The VEGA Planning Toolkit: a Flexible Tool for Planning Systems Development

Marc Niezette[1]
VEGA Space GmbH, Darmstadt, 64293, Germany

This paper introduces the VEGA Planning Toolkit (VPT), a flexible tool that supports the development of automated planning systems. VPT is the successor of the Enhanced Library for Operational Planning Systems (EKLOPS), which has been successfully used to support the daily operations of Envisat, Mars Express, Venus Express, and ESTRACK, the ESA ground station network. The paper presents the characteristics of the tool and its origin and provides an example of deployment in the context of Earth Observation mission planning.

I. Introduction

Over the last few years, the interest for off-the-shelves ground segment solutions able to support a large range of EO missions has grown significantly, mostly driven by the expansion of the market for EO satellites, and enabled by the development of standards.

There is therefore a need for flexible components that can be integrated smoothly in ground segment solution, and support standards when available.

With this in mind, VEGA has developed a core control segment solution, which can be used as a pre-integrated set of components. The core control segment is organized around the ESA SCOS-2000 Mission Control System, complemented by a number of VEGA tools that covers planning and operation preparations.

The VEGA Planning Toolkit (VPT), which is now the basis of the VEGA solutions for planning systems development, implements the planning component in the VEGA core control segment. The toolkit is organized around the adaptability and configurability of a number of elements: (a) types of input; (b) outputs to different targets of different expressiveness; (c) planning logic and work flow; (d) planning algorithms; (e) user interface.

II. Origin of the Tool

VEGA Space has developed for many years numerous planning systems for ESA, including flight operations segment mission planning systems for ENVISAT, Mars-Express, Venus-Express, Gaia, etc., ground segment planning systems for the ESA station network (ESTRACK planning System), test planning systems, etc.

To effectively carry out all these developments, VEGA developed an approach to planning system implementation based on the use of a mission planning kernel, the Enhanced Kernel for Operational Planning Systems (EKLOPS).

EKLOPS originated initially from the ENVISAT FOS Mission Planning System in 2002, and has since then been incrementally extended to address the needs of the successive planning system implementations led by VEGA.

[1] Business Development Manager, Ground Segment Systems, Marc.Niezette@vega.de

In 2009, the tool was considered by VEGA for re-use for the development of several non-ESA mission planning systems. Although the applicability of EKLOPS to the implementation of these systems was not an issue, and the available infrastructure software was directly re-usable, a number of factors have led to a different approach:

1. An essential benefit of the re-use of EKLOPS is the link to the ESA/ESOC ground systems developments. This ensures the operational validation of the kernel, and the long terms maintenance of the software as part of ESA operational systems developments.

   ESOC have in the last years started the development of their own infrastructure software for flight control segment mission planning, the Mission Planning System Framework (MPSF). It is expected that future mission planning system development will all be based on this platform. VEGA Space is currently leading the first development of mission planning system based on MPSF for the Rosetta and BepiColombo missions.

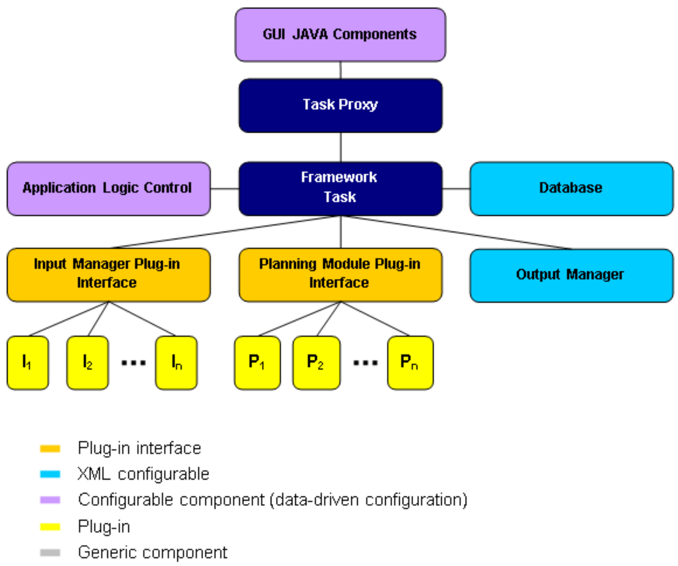

Figure 1. EKLOPS Architecture.

