SpaceOps-2025, ID # 16

ESTRACK Deep Space Cross support using CCSDS Service Management

Martin UNAL[a], Matthew REED[b], Szymon STUGLIK[c], Colin Haddow[d]

[a] ESA, European Space Operations Center, Darmstadt, 64293, Germany, martin.unal@esa.int
[b] SERCO, European Space Operations Center, Darmstadt, 64293, Germany, matthew.reed@ext.esa.int
[c] ESA, European Space Operations Center, Darmstadt, 64293, Germany, Szymon.Stuglik@esa.int
[d] Scotty Consulting, Darmstadt, 64293, Germany, Scotty.Consulting@Scotty-Enterprizes.com

Abstract

The latest ESTRACK Deep Space Terminal loading forecast shows that the success of ESA's scientific and robotic exploration missions is exhausting ESTRACK Deep Space network capacity. Consequently, access to external resources is a necessity to overcome saturation of demand at peak periods. As institutional partners face a similar situation, both in terms of success and demand, when a high level on one network corresponds to a low level on the other, collaboration can go both ways.

Furthermore, commercial players and suppliers are investing in the deep space sector. New entrants are interested in using ESTRACK as a broker, whether to sell or access resources. This means that ESTRACK's service management implementation must be able to act as a supplier, while also acting as a user to request external resources on behalf of third-party missions.

Over the past decade, ESTRACK has established CCSDS-based service management interfaces with numerous partners. With the multiplicity of interfaces, standardisation is the Holy Grail to quest for. As the need for standardisation is faster than standardisation, draft standards have been implemented. This was a unique opportunity to prototype and test CCSDS service management concepts.

One of the challenges was to maintain existing interfaces while establishing new ones, without altering the overall workflow of the ESTRACK scheduling office. It is essential to keep the complexity of the workflow below a certain threshold in order to avoid excessive training efforts and operator errors.

The result was the development of a multi-purpose scheduling portal (EMSP) that enables the ESTRACK planning office to operate with internal and external stakeholders, as user or supplier. It has enabled us to put into practice various types of interfaces and processes with multiple partners, and to identify commonalities, similarities and differences between them.

This document aims to present what has been achieved and learned in our attempt to standardize CCSDS service management. Based on these findings, it proposes recommendation on potential ways forward.

Keywords: ESTRACK CCSDS Cross-Support Service Management

Acronyms/Abbreviations

Delta Differential One-Way Ranging (DDOR), European Space Agency (ESA), ESA tracking Network (ESTRACK), ESTRACK Management System web Portal (EMSP), ESTRACK Planning System (EPS), ESTRACK Scheduling Office (ESO), Consultative Committee for Space Data Systems (CCSDS), Launch and Early Operation Phase (LEOP), Mission Planning System (MPS), Multiple Spacecraft per Aperture (MSPA), Simple Schedule Format (SSF), Service Management Utilization Request Formats (SMURF), Service Package Data Formats (SPDF). Extensible Markup Language (XML), XML Schema (XSD), Extensible stylesheet language (XSL)

1. Introduction

In the last two decades the number of ESA flying spacecraft has tripled, while ESTRACK capacity has remained stable. As consequence, the usage of third-party networks is essential to fulfil mission requirements.

On the other direction, ESTRACK is one of the few networks with Deep Space capacity. Numerous collaborations with other partners, both commercial and state-run, have been established. In particular the growing Lunar exploration program requires performance levels that are not available yet on the market.

ESTRACK Scheduling Office had to developed tools and procedures to interface with all parties, both as user and provider. Currently ESTRACK is interfacing with 6 providers and serving 5 external partners. This has highlighted the need for booking and scheduling service management standardisation. Whenever possible, the CCSDS Service Management standard [1] has been used for ESTRACK interfaces.

This document analyses the results of this exercise. It turns out that even if standardisation is not (yet) in place, there is a significant level of similarity between the practices.

2. Inside the box – ESTRACK Scheduling Office tools & processes

ESTRACK planning is created in multiple ways. For most ESTRACK users, an initial booking is automatically generated by the ESTRACK Planning System (EPS). The EPS is a smart solver able to implement and optimise network allocation based on mission agreements with Standing Orders and constraints.

At the start of each project, a mission agreement is established. It lists the services available and the information exchange interface for their implementation.

