Flexibility & Autonomy in Mars Express Planning using LMP

R. Steel & E. Rabenau

Telespazio VEGA Deutschland GmbH, Darmstadt, Germany
name.surname@vega.de

NOVA Space Associates Ltd, ESOC, Darmstadt, Germany
name.surname@esa.int

Abstract[1]

The paper sets out to introduce some of the planning problems faced by the Mars Express Mission Planners, detailing the Medium Term Planning (MTP) constraints and the change in planning strategy needed for the Very Short Term Planning (VSTP) due to an unexpected spacecraft anomaly which occurred back in November 2011. We illustrate the MTP communications planning problem and how the previously unused Language for Mission Planning (LMP) functionality, of the already operational planning system, were employed to determine the station usage plan and communications operations in an iterative plan refinement approach. The paper further illustrates the activity-based planning strategies that were needed to accommodate a spacecraft anomaly that changed the complete short term planning concept for Mars Express. We look at the way the operational planning system has been adapted and is being used to provide safety and, ultimately, automation in the production of the commanding products. Finally, concluding with some observations and lessons-learnt and how AI techniques could be applied to improve the planning process.

Introduction

As with most other space missions, Mars Express suffered from a number of anomalies. Some of them had an impact on the operations concept. This was also the case with an anomaly discovered in November 2011, that affected the solid-state mass memory and the way it is to be operated.

The paper discusses two representative planning problems that have been solved using the Language for Mission Planning (LMP) functionality of the existing operational planning system used by the Mars Express mission. The first involved a migration of functionality from an existing toolset into the main planning process and application, ensuring its usage and maintainability. This toolset dealt with the planning of station allocations and communication opportunities needed for transmission of data to Earth and the uploading of commanding to the spacecraft.

The second problem discussed is one that has arisen from the solid-state mass memory anomaly affecting the commanding and safety handling of the spacecraft. Due to the anomaly the command resources of the spacecraft were dramatically limited, requiring that the tools supporting the planning process be adapted to cope with the new dimensions.

It must be noted that the authors originate from the operations side of the space domain where the terms planning and scheduling respectively refer to the scheduling of activities on a timeline and the generation of the executable command schedules that are sent to the spacecraft.

Background

Mars Express was launched with the Beagle Lander in June 2003, arriving at the Red Planet 6 months later when it released its passenger that was unfortunately lost upon entry, descend or landing. Nonetheless the orbiting spacecraft has been a success story. Mars Express has been surveying the Martian surface and environment for almost 10 years now taking high resolution and 3D images of the planet , performing spectral analyses of its atmosphere and the boundary layers below the surface of Mars with a radar. Throughout these years Mars Express has been supported by a number of tools developed for ground support to ensure the operational success of the mission. These are the tools that are used on a daily basis for operations which have to adapt to the needs of the mission as it evolves over the years, evolutions which can be due to degradation of components on the spacecraft, unexpected failures and other such anomalies. One such tool is the Mission Planning System of the Flight Control Team.

---

[1] The contents of this paper are the views of the authors and do not necessarily reflect the views of European Space Agency.

MPS2010 – Communications Planning

In 2009 an evolution of the operational planning tools was conceived. The previous stand-alone MPS++ toolset that was written in the IDL programming language was to be incorporated into the main MPS application after successfully proving the operational concepts and validating operational requirements. The MPS++ toolset had been developed by the mission planners to cope with the changes to the operations concept that mainly surfaced after launch. Its main functions were to generate a de-overlapped station allocation plan, issuing station release notifications and station usage periods for transmission to the station networks.

Through the flexibility and configurability that had been developed over the years prior to this in the main MPS and in conjunction with developments on related missions also using the tool, this task became mostly a matter of configuration of LMP rules to be applied to the planning problem. To understand what was really needed though required an understanding of the constraints that drove the problem.

