Customer	**: ESA**	**Document Ref**	**: VEGA-MPS-APS-FR-036**
Contract No	**: 18316/04/NL/AG**	**Issue Date**	**: 18 June 2007**
WP No	**: WP-5**	**Issue**	**: 1**

Title :

Space Systems for On-line Flexible Science Operations
Final Report - Executive Summary

Abstract : This document presents the Executive Summary of the Final Report of the Study on Space Systems for On-line Flexible Science Operations, which specified the APS prototype system intended to enhance operation of ESA's science operations by a largely automated planning, scheduling and optimisation process for the various scientific observations of a given mission at a science planning level.

Author : ____________________
Robin Steel

Approval : ____________________
Marc Niezette, VEGA IT GmbH

Accepted : ____________________
Eric Bornschlegl, ESA

Distribution :

Hard Copy File:
Filename: VEGA-MPS-APS-FR-036-1.doc

VEGA IT GmbH
Europaplatz 5, D-64293 Darmstadt, Germany
Tel: +49 (0)6151 8257-0
Fax: +49 (0)6151 8257-799
www.vega-group.com

TABLE OF CONTENTS

AMENDMENT POLICY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

AMENDMENT RECORD SHEET

ISSUE	DATE	DCI No	REASON
1	18 Jun 2007	N/A	Initial Issue

1. INTRODUCTION

1.1 Purpose and Scope

The projected Space Systems for On-line Flexible Science Operations prototype, the Automated Planning System (**APS**), is intended to enhance operations of the European Space Agency (**ESA**) science planning by a largely automated but flexible planning, scheduling and optimisation process for the science observation resources. The APS study established a reference model, specified the user-level requirements for future science planning systems.

This Executive Summary of the Final Study Report

- outlines the background and objectives of the study;
- summarises the work performed;
- provides a top level description of the Automated Planning System concepts, and
- reviews the documents produced by the study.

1.2 Structure of the Document

After this introduction, the document is divided into a number of major sections that are briefly described below:

2 SPECIFICATION AND DESIGN

This section provides an outline of the specification and design of the APS prototype which was produced in the context of the Space Systems for On-line Flexible Science Operations study. It also illustrates the concepts behind the APS and how the system interacts to internal and external entities.

3 OPERATIONAL EVALUATION

Within this section we describe the operational evaluation steps performed within the study and any preparation steps necessary for the evaluation.

4 CONCLUSION

In this section we give our final conclusion on the study and its outcome.

5 GLOSSARY

The Glossary contains definitions of acronyms, abbreviations and terms used throughout the document.

1.3 Referenced Documents

The following is a list of documents with a direct bearing on the content of this report. Where referenced in the text, these are identified as [n], where 'n' is the number in the list below:

[1] Space System for On-line Flexible Sciences Operations (APS), System Specification Document, APS-RAL-SDD-0001, Issue 3.0, February 2006

[2] APS Software Interface Requirements Document, APS-RAL-SIR-0001, Issue 2.0, February 2005

[3] APS Software Design Document, VEGA-MPS-APS-SDD-027-1, Issue 1.0, January 2006

[4] Identified GOW definitions for APS, VEGA-MPS-APS-TN-002-1, Issue 1.0, December 2005

[5] Representation of GOW definition for APS, plan evaluation and plan optimisation process, VEGA-MPS-APS-TN-003-1, Issue 1.0, May 2006

[6] APS End to End Test-plan Preparation (Venus Express case), ESA Memo, SOP-RSSD-MEM-021/2-, Issue 1.0, April 2007

[7] APS Evaluation Report, VEGA-MPS-APS-REP-034-1, Issue 1.0, May 2007

[8] APS Software User Manual (SUM), including SW installation, VEGA-MPS-APS-SUM-035-1, Issue 1.0, May 2007

[9] ESA Ground Segment Software Engineering and Management Guide, Parts A, B, and C, BSSC (2002) 1, Issue 1.0, March 2002

1.4 Definitions of Terms

The following terms have been used in this report with the meanings shown.

Term	Definition
Target Opportunity Window	This is a period of time which has been derived from several conditions to provide an opportunity window for when observations or activities can be performed.
Goal	A set of observation/operations with associated conditions attached which when met will allow the observation/operations to be performed.