It may also include of one-to-many standing order describing services planning requirements and constraints. These standing orders are translated into a machine-readable format within the EPS configuration.

Based on this input, ESTRACK Scheduling Office uses the EPS solver to process the combination of all users' standing order. The tool generates a consistent booking fulfilling all user requirements. The output is distributed to each individual user, internal and external. This long-term planning horizon covers typically 4 weeks for the low earth orbit network, and up to 12 months for Deep Space assets.

During a second phase, users can request modifications to this initial baseline, to either, add, modify or delete bookings. They usually include the outcome of their iteration with science or navigation teams.

It happens also that a support is "spontaneously proposed" by providers. Either based on a similar standing order planning or agreed via another channel (e.g. email).

The ESTRACK Management System Portal is the tool used to interface with all external systems, either as user or provider. It can handle these multiple types of inputs, host negotiations, and support ESO in the resolution of discrepancies.

A change request, irrelevant of its origin, is checked against:

1) Validity: format, ID and privilege are correct.
2) Conflict: request is blocked until all conflicts are resolved.
3) Technicality. Service is available to the user at the requested terminal, the visibility is compliant with the activity.
4) (For external provider) Availability. The request is forwarded to the provider.

Valid requests are going through the states described in Figure 1.

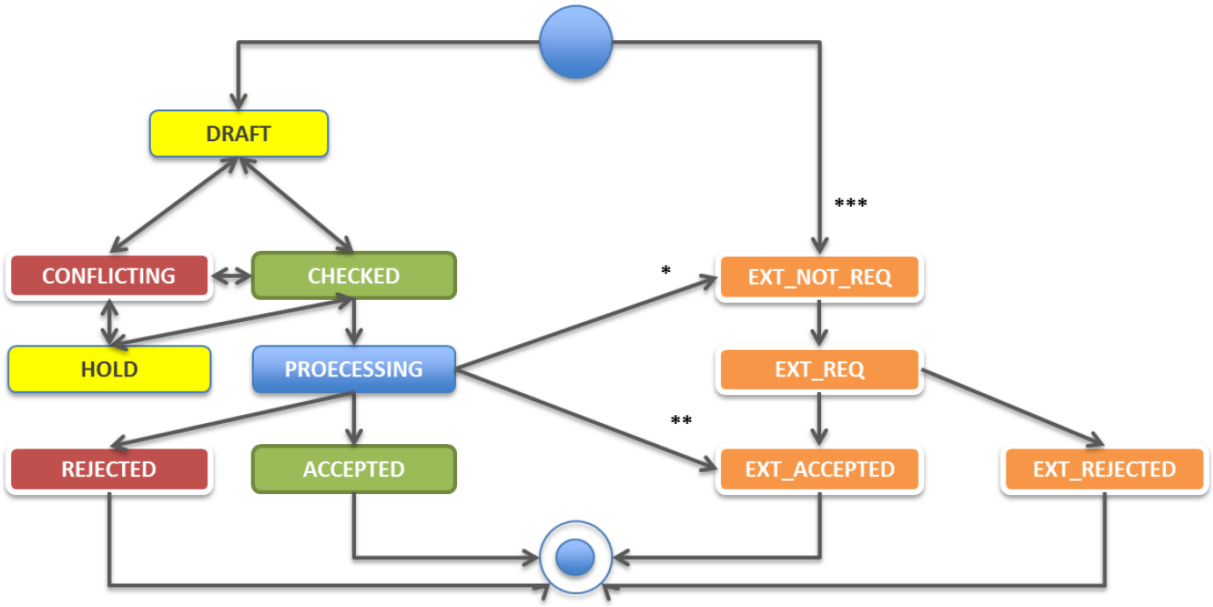

Figure 1 EMSP request state machine [5]

* If ESA mission requested support on external ground station, i.e. origin network is ESTRACK, but the ground station is external.

** If external provider suggested the support, i.e. origin network is not ESTRACK, and the ground station is external.

*** support created by planning or other means, no support update attached, and the ground station is external.

ESTRACK maintains individual interfaces with all users and providers. But partners, user or provider, are shielded from each other's complexity. They only need to take care of their own interface. How requests are coming and forwarded between them is an ESTRACK problem.

