Benefits for the standardization of Mission Planning Services from Advanced Planning Experiences

Simone Fratini, Mehran Sarkarati, Nicola Policella, Alessandro Donati, Mario Merri, and Mariella Spada

European Space Agency, ESA/ESOC, Robert-Bosch-Str. 5, Darmstadt, Germany
name.lastname@esa.int

Abstract

The ambitious goal of the Spacecraft Monitoring & Control (SM&C) Working Group of the Consultative Committee for Space Data Systems (CCSDS) is to define a set of standardized, interoperable mission operation (MO) services, which allow rapid and efficient construction of co-operating space systems. Such services will have to be general enough to cope with the various needs of existing and future missions, but at the same specified enough to be practically usable.

This paper presents some ideas to deal with this difficult task drawn from existing architectures and interfaces used in AI advanced software systems for planning and scheduling.

CCSDS Mission Operations (MO) Services

The Spacecraft Monitoring & Control (SM&C) Working Group of the Consultative Committee for Space Data Systems (CCSDS), which sees the active participation of 10 space agencies, has been working since 2003 on the definition of a service oriented architecture for space mission operations. The ambitious goal of the WG is to define a set of standardized, interoperable mission operation (MO) services, which allow rapid and efficient construction of co-operating space systems (Ground Segment, but also part of the Space Segment).

For this purpose the WG has defined a MO layered service framework, shown in Figure 1 which allows mission operation services to be specified in an implementation and communication agnostic manner. The core of the MO service framework is its Message Abstraction Layer, MAL, which ensures interoperability between mission operation services deployed on different framework implementations.

The MO services are defined in compliance to a reference service model using an abstract service description language specified by the MAL. For each concrete software implementation and communication technology the abstract service contracts must be bound to that particular technology. The MAL layer provides in turn standardized interfaces in form of Application Programming Interfaces (API) towards both upper and lower layers.

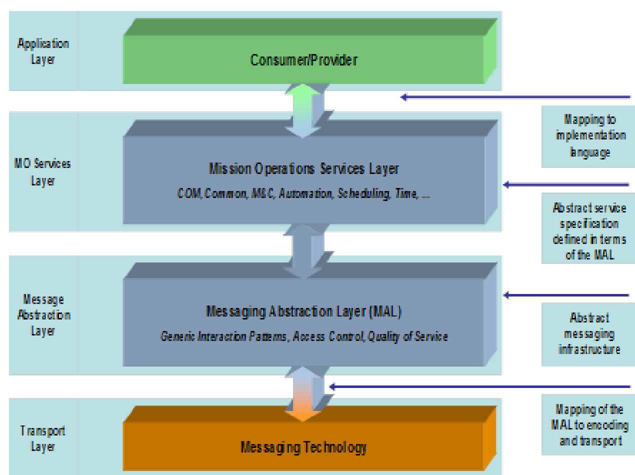

Figure 1. The Layered Architecture of the MO Framework

Boundaries of generic Planning Services

In preparation for specification of generic, reusable planning services in the domain of space mission operations the CCSDS MO working group has organized in 2012 two call-for-interest workshops in Europe and in the USA. The objective of these workshops has been:

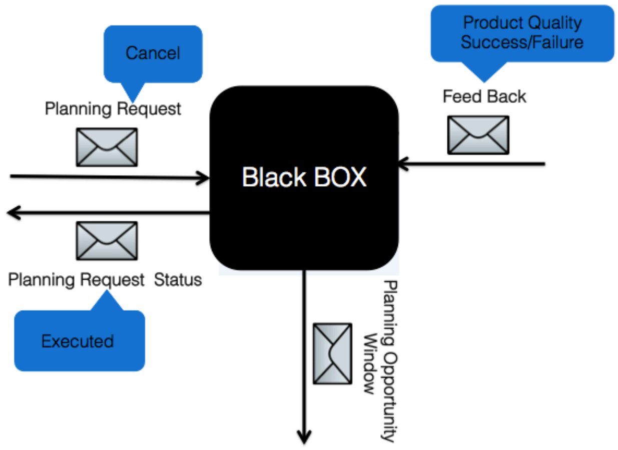

Figure 2. Boundaries of planning systems

1. To determine if the community has the interest and is of the opinion that standardized generic planning services can be specified, such that they can be reused across missions;
2. Identify the candidate areas in the space missions planning domain where generic planning services would have the highest potential for cross-mission standardization and reuse.