2. SPECIFICATION AND DESIGN

2.1 Introduction

In the past few years, with the launches of Smart-1, Mars Express, and Venus Express, ESA has directed its effort in developing programmes for planetary science missions, which will continue in the near future with the launch of the missions of the Aurora program.

The increasing complexity of the instruments, required to push the scientific boundaries, leads to an increasing complexity of payload operations. Cutting edge science is always likely to push the envelope of mission resources (e.g. in terms of data volume, power, pointing) and thus require a substantive capability to develop plans for operations and to validate that they are feasible within the available resources. This is reinforced by the development of on-board autonomy; planning must be able to allow for the consequences of that autonomy.

VEGA, in collaboration with the STFC Rutherford Appleton Laboratory (**RAL**), has carried out a study for the ESA in the automation of the planning science process, which aimed at providing an initial platform for the development of an automated planning tool, for supporting and significantly speeding up the science planning process.

The objectives of the study were:

- To automate the generation of the Science plan using a goal based approach with an adaptable optimisation mechanism;
- To show that this can be achieved using a framework which is flexible, adaptable and pluggable
- To formalise in the system the requests from the Principle Investigators (**PIs**)
- To prototype the system and demonstrate using the Venus Express (**VEX**) case

2.2 Planning Deep Space Planetary Missions

A scientific spacecraft orbiting a planet is a platform carrying instruments for which activities have to be planned in order to satisfy scientific and technical constraints as well as to provide a maximum scientific return. Deep space planetary missions are no exception and require such planning. However, they carry some specific features that the planning must consider. The most prominent one is certainly the long distance separating the spacecraft from the Earth. This implies:

- Communication time delays (tens of minutes to hours) that prevent real time operations, necessitate careful “time co-ordination” of the space link (communication) assets, and require spacecraft autonomous operations.
- distance variation change the available resources typically as the inverse square law (e.g. heliospheric distance changes power, Earth distance changes link budget)
- long cruise journey duration with sporadic activities (although the ITT suggests only the in-orbit planetary phase be considered)

Planetary missions can also include lander(s), which implies that co-ordination of the lander communications via the orbiter is required.

The planning of the operations of a planetary deep space mission relies on environmental event predictions, which are produced by an environment simulator on the basis of orbital event predictions provided by ESA’s Flight Dynamics.

The scientific observations are scheduled based on the environmental constraints (visibility of the zone to be observed, illumination of the zone, altitude of the spacecraft over the zone), and spacecraft resource constraints (memory, power, slew capability).

2.3 System context

The general context of APS is illustrated in Figure 1.

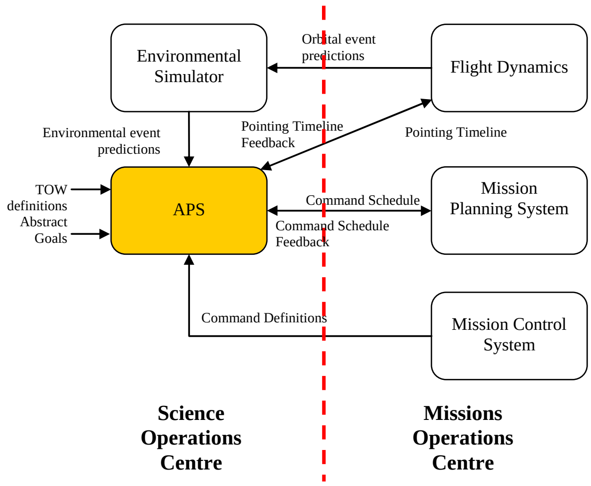

Figure 1: APS Context

The APS is typically deployed in the Science Operations Centre. It takes as input (1) *Target Opportunity Windows (**TOW**) definitions* representing the environmental conditions (typically zone visibility, illumination, and altitude) under which basic S/C operations can be performed; (2) *Abstract Goals* representing the observations that the PI's want to perform as combinations of basic S/C operations, with associated priorities; (3) *Environmental event predictions* provided by an environment simulator; and (4) *Command Definitions* directly extracted from the Mission Control System database at the Mission Operations Centre (**MOC**).

