The Mission Planning System for the Firebird Spacecraft Constellation

Maria Theresia Wörle*, Andreas Spörl*, Jens Hartung*, Christoph Lenzen* and Falk Mrowka†

Deutsches Zentrum für Luft- und Raumfahrt e.V., German Aerospace Center, Münchener Straße 20, 82234 Weßling, Germany

The Firebird mission comprises the two spacecraft TET (launched July 22nd 2012) and BIROS (launch foreseen for May 25th 2016), both carrying a combined infrared-optical camera system as primary payload as well as several additional hard- and software experiments. Our Mission Planning team at the German Space Operations Center (GSOC) is responsible for generating conflict-free timelines for commanding payload and so-called background sequence operations for both spacecraft in accordance with all given spacecraft and ground-related constraints and customer requirements. Therefore, a Mission Planning system has been prepared and is continuously developed further with continuously changing space segment capabilities throughout the different project phases. The paper at hand describes the main components and their set-up, e.g. the semi-automated planning tools and the newly implemented interactive order interface for the customers. Furthermore, the decision to which extent a combined system is set up for both spacecraft as well as the advantages of being able to rely on a generic, configurable tool suite, modeling language and scheduling algorithm assembly are discussed.

I. Introduction

WITH the end of the TET-1 OnOrbitVerfication mission[1,2] in November 2013 the so-called Operational Phase 1 of the Firebird mission[3] started with TET ("TechnologieErprobungsTräger") being the first of two spacecraft dedicated to the detection and monitoring of high-temperature events on Earth and performing other Earth observation tasks with a combined infrared-optical camera system. This year (currently foreseen launch date: May 25th), TET will be accompanied by the BIROS ("Bi-spectral InfraRed Optical System") spacecraft which carries the same camera system as primary payload and several additional technological experiments for hard- and software innovations[4]. The camera system comprises a bi-spectral infrared sensor and three optical channels, with an interface to perform onboard processing of the data for hot-spot recognition.

Although the Firebird mission is a non-commercial project and the overall approach is rather experimental, a continuous reliability of the mission operations is required from GSOC to maximize the scientific output and amount of data for the end-user community. Integrated in the ground segment for mission operations, the Mission Planning system (MPS) is responsible for generating consistent, conflict-free timelines for commanding routine payload and so-called background sequence operations of the two spacecraft TET and BIROS. This background sequence (BGS) contains the sequence of to-be-commanded Flight Operations Procedures (FOPs) for several regular housekeeping dumps and transmitter switches.

Therefore, a semi-automated planning system has been established that collects requests and information from users, spacecraft and groundstations, and generates the executable timelines in daily planning runs with an order deadline of six hours. It reuses generic GSOC planning, analysis and visualization functionalities combined with generic as well as project-specific modeling, algorithms and plug-ins. Furthermore, a new application was developed to enable the user to preview the spacecraft visibilities, calculate future opportunities to acquire primary payload data of regions of interest on Earth, and allow for creating the according Planning Requests that are then sent to the core MPS.

One special challenge of the Firebird mission is that onboard re-configurations as well as changes of the on-ground requirements made and still make necessary several extension stages and adaptations of the MPS necessary.

* Mission Planning System Engineers, Mission Operations Department, German Space Operations Center.
† Head of Mission Planning Team, Mission Operations Department, German Space Operations Center.

The same will apply for the Operational Phase 2 with BIROS included into the space segment. Changes of the onboard software and other capabilities are expected to take place throughout the whole mission lifetime. In addition, some of the experiments onboard optionally might become integrated into the nominal payload operations[4].

The paper at hand describes the components of the Firebird MPS and how they are set-up from generically available GSOC Mission Planning tools and libraries. It will be explained to which extent it was decided to combine the systems for TET and BIROS into one unified MPS, and examples will be outlined how much we profit from being able to rely on the well-proven, highly configurable generic basis to cope with the given requirements and continuous change requests.

II. The Components of the Firebird Mission Planning System and their Set-up

Figure 1 gives an overview of the Firebird Mission Planning system and shows how it is embedded within the mission operations ground segment and which internal and external interfaces are to be served.

In the following, these components and their main functionalities and set-up will be described in more detail.

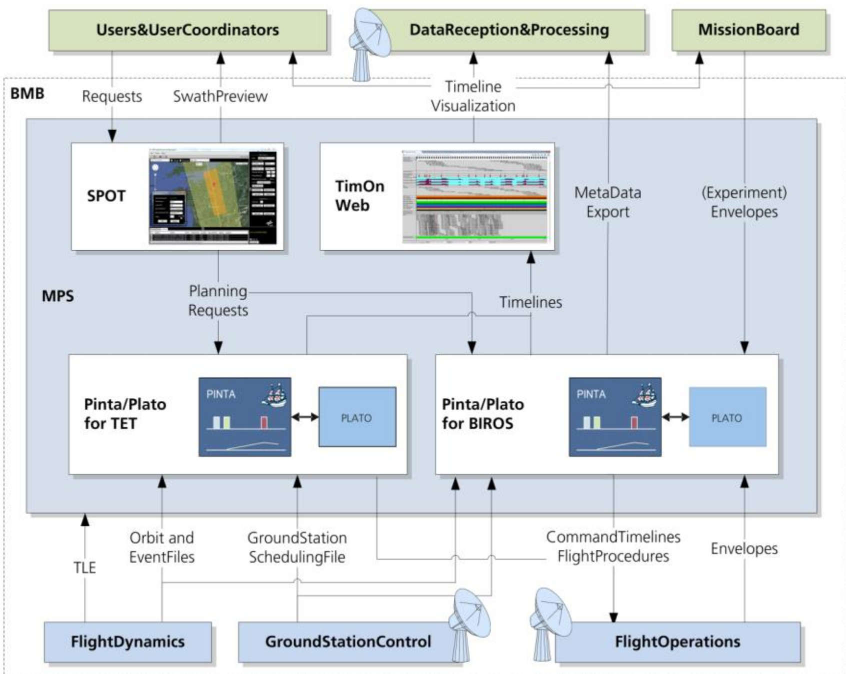

Figure 1. Overview: The Firebird MPS, its components and its interfaces. It is shown how the MPS is embedded in the ground segment for operations (BMB, "Bodensegment Mission und Betrieb") and which relations to other project entities exist.

