

Use Cases and Algorithms of the EnMAP Mission Planning System

Sven Prüfer^{*}, Christoph Lenzen[†], Sebastian Wiesner[‡], Jonas Krenss[§], Maria Th. Wörle[¶] and Falk Mrowka^{||}
*Deutsches Zentrum für Luft- und Raumfahrt e. V., German Aerospace Center
Münchener Straße 20, 82234 Weßling, Germany*

The Environmental Monitoring and Analysis Program mission (EnMAP) is a German hyper-spectral Earth observation mission, currently scheduled for launch in 2022. The EnMAP Mission Planning System (MPS), developed and operated by the German Space Operations Center (GSOC), is responsible for compiling and maintaining a conflict-free timeline for the EnMAP routine operations. This timeline must not violate any constraints on ground or on board the spacecraft (e. g. regarding power or onboard memory) and it will be regularly commanded to the spacecraft and communicated to ground stations.

The EnMAP planning system is a fully automated system, which provides immediate feedback on any planning relevant information. It is based on GSOC's Reactive Planning framework, which consists of multiple parallel services exchanging information via dedicated messages. As it is the first implementation of Reactive Planning, there are various design decisions that need to be considered carefully in order to keep the generic part as useful as possible for later missions, which is an important aspect of the overall design and development process.

The core planning component of the Reactive Planning framework and thus the EnMAP planning system is the so-called Master Processor for which we describe the various inputs and the respective algorithms used to adapt the model accordingly. This includes the modelling of activities, resources and constraints.

I. Introduction

A. The EnMAP Mission

The Environmental Monitoring and Analysis Program mission (EnMAP) [1] is a German hyperspectral Earth observation mission currently scheduled for launch in 2022. Flying in a sun-synchronous polar orbit with a local time of 11 h (descending node), it will provide a global revisit capability of four days (within an off-nadir range of $\pm 30^\circ$); one full repeat cycle lasts 27 days (398 revolutions). Its dual-spectrometer instrument [2] covers an optical spectral range between 420 and 2450 nm with a spectral sampling distance of 4.8–12 nm and a ground sampling of 30 m, respectively. Together with a high signal-to-noise ratio, these optical characteristics will enable environmental studies with an unprecedented level of detail w.r.t. current spaceborne sensors.

These environmental studies comprise in particular vegetation [3], coastal and inland waters [4], soil [5], geology [6], snow and ice [7], or urban studies [8].

A full description of the EnMAP mission is given by [1], whereas the EnMAP science plan [9] addresses the mission's scientific applications in detail.

B. MPS Design

The EnMAP Mission Planning System (EnMAP MPS) is one of four subsystems within the EnMAP Mission Operations System and is responsible for computing and maintaining a conflict-free timeline for bus and payload

^{*}Mission Planning System Engineer, Mission Technology Department, German Space Operations Center, Sven.Pruefer@dlr.de

[†]Mission Planning System Engineer, Mission Technology Department, German Space Operations Center, Christoph.Lenzen@dlr.de

[‡]Mission Planning System Engineer, Mission Technology Department, German Space Operations Center, Sebastian.Wiesner@dlr.de

[§]Mission Planning System Engineer, Mission Technology Department, German Space Operations Center, Jonas.Krenss@dlr.de

[¶]Deputy Head of Mission Planning Team, Mission Technology Department, German Space Operations Center, Maria.Woerle@dlr.de

^{||}Head of Mission Planning Team, Mission Technology Department, German Space Operations Center, Falk.Mrowka@dlr.de

operations. It is developed and operated by GSOC. For an overview and an explanation of the various novel mission planning concepts introduced by the EnMAP MPS, see [10].

In order to create valid timelines according to the mission requirements, the EnMAP MPS automatically ingests inputs from various other subsystems. This includes

- Earth observation requests from scientific users,
- calibrations,
- orbit updates,
- orbit maneuver opportunities and
- ground station availabilities.

The EnMAP mission supports multiple types of Earth observation requests from scientists, collectively referred to as *acquisition requests* in this paper. To keep the timeline conflict-free the EnMAP MPS models multiple resources such as power and on-board memory and respects corresponding constraints. Furthermore, it implements the mission policy by prioritizing between competing requests in case of conflicts. Another interesting aspect is that the MPS takes into account cloud coverage and sun glint when planning acquisition requests.

The EnMAP MPS regularly informs other subsystems about the current planning state; these exports include for example

- to-be-commanded timelines to the Flight Operations System,
- acquisition request status updates to the scientific users,
- chosen maneuvers for the operators as well as the Flight Dynamics System and
- booked downlink passages from the Neustrelitz Ground Station.

From a software point of view, the EnMAP MPS consists of four main components: *MPS Import Handler*, *MPS Core*, *MPS LEOP Tools* and *MPS Cloud Coverage Handling*. The role of these individual components is explained in [10]. Here, we will mostly focus on the MPS Core component which implements the planning logic for routine operations in EnMAP. It implements the Reactive Planning framework (see Section I.B.1) for the EnMAP mission.

1. Reactive Planning

Reactive Planning (RePL) is GSOC's next-generation mission planning framework, first introduced as Incremental Planning in [11]. It is based on the modern Plains scheduling library developed by GSOC, see e. g. [12] for some Plains concepts. The EnMAP MPS is the first implementation of RePL. In general, it adheres to reactive principles, i. e. responsiveness, resilience, scalability and message driven communication, see e. g. [13]. Here, responsiveness refers to the ability of the system to respond to requests in a timely manner, whenever possible. Resilience means that the system stays responsive even in case of a failure and is able to recover from some failure states. This is achieved by splitting RePL into multiple smaller single-purpose services that have a well-defined boundary and communicate via asynchronous message-passing. Figure 1 shows an overview of the various RePL services that are used by the EnMAP MPS.

At the core of the Reactive Planning system is the Master Processor whose purpose is the actual scheduling of activities and updating of the planning state. Due to the separation of RePL into separate services that communicate via messages, the Master Processor can be modelled as a state machine. Its state consists of the current planning model and state transitions correspond to individual algorithms applied to this model due to received messages. The Master Processor is not concerned with the origin of these messages, it merely knows how to react to each one given the current state. After an algorithm in the Master Processor has changed the state, it will be persisted in a database to be resilient against failures. Other RePL components that need to react to planning state changes receive a stream of these events by listening to the database.

Furthermore, some Reactive Planning components act as gatekeepers or preprocessors for the Master Processor and reduce its task set further. For example, the Input Manager keeps track of a queue of messages that need to be processed next, orders them according to priority and can even trigger an interruption of the current calculation of the Master Processor in case of a very urgent message. In EnMAP in particular, the Acquisition Request Ingestor filters certain invalid requests before they even reach the Master Processor.

