The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Practices

CROSS SUPPORT SERVICE MANAGEMENT—BEST PRACTICES

DRAFT RECOMMENDED PRACTICE

CCSDS 902.11-R-1

RED BOOK

April 2026

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Practices

CROSS SUPPORT SERVICE MANAGEMENT—BEST PRACTICES

DRAFT RECOMMENDED PRACTICE

CCSDS 902.11-R-1

RED BOOK

April 2026

AUTHORITY

Issue:	Red Book, Issue 1
Date:	April 2026
Location:	Not Applicable

(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the email address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not in themselves considered binding on any Agency.

CCSDS Recommendations take two forms: **Recommended Standards** that are prescriptive and are the formal vehicles by which CCSDS Agencies create the standards that specify how elements of their space mission support infrastructure shall operate and interoperate with others; and **Recommended Practices** that are more descriptive in nature and are intended to provide general guidance about how to approach a particular problem associated with space mission support. This **Recommended Practice** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommended Practice** is entirely voluntary and does not imply a commitment by any Agency or organization to implement its recommendations in a prescriptive sense.

No later than five years from its date of issuance, this **Recommended Practice** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Practice** is issued, existing CCSDS-related member Practices and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such Practices or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new Practices and implementations towards the later version of the Recommended Practice.

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Practice is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

https://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the email address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Études Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- State Space Corporation (Roscosmos)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Science Policy Office (BELSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Netherlands Space Office (NSO)/The Netherlands.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Practice. Its 'Red Book' status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organization. As such, its technical contents are not stable, and several iterations of the document may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 902.11-R-1	Cross Support Service Management—Best Practices, Recommended Practice, Draft 20260324	February 2026	Original issue
CCSDS 902.11-R-1	Cross Support Service Management—Best Practices, Draft Recommended Practice, Draft 20260416	March 2026	Updates after CESG review.
CCSDS 902.11-R-1	Cross Support Service Management—Best Practices, Draft Recommended Practice, Issue 1	April 2026	Corrected border round Fig 1-1.

CONTENTS

Section Page

1 INTRODUCTION 1-1
1.1 PURPOSE AND SCOPE 1-1
1.2 APPLICABILITY 1-2
1.3 RATIONALE 1-2
1.4 DOCUMENT STRUCTURE 1-2
1.5 DEFINITIONS 1-3
1.6 NOMENCLATURE 1-3
1.7 CONVENTIONS 1-4
1.8 REFERENCES 1-4
2 OVERVIEW 2-1
2.1 GENERAL 2-1
2.2 DATA ENTITIES 2-1
2.3 MAPPING TO W3C XML SCHEMA 2-1
3 CAPABILITY SETS 3-1
3.1 OVERVIEW 3-1
3.2 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS 3-5
3.3 SERVICE PACKAGE DATA FORMATS 3-12
3.4 SIMPLE SCHEDULE DATA FORMAT 3-16
3.5 USAGE NOTES 3-23

ANNEX A SECURITY, SANA, AND PATENT CONSIDERATIONS (INFORMATIVE) A-1
ANNEX B XML SCHEMA ORGANIZATION AND PACKAGING FOR THE VARIOUS INFORMATION ENTITY CLASSES (INFORMATIVE) B-1
ANNEX C ABBREVIATIONS AND ACRONYMS (INFORMATIVE) C-2

CONTENTS (continued)

Figure Page

1-1 Best Practises in the Context of Space Communication Cross Support Service Management 1-2
3-1 SM CS-1 – SMURF Class Diagram 3-5
3-2 SM CS-1 – SPDF Class Diagram 3-12
3-3 SM CS-1 – Simple Schedule Data Format Class Diagram 3-16

Table

3-1 Class SrvMgtUtilReqHeader Parameters 3-6
3-2 Class NewOnlineSrvPkgReq Parameters 3-7
3-3 Class ReplaceSrvPkg Parameters 3-8
3-4 Class DeleteSrvPkg Parameters 3-8
3-5 Class BasicPass Parameters 3-9
3-6 Class ApertureSelection Parameters 3-10
3-7 Class ApertureReference Parameters 3-10
3-8 Class OnlineSrvPkgType1 Parameters 3-11
3-9 Class OnlineSrvPkgType1Details Parameters 3-11
3-10 Class SrvPkgHeader Parameters 3-13
3-11 Class ServicePkgBody Parameters 3-14
3-12 Class ServicePkg Parameters 3-14
3-13 Class SimpleScheduleHeader Parameters 3-17
3-14 Class ScheduledPackage Parameters 3-18
3-15 Class ScheduledActivity Parameters 3-19
3-16 Class ServiceInfo Parameters 3-20
3-17 Class ServicePackageXRef Parameters 3-20
3-18 Class ScheduledPkgAssn Parameters 3-21

1 INTRODUCTION

Service Management (SM) enables cross-agency coordination of ground station tracking support for space missions. SM provides standardized formats for submitting tracking requests, expressing communications geometry and schedules, and specifying operational parameters (e.g., carrier frequency, modulation, encoding, data rates) to ensure that missions can reliably obtain telecommand, telemetry, and ranging capabilities from any participating ground station network.

