastically.

In this section, we present how agile software development and overlapping developer and operator roles have improved the operational workflows with respect to these three parts, and increased the data quality of HYPSO-1 since launch. Possible improvements to operational procedures were identified by qualitatively assessing the proportion of time required to perform a certain task out of the total time spend by a team member on operations related tasks.

3.1. Planning and Scheduling Software

The tasks performed during the planning stage for imaging are 1) selection of imaging targets, 2) checking suitability of targets with respect to overpass opportunities, pointing angles within acceptable range, and predicted cloud cover, and 3) generation of an imaging schedule in the form of a command script to be uploaded to the satellite.

At launch, the ground software for planning was consisting of the Ansys Systems Tool Kit (STK) and a MATLAB script for computing the attitude parameters for pointing HYPSO-1 to targets at specific times. Targets were checked manually for passes using STK and cloud cover. STK was tedious to use, as simple actions like determining future passes over targets took minutes per target, and overpasses were shown in comparatively hard to read text files. The number of targets to be imaged were few, as checking each target took time. Schedules were written and uploaded manually. This was the main reason for limited capture capacity in the initial months of operation, see Figure 2.

Python based tools were made to automate these steps. A map-based Graphical User Inteface (GUI) was made to check and visualize overpasses to aid target selection, requiring less than a minute to check a target. A minimal set of parameters required to write a imaging schedule script was identified. Thus, it was possible to generate the imaging schedule scripts automatically. Upload of the schedule was automated using a satellite overpass tool delivered by the satellite bus manufacturer.

By the beginning of 2023, also overpass and cloud cover checking was automated using self-developed tools. These tools relies on open-source software and Application Programming Interfaces (APIs), such as cloud cover provided by met.no. The downlink capacity of HYPSO-1 was saturated, resulting in consistent performance of more than 100 images per month, see Figure 2. The imaging capacity of HYPSO-1 was found to be 5-6 images per day. Due to automation, the list of potential targets has reached ca. 150. The 5-6 prioritized image captures per day are automatically selected depending on pass availability and cloud cover forecasts.

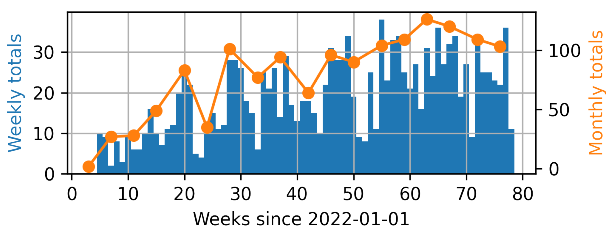

Fig. 2. Lifetime histogram of images from HYPSO-1 per week and per month.

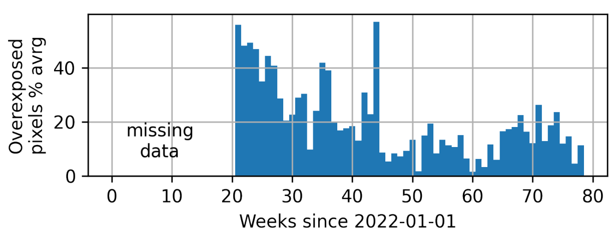

Fig. 3. Average amount of overexposed pixels in an image per week.

3.2. Ground Processing Software

Data handling on ground was found to be equally time consuming as planning due to the manual work required. Initially, data was downlinked manually, requiring time intensive operator presence during a pass. The data was stored in a raw format in a location for internal data sharing. Automatic data downlinking using software from the satellite bus provider was one of the earliest improvements for operations. The downlinked data is stored in a satellite bus provided database, from which the data need to be extracted to perform processing. Further improvements are association of the hyperspectral data with Attitude Determination and Control System (ADCS) telemetry for direct georeferencing, standardized data formats and a metadata database. Moreover, the team is currently developing an automated calibration and correction pipeline to disseminate data at levels L1 and L2.

3.3. On-board software

Initially, on-board processing included binning and compression. Recently, on-board processing has been expanded with semantic segmentation. It is also possible to generate over and underexposure masks, generate RGB thumbnails on-board, and perform direct georeferencing. However, this is not included in nominal operations yet.

3.4. Data Quality

Due to the map-based satellite overpass tool, more time was available for more detailed cloud cover checking, resulting in an improvement of data quality in terms of overexposed pixel content, see Figure 3.

3.5. Discussion

In one way, the HYPSO team was forced to work in an agile, self-organizing way. Every member had duties in addition to tasks relating to operating and developing software for HYPSO-1. For example, students typically also conducted research that are related to the use of HYPSO-1 data and develop its data processing algorithms. Thus it is natural to optimize for time and data quality. In another way, the proposed method may be tended towards naturally for new small satellite operators, if operations are not planned in depth, and the operators will need to learn a lot in a short period of time, while the remaining lifetime of the small satellite decreases day by day. However, it may be just as likely that operations are neglected considering the time investment. In addition, initially after launch, a degree of manual work is accepted due to the novelty of operating one's own satellite.