To summarize, the general flow of information from the point of view of the Master Processor thus looks as shown in Fig. 2. An algorithm in the EnMAP Master Processor can be described as a map

$$(\text{Current Planning State, Input Message}) \mapsto \text{New Planning State.}$$

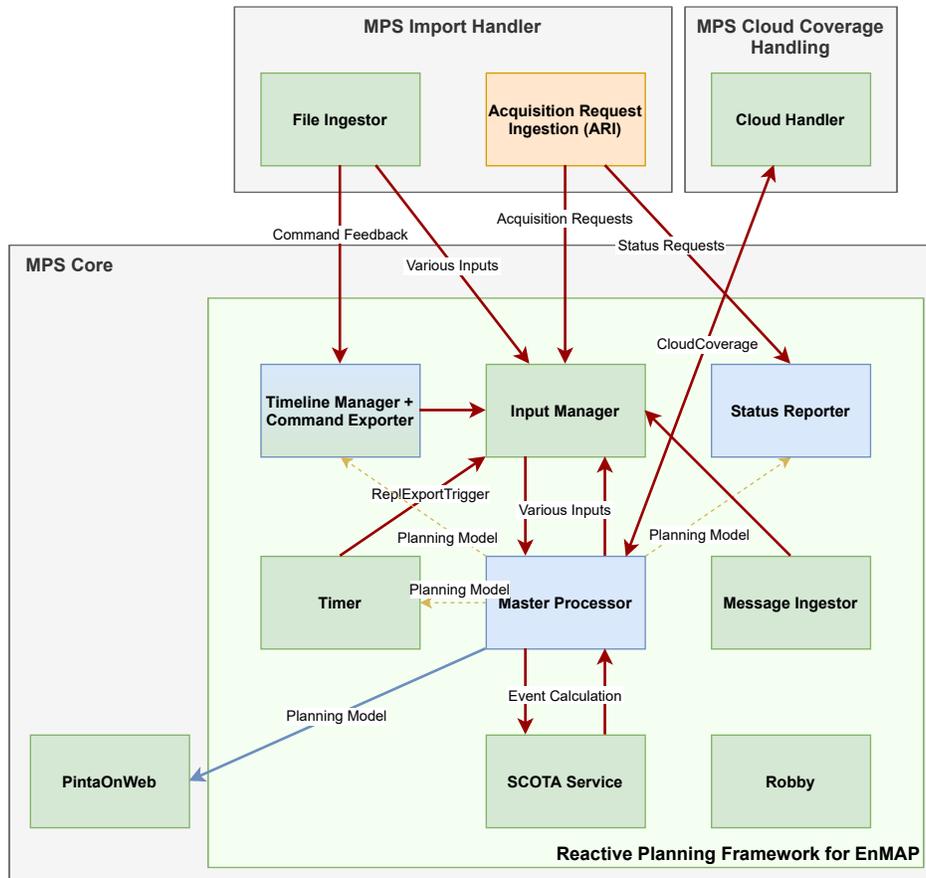


Figure 1 A current overview of the Reactive Planning framework as implemented for the EnMAP MPS. This can be understood as a zoom of Fig. 1 from [10]. Arrows correspond to RePL-internal interfaces: red—messages, blue—file transfers, yellow dashed—database. Green components are completely generic, blue ones are adapted to EnMAP and the orange ones are EnMAP-specific. Robby is the RePL control unit and has message interfaces to all RePL components.

2. The EnMAP Master Processor

The EnMAP Master Processor is a mission-specific implementation of the Reactive Planning Master Processor library. It uses the general infrastructure as described in Section I.B.1 but supplements it by implementing concrete algorithms for the EnMAP-specific inputs as well as adding a few EnMAP-specific interfaces.

These additional interfaces are due to some algorithms requiring input from external services at the time of their execution, for example

- cloud information for an acquisition request may need to be fetched from the Cloud Handler,
- observation opportunities for acquisition requests need to be calculated by the *Spacecraft Orbit and Ground Track Analysis tool* (SCOTA [14]),
- EnMAP orbit information can be fetched from Flight Dynamics, or
- a guidance list for steering the precise satellite attitude during an acquisition or a downlink can be calculated by the Guidance List Service.

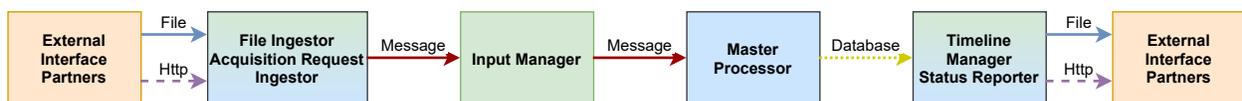


Figure 2 Flow of information within Reactive Planning as seen from the Master Processor point of view.

Table 1 Main EnMAP input types that trigger an algorithm within the EnMAP Master Processor.

Interface Partner	Input Type	Description
Flight Operations System (FOS)	TelemetryFeedback	MPS relevant changes in the telemetry. Not implemented yet.
	OutageRequest	An operator may reserve a time interval for e. g. manual operations. During this time MPS may not plan e. g. any acquisitions.
	OutageCancel	A previously requested outage can be cancelled by an operator.
	CyclogramUploadStatus	Request for status of certain instrument calibration.
	ReplCommandFeedback	After exporting a timeline for commanding, FOS reports back to MPS whether the commands were successfully uplinked.
Flight Dynamics System (FDS)	ManeuverOptions	A list of possible maneuvers calculated by FDS. The EnMAP MPS needs to choose among those.
	ManeuverSelectionAcknowledge	Acknowledge that FDS has registered the chosen maneuver.
	TwoLineElement	An update of the current orbit as a two-line element.
	ReplOrbitRelatedInformation	An update of the current orbit and its related events.
Ground Data System (GDS)	Schedule	An update of one week of possible ground station contact availabilities for uplinks.
Neustrelitz Groundstation (NSG)	AvailabilityInfo	An update of one week of possible ground station availabilities for downlinks.
	ReceptionReport	Report about downlink saved by downlink ground station. Mainly used for post-processing.
	PassRequestResponse	Confirmation or rejection of ordered downlink opportunity.
Data Information Management System (DIMS)	SubmitAcquisitionRequest	Add a new acquisition or calibration request.
	CancelAcquisitionRequest	Cancel a previously submitted acquisition or calibration request.
	CloseAcquisitionRequest	Once an acquisition or calibration request has been done to the customers liking, it can be closed.
	SignalRedo	If the result of an image acquisition was not satisfactory and needs to be repeated, this so-called <i>redo</i> can be triggered.
Cloud Coverage Handling	CloudInfo	Retrieve the cloud information used for a particular acquisition.
	CloudForecastNotification	A new cloud forecast has been obtained and current requests can be updated accordingly.
Reactive Planning	ReplModelUpdate	Modify the planning model directly by importing a Plains model patch to apply.
	ReplTriggerExport	Trigger a scheduled activity, e. g. a command export 45 minutes before an uplink.

In those cases the Master Processor uses a request-response pattern and blocks the execution of the algorithm until it has received an answer. The EnMAP Master Processor was designed, however, to remain responsive even during such time-consuming algorithm steps.

The inputs for the EnMAP Master Processor come from various other EnMAP systems and subsystems as well as from generic Reactive Planning components and operators, see Table 1 for an overview. As described earlier, each one of those inputs is associated with one particular algorithm. However, some of them share of course common sub steps, and special care was taken to extract these into reusable methods. One example of a common sub step is trying to add an image acquisition on the timeline. This is of course needed whenever an acquisition request from a customer is received, but it is also necessary whenever a part of the timeline needs to be replanned due to some other input, such as a cancelled downlink, an interfering maneuver or an outage request.