1.1 PURPOSE AND SCOPE

1.1.1 PURPOSE

The purpose of this Recommended Practice is to provide guidance on the use of Cross Support Service Management standards to facilitate interoperations in the context of ground station service management.

1.1.2 SCOPE

This book defines the SM Data Entities used in the SM Recommended Standards and Practices. As of this issue, the following are addressed:

- Cross Support Service Management—Service Management Utilization Request Formats (see Ref. [4]);
- Cross Support Service Management—Service Package Data Formats (see Ref. [5]);
- Cross Support Service Management—Simple Schedule Format Specification (see Ref. [6]).

Figure 1-1 puts the Cross Support Service Management—Best Practices into context with the standards that together form the Space Communication Cross Support Service Management.

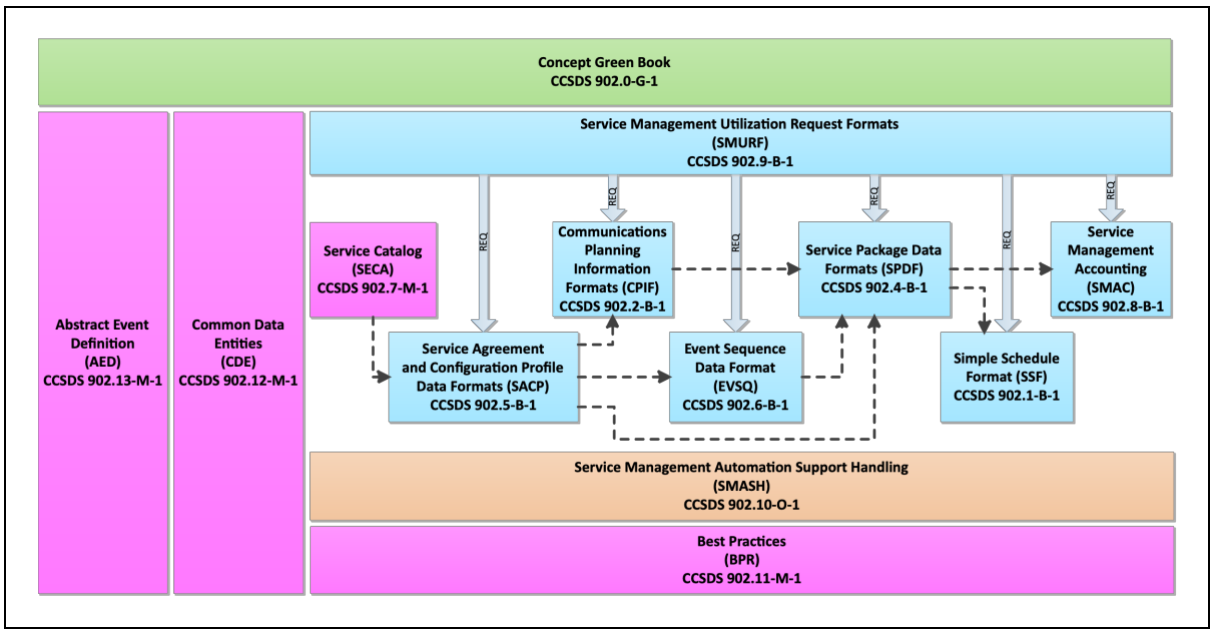

Figure 1-1: Best Practices in the Context of Space Communication Cross Support Service Management

1.2 APPLICABILITY

This Recommended Practice applies across the whole scope of SM. However, not all capabilities outlined in this document are applicable to all SM Recommended Standards and Practices. Different missions require different levels of SM functionality based on their complexity.

1.3 RATIONALE

This document illustrates how the SM Recommended Standards and Practices can support missions of varying complexity. This is done by defining various SM Capability Sets (CS). Capability Set-1 (CS-1) outlines the simplest approach for supporting a straightforward mission with no complex scheduling requirements.

1.4 DOCUMENT STRUCTURE

This document is organized as follows:

a) Section 1 provides the purpose, scope, applicability, and rationale of this Recommended Practice and identifies the conventions, terminology, and references.

b) Section 2 provides a brief overview of the common data entities.

c) Section 3 provides details about the SM CS.

d) Annex A discusses security, Space Assigned Numbers Authority (SANA), and patent considerations.

e) Annex B provides an overview of XML schema organization and packaging for the various information entity classes.

f) Annex C contains a list of abbreviations and acronyms.