In addition, as most support are automatically planned by EPS, it may happen that a user get its full support requirement implemented without the need to issue a single request. The overarching goal is to make the interface as simple as possible for to the user, and to limit the volume of request to the bare minimum. This is typically the case for low earth orbit and astronomical observatory missions

3. Interfacing the world

3.1. Overall process

EMSP manages supports and requests. This looks trivial in a self-standing application, but tracking changes requested or unexpected between multiple parties is not.

Let start easy with a user requesting a support. User demand, Provider accept. Support is confirmed. So far, so good.

Now, the support gets modified by the provider without prior notification. This can happen for various reasons. Due to LEOP, maintenance or failure, it is common to have support shortened or moved to another terminal. But this requires a channel of communication in the provider to user direction.

Many common situations require a push from the provider to the user. A request for modification is pending due to conflict. It is neither accepted nor rejected until a decision is taken. For Deep Space missions, this type of negotiation can last months.

It will happen that the parameters defining your support, such as service type, parameter value, IDs change in the provider database. A request for support cancellation is rejected. Has the support been already cancelled or is it still present in the provider databases?

Whatever it is for good or bad reason, discrepancies must be detected and handled.

The SSF data format [2] has proven to be a good way to close the loop in all circumstances. In ESTRACK implementation it has been used to solve all the point presented above. Implementation solely based on immediate confirmation will not cover all cases.

On top of technical issues, cross support has to bridge the gap between different work cultures and processes. ESTRACK has put a huge effort in the EMSP operators' interface to guide ESO in understanding situations and help schedulers to handle discrepancies [4].

Over the last two decades, ESTRACK has developed two generations of inter-network synchronisation applications. Even with full experience of the topic, It took two years each time to have it correct. Due to the multiple months range covered by ESTRACK planning and the complexity of the mission profiles served, it has to be expected the unexpected to happen.

As a matter of fact, both external users and providers shall be expected to do whatever they want, or to be fair whatever they need to do. The task is to go with it, not against.

Launches will be delayed, missions will adapt to adverse situations, databases will crash, and unique IDs will be reset. EMSP has been designed to survive all of this and straighten things out in the most effective way.

3.2. Interfacing with Users

ESTRACK users are very different in nature. They range from missions designed in the previous century to Moon landing start-ups.

So far only few mission planning systems can handle all the concepts required to exchange with a service provider:

- Generate time ranged Standing Order Refinement (e.g. to decrease the number of passes during weekends)
- Ingest allocated supports
- Check validity of the plan
- Edit and export change requests
- Close the loop (are my requests implemented? Was anything changed without my consent?)

Mission Planning Systems will only support the functionalities required for their own projects. Some type of interactions will be essential to some mission concepts, and irrelevant to others.

To allow all types of users to operate, the EMSP offers all the functionalities, both to human and machine. Users can consult the web portal or interface system to system. The EMSP supports REST and file-based exchange.

Both manual intervention and machine generated requests are processed in the exact same way. It is possible to connect the web portal to edit a request created in the system via a file.

A good example of mission using all functionalities has been the GAIA mission.

GAIA objective was to measure stars positions viewed from a Lagrange-2 halo orbit. Depending on the density of stars in the field of view, the daily downlink requirement varied between 6 to 24 hours. GAIA MPS has generated standing order requirements to specify tracking time requirement for each single day of the entire 10 years mission lifetime. This powerful interface has played a key role in GAIA operations planning.

On the other end, this functionality is unnecessary for a mission with a stable demand profile. To keep things simple, ESTRACK policy is to expose user only to the strictly necessary level of complexity. As long as requirements are stable, the EPS configuration can host a default behaviour definition on behalf of a user. This allows to support complex requirements without exposing unnecessary burden to users.

3.3. Interfacing with provider

ESTRACK acts as a service broker. It interfaces with external providers on behalf of all its users.

The goal is now to interface with all of them with a single simple and efficient tool. It is not easy, but far from impossible. It turns out that behind the legacy implementation and the current practice, the nature of the problem has not changed much. The commonalities are discussed further in the next chapter.

When the provider is already an established entity, it has its legacy interface in place. It is up to the user to adapt.

On the other hand, newcomers or private network going on the market will be looking for existing standards to build their initial interface implementation. This second category has showed interest in the CCSDS Service Management publications.

The approach used by ESTRACK is simple and can be applicable to any existing implementation.