These inputs are used to schedule the main planning objects, which are

- Acquisition Requests,
- Calibration Requests,
- Downlinks,
- Uplinks,
- Maneuvers, and
- Outages.

Some inputs from Table 1 have a special workflow and are common enough outside the EnMAP MPS that they were partially implemented or defined on a Reactive Planning level. For example the `ReplModelUpdate` is a general interface for tools or operators allowing to update the model directly, e.g. for changing configurations or modifying the plan manually. This is used for example by `PintaOnWeb`, GSOC’s next-generation mission planning GUI tool, the successor to `Pinta`, see e.g. [15, 16]. Another example is the `ReplOrbitRelatedInformation` which is a file containing relevant orbit events calculated by Flight Dynamics that is very common in GSOC missions. Parsing and inclusion of its content into the planning model are done by the generic Reactive Planning code, but the actual updating of the timeline needs to be implemented in an EnMAP-specific algorithm. This is because a change of the orbit data requires e.g. a recalculation of image acquisitions and therefore depends on EnMAP-specific interfaces. The third example is the `ReplCommandFeedback` workflow and concerns the way how the EnMAP MPS exports timelines to the Flight Operations System, where they are commanded to the spacecraft. As comparing exported and uplinked timelines may be computation-intensive this was outsourced to the Timeline Manager component* and a particular RePL communication pattern established. This is described together with the associated EnMAP Master Processor algorithm in Section III.C.

II. Modeling the Planning Problem

A. Basic Model

The Reactive Planning Master Processor models the planning state using GSOC’s generic modelling library called *Plains* (Planning in Scala), which is the successor to *Plato* (Planning Tool), see [17]. Its main objects are activities, resources, constraints and timeline entries.

Activities correspond to planning elements that may be associated with timeline entries in the current timeline and may influence the value of a resource. Plains supports parent–child relationships between activities and this is extensively used in the EnMAP Master Processor to keep a hierarchical structure of relevant objects. For example, acquisition activities are saved in a hierarchy corresponding to their specialization and creation order. All acquisition requests are children of a root acquisition activity and contain the particular acquisition activities[†] necessary to implement it.

In general, there is a common scheme in the planning process in the sense that multiple types of activities go through similar steps:

- An acquisition or calibration request can be ordered, afterwards its *opportunities*, i. e. times when it is physically possible to satisfy the request, can be determined and among those, the Master Processor may plan one which is then commanded to the spacecraft. The activity corresponding to the chosen opportunity will be called acquisition or calibration, respectively.
- A ground station may signal to the EnMAP MPS that they are available at a certain time, the corresponding activity is called an *availability*. The actual physical visibilities during an availability are *opportunities* and when such an opportunity is booked, the corresponding activity is called a transmission.
- Possible maneuvers to keep EnMAP in its reference orbit or for collision avoidance are calculated by the Flight Dynamics System and are then sent to the EnMAP MPS. They correspond to *Maneuver Options* out of which a maneuver is chosen by the Master Processor and communicated to FDS.

Notice that e. g. a single acquisition request thus corresponds to multiple activities in the above hierarchy, which allows, however, to implement constraints and parameters at the correct logical level. For example, parameters specified by the scientific customer are parameters of the acquisition request activity, whereas timing parameters required for the command export that depend on the precise opportunity are parameters of this opportunity. The same holds for constraints as e. g. the timeline entry for an opportunity needs to lie within the timeline entry of the corresponding acquisition request,[‡] whereas the actual acquisition has constraints that it must not take place during a maneuver or that there needs to be a minimum separation between two planned acquisitions.

B. Plains Templates

One interesting feature in Plains is the possibility to define *template activities* which serve as an instruction how to instantiate a set of activities with corresponding resources and constraints or even further templates for a common use

*Notice that the Timeline Manager is also responsible for exporting timelines to FOS in the first place. This means that it is already aware of the expected timeline when it receives feedback and serves therefore as a good first processing place for the command feedback.

[†]Notice that there may be multiple possibilities to take an image of a particular region, e. g. at a later orbit.

[‡]The timeline entry of an acquisition request corresponds to the user-specified validity period.

case. This way, the Plains planning model itself knows how to model certain common scenarios, such as acquisitions or down- and uplinks. For example, acquisition requests by scientific users are modelled as child activities of the root activity *AcquisitionRequests*. This root activity contains templates as children for the various types of acquisitions. Such a template can then be instantiated by filling in the missing parameters at the time of creation. The template for e. g. an *SpcPointRequest* contains:

- Fixed parameters for all *SpcPointRequest*, such as the initial state of the request or the initial count 0 of repeated retries of the request.[§]
- So-called variable declarations, i. e. instructions how to fill in parameters from given inputs at instantiation. For example, an *SpcPointRequest* has as parameters rational coordinates of the target point which need to be specified when instantiating the activity. Such a variable declaration may also specify the creation of an instance-dependent resource modification or constraint.[¶]
- An *opportunity* activity, whose children are the individual opportunities an acquisition request may have. Notice that this activity again has child templates for the activities that represent the individual opportunities together with their variables and constraints.

This is illustrated in Fig. 3. Notice that there are in fact four children templates of an opportunity. This models the fact there are two binary choices for an acquisition that influence the constraints that need to be obeyed. Concretely, an acquisition may be planned with or without preheating and with or without keeping the instrument in stand-by mode afterwards. This corresponds to four different acquisition templates with different constraints, which are all instantiated when an acquisition request is added to the model. However, the planning algorithm will then only create timeline entries for at most one of the four acquisition types, modelling the fact that this particular setting has been planned.

C. Resources

The EnMAP Master Processor models numerous resources, most of them are initialized at project creation, however, corresponding constraints and resource modifications are defined together with the respective activities. See Table 2 for a few examples. Some constraints are valid all the time and are thus associated with the *planning horizon*, see Section II.D, i. e. the time interval which is currently planned. This includes various bounds for e. g. *InstrumentState*, *PayloadDataMemory*, *Power* and *TcExecutionIndicator*. These resources are modified by adding timeline entries to the timeline for corresponding activities with a resource modification. As examples, an acquisition will increase the *Power* resource via a linear function in time, an acquisition with preheat will change the *InstrumentState* resource for a certain amount of time to a different level than an acquisition without preheat and an acquisition or a calibration will increase the *PayloadDataMemory* resource by some amount that depends on the size of the image. Notice that the last resource modification is even more complicated as EnMAP compresses image data and thus the *PayloadDataMemory* consumption caused by an acquisition is not monotonous in time.

Although we do not discuss outages in much detail in this paper, let us mention briefly that they have various constraints on the resources mentioned in Table 2. For example, if an operator requests an outage for manual operations that requires a certain amount of available TTTC slots, or if they require that no payload data remains in memory or that a certain amount of power is available at the start of the outage, this can be easily modelled using the above resources. As the outages take precedence in the scheduling algorithms, these constraints will in fact restrict e. g. possible acquisitions and thus have a non-local influence on the timeline as soon as they are added to the model and trigger a replanning of some time interval.

