

SpaceOps-2023, ID # 201

A new approach to inter-agency station scheduling at GSOC

Edoardo Barbieri^{a*}, Marcin Gnat^b, Pierre-Alexis Lagadrilliere^c, Wolfgang Frase^d

^a German Space Operations Center, LSE Space GmbH, Oberpfaffenhofen, 82234 Wessling, Germany, edoardo.barbieri@linspace.de

^b Communication and Ground Stations, German Space Operations Center, German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Wessling, Germany, marcin.gnat@dlr.de

^c Communication and Ground Stations, German Space Operations Center, German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Wessling, Germany, pierre.lagadrilliere@dlr.de

^d German Space Operations Center, Freelancer at Hays AG, Oberpfaffenhofen, 82234 Wessling, Germany, wolfgang.frase@dlr.de

* Corresponding Author

Abstract

Cross support to space operations is facing increasingly more complex challenges: a rising number of satellites and space mission users and consequently, an increasing number of mission requirements. Until 2010, an established space agency like DLR was servicing a small number of costly missions, providing ground station and cross support services to a handful of projects and ground stations. Nowadays, especially thanks to the developments in nanosatellite technology, the same agency is facing an increasing number of smaller missions and potential customers each having specific requirements and needs. To supply such demand, space agencies have initiated a process of standardization in the management of Cross Support Services, providing recommended standards whose goal is to facilitate the process of automation of such services. Among these recommended standards there is a set of upcoming recommendations, which focus on the specific task of station booking. This task is being supported mainly by the Service Management Utilization Request Format and the Service Package Data Format. Those can be described as structured data entities whose content is used to request and convey the cross-support information between mission users and antenna providers, which is necessary to execute a requested contact. The future use of these new standards and new station booking software will have a large impact on operations of the future.

In this paper we will first provide a detailed introduction to the specification of the Service Package Request and the Service Package Data Formats with focus on the new concepts introduced by the standard in the context of cross support services. Then we will show a real implementation of the standard itself, as adopted in the new scheduling system that is being developed at GSOC. Furthermore, we will analyze the impact from the introduction of such a standard for the spacecraft operation community at GSOC, and finally explore the future possibilities that the standard offers to operations at space agencies. The aim of this paper is to give an overview of the Service Package Request and Service Package data specifications and a key to understanding the newly recommended CCSDS standard for Cross Support Service Management, with the hope that more and more agencies and companies will adopt it in the future.

Keywords: CCSDS-CSS, Service Package, Service Management

Acronyms/Abbreviations

Consultative Committee for Space Data Systems (CCSDS), Deutsches Zentrum für Luft- und Raumfahrt (DLR), Cross Support Service Management (CSSM), Cross Support Services (CSS), Cross Support Service System (CSSS), Service Management (SM), Ground Station Scheduling Next Generation (GSSNG), German Space Operations Center (GSOC), Interface Control Document (ICD), Representational State Transfer (REST), Space Link Extension (SLE), Service Management Utilization Request Formats (SMURF), Extensible Markup Language (XML).

1. Introduction

In the past twenty years the Service Management working group of the Consultative Committee for Space Data Systems (CCSDS) Cross Support Services Area has focused on the development of a set of standards for the interoperability of those services that are functional to space missions, but not directly involved in real-time operations. As described in [1], the Service Management standard suite provides those data structures and formats that enable the collection and exchange of all information necessary to agree, configure, schedule, prepare, access, and report on Space Communication Cross Support (SCCS) services. Alongside the publication of the standards, the Cross Support Services - Service Management (CSS-SM) working group has created schema files for the direct implementation of the data structures described in the SM catalogue. Such schemas are an invaluable source for software developers and engineers who will eventually adopt such standards in their systems. In the past years at DLR/GSOC a small group of developers and Ground Data Systems engineers and managers has been working on a scheduling system whose aim is, among others, to support the Service Management data formats and have them integrated into the routine scheduling operations. This has been possible mainly thanks to the provision of the schema files by the CSS-SM working group.

The adoption of CSS-SM recommended standards has meant for the scheduling team a paradigm shift in the process of booking and scheduling ground station resources for space missions. Before CCSDS, the booking of resources (as in apertures, antennas, and terminals) was done on a per-contact basis. The scheduling office at GSOC would collect the scheduling requests from various space missions, where every requested resource allocation was connected to a specific spacecraft-station contact, and where “contact” is defined as the period of time in which a spacecraft surpasses a certain elevation threshold with respect to a specific ground station, and as a consequence of this phenomenon, a communication link can be established. Scheduling requests were therefore treated as “contact requests”, where in most cases the only way for a service user to uniquely identify a request was by looking at the contact start and stop times or (when present) at the orbit number of the spacecraft’s trajectory. It was clear that this mechanism could hold as long as there are humans to supervise the whole scheduling process when dealing with multiple space missions and multiple ground stations. For example, a clear case in which this process breaks, occurs when the start and stop times of the contact are intended to change. This could happen for example due to orbital maneuvers at the spacecraft or due to the recalculation of a more precise trajectory. When such a case occurs, it becomes complicated for an allegedly automated system to propagate the information consistently. In fact, without human interaction, there shall be rules to unequivocally determine what happens to the “contact request” if the requested times have changed. But to do so both the requestor and provider need to agree on a clear way to uniquely identify such “contact requests”, so that they can act upon them, by for example changing the start and stop times. What appears to be a trivial exercise when performed by humans, becomes a complicated puzzle when one tries to implement rules for the automatic handling of the “contact requests”, especially when a multitude of ground station service providers, each with their views on the matter, are involved.