A. SPOT – The Swath Preview and Ordering Tool

As the Firebird mission's main goal is the operation of the two spacecraft's infrared-optical camera systems acquiring Earth observation data, the users that are allowed to create so-called Planning Requests for future acquisitions need the possibility to preview the orbital position and therefore which region on Earth the instrument is able to acquire/monitor when. In other missions operated at GSOC, the Mission Planning team already delivers the so-called Swath Preview service to other parts of the ground segment with the help of our generic orbit propagation and event calculation library SCOTA. There, only the core calculations are provided as a service, whereas the order interface to external customers itself has been established by others.

For the Firebird mission, a new approach was chosen: The ordering system should have a direct interface to GSOC and so was decided to be established on our own. SPOT, the Swath Preview and Ordering Tool, was set up as a GUI application and validated throughout the first months of TET. Figure 2 shows a snapshot of its main window and a little selection window.

The user authentication and configuration as well as the provision of a current TLE (Two-Line Element) for the orbit propagation and the map pane are handled by a server at GSOC while all other functionalities are handled on client side where installed.

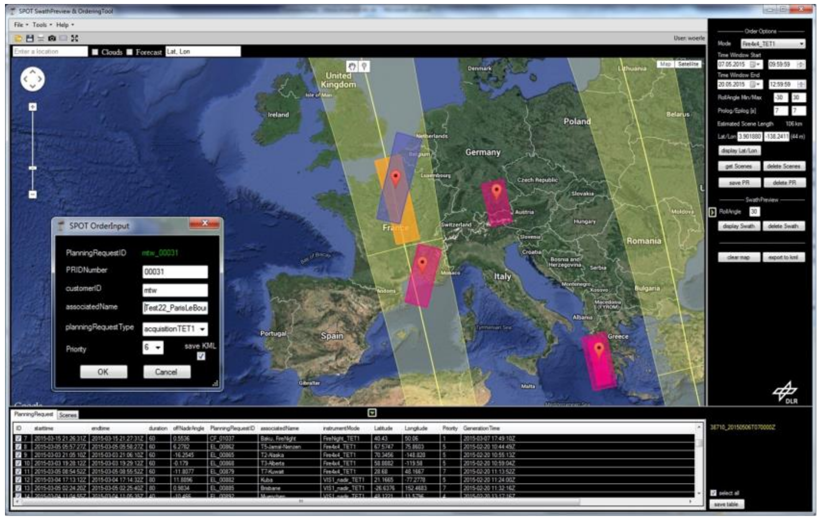

Figure 2. Snapshot of the Swath Preview and Ordering Tool. The pane is using Google Maps. Light yellow: swath of TET for the specified timeframe. Purple: re-loaded Planning Requests. Blue and orange: Future night and day visibilities of the selected target. Furthermore, the selection window for specifying parameters of a Planning Request to be newly created is shown.

Among others, SPOT comprises the following functionalities: For the TLE currently loaded and the selected input parameters, such as the desired period of time, it

- provides orbit propagation and the visualization of the spacecraft ground track and the potentially visible swath,
- provides calculation and visualization of acquisition opportunities for the currently selected point of interest and parameter restrictions, i.e. the start and stop times and roll angle are determined to see this point of interest based on the currently available orbit information, and the list of possible scenes to acquire the interesting area during the considered timeframe are displayed to the user for selection, and
- enables the interactive generation, storage and visualization of Planning Requests for one or more of the selected acquisition opportunities, specifying e.g. a priority value as additional parameter. This way it is ensured that Planning Requests in the correct XML format are generated for transfer to GSOC, and that the content represents an acquisition of the region of interest that would be feasible with the current orbit information and parameter restrictions, e.g. the maximally allowed roll angle.

SPOT supports different instrument modes for the calculations and visualization that differ in the (number of) camera channels to be used and thus the Field of View.

Furthermore, for instance, it enables the user to reload previously created Planning Requests and to save the currently selected visualization to KML files.

Thought as a prototype first, SPOT has proven robust and very useful and usable for the customers working with it immediately. Some smaller improvements and new features to support the users in doing their job were included since then. With BIROS starting its operational phase, a new version of SPOT will be launched. The decision was made to integrate both spacecraft in one tool, with both satellites' TLEs available. This way the user can view and list the acquisition opportunities of both TET and BIROS together and decide which spacecraft to choose for an acquisition. Criteria for such a decision could be which spacecraft sees a region of interest earlier, or by which one with a looking angle or lighting conditions fitting better the scientific purpose of the data.

In addition, it will be even possible to add a third, non-Firebird satellite to the selection for the calculations and visualization of orbits and opportunities. Having the possibility to view the ground tracks of other spacecraft as well during a specified timeframe will be used to find e.g. opportunities for adjacent acquisitions for campaigns to compare and/or combine the gathered data of similar or complementary instruments.

B. TimOnWeb – Providing the Timeline Online via the Web

On the other end of the MPS, TimOnWeb is a graphical display tool to visualize the mission timeline and provide insight into an excerpt of the current state of the planning model to the users and other parts of the ground segment, e.g. the Flight Directors, via a website accessibly from the internet.

After each planning run, the latest resulting timeline is broadcasted via this interface. All requested and the actually planned and commanded Planning Requests with their parameters, the groundstation contacts, sun- and shadow phases and the fill levels of the onboard memory partitions are shown, amongst other information. Figure 3 provides a snapshot of the current view for TET.

More detailed information about the implementation of TimOnWeb can be found in Ref. 5.

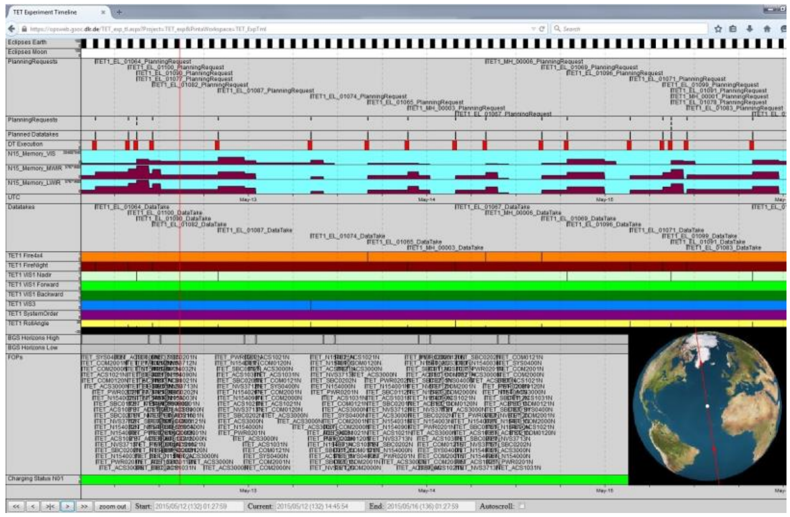

Figure 3. Snapshot of TimOnWeb.
The Web front-end gives insight into the current planning state.