It has been the common shared view of the community that Planning problems in space domain can be very diverse. It is therefore not an easy task to define generic, reusable planning services. The size of the problem, the location of the planning logic (on the ground or on board the spacecraft or even a hybrid solution), the decision-making authority (AI decision making vs. human based conflict resolution or a so-called mixed initiative hybrid solution) and the distribution of the planning knowledge (centralized vs. decentralized planning) are factors which contribute to the diversification of space mission operation planning problems.

As a result the community has recommended in both workshops that standardisation work shall initially focus on the boundaries of a black-box planning system for space missions (Figure 2). Mainly concentrating on the aspects of handling the submission of "Planning Requests" and the related feedback information.

In this context, it was recognised by the community that standardising the terminology shall be the very first and an important step. It was the shared understanding of the community that standardising the services related to specification of the planning problem, the involved constraints and the specifics of the solvers would be a much more challenging attempt, which shall only be addressed as a second step.

Since this black box should provide services very general but at the same time detailed enough for being practically usable, we will try in the next Section to get more in detail about these services, in order to open a bit this black box.

When the Black Box becomes Grey

Finding the right balance between the level of abstraction applied to the black box and the assumptions which can be made on it versus the level of standardisation required for specifying generic boundary planning services will represent one of the main challenges of the standardization process.
The less is known about the implementation of the black box and its adopted planning/scheduling technique, the more difficult it becomes to specify a Planning Request and to interpret the resulting feedback (and the generated Plan). This is due to the fact that the information model adopted by each planning system, which includes the planning request, the resulting feedback, and the plan, is strongly dependent on the adopted planning technique. To give an example, the planning information model of a CSP scheduler or a Linear Planning solver compared to a goal driven planner or an HTN planner can be very different, consequently the input to each of these planning systems will contain a different set of information. Also the overall business process (the work flow of which the planning process is a part, e.g. how the planning of ground station allocation fits into the overall mission operations concept for a particular mission, what are the involved planning cycles and how is the conflict resolution process, etc.) plays an important role in the content of the exchanged information at the boundaries of the planning system, hence in the specification of the boundary planning services. Without making rigid assumptions in this regards, which must then be met by all compliant systems, it becomes quite difficult to agree on an abstract information model for planning requests and resulting feedback from the planning system and the eventual plan.

In order to frame the boundaries of any planning system, two extreme cases can be considered:

1. All relevant semantics for planning is defined within the black box (regardless of if planning data model is centralised or distributed within the black box) and is considered proprietary to the planning system (the black box). The boundary services such as for submitting planning requests would in this case contain only minimal syntactic information, basically by referring to pre-defined activity units (e.g. by providing the ID of the requested activity). All the related information for the requested

activity (e.g. the constraints, the goals, tasks and actions which are related to the activity, etc.) are not provided as part of the request. This approach relies however on agreeing on a set of common assumptions about availability and unambiguity of the relevant planning information for each “requestable” planning activity. In practice such assumptions prove often to be too optimistic so that a minimum set of semantics in the form of constraints and context parameters must be provided along with the reference to the name or ID of the pre-defined activity (e.g. temporal or global constraints or resource constraint parameters specific to each request). To give an example the resource consumption for an Action (e.g. take an Image) can often not be specified up-front as a fixed number and attached to the definition of the activity in the information model of the planner, since it is often dependent on other constraints in the system (e.g. the attitude of the S/C, the current state of a number of systems or the predecessor and successor activities in the plan). Also temporal and global constraints which put multiple requests in dependency with each other must often be assigned and submitted case by case to individual requests.