D. Planning Horizon and Earliest Commanding

The *planning horizon* is the time interval which is considered in the current or next planning run. The EnMAP algorithms plan activities and consider constraints during this interval. As commands can only be uplinked to the spacecraft during an uplink, there is generally no point in modifying spacecraft activities before the next uplink opportunity. This point in time when the next commands can be uplinked is known as *earliest commanding* (EC) and serves as the starting point for most algorithms. It is modelled as an activity with a single timeline entry that is moved

[§]An acquisition request may be redone if the resulting image is of a too low quality. The EnMAP Master Processor keeps track of this count in order to fail the request once it has been retried too often.

[¶]For example, an image acquisition that is to be commanded, requires execution of telecommands during its timeline entry. In order to avoid executing too many commands at the same time, a planned acquisition in a template is defined such that on instantiation it is equipped with a resource modification modelling this telecommand execution. On the other hand, there is a global upper bound defined on the corresponding resource *TcExecutionIndicator* making sure that the constraint is obeyed.

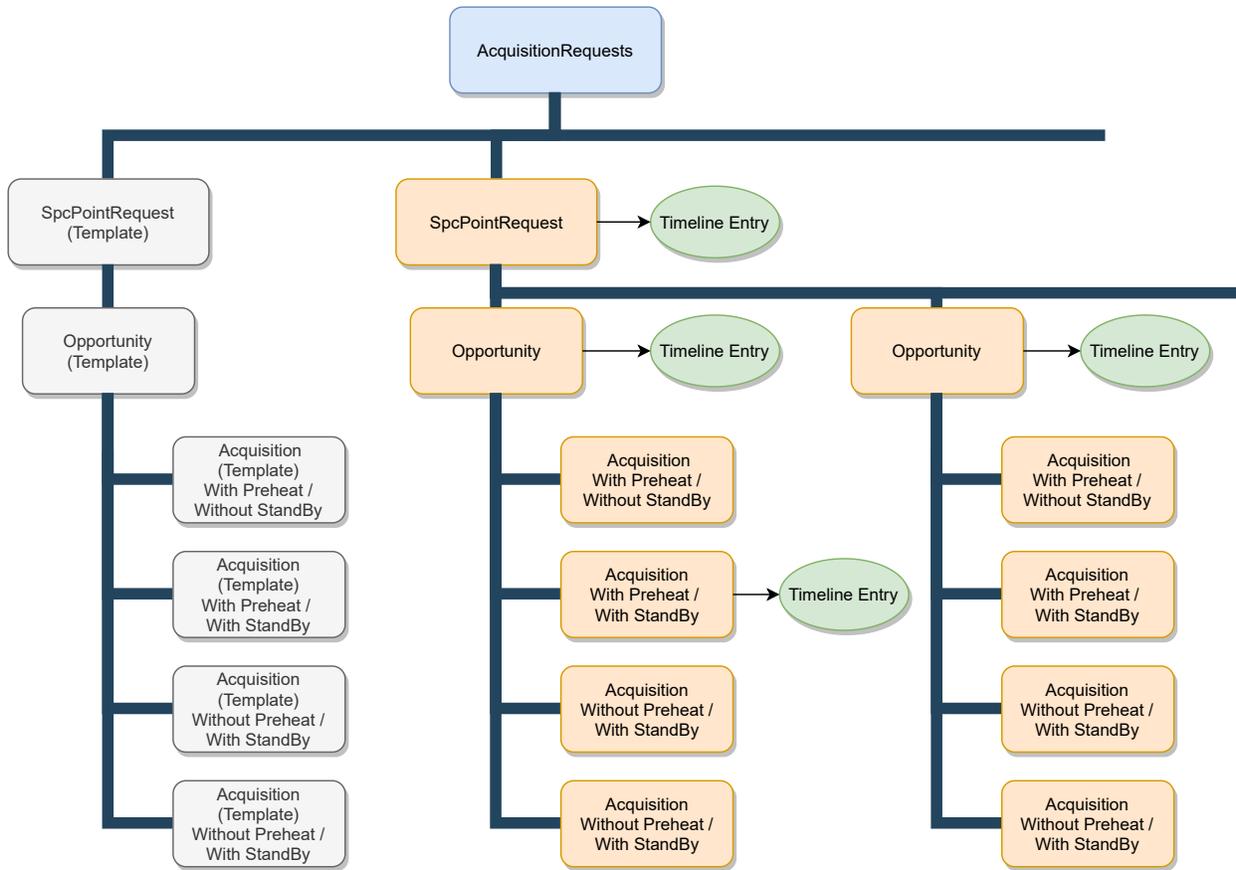


Figure 3 The activity hierarchy associated with an Acquisition Request. Notice that the acquisition with a timeline entry implies that this particular opportunity was planned by the MP to complete the acquisition request.

whenever the earliest commanding changes.

Since in the EnMAP mission commands are sent to the spacecraft for the time from an uplink until the next uplink, the planning horizon nominally starts with the next uplink (i. e. earliest commanding) and is moved to the subsequent uplink at command export for the current earliest commanding. This is illustrated in Fig. 4. Notice that the length of the planning horizon is configurable parameter currently set to 14 days and is determined by the overall EnMAP mission as the value depends on the EnMAP orbit.

This shift of earliest commanding and planning horizon does not only happen during command export but may also be triggered by manual modification or FOS outages that require the EnMAP MPS to skip an uplink. The logic for determining and modifying earliest commanding and planning horizon are thus extracted to reusable methods to be called from various algorithms.

E. Orbit Handling

The EnMAP orbit is calculated and extrapolated by the EnMAP Flight Dynamics System (FDS) and offered in form of the Flight Dynamics service (*FD service*) as well as via Two-Line-Elements (*TLE*) to the users such as e. g. the EnMAP MPS. The FD service is a web server, where one can access an orbit by ID and receives Chebyshev polynomials, allowing for a higher orbit precision for a longer time in the future. Orbit calculations needed by the EnMAP MPS can be configured to use both TLE's or the FD service. When using the FD service, the MPS is said to be in *Orbit-Reference-Mode*. Notice, however, that event calculations based on the orbit data are done not by the Master Processor itself but by SCOTA, which offers a rich and well-tested event calculation library.

The Flight Dynamics System regularly sends updated Two-Line-Elements and Orbit-Related-Information (*ORI*) to

Table 2 Examples of EnMAP resources and some constraints including some activities associated with them.

Resource	Relevant Activities	Description
TttcMemory	Any activity requiring TTTC	Models number of consumed TTTC slots, Upper Bound during planning horizon
PayloadDataMemory	Acquisitions, Downlinks	Models consumed number of bits in payload data memory, Upper and Lower Bounds during planning horizon, takes into account compression of data and is reduced by Downlinks
Power	Any activity modifying the power consumption or supply	Includes lots of constraints and non-trivial resource modifications, <i>InstrumentOff</i> activities mark times when the spacecraft turns towards the Sun and power is recharged
InstrumentState	Instrument activities	Finite amount of states, used to model no-overlap regions around instrument activities
ManeuverIndicator	Maneuver	Binary resource indicating when a maneuver takes place, various instrument and downlink activities cannot take place during a maneuver
OutageIndicator	Outage	Binary resource indicating when an outage takes place, in fact there are multiple types of outages modelled by separate resources and with different constraints
HsiAuxFiles	Downlinks	Model number of current HSI files, Upper Bound of 5, deleted by individual commands to make sure they are always downlinked
TcExecutionIndicator	Any activity requiring TTTC	Binary variable modelling execution of telecommands, used to make sure that minimum distance between two telecommands is at least one second

MPS which contains orbit numbers, information about sun shadows, calibration possibilities as well as ground station contacts. In Orbit-Reference-Mode, upon receiving such an ORI file the EnMAP MPS inquires the ID for the newest orbit and uses this for any calculations requiring detailed orbit information.

