Massaging the plan with the Language for Mission Planning

Bruno Teixeira de Sousa
ESOC /ESA, OPS-OPV (Venus Express Flight Control Team)
Robert Bosch Strasse 5, 64293 Darmstadt, Germany, bruno.sousa@esa,int

I. Abstract

In today's world, the fundamental problem of mission operations is no longer related to establishing access to the spacecraft (monitoring and control).

The connectivity to the spacecraft is already a given condition:

- Mature and extensively tested Mission Control Systems
- A network of ground stations of different ranges and covering the complete sky
- Fast digital connectivity to any point in the globe
- Fast digital connectivity via the space link on X and K band to most of the near by solar system
- Large data storage capacity on-board in solid state memories
- Large networks of sensing devices to monitor the platform (thousands of TM parameters)
- Programmable autonomy on-board (Mission Timeline, SW routines, On-Board Control Procedures, etc.)

All this allows the mission operator to have a clear view and full control capacity over the spacecraft.

The core problems in mission operations today are **planning** and **planning**:

On one side planning and scheduling of spacecraft operations within available resources and constraints - e.g. prioritisation of payload activities, maximizing scientific data return, power management, battery degradation management, data production, storage and download management, thermal constraints, consumables, wheel momentum, illumination conditions, payload restrictions, system wide mode or environmental incompatibilities, etc. - Some of these may involve complex optimization processes.

And on the other side planning and scheduling of ground resources and operations (e.g. station and pass duration allocation to multiple missions, station configuration, operators on shift, Mission Control System automation schedules, orbit determination and collision avoidance, ground science with radio tracking, etc.). Here as well the level of complexity can be quite high.

In a typical mission the complete planning process is a distributed one, where different particular tasks are performed by particular stakeholders who interface among each other with intermediate planning products, and interface to the spacecraft and ground systems with consolidated products. As such there is not a single Mission Planning System that does it all.

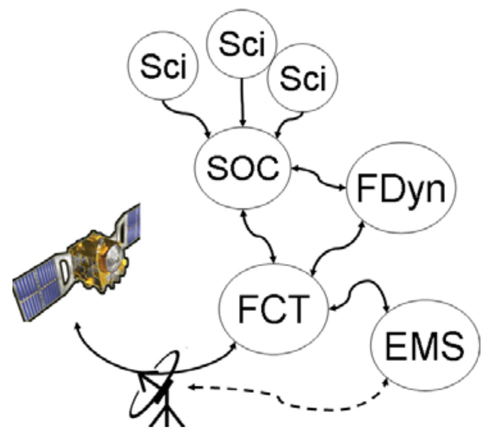

Figure 1 - Mission Planning Interfaces at VEX.

Nevertheless, and for what concerns the tasks allocated to the Venus Express (VEX) Flight Control Team (FCT) when designing the operations planning concept, a planning system was envisaged and put into place and is commonly referred to as the Mission Planning System (MPS).

This particular system interfaces with at least 5 external stakeholders:

- Science Operations Center (SOC) – provides communications skeletons and data bit rate usage over time, and receives operations requests in return.
- Flight Dynamics (FDyn) – receives orbital and attitude related events and profiles,

pointing modes and AOCS operations requests.

- ESTRACK Management System (EMS) – receives ground station allocation plans, and produces station allocation modification requests and station control events and parameters.
- JPL Deep Space Network (DSN)– receives ground station allocation plans and produces station control events and parameters.
- Mission Control System and Spacecraft Controllers (SPACON) – produces TC stacks for upload and pass sheet data for the human controllers.

Its main features are the capabilities to:

- Interface with the identified stakeholders by ingesting and generating agreed products.
- Consolidate a full operations scenario based on specific requests from external parties and systematic routine requests for the platform by the FCT (TX actuation, dumps, etc.).
- Validation of the operations against environmental conditions modelled by external parties or derived modes modelled internally by the MPS.
- Modelling of power and data resource usage and check associated constraints.
- Expansion of commanding entities to low level, time ordered, ready for upload mission TC timelines.