The APS produces plans at several levels of abstraction. The initial output of a planning session is a Pointing Timeline, which is sent to Fight Dynamics of the MOC side for validation. Upon confirmation of the validity of the proposed Pointing Timeline, the APS can proceed with the completion of the planning within these pointing constraints, and produce a Command Schedule, which is sent to the Mission Planning System of the MOC for final validation before upload and execution.

2.4 System architecture

The objective of the APS is to produce a plan of high-level activities that maximises the scientific return of the mission according to a given criteria. The APS planning process is depicted in Figure 2.

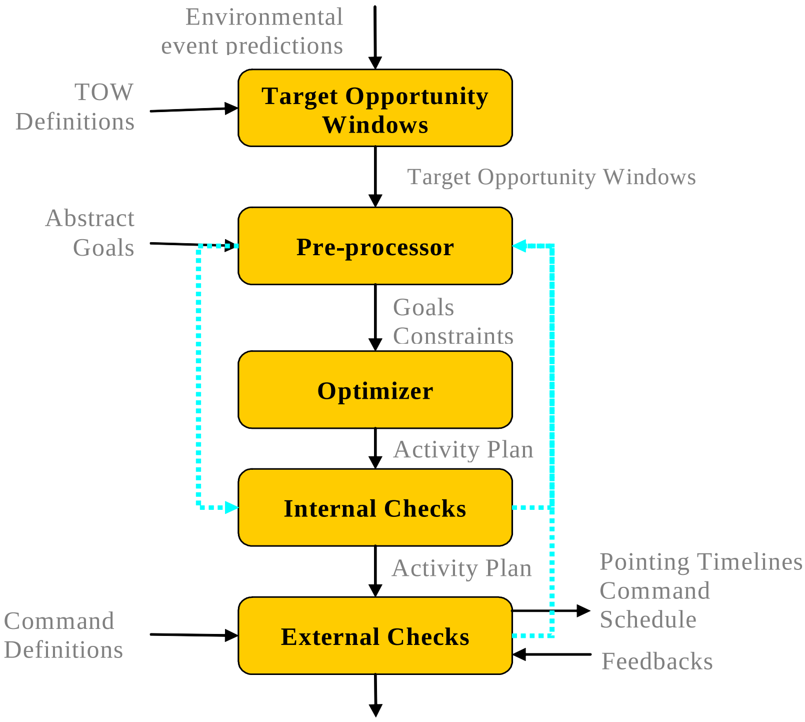

Figure 2: APS Planning Process

2.4.1 Data input and output

There are various inputs that the system requires to operate in a meaningful manner; most of these are in an ASCII based format. Some originate from external entities and are passed into the system via a specific interface. Others are produced by the users of the system and include the TOW definitions and goal definitions.

Through the re-use of the Enhanced Kernel Library for Operational Planning Systems (**EKLOPS**) components, the APS is capable of producing many file output formats. This is achieved by using a generic XML templated output mechanism which allows for a greater flexibility when interfacing with other systems. All of the files generated by the system make use of this functionality, including the generation of the Pointing Timeline Request (**PTR**) and the plan view which is passed to an external Gantt chart viewer and/or an external resource plot viewer for graphical display.

2.4.2 TOW generation

A TOW is a time period in which all conditions for execution of a specific S/C basic operation hold. The first task of the APS is to determine these periods from the TOW definitions and the event predictions provided by the environmental simulator. Figure 3 illustrates the TOW generation process for two types of operations, with a combination of visibility and illumination conditions. Equation 1 provides the corresponding TOW definition for operation_A.

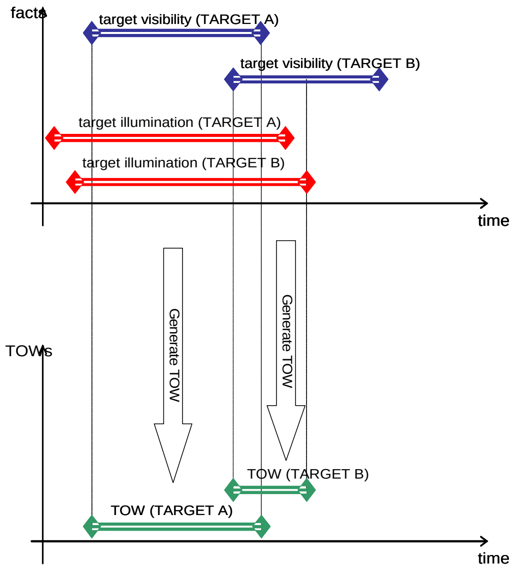

Figure 3: TOW Generation

```
fact(?id1, target_visibility, TARGET A, ?tvS, ?tvE)
^
fact(?id2, target_illumination, TARGET A, ?tiS, ?tiE,)
^
overlap( ?tiS, ?tiE, ?tvS, ?tvE, ?toS, ?toE )
->
activity( ?newId, operation_A, TOW, ?toS, ?toE)
```