To overcome this puzzle, CCSDS approaches the problem firstly by establishing a common vocabulary: the ground station equipment is now called Cross Support Service System (CSSS) [1], the station provider is, therefore, the Provider CSSS, while the mission is now the User CSSS, whereas the spacecraft will be referred to as Space User Node. Moreover, what was previously called “contact request” is now referred to as Service Package. The “Service Package Data Formats” proposed book [2] outlines in detail the Service Package data structure and its parameters, as intended by the CSSDS-CSS group. Here not only the vocabulary has changed, but the whole concept of “resource allocation” did. The Service Package Information entity is a “wrapper” for a set of space communication services that are scheduled at a Provider CSSS [1, 5-13]. The Service Package wraps now services that are meant to be executed by a ground station aperture for a specific spacecraft within a specific time frame given by the start and stop time parameters. Whether this time frame occurs at the time of a “contact” is optional. Additionally, when multiple services are present in a Service Package, each individual service may be executed at different time frames, which may or may not overlap with each other. For example, a typical Service Package may contain a Telemetry Service and a Telecommand Service. While the Telemetry service may extend to the whole duration of the Service Package, the Telecommand service may extend only to a fraction of the time, perhaps only when the spacecraft has crossed a certain elevation threshold, to mitigate risks of data loss. In the next chapters, a more detailed representation of the Service Package data structure and all the parameters that are most relevant for the scheduling of ground station resources will be provided.

Once such vocabulary has been established, CCSDS goes further by defining a set of messages to be exchanged among the several nodes of a CSSS network: these go by the name of “Service Management Utilization Request Formats” (SMURF) and they are thoroughly described in [3]. The SMURF book describes in detail what kind of messages are to be exchanged by the CSSS User and CSSS provider, as well as their content. In this paper one of the

messages from SMURF will be analyzed: the “Online New Service Package Request” [3, 3-14], as it is particularly relevant for scheduling applications.

Finally, CCSDS provides a simplified format to exchange scheduling information of allocated resources: the so-called Simple Schedule Format, outlined in “CSSM – Simple Schedule Format Specification” [4], implements a data structure containing the basic information related to a scheduled Service Package, which is going to be executed at a Provider CSSS. Such data structure will most likely be issued by the Provider CSSS to be transferred to the User CSSS. However, the current proposed and published Cross Support (CS)–Service Management books [4] do not freeze the usage of the mentioned data formats, or at least not yet. The ifs and hows of Service Management data formats are very much still open to the interpretation of ground engineers and software developers. This is due to the fact that CS – Service Management concept is still an open topic and there are still discussions ongoing between member agencies of CCSDS. Hopefully in the upcoming years, the CSS – SM working group will manage to complete the recommended standards, but at the time of this paper the work is still ongoing. For this reason, throughout this paper the subject of the analysis will be the DLR approach to the implementation of such recommended standards. Thanks to its involvement in CCSDS working groups, the GDS team of DLR is at the forefront in the implementation of the CSS–SM standards. This is reflected in the implementation of a new scheduling software, GSSNG, whose aim is to support the Service Management data formats, as well as cope with legacy scheduling procedures. The result is a hybrid system, which can interface with ground station providers using their proprietary protocols, and provide CCSDS-based APIs for mission users and providers at the same time. A brief overview of a CSSM provider interface of GSSNG will be given in the final part of this paper.

2. Service Package Data Format

2.1 Service Types

The Service Package Data Format (SPDF) implements a standard data structure to express a package of one or more services [2, 1-1] from a service provider to a mission user. With the word “service” is meant anything that a ground station provider could offer to a space mission. Typically, these will be telecommand, telemetry, and tracking (TT&C), but service types are not limited to these three. In fact, CCSDS specifies as many service types as listed in Table 1. It can be noticed that both online and offline services are included, at least according to the draft version (red book). Online service means that at the time of the service execution, an active communication link with the spacecraft exists, whereas offline service means that the service execution occurs solely and is limited to the ground.

Table 1. List of service types

<u>Service Type</u>
Apa-az/el
Apa-x/y
Deltador
Doppler
Offline-tm-recording
Offline-tm-provision
Rf-only
Ranging
Reserved
Tbd
Telecommand
Telemetry
Test
Unused
VLBI

2.1.1 Service Types in GSSNG

In the DLR implementation of the CSSM concept all Service Types mentioned in Table 1, are considered. Nevertheless, only a subset of those is actually used when scheduling ground station resources. This is because traditional missions are not yet aware of the full potential the standard offers, and they tend to look at any service

different from Telemetry and Telecommand as a special service request. For this reason, the focus of GSSNG has been mainly on the two mentioned services, and a convenient form was designed for the GSSNG user to fill in all the parameters which might be relevant to the service execution. It is important to say that in reality, the configuration parameters of a ground station terminal outnumber by far the parameters considered by the GSSNG service profile. Currently, the goal of GSSNG is not to provide all parameters a ground station needs to operate but to model only those parameters that are known and understood by both the mission user and the station provider. The former objective might be pursued in the future, once the proposed blue book in [5] will be published. For the moment the focus of Service Profiles is to bridge the knowledge gap between the mission user and the service provider and provide a common vocabulary. In Figure 1 there can be seen an example of a Telemetry service form. The parameters have been divided into five thematic macro-areas:

- *Antenna parameters*: contains typical parameters which constrain the execution mode of the terminal, such as polarization and tracking mode.
- *Physical channel*: here are the actual physical parameters of the telecom link, such as frequency, modulation, and polarization. In Uplink's case, there might also be sweep frequencies, doppler offset, compensation, modulation index, EIRP, etc.
- *Sync and channel coding*: parameters linked to the coding and decoding of transfer frames, such as CADU length, Symbol decoding technique, or in the case of Uplink, CLTU length, Sequence Pattern, etc.
- *Data delivery production*: Flags indicating what to do with the downlinked data, for example TM recording (yes or no).
- *Data transfer services*: typically, SLE service parameters like Service Instance Id, delivery mode, virtual channel Id, spacecraft Id, etc.

A similar form has been prepared for all other service types, but as the required parameters are different for each service, there exist two additional thematic macro-areas across the different services:

- *Radiometric data production*: parameters relative to specific radiometric services such as RF only and VLBI, indicating specific usage of the collected radiometric data
- *Offline data storage*: parameters relative to the storage of data after specific offline services.