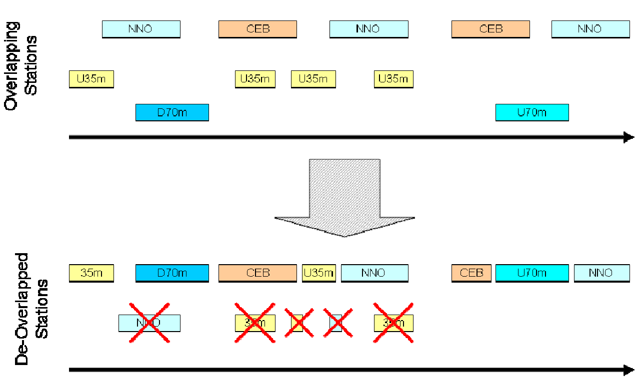

Figure 1 : Station De-Overlapping

Constraints Overview

There are many levels of constraints that have to be considered for the Communications planning for Mars Express. They range from hand-over times between stations to minimum and maximum transmitter usage durations. Even the type of station has an influence on the constraints of the problem be it an ESTRACK station, DSN station, station with uplink, station without uplink, 35 meter diameter dish or 70 meter diameter dish. These constraints can be grouped into three main areas, those for station de-overlapping, those for downlink communication and those for uplink communication.

The station de-overlapping constraints handle the selection of a station or part of a station based on its dish size, its uplink/downlink capability and usable duration.

```
/*
*  Generates the sweep markers for non 70m DSN stations  taking into account the
* minimum TC duration,  the station offset, owlt and sweep duration
*/
environmentVar("MIN_TC_ON", ?minTcDur)
^ environmentVar("SWEEP_DURATION", ?sweepDur)
^ environmentVar("ESTRACK_OFFSET", ?stationOffset)
^ fact(?id1,"Station_Allocation", "Station_Selected", ?gst, ?get)
^ parameter(?id1, "StationId", ?station)
^ (like(?station, "Norcia") v like(?station,"Cebre"))

// Adjust for 10 deg on ground
^ fact(?idDeg, "Station_Allocation", "TX10Deg", ?txs, ?txe)
^ overlaps(?gst, ?get, ?txs, ?txe, ?st, ?et)

^ fact(?id2, "TC_Pre-Switchings", "resourceMarker", ?sst, ?set)
//  Calculate the TC start and end times here and only create
//  sweep markers if the allowed minimum duration is maintained.
^ owlt(?sst, +1, ?tcSt)
^ owlt(?set, -1, ?tcEt)
^ ?dur <- ?tcEt - ?tcSt
^ ?dur > ?minTcDur   // Minimum allowed TC duration

^ parameter(?id2, "startValue", ?val)
^ ?val > 0.0
^ owlt(?sst, -1, ?sstOG)
^ owlt(?set, -1, ?setOG)
^ overlaps(?oid, ?st, ?et, ?setOG, ?setOG, ?ost, ?oet)
^ ?gsst <- ?sst - ?sweepDur     // Sweep duration onboard
^ owlt(?gsst, -1, ?gsstOG)
^ ?possSweep <- ?gsstOG - ?stationOffset      // Station Offset

^(

// Clause for basing timings on the ground station start time or 10 degree start
(?st > ?possSweep
^ ?sweepStart <- ?st + ?stationOffset       // Station Offset
^ owlt(?sweepStart, +1, ?obSweepSt )
^ ?obSweepEt <- ?obSweepSt + ?sweepDur  // Sweep duration onboard
^ owlt(?obSweepEt, -1, ?sweepEnd )
^ owlt(?obSweepEt, +1, ?relTcSt)
^ ?durTc <- ?tcEt - ?relTcSt
^ ?durTc > ?minTcDur  // Minimum allowed TC duration
^ ?relSt <- ?st + 000.00:00:00.000
)
v

// Clause for basing the timings on the sweep end on-board (i.e. TYON)
(
?st <= ?possSweep
^ ?obSweepEt <- ?sst + 000.00:00:30.000
^ ?obSweepSt <- ?obSweepEt - ?sweepDur    // Sweep duration onboard
^ owlt(?obSweepSt, -1, ?sweepStart )
^ owlt(?obSweepEt, -1, ?sweepEnd )
^ ?relSt <- ?sweepStart - ?stationOffset      // Station Offset
))

^ parameter(?id1, "dsn", ?dsn)
^ parameter(?id1, "support_mode", ?mode)
-> activity(?nid2,"MEF", "SWES", ?sweepStart, ?sweepEnd)
-> parameter(?nid2, "support_mode", ?mode)
-> parameter(?nid2, "StationId", ?station)
-> parameter(?nid2, "dsn", ?dsn)

-> activity(?nid3,"MEF", "SWXS", ?obSweepSt, ?obSweepEt)
-> parameter(?nid3, "support_mode", ?mode)
-> parameter(?nid3, "StationId", ?station)
-> parameter(?nid3, "dsn", ?dsn)
-> activity(?nid4,"MEF", "SWXE", ?obSweepEt, ?obSweepEt)
-> parameter(?nid4, "support_mode", ?mode)
-> parameter(?nid4, "StationId", ?station)
-> parameter(?nid4, "dsn", ?dsn)
-> activity(?nid5,"MEF", "SWEE", ?sweepEnd, ?sweepEnd)
-> parameter(?nid5, "support_mode", ?mode)
-> parameter(?nid5, "StationId", ?station)
-> parameter(?nid5, "dsn", ?dsn)

-> activity(?nid6,"Station_Allocation", "SSOB", ?sstOG, ?setOG)
-> activity(?nid7,"Station_Allocation", "ReleaseBlock", ?relSt, ?obSweepEt)
```