2. The interfaces (or API) of the black box allow the specification of all the information required by the planning system for carrying out the task of planning. This would however either require exact knowledge of the adopted planning technique and the corresponding proprietary information model or require agreeing on an standardised abstract planning information model. In order to specify generic boundary Planning services obviously the latter must be the case. A number of initiatives and languages in AI planning (such the PDDL language[1] for instance) are already attempting on specifying such a generic and abstract planning data model. The Black Box would in this case either translate the submitted planning information along with the Planning Request from the standard data model to its proprietary information model or be implemented in a way that it would work directly with the standardised information model.

The pros and cons of the two extremes are at hand, while the boundary planning services of the first approach would be much easier to specify from an standardisation point of view (much less to agree on, and only at syntactical level since the information model in question would be very small), it relies on a large number of assumptions which can impact significantly the implementation of the black boxes, hence again a source of debate at standardisation level. Also the usability of the resulting generic services in real world complex planning solutions can be questionable and should be demonstrated.
The second approach would require much more effort at standardisation level as many agreement must be reached to come up with a generic abstract planning information model which is independent from the actually adopted planning technique. The experience of other initiatives in this direction such as PDDL are also not very promising, as specialised dialects and extensions have proved to be necessary to address specific needs of certain planning techniques (e.g. expression of temporal and generic constraints).

The difficult task of a standardisation working group for planning services should find the right balance between a purely syntactical standard (which would lead necessarily to a pre-defined agreement among the developers of different software systems to entail the interoperability), and a more generic interface based on a semantic description of the data and processes (which conversely would entail very powerful, automatic interoperability among the systems but would require a wide agreement impossible to reach at the current stage).

A good middle point can be an agreement on the *syntax and semantic of a limited set of services* provided by the system to manipulate low level information. From our viewpoint this can be obtained looking at some previous works in the area of advanced planning solutions developed both at ESA and at other space agencies. Following sections will elaborate on how the experience gained from a set of planning initiatives at ESA can be used in contributing to identifying such a set of services.

Advanced Concepts for Future Mission Planning Design

The focus of this section is on how experiences and lessons learned at ESA on AI based advanced mission planning solutions could bring benefits and contribute to the process of mission planning services standardization. Services and APIs implemented in these systems can be considered as a starting point for a discussion about how designing practically usable planning systems services. Moreover, as proved from the series of IWPSS workshops, the space domain has been often a fertile field for the introduction of AI based advanced planning and scheduling technologies.
One of the aims of a standardization of Mission Planning Services is to allow rapid prototyping design and implements the concepts of re-using modules between different missions (to shorten software development time and cost, as well as the training of mission operation engineers).

In this context, a concept that has demonstrated to be very useful is the *Model-based approach*. This allows reusing of software modules across different missions because of the great flexibility of the symbolic representation of goals, constraints, logic, parameters to be optimized, and so on. As the system is not designed for achieving (possibly parameterized) goals in a given domain but for manipulating symbolic entities, the software deployment and test is substantially independent from the specific mission. However, it is worth highlighting how a great effort and amount of time in general might be necessary to both understand the domains and the problems, capturing all the specificity, and to create a model for these domains if proper symbolic constructs are not available for modeling. Moreover, as pointed out in the previous section, to design a generic API for a model-based planning system would require an agreement on (at least) a common language to specify models, problems, constraints and so on. And this is obviously quite far from the reality right now.

To cope with modeling issues, the cognitive distance between modeling primitives and the objects to be modeled has to be as small as possible. To have a real chance of getting to a general agreement on how planning information is specified and manipulated (pre-requisite to discuss a standard), a good starting point would be the greatest common denominator among all the different types of information in use in mission planning systems, i.e. *time tagged data*. Hence the problem of discussing a generic API for a planning system can be specified as a problem of specifying an API to manage sequences of time tagged data to achieve some objectives.