The form is titled "Telemetry-Details" and includes the instruction "Fill formular for Telemetry service details". It is organized into five distinct sections, each with a blue header:

- Antenna**: Contains "Polarisation" (dropdown menu set to "RHC") and "Antenna Tracking Mode" (dropdown menu set to "AUTOTRACK").
- Physical Channel**: Contains "Frequency [MHz]" (text input field with "2,034"), "Frequency Band" (dropdown menu set to "S-NE"), "Modulation" (dropdown menu set to "BPSK"), and "Symbol Rate [1-40000000]" (text input field).
- Sync and Channel Coding**: Contains "ASM" (dropdown menu set to "CCSDS Pattern"), "CADU Length [48-3278]" (text input field), "Derandomization" (checkbox checked), "FECF" (checkbox unchecked), and "Symbol Decoding" (dropdown menu set to "Convolutional").
- Data Delivery Production**: Contains "Telemetry Recording" (checkbox checked).
- Data Transfer Services**: Contains "Real-Time Transfer" (checkbox unchecked), "Spacecraft ID" (text input field), "VCID [1-7]" (text input field), "ServiceInstanceID" (text input field), and "Delivery Mode" (dropdown menu set to "Online Complete").

Fig. 1. GSSNG Form for Telemetry Service details

This being said, it is necessary to mention that all data entered in forms similar to Figure 1 is persisted by GSSNG using the abstraction of Service Profile. Since every Service Profile is uniquely identified by an Id, this can be used by other entities as a reference to a specific Service Profile, thus the set of parameters that the profile specifies. In GSSNG Service Profiles are then wrapped by Configuration Profiles (they will be explained in chapter 2.3) and these are referenced by a specific Service Package. This reference chain provides the link for the Provider to what has to be done and at what time. Additionally, it is foreseen by CCSDS that each Service within a Service Package has its start and stop time. This would allow mission users to specify when they wish to execute which service, without the need for voice contact between space and ground operators. For example, when servicing a MEO satellite, there could be TM and TC services at the beginning of the pass, followed by a Ranging service, and then again TM and TC services at the end of the pass, bringing the number of services contained in the service package to 5. However, GSSNG is not supporting this feature yet, as not required for routine scheduling operations with current missions and because special cases are still handled by the operators via voice loops.

2.2 Service Package Types

Now, it is necessary to mention that there exist two types of Service Package formats, namely Type 1 and Type 2 SPDF. The main difference is that Type 2 contains the complete content of the Configuration Profiles and Event Sequence data entities, while Type 1 is a shorter representation and contains only the references of such entities. Because in current scheduling operations at DLR, the Type 1 SPDF is sufficient to fulfill all operational requirements, this will be the only focus of this paper. Although there would be a lot more to say about Type 2, it would exceed the scope of this analysis. In particular, the focus will be on Type 1 Service Packages for online services, since it is the only SPDF description supported by DLR scheduling software at the moment. From a design perspective, the Type 1 SPDF is itself composed of many different classes, which are extensively described by the UML diagram from the SPDF book [2, Figure 3-2]. The most important attributes from the Service Package Entity class diagram are summarized in Table 2. The column “Parent Class” refers to the Extensible Markup Language (XML) element to which the parameter or parameters belong. The information was extrapolated by putting together several classes description from the CCSDS SPDF book [2].

Table 2. Attributes of Service Package Entity subclasses

Parameter	Parent class	Data Type	Description
<i>servicePkgUsage</i>	ServicePkgHeader	Enumeration The following values are permitted: - INTERFACE-TEST - OPERATIONAL - OTHER	Mandatory parameter. Expresses the intended usage of the contained set of service packages.
<i>spaceUserNode</i>	ServicePkgBody	xsd:string	Mandatory Parameter. The spacecraft of the requested or scheduled package.
<i>serviceReqRef</i>	ServicePkgBody	xsd:string	Mandatory parameter. Service request identifier.
<i>servicePackageID</i>	ServicePkg	xsd:string	Mandatory parameter. Unique string identifier.
<i>beginningOfTrack</i>	ServicePkg	CCSDS ASCII Time Code B	Mandatory parameter. Time at which the Service Package is scheduled to start.
<i>endOfTrack</i>	ServicePkg	CCSDS ASCII Time Code B	Mandatory parameter. Time at which the Service Package is scheduled to stop.
<i>servicePackageStatus</i>	ServicePkg	xsd:string	Optional parameter. This may be used by the provider to indicate the status of the service package.
<i>siteRef</i>	ApertureReference	xsd:string	Name of the site (ground or spacecraft) to which the event refers.
<i>apertureRef</i>	ApertureReference	xsd:string	Optional Parameter. This may be used to specify the identifier of the aperture to which the event refers.
<i>trajectoryRef</i>	OnlineSrvPkgType1	xsd:string	Optional parameter. A reference to the existing trajectory data that is to be used.
<i>configProfileRef</i>	OnlineSrvPkgType1Details	xsd:string	A reference to the existing configuration profile that is to be used.

The parameters described in Table 2 are the typical parameters that can be found in a SPDF. With these, it is possible for station and ground operators to uniquely identify a specific service provision, as well as configure the ground station’s selected aperture and execute the service or services. The link to which or what kind of services are to be executed is provided by the Configuration Profile Reference. A configuration profile is itself an additional

entity defined by CSS-SM, in the proposed blue book CSSM - Service Agreement and Configuration Profile Formats [5].

2.3 Configuration profile and Services

The Configuration Profile class is a wrapper for a set of Service Profiles, which themselves contain all the information useful to configure the ground station and ground data connections necessary to execute the service, as explained in 1.1.1. For example, for a Telecommand service, a Telecommand Service Profile may contain information regarding antenna polarization and tracking mode, as well as parameters regarding the physical channel such as frequency band, modulation parameters, EIRP, channel synchronization and coding, but also data transfer service parameter, to configure SLE connections. The data stored in a profile can be referenced to in a Configuration Profile and fetched at the time of its utilization.