Figure 2 : Sweep generation LMP example

The antenna dish size has implications for the downlink bit rate and determines the support of radio science activities, e.g. bi-static radar, solar corona and occultation measurements. A station with uplink is preferred over a station with downlink only (the DSN supports so-called multiple satellites per antenna tracking – MSPA) to provide the necessary windows of opportunity to send commands to the spacecraft for later execution.

Downlink communication is mainly governed by the requested science observations and the visibility of the spacecraft to earth. Additionally, the transmitter on the spacecraft has operational constraints that need to be adhered to, such as the transmitter configuration time, minimum switch on duration, the maximum switch on duration and the minimum switch off duration between any two switch on activities. These constraints are in place to ensure the health and safety of the transponders along with the limiting of power usage by the transmitters. Once these have been resolved then the downlink operations can be defined using these initial communications windows taking into account the minimum and maximum downlink durations, although this is not the final story for the downlink windows.

The uplink plan (see Fig. 2) is quite complex as the example LMP snippet shows with the generation of sweep activities for the ESTRACK stations used by Mars Express. An uplink plan consists of several components that need to be performed for an uplink window to exist. The first of these being the uplink sweep activity from the station to obtain a stable lock on the satellite. This is governed by several factors, i.e. the elevation of the satellite above the horizon, the real duration of the uplink window following the sweep activities and the offset that needs to be applied depending on the type of station. Additionally, the generation of sweep activities also has an impact on the downlink windows which need to be interrupted for this activity.

All throughout planning the one-way light time has to be taken into account, as some timepoints are provided in ground time and others in on-board time, in order to establish a consistent time base.

Iterative Approach

To achieve the goal of producing a de-overlapped plan of station allocation and communication usage, a plan refinement approach was employed. This entailed making an initially very coarse plan refinement of the station allocations taking into account the simple constraints to de-overlap different station types depending on their basic characteristics, basic meaning their dish size and uplink/downlink capability.

The next step is to apply the constraints that determine if it is possible or required to communicate with the ground, initially forming transmitter activities governing the switching on and switching off. Through this activity additional station time could be released or even whole stations released due to them not being needed anymore. The station allocation plan is then refined to reflect this new state. Within this process the transmitter minimum allowed switch off time is checked and where necessary transmitter periods are joined together, causing a re-evaluation of the transmitter maximum switch on time and further refinement to the transmitter timeline. Downlink opportunities are then imposed on top of the transmitter timeline taking into account the constraints mentioned previously for durations.

Following this we further refine the plan by defining the uplink communication activities based on the constraints provided. This step then affects the result of the previous refinement because the downlink windows need to be interrupted to accommodate the newly generated uplink communications, forcing a re-validation of the previous constraints to ensure consistency is maintained. Several iterations area performed using many LMP rules to obtain the final communications plan.

Where could AI techniques be used?

The planning of the communications and de-overlapping of stations in the manner discussed in the previous paragraphs is not necessarily an optimal one. From the first step, and throughout, we have reduced the problem to make later processing easier, but we have made decisions that may not necessarily be optimal for the resulting solution. The details and constraints imposed by the pointing timeline could have made a difference to how the stations were de-overlapped in the first instance if this information was known earlier. It could be that we reduced the usage of one station for another only to release the second station later because the transmitter could not be switched on for the minimum allowed duration. In this case keeping the first station may have provided more downlink time or even more uplink time. Another scenario would be that we had joined two potential transmitter periods together because of the minimum period between them being too short, only to cut the whole transmitter period because it violates the maximum switch on constraint when it might have been possible to just cut the start of the second transmitter period, effectively forcing the minimum off period, to allow for more transmitter time.