   With the exception of the extensions of existing systems, EKLOPS is therefore unlikely to be significantly used in future ESA ground software developments, and one of its key benefits is now obsolete.

2. In the context of the new developments carried out by VEGA, contractual rules typically enforced by ESA, such as the delivery of the source code to the customer, are not applicable. This relaxes the constraints in terms of tools and COTS that can be used to support the solution.

3. Re-using ESA software for commercial developments introduces constraints in terms of licensing, which are not always easy to address, and may lead to consider alternative solutions when available.

4. The applications of advanced planning techniques as those demonstrated by VEGA Space consortia in studies such as the Advanced Planning and Scheduling Initiative of ESA requires to extend the planning system applications to support more elaborated knowledge and plan representations.

5. The evolution of EKLOPS in the last years, where it has been used to support a wide variety of systems in parallel, has led to the development of several branches of the software. The point was reached when a re-engineering of the tool was needed, through which the strength of each system could be brought in a consolidated infrastructure that would support the future planning systems development. Significant work was needed to bring together the various branches of EKLOPS, which could instead be used to set the basement of a new system.

6. The type of functionality needed by the new planning systems requires a significant evolution of the modeling framework supported by EKLOPS, which would have to be implemented in all cases.

All these arguments have led to develop a new planning toolkit, integrating some elements of EKLOPS with the result of planning studies and the needs identified during the requirement analysis of the future planning systems developments.

III. Requirements Elicitation

VPT is designed as a core planning system consisting of a host application and the toolkit required to build specific solutions. Additional libraries and functionalities are provided around the core to support the fast development of applications in specific domains. The components of the extensions are separated from the component of the core and constitute a separate software element.

The requirements for the core toolkit come essentially from two sources:

- The requirements underlying EKLOPS, which has demonstrated the successful use of rule-based and constraint reasoning techniques for the development of planning systems for numerous space missions. VPT being the successor of EKLOPS, it is expected that all applications previously built using EKLOPS can also be developed within the new framework.

- Lessons learned from ESA studies performed by VEGA with academic partners in the application of advanced planning and scheduling techniques (essentially Artificial Intelligence techniques) to space mission planning. This includes the Advanced Planning and Scheduling Initiative study from ESA.

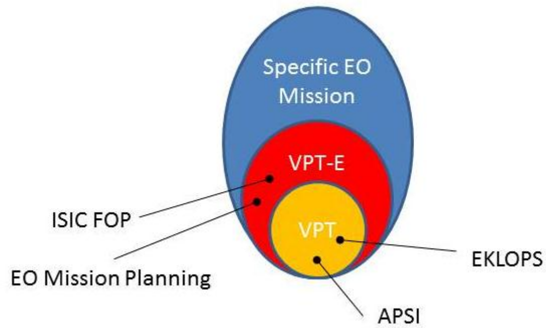

Figure 2. Scope of VPT, VPT-E, and specific planning systems. *Impact of EKLOPS, APSI, and specific system requirements on the VPT infrastructure*

The initial objective of the toolkit is to support Earth Observation mission planning in terms of EO optical payload planning, EO flight operations planning and ground station planning. The extension of the core including this functionality is VPT-E (VPT for Earth Observation), and its requirements are derived from three main sources:

- The requirements from Flight Operations Planning system of the International Space Innovation Centre (ISIC).

  The UK Space Agency has started in 2010 the deployment of a ground segment including both control and data processing facilities at the International Space Innovation Centre (ISIC) at Harwell, Oxfordshire. VEGA Space has contributed to the deployment of the ISIC ground segment, including the Flight Operations Planning system, under the coordination of EADS Astrium.