Protect your internal implementation by decoupling internal data structures from external interfaces to avoid ripple through effects. It is a mistake to try to stick close to a anybody else design, even if it is a published standard. Technology evolves, new mission profiles are more complex (e.g. multiple uplink). Support scenarios are going for new challenges such as DDOR and Multi Spacecraft per Aperture.

External changes will impact your interface; internal change will impact your design and implementations must adapt. It is important to keep the freedom to update and adapt your own implementation at your own pace. Furthermore, expect other parties to apply a similar approach. Even limited to your organisation, as soon as you have more than one party to interface with, it is extremely difficult to have all parties adapting at the same time. In the context of cross support, it is unrealistic to assume that a single interface can be used with all parties at any given time, not even considering the necessary evolutionary upgrade over time.

To face these challenges, the EMPS interface design ensures that:

- All point-to-point interfaces are independent. Any single interface can be created, maintained or destroyed without impacting the core design or any other interfaces.
- The core interface is fully shielded from the outside. Adaptation to the external world is done in the shielding layer.

Figure 2 - XSL shielding – describes the concept. All exchanges are done via files in XML format. XML has proven very efficient to package information between multiple parties. It is agnostic to implementation, and it makes it easier for all parties to validate data format against XML Schema. The conversion from XML to another format such as JSON or ASCII is also supported.

XML transforms is used to map third party information to a stable core implementation. These User-Provider translators shield both parties from each other's changes.

With the push of standardisation, the number of translator types will be reduced, but until then, it leaves the capacity to support different generation of interfaces at the same time.

As the transforms are independent from the core EMSP software, they can even be edited without halting the application.

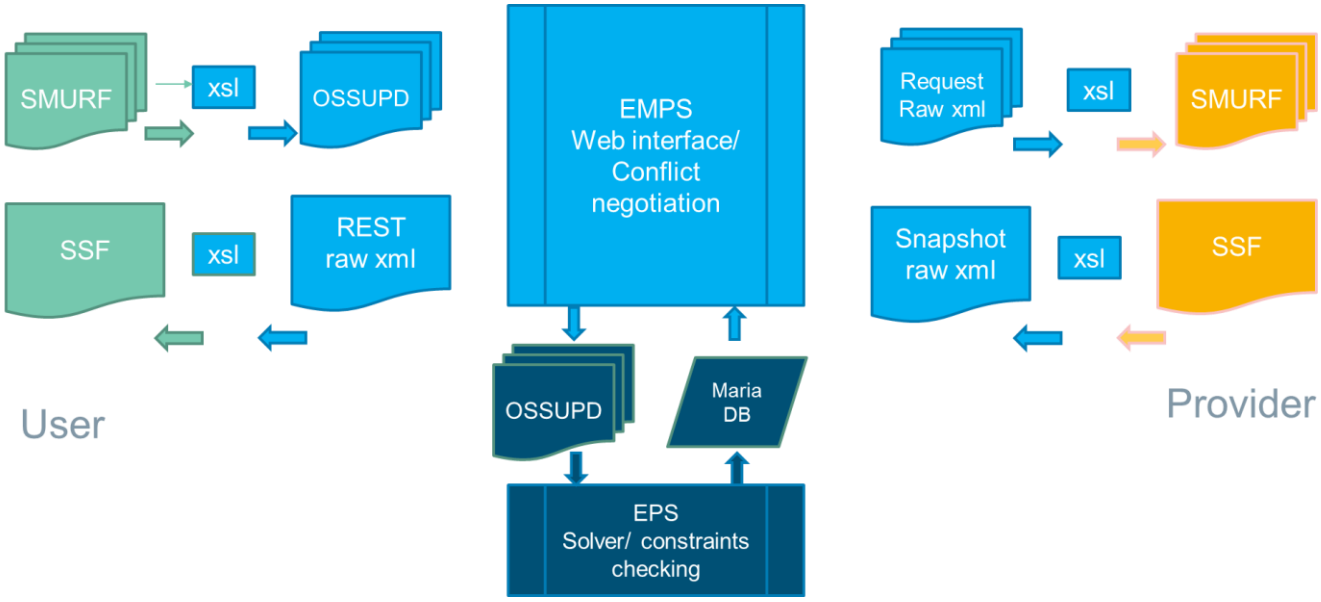

Figure 2 - XSL shielding

Mapping addresses following concepts:

- The organisation in service layer such as Telemetry or Telecommand with independent parameters
- The capacity to store user or provider parameters. And to feed them back as required.
- Handling of user and provider unique ID
- Model requests as state machine for external provider acceptance cycle
- Advanced matching protocol to keep the internal and external databases consistent.
- Operator friendly interface: Availability of information in a single view, possibility to halt and restart process at will.