Traditionally all such information is handled during a mission preparation phase and all the configuration parameters are prepared and tested before the launch. Nevertheless, with such a data structure at hand, CCSDS embraces future scenarios, where thanks to a higher level of interoperability it will be possible for mission managers to extend on-demand the mission's ground network with new ground station resources, or else to just change some configuration parameters to adapt the terminal or ground equipment to changing requirements.

2.4 The Service Package in GSSNG

The concept of Service Package has been extensively modeled in the new scheduling software of DLR. The Service Management schema files have been used as a basis, to model a simpler data structure in the form of an Entity-Relationship database schema. The main data entities of such a database have been inspired by the CS-Service Management entities. In particular, the concepts of Service Package, Service Package Group, Service Request, Configuration Profile, Service Profile, Trajectory and Aperture, have all been inspired by the CSSM concept. The reason for the simplification of the schema is that many of the classes or entities recommended by CCSDS can be constructed on the fly with data persisted in a shallower form. When it is said that GSSNG supports CCSDS data structures, it is meant that GSSNG can provide on-demand such data structures from the data persisted in its database. This is done using automatic converter software, which takes as input both CCSDS classes (generated from XML schema files), and GSSNG persisted data.

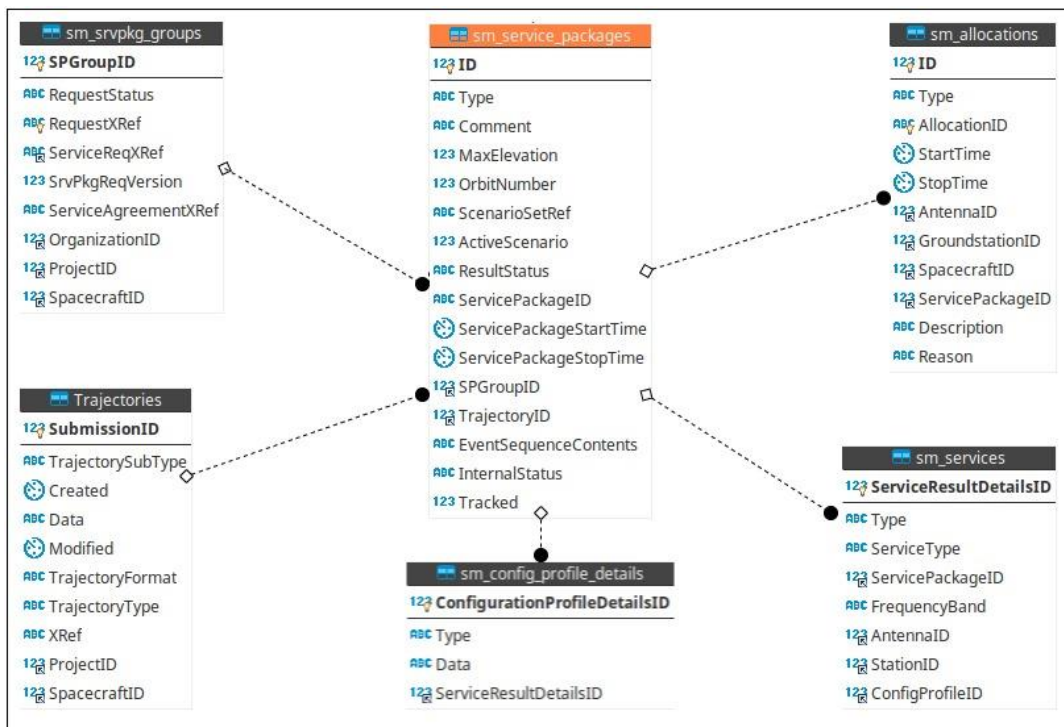


Fig. 2. Service Package ER diagram

The GSSNG Service Package data is actually constructed itself from different tables in the database which have relationships with each other. The Service Package ER diagram is shown in Figure 2. This shows that any current representation of a Service Package is dependent on the current state of many other tables, which themselves contain unstructured data of entities from the persisted data model. This means that any action a user performs via the GSSNG GUI could potentially lead to changes in the representation of one or more service packages. For example, if the Configuration Profile content of one specific Configuration Profile is changed, by for example adding or removing a Service, that will change future reconstruction of all Service Packages that have a reference to that Configuration Profile. The same discourse can be done for Trajectories and Antennas. This allows GSSNG users to easily create or modify large numbers of Service Packages with a few GUI interactions.

What is a Service Package in the GSSNG context? It could be described as an abstract data structure, composed of a core set of parameters, a set of reference parameters with relationships (in the language of databases these are called “Foreign keys”) to other existing entities in the database, and other accessory parameters.

The core set of parameters consists of:

- Service Package Identifier
- Service Package Start and Stop times
- Service Package Status

While the reference set of parameters consists of:

- Ground station reference
- Spacecraft reference
- Configuration Profile reference
- Service Profile reference(s)
- Trajectory reference

Other accessory and optional parameters of a GSSNG Service Package are for example:

- Aperture reference (optional)
- Max elevation
- Orbit number
- Comment field

It is evident the similarity of the first two sets of parameters, with the parameters defined by CCSDS-SPDF and listed in Table 2. The main difference is that CSSM describes merely a data format that is intended for data exchange between partner agencies or CSSS Users and Providers, whereas GSSNG defines an abstract entity that exists in form of structured dynamic data, and because of its nature can take any representational form that is desired to assume. For example, it is possible to easily convert it to SPDF by combining the XML schema from CSSM with the Service Package data in the GSSNG database. An example result can be visualized in Figure 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<servicePkgEntity xmlns="urn:ccsds:schema:csm:1.0.0">
  <servicePkgHeader originatingOrganization="GSOC-GSSNG"
    generationTime="2023-003T14:44:23.763884Z"
    version="1"
    servicePkgUsage="OPERATIONAL"/>
  <servicePkgBody spaceUserNode="BOS"
    serviceReqRef="GSOC-Legacy:service-request-id-1670953012041"
    serviceReqVersion="1">
    <servicePkg servicePackageID="GSOC-GSSNG:srv-pkg-id-1670953012819"
      beginningOfTrack="2023-009T08:45:48Z"
      endOfTrack="2023-009T08:54:37Z"
      servicePackageStatus="SCHEDULED">
      <onlineSrvPkg trajectoryRef="System-TLE:BOS">
        <onlineSrvPkgDetails configProfileRef="BIROS_NOMINAL_prof"/>
      </onlineSrvPkg>
      <apertureReference siteRef="WHM"
        apertureRef="S69"/>
    </servicePkg>
  </servicePkgBody>
</servicePkgEntity>
```

Fig. 3. Example of SPDF generated for BOS mission

2.5 The Service Package Status

One of the most relevant parameters for scheduling resources is the Service Package Status. In the CCSDS book [2, Table 3-4] it is described as an optional parameter that may be used by the service provider to announce the status of the service package in their system and its type is a simple string. CCSDS does not constrain this field further, leaving its usage open to the interpretation of the developers. In GSSNG this field has been extensively used to keep track of the life cycle of the Service Package. Here the type was constrained to Enumeration, which can assume all the following values:

- **CREATED:** The Service Package was generated after a Service Package Request, and it is now persisted in the GSSNG system for further processing.
- **SCHEDULED:** The Service Package was acknowledged and approved by the Service Provider to whom it concerns. The resources to execute the support have been granted and allocated.
- **EXECUTING:** The Service Package is being executed. This status will remain for the whole duration of the service(s) execution.
- **ARCHIVED:** The Service Package has successfully completed its execution.
- **REJECTED:** The Service Package has been rejected by the Service Provider. No resource will be allocated to the execution of the Service Package.
- **CANCELLED:** The Service Package request has been withdrawn by the user, or the Service Package was previously accepted and the acceptance has been withdrawn by the Service Provider
- **ABORTED:** This status occurs when a scheduling action (deletion) is performed on a Service Package during the course of its execution.

In Figure 4 the state machine of the Service Package Status parameter is shown. This gives a good overview of a Service Package life cycle. Immediately after its creation, a Service Package has the Status CREATED, when it encounters the first logic block: this is when feedback from the ground station provider is usually expected. In case of a negative response, the SP assumes the status REJECTED and ends its life cycle. On the contrary, in case of a positive response, it takes the status SCHEDULED and can continue until its execution, or until the request is withdrawn by the user or provider. In that case, the status transitions to CANCELLED, and the SP again ends its life cycle.

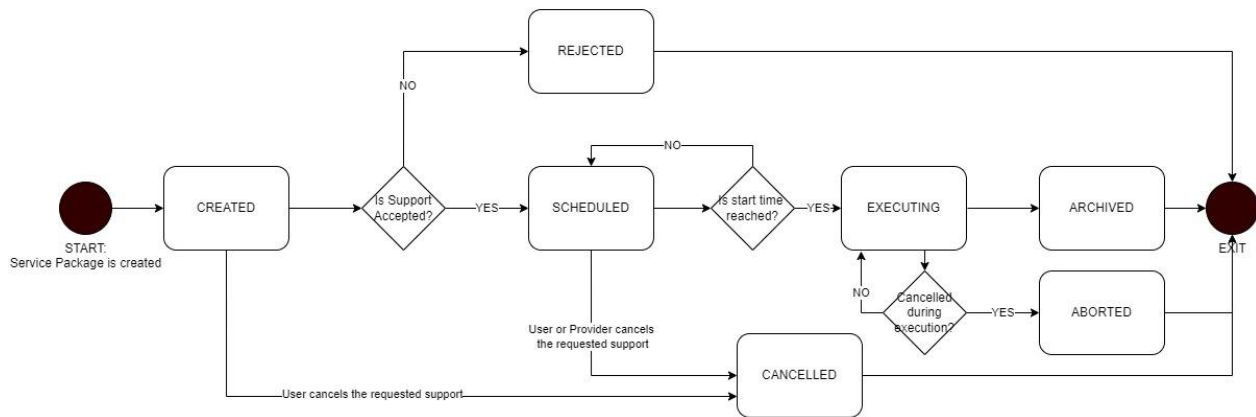


Fig. 4. Service Package Status State Machine in GSSNG

3. Service Package Requests

In the SMURF book [3] several request formats are defined, to provide a common framework for submitting requests between CSSS nodes. Since the common currency of CSSS nodes is the Service Package entity, SMURF provides several request formats, released as XML schemas, to submit requests targeting new or existing Service Packages that are persisted at the other node or nodes. These go under the name of Service Package requests [3, 1.3.2.8]. By definition, the Service Package requests are meant to be submitted by a CSSS User to a CSSS Provider. Different specializations of Service Package Requests are provided according to what type of operation is desired. For example, the NewOnlineSrvPkgReq is a concrete specialization of the abstract class ServicePackageReq [3, 3.6.4] and it is used to submit a new Service Package request by the User to the Provider. In other words, this is similar to what in the past for many agencies was called Schedule Request or Contact Request. Other similar formats provided by specialization of the class ServicePackageReq are:

- ReplaceSrvPkg [3, 3.8.5]
- DeleteSrvPkgReq [3, 3.8.3]
- DeleteSrvPkg [3, 3.8.4]

Since they all inherit from the same abstract class, they share similar data structures. The classes above are all contained in a `srvMgtUtilReq` (Service Management Utilization Request) wrapper class, where a `srvMgtUtilReqHeader` defines several attributes, necessary to constrain the request into a time frame and a context, for example by stating the originating organization of the request, the purpose, description and a version number for tracking changes. Figure 5 shows an example of SMURF request, in particular the `newOnlineSrvPkgReq`, which is meant to forward the information from the CSSS User to the CSSS Provider that a new Service Package is requested and it shall be created at the Provider side, using the content of the request as constraints.