Fast software development cycles cause workflows to quickly become outdated. This puts a strain on documentation and maintaining the knowledge of how to operate the satellite within the team. It is imperative that a core part of the team stays active and involved with operations to stay up to date, and engage in knowledge transfer via peer-to-peer training.

4. CONCLUSION

The toolchain for small satellite operations was steadily improved over time, as operations learned various details about the platform and adjusted to different conditions. The improved toolchain has caused a significant reduction in the overall time required to operate the HYPSO-1 satellite, as well as an increase in data quality. Development based on agile principles with integrated software development and operations has been shown to be beneficial.

It is found to be acceptable for inexperienced teams to not have an exhaustive operational design concept ready by launch. Inexperienced teams most likely struggle to identify the most pressing tasks needing automation prior to launch, so some efforts can prove worthless once the satellite is in orbit. It is therefore valuable to focus on a minimum operation at start, and then develop according to emerging needs, as outlined Section 2.3. Agile software development methods using powerful modern open source tools make it possible to develop ground software tools for planning and data processing as part of satellite operations.

First, operators were being busy scheduling a few images over a time weeks time window, which took many working hours each time. Now, a single operator can schedule images at saturated downlink capacity over a time window of 3-4 days, as well as running the ground processing pipeline to store the images into an archival format. All taking about an hour of operator time.

As of 2023, open source programming tools have become easy enough to use such that integrated software development and satellite operations are more efficient for small satellite operations.

5. REFERENCES

[1] E. Kulu, "Nanosatellite launch forecasts - track record and latest prediction," in *36th Annual Small Satellite Conference*, 2022.

[2] M. N. Sweeting, "Modern small satellites-changing the economics of space," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 343–361, March 2018.

[3] T. Gathmann and L. Raslavicius, "Systems approach to the satellite operations problem," *IEEE Aerospace and Electronic Systems Magazine*, vol. 5, no. 12, pp. 20–24, Dec 1990.

[4] J. L. Anderson, F. J. Kurfess, and J. Puig-Suari, "A Framework for Developing Artificial Intelligence for Autonomous Satellite Operations," in *ESA Special Publication*, H. Lacoste, Ed., Sept. 2009, vol. 673 of *ESA Special Publication*, p. 4.

[5] M. E. Grøtte, R. Birkeland, et al., "Ocean color hyperspectral remote sensing with high resolution and low latency–the hypso-1 cubesat mission," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–19, 2021.

[6] S. Bakken, M. B. Henriksen, et al., "Hypso-1 cubesat: First images and in-orbit characterization," *Remote Sensing*, vol. 15, no. 3, 2023.

[7] A. Dallolio, G. Quintana-Diaz, et al., "A satellite-usv system for persistent observation of mesoscale oceanographic phenomena," *Remote Sensing*, vol. 13, no. 16, 2021.

[8] A. Oudijk, O. Hasler, et al., "Campaign for hyperspectral data validation in north atlantic coastal waters," in *12th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Rome*, 2022.

[9] Ambysoft Inc, "The Criteria for Determining Whether a Team is Agile," https://agilemodeling.com/essays/agilecriteria.htm, accessed on 03 July 2023.

[10] Agile Alliance, "Agile 101," https://www.agilealliance.org/agile101/, accessed on 03 July 2023.

[11] K. Foo, M. Tissera, R. Tan, and K. Low, "Agile development of small satellite's attitude determination and control system," in *2023 IEEE Aerospace Conference*, Mar. 2023, pp. 1–11.

[12] K. Beck, J. Grenning, et al., "Manifesto for Agile Software Development," https://agilemanifesto.org/, 2001, accessed on 03 July 2023.

[13] C. Drumond, "Is the Agile Manifesto still a thing?," https://www.atlassian.com/agile/manifesto, accessed on 03 July 2023.

[14] Planet Labs PBC, "What Is Agile Aerospace? Learn Planet's Approach," https://www.planet.com/pulse/what-is-agile-aerospace-learn-planets-approach, 2019, accessed on 03 July 2023.

[15] S. Bakken, A. Danielsen, et al., "A modular hyperspectral image processing pipeline for cubesats," in *2022 12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Sep. 2022, pp. 1–5.

[16] S. Bakken, R. Birkeland, et al., "Testing of software-intensive hyperspectral imaging payload for the hypso-1 cubesat," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, Jan 2022, pp. 258–264.

[17] S. Bakken, E. Honoré-Livermore, et al., "Software development and integration of a hyperspectral imaging payload for hypso-1," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, Jan 2022, pp. 183–189.