- The requirements from payload planning of Earth-Observation missions supported internally within the Telespazio group, to which VEGA Space belongs.

  As part of their numerous ground segment developments, Telespazio, VEGA Space's mother company, is developing payload planning systems for EO missions. VPT-E is expected to become the de facto company standard for these activities.

- Available specifications for planning systems for ESA EO mission, such as the Sentinels.

IV. VEGA Planning Toolkit

A. General Architecture

The general architecture of VPT takes advantage of the lessons learned from the EKLOPS developments, and redistributes the functionality of EKLOPS across four main elements, which are described in Table 1.

Table 1. VPT Main Elements.

Module	Description
Plan Engine	The core functionality is encapsulated in a set of C++ libraries that are used by a running server task.
MMI	The Man Machine Interface (MMI) is developed in Java. The communication to the planning core is provided by CORBA, so that the MMI is completely decoupled from the planning core.
Data Repository	The data repository is responsible for holding the input/output data of the MPS core and the working configuration and plans.
Helper Applications	With additional (small) applications direct access to the MPS database is provided, ensuring an independent way for database modifications (import/export) e.g. in case of malfunction of the planning core.

B. Plan Engine

The Plan Engine is the core of the system. It is a layered application in five levels, as illustrated in Figure 3. The various component of the Plan Engine are described in Table 2.

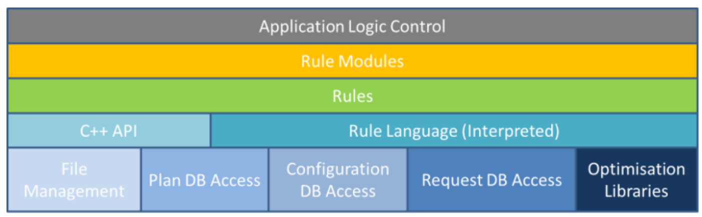

Figure 3. Plan Engine Components.

Table 2. Plan Engine Components.

Module	Description
File Management	Libraries of configurable file handlers, parsers and report generators, which are used to read the planning input files and generate the final output of the planning process, such as executable schedules.
Plan DB Access	Libraries of accessors to the database of plans and their content.
Configuration DB Access	Library of accessors to the planning configuration, including the various models that are required for planning (components behavior, resources, etc.).
Request DB Access	Library of accessors to a request database.
Optimization Libraries	Libraries of algorithms (Linear programming, AI algorithms, etc.) applicable within the planning modeling framework.
C++ API	An API to all lower-levels components of the Plan Engine, which can be used to build specific rules in C++ when needed.
Rule Language	Interpreted rule language supported by the Plan Engine. Typically all rules are expressed in that language, which is accessible to the planning operators.
Rules	Condition-action rules that implement the planning algorithms, and are either written in the Rule Language, or in C++ using the provided API.
Rule Modules	Logical grouping of rules managed by the Application Logic Control.
Application Logic Control	Controls the way the rule modules are run, based on a set of properties such as priorities, dependencies, mutual exclusions, etc. The application control identifies the rule modules that have to be (re-)run when specific new inputs are given to the Plan Engine.

C. Data Repository

The Data Repositories cover the persistent repositories for the data handled by the system, which summarized in Table 3.

Table 3. Persistent Planning Data.

Module	Content Description
Configuration	Rules and Rule Modules Environmental models (visibility, eclipse, etc.) Component models (system elements state machines, etc.) Resource models (resource profiles, resource behaviors, etc.) Configuration parameters Service definitions (request templates) Input file descriptors (for import of input data) Output file templates (for output generation) Copy of applicable satellite database and operations procedures database for the commands, command sequences and procedures definitions
Plans	Plans generated by the system, including events, activities, resources, and constraints.
Requests	Planning requests (reference to a service with associated geographic constraints, temporal constraints, timeliness constraints, etc.)
Dynamic Inputs	All data inputs received from external sources
Outputs	All the data outputs generated from the repositories above

Table 4. Persistent Data Areas.