Equation 1: TOW Definition

2.4.3 Pre-processor

The second step of the processing consist in interpreting the abstract goals provided as input to APS and converting them into a finite number of candidate goals that can be handled by the optimization algorithm.

In the context of the APS, a goal is a definition of what observations/operations need to be carried out with associated constraints specifying timings, periodicity, durations, etc. to implement a PI's request.

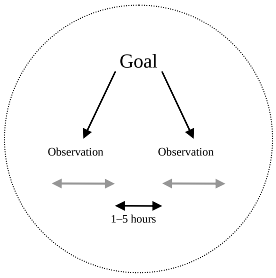

Figure 4: Example Goal Definition

For example, two observations need to be carried out one after the other with a minimum of 1 hour between the observations and a maximum of 5 hours between them, as depicted in Figure 4.

Figure 5 illustrates such operations combinations in the case of the Solar Occultation Observation.

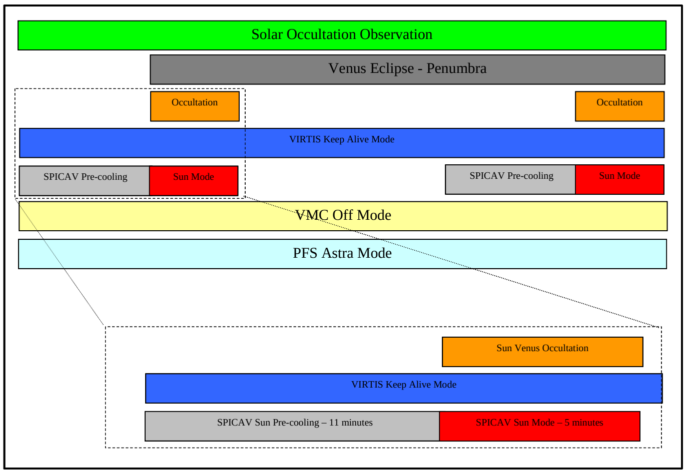

Figure 5: Solar Occultation Observation

The goals that are initially presented to the APS are abstract goals, which specify the combinations of operations that are required, together with a repetition factor (every orbit, every second orbit, as often as possible, etc.). Normally no absolute timing information is included in the goal definition, although this may be required in very specific observations or to indicate a timeout point for a specific goal.

These abstract goal definitions also contain prioritization information, at a global goal level and at an operation level, for the optimizer to make use of during its processing and solving.

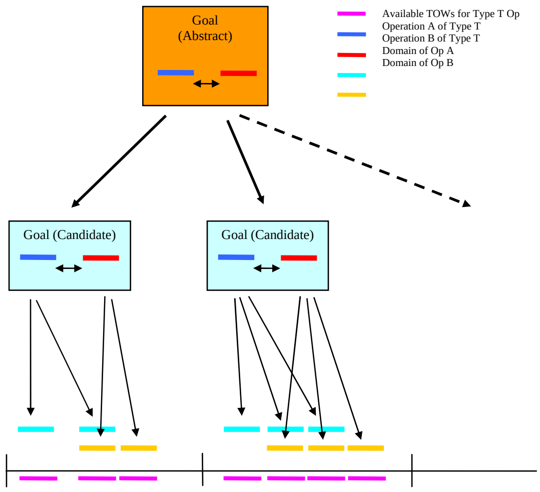

Figure 6: Abstract Goal Expansion

The goals can also include optional observations/operations, which would increase the value of the goal from the PI's perspective, but are not mandatory for achieving the minimum science return expected from the observations or to achieve the goal.

The pre-processing makes use of the actual plan context (number of orbits to plan, generated TOW's, etc.) to expand the abstract goals out into sets of candidate goals, and tries to reduce the temporal domain of implementation of each candidate goal to a minimum prior to presenting these goals to the optimization layer.