The complete life cycle of any given operations scenario, from planning to operation, can last up to a year since it starts being drafted until it's fully refined. We can safely say that more than 95% of all operations uploaded by the FCT to the spacecraft have been at more than one stage (typically 3) validated and processed by the MPS. This makes the MPS a very critical and central system to the whole operations process efficiency and safety.

An operations plan is in the end an enormous collection of inputs from different parties that needs to be made consistent, valid, safe and complete. The mechanism by which the MPS is able to "massage" a plan – extracting relevant relations and identifying relevant events or time windows, performing systematic routine space and ground operations scheduling, and performing plan validation – should fulfil the following requirements:

- Rule based
- Organised in an editable logical workflow
- Flexible and easily programmable by an operator
- Unambiguous and predictable

To achieve this, the Venus Express team expanded the MPS in 2009 with a new component called the Language for Mission Planning (LMP). It allows the operator to define logical, linear temporal rules that can be systematically applied over the data elements of a plan to verify relations among them and to generate new activities or modify existing ones.

LMP empowers the planning team to define directly the planning rules without the need to request software expert intervention, thus reducing dramatically the turn around time to deploy new or modified rules to the planning process – a must in an ever changing operational environment.

This paper is a follow-up on a previous one [1] presented at SPACEOPS-2010, in Huntsville, where the LMP implementation and syntax were presented. The present paper presents LMP scope in the context of a broader Planning and Scheduling system, describes its current usage in VEX and other projects, and the way it's been deployed operationally.

II. Scope and Rational for a Language for Mission Planning

LMP is a language for *querying* parameterized data from a plan and *manipulating* it – it is in that sense a plan data preparation language, hence the usage of the expression "massaging" the plan. As such it is a very useful tool for Planning systems that need to generate plan optimizations, because it allows determining, from a very populated and complex plan, the time periods or activities that are relevant for an optimization process (gaps to fill, windows of opportunity, particular alignments of data, etc.).

In Venus Express there is currently no optimization problem to solve (the payload utilization plan is pre-optimized by the SOC, and the attitude and trajectory plan are optimized by FDyn), so LMP is mostly used in the systematic planning solution loop mode (see Figure 2).

However this capability to prepare datasets for AI engines has been demonstrated in the EMS [5, 6] system which makes use of LMP to determine Service Opportunity Windows that are then further processed by AI Algorithms to calculate an optimal ground station allocation plan for all missions.

Also in Mars Express, there is a case where LMP is currently used to determine a ground station utilization plan by selecting, based on rules, whether to take available ESTRACK or DSN (NASA) tracks, and then passes this information to an AI planner, called MEXAR[2, 3, 4], that provides an optimal solution for the data downlink strategy.

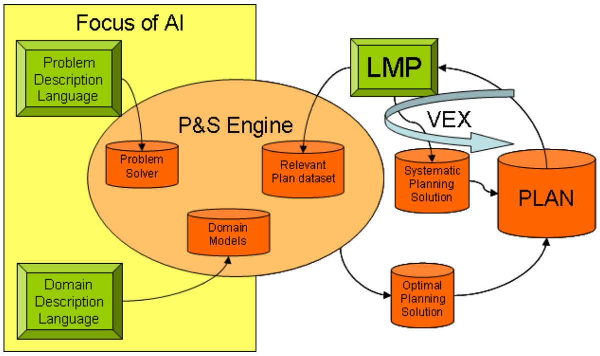

Figure 2 - The scope of LMP within a simplified AI Planning System schematic.

LMP could also be used as an important complement to systems making use of the APSI [7, 8] planning framework, also developed by ESA, which covers approximately the left side capabilities shown in Figure 2. At the moment there are no examples of this setup.

An important feature of LMP is that it is of a declarative nature, meaning that the user expresses **what** he wishes to happen, and **not how** the System should accomplish it. The user basically expresses that "WHEN these conditions" are met in the plan "THEN these other conditions" should also be met. The way they are met is transparent to the user, thus requiring less "programming" skills.

Giving system configuration information in the form of a language will typically give much more flexibility than option switches and parameter set via a graphical user interface and/or in system configuration files (the latter was the case for VEX before LMP). A few simple grammar rules can allow the user to build complex expressions in ways that need not have been foreseen by the system developers.