Repository Area	Description
Input Area	Area where all the external data made available to the system are received and validated.
Holding Area	Area where all the valid external input is stored before being made available to the planning system by the operator.
Master Area	Area where all valid inputs made available to the planning system and all valid plans selected for execution are stored.
Working Area	Area where temporary data are generated (plan under construction, configuration changes before commit).
Delivery Area	Area from where all outputs delivered to the external world are disseminated.

The repositories are implemented as a set of ASCII file repositories and databases. All non-ASCII data can be exported and imported to XML format.

The data repository is organized around two dimensions: the data type (request, plan, etc.) and the planning processing areas, which are described in Table 4. Table 5 provides the distribution of the data types across the repository areas during the overall planning process.

Table 5. Data Repository Organization.

	Input	Holding	Master	Working	Output
Configuration			●	●	
Requests	●	●	●	●	
Plans			●	●	
Dynamic Inputs	●	●	●		
Outputs				●	●

D. MMI

The EKLOPS experience has demonstrated that the MMI of the planning system was highly depending on the specific needs of the users, and was therefore subject to specific implementation for specific systems, while the core of the planning application would stay the same. It is for instance usual to make use in the MMI of the concepts that are known and mastered by the operational teams. The evolution of the tools available for MMI developments also calls for a regular update of the MMI to stay in line with the recent technological developments.

The VPT MMI is therefore kept separated from the core plan engine, with which it communicates via CORBA. This has the advantage that the MMI can be changed according to specific needs while keeping the backend (the plan engine) unchanged.

The frontend is developed in Java, such that it can make use of elaborate libraries for graphical interfaces (e.g. SWT).

The MMI cannot directly change the data repositories or the MPS configuration database. It is used to view output files that are generated by the plan engine, visualize the plan (e.g. using Gantt charts) or command the Plan Engine that in running in background.

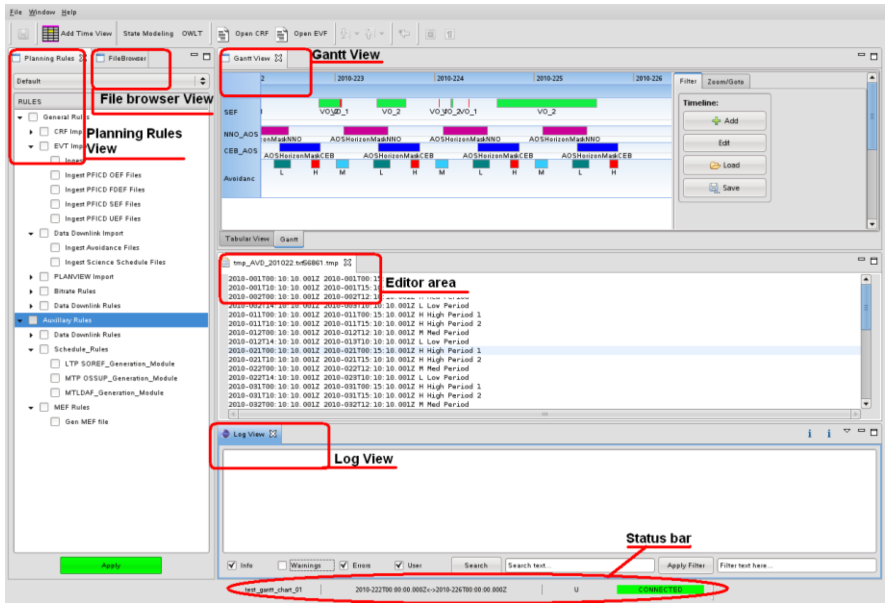

Figure 4. Example MMI.

The key elements of the MMI are described in Table 6. An example implementation is provided in Figure 4.

Table 6. Persistent Planning Data.