2.4.4 Optimizer

The optimization layer is responsible for the extraction and resolution of the optimization problem created by the pre-processor. It consists of an adaptor and a solver. The adaptor extracts the problem and encodes it in a form that the solver can handle, and converts back the solution provided by the solver in components of the plan.

The solver is modular and can be selected depending on the characteristics of the problem, i.e. the variables, constraints, domains, and evaluation function that describe it.

The test performed with the prototype software use mixed integer programming (MIP) for modelling of the problem, and rely on Gnu LP solver for its resolution.

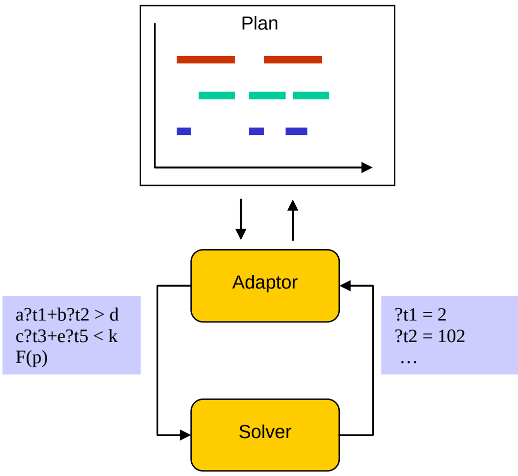

Figure 7: Optimization Layer

2.4.5 Internal checking

The problem solved by the optimization layer relies on abstract resource models (power, memory, etc.), as complex resource models would not be suitable to support complex reasoning methods. These models may not reach the level of accuracy required for ensuring a safe execution of the plan.

Therefore a number of internal checks of the plan against more complex resource models are integrated in the planning logic as post-processing after initial plan generation. In case of failure of some of these checks, a plan repair process would be triggered.

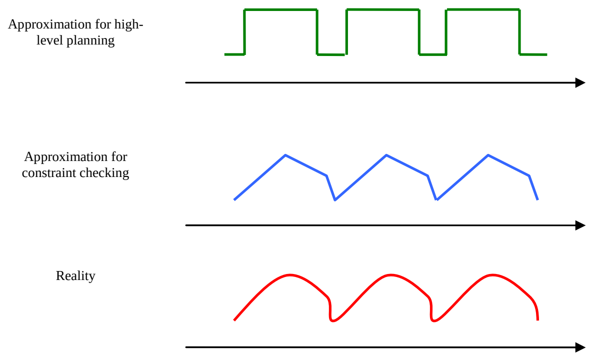

Figure 8: Resource Models

2.4.6 External checking

Some of the most complex checks required to complete the plan validation can only be provided by external checkers. For instance, a typical ESA scenario would involve an intermediate validation by Flight Dynamics of the pointing timelines created by APS and a final check by the MOC Mission Planning System against models of the mission resources.

2.5 Implementation

The core of the prototype planning tool that was developed relies on the use of the EKLOPS, a mission planning system development framework that VEGA developed for ESA to support the development of mission planning systems.

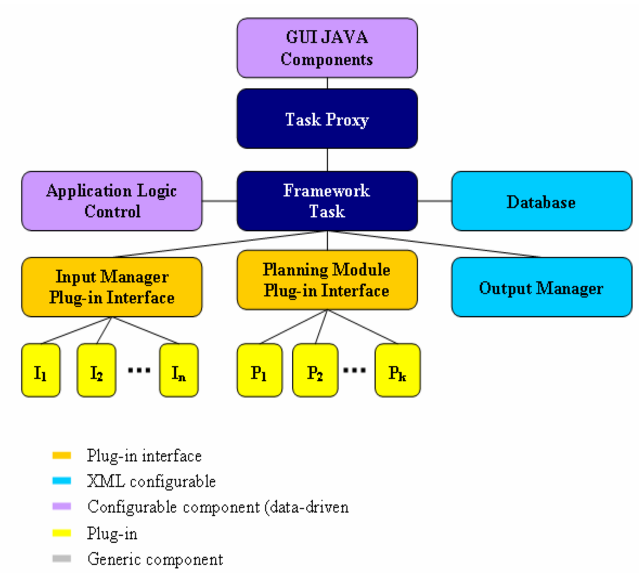

Figure 9: EKLOPS Architecture

In order to support the required functionality, the planning kernel was extended to integrate three major new components that increase significantly the power of the system and its applicability to a larger number of problems.