One obvious alternative option would be to make the planning system scriptable using an off-the-shelf scripting language. This would give the user full control over the planning process, but also the full burden of algorithm development, optimization, de-bugging, etc. Typically, a spacecraft operations planner is not expected to possess the necessary skills nor have the time for this.

III. Nature of Planning Data

LMP is ideal to manage planning data that has the following characteristics:

- Represents atomic occurrences at instants/periods in time
- Relevant relations between data items are of temporal nature: Before, After, Overlaps, Contains, Exists, etc.
- Nature of manipulation: Create, move, interrupt, remove

Such data items are referred to, in this context, as facts. Facts are logically grouped into Elements and are characterised by having a name (descriptive of its function), a start and end time (an absolute UTC time stamp) and parameters (attributes to complement the classification of the fact).

A plan populated with facts[1] can then be queried on any of the properties mentioned above and new facts can be created by assigning values to these properties, or existing ones can be modified.

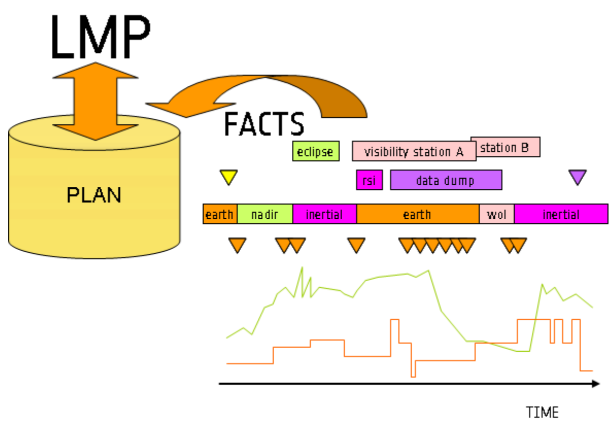

Figure 3 - Example of data items that populate a plan

Facts are used to represent to following types of relevant planning entities:

- Modes of instruments or systems and the transitions between them
- Commanding entities (procedures, command sequences, individual TCs, directives, tasks, etc.)
- Events (as in orbital or attitude events, derived synthetic events, ground station related events, etc.)

[1] Note that the facts have to be somewhere configured, known and supported by the underlying planning repository. In the case of VEX MPS, all the facts, their names (type), their grouping into elements, and associated parameters are pre-configured in the planning system configuration repository and made accessible to LMP rules.

- Resource markers – these are special types of facts[2] used to represent (and store) the evolution of a numerical quantity over time (for example the evolution of power consumption).

Typically a plan is initially partly populated with data originating from external parties - such as orbital information provided by Flight Dynamics, operations requests from scientists or station allocation by a network provider - and then it is further expanded and validated.

IV. Usage examples from VEX

This chapter gives real examples of how VEX operational planning rules have been translated into LMP rules for different purposes. As already mentioned the usage of LMP made in VEX is of a pure systematic planning nature, with no plan optimization being required. But this does not limit the scope of LMP to only this type of usage.

The main types of planning activities where LMP is useful for VEX are:

- Deriving new events – These are required internally for checks, for plan visualization, for driving the scheduling of operations, and to produce event files for external parties.
- Auto-Scheduling of operations – inserting routine conditional sequences automatically in the plan
- Conflict Checks – Alerting the planner that un-desired conditions are met in the plan

i. Derive new events from existing data

LMP is extremely useful at deriving new events based on other known data in the plan (received from external sources, or calculated by previous rules).

As an example the determination of the spacecraft communications periods from the station allocation events will be demonstrated. These are the periods when it is expected the spacecraft to have its transponders active, they are used, among other things to calculate the amount of data that can be downloaded and assign actual transponder commands. These events are called STTM (start of Telemetry)

[2] Each fact represents a segment of the complete plot. The start and end value of the segment are stored in two fact parameters and each segment follows the next continuously (separated by 1 micro-second) in a plan.