Repository Area	Description
Plan Management	Provides access to the plan management functionality, such as creation for a given time range, deletion, etc.
File Management	Visualize the list of files available for import, and provide access to a viewer.
Resource Status Edit	Visualization and edit of the resource profiles.
Gantt View	Graphical representation of the plan as a Gantt Chart.
Rule Control	Access to the rule application control where the user can control the work flow of the planning system.
Request Selection	Selection of requests for the planning session, and change of requests parameters (priority, etc.).

V. VPT for Earth Observation

The core toolkit presented above is basically an open development toolkit that allows the planning system developers to implement specific planning solutions within a modeling framework, using the domain representation and the algorithms provided by the toolkit.

The extension of the core for Earth Observation missions provides on top of this an application logic in several steps, which can be implemented by selecting and configuring building blocks. The typical planning workflow is depicted in Figure 5.

Figure 5. VPT-E Work Flow.

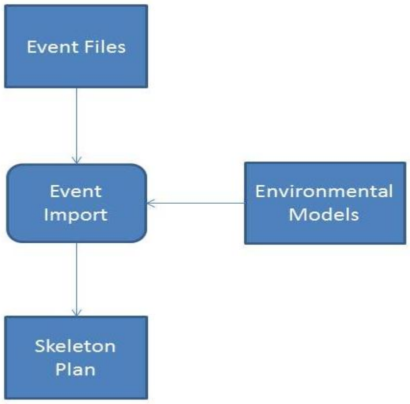

Figure 6. Event import.

A. Event Import

The first step in the process is the ingestion of all events that support the planning, such as orbital events, (un)availability windows, manoeuvres, etc. The events are provided in event files, and are matched against environmental models at ingestion to deduce the corresponding environmental states. The result of the ingestion is a skeleton plan including only events and environmental states.

B. Pre-Planning

The Pre-Planning consist in the generation of all static elements of the plan that can be generated directly from the planning requests and imported events without the need for specific decision algorithms. This includes the generation of the visibility windows or the observation target, the evaluation of the observation conditions, the stations visibility, etc. and the combination of those conditions for the generation of opportunity windows in which all environmental conditions required to implement operations (observations, downlink, etc.) are met. The result is an enhanced skeleton plan, including derived events, derived environmental states, and request-specific opportunity windows.

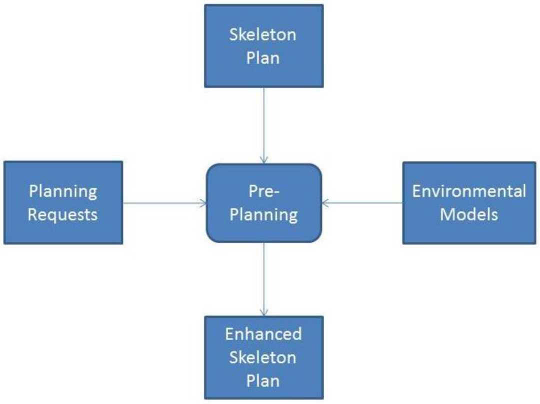

Figure 7. Pre-Planning.

C. Plan Generation

The Plan Generation is the core of the process. At this level the planning requests are implemented on the plan as a set of activities consistent with the opportunity windows of the enhanced skeleton plan, the model of the components supporting the activities (essentially state models of the on-board equipment), and abstract resource models such as power and memory models.

The Plan Generation is performed according to a search strategy that takes into account the priority of the requests and an associated cost function to be optimized on the plan. The search strategies supported are based on local search, and require a fast consistency check of the candidate solutions against the various models involved at this stage. This calls for the use of simplified models, essentially for resources. At this stage complex models cannot be used effectively by the algorithm.

The result of this step is a candidate plan for validation.