Among the proposal from the AI planning community, the closest one to the problem of managing time tagged data can be considered the so called *timeline-based paradigm* [3][4][5][6], where the planning problem is conceived as a problem of assigning values to sequences of ordered time intervals (the timelines). This approach to planning has proved to be particularly suitable for space applications, mainly because it is very close to the way problems and constraints are naturally represented in space applications. It is possible to state that historically, instead of having been injected into the control rooms to solve specific planning problems, this paradigm comes from the experience and the daily problems in the control rooms.
There are already software and platforms based on AI timeline planning in use at NASA and ESA (EUROPA [7], ASPEN [8], APSI [9], GOAC [10] among the others). These platform unfortunately do not use a standardized language or API, but follow a conceptually similar approach. The lack of a standardization in the languages and information lead to a consequent objective difficulty in spreading and re-using data, software solutions, and models among these platform. Nevertheless the similarities at the level of the services they provide can be analyzed to draw a possible starting point for a discussion on a standardization.

In fact all these platforms aim all at providing a set of similar services to implement planning and scheduling algorithms as well as complete “end-to-end” applications. A common key aspect is that they provide high-level support to (or to a subset of):

(1) Represent and Manage Domains and Timelines
(2) Model and Represent Domain Theories, Problems and Solutions
(3) Problem Solving with Timelines
(4) Timeline Validation and Verification
(5) Timeline Execution.

The services listed above can represent indeed a starting point for identifying classes of services that a generic Mission Planning system could provide.

Classes of Mission Planning services

When thinking to the possible set of services drawn from timeline-based systems, a possible break down to classify these services could be: (1) services to manipulate the basic entities that constitute a planning problem and its solution; (2) services to interact with the system, to post problem, control the solving process and provide feedback to the system; (3) services to evaluate and execute solutions.

The services to represent the basic entities that constitute a planning problem and its solution should entail the capability of representing and manage sequences of temporal tagged data (timelines), as well as events, activities and simple temporal and parameter relations among them. More in detail, a real standard should discuss at least how to represent:

- bounded or grounded time points,
- temporal constraints (e.g., minimal and maximal distance and duration constraints),
- overlapping and not overlapping constraints,
- integer, real, set, and generic parameters.
- finite and/or infinite states,
- consumable and renewable resources,
- functional parametric dependencies between data and data temporal tags.

Regarding activities and events representation, the standard should address the problem of how to represent tasks, activities and events, as well as their controllability properties (in order to execute the plans).

Regarding the services to interact with the system, they should generically allow the ability of representing problems, manage solving and optimization processes (with the possibility of escaping into arbitrarily coded timelines, constraint models and solving processes for maneuvers, power, mobility, thermal, etc.). In practice this can be reduced to the capability of stating and propagating constraints, querying the status of the timelines, detecting and reporting conflicts in the above constraints, extracting timelines from tasks, events, activities at various levels of flexibility (in terms of groundization of temporal information and data parameters), querying if a specific placement of an activity/task/event will violate the constraints, adding/retracting dynamically activities/tasks/events from timelines, or synchronizing timelines with events representing triggers or tasks/activities representing external inputs.

The services to use and manage solutions should permit the possibility of validating and verifying properties on timelines; evaluate solutions (quality, robustness, flexibility and so on); executing timelines, monitoring the process; store and visualize timelines in an efficient manner; intersect, synchronize, merge and split timelines; support problems at various level of granularity and temporal horizon (from LTP to STP, from science planning to operations, from on-ground activities to on-board activities, from EO missions to Deep Space missions, from sully manned till fully autonomous).

It is certainly too difficult (and probably not necessary) to standardize the way this information is manipulated inside the black box (i.e. what kind of process is applied to use the model), but having an agreement on a conceptual approach based on the semantic of the services listed above entails the possibility of finally standardizing even higher levels services close to the final users.