4. Cross Support Interface Commonality

4.1. Data

Through all the interfaces ESTRACK has established with third party users and providers, these 5 items are always present:

- Username, typically the spacecraft codename. (SC)
- Resource name, either the site or the terminal. (GS)
- Start of Support aka Beginning of Track (BOT)
- End of Support aka End of Track (EOT)
- Service Definition. A pre-agreed ASCII string, or a free text comment. (Service/Comment)

To serve all cases and protocol it is sufficient to support in addition:

- Service parameters (e.g. telemetry rates or coding)
- User ID
- Provider ID

With this minimal set, ESTRACK has been able to establish operational interfaces, both as user and provider, with 6 agencies and 5 commercial entities.

4.2. Format

The differentiation starts with the format. This is the topic for which the CCSDS SM standard is the most beneficial.

XML usage is widespread, but obviously the mapping differs for each party. This easily solved by XSL transform.

Since 2017, ESTRACK has started to use the published SSF and the draft SMURF (white book0.06 at the time [3]) for all new interface design. Since then, this has simplified all implementation and procedures.

4.3. Process

In short, two protocols exist to request support: replace all (bulk request over a time range), or single support request driven.

The legacy one is "replace all". The request file contains a time range information, usually in the header or filename, and a list of support to be implemented. Any support previously requested contained in the time range is deleted.

This approach is adapted to low complexity exchanges. As long as the requests concern only a single spacecraft over a limited range it does the job.

In the context of a Deep Space mission planning covering twelve months and multiple missions, this is not practical.

The SMURF protocol implements add/modify/delete individual support requests. It allows changes tracking on a per request basis, which is convenient when less than 1% of a yearly plan need to be modified.

The drawback is the increase in complexity of the booking system. Storing and managing requests linked to each support come with a cost. As the request state machine goes into third party system, anomalous behaviour root cause can be on either side making troubleshooting challenging.

The robustness of the synchronisation logic is essential to make this approach beneficial.

4.4. Service Definition

This is the least standardized part of all. The notion of selecting a support type is common to all, but how to organise and name them is provider specific.

The most common practice is to establish a list of keywords to be used by both parties to refer to specific configurations. A common practice is to use the band (e,g, XBand, SBand) or a keyword differentiating a linited number of configurations (e.g. Mode1, Mode2).

For more complex activities such DDOR, MSPA or mission with elaborated space link, the usage of additional parameters is a must.

Some good practices are beneficial when establishing the bilateral user-provider agreement.

Even if the interface definition is a bilateral effort, the support concept must match existing implementation. It is not recommended to inject radical changes into the provider design for every user.

Another important point not to forget is the human element in the system. When automation collapses, or must be bypassed for emergency, it shall be possible for a network operator to implement the service from instruction exchanged over a voice loop. Even if you push automation to the utmost point, prepare for the day it breaks, you prepare to be doomed.

The key message to properly define service in cross support is:

- Keep it simple. Only consider the minimal set of information absolutely required,
- Never allow for redundant, potentially contradicting, information (they will contradict),
- Keep in mind the limit of the target running the automation (don't break a running system)
- Respect operators. Team working 24/7 have limited access to background information. (target for efficiency, not for scapegoat)

Any of these recommendations you disregard will teach you something…

4.5. Support execution

ESTRACK network operations are fully automated. Each service type is linked to a scheduling template listing all automated tasks to be run to the target system.

Based on the service request and the parameter provided, the templates are used to instantiate each planned allocation into an executable timeline.

For complex activity pattern, ESTRACK has published a catalogue of "user scheduled events" which can be requested withing an allocated support.

The events are basic actions listed in absolute time (e.g. start/stop uplink. Start stop recording). They are commonly used for mission profile which includes handling of planetary occultation or activity requiring changes to the space link properties at arbitrary absolute time.

The introduction of this event catalogue has been very successful and is now a de facto standard applicable to all ESTRACK user and provider.