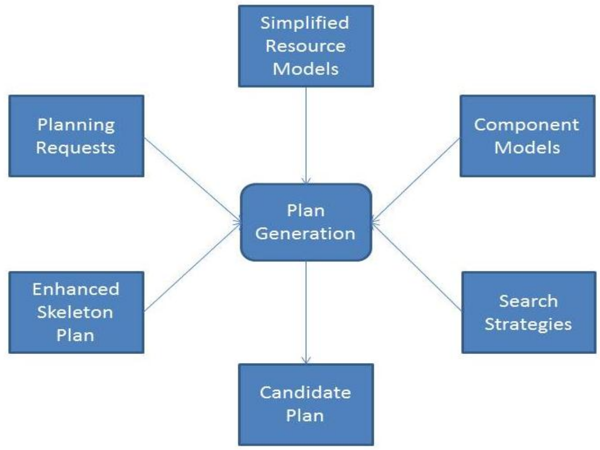

Figure 8. Plan Generation.

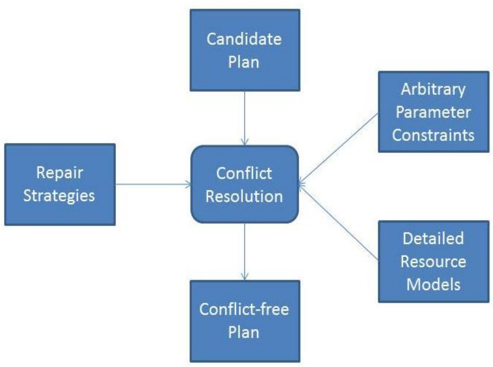

Figure 9. Conflict Resolution

D. Conflict Resolution

The candidate plan produced at Plan Generation must still be validated against complex resources models and arbitrary parameter constraints. At this stage the plan is fully committed, i.e. the value of all parameters involved in the constraint checking, such as the exact timing of each activity, is set.

The Conflict Resolution identifies the remaining conflicts in the plan, and solves them on the basis of repair strategies. In most of the case, the repair leads to a de-scoping of activities on the plan which resolves the conflict. Note that it is not expected at this stage to resolve simple conflicts between requests or activities, such as conflicting overlapping observation. These conflicts are assumed to have been avoided at plan generation.

E. Output Generation

The final step of the planning process is the generation of the necessary output from the generated plans. This includes the schedules for spacecraft, ground stations, and control systems, as well as various types of reports.

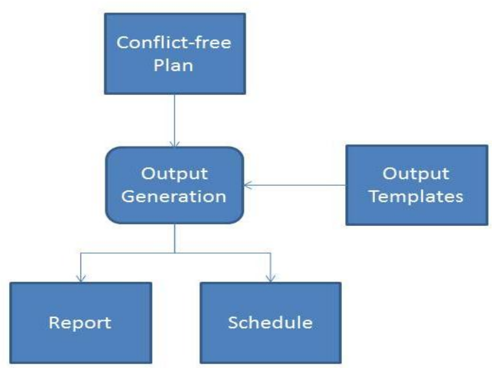

Figure 10. Output Generation.

VI. VPT-E in the VEGA Core Control Segment

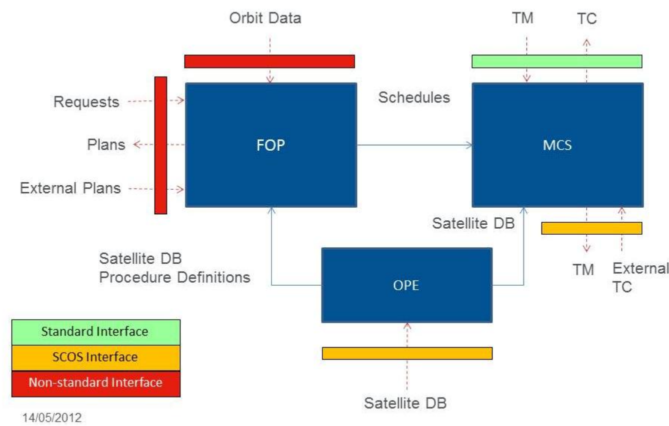

Figure 10. VEGA Core Control Segment.

Relying on the experience gathered in many studies and operational development for ESA, VEGA has integrated the core of a Control Segment including a Mission Control System based on ESA's SCOS-2000, a Flight Operations Planning System (FOP) based on VPT-E, and an Operations Preparation Environment (OPE) based on the VEGA Operations Toolkit Procedure Editor (VOT-PE).