1.5 DEFINITIONS

For the purposes of this document, the following definition applies:

agency: A satellite operator or satellite service provider.

1.6 NOMENCLATURE

1.6.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Practice:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.6.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is offset from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.7 CONVENTIONS

1.7.1 UNIFIED MODELING LANGUAGE

The Unified Modelling Language (UML) is a general-purpose developmental modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. It supports various diagram types, including class, package, sequence, and activity diagrams. This document follows the notation, semantics, and conventions imposed by the Version 2.4.1 UML specification of the Object Management Group (OMG) (reference [2]).

The document uses class diagrams only. A UML class diagram describes the structure of a message, its parts, and how those parts interrelate. A UML class, represented in the diagram as a box, represents a data set. Class diagram conventions include composition, generalization, multiplicity, and constraints. Enumeration notation is also used but only when it is involved in a composition constraint.

1.7.2 XML SCHEMA DATATYPES

Many of the datatype definitions in this Recommended Practice are the same as definitions defined by XML Schema, Part 2 Datatypes (see reference [3]). The string ‘xsd:’ serves as a prefix for these datatypes.

1.8 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *Time Code Formats*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 301.0-B-4. Washington, D.C.: CCSDS, November 2010.

[2] *Unified Modeling Language (UML)*. Version 2.4.1. Needham, Massachusetts: Object Management Group, August 2011.

[3] David Peterson, et al., eds. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. Version 1.1. W3C Recommendation.

[4] *Cross Support Service Management—Service Management Utilization Request Formats*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.9-B-1. Washington, D.C.: CCSDS, Forthcoming.

[5] *Cross Support Service Management—Service Package Data Formats*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.4-B-1. Washington, D.C.: CCSDS, Forthcoming.

[6] *Cross Support Service Management—Simple Schedule Format Specification*. Issue 1 Cor. 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.1-B-1 Cor. 1. Washington, D.C.: CCSDS, December 2021

[7] *CCSDS Spacecraft Identification Field Code Assignment Control Procedures*. Issue 7. Recommendation for Space Data System Practices (Magenta Book), CCSDS 320.0-M-7. Washington, D.C.: CCSDS, November 2017.

[8] *Abstract Event Definition*. Issue 1 Cor. 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 902.13-M-1 Cor. 1. Washington, D.C.: CCSDS, February 2024.

2 OVERVIEW

2.1 GENERAL

This section provides a high-level overview of SM data entities, which structure the information exchanged between mission operations centres and ground station networks.

2.2 DATA ENTITIES

SM data entities are defined as XML constructs in XML Schema files. Within each entity, data elements are either mandatory (suitable values must be present) or optional (values may be present).

It is possible to extend the contents of the data entities by defining additional parameters. The content of any additional parameters so defined is outwith the scope of this document and should be documented in an ICD agreed upon by the parties involved.

2.3 MAPPING TO W3C XML SCHEMA

This Recommended Practice includes the specification of a mapping to World Wide Web Consortium (W3C) eXtensible Markup Language (XML) schema. The normative mapping of this Recommended Practice to W3C XML schemas is a virtual annex to this Recommended Practice, contained in a stand-alone set of schema files.

3 CAPABILITY SETS

3.1 OVERVIEW

Service management supports operations ranging from simple LEO missions, through constellations, to complex deep space missions. This range of mission complexity requires flexible SM to support the diverse requirements.

However, complexity is not always required. Service Management Capability Sets (CSs) can be defined to highlight appropriate elements for specific use cases. This document currently defines SM CS-1, which represents the simplest case for missions with basic requirements.

3.1.1 CONVENTIONS

In the following sections, *abstract* classes in the class diagrams are highlighted in green for clarity. Unlike the Recommended Standards/Practices where these classes are defined, this document does not discuss the abstract classes directly. Instead, it describes their parameters where the abstract class is realized into a concrete class.

The class diagrams presented in this document are subsets of original diagrams, showing only classes relevant to a particular SM CS. Nonetheless, all SM CSs defined in this Recommended Practice are fully compatible with the "complete" version, and any XML file produced will validate against the standard SM Schema files.

3.1.1.1 Colour Coding

In the tables describing parameters of the various classes, the following colour coding is used:

- Red background indicates that the parameter is optional in the overall context of Service Management. Within the context of a particular SM capability set, however, it is not required and, therefore, is not specified when that SM CS is used.
- Yellow background indicates that the parameter is optional in the overall context of Service Management and may be specified for a particular SM CS.
- Green background indicates that the parameter is mandatory or optional in the overall context of Service Management but for a particular SM CS, it is considered mandatory and shall be specified.