C. Pinta/Plato for TET and Pinta/Plato for BIROS

The core planning components of the Firebird mission are Pinta/Plato for TET and Pinta/Plato for BIROS. They are similarly built and as their naming tells, they comprise GSOC's generic "Program for INteractive Timeline Analysis" Pinta and generic planning library Plato ("PLAnning TOol").

Both operate on a representation of the respective mission's planning problem, the planning model:

1. The Planning Model

A GSOC planning model is also referred to as the Pinta or Plato or simply current project. In Ref. 6, a detailed overview of the modeling language can be found. Briefly described, the model contains groups, tasks, parameters, and the resources and constraints to be considered. Among others, we have OrderedTimeDependencies between the tasks, Bounds of the resources and Allocating-, Accumulating- and ComparingResource Dependencies between the tasks and resources, with all but the inter-task constraints being specified via profiles over time. Scheduling and un-scheduling of the tasks is done by adding/removing timelineentries to/from the project timeline. The resource modifications become active as soon as a task is scheduled, while the constraints of the other tasks already scheduled as well as the resources themselves indicate conflicts, i.e. whether the modification leads to a conflict-free new project state or not.

The so-called StartUpProject with the initial setup of the planning model comprises merely horizon and indicator tasks and tasks for the FOPs to be planned. Furthermore, it already contains basic resources that will have to be used and considered throughout the process or for display purposes, with their initial fill level and potential Upper- and LowerBound profiles.

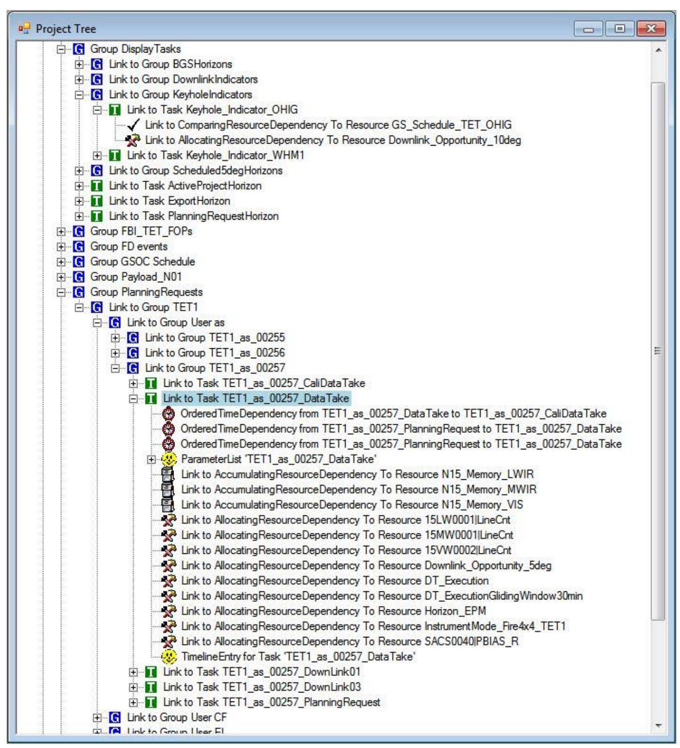

Figure 4. Snapshot of a part of the TET Planning Model.

Then, importers usually add further groups, tasks, resources and constraints, and the operators manually and the scheduling algorithm(s) automatically create, shift and remove timelineentries of the tasks to/on/from the timeline, which can cause, remove or avoid conflicts on the resources and in-between the tasks, depending on the respective objective.

In general, the approach of a mission planning system and its planning runs of course is to transfer an old to a new project state with a conflict-free timeline to be further processed.

The basic planning model for TET as well as for BIROS is composed and displayed in Pinta via the so-called Project Tree. An excerpt of this can be seen in Fig. 4.

2. Pinta and its Plug-ins

Pinta at GSOC is used for all projects and/or phases of projects whenever a (semi-)manual planning and/or (semi-)automatic inspection or crosscheck of an automatically generated planning result is requested. It is a GUI application with the help of which a planning model can be constructed using the GSOC modeling language[6] (see above). The planning model and the current timeline can be visualized and modified e.g. via drag-and-drop of model contents. Figure 5 shows an example snapshot of the Pinta GUI.

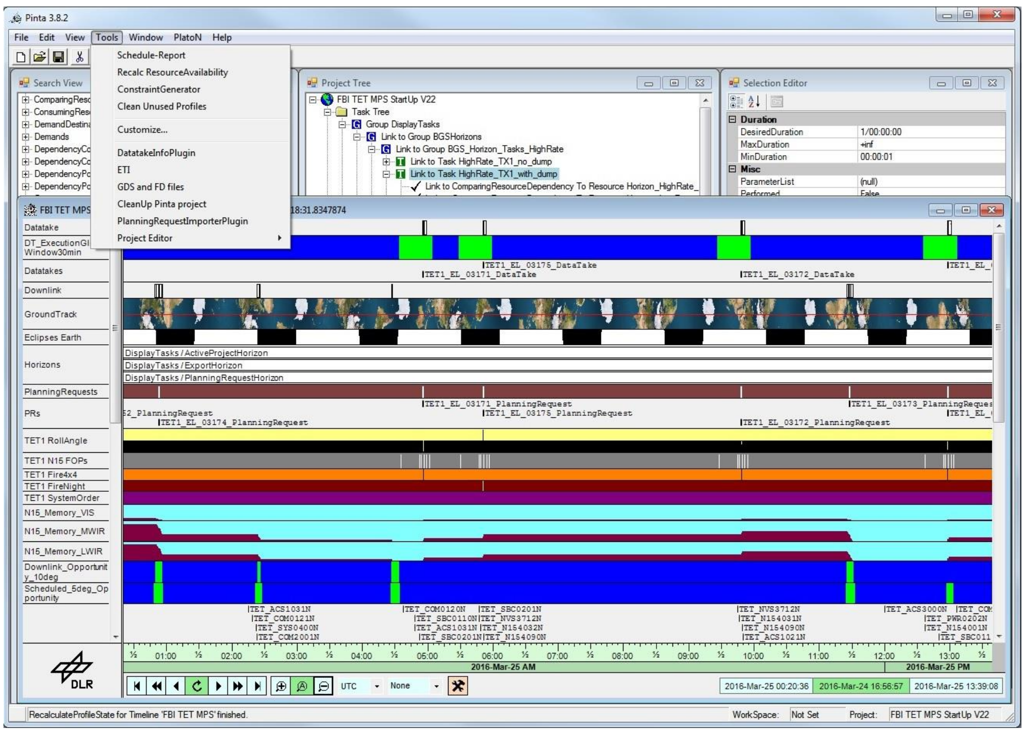

Figure 5. Snapshot of the Pinta GUI with a TET timeline loaded.