The concepts and services sketched in this section are certainly meant as a challenge for future mission operation frameworks. Nevertheless the challenge is not impossible, since most of the concepts and services have already been implemented, tested, and successfully used both at NASA and ESA. In the following section we sketch a possible architecture for the black box in Figure 2 inspired to some ESA advanced timeline-based planning systems that have implemented some concepts and services described so far. While on one side none of these systems have implemented all the services and often there are not standard interfaces to access them, on the other hand the current state of the art demonstrates that is feasible to take up this challenge.

A look into the grey box

In the past decades, the space domain has been a fertile area for efficiently applying automated planning and scheduling solutions based on "timeline synthesis". Many applications shown the benefits of AI techniques applied to space mission scenarios. This approach is in fact common to solid works in NASA's space domain such as, HSTS[3] and RAX-PS/EUROPA[7], ASPEN [8] as well as in some ESA applications and architectures, like MEXAR2[11], SKeyP[12], the APSI framework[9] and the GOAC system [10]. ESA has funded research and deployment on timeline based planning and scheduling in the past 10 years through external studies involving industrial partners and various European research institution. The approach followed was twofold. In "product driven" activities solutions for specific mission problems have been developed. Such an approach inevitably entails a huge implementation effort in terms of development: specification extraction, design, coding, and maintenance. Conversely, in "process driven" activities general purpose tools for facilitating the design and synthesis of new products have been developed. The general pursued idea is the one of improving the "process" of tool development, taking advantage of the state of the art AI planning and scheduling technology (see [13] for a more detailed discussion). This line of research and deployment lead to the design of a generic Timeline-based Representation Framework (TRF) which high-level architecture is sketched in Figure 3.

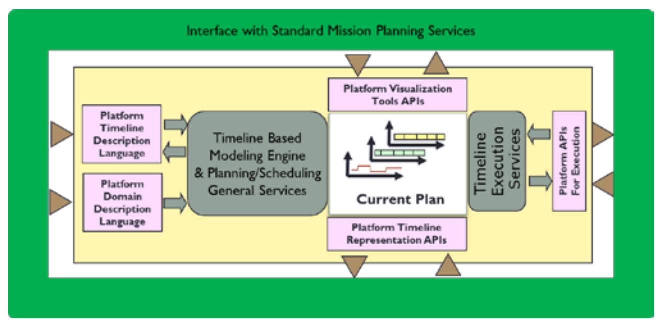

Figure 3. Architecture of a Timeline-Based P&S Framework

The architecture is designed around a database of timelines representing the current status of the plan/schedule. The solution database provides APIs for manipulating and visualizing timelines. Two engines provide (1) basic services for planning and scheduling and (2) basic services for monitoring timeline execution. Two languages provide the entry point for describing models and problems, while on the execution side an API provides access to the platform execution and monitoring services. Around the core, standard external services can use the basic functionalities for application design and implementation trough an adaptation layer (in green in the Figure). HCI services for instance can be built on top of the platform APIs for timeline visualization, specific solvers and V&V tools can access the timeline database through the APIs for manipulating timelines, Knowledge Engineering environments can be built for simplifying model design and management, executors can be interfaced with the APIs for execution and monitoring.

This planning and scheduling architecture definitely relies on its own languages for defining constraints, rules and problems, data formats for representing plans and schedules, and specific APIs to access the platform services. Nevertheless the classes of services identified in this paper can be easily mapped into internal services by means of the adaptation layer, drastically reducing the complexity of the standardization process. Hence the point of the standardization is not the specific syntax of a given language or a specific API for low level services, but what we need is an agreement on a few general and simple classes of services to manipulate timelines.

Conclusions

Following the discussion started in the frame of the CCSDS Spacecraft Monitoring & Control (SM&C) Working Group, in the paper we discussed how the state of the art in the area AI planning scheduling can provide useful examples and guidelines to the mission planning services standardization process.