3.1.1.2 Identifiers and References

For CCSDS SM, keeping track of the transactions and resultant data entities is important for facilitating accountability, especially in automated systems. To that end, CCSDS Service Management adopts the following approach:

1. When a data entity is created, it is given an Identifier (ID);
2. An existing data entity is referenced by its ID.

For example, if a new service request is submitted using the SMURF `NewOnlineSrvPkgReq` class (see Section 3.2.1.3), the User Cross Support Service System (CSSS) must provide two IDs:

- **requestID**: the unique Identifier of the SMURF request;
- **serviceReqID**: the unique Identifier of the service request contained in the NewOnlineSrvPkgReq, which creates a service package request and could be of the form *XMM-Service-Request-0001*.

Subsequent actions on the service package request need to reference the serviceReqID. For example, to modify the service package request using the DeleteServicePkg class (see Section 3.2.1.5), the following needs to be specified:

- **requestID**: the unique Identifier of the SMURF request submitted by the User CSSS;
- **serviceReqRef**: the Reference to the Identifier of the service request contained in a previously submitted NewOnlineSrvPkgReq request. To modify such a service package request with a **serviceReqID** of XMM-Service-Request-0001, the reference contained in **serviceReqRef** should be *XMM-Service-Request-0001.*

Similar considerations apply to the Identifiers and References contained in other data entities used in Service Management.

The Identifier may not always be created during an operation directly supported by Service Management. For example, SM CS-1 does not support operations to submit Service Agreements or Configuration Profiles, but it is does support references to the Identifiers of Service Agreements (see section 3.2.1.3) and Configuration Profiles (see section 3.2.1.10). In such cases, it is necessary for the participating parties to agree on the appropriate Identifiers for the data entities (possibly documenting them in an ICD), so that these Identifiers can then be Referenced by the data entities.

3.1.2 STATES

Conceptually, once a SM Utilization Request has been received by a service provider, syntactically validated, and accepted into the providing agency’s processing system, it undergoes several state changes during its processing. The resulting Service Package undergoes the following state transitions:

- CREATED;
- SCHEDULED;
- DELETED;
- ARCHIVED;
- EXECUTING;
- ABORTED.

The exact processing details depend on the provider agency's system but can likely be mapped to the above states. Ideally, when a request transitions from one state to another, this should be reported to the requesting party, so that they are aware of the latest status of their request. For CS-1, it is assumed that adopting implementations can use their current message exchange mechanisms (e.g., APIs, file transfer) to adequately provide status information.[1]

3.1.3 ASSUMPTIONS

SM CS-1 is the simplest usage of SM. In defining its use case, the following assumptions are made:

1. Any required Trajectory information exchanged between the parties is outwith the scope of SM CS-1. The mechanism for this should be documented in an ICD agreed upon by the involved parties.

2. Any required Service Instance Configuration Files (SICF) exchanged between the parties is outwith the scope of SM CS-1. The mechanism for this should be documented in an ICD agreed upon by the involved parties.

3. Any required Configuration Profiles exchanged between the parties is outwith the scope of SM CS-1. The mechanism for this should be documented in an ICD agreed upon by the involved parties.

4. Any required Event Sequences exchanged between the parties is outwith the scope of SM CS-1. The mechanism for this should be documented in an ICD agreed upon by the involved parties.

5. Delta DOR scan patterns are not required.

6. Only basic constraints are required to specify the required station support, that is:

    a. Preferred start time;

    b. Optional window around preferred start time;

    c. Preferred duration;

    d. Optional flexibility around preferred duration.

7. Aperture selection can be done by one of the following:

    a. specifying both site and aperture references;

    b. specifying only a site reference and deferring aperture selection to the provider;

[1] A recommendation for standardized message exchange is being developed. However, CCSDS SM can be implemented with existing message exchange mechanisms.

c. deferring both site and aperture selection to the provider (i.e., no aperture selection criteria is specified).

8. Selecting an aperture by specifying its location on a celestial body is not required.

9. Reporting of request state transitions is not required.

If any of these assumptions do not hold, an implementation will likely need to perform a local analysis to determine whether CS-1 can be used to achieve interoperability and to what degree.

3.1.4 PREREQUISITES

The following are necessary prerequisites for the use of SM CS-1:

1. A Service Agreement exists between the parties involved.

2. At least 1 Configuration Profile has been defined.

3. The parties involved have agreed on unique name(s) for the Configuration Profile(s).

4. If required, the parties involved have agreed on unique name(s) for the Event Sequence(s).

5. If required, the parties involved have agreed on a mechanism for the return of request status and any associated timing constraints. This may be documented in a bilaterally agreed ICD.

NOTE – The Service Agreement, Configuration Profile, and Event Sequence may not conform to those specified in SM recommendations, i.e., they can be agreed bilaterally and defined in an ICD.

3.2 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS

3.2.1 OVERVIEW

The following class diagram shows the classes of the Service Management Utilization Request Formats (SMURF) that are relevant to SM CS-1.