This is where AI techniques could be introduced, performing searches of the possible plan space to improve the quality of the final communications plan.

Some AI technology is already being used during the medium term planning in the form of MEXAR2 [Cesta et al., 2007], which is a tool that determines the data downlink strategy from the various packet stores of the Solid State Mass Memory.

MPS2012 – Activity-based planning

The next major evolution resulted from a spacecraft anomaly. The connection between the command stack and the execution stack became intermittently corrupted the data upon transfer from one to the other, causing the fault detection to trigger. A new commanding concept had to be devised quickly and the current tools adapted to support this new strategy.

To bring this into context, previously it was possible to load a whole week's worth of commanding, approximately 3000 commands, onto the on-board commanding queue in one go. These would then be transferred to the immediate queue for execution. Up until then, this had worked flawlessly . A new approach had to be taken that would verify that the commanding had been successfully loaded onto the immediate queue before they are executed. To achieve this an existing mechanism was employed to disable and enable commanding on an Sub-Schedule Identifier (SSID) level but this meant that the commanding had to be grouped into sets of commands small enough to be held in the short-term queue. This gave rise to the "activity-based" planning where related commanding is grouped together and only executed if all commands from the group have successfully been loaded on the short-term queue.

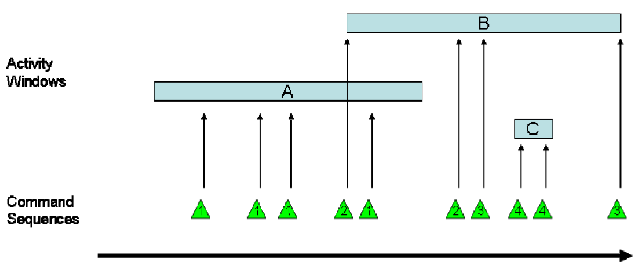

Figure 4:Sequence Allocation to Activity Windows

Approach used

To facilitate the activity based planning the current MPS had to be re-configured and adapted to support the handling of SSID's and the grouping of commanding into so called Activity Windows which needed to be defined in some way. What defined an activity window? This was one of the first hurdles that needed to be jumped. In the beginning the definition of these activity windows was not known but the mechanisms to support this had to be in place as quickly as possible. Fortunately, the use of the LMP for this task meant that we could easily define and redefine these windows as we needed and as the mission evolved. This became an asset to the mechanism because the criteria for defining these activity windows changed as more experience was gained following the anomaly.

So the first step was to define the initial activity windows based on the criteria provided by the mission and the experiences being made. Once these initial activity windows have been made we assigned the relevant commanding to these windows again using LMP rules to map the sequences to the relevant windows (fig 5).

The ultimate goal of this activity-based planning was to produce the same commanding as previously obtained but

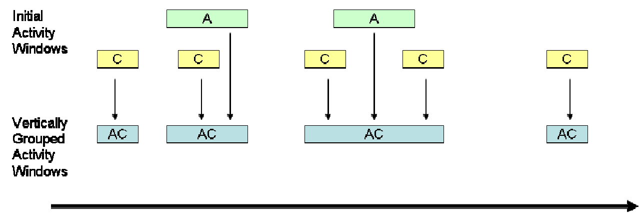

Figure 3: Vertical Grouping of Activity Windows

by splitting the resulting command schedule up into smaller commanding files (chunks) that were independent of each other as much as possible, meaning that if one commanding file failed to load successfully it would not prevent other ones from loading or executing successfully. This meant that a constraint had to be introduced prohibiting that two activity windows of the same type overlap with each other. Additionally, very early on in the evolution of this mechanism it was observed that the number of commanding files being produced was often excessive due to the definitions of the activity windows. A balance had to be made between the number of commanding files to be uploaded and the acceptable level of failure should a commanding file not be successfully loaded onto the short-term queue. This is where the notion of vertical and horizontal joining of activity windows came in.