Apart from many other features, Pinta contains the ability to include generic as well as project-specific plug-ins and to run Plato algorithms on the currently available planning model to modify the current timeline (see below).

Generic plug-ins and features configured and integrated for TET and BIROS are

- the *CleanUpTool:* Before every planning run the current planning model state has to be cleaned in order to avoid runtime problems due to too much information stored in it and the underlying database. It is configurable for which model contents and up to which point in time the clean-up shall take place. This way it is ensured that only information is autonomously deleted that has no influence on the current planning state and future planning runs anymore. Previous states of the model are archived.
- the *GSOCFileImporter:* This plug-in is invoked to ingest input files from other parts of the mission operations ground segment (see Fig. 1): On a regular basis new TLEs and Event files from Flight Dynamics and so-called Schedule files from the Groundstation Scheduling office are received, and at the beginning of a planning run the latest of each type is processed. The content of the Event and Schedule files is filtered according to the current configuration of the Importer and is inserted into the planning model as scheduled tasks with constraints to modify the fill level profiles of the according resources.
- the *ExecutionTimelineImport:* This plug-in originally was implemented to load a complete result timeline of FOP sequences that has been generated in fully-automated planning runs, to allow for a visual cross-

check for its consistency and validity supported by the stored set of constraints. For the Firebird mission the functionality is reused to ingest input XMLs from the principle camera investigator that contain a FOP snippet to be used for so-called SystemOrders. For the experimental acquisitions taken for these special requests, the camera is not commanded to one of the standard acquisition modes but configured with a FOP containing the ingested snippet.

- and the *ExecutionTimelineExport:* After having created a new timeline via Plato algorithms (see below in II.C.3) and, potentially, manual interaction, the ExecutionTimelineExport is triggered for the upcoming export horizon. As configured, it fetches all planned FOPs in the timeline during this timeframe and combines them to an XML containing the sequence of FOPs with the according start times of the timelineentries. The list of commands inside is filled with all necessary information from the underlying FOP database, parametrized with the information from the model. For example, some attitude control commands are given a roll angle value that is read by the Export from the according resource's fill level at the start time of the according task's timelineentry.

Furthermore, for Firebird three new plug-ins have been developed and are integrated via Pinta:

- the *PlanningRequestImporter:* The Planning Requests the users have created with SPOT and transferred to GSOC are ingested by this plug-in. This includes re-calculating the actual start and stop time and roll angle according to the newest available TLE before upcoming planning runs for all Planning Requests in the current planning horizon. Therefore, the SCOTA library with its event calculation is included in the PlanningRequestImporter as it is included in SPOT (see II.A). For each of the Planning Requests a group with datatake, calibration datatake and downlink tasks is added to the planning model, with constraints added that map all their resource dependencies and scheduling restrictions. For instance, the consumption and release of memory space are implemented as so-called AccumulatingResourceDependencies whereas the prohibition to execute datatakes in parallel to downlink transmissions is included via a so-called ComparingResourceDependency.
- the *DatatakeInfoFileGenerator:* This plug-in creates the so-called DatatakeInfo XMLs that contain all relevant information from the model for each finally planned Planning Request to be transferred to the data processing facility. It is notified about which payload data will be received during the upcoming groundstation contacts and what are the finally commanded start and stop time and roll angle of the acquisition. Thus, the facility can combine this attitude information and the correct additional telemetry data from the spacecraft with the payload data when processing the acquisition and generating higher-level imagery from the downlinked raw data.
- and, for BIROS only, the *EnvelopeRequestImporter:* This functionality is needed to ingest so-called Envelope Requests (see Fig. 1). They will be sent in an XML-format to MPS to announce time intervals for special onboard activities (maneuvers, experiment executions or other maintenance phases) during which the nominal mission shall be interrupted and/or restrictions on the BGS are imposed (see IV.B.2).

3. *The Plato Algorithm(s)*

As already mentioned, Pinta allows for invoking Plato algorithms. Pinta/Plato for TET and Pinta/Plato for BIROS make use of this and comprise scheduling algorithms to automatically schedule and re-schedule indicator tasks, datatakes and downlinks to prepare a conflict-free timeline for the upcoming planning horizon. Furthermore, timelineentries of the FOP tasks are planned accordingly for the commanding timeframe from the next to the next but one uplink session.

These scheduling algorithms however have not been implemented from-scratch but composed via the generic algorithm assembly available in our Plato library that is the basis of any Mission Planning system set-up at GSOC. The core idea of this set of algorithms, filters, etc. is to have a reusable, configurable suite of functionalities that can be variously combined and configured and operate on a planning project available in the GSOC modeling language.

Besides the logic implemented and compiled in the code of Plato each of these algorithms and many of the filters are invoked or invoke each other with a parameter set represented via an object derived from an XSD schema. Thus creating a Plato algorithm for a certain use case means creating a sequence and/or cascades of XMLs that conform to these schemas referencing each other starting from one algorithm root.

The suite of available algorithms, filters and their configuration possibilities have been extended and continuously further developed during the implementation and support of various Mission Planning projects. In general, in the meantime this framework has become similar to a new domain-specific language: One function can be invoked from another with long lists of variables, features exist like global and/or temporarily available variables, transactions, case decisions, loops, etc. All of these abstractions act on the high level of planning algorithms working on and modifying a given Plato project with its structure of objects which themselves are combined and dependent on each other via a set of constraints.

The big advantage of our approach for creating planning algorithm is, on the one hand, that without the necessity for recompilation of the whole planning tool or even parts of it, it is possible to modify the complete behavior of the scheduling algorithm(s) and thus produce another planning result without big administrative overhead. Of course, this is not allowed to lead to make use of the given possibilities with too much levity because the modeling and algorithm suite are too mighty and more complex than a simple/customary configuration change of a software component. However, and these are the other big advantages, the underlying basic algorithms are thoroughly tested and are mostly already validated in other missions or for other use cases. Moreover, they are written in a robust style to detect and signal invalid input as far as possible and in addition to the already good, useful consistency checks the usage of XMLs –which have to conform to their underlying XSD schema– already provides.