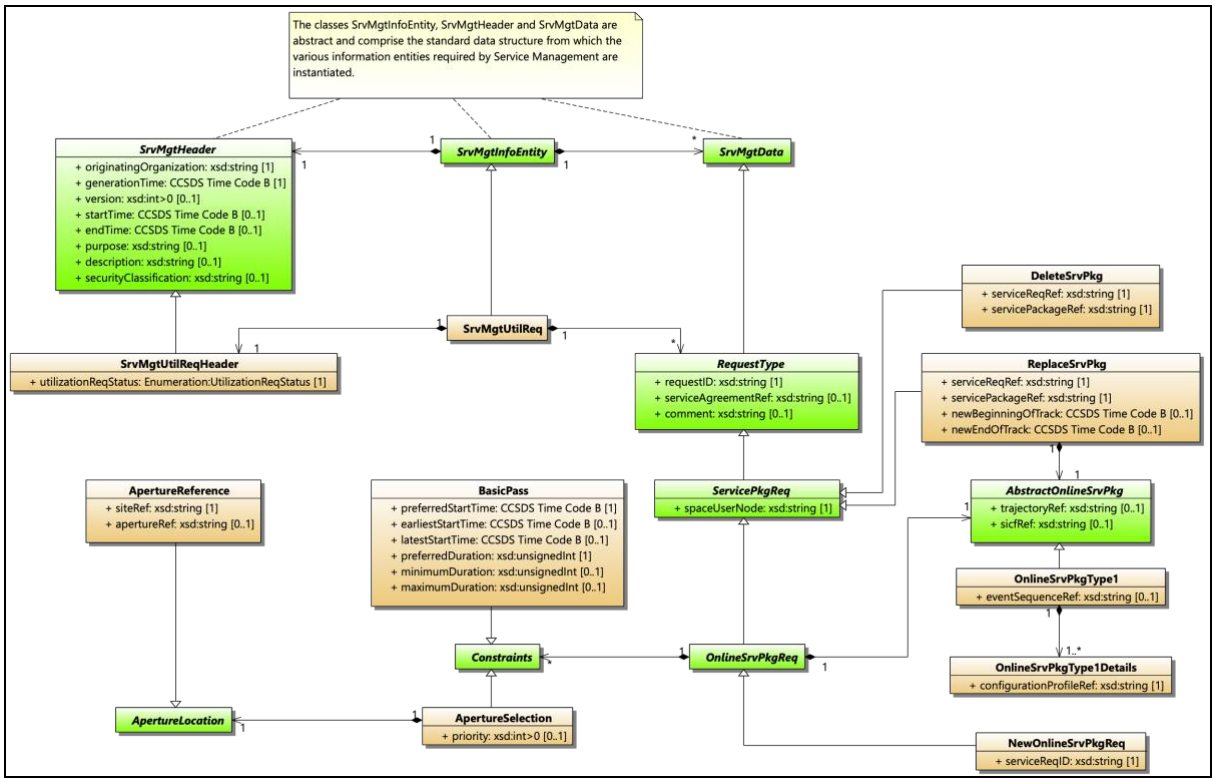

Figure 3-1: Service Management Capability Set-1 – Service Management Utilization Request Formats Class Diagram

For SM CS-1 the SMURF supports three types of requests:

1. Submit a new service package request (class NewOnlineSrvPkgReq);
2. Replace an existing service package (class ReplaceSrvPkg);
3. Delete an existing service Package (class DeleteSrvPkg).

These and other required concrete classes (as shown above in the class diagram) are described in the following sections. Full descriptions of SMURF classes and parameters can be found in reference [4].

3.2.1.1 Class SrvMgtUtilReq

This class is effectively the 'wrapper' for a SM Utilization Request. It does not have any parameters but contains the following classes:

– One instance of the SrvMgtUtilReqHeader class (see section 3.2.1.2);

– Zero or more instances of one of the following classes:

- NewOnlineSrvPkgReq (see section 3.2.1.3);
- ReplaceSrvPkg (see section 3.2.1.4);
- DeleteSrvPkg (see section 3.2.1.5).

3.2.1.2 Class SrvMgtUtilReqHeader

This class contains the 'header' for the SM Utilisation Request. It does not contain any other classes but contains the parameters as described in table 3-1.