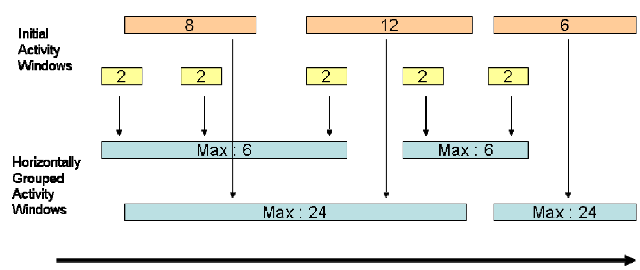

Figure 5: Horizontal Activity Window Grouping

Horizontal grouping (Fig. 5) allows for several activity windows of the same type to be grouped together into a single group based on a maximum number of commands allowed within a chuck of that group type. Vertical grouping (Fig. 4) on the other hand allowed different activity window types to be grouped into the same commanding file. This was needed to cater for the cases where multiple criteria for a given group type were needed for the same activity windows.

Once all the groupings have been performed one last pass is made to determine what we called the final activity windows. These windows represent the period from the first command to last command of a file of grouped commanding which are eventually generated as mini-command schedules to be uplinked to the spacecraft for later loading and execution.

Trigger Generation

Once the final activity windows have been determined with their commanding associated to them, we are now able to generate the triggers that actual load these commanding files onto the short term stack from there storage location on the SSMM. The process here is a simple one of calculation initially, determining when the commanding files have to be loaded to allow all the commands contained within them to be load onto the short-term stack before the first command from the commanding file has to be executed. Or is it? I guess you can already see a possible catch here. What if two commanding files overlap when being loaded onto the short-term queue? This situation is not allowed because of the sequential nature of the command loading mechanism, so additional constraints had to be applied to ensure that no two commanding files of commanding are being loaded onto the short term queue at the same time and with enough gap between them to not cause additional problems. This we achieved using the LMP with a multiple pass approach which progressively resolved any overlap conflicts of commanding file by applying a given set of resolution constraints to the commanding files in conflict. Possibly not the most efficient solution but it is effective.

De-ghosting

Due to the change in operational commanding of the spacecraft an unwanted side effect manifested itself allowing so called ghost files to exist. The SSMM, which stores the command stack files before they are loaded onto the execution stack, can be viewed as a basic file stack. Files are only removed from the stack if they are taken from the top of the stack. Any file removed from within the stack simply creates a gap in the stack which cannot be used again until all files on top of it are also removed, effectively creating a ghost file. This meant that an additional activity had to be introduced into the planning cycle to allow for the ghost files to be manually removed from the SSMM to free up the space for newer commanding files.

So how does this effect the activity based planning? In a nutshell it doesn't directly affect the activity based planning but affects the loading of the commanding files onto the short term queue. For a de-ghosting activity to be successful all commanding during the de-ghosting period has to be loaded onto the short term queue allowing the SSMM to be emptied completely to remove the ghost files. This means that all triggers that load command files onto the short term stack have to be executed before the de-ghosting period starts. It also means that the amount of commanding during a de-ghosting period is limited by the available space on the short term stack before the de-ghosting activity commences. For operational safety a de-ghosting period is defined as lasting from the start of a given station pass until the end of a backup station pass which can be many hours apart.

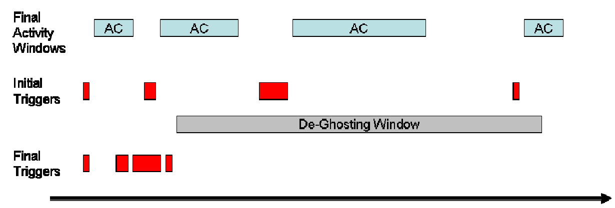

Figure 6: De-Ghosting Adjustments

The main difficulty here was that these triggers had to be executed as late as possible to prevent overfilling the short-term queue but maintaining the constraints of overlap between commanding files.

Where could AI techniques be used?

An obvious place where AI could be used within this context is in the handling of ghost files on the SSMM. These ghost files are created because it is not possible to empty the file stack sufficiently to remove them. With the correct strategy it should be theoretically possible to keep the number of ghost files within an acceptable limit or even reduce the number of ghost files on the stack as the mission progresses without the need to perform regular de-ghosting activities which reduces the observation capabilities during this period.