The Ground Station scheduling office automatically assigns Venus Express a daily communications pass (or track) on Cebreros, our prime ground station, or alternatively in New Norcia if the prime is not available, according to a previously agreed baseline allocation rule. The actual times of the assigned tracks are provided to the FCT via an agreed interface – a file called Plan View containing the allocation of all stations to all missions over a time frame of approximately one year, and updated weekly. Each track has an associated parameter called "support_type" containing the string "daily_TTC_vex". This distinguishes the purpose of this track from other possible ones, like radio science. Out of that file only the allocations for our mission are extracted into the plan, which then become facts (named CEBREROS or NEW_NORCIA), grouped under an element called Station_Tracking.

```
// Queries all Cebreros tracks
fact(?id1, Station_Tracking, CEBREROS, ?taS, ?taE)
// binds variable ?type to the values of parameter support_type
^ (parameter(?id1, "support_type", ?type)
//  filters the ones that contain the string daily_TTC_vex
^ like(?type, "daily_TTC_vex"))
// offsets the start time of those events by -6 minutes
^ ?offsetS <- ?taS - 000.00:06:00.000
// offsets the result by minus OWLT
^ owlt (?offsetS, -1, ?start_time)
// offsets the end time by 2 minutes
^ ?offsetE <- ?taE + 000.00:02:00.000
// offsets the result by minus OWLT
^ owlt (?offsetE, -1, ?end_time)
// calculates the duration by subtractinf end from start
^ ?duration <- ?sptm_time - ?sttm_time
// create new events STTM at the calculated times
-> fact( ?newId2, Station_Communications, STTM, ?start_time, ?end_time)
// set the parameter duration accordingly
-> parameter( ?newId2, duration, ?duration)
```

Figure 4 - LMP rule to derive Communication periods.

The activation of the transponders is routinely initiated 6 minutes prior to the start of the track to give time for warm-up and interrupted 2 minutes after the end of track, but all this in Spacecraft Time, thus shifted by one negative factor of OWLT[3] (for a spacecraft in Venus). This rule has been implemented in LMP as shown in Figure 4[4].

[3] OWLT – One Way Light Time.

[4] The editor (Notepad++ with user defined LMP plug-in) highlights the language features: operators in blue, variables in green, quoted strings in grey, fact/element/parameter names in black and comments in italic brown. The symbol "^" represents an AND/Conjunction of conditions; the symbol "->" indicates the statements that are produced as a consequence of the rules conditions being fulfilled; fact, parameter, like and owlt are some of the language available operators. The same operator (e.g. fact/parameter) used in the condition part of the rule

ii. Auto Scheduling of Operations

LMP is also very useful for creating rules that automatically schedule routine spacecraft command sequences under given conditions. There are 4 areas where routine automatic commanding is required in VEX and they're implemented with LMP:

- Interrupt storage of data in the mass memory during on-board clock wrap-around events
- Bit rate adjustment at given events (function of spacecraft distance to Earth) or at DDOR[5] events (always performed at 22 kbps)
- Stopping the dump of science data at the end of a pass
- Full management of the Transponder activation for Communications, DDOR and Radio Science passes.

The activation of the VEX transponders needs to be scheduled from ground at the correct time for a given ground track and with parameters that define the configuration of the transponder according to the activity being performed, including whether it includes:

- uplink carrier
- telemetry modulation and its settings
- usage of on-board transponder 1 or 2
- usage of USO[6]

To cope with the different configurations and activate the transponder correctly different command sequences which include the different necessary parameterizations of the commands, have been instantiated that need to be executed at specific times.