A more detailed insight into this approach and an overview of most of the available algorithms and filters can be found in Ref. 7.

Regarding the TET and/or BIROS planning problem, here just some examples for basic algorithms shall be given and outlined only very shortly to understand the principle:

- *ChooseValuesToConsider:* The time range to be forwarded to a sub-algorithm can be determined via various criteria, e.g. respective to the execution time of scheduled horizon tasks.
- *ObjectSelection:* Various filters can be applied to determine to which sub-algorithms which of the currently considered group(s) and/or task(s) are to be forwarded in which order.
- *ValueSelection:* When having found the next task to schedule, various filters can be applied to determine whether and with which execution time it is allowed to be scheduled. The invocation of sub-algorithms with time ranges derived from the new timelineentry's execution time is possible.
- *ConstraintIgnorer:* This allows to temporarily deactivate constraints during the execution of a sub-algorithm, which can be necessary to handle circular dependencies. It enables scheduling and/or un-scheduling tasks with a potential conflict first before trying to repair the solution by scheduling and/or un-scheduling other tasks.

A composition of these algorithms is used for TET and BIROS to schedule the datatake, its calibration datatake and their downlink tasks for a Planning Request. Due to the memory behavior of the spacecraft (first-in first-out principle) this can include un-scheduling of downlink timelineentries of other datatakes that have already been planned but have a later acquisition time than the new one. Then the downlinks of the new datatake have to be inserted before the ones of the first succeeding datatake, i.e. all upcoming downlink timelineentries are to be removed and re-added.

Other examples for the application of the Plato algorithms and their usage will be mentioned in chapter IV.A, in which the evolution of the BIROS core planning system from the one for TET is discussed.

4. *Manual Operations when Performing the Planning Runs*

As already mentioned, Pinta/Plato for TET and Pinta/Plato for BIROS are set up as semi-automated and thus interactive Mission Planning tools. The daily tasks left to the operator are the following:

- performing and monitoring the planning runs as specified in the according ground operations procedure
- execution of recommendations for off-nominal operations and/or additional procedures for special planning tasks for additional payloads, if applicable
- user help-desk for inquiries of the users, e.g. about the reason why a request could not be planned

In summary, there are several manual steps to be performed in addition to the capabilities of the software components. In a more complex Mission Planning system than the one for Firebird, most of these are included into a fully automated software component or component assembly. However, for this mission it was decided that the described approach fits best the given requirements for reliability combined with flexibility, and, not to forget, cost efficiency.

III. The Evolution of the TET Mission Planning System and Pinta/Plato for TET

A. Mission Planning for TET before the Firebird Mission

During the OnOrbitVerification (OOV) mission, the TET spacecraft was carrier for eleven newly invented technological hardware components to be proven in space, with one of them being the infrared-optical camera system. Mission Planning at GSOC originally only comprised the import of the Event and Schedule files and input from an Excel file from the customer containing the execution times and steps for the complete experiments pre-calculated for the whole year of operations in advance. The main goal of the Mission Planning software then was to translate this to a sequence of FOPs combined with the BGS operations and generate a weekly ExecutionTimeline XML. A more detailed description of this system and the first steps to enable repairing the preplanned timeline according to actual circumstances and gaining more flexibility can be found in Ref. 2.

B. Realizing the Firebird Mission on TET

With the Firebird mission becoming "owner" of TET, the objective of and therefore the requirements to the operations changed, whereas the spacecraft with its capabilities first stayed the same.

The basic drivers for a reorganization of the Mission Planning components were to

- have the focus on the operations of the infrared-optical camera system,
- make the data it can deliver accessible to a bigger user community for scientific and Earth-observation applications,
- enable target-oriented acquisitions of flexible duration and allow several different, pre-configured but spontaneously selectable camera configurations,
- make the generation of the data reliable, incl. avoiding the loss of already acquired data or non-accessibility of expected data because of not available memory, and
- increase the number and amount of acquired data by avoiding the necessity for huge security margins.

Therefore, on the one hand, the SPOT tool and a notification to the groundstation and processing unit about exactly which data can be expected were established.

On the other hand, the PlanningRequestImporter as described in chapter II.C.1 was developed and the planning model was set up with the Planning Requests, datatakes and downlinks together with related resources and constraints as new entities. The memory fill level had to be modeled, and after the correct data rates had been found out, the fill level modifications were linked to the datatakes and downlinks.

It had to be considered that the datatakes in the different modes fill the three memory partitions for data from the MWIR, LWIR and VIS channels with a different data rate depending on to which one or two other partitions the data is written in parallel during which part of the acquisition. The same applies for modeling the release via accordingly different slopes for the constraint profiles of the downlink tasks. For example for the "Fire4x4" mode, there is a timeframe for which only the LWIR partition is filled/released, then all three partitions are filled/released in parallel and then only the VIS partition is filled/released for some more time.

Last but not least, the OOV scheduling algorithm that in its latest version generated the BGS and concentrated on "repairing" and shifting[2] the input from the pre-planning Excel sheet, had to be split and newly composed for the scheduling of the datatakes and downlinks (see II.C.3).

C. Examples for Continuous Adaptations of the TET Mission Planning System

With onboard software updates and further change requests of the mission, adaptations of the planning model and the Export functionality had to be continuously made. Among these were introducing the SystemOrders (see ExecutionTimelineImport in II.C.2) and shortening of the order deadline, but two other examples should be given here to show how the MPS has to be kept up-to-date with the spacecraft behavior and on-ground decisions.

1. Changing Constraints and Resource Availabilities

When the experience with numerous acquisitions under different spacecraft conditions had been achieved and the power-thermal spacecraft behavior could be evaluated, it was, for example, discovered that some of the restrictions for scheduling datatakes could be relaxed, and that a new way of controlling the solar-panels was advisable. For the latter, only the rule for scheduling an additional FOP in advance to every acquisition had to be added to the XML scheduling algorithm sequence via copy and paste, and the model, import and export plug-ins could stay as-is. Conversely, the restrictions for using the camera system are mapped to constraints of the datatake tasks to instrument-is-active indicator resources. For their relaxation then only the adding of these constraints upon creation of the datatake tasks for new acquisitions had to be switched off in the PlanningRequestImporter. The scheduling algorithms did not have to be touched.