Another place where AI techniques could be employed is in the actual horizontal and vertical grouping of activities. Our approach follows a simplistic scanning approach but you could imagine an approach that would minimize the number of commands in the short-term stack at any given time by grouping the activities differently to this taking the allowed maximum number of TCs into account. This could allow for additional activities to be performed due to commanding stack space being freed up more often.

Conclusion

Flexibility is a key function of an operational planning system that needs to be built in from day one. The mission planning system of Mars Express has shown its maturity on many occasions, the ones discussed being the most prominent ones. Through the use of the LMP it was possible to provide the necessary adaptations to cope with the changes of the operations concept. Furthermore, the use of LMP has set the ground for the mission to advance further towards automation on-ground, providing products that are generated in a common thread without further intervention by a human.

So where could AI techniques be used in the problems described within this paper? This is a very subjective question and one which hopefully provokes thought for future missions. We have given some examples throughout the text but the possibilities of introducing AI techniques into such problems and operational missions are not limited to these. There are many more avenues to be explored and opportunities to be grasped in the current and future missions. .

References

Noll, J., and Steel, R., 2005. *EKLOPS: An Adaptive Approach to a Mission Planning System.* IEEE Aerospace co conference, March 2005, Big Sky Montana, USA.

Bru, T., Rabenau, E., Niézette, M., Shaw, I., IWPSS 2004, *Mars Express Mission Planning: An example of successful software reuse*, IWPSS 2004, Darmstadt, Germany 2004

M.Niézette, I.Shaw, E.Noréus, R.Steel, J.Noll, S.Kellett, H.Schönhardt, 2005, *Planning with EKLOPS: Lessons Learned and Latest Developments*, in Proceedings of the Sixth International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations (RCSGSO 2005), Darmstadt, Germany.

Shaw, I., Steel, R., Ercolani, A., Sousa, B., 2006, *Venus Express Mission Planning System – A Further Reuse Of EKLOPS*, in Proceedings of the Seventh International Symposium on Space Mission Operations and Ground Data Systems (SpaceOps 2006), June 2006, Rome, Italy.

Niézette, M., Shaw, I., Noréus, E., Steel, R., Noll, J., Dreihahn, H., Schönhardt, H., 2006, *Efficient Planning Systems Development with EKLOPS*, in Proceedings of the Ninth International Conference on Space Mission Operations (SpaceOps 2006), June 2006, Rome, Italy.

Sousa, B., Accomazzo, A., Shaw, I., Steel, R., 2006, *VEX MPS – Supporting the Venus Express Mission Planning Concept*, poster contribution in the International Workshop for Planning & Scheduling in Space (IWPSS 2006), October 2006, Space Telescope Science Institute, Baltimore, MD. USA.

Niézette, M., Steel, R., Hutchinson, G., Bornschlegl, E., 2007, *Automated Planning of Science Planetary Missions with EKLOPS*, in Proceedings of the Data Systems In Aerospace (DASIA) conference, May 2007, Naples, Italy.

Niézette, M., Damiani, S., Dreihahn, H., Noll, J., Noréus, E., Shaw, I., Steel, R., 2007, *Extending EKLOPS: New applications of the Mission Planning Kernel*, in Proceedings of the Seventh International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations (RCSGSO 2007), Moscow, Russia.

[Cesta et al., 2007] Cesta, A., Cortellessa, G., Denis, M., Donati, A., Fratini, S., Oddi, A., Policella, N., Rabenau, E., and Schulster, J. (2007). *MEXAR2: AI Solves Mission Planner Problems*. IEEE Intelligent Systems, 22(4):12–19.

Niézette, M., Steel, R., Shaw, I., Dreihahn, H., Damiani, S., 2008, *Test Planning using EKLOPS*, in Proceedings of the Tenth International Conference on Space Mission Operations (SpaceOps 2008), Heidelberg, Germany.

Steel, R., Niézette, M., Bornschlegl, E., 2008, *Automated Planning of Science Planetary Missions: Reality or Myth?*, in Proceedings of the Tenth International Conference on Space Mission Operations (SpaceOps 2008), May 2008, Heidelberg, Germany.

Sousa, B., Noreus, E., 2010. Use and abuse of the Language for Mission Planning by Venus Express. AIAA-2010-2101. In *SpaceOps 2010.*