The working group community has recommended to focus on the boundaries of the black-box planning system. This black box should provide services very general but at the same time detailed enough for being practically usable. Therefore the challenge is to find a balance between two opposite solutions: a purely syntactical standard (which would lead necessarily to a pre-defined agreement among the developers of different software systems to entail the interoperability), and a more generic interface based on a semantic description of the data and processes (which conversely would entail very powerful, automatic interoperability among the systems but would require a wide agreement impossible to reach at the current stage).

In the paper we have discussed as a realistic solution should be defining an agreement on the *syntax and semantic of a limited set of services* provided by the system to manipulate low level information. We also showed as a promising starting point would be to exploit common denominator among all the different types of information in use in mission planning systems, i.e. *time tagged data*. focusing on the specification of an API to manage sequences of time tagged data to achieve some objectives.

The different software and platforms based on AI timeline planning are in particular already showing similarities at the level of the provided services which can be considering in the standardization process.

In conclusion, the standardization process should look for a few general and simple classes of services to manipulate timelines. In this paper three of these classes have been identified: services to manipulated the basic planning entities, services to interact with the planning system, and services to evaluate and execute plans.

References

[1] Fox, M.; Long, D. (2003). "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains". Journal of Artificial Intelligence Research (JAIR) 20: 61–124.

[2] Chien, S., Johnston, M., Frank, J., Giuliano, M., Kavelaars, A., Lenzen, C., Policella, N. A Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations. Proceedings of the 12th International Conference on Space Operations, SpaceOps 2012.

[3] N. Muscettola, “HSTS: Integrating Planning and Scheduling”, In Zweben, M. and Fox, M.S., editor, Intelligent Scheduling, Morgan Kauffmann, 1994.

[4] J. Frank and A. Jonsson, “Constraint Based Attribute and Interval Planning”, Constraints, 8(4):339–364, 2003.

[5] Fratini, S.; Pecora, F.; and Cesta, A. 2008. Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. *Archives of Control Sciences* 18(2):231–271.

[6] Verfaillie, G.; Pralet, C.; and Lematre, M. 2010. How to model planning and scheduling problems using constraint networks on timelines. *Knowledge Eng. Review* 25(3):319–336.

[7] Europa, “Europa Software Distribution Web Site”. http://code.google.com/p/europa-pso/wiki/EuropaWiki, 2008.

[8] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, and D. Tran, “ASPEN - Automated Planning and Scheduling for Space Mission Operations”, In Proceedings of SpaceOps’00, 2000.

[9] Fratini, S. and Cesta, A. “The APSI Framework: A Platform for Timeline Synthesis”. In Proceedings of the 1st Workshops on Planning and Scheduling with Timelines, Atibaia, Brazil, 2012.

[10] Fratini, S., Cesta, A.,De Benedictis, R., Orlandini, A. and Rasconi, R. “APSI-Based Deliberation in Goal Oriented Autonomous Controllers”. In Proceedings of the 11th Symposium on Advanced Space Technologies in Robotics and Automation, ESA-ESTEC, April 12-14, 2011.

[11] Cesta, A., Cortellessa, G., Fratini, S., Oddi, A. and Policella. N. “An innovative product for space mission planning: an a posteriori evaluation.” In Proceedings of International Conference on Automated Planning and Scheduling, 2007, Providence, RI.

[12] Policella, N., Oliveira, H., Siili, T. Managing SOHO's Keyhole Periods: Problem Definition and Solving Model. In Proceedings of the International Workshop on Planning and Scheduling for Space, IWPSS-09, 2009, Pasadena, California, USA.

[13] Cesta, A., Cortellessa, G., Fratini, S., and Oddi, A., Bernardi, G., Deploying Interactive Mission Planning Tools – Experiences and Lessons Learned. Journal of Advanced Computational Intelligence and Intelligent Informatics. Vol.15 No.8, pp. 1149-1158, 2011.