The VEGA Core Control Segment benefits from the configurability of its three components, while providing a higher level building block for missions based on CCSDS and ECSS PUS standards.

The UK Space Agency has started in 2010 the deployment of a ground segment including both control and data processing facilities at the International Space Innovation Centre (ISIC) at Harwell, Oxfordshire. VEGA Space has contributed to the deployment of the ISIC ground segment under the coordination of EADS Astrium.

The contribution of VEGA was the configuration and installation of the Core Control Segment as part of the ISIC Flight Operations Segment, and was the first deployment of VPT-E.

The component whose configuration is the most affected by the deployment for a specific mission is the planning system. The FOP is configured using the basic configuration features of VPT-E:

- The input templates, which support the configuration of the mission-specific input data file formats into the planning system;
- The output templates, which support the configuration of the external products (schedules and reports);
- The environmental models that are used at event import;
- The components models that are used in the plan generation;
- The search strategy, implemented as a set of rules;
- The repair strategy, implemented as a set of rules.

In the ISIC-FOS, the FOP is the central point of all the planning operations and provides visibility of all the activity for CCSDS missions. It must be able to ingest plans from multiple payload operations facilities and co-ordinate the resources so that there is no conflict of resources between the separate missions running out of the Harwell ground segment. The FOP is responsible for:

- Scheduling satellite contacts for uplink of commands and downlink of telemetry
- Co-ordination of ground resources availability (e.g. external TT&C station)
- Declaration of the availability of satellite to the payload planning system (e.g. not available during manoeuvres)
- Verification of mission schedules prior to transfer to the MCS.

The FOP is integrated in a larger programming chain, and focuses on the integration of a number of inputs (e.g. ground station allocation plans, payload operations plans, manoeuvre plans, etc.) produced by different systems into a consolidated conflict free flight operations plan. Thus, the high level requirements on the FOP can be summarized as:

- Initiate the planning cycle and guide it trying to maximize the payload operations return and ensuring the spacecraft safety at the same time
- Produce conflict free integrated flight operations plan

- Derive executable schedules from the integrated flight operations plan for different platforms (e.g. spacecraft on-board schedules, ground station schedules, MCS schedules

The planning cycle is based on exchange of planning data between the Payload Operations Planning (POP) system external to the control segment and the FOP. The POP system provides to the FOP the expected downlink scenarios in terms of the preferred station contact. The FOP produces an initial skeleton plan including events and resource profiles that will be used by the POP system for the payload operations scheduling. The payload operation plan produced by the POP System will be then sent back to the FOP which will check it against the complete set of operational constraints. A feedback and updated plan report then will be sent back to the POP. The cycle continues until the time for dissemination of the executable schedules to the various platforms (spacecraft, mission control system, and ground stations) is reached. Then, the schedules are distributed and executed. The FOP produces:

- Acquisition plans for ground stations;
- A consolidated mission timeline for spacecraft activities;
- Operations plan for the Satellite Control System.

These three logical outputs are converted to executable schedules (e.g. on-board schedule, ground station schedule, etc.) by the output generation functionality of the FOP.

The implementation of the ISIC FOP has significantly impacted the requirements of VPT-E and has highlighted the needs for additional flexibility in the setup of the planning work flow. The interface to the higher-level elements of the programming chain, in the case of ISIC-FOS the Payload Operations Planning system, are essential and require the capability to implement specific high-level protocols. In addition, the interface to Ground Stations at management level to support the booking and configuration of the stations cannot today rely on the CCSDS SLE Service Management standard.

VII. Conclusion and Future Work

VEGA has produced the initial version of VPT in the last months of 2010. Since then it has been integrated to the VEGA Core Control Segment and used in the ISIC FOP implementation.

This initial deployment has provided a significant feedback from both the users of the system and the developers, which is now used to drive the future evolution of the infrastructure.

VPT and the VPT-E extension are used today for the developments of three operational mission planning systems.