Another aspect that is related to orbit handling is the calculation of detailed attitude quaternions whenever a precise attitude control of the spacecraft is required. These so-called *guidance lists* are calculated by another web service from FDS, the guidance list generator (GLG). The EnMAP MPS can request calculations of guidance lists using the newest orbit whenever this is necessary for appropriate activities, such as e. g. maneuvers, downlinks and acquisitions.

These orbit-related interfaces are summarized in Fig. 5.

F. Benefit

To schedule a single acquisition out of multiple potential opportunities EnMAP algorithms usually order them by their *benefit* and then add them to the timeline in descending order. Therefore, the EnMAP Master Processor provides a function that maps a planning model and an opportunity of an image acquisition to a rational number. This function is a sum of eight terms with configurable weights and tries to incorporate in particular

- the priority of the initial acquisition request,
- cloud forecasts that are better than the statistical cloud coverage,
- statistical cloud coverage of the target, and
- robustness of the timeline against moving of already planned acquisitions or adding new ones.

For a more detailed description in particular of the cloud coverage influence on the benefit, see [10].

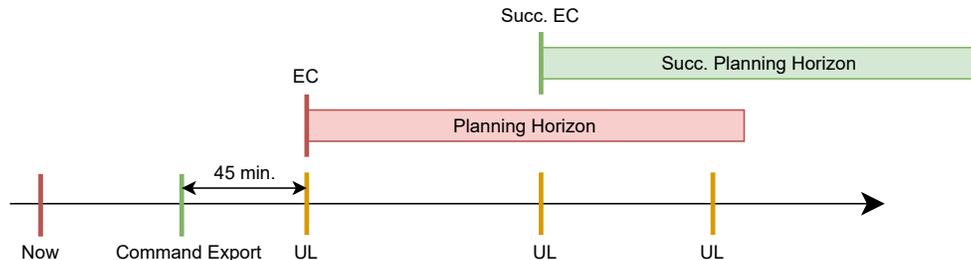


Figure 4 Shift of planning horizon and earliest commanding after command export. *UL* refers to uplinks.

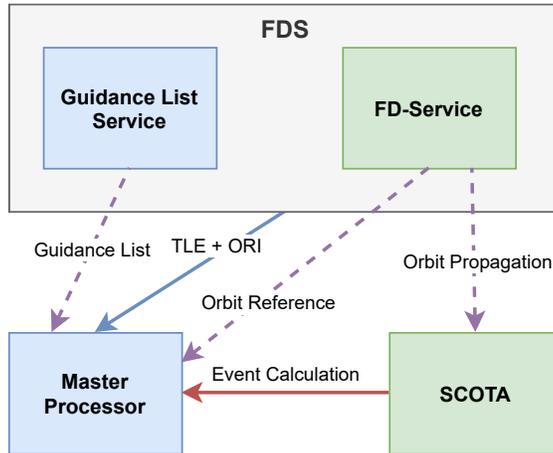


Figure 5 Orbit related interfaces from the point of view of the EnMAP Master Processor. As in the other figures, blue refers to file transfers, red to messages and purple dashed to HTTP requests. Notice that the arrows show the flow of information, but some of these interfaces are triggered by request of the Master Processor.

III. Use Cases and Their Algorithms

This section describes a couple of scenarios within EnMAP and their corresponding algorithms. We will start with the most important one, planning acquisition requests, as this is the main objective of EnMAP. In order to command such a user request to the spacecraft it is necessary to schedule uplinks and export corresponding command files before such an uplink, and we will thus look at these use cases afterwards.

A. Acquisition Requests—SpcPointRequest

As explained in Section I.B, an acquisition request refers to a request by a scientific user to take a picture of a certain region on Earth using EnMAP’s multi-spectral camera. The user interacts with DIMS, see Table 1, which serves as the user interface and adds various inputs to those from the user. An acquisition request arriving at the EnMAP MPS therefore supplies various administrative settings as well as image-related configurations. In particular, there are multiple types of acquisition requests, namely SpcPointRequest, SpcAreaRequest, AreaCoverageRequest and CalibrationRequest. The simplest one is the SpcPointRequest, where the camera points towards a user-defined point target while passing by. This is the use case we want to explain further in this section. An SpcPointRequest as received from DIMS contains in particular the following data:

- a target point defined via its longitude and latitude,
- an acquisition time window, i. e. a time interval during which the request may be satisfied,
- the length of the swath used for the image,
- the required direction (descending or ascending) of the spacecraft over the target,
- a maximum off-nadir angle under which the camera may take the image,
- restrictions on allowed sunglint during the acquisition,
- restrictions on allowed cloud coverage probability at target, and
- a maximum number of allowed repetitions of the image acquisition in case the resulting image data is not considered adequate.

The algorithm is illustrated in Fig. 6. After checking that the ID of the request has not been used yet, the first step of the EnMAP algorithm is to create corresponding activities with all the specified parameters in the Plains model of the EnMAP Master Processor as described in Section II.B. This means that a template is instantiated and various child activities and further templates are added and the inputs from the user are attached to the corresponding activities. This includes in particular sunglint and cloud information as well as a timeline entry for the acquisition time window. Notice that creating the individual opportunities is the most time-consuming task here, as they depend on various external calculations involving the orbit.

Using the parameters specified by the user as above, a request to SCOTA is sent to calculate all acquisition opportunities during the intersection of the acquisition time window and the planning horizon. The replies contain

swathes implementing the request and are processed individually. For each one of them the algorithm creates the new opportunity and searches for a corresponding old opportunity by checking for overlaps within a configurable maximal time range. If it finds an old opportunity it is either deleted together with its descendants or, if it was already commanded, new opportunity is dropped.

The creation of the new opportunity from the data received from SCOTA includes calculating and filling in the timing information for a potential later commanding as well as the concrete cloud and sunglint information for this particular time, if the request has conditions for those. In order to get this information, the EnMAP Master Processor requests the current cloud information from the Cloud Handler and adds this data to the opportunity. The sunglint information is already contained in the SCOTA response. Furthermore, the timeline entry for the opportunity is created and, if an acquisition was planned beforehand and the opportunity is updated, the acquisition together with its resource modifications and constraints is updated.

Notice that at this point no new acquisition has actually been planned, only opportunities (and already scheduled acquisitions) have been updated according to the latest data. The next step is to update the benefits according to Section II.F which can then be used to actually plan an acquisition for the request if possible. Notice that as described in Section II.B an acquisition request opportunity has multiple possible types of acquisitions corresponding to preheat or stand-by options and this distinction is also part of this step. The choice is done as follows: The algorithm tries to add the acquisitions satisfying their constraints one by one and compares them pairwise returning the acquisition which increases the benefit the most, or if equal, the one which reduces the time until the request is finished the most. If this is also equal, an arbitrary choice is made.