With this possibility to task the satellite with a higher workload, the amount of onboard memory became the new bottleneck for the acquisition of datatakes, and new groundstations had to be added to the schedule to facilitate a higher data throughput. For the core scheduling algorithm this could be implemented transparently. Only the filters of the FileImporter had to be re-configured to ingest resources and tasks for elevation events and Schedule information for these new groundstations. In addition, a preparatory step in the scheduling algorithm to create timelineentries for tasks that indicate the combined availability of elevation event and availability of the station for the contact had to be copied.

2. Consideration of Groundstation Antenna Keyholes

Another change request for the TET operations regarded the inclusion of so-called "keyholes" into the downlink planning. A keyhole occurs when a groundstation antenna, due to its construction, cannot track the spacecraft at high elevations. This means that the transmission has to be interrupted for a short time interval to avoid losing data. For the stations we operate for Firebird this applies for one of the antennas in Weilheim (Germany) and the antenna in O'Higgins (Antarctica).

Originally, only groundstation contacts over Weilheim and Neustrelitz (also located in Germany) were in the Schedule, and a manual selection of the station contacts took place outside of MPS that avoided such a high-elevation pass. However, then the decision was made to automatically use certain contacts per day, and the planning system had to cope with the necessity to detect and handle keyhole passes correctly, including pausing the transmission of payload data during the critical time interval.

This could be solved in the following way: The Plato algorithm sequence was extended by scheduling so-called KeyholeIndicators wherever the threshold for the maximum value of the according groundstation elevation resource is reached, with according offsets. As soon as these Indicators are scheduled, they in turn reduce the value of the full-downlink-available resource so that the novelty was immediately transparent to the existing planning algorithm for scheduling datatakes and downlinks. No further change became necessary because for this part the gap was considered like any gap between adjacent groundstation contacts, whereas for the rest of the planning the groundstation contact could still be considered as a whole, since relying on other resources' fill levels. For instance, the planning of the FOPs for the BGS was not affected and automatically no superfluous switching off and on again of the downlink transmitter around this event was initiated.

Later on, when downlink contacts over O'Higgins were added to the Schedule to extend the overall available downlink time, scheduling the KeyholeIndicator tasks only had to be extended towards considering the according elevation events of this station as well the same way.

IV. The Evolution of Pinta/Plato for BIROS

A. Developing Pinta/Plato for BIROS from the Experience made for TET and other Projects

With having all the experience made for TET and having the already existing MPS for this similar (but by far not equal!) spacecraft, of course the initial idea was to duplicate and reconfigure the existing tools where possible and viable which then formed the basis for setting up the BIROS Mission Planning system. However, the awareness has to be given to duplicate "correctly" and to incorporate improvements that had been made on the generic tool and algorithm basis and experience gathered in other missions. These should not be neglected in order to simplify the new planning system as much as possible and to avoid dragging along "dead weight". Such ballast can have been created via workaround solutions for a certain problem to be handled on-the-fly (and left as soon as the problem is solved and other work has to be done). At least when having a second look at a software solution now, these should be overcome at last and everything should be re-assessed for a cleaner and less complex implementation.

To enhance the usability of the planning tool, it has to be avoided that the tool for BIROS contains remainders from the one for TET that could lead to confusion when not expecting them and thus not knowing their influences on the planning and export functionalities. Therefore, having a closer look at every single entity upon set-up of the new system is worth the time to save it later on during debugging.

In addition, in the two to four years between converting the OOV to the Firebird/TET MPS resp. starting the first TET MPS at all, other Mission Planning systems have been developed by our Mission Planning team at GSOC (e.g. the Link Management System[8] for EDRS and the MPS for TDP-1[9]). As side-products, new generic Plato algorithms were developed and some of the old ones were enhanced or expanded. The so-called backward-compatibility is always ensured, i.e. previous planning algorithm configurations are still supported by newer versions of the Plato library and especially newer versions of the dedicated algorithm snippets themselves, but due to this evolution there might exist better/simpler alternative solutions to face some of the Firebird requirements in the meantime.

In detail, in Ref. 2, the StaticSchedulingRules algorithm, the FilterAlgorithm, and their usage for TET were described. Some years ago, these were the first algorithms with the philosophy of being configured via XMLs while keeping the compiled software stable, i.e. almost the starting point of bringing a big amount of rather online flexibility to already installed planning software components. By now, they provide a huge scope of functionality that has been created for the TerraSAR-X/TanDEM-X MPS[10,11] and constantly extended in these times. With the extension of the algorithm suite and the urge to freely combine rather small snippets to bigger algorithms (see chap. II.C.3), also new algorithms were developed that contain parts of the functionalities of the two "big old ones" and in combination replace and even expand their capabilities. For example, the StaticSchedulingRules (among other features) allow creating timelineentries of selectable tasks including the possibility to avoid conflicts etc. However, it is not possible to consider combined opportunities of multiple resources and to loop over opportunity occasions easily, so this had to be worked around by adding additional constraints and specific resource modification profiles.

With one of the new algorithms, the so-called OpportunitySelection, time intervals can be found for example unifying or intersecting different opportunities of multiple resources (including offsets, minimum durations, etc.). The resulting set of time intervals is then freely processible in a sub-algorithm to be specified, all at once or one after the other. In our use case, it is handed over to a series of other algorithms that first filters for the objects to be planned and then, according to the specific object filters, for the correct time to be planned according to further restrictions.

B. The Extension Stages of the Mission Planning System for BIROS

As far as foreseen until now, for BIROS at least four major extension stages for the core planning system can be distinguished. For each of these most probably one or more new configurations of the planning model and planning algorithm are needed. This leads to several consecutive versions of the Pinta/Plato for BIROS application.

1. LEOP Support

During the Low Earth and Early Orbit Phase (LEOP) of BIROS the semi-automated planning system shall cover only a very reduced scope of recurring operations. Apart from supporting the mission by maintaining the so-called Sequence of Events manually with our new generic SoEEditor tool (which is also based on Pinta, see Ref. 5), a first version of Pinta/Plato for BIROS will be used to generate a new BGS whenever new orbit or groundstation information is received. Downlink sessions are supported that are formed by one or more overlapping or shortly adjacent groundstation contacts with configurable offsets. Furthermore, KeyholeIndicator tasks are scheduled for the according events in order to enable a notification of the Flight Director about these.

2. Commissioning and Validation Phase Support

After the LEOP, an extended version of Pinta/Plato for BIROS will have to be activated for daily planning runs. With starting more complex spacecraft operations and the check-out of all the instruments, dedicated different dump modes will have to be used, and attitude mode switches will be included in the automatic planning and commanding as soon as they have proven to work as expected in space.