The plan representation has been extended to allow for the explicit representation of constraints and dependencies on the plan itself, so that the plan can actually be used to share the representation of the planning problem across different planning modules, instead of being only a representation of a partial fully committed solution. The constraint representation modules have been extended to support the explicit representation of temporal and resource constraints

A query language on the plan and on the planning databases has been integrated into the system, so that the TOW generation can easily be implemented on the basis of the production rule mechanism of the original tool.

Lastly, a number of constraint-based reasoning and optimization modules have been integrated into the system to implement the search and optimization algorithms.

Table 1 summarises the key system characteristics of the study and the prototype implementation of the APS.

Table 1: APS Prototype Characteristics

Required System Characteristic	Summary of Technical Approach
Coverage of planetary-type missions	Definition of specific planning layers
Modelling of mission characteristics (generic/specific)	Mission model, supported operational approach
Advanced interfacing functionalities with the users	Descoped (in the prototype)
Flexibility, Modularity and Evolution capability	Object Oriented adaptive approach (Plug-In), constraint/model-based approach
Cross-platform inter-operability	Model-based software engineering, LINUX
Portability of the system to interface in any SOC with (SCOS-2000) MOC	ASCII files interfaces, importer for SCOS-2000 Mission Information Database (**MIB**)
Long-term maintainability	ASCII files interfaces, C++, modular design, decoupling of MMI and processing
Re-use of open-source software, and software developed under ESA IPR	Core system derived from ESA MOC planner, integration of freeware solver

Table 2 below summarizes the extent of the current prototype implementation of the APS.

Table 2: APS Prototype Implementation Status

High-Level Requirement	Extent of the Current Implementation
Interface with FD/MOC (static)	ASCII interface, configurable event definitions
Interface with External Simulator (static)	ASCII interface, configurable event definitions
Generation of additional secondary events	Configurable using logical statements (LMP)
Generation of intervals from events	Configurable in Mission Model
Import of Target Opportunity Windows	Supported
Identification of Target Opportunity Windows	Configurable using logical statements (LMP)
Completion of Target Opportunity Windows	Not supported in the prototype
Modular planning algorithm	Plug-in interface for processing modules (in C++ or LMP), separation of optimisation algorithm in the design
Grouping of related activities into goals	Supported
Memory Utilization Conflict Detection	Abstract model in the Mission Model for plan

High-Level Requirement	Extent of the Current Implementation
	construction, more accurate model for plan checking and repair
Power Conflict Detection	Checking using models similar to the MOC planner models
Pointing Conflict Detection	Detection of conflict between pointing requirements
Thermal Constraint Checking	Event-based, similar to the approach followed in the MOC planners
Slew Conflict Detection	Abstract model in the Mission Model for plan construction, more accurate model for plan checking and repair
Operational Conflict Detection	Conflicting modes specified in Mission Model, other conflict can be specified and checked using LMP
Downlink planning	Supported
Goal priorities	Supported
Optional Operations	Supported (entire goal or operations in goal)
Modular optimisation algorithm	Separation of optimisation algorithm in the design, replacement may require an adaptor (if not Mixed Integer Problem (**MIP**) solver)
Relative distribution of goals to users	Only possible via utilisation function, not demonstrated in the prototype
Fixed or proportional goal value	Partly supported by optional operations in goals
Utility function as sum of individual values	Supported
Tracking of Past Planning Activities	Covered by master plan history, limited use in the algorithm
PTR Generation	Covered by output configuration
Payload Operation Request (**POR**) Generation	Covered by output configuration
Template-based Output Generation	Supported
Interactivity with External Simulators (dynamic)	Not covered in the prototype
Use of external checkers (FDS, MOC)	Interface to read back the feedback not present
Plan regeneration – automatic repair	Not covered in the prototype
Plan post optimisation update	Not covered in the prototype

3. OPERATIONAL EVALUATION

3.1 Introduction

Part of the evaluation process for the APS was to configure the system for use with the Venus Express, using the mission as the bases for the test scenario. A workshop was setup to demonstrate this and to try out the system.