Table 3-1: Class SrvMgtUtilReqHeader Parameters

Parameter	Description	Data Type	Data Units
`originatingOrganization`	Organization that generated the information entity. Permitted values registered in SANA.	xsd:string	n/a
`generationTime`	Generation time of information entity in CCSDS ASCII Time Code B format (reference [1])	xsd:string	UTC
`version`	Optional version number of the information entity. Increments when regenerated with the same concrete type, status, `startTime`, and `endTime`. Must increase, but not necessarily by 1.	xsd:integer >0	n/a
`startTime`	Optional start time for the information entity in CCSDS ASCII Time Code B format (reference [1]). ***Not required or specified for SM CS-1.***	xsd:string	UTC
`endTime`	Optional end time for the information entity in CCSDS ASCII Time Code B format (reference [1]). ***Not required or specified for SM CS-1.***	xsd:string	UTC
`purpose`	Optional purpose of the information entity, specified in an ICD between relevant parties.	xsd:string	n/a
`description`	Optional description of the information entity contents, specified in an ICD between relevant parties.	xsd:string	n/a

Parameter	Description	Data Type	Data Units
securityClassification	Optional security classification. Definitions and use specified in an ICD between relevant parties.	xsd:string	n/a
utilizationReqStatus	Status of the SM utilization request. The following values are permitted: – TEST (indicates that the request has been generated for test purposes only); – OPERATIONAL (indicates that this is an operational request).	Enumeration	n/a

3.2.1.3 Class NewOnlineSrvPkgReq

This class contains a new Service Package Request which creates a Service Package Request. It contains zero or more instances of permitted Constraint classes, which are, in the context of SM CS-1:

– BasicPass (see section 3.2.1.6);

– ApertureSelection (see section 3.2.1.7).

It also contains exactly one OnlineSrvPkgType1 class instance (see section 3.2.1.9) and the parameters described in table 3-2.

Table 3-2: Class NewOnlineSrvPkgReq Parameters

Parameter	Description	Data Type	Data Units
requestID	Unique request ID assigned by the requestor.	xsd:string	n/a
serviceAgreementRef	Optional reference to the service agreement under which the requested services are provided.	xsd:string	n/a
comment	Optionally provides ad hoc information.	xsd:string	n/a
spaceUserNode	User of requested services. Must be spacecraft names as registered in SANA.	xsd:string	n/a
serviceReqID	Unique request identifier.	xsd:string	n/a

NOTE – It is the responsibility of the User CSSS to ensure that the serviceReqID is unique for the combination of serviceAgreementRef and spaceUserNode for which the request is being submitted. The provider CSSS can then assume that the combination of serviceAgreementRef, spaceUserNode, and serviceReqID constitutes a unique identifier for a service request.

3.2.1.4 Class ReplaceSrvPkg

This class contains a request to Replace a Service Package, i.e., a Service Package that replaces he one that was previously generated. The servicePackageRef parameter, assigned by the

provider CSSS when a new service package is generated, identifies the Service package to replace. It contains only one OnlineSrvPkgType1 class instance (see section 3.2.1.9) and the parameters described in table 3-3.

Table 3-3: Class ReplaceSrvPkg Parameters

Parameter	Description	Data Type	Data Units
`requestID`	Unique request ID assigned by the requestor.	xsd:string	n/a
`serviceAgreementRef`	Optional reference to the service agreement under which the requested services are provided. NOTE – Must match the `serviceAgreementRef` specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	n/a
`comment`	Optionally provides ad hoc information.	xsd:string	n/a
`spaceUserNode`	User of requested services. Must be spacecraft names registered in SANA. NOTE – Must match the `spaceUserNode` specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	n/a
`serviceReqRef`	Identifier of the service request that generated the service package being replaced, i.e., the `serviceReqID` from the original NewOnlineSrvPkgReq.	xsd:string	n/a
`servicePackageRef`	Reference to the generated service package being replaced.	xsd:string	n/a
`newBeginningOfTrack`	Optional new start time for a pass (in UTC).	CCSDS ASCII Time Code B (reference [1])	UTC
`newEndOfTrack`	Optional new finish time for a pass (in UTC).	CCSDS ASCII Time Code B (reference [1])	UTC

3.2.1.5 Class DeleteSrvPkg

This class contains a request to Delete a Service Package, i.e., the reference to a Service Package to delete. The `servicePackageRef` parameter, assigned by the provider CSSS when a new service package is generated, identifies the Service package to delete. It does not contain any other classes, but contains the parameters described in table 3-4.

Table 3-4: Class DeleteSrvPkg Parameters

Parameter	Description	Data Type	Data Units
`requestID`	Unique request ID assigned by the requestor.	xsd:string	n/a

Parameter	Description	Data Type	Data Units
serviceAgreementRef	Optional reference to the service agreement under which the requested services are to be provided. NOTE – Must match the serviceAgreementRef specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	n/a
comment	Optionally provides ad hoc information.	xsd:string	n/a
spaceUserNode	User of the requested services. Must be spacecraft names registered in SANA. NOTE – Must match the spaceUserNode that was specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	n/a
serviceReqRef	Identifier of the service request that generated the service package(s) being deleted, i.e., the serviceReqID specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	n/a
servicePackageRef	Reference to the service package being deleted.	xsd:string	n/a

3.2.1.6 Class BasicPass

This class may be used to specify the preferred start time and duration of a pass along with the acceptable earliest and/or latest start times and minimum and/or maximum acceptable duration. It does not contain any other classes but contains the parameters described in table 3-5.