For a normal Communications pass the sequences used are called ATTF600A[7] to switch ON the transponder and ATTF610A to switch off and they need to be scheduled at the STTM events (as calculated in the previous example). We're also required to enable (and correspondingly disable) the routing of the Telemetry to the Radio channel at 10 minutes before the start (and 10 minutes after the end) of the pass; this is done with sequences ADMS820D and ADMS820C respectively.

```
// Queries all the STTM events calculated before
fact(?id1, Station_Communications, "STTM", ?taS, ?taE)
// offsets the start by -10 minutes
^ ?off_hk_start <- ?taS - 000.00:10:00.000
// offsets the end by 10 minutes
^ ?off_hk_end <- ?taE + 000.00:10:00.000
// schedules TC sequence to enable data routing
-> fact( ?newId1, DMS, ADMS820D, ?off_hk_start, ?off_hk_start)
// schedules TC sequence to switch on Transponder
-> fact( ?newId2, TX, ATTF600A, ?taS, ?taS)
// schedules TC sequence to switch off Transponder
-> fact( ?newId3, TX, ATTF610A, ?taE, ?taE)
// schedules TC sequence to disable data routing
-> fact( ?newId4, DMS, ADMS820C, ?off_hk_end, ?off_hk_end)

// Queries all transpoder switch on sequences
fact ( ?id1, TX, ATTF600A, ?taS, ?taE)
// And all the switch offs
^ fact ( ?id2, TX, ATTF610A, ?tbS, ?tbE)
// offsets both end times by one hour
^ ?taoff <- ?taE + 000.01:00:00.000
^ ?tboff <- ?tbE + 000.01:00:00.000
// filters only the occasions when they overlap
^ overlaps(?id3, ?taS, ?taoff, ?tbS, ?tboff, ?toS, ?toE)
// removes those sequences from the plan
-> remove(?id1)
-> remove(?id2)
```

Figure 5 – 2 LMP rules, one to schedule command sequences to activate and de-activate the on-board transponders and telemetry routing, and the other to remove transponder cycling closer than one hour

Furthermore, an additional constraint has been introduced that when the switch OFF (ATTF610A) and ON (ATTF600A) are scheduled within an hour of each other, they shall be removed from the plan to prevent excessive TX re-cycling. This can happen when a DDOR and Communications pass are too close to each other or overlap. These rules have been implemented in LMP as shown in Figure 5.

iii. Conflict Checking

LMP is also helpful to check a plan for situations where operations are conflicting with flying rules and constraints. This may arise because, on one hand, different parties are responsible for planning different aspects of the mission: the Science Operations Centre (SOC) will activate the payloads, the Flight Dynamics team will control the attitude and orientation of the Spacecraft, the FCT will activate the transponders; and on the other hand there are flying rules that affect simultaneously these different aspects, for example:

- Payload Aspera may not be ON (SOC) during a Wheel-Off- Loading manoeuvre (FDyn) and for one hour after, as gases from the thruster may damage the instrument.
- Transponder may not be ON (FCT) when the spacecraft cold faces are illuminated by the Sun (FDyn) due to thermal load limitations.

---

results in a query/filter, and used in the statements. part of the rule results in the creation/modification of facts.

5 DDOR - Delta Differential One-Way Ranging – Used to determine accurately Spacecraft position and velocity

6 USO – Ultra-Stable Oscillator

7 This is the unique code used to identify a command sequence in the Mission database. The naming convention is a mission specific choice.

Because it's fundamental to be absolutely sure that such cases are thoroughly detected, and do not result in violations of flight rules, LMP is used to scan the plan for such situations and raise alarms. The figures below show the rules defined above expressed in LMP.

```
//Queries all periods when WOL is active
fact(?id1, "S/C pointing", "MAINT_WOL", ?taS, ?taE)
// Offsets the end time by one hour
^ ?end <- ?taE + 000.01:00:00.000
// Queries all the periods when ASPERA is ON
^ fact ( ?id2, "ASPERA", "AS-ON", ?tbS, ?tbE)
// Checks the time periods when they overlap
^ overlaps (?id3, ?taS, ?taE, ?tbS, ?tbE, ?toS, ?toE)
// Raises the corresponding warning
-> message (Warning, "ASPERA ON during WOL between time ",
            ?toS, " and ", ?toE)
```