For the chosen opportunity to schedule, the corresponding acquisition activity is added by a general method that replans instrument related activities during a specified interval. The interval used for replanning here covers the opportunity timeline entry as well as possible already existing acquisition timeline entries for this opportunity. This step, called *RePlanSurrounding* in Fig. 6, first removes all instrument activities from the timeline including various resource modifications associated with them and all downlinks starting with the beginning of the interval to be replanned. Notice that replanning of downlinks at this step is necessary as added or modified image acquisitions may make it necessary to schedule additional downlinks or move existing ones. In a second step the cleared interval is refilled by trying to schedule opportunities in descending priority order, i. e. according to their benefit, and only afterwards scheduling downlinks as needed. This means that an image acquisition may block a downlink if there is still enough memory available or some other downlink in the future can be used. Notice that when these acquisitions are replanned, the MP also updates the associated guidance lists, such that they are always kept up to date.

In case replanning downlinks requires uncheduling of opportunities we then try to schedule those again outside the opposing downlink to see if there is another possibility. As this may cause a planning loop if e. g. other possibilities interfere with other downlinks, the algorithm keeps track of an *algorithm depth* corresponding to the amount of replans there were done so far. If this configurable number is reached, the algorithm aborts and returns the last result.

Notice that the functionality for replanning an interval of instrument activities including downlinks and the subsequent conflict resolution is needed at various places and is thus kept rather general and reused in many algorithms.

B. Uplinks

Whenever GDS sends an updated ground station schedule to MPS this is processed by the Master Processor in order to update uplink activities and all activities that might be interfering with the new uplink schedule. Notice that for EnMAP the actual booking for the uplink passages is done before processing within the EnMAP MPS and all supplied uplinks from GDS can and should be used by the Master Processor. The algorithm is illustrated in Fig. 7.

A GDS schedule contains ground station passages for multiple satellites and ground stations. The first step for this algorithm is thus to filter for relevant entries, i. e. those for EnMAP and for the Weilheim ground station which is currently the only supported one. The schedule from the file replaces all uplinks in a specified time interval which is typically two weeks.

The algorithm then removes all availability timeline entries for the ground station and adds the new ones according to the new input. Furthermore, it removes all opportunity timeline entries in the time interval to be updated and then adds new ones according to a request via SCOTA. Notice that it is possible that an uplink opportunity overlaps with the boundary of the planning horizon, in which case the algorithm merges the corresponding timeline entries.

After updating the opportunities it is necessary to schedule the actual uplinks and update the rest of the model. The first step here is to remove all transmissions (corresponding to planned uplinks) in the interval to be updated and after the current earliest commanding. The algorithm then plans new transmissions for every opportunity by finding the

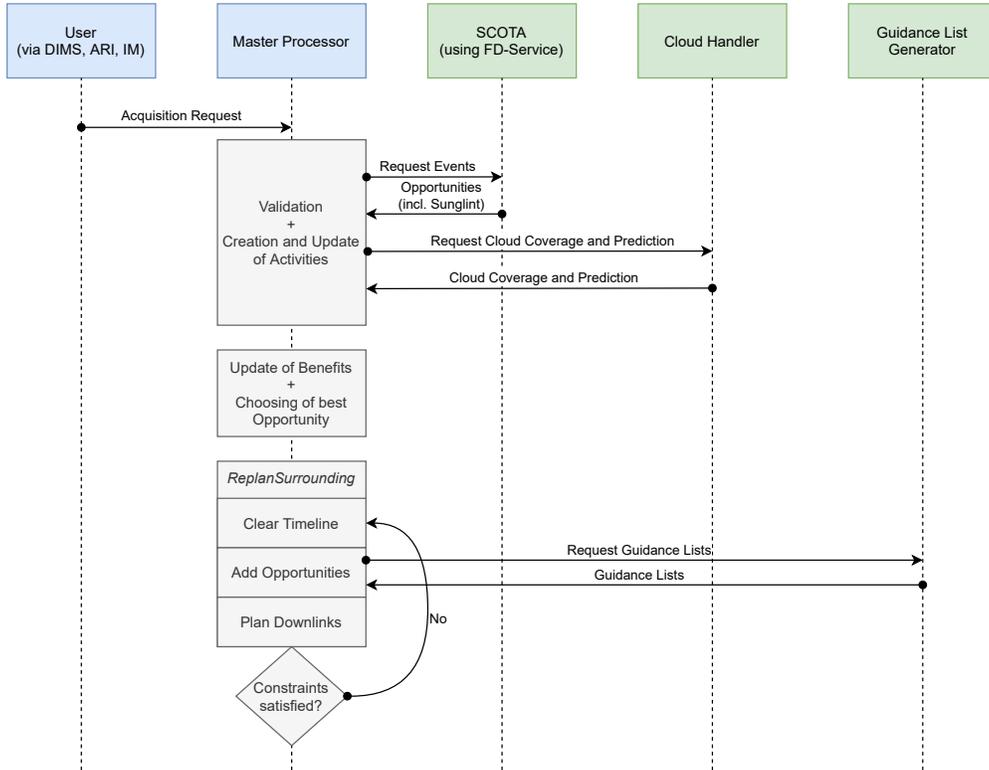


Figure 6 Schematic description of the algorithm for processing an SpcPointRequest.

largest interval contained within an opportunity which satisfies all constraints.

Next, the new earliest commanding is determined and updated as described in Section II.D. Notice that it is possible that the old upcoming uplink was moved to a slightly later time by the update and thus the earliest commanding may move. However, the algorithm does not support moving earliest commanding to a newer time since commands until the old earliest commanding have already been uplinked.

Once the earliest commanding is updated, the algorithm sets the new planning horizon, updates the timeline entry modelling the default power supply for the far future, sets the instrument to *off* between the new earliest commanding and the end of the planning horizon. Notice that it is possible that due to the movement of uplinks e. g. an additional image acquisition is now taking place during two uplinks and thus the buffer for time-tagged telecommands might not suffice. In that case the algorithm removes acquisitions until the TTTC buffer has no more conflicts and then recreates downlinks as needed similarly as described in Section III.A.

C. Command Export and Command Feedback

The command export workflow is a bit more involved than the previous examples as it contains asynchronous interfaces with the commanding system as well as an intermediate Reactive Planning component, the Timeline Manager. As it is possible that uploading of commands fails, the Master Processor needs to be prepared for a failure. The general workflow is described in Fig. 8.

As command export is a functionality that is required for most satellite missions at GSOC the basic interface is implemented as part of the generic Reactive Planning Master Processor but the concrete modification for the planning model in case of an export is defined as an EnMAP-specific function.