Table 3-5: Class BasicPass Parameters

Parameter	Description	Data Type	Data Units
PreferredStartTime	Preferred start time for a pass in UTC: – PreferredStartTime ≥ earliestStartTime when specified; – PreferredStartTime ≤ latestStartTime when specified.	CCSDS ASCII Time Code B (reference [1])	UTC
earliestStartTime	Optional earliest start time for a pass in UTC.	CCSDS ASCII Time Code B (reference [1])	UTC
latestStartTime	Optional latest start time for a pass in UTC. NOTE – If both start times are specified, then latest ≥ earliest.	CCSDS ASCII Time Code B (reference [1])	UTC
preferredDuration	Preferred pass duration: – preferredDuration ≥ minimumDuration when specified; – preferredDuration ≤ maximumDuration when specified.	xsd:unsignedInt	Secs
minimumDuration	Optional minimum duration for a pass. Defaults to preferredDuration.	xsd:unsignedInt	Secs

Parameter	Description	Data Type	Data Units
maximumDuration	Optional maximum duration for a pass. Defaults to preferredDuration. NOTE – If both duration values are specified, then maximum ≥ minimum.	xsd:unsignedInt	Secs

3.2.1.7 Class ApertureSelection

This class specifies an aperture for the provided service. In the context of SM CS-1, it contains only one ApertureReference class instance (see section 3.2.1.8) and the parameter described in table 3-6.

Table 3-6: Class ApertureSelection Parameters

Parameter	Description	Data Type	Data Units
priority	Optional priority for aperture preference if multiple ApertureSelection classes are instantiated. Lower values indicate higher priority, with 1 being the highest and the default.	xsd:integer > 0	n/a

3.2.1.8 Class ApertureReference

This class specifies the location of an aperture by the names of the site where it is located and the aperture. It does not contain any other classes, but contains the parameters described in table 3-7.

Table 3-7: Class ApertureReference Parameters

Parameter	Description	Data Type	Data Units
siteRef	Site name. Permitted values registered in SANA.	xsd:string	n/a
apertureRef	Optional aperture identifier. Permitted values registered in SANA.	xsd:string	n/a

3.2.1.9 Class OnlineSrvPkgType1

This class specifies data relevant to an online service package. In the context of SM CS-1, it must contain one or more instances of the class OnlineSrvPkgType1Details (see section 3.2.1.10). Additionally, it contains the parameters described in table 3-8.

Table 3-8: Class OnlineSrvPkgType1 Parameters

Parameter	Description	Data Type	Data Units
sicfRef	Optional reference to the existing SICF. ***Not required or specified for SM CS-1.***	xsd:string	n/a
trajectoryRef	Optional reference to existing trajectory data. ***Not required or specified for SM CS-1.***	xsd:string	n/a
eventSequenceRef	Optional reference to existing event sequence.	xsd:string	n/a

3.2.1.10 Class OnlineSrvPkgType1Details

This class may be used to specify details relevant to an online service package. It does not contain any other classes but contains the parameter described in table 3-9.

Table 3-9: Class OnlineSrvPkgType1Details Parameters

Parameter	Description	Data Type	Data Units
configurationProfileRef	Reference to the existing configuration profile.	xsd:string	n/a

3.3 SERVICE PACKAGE DATA FORMATS

3.3.1 OVERVIEW

The following class diagram shows the classes of the Service Package Data Formats (SPDF) that are relevant to SM CS-1.