Figure 6 - LMP rule to verify payload Aspera activation during WOL restriction periods

```
// Queries all the periods when Transponder X-band is ON
fact ( ?id1, X_Band, "X_ON", ?taS, ?taE)
// And all the periods when there is an illumination in one of the cold faces
^ fact ( ?id2,Face_Illumination,?state, ?tbS, ?tbE)
// Filters for the events containg an S character
// indicating start, e.g. IPYS - start illumination on plus Y face
^ like (?state, "S")
// Filters the periods when they overlap
^ overlaps ( ?id3, ?taS, ?taE, ?tbS, ?tbE, ?toS, ?toE)
// Raises the corresponding warning
-> message (Warning, "Cold Face illumination violation when TX 1 ON due to ",
            ?state, " between time ", ?toS, " and ", ?toE)
```

Figure 7 - LMP rule to verify Transponder activation during Cold face illumination periods

V. Bringing LMP into operations

LMP was introduced operationally in the VEX MPS in 2009 to address a problem of scheduling transponder activation to perform different types of radiometric activities, and it soon became evident that it performed rather well coping with a great number of other planning rules.

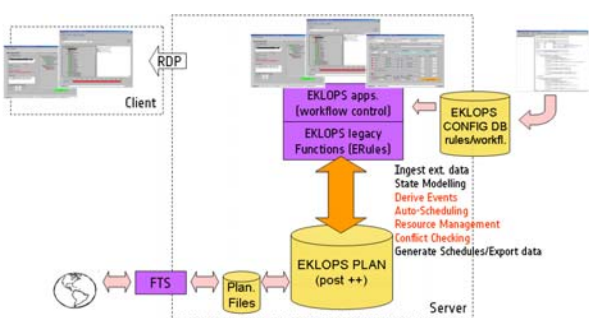

Figure 8 - The original layout of the VEX MPS system: the functions highlighted in red were deficient and lacking flexibility. The rule/workflow configuration (in XML) was managed in simple text editors. The MMI had many separate windows/applications for different functions and was rather cumbersome to use.

Today in the VEX mission, when performing the weekly short term operations preparation, the Mission Planning System, will execute a workflow containing a total of 83 rule modules (each containing rules that perform ingestion of planning products, deriving events, scheduling operations, modelling power and data, checking conflicts and exporting command stacks for the spacecraft and other products), out of those, 47 modules are containing LMP rule sets (the remaining modules use MPS legacy functions), that corresponds to more than a half of the total.

The rules and workflow are managed by elements of the Flight Control Team (no software development involved) and kept under configuration and version control. They are tested in an offline representative environment before being deployed to the actual operational system. This setup allows that any modifications to flight rules or operations strategies, identification of new constraints or plan checks, implementation of new on-board activities, can be achieved (in an automated fashion) and deployed in a very short time (less than a week).

Besides VEX and EMS systems, LMP is also heavily used now by the **Mars Express** mission in particular to manage their uplink and downlink opportunities and sorting which station to use when overlapping ESA and JPL tracks are available to them. A similar usage solution is being currently implemented by the **Cluster** mission and is being brought into operations.

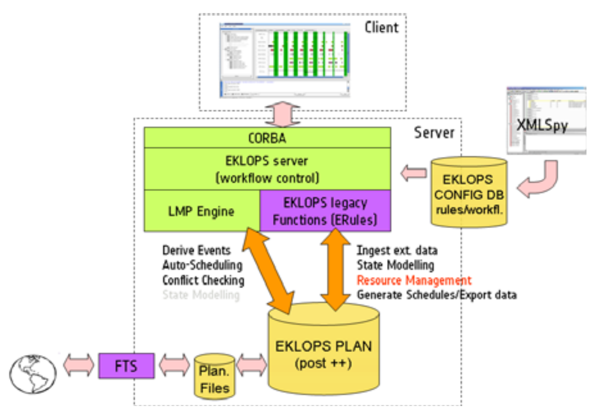

Figure 9 - Current layout of the VEX MPS system. The client-server architecture has been modified. The MMI has been completely redone. LMP engine has been integrated. Configuration is now managed with a tool set consisting of XMLSpy® (for planning items, interfaces and workflow) and Notepad++ (for LMP rules). The

system is still deficient in the area of resource modelling.

All these systems are based on the EKLOPS [9, 10] planning framework developed by VEGA, and the LMP implementation is integrated as an extended library.