For external support, 3rd party providers are presented with the same level of information as the one used to create ESTRACK automation (service type, parameters and events). It is up to them to implement the service. As long as the service is provided, it doesn't matter if the tasks are run by automation or executed manually by an operator.

5. Lessons Learned

5.1. Things that worked well

It is recommended to minimize the information that shall be exchanged routinely. It makes implementation simpler and easier to maintain.

Leave it to classical documentation to describe complex configuration details and operation concepts. The parameters defined in the interface shall only allow access to tested configuration. The implementation shall prevent users from mistakes.

The usage of centralized automatic constraint-based planning, either by the user or provider, reduces significantly the effort on both sides. It provides ESTRACK user with a booking matching its requirement without effort and allows optimisation of the network load.

REST interface is very convenient for user to poll data and scripting. However, file-based interface is more robust than REST only and goes both ways. Keep in mind that cybersecurity rules are far from being standard or even stable. Some communications port can be blocked with little to no prior notice, invalidating operational channel and killing live protocol such as REST. However there will always be a way to transfer a file between two systems if needs be.

The capacity to work with files, in parallel to REST, has allowed ESTRACK to continue operating despite adversity.

XML is well suited for interface definition between multiple parties. It enables local implementation validation against XSD, drastically reducing testing effort. In theory, any structured format, such as JSON, with transformation capability can do.

5.2. It looked like a good idea…

Some attempts have been made to standardise all possible aspects of any mission configurations. The result has been a huge catalogue listing all possible fields that could describe any possible services. To say the least, this approach is ambitious, and nearly impossible to maintain up to date in line with latest technological evolution. It is not clear if this idea will make it through. It may be more successful within the community of software defined radio users where configuration file are more stable than for hardware based solution.

Concerning service description, the CCSDS proposes a list of operational services using functional levels such as: Online Telemetry, Uplink, Ranging, DDOR, Offline Telemetry….

This looked very nice on paper but turned out to be awkward to use. To describe what the user needs with these concepts, each sub-service (e.g. telemetry, telecommand) would have to be layered and booked individually for each single support. An extra effort with no added value.

It is more convenient to agree on a meaningful service name packing everything. E.g. a "TTC" service includes telemetry, uplink, ranging and calibration with agreed timings. That has been so far the best trade-off among cost, complexity and robustness.

An interesting point is the individual request confirmation concept. This has been rated as essential during the EPS design phase… but never used. Due to the nature of the ESTRACK scheduling process, requests can be pending in the system for days to allow negotiations between users. The same applies to ESTRACK providers. It usually takes at least 24h between the request and the reply. The best way to close the loop in this situation is to rely on the SSF reply listing everything the provider has agreed to support, keeping in mind that a missing support may be pending a decision. To avoid maintaining two types of confirmation process (supports list & individual request confirmation), it has been decided for ESTRACKto only rely on the pass listed in the SSF reply for confirmation process.

Individual request confirmation concept (aka CCSDS SPDF) is still useful in a different context. ESTRACK is interfacing with only one provider supporting this concept. The drawback is that there is no room for negotiation, and when the provider changes its plan (e.g. Launch scrub or antenna failure) another protocol is needed to push back the information to the user (e.g. SSF).

In the end, the implementation must be able to close the loop by importing the providers actual support list (SSF). Individual request confirmation is nice to have.

In ESTRACK cases no user has shown interest so far in supporting two different protocols for similar purpose. They only rely on the published pass listing pushed to them.

6. Way forward

6.1. Service definition

The approach considering having a standard as a prerequisite to any implementation has not been successful so far. And it is unlikely that existing systems will align overnight and all at the same time. However, the need for interoperability is present.

It remains important for all to converge whenever it is possible. ESTRACK Scheduling Office has proven it is possible to do it in a step-by-step approach.

Request of service can be standardised today for what concerns the format, basic parameters, and service definition parameters.

Services themselves are not standard and will probably not be in any near future. They are mission specific, linked to the technology and the operational concept. This part is better covered by case-by-case bilateral agreement.

Authorized parameter values shall be restricted to the essential: Only what needs to change and is easy to communicate over a voice loop. Service must be tested and validated for each possible parameter before being used. It shall not be possible to change on the fly a parameter accessing a configuration not previously tested.

Standardisation will help. But not everything needs to be standard. In some cases, it is beneficial to tailor the interface details as close as possible to the mission specificity.