In general, implementing these additional requirements means multiplying the features already available in the LEOP support version and integrating functionalities as already used for TET to schedule so-called TransmitterHorizonIndicator tasks that modify according indicator resources, which then are used to decide upon which kind of FOP is to be scheduled and thus exported.

In addition, some of the experiments carried by BIROS are also likely to be executed already during the validation phase. Technical details can be found in Ref. 4. They are operated outside the MPS, however we have to support that by commanding special BGSs or only scheduling a reduced amount of FOPs in order not to disturb their execution. Therefore, the special EnvelopeRequestImporter plug-in (see II.C.2) will be used to ingest the so-called Envelopes into the planning model.

Another purpose of the Envelope Requests is to announce timeframes during which operations of the main payload shall be blocked, due to either experiment or maneuver execution or other foreseeable outages. This especially becomes relevant in the next phase:

3. Nominal Payload Operations

As soon as all necessary instruments and components of the spacecraft are successfully tested and the foreseen FOP sequences to command them are proven in space and/or adapted to the evaluated circumstances, the nominal payload operations can start. For this phase, the planning system normally should already have been developed and undergoing integration and system tests as well. However, the spacecraft onboard software of the BIROS Payload Processing Unit is still partly under discussion and thus not yet fully prepared by the spacecraft manufacturer. Nevertheless, it is clear that the preparation has to be started early enough in advance to the mission relying on our support, so we will have to base it on assumptions to some extent and foresee possibilities for last-minute changes and functionality add-ons.

Some changes for the BIROS planning system compared to the TET planning system which are already known are that the separated storage of camera payload data in the three partitions linked to the channels will be obsolete; however, other partitions will have to be handled then. For instance, there will be acquisition modes triggering onboard data analysis, for the use case of detecting and monitoring so-called High-Temperature Events. The resulting data of this mechanism is then stored in a dedicated memory partition the downlink of which has to be commanded explicitly. Next, acquisitions in parallel to downlink contacts probably are allowed, but will be dependent on the memory configuration. Furthermore, a feature to enable double downlinks with a dedicated near-real-time (NRT) downlink as soon as possible after the acquisition of the data of certain Planning Requests shall be implemented in addition to the general first-in first-out downlink approach that will persist for BIROS.

Therefore, the planning model for TET can be re-used to some extent, but has to be adapted accordingly, now and for all the upcoming onboard software design decisions. The same applies for the PlanningRequestImporter, ExecutionTimelineExport and the scheduling algorithm to cope with the new and/or changed requirements. Here again, the modular set-up and flexibility to incorporate modifications and make use of re-configuration options are very beneficial.

4. Inclusion of Experiments and Optional Operations Enhancements

One of the software experiments of BIROS will be VAMOS, developed within our Mission Planning team at GSOC for “Verification of Autonomous MissionPlanning Onboard a Spacecraft”. More detailed information about this can be found in several publications, for example in Ref. 12. The core Mission Planning system on-ground will have to include according features to prepare the onboard operations and enable the autonomous decisions without risking exceeding limitations or loose/overwrite data of the primary payload on the one hand and without losing the capability of commanding operations from ground in parallel on the other hand at the same time.

As well as VAMOS, some of the other experiments carried on BIROS, which are executed via MPS-external commanding in the earlier phases, might, if proven successfully, be selected for integration into the nominal operations[4]. If this is the case, then the planning algorithm would become responsible for scheduling regular activities of these additional instruments and/or software components and combine them with the main payload and BGS operations.

The complexity for these adaptations will vary. Including, for example, the next-generation High-Torque Wheels into nominal command timelines might only mean exchanging the customary with new attitude control FOPs and reducing the offsets to be kept in-between attitude changes and data acquisition. Integrating the OSIRIS laser communication terminal for optical downlink however would probably impose some more modification effort, e.g. reorganizing the downlinks completely instead of only modifying the data transmission rates. Nevertheless, the flexible set-up of Pinta/Plato for BIROS will still allow for integrating/implementing such changes with acceptable risks and time- as well as cost-effort.

V. One Mission Planning System or two?

Since we have two similar spacecraft within one mission, each of which provide similar main mission goals, the question obviously arose whether to choose the set-up as described in Chapter III.C, i.e. doubling the planning tool, or to prepare one combined Mission Planning system for scheduling the operations of both TET and BIROS together.

For the interfaces to the users, SPOT and TimOnWeb, BIROS will be integrated into the TET components to provide a combined, user-friendly and comfortable solution. However, for the core planning system, the decision was the latter one, which shall be reasoned a bit in the following:

On the one hand, the main payload, the infrared-optical camera system, and the spacecraft bus are the same for both spacecraft. Thus, there are commonalities among the to-be-commanded tasks. On the other hand, the Payload Processing Unit is different, the memory management, the commanding workflows, and the additional functionalities to be operated are different, there will be different FOPs available to command the bus, and, having a closer look, even the configuration of the camera acquisitions will be slightly different. Still, this would not prohibit merging the planning systems to one: The models of the spacecraft behavior including resources, constraints and tasks to be planned could be kept in parallel inside one planning model as well as the separate ground resources. The import and export functionalities are able to filter and identify the input and output belonging to each spacecraft, the planning algorithm could be combined, etc.

For a formation-flying mission such as the TerraSAR-X/TanDEM-X mission the need to integrate the planning for both spacecraft into one system[11] results from the requirements to balance the workload, perform combined acquisitions, not disturbing each other when doing separate acquisitions, and competing for on-ground resources for the downlink of payload data. None of the latter three points applies for TET and BIROS that will form a constellation separated far enough to have different contact times for the same groundstations. In contrast, the first point might be applicable, and, precisely because of having the distinct orbits and visibilities not only for the groundstations but also the acquisition targets of interest, another idea could become interesting: to let the planning system choose the spacecraft to perform a request by having the knowledge about which one is able to perform the acquisition and downlink of the data earlier, according to all contributing factors.

However, with a proper near-real-time support existing only for BIROS (see chapter IV.B.3) and the assumption that the acquisition opportunities will differ in time too much anyway, it is currently considered sufficient for such use cases to provide the user with the combined preview in SPOT and let him decide on the spacecraft upon creation of each Planning Request (see II.A), knowing of course that he cannot completely foresee the planning success then. Furthermore, although the camera hardware is the same on both spacecraft, the onboard interfaces and memory capabilities are improved for BIROS, so that e.g. the Field of View of the visual channels can be increased. Thus, the software configuration will differ slightly and so will the acquisition modes and/or the resulting properties of acquired data. In addition, a higher variety of acquisition modes for BIROS is in discussion. Therefore, for most acquisitions the user himself will want to keep the sovereignty to decide on the satellite to task with a certain request anyway.