In preparation for this workshop a set of goal definitions and target opportunity window definitions needed to be defined. The system also needed to support all the file interfaces to be used by this configuration of the system. To aid in this task a document was produced (see [6]) outlining the components that would make up an end-to-end test scenario based on the Venus Express mission. The two day workshop was then used to enhance this understanding of the end-to-end test specification and the goals that would be needed to define this test scenario. The end-to-end test case was then evaluated and report produced (see [7]).

3.2 Prototype results

As part of the prototype validation, a test case scenario was created based on the Venus Express mission taking into account the current scientific and operational constraints. The science planning for the Venus Express mission is split into ten science cases each requiring different environmental conditions to perform the required science. For the validation of the system, eight of the ten science cases were selected, along with the mandatory maintenance window allocation and earth communication windows.

For each science case a number of TOWs had to be initially generated for the goals to refer to. An example of one of these TOW definitions is given in Equation 2 and illustrates the opportunity window for a solar occultation observation.

```
fact(?id1, Venus_Occultation, 1000km, ?tiS, ?tiE)

^ fact(?id2, Venus_Occultation, 0km, ?taS, ?taE)

^ overlaps(?id3, ?tiS, ?tiE, ?taS, ?taE, ?toS, ?toE)

^ ?tmS <- ?tiS - 000.01:30:00.000
```

Equation 2: Solar occultation TOW definition

Figure 10 and Figure 11 illustrate one solution which was produced by the system as part of the Venus Express test case scenarios. They show the goals, which have been selected by the optimizer module of the application, to be planned and the TOWs relating to these goals, along with the derived pointing timeline to carry out the selected science activities.

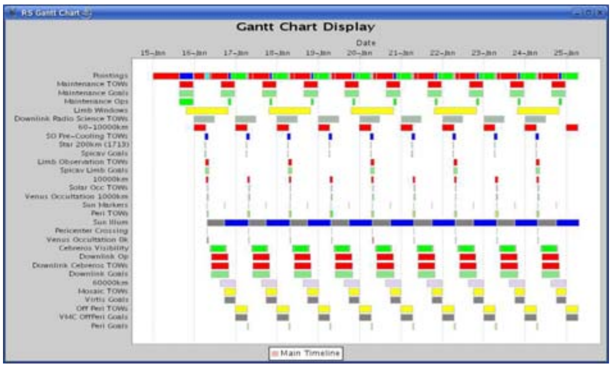

Figure 10: Solution generated by the APS

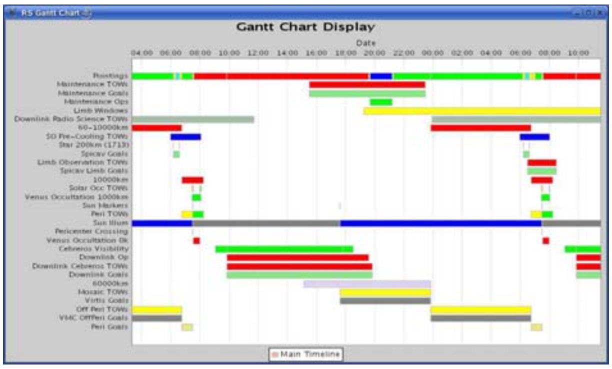

Figure 11: Solution generated by the APS (2)

3.3 Known limitation of the prototype

3.3.1 Development architecture

The prototype has been developed on a 32-bit architecture. This means that the virtual address space used by the system for its persistent storage is limited. Due to the re-used implementation of this persistent storage functionality the size of each database within the system is also limited. On a 64-bit architecture these limitation are not as much an issue. It is feasible to re-compile the prototype for a 64-bit architecture but this is not within the scope of this study and has not been tested.

3.3.2 Third party solver

During testing of the prototype using the Gnu LP solver it was discovered that this third party library has some interesting artefacts depending on the dimensions of the plan and the way the encoding of the plan has been implemented. This can result in a plan seemingly having no solution. Unfortunately this is not reported by the solver and is not easily identified. It may be possible using the many various parameters that can be set on the solver to make this more robust but this is not within the scope of the study to investigate this further.