The command export is triggered by the RePL Timer component which sends a `ReplExportTrigger` message to the MP exactly 45 minutes before an uplink timeline entry. This triggers an EnMAP algorithm taking care of various settings relevant for the export only, e. g. it will set timeline entries for uplink and downlink related activities that don't matter for planning but need to be included in the timeline such that the Timeline Manager exports telecommands for these activities. This includes TC for autonomous telemetry, auxiliary file deletion and download flags. It also *locks*

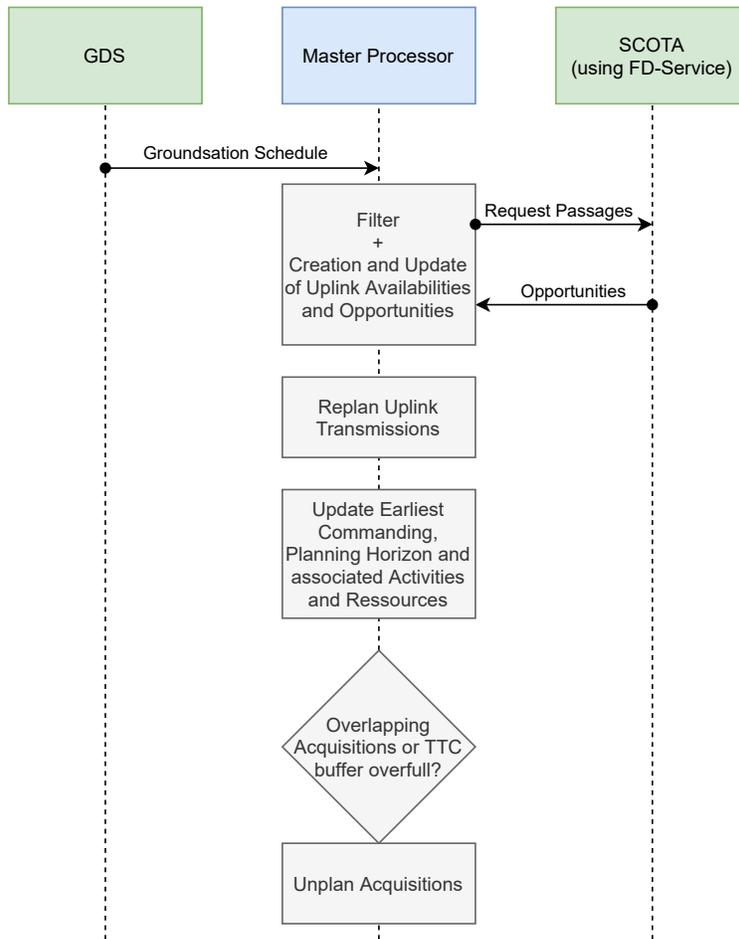


Figure 7 Schematic description of the algorithm for processing a GDS schedule.

timeline entries between earliest commanding and the succeeding uplink such that any subsequent algorithms are aware that these timeline entries cannot be modified or deleted while the command export loop is running. The only later modification for these timeline entries is possible by the corresponding command feedback. At the end, it also shifts the earliest commanding as well as the planning horizon to the succeeding uplink.

The MP creates a Plains project parameter to inform the Timeline Manager that it should export commands. In case the uplink takes place during a FOS outage which blocks timeline exports it will furthermore let the Timeline Manager know to consider the command export immediately as failed. This way the EnMAP MPS can reuse the nominal workflow as the Timeline Manager needs to be able to deal with failed command exports in any case.

Next, the Timeline Manager (TM) inspects the timeline and exports telecommands according to its configuration. Here it is necessary to identify activities, their timeline entries as well as project and activity parameters from the planning model and map them to flight operation procedures (FOP's) in a configurable way. As FOP parameters may use very different formats compared to the planning model, the Timeline Manager offers a set of common functions and transformations to get the expected result. Nevertheless, it depends on quite a bit of EnMAP-specific code.

Once the TM has exported telecommands it waits for the feedback from the commanding system. In particular, it remembers the exported commands together with the activities they correspond to. This means that once it receives feedback from FOS it can calculate the difference between exported and uplinked commands and thus determine a patch changing the modelled timeline for the exported time interval to the actual uplinked timeline. This patch is then sent as part of a `ReplCommandFeedback` message to the Master Processor via the Input Manager.

When processing the command feedback, the Master Processor first makes sure that it did indeed expect this feedback. As it is non-trivial to process command feedbacks out of order both Timeline Manager and Master Processor reject any unexpected feedbacks. After ingesting one, the Master Processor first removes timeline entries of non-executed

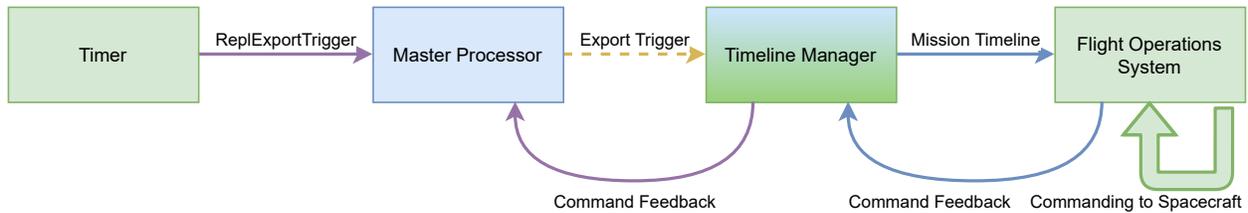


Figure 8 Complete workflow for a command export.

commands and adapts the timeline according to the potentially partially executed activities. For example, if a file deletion failed, the MP needs to model the still existing file on the satellite. As another example, if an image acquisition did not get uplinked the heater for the preceding uplink will get turned off after a timeout on the satellite and thus the planning model needs to be adapted accordingly. Once this is done, the acquisition requests whose commanded acquisitions could not be executed are replanned after the next earliest commanding.

IV. Conclusion and Outlook

So far we have seen how the EnMAP Master Processor makes use of various features of the Plains modelling language and implements the functionality of the Reactive Planning framework. The generic functionality is separated from the EnMAP-specific functionality and reusable for future satellite missions at GSOC.

Of course, there are always many aspects that one may improve. In Section III.A we have described how the Master Processor schedules `SpcPointRequests`, but there are also various types of calibration requests as well as `SpcAreaRequests` in which one wants to take an image covering a big part of a specified polygon on the surface of Earth. These types of requests follow a similar, but different scheduling approach. However, there is one additional type of request that is left for future development: the `AreaCoverageRequest` which corresponds to a large surface polygon that needs to be covered in multiple swaths. Since this poses interesting questions not just on the scheduling side but also on the mathematics of spherical polygons, GSOC’s SCOTA will be extended accordingly.

There are various places in the Master Processor where one may improve the efficiency by incorporating up-to-date telemetry into the planning input. This feature is called `TelemetryFeedback`, but it has not been implemented yet.

Besides new features there are a few things that might be pulled out of the EnMAP MPS and reused in a generic way. One such example is the generic instrument chain which describes a situation as with image requests in EnMAP where one wants to try to schedule a sequence of activities with some common logic how to deal with intermediate states such as e. g. instrument modes. This code is currently being generalized and moved to the generic Plains planning library.

Of course, EnMAP will not remain the only implementation of the Reactive Planning framework. In fact, the MPS for the TDP-1 mission is being reimplemented at the moment using RePL. How this rewrite and update is done while the mission is flying, will be the content of [18].