So, for the time being, it was eventually decided by the project to keep the core planning systems separate in favor of having distinct order deadlines as well. The operators will trigger the timeline and command generation separately at the respective planning run start times in advance to the uplink session of the respective spacecraft. They will use the same GUI, Pinta, but another content of the planning model and configuration and functionality composition of the Importers, Exporters, and Plato scheduling algorithm.

VI. Conclusion

For the Firebird mission a semi-automated Mission Planning system has been developed, based on a set of generic tools and libraries. Starting from the support of the TET OOV mission via the renewal for TET becoming part of the Firebird mission up to adding the operations of BIROS as a second spacecraft, several modifications and requirement changes had to and will have to be incorporated. Further extension stages are expected to be required in future mission phases. It is already known that several updates and/or reconfigurations of the BIROS onboard software and commanding structure will be implemented throughout the mission lifetime, as it already applied for TET, and that the role of some of the additional experimental payloads might change. To cope with such a continuous evolution of the onboard capabilities and on-ground requirements, the Mission Planning system must be flexible and upgradeable, sometimes even on short notice, while always ensuring a reliable operational availability.

To support this, we reuse an assembly of generic functionalities configured to the project-specific needs and complemented by project-specific extensions. The specific planning problem is mapped to our GSOC modeling language and planning algorithms are composed from the extensive suite of relatively freely combinable, configurable Plato algorithms. Both are then embedded in our interactive planning tool Pinta with its generic configurable plug-ins and the possibility to include further project-specific plug-ins for additional features. Only the latter and the Swath Preview and Ordering Tool were newly developed for Firebird. However, these components again were set up in a way that other missions with the need for similar customer interfaces could reuse them with only some minor adaptations.

The core approach of the Mission Planning software development at GSOC is that if the already given capabilities are not sufficient, and as soon as it is predictable that the respective features might become useful for other tasks and/or projects in the future, an attempt is made to extend or enhance the overall tool and/or algorithm suite by new functionalities rather than implementing mission-specific solutions. This leads to having available a well-proven and validated, configurable and modular framework. Therefore, on the one hand, solutions for new missions to be operated can be generated with a reduced implementation effort in comparison to setting up everything from scratch, especially in terms of testing the basic functionalities. On the other hand, many modifications can be performed as reconfigurations while keeping a robust and reliable system at the same time.

As this concept has already proven very useful, especially for the first part of the Firebird mission on TET, we assume we will be well-prepared for the upcoming challenges during the future project phases as well.

References

[1]Axmann, R., Mühlbauer, P., Spörl, A., Turk, T., Föckersberger, S., and Schmolke, J., "Operations Concept for 11 different Payloads on the TET-1 Mini-satellite", *The 2010 4S Symposium - Small Satellite Systems and Services*, Funchal, Madeira, Portugal, May 31-June 4, 2010.

[2]Spörl, A., Lenzen, C., Wörle, M.T., Hartung, J., Mrowka, F., Braun, A., and Wickler, M., "Mission Planning System for the TET-1 OnOrbitVerification Mission", *SpaceOps 2014, 13th International Conference on Space Operations*, Pasadena, CA, May 5-9, 2014.

[3]Reile, H., Lorenz, E., and Terzibaschian, T., "The Firebird Mission - A Scientific Mission for Earth Observation and Hot Spot Detection", in: "Small Satellites for Earth Observation", Digest of the 9th International Symposium of the International Academy of Astronautics, Wissenschaft und Technik Verlag, ISBN 978-3-89685-574-9.

[4]Lesch, T., Mühlbauer, P., Lenzen, C., and Wörle, M.T., "Firebird: A Spacecraft Constellation for Monitoring High-Temperature Events on Earth Including Promising Innovations", *SpaceOps 2016, 14th International Conference on Space Operations*, Daejeon, South Korea, May 16-20, 2016.

[5]Hartung, J., Nibler, R., Spörl, A., Lenzen, C., Wörle, M.T., and Peat, C., "GSOC SoE-Editor 2.0 – A Generic Sequence of Events Tool", *SpaceOps 2016, 14th International Conference on Space Operations*, Daejeon, South Korea, May 16-20, 2016.

[6]"Planning Modelling Language", URL: http://www.dlr.de/rb/en/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-MIB/GSOC_Modelling_Language.pdf [cited 17 March 2016].

[7]Lenzen, C., Wörle, M.T., Mrowka, F., Spörl, A., and Klaehn, R., "The Algorithm Assembly Set of Plato", *SpaceOps 2012, 12th International Conference on Space Operations*, Stockholm, Sweden, June 11-15, 2012.

[8]Göttfert, T., Grishechkin, B., Wörle, M.T., and Lenzen, C., "The Link Management System for the European Data Relay Satellite Program", *SpaceOps 2016, 14th International Conference on Space Operations*, Daejeon, South Korea, May 16-20, 2016.

[9]Rossmanith, G., Kuhlmann, S., Grishechkin, B., Schlepp, B., Pitann, J., Lutzer, M., and Tröndle, D., "Laser Communication in Space: The TDP-1 Mission Control Center and its current Operational Experience", *SpaceOps 2016, 14th International Conference on Space Operations*, Daejeon, South Korea, May 16-20, 2016.

[10]Lenzen, C., Wörle, M.T., Mrowka, F., Geyer, M. P., and Klaehn, R., "Automated Scheduling for TerraSAR-X/TanDEM-X", *IWPSS 2011, International Workshop for Planning and Scheduling in Space*, Darmstadt, Germany, June 08-10, 2011.

[11]Geyer, M., Mrowka, F., and Lenzen, C., "TerraSAR-X/TanDEM-X Mission Planning - Handling Satellites in Close Formation", *Space Ops 2010, 11th International Conference on Space Operations*, Huntsville, AL, April 25-30, 2010.

[12]Lenzen, C., Wörle, M.T., Göttfert, T., Mrowka, F., and Wickler, M., "Onboard Planning and Scheduling Autonomy within the Scope of the Firebird Mission", *SpaceOps 2014, 13th International Conference on Space Operations*, Pasadena, CA, May 5-9, 2014.