```
<srvMgtUtilReq xmlns="urn:ccsds:schema:cssm:1.0.0">
  <srvMgtUtilReqHeader
    originatingOrganization="DLR"
    generationTime="2022-229T08:00:13.000Z"
    version="1"
    startTime="2022-230T12:18:00.000Z"
    endTime="2022-230T12:48:00.000Z"
    purpose="GSOC BOS requests"
    description="Additional pass request"
    utilizationReqStatus="OPERATIONAL"/>
  <newOnlineSrvPkgReq
    requestID="REQ-ID-2022.186.10.07.02.581830-1239936"
    serviceAgreementRef="BOS_Routine"
    comment="GSOC GDS"
    serviceReqID="REQ-ID-2022.186.10.07.02.581830-1239936"
    srvPkgReqVersion="1">
    <spaceUserNode>BOS</spaceUserNode>
    <apertureSelection priority="1">
      <apertureReference siteRef="WHM" apertureRef="S69"/>
    </apertureSelection>
    <basicPass
      preferredStartTime="2022-230T12:18:00.000Z"
      latestStartTime="2022-230T12:18:00.000Z"
      earliestStartTime="2022-230T12:18:00.000Z"
      preferredDuration="8210"/>
    <onlineSrvPkg trajectoryRef="System-TLE:BOS">
      <onlineSrvPkgDetails
        configProfileRef="provider:configuration_profile_BOS"/>
      </onlineSrvPkg>
    </newOnlineSrvPkgReq>
</srvMgtUtilReq>
```

Fig. 5. Example of SMURF – `newOnlineSrvPkgReq` generated for BOS mission

Since it is a Service Package request, it shares many of the parameters from the SPDF data structure analyzed in the previous chapter. There are a few differences in the formatting of parameters as XML attributes vs elements (for example `spaceUserNode`), but the concept doesn't change. The one main difference is in how the times of the request are presented. In Figure 5 the request refers to a specific pass of the satellite BOS. Therefore, in the body of the request the tag "basicPass" appears followed by `preferredStartTime` and `preferredDuration` as only required parameters [3, 3.10.4]. This shows that the message is carrying a request which may be partially, entirely, or not at all fulfilled. It is up to the provider to generate a response with the exact times of the service provision. In GSSNG, when a new Online Service Package Request is received, a Service Package is generated according to the exact `preferredStartTime` of the request, translated to Service Package `BeginningOfTrack` time, and where the `preferredDuration` is summed to that to compute the Service Package `EndOfTrack`. This is further stated by the two

optional parameters latestStartTime and earliestStartTime. It was implemented this way for simplicity, and because DLR missions require exact tracking times as what is requested. But the standard designed by CCSDS aims at offering this kind of flexibility, where a user could specify a time window in which they want the tracking to start, plus a wished duration, such that the provider can try to offer the service on a best effort basis, also when the requested resource is partially allocated to another mission at those times.

4. Simple Schedule Format

In the published blue book CSSM – Simple Schedule Format Specification [4], CCSDS wants to provide a standard format for conveying scheduling information from a Service Provider to one or more Service Users and involved parties. The Simple Schedule is meant to be intended as a snapshot of the current state of everything that is scheduled at a ground station or set of ground stations. In Figure 6 an example of a Simple Schedule is shown. The data is taken from the running instance of GSSNG at GSOC. The Simple Schedule was in this case cut to show only two Service Packages for shortness and simplicity. The missions of this example are BIROS and EnMAP, as it can be found in the attribute “user” of the ScheduledPackage class (EN1 is an acronym for the spacecraft of mission EnMAP, whereas BOS stands for BIROS). The main difference from SPDF in Simple Schedule is the presence of additional information, typically known by the CSSS Provider, such as the Activity Times of a pass. With Activity Time, reported in Simple Schedule as BeginningOfActivity and EndOfActivity it is meant the time period at which the allocation of the resource starts, which includes the time taken by the ground station provider to configure and setup the antenna before the pass, until the time at which the allocation of the resource ends, which includes the time for necessary teardown activities. During such Activity time, it is normally included the Tracking time, designated by beginningOfTrack and endOfTrack parameters.