One more major feature that complements the Reactive Planning framework is the possibility to monitor and manipulate the planning state in a manual way. This will be offered by the upcoming `PintaOnWeb` web application, see [16] for a current description of the predecessor PINTA.

In any case, the EnMAP MPS will certainly evolve during the EnMAP mission and be adapted to the changing mission requirements, and we are looking forward to this challenge.

Acknowledgements

Supported by the DLR Space Administration with funds of the German Federal Ministry of Economic Affairs and Technology on the basis of a decision by the German Bundestag (50 EE 0850).

References

- [1] Guanter, L., Kaufmann, H., Segl, K., Foerster, S., Rogass, C., Chabrillat, S., Kuester, T., Hollstein, A., Rossner, G., Chlebek, C., Straif, C., Fischer, S., Schrader, S., Storch, T., Heiden, U., Müller, A., Bachmann, M., Mühle, H., Müller, R., Habermeyer, M., Ohndorf, A., Hill, J., Buddenbaum, H., Hostert, P., van der Linden, S., Leitão, P. J., Rabe, A., Doerffer, R., Krasemann, H., Xi, H., Mauser, W., Hank, T., Locherer, M., Rast, M., Staenz, K., and Sang, B., “The EnMAP Spaceborne Imaging

- Spectroscopy Mission for Earth Observation,” *Remote Sensing*, Vol. 7, No. 7, 2015, pp. 8830–8857. doi:10.3390/rs70708830, URL <http://www.mdpi.com/2072-4292/7/7/8830>.
- [2] Sang, B., Schubert, J., Kaiser, S., Mogulsky, V., Neumann, C., Förster, K.-P., Hofer, S., Stuffer, T., Kaufmann, H., Müller, A., Eversberg, T., and Chlebek, C., “The EnMAP hyperspectral imaging spectrometer: instrument concept, calibration, and technologies,” *Proc. SPIE*, Vol. 7086, 2008, pp. 708605 1–15. doi:10.1117/12.794870, URL <http://dx.doi.org/10.1117/12.794870>.
- [3] Knipling, E. B., “Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation,” *Remote Sensing of Environment*, Vol. 1, No. 3, 1970, pp. 155–159. doi:10.1016/S0034-4257(70)80021-9, URL [https://doi.org/10.1016/S0034-4257\(70\)80021-9](https://doi.org/10.1016/S0034-4257(70)80021-9).
- [4] Lee, Z., Carder, K. L., Mobley, C. D., Steward, R. G., and Patch, J. S., “Hyperspectral remote sensing for shallow waters: 2. Deriving bottom depths and water properties by optimization,” *Appl. Opt.*, Vol. 38, No. 18, 1999, pp. 3831–3843. doi:10.1364/AO.38.003831, URL <http://ao.osa.org/abstract.cfm?URI=ao-38-18-3831>.
- [5] Ben-Dor, E., Chabrilat, S., Demattê, J. A., Taylor, G., Hill, J., Whiting, M., and Sommer, S., “Using Imaging Spectroscopy to study soil properties,” *Remote Sensing of Environment*, Vol. 113, 2009, pp. S38–S55. URL https://www.researchgate.net/publication/223042783_Using_Imaging_Spectroscopy_to_study_soil_properties.
- [6] Hunt, G. R., “Spectral signatures of particulate minerals in the visible and near infrared,” *Geophysics*, Vol. 42, No. 3, 1977, pp. 501–513. doi:10.1190/1.1440721, URL <http://dx.doi.org/10.1190/1.1440721>.
- [7] Dozier, J., Green, R. O., Nolin, A. W., and Painter, T. H., “Interpretation of snow properties from imaging spectrometry,” *Remote Sensing of Environment*, Vol. 113, 2009, pp. S25–S37. doi:10.1016/j.rse.2007.07.029, URL <https://doi.org/10.1016/j.rse.2007.07.029>.
- [8] Heldens, W., Heiden, U., Esch, T., Stein, E., and Müller, A., “Can the Future EnMAP Mission Contribute to Urban Applications? A Literature Survey,” *Remote Sensing*, Vol. 3, No. 9, 2011, pp. 1817–1846. URL <http://elib.dlr.de/70793/>.
- [9] Guanter, L., Kaufmann, H., Foerster, S., Brosinsky, A., Wulf, H., Bochow, M., Boesche, N., Brell, M., Buddenbaum, H., Chabrilat, S., Hank, T., Heiden, U., Heim, B., Heldens, W., Hill, J., Hollstein, A., Hostert, P., Krasemann, H., Leitão, P. J., van der Linden, S., Mauser, W., Mielke, C., Müller, A., Oppelt, N., Roessner, S., Röttgers, R., Schneiderhan, T., Staenz, K., and Segl, K., “EnMAP Science Plan,” Tech. rep., GFZ Data Services, 2016. doi:10.2312/enmap.2016.006, URL <http://doi.org/10.2312/enmap.2016.006>.
- [10] Fruth, T., Lenzen, C., Gross, E., and Mrowka, F., “The EnMAP Mission Planning System,” *2018 SpaceOps Conference*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-2525, URL <https://doi.org/10.2514/6.2018-2525>.
- [11] Wörle, M. T., Lenzen, C., Göttfert, T., Spörl, A., Grishechkin, B., Mrowka, F., and Wickler, M., “The Incremental Planning System – GSOC’s Next Generation Mission Planning Framework,” *13th International Conference on Space Operations*, Pasadena, California, USA, 2014. URL <http://elib.dlr.de/89586/>.
- [12] Lenzen, C., Klaehn, R., Prüfer, S., and Wörle, M. T., “Polygon stacks and time reference conversions,” *CEAS Space Journal*, 2020. URL <https://elib.dlr.de/133904/>.
- [13] Bonér, J., Farley, D., Kuhn, R., and Thompson, M., “The Reactive Manifesto,” , 2014. URL <https://www.reactivemanifesto.org/>.
- [14] Gross, E., Fruth, T., Dauth, M., Petrak, A., Hinze, A., and Mrowka, F., “SCOTA – The Mission Planning Orbit Analysis Tool at GSOC,” , 2021. To Be Published.
- [15] Nibler, R., Mrowka, F., Wörle, M. T., Hartung, J., and Lenzen, C., “PINTA and TimOnWeb - (more than) generic user interfaces for various planning problems,” *IWPSS 2017 - 10th International Workshop on Planning and Scheduling for Space*, 2017. URL <https://elib.dlr.de/114095/>.
- [16] Nibler, R., Mrowka, F., Hartung, J., Schneider, A., and Brogl, S., “PINTA – one Tool to plan them all,” , May 2021. To Be Published.
- [17] Lenzen, C., Wörle, M. T., Mrowka, F., Spörl, A., and Klaehn, R., “The Algorithm Assembly Set of Plato,” *SpaceOps 2012 Conference*, SpaceOps Conferences, American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-1255731, URL <https://doi.org/10.2514/6.2012-1255731>.
- [18] Wörle, M. T., Lenzen, C., Wiesner, S., Prüfer, S., Petrak, A., and Müller, K., “Replacing the TDP-1 Mission Planning System – more than just another Technical Demonstration Project,” , 2021. To Be Published.