3.3.3 Initial Goal definitions

Specification of two or more of the same operation (i.e. two of the same pointing mode) within a single goal definition file is not possible in the current implementation. This is related to the goal definition files structure rather than the internals of the system. The underlying system can handle more than one operation of the same type being specified within a goal but the input definition of the goal does not support this. To accommodate this, the goal definition file handling would need to be extended.

3.3.4 Slew handling

The slew handling and generation within the prototype uses a fixed slew model for demonstration purposes, although it has been designed so that this can be exchanged for a more dynamic slew model in future incarnations.

3.3.5 Inter-goal constraints

Another limitation of the prototype is the definition of inter-goal and inter-instrument constraints. Even though the internals of the system can support these types of constraints, there is no mechanism in place for them to be specified or configured by the user. Further study would also be need to understanding the impact of these types of constraints on the overall problem domain.

3.4 Performance

Measuring the real performance of the system requires more research as it is not a simple task. The performance of the prototype depends a lot on the dimensions of the problem to be solved. The more elements that have to be considered and the more conflicting constraints a problem has the more complex it is to find a solution and hence the longer it will take to compute the solution, if one can be found.

Within the test scenario that has been illustrated earlier in this document, the performance was fair with a solution being produced within 1 hour on a shared machine with only a 1Ghz processor.

A mechanism has been added to the solver rule to allow some control over the solver. This are specified in the form of rule configurations parameters and control some timeouts for the solver to use. Using this mechanism, it can be set up to drop out of the processing after a given amount of time, a certain number of cycles or after a given percentage of completeness. In the context of the project, this can facilitate the functionality to swap the solving algorithm to another approach if the processing seems to take too long to arrive at a conclusion.

3.5 Further scenarios

A further test of the system would be to configure all the rules for each star that can be used with the Spicav star occultation observations, giving each star a relative priority/weighting and seeing how the system performs in this scenario. The solver could then be left to find a solution. Another possible similar scenario would be to do the same sort of thing with limb observations.

4. CONCLUSION

- The APS prototype provides a core framework for automated planning of science missions
- APS is a modular system based on an object-oriented adaptive framework
  - Mission Model includes operations and associated constraints
  - Configurable input and output interfaces
  - Domain-specific logical language for query and conditions (TOW generation)
  - Optimisation algorithm decoupled from the core of the system
  - Plug-in interface for additional planning modules
- The system is available under a specific license agreement from ESA

4.1 Study Outputs

4.1.1 APS Prototype software

This is the main output of the study and consists of a functional prototype allowing the specification of TOWs, goal definitions and mission model through the use of configuration files. The prototype has been built and designed with flexibility and reuse in mind. It illustrates this by incorporating an external Linear Problem (**LP**) solver to perform the optimisation step. The actual planning steps performed and the order they are performed in can be configured through the XML configuration files. A software user manual (see [8]) was also produced to accompany the APS prototype.

4.1.2 APS Technical Notes ([4] & [5])

Two technical documents were produced for the APS. The first containing the initial understanding of some of the science cases of the VEX mission that could be later used to define the TOW definitions and goal definitions for testing and illustration purposes. The second covers the technical aspects of the evaluation function and algorithmic approaches to producing an optimised plan using weighted goals and goals with associated priorities. It also highlights the problems that need to be considered with regards to planning using a goal base approach.

4.1.3 APS End-to-end Scenario Evaluation report [7]

This report outlines all the TOW and goal definitions defined for the performance of an end-to-end test scenario based on the VEX mission. It then illustrates, at various stages, the prototype as it performs its tasks showing the production and removal of the candidate goals which have been derived from the initial abstract goals provided to the system. Finally, the report gives a summary of the results produced by the prototype and indicates the main limitations to the prototypes implementation.

5. GLOSSARY

The following acronyms and abbreviations have been used in this report.

APS	Automated Planning System
EKLOPS	Enhanced Kernel Library for Operational Planning Systems
ESA	European Space Agency
LP	Linear Problem
MIB	Mission Information Database
MIP	Mixed Integer Problem
MOC	Mission Operations Centre
PIs	Principle Investigators
POR	Payload Operation Request
PTR	Pointing Timeline Request
RAL	STFC Rutherford Appleton Laboratory
RD	Reference Document
TOW	*Target Opportunity Windows*
VEX	Venus Express

This Page Is Intentionally Blank