```
<simpleSchedule xmlns="urn:ccsds:schema:csm:1.0.0">
  <simpleScheduleHeader
    originatingOrganization="GSOC-GSSNG"
    generationTime="2023-003T13:05:56.617668Z"
    version="1"
    startTime="2023-009T00:00:00Z"
    endTime="2023-016T00:00:00Z"
    status="OPERATIONAL"
    inclusionType="OVERLAP_INCLUSION"/>
  <scheduledPackage scheduledPackageId="GSOC-GSSNG:srv-pkg-id-12819"
    user="BOS"
    comment="">
    <servicePackageXRef serviceAgreementRef="n/a"
      servicePackageRef="GSOC:srv-pkg-id-12819"/>
    <scheduledActivity scheduledActivityId="GSOC:srv-pkg-id-12819-1"
      activityStatus="COMMITTED"
      siteRef="WHM"
      apertureRef="S69"
      beginningOfActivity="2023-009T08:25:48Z"
      beginningOfTrack="2023-009T08:45:48Z"
      endOfTrack="2023-009T08:54:37Z"
      endOfActivity="2023-009T08:59:37Z"
      orbitNumber="36385">
      <serviceInfo serviceType="TELEMETRY"
        frequencyBand="S-NE"/>
      <serviceInfo serviceType="TELECOMMAND"
        frequencyBand="S-NE"/>
    </scheduledActivity>
  </scheduledPackage>
  <scheduledPackage scheduledPackageId="GSOC-GSSNG:srv-pkg-id-13013"
    user="EN1"
    comment="">
    <servicePackageXRef serviceAgreementRef="n/a"
      servicePackageRef="GSOC:srv-pkg-id-13013"/>
    <scheduledActivity scheduledActivityId="GSOC:srv-pkg-id-13013-1"
      activityStatus="COMMITTED"
      siteRef="WHM"
      apertureRef="S67"
      beginningOfActivity="2023-009T10:04:03Z"
      beginningOfTrack="2023-009T10:24:03Z"
      endOfTrack="2023-009T10:34:52Z"
      endOfActivity="2023-009T10:39:52Z"
      orbitNumber="4168">
      <serviceInfo serviceType="TELECOMMAND"
        frequencyBand="S-NE"/>
      <serviceInfo serviceType="TELEMETRY"
        frequencyBand="S-NE"/>
    </scheduledActivity>
  </scheduledPackage>
</simpleSchedule>
```

Fig. 6. Example of Simple Schedule Format

Another clear difference is that here we find an explicit reference to the Services to be executed during the pass, thanks to the class ServiceInfo. In this service type information can be found, where in this case it is only one parameter, the Frequency Band, but optionally additional parameters can be set. Here one could find for example specific start and stop times for the Service when it differs from the Service Package beginning and end of Track. This Simple Schedule format can be fetched from GSSNG in principle by every user with the right privileges and access to the GUI or the internal API. Furthermore, it is being fetched by automated software developed by a software team at the DLR ground station in Weilheim, for providing automated schedule information to the operators and the ground station control system. In the future, this is expected to be the main schedule format to be fetched by users for mission planning procedures.

5. GSSNG implementation of CS - Service Management

As mentioned in the introduction, a final blue book on CSSM – Service Management has not been proposed yet. Nevertheless, in the past few years, DLR has been trying to give its solution to the scheduling problem using CSSM existing data formats, in particular combining both SMURF [3] and SPDF [2] to design a novel solution for an automated interface between CSSS User and Provider. The hybrid solution was designed in particular to automate the scheduling of the DLR-Neustrelitz ground station via GSSNG. The solution was tailored to the requirement of automating five specific use cases, which result in the interaction between GSSNG and the ground station scheduling system, where GSSNG is intended to be here the CSSS User, while the ground station is the CSSS Provider. The use cases are the following:

1. Schedule Request from User to Provider
2. Acceptance or Rejection of Schedule Request
3. Cancellation of Schedule Request by User
4. Modification of Schedule Request by User
5. Cancellation of Schedule Request by Provider

To automate such cases a combination of SMURF and SPDF is used. The transport layer was chosen to be the HTTP/REST framework, hence a set of POST requests which encapsulate the CCSDS data messages in their request body. In general, the User will use SMURF messages, while the Provider will send only SPDF. The Provider response to the User request is therefore communicated via the Service Package Status field. Because of the asynchronous nature of scheduling decisions, it was not possible to establish a simple CLIENT–SERVER communication pattern. In fact, to provide maximum flexibility, it was decided that both sides should accommodate both a CLIENT and a SERVER application. This way it is possible to exchange CCSDS messages between the two sides without a polling mechanism and with complete autonomy. In detail, what happens in each scenario is presented in the following sub-chapters.

5.1 Case 1: Scheduling Request User to Provider

The User or one of its customers introduces the request in the GSSNG scheduling system. The request format is generated at GSSNG and sent via the REST POST method, to the Provider’s scheduling system. After receiving the request, the Provider acknowledges the request by sending a SPDF back to User. The whole process can be described in Figure 7.

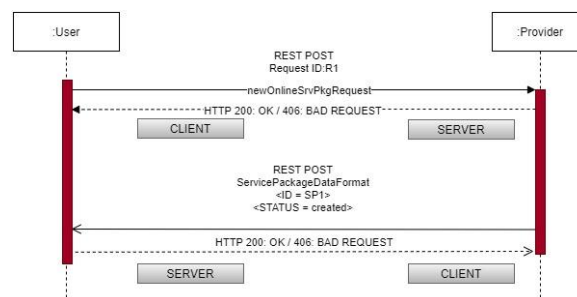


Fig. 7. Use case 1 – User sends Schedule Request

5.2 Case 2: Response of request Provider to User

After checking internally for available resources, the Provider reaches a decision on the schedule request and decides whether to provide support or not for the requested pass. After the decision has been taken, the Provider's system has to notify the User's system about the scheduling decision. It does so by sending the Service Package in SPD Format with the status updated to either ACCEPTED or REJECTED. This scenario is presented in Figure 8.

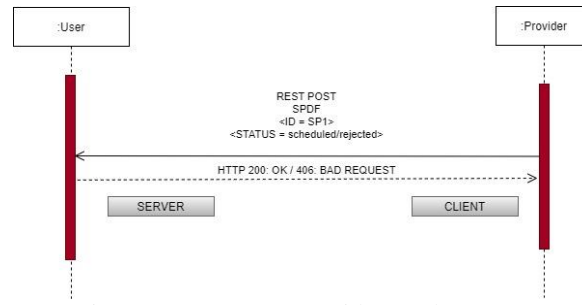


Fig. 8. Use case 2 – Provider sends a response

5.3 Case 3: Cancellation of scheduled service package request by User

The User withdraws the requested passage (this may happen for several different reasons). It does so by sending via REST POST a Deletion request (DeleteSrvPkgReq, [3, 3.8.3]). The Provider acknowledges the withdrawal and sends the SPDF of the cancelled Service Package with the Service Package status updated to "cancelled". Figure 9 describes this scenario.