6.2. Planning

An attempt has been made to consider constraint-based request in the CCSDS books (e.g. Give me a track of 2 to 4 hours tomorrow between 10h00 and 20h00 UTC). The SMURF book defines a set of enhanced constraints for this purpose.

The standardisation of standing orders looks simple (give me one pass of at least 6 min uplink per day) but is in reality extremely complex. To be of any use, a standing order definition must be able to express preference, avoidance and constraint. The logic behind constraint-based planning requires a level of expertise that make it beyond the reach of regular users. A simpler option is to allow users to query network free time as input to its own booking system.

The functionality is very powerful, but today it is better off as a proprietary solution rather than a generic service. The ESTRACK Planning System supports the definition of standing orders to express extremely complex scenarios. But its complexity makes it inadequate for a standard interface. Even as an ESTRACK internal product, each standing order implementation is highly tuned to user specificities. This is better described using classical documentation (e.g. Mission Agreement or Scheduling ICD) where background information and graphics can be included.

In addition, the prime purpose of standing order planning being the optimisation of network allocation, it is anyway possible only in a centralized system. EPS accounts published 3rd party network availability into consideration for its global optimisation.

What is currently missing is the possibility to express flexible requests for network not able or willing to publish their availability. The possibility to express earliest track start time, latest track end time and minimum and maximum

accepted track duration is proposed in the SMURF book but is not supported by any implementation so far. This would allow to complete a globally optimised solutions in a single iteration.

7. Conclusion

Legacy space players with integrated resources have reached network saturation. Most of the newcomers in space operations are either satellite operators looking for network services, or network operators looking for customers. As consequence the demand for cross support is expanding.

At this time, major providers are pushing to impose their individual solution, mainly based on proprietary REST API.

In parallel, the CubeSat community is building bricks of interoperability in an uncoordinated but prolific effort.

These two last approaches are very targeted to answer their creators' specific needs. They will probably not scale up easily to higher level of complexity such as Deep Space operations.

ESTRACK, with other partners, have tried to test and promote the CCSDS SM proposals into operations. The CCSDS request (SMURF) and reply (SSF) support all the operational concepts (and much more). They cover the essential dataset and a stable format to package and exchange information.

CCSDS Service Management working group also explores aspects of cross support beyond support request. Some related topics would benefit from standardisation to avoid recurring implementation effort. For example, harmonizing Service Management Accounting (SMAC) to simplify contractual discussion. E.g. What shall be in a pass report? How shall be defined support success and failure criteria?

More ambitious topics are also discussed such as the Service Agreement and Configuration Profile Data Format (SACP) and the Service Management Automation Support Handling (SMASH). The objective is to formalize all aspects defining mission operation in a machine-readable format. The challenge is that reality is moving much faster than standardisation. Space Agencies purpose is to try new technology and new concept to prepare future standards. Innovation works better within a trial-and-error approach than within predefined standard.

However, for CubeSat with recurrent hardware operated by many different parties, it makes sense to converge on standardisation already today.

The current problem is the lack of a common forum to work on standardisation. Only big institutional players are currently represented in standardisation bodies such as the CCSDS. Universities operating small spacecrafts are unlikely to have the resources to work on standardisation.
To be complete major commercial players are more interested in imposing their solution than to abide to a standard.

Between published and de facto standards, it is unclear which one will prevail in the next decade. But having one will be beneficial to all.

Acknowledgements

A special thanks to Vladimir Nazarov, Yaroslav Markov and Fedor Korotkov who cooked with us the first CCSDS SM interface for ExoMars project.

References

[1] "Space Communication Cross Support— Service Management—Service Specification", Recommended Standard, CCSDS 910.11-B-1, Blue Book Issue 1, August 2009

[2] "Simple Schedule Format Specification", Draft Recommended Standard, CCSDS 902.1-B-1, Blue, May 2018.

[3] "Service Management Utilization Request Formats", Proposed Draft Recommended Standard, CCSDS 902.9-W-0.06, White Book Draft July 2018.

[4] H. Dreihahn, C. Haddow and M. Unal, "CCSDS compliant Ground Network Schedule Interfaces from an ESA perspective", *Spaceops*, Daejeon, Republic of Korea, 2016.

[5] "EMS Portal - Software User Manual", DHSO-ESOC-EMSP-SUM-1001-HSO-GIB.