VI. Conclusion

When the mission planning and scheduling rules are linear – in the sense that they don't involve optimization decisions (maximizing or minimizing resource usage, for which some form of artificial intelligence is required) – then they can be expressed in an unambiguous way that can be interpreted and systematically applied by a planning machine.

LMP uses elements of set theory and logical connective, combined with the manipulation of *time related* facts to give the mission planner the full flexibility to express relevant rules to manipulate (massage) the plan. LMP is designed to specifically cope with the time and parameterized nature of the data being handled. It is therefore suitable for any planning domain.

LMP is a simple language, easy to learn, with a small set of operators and an uncomplicated structure making it painless to adopt. The extension of a mission may be linked to modification of the initial operational concept. LMP offers the flexibility needed for adaptation of the planning processes.

It is possible that we'll see other implementations of LMP appearing as new missions take the concept on board, as is the case for **Bepi-Colombo** and **Rosetta**. The **Gaia Mission**, which is in the process of receiving and validating their MPS, is also using LMP but they have created an additional layer whereby their own macro language gets expanded to LMP expressions, allowing them to use both.

In the future we're planning on adding a few more operators to complement the currently available functionality. We're also prototyping an editor that will combine the capability to write syntactically correct rules, build a workflow with them, and then test their effect on plan data.

As such LMP is not a powerful planning and scheduling engine in itself, in the sense that it is not smart and it does not allow you to find optimal solutions, but it is a very powerful mechanism to automate systematic plan searches and modifications (which in some missions is the main bulk of the work), and to find the relevant datasets that are needed to be considered for an optimization process by AI engines. This makes LMP a potent ally to AI P&S engines.

Acknowledgements

The author would like to acknowledge the work of Erik Noréus and Robin Steel, and the rest of the mission planning systems development team at VEGA for their outstanding work in creating and implementing this feature. I would also like to thank Octavio Camino, the VEX Spacecraft Operations Manager, for making it possible to deploy this implementation operationally.

References

Papers

[1] Use and abuse of the Language for Mission Planning by Venus Express – B.Sousa, E.Noréus, AIAA-2010-2101, SPACEOPS 2010, Huntsville, USA

[2] Studying Decision Support for MARS EXPRESS Planning Tasks: A Report from the MEXAR Experience - Amedeo Cesta, Gabriella Cortellessa, Angelo Oddi and Nicola Policella, IWPSS 2004

[3] Mexar2 - An Operational Tools for Continuous Support to Mission Planning - Amedeo Cesta, Gabriella Cortellessa, Simone Fratini, Angelo Oddi, and Nicola Policella, IWPSS 2006

[4] Using an Artificial Intelligence Tool to Perform Science Data Downlink Planning as Part of the Mission Planning Activities of Mars Express - Erhard Rabenau, Amedeo Cesta, Angelo Oddi, Alessandro Donati, and Thomas Léauté, IWPSS 2006

[5] Automated Allocation of ESA Ground Station Network Services - Sylvain Damiani, Holger Dreihahn, Jörg Noll, Marc Niézette, and Gian Paolo Calzolari

[6] Improving Performance and Interoperability of the ESTRACK Planning System, Alexander Hoffmann, Holger Dreihahn, Marc Niézette, Gerhard Theis, IWPSS 2009

[7] Advanced Planning and Scheduling Initiative: MrSPOCK AIMS for XMAS in the space domain - R. Steel et al. - IWPSS 2009, Pasadena, USA.

[8] MrSPOCK: Long-term Planning for the ESA Mars Express Mission - Amedeo Cesta, Gabriella Cortellessa, Simone Fratini, and Angelo Oddi - IWPSS 2009, Pasadena, USA

[9] Noll, J. and Steel, R. 2005. EKLOPS: An Adaptive Approach to a Mission Planning System. Proc. Of IEEE Aerospace conf.

Computer Software

[10] EKLOPS 2.0 (Mission Planning System Kernel Software and package) including LMP libraries (as described in this paper) available under ESA licensing conditions for Solaris. EKLOPS 3.0 for Linux, as used by EMS, has currently small deviations from this paper.