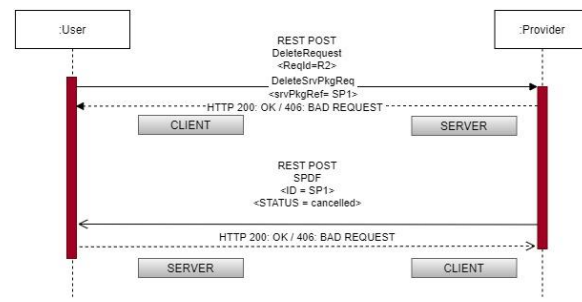


Fig. 9. Use case 3 – Cancellation of Request by User

5.4 Case 4: Modification of scheduled service package request by User

The User changes the requested booking (this may happen for several different reasons), for example by changing the Configuration Profile of the pass or the requested start/stop times. It does so by sending via REST POST an update request (ReplaceSrvPkg, [3, 3.8.5]). The Provider acknowledges the updates and sends the SPDF of the modified Service Package including the updated parameters in the body of the XML message. This scenario is described by the diagram in Figure 10.

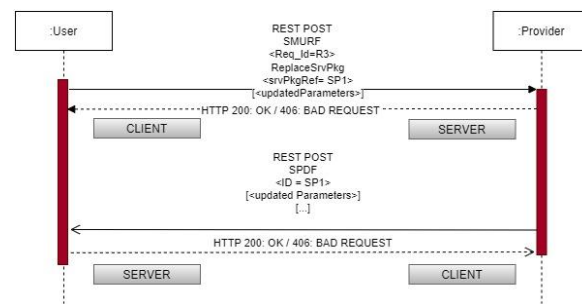


Fig. 10. Use case 4 – Modification of Request by User

5.5 Case 5: Cancellation of scheduled service package request by Provider

This scenario applies to circumstances in which a previously accepted Service Package request, cannot be supported by the Service Provider anymore. Reasons may be an antenna failure, the allocation of the resource for a higher priority mission, or others. The Provider must notify the User about the cancellation of a previously accepted pass. It does so by sending a REST POST with the SPDF of the Service Package and its status updated to "cancelled". A diagram of case 5 is shown in Figure 11.

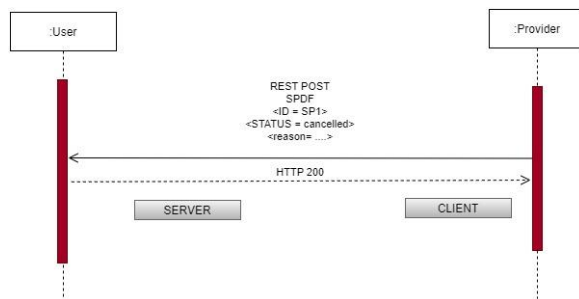


Fig. 11. Use case 5 – Cancellation of Request by Provider

5.6 Conclusions

The solution analyzed above is still under development at GSOC for the GSSNG system, and technical documentation and ICDs are still under review by all involved parties. One open point was for example the usage of the HTTP response to POST request. On one hand, it could be used to directly deliver feedback on the Service Package request (e.g. 200-OK means accepted, 400 means rejected). On second thought, it was decided to use such a response to provide information on the validity of the requests. Therefore, when getting status 200 means that the previous request was successfully validated by either party and considered formally correct. On the contrary, when getting the status 406 it means that the request was invalid, therefore it must be corrected and sent again. It may happen when some of the parameters in the request body are not recognized by either party or when the user is asking for a resource or a spacecraft not supported by the provider. Another hot topic is the timeliness and expiration time of the messages. Until what time before the pass can the user send a SMURF request? Until when is a deletion request considered valid, if no response is received from the provider? In the first case, it was decided to leave the answer to mission-specific ICDs, because there are different requirements for each mission. In the second case, it was deemed better not to introduce an expiration time for the deletion request, which will remain valid as long as it was acknowledged by the first HTTP OK response from the Provider.

This and other technicalities were thoroughly debated and solutions were provided. Finally, when all the minor details will be taken care of, and the interface will be successfully implemented and tested, which we expect to happen no later than Q3-2023, it will represent the DLR's first operational system to introduce the CCSDS Service Management concept into daily operations.

Conclusion

With this paper a brief introduction of the main CSSM concept was given, with a particular focus on the data structures and formats that are extensively used in routine scheduling operations at DLR. Furthermore, the newly developed scheduling software GSSNG was used as a demonstrator to generate scheduling data formats according to CCSDS standards from real scheduling information. The result shows that CCSDS-recommended formats are suitable to the requirements of ground operations and are being developed, tested, and used in some cases already at present. This paper focused particularly on three concept data formats from CCSDS CSSM and their application at GSOC: The Service Package Data Format, the Service Package Request and the Simple Schedule Format. Additionally, it introduced a novel concept for a future Service Management interface between GSOC and the DLR ground station Neustrelitz. Despite the concept being still to be tested, it shows promising features that would make the automated booking of ground station resources possible using CCSDS data formats.

Acknowledgments

The authors would like to thank the GDS team as a whole, for making scheduling operations possible and all the other teams at GSOC for running tirelessly the biggest space control centre in Germany and making space accessible to scientific and commercial communities. Special thanks go to the colleagues of Neustrelitz ground station for

supporting us in the challenge of implementing the first Cross Support Service Management interface. Other special thanks go to the CCSDS Cross Support Working Group for developing the standards that are the subject of this paper, and for making interoperability between space agencies possible. Personal thanks go to every person who accepted to review this paper and gave their feedback.

References

- [1] *Extensible Space Communication Cross Support—Service Management—Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1. Washington, D.C.: CCSDS, September 2014.
- [2] *Cross Support Service Management—Service Package Data Formats*. Issue 0. Proposed Draft Recommendation for Space Data System Standards (Proposed Red Book), CCSDS 902.4-R-1. Washington, D.C.: CCSDS, August 2022.
- [3] *Cross Support Service Management—Service Management Utilization Request Formats*. Issue 0. Proposed Draft Recommendation for Space Data System Standards (Proposed Red Book), CCSDS 902.9-R-0. Washington, D.C.: CCSDS, August 2022.
- [4] *Cross Support Service Management—Simple Schedule Format Specification*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.1-B-1. Washington, D.C.: CCSDS, May 2018.
- [5] *Cross Support Service Management—Service Agreement and Configuration Profile Formats*. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.5-B-1.