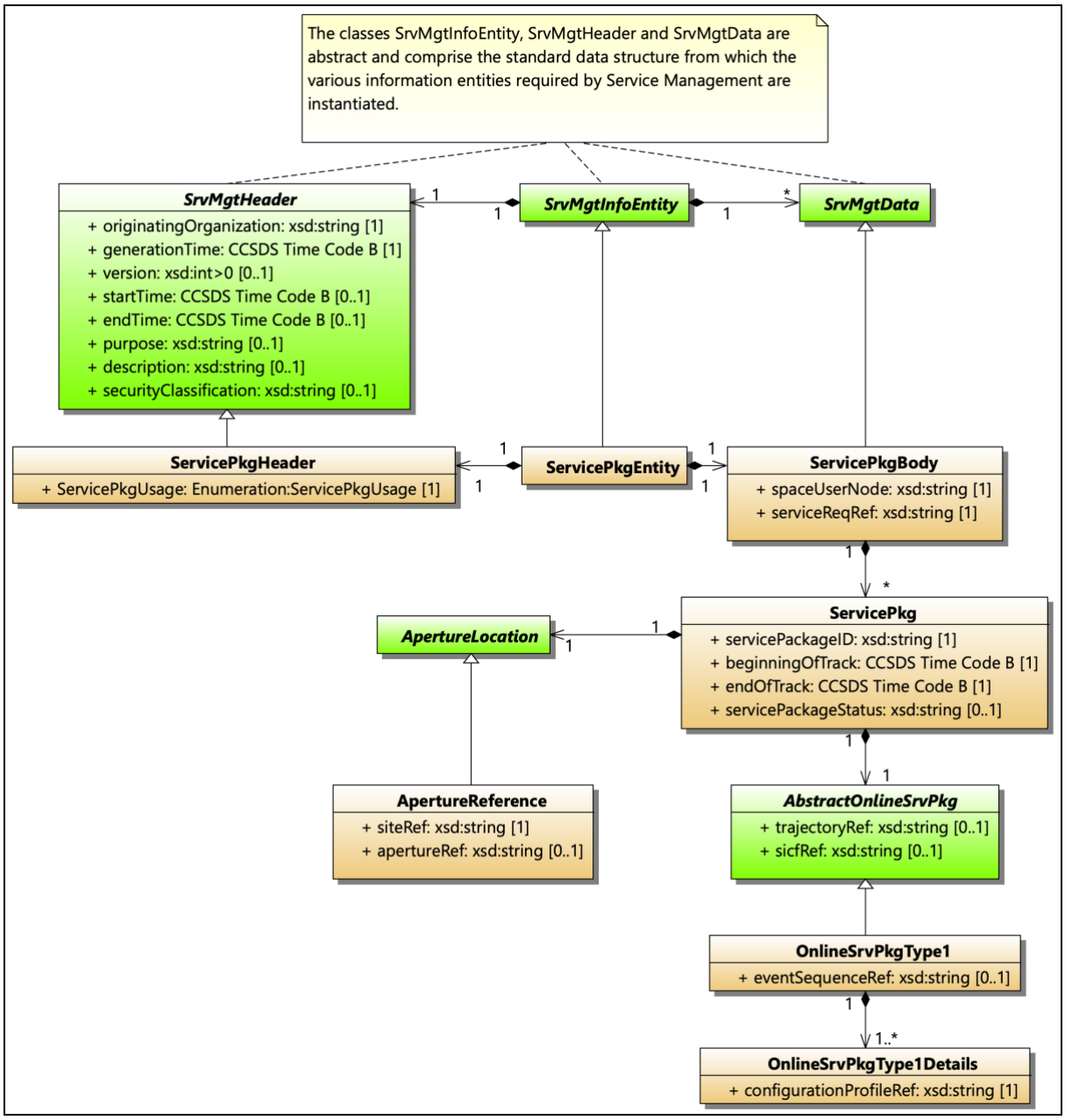

Figure 3-2: Service Management Capability Set-1 – Service Package Data Formats Class Diagram

The concrete classes shown in the above figure are described in the following sections. Full descriptions of all SPDF classes and parameters can be found in reference [5].

3.3.1.1 Class SrvPkgEntity

This class is the 'wrapper' for the Service Package data. It does not have any parameters but contains the following classes:

– One instance of the SrvPkgHeader class (see section 3.3.1.2);

– One instance of one of the SrvPkgBody class (see section 3.3.1.3).

3.3.1.2 Class SrvPkgHeader

This class contains the 'header' of the Service Package data. It does not contain any other classes, but contains the parameters described in table 3-10.

Table 3-10: Class SrvPkgHeader Parameters

Parameter	Description	Data Type	Data Units
`originatingOrganization`	Organization that generated the information entity. Permitted values registered in SANA.	xsd:string	n/a
`generationTime`	Generation time of information entity in CCSDS ASCII Time Code B format (reference [1])	xsd:string	UTC
`version`	Optional version number of the information entity. Increments when regenerated with the same concrete type, status, `startTime`, and `endTime`. Must increase, but not necessarily by 1.	xsd:integer > 0	n/a
`startTime`	Optional start time for the information entity in CCSDS ASCII Time Code B format (reference [1]). ***Not required or specified for SM CS-1.***	xsd:string	UTC
`endTime`	Optional end time for the information entity in CCSDS ASCII Time Code B format (reference [1]). ***Not required or specified for SM CS-1.***	xsd:string	UTC
`purpose`	Optional purpose of the information entity, specified in an ICD between relevant parties.	xsd:string	n/a
`description`	Optional description of the information entity contents, specified in an ICD between relevant parties.	xsd:string	n/a
`securityClassification`	Optional security classification. Definitions and use specified in an ICD between relevant parties.	xsd:string	n/a

Parameter	Description	Data Type	Data Units
servicePkgUsage	Mandatory. Intended usage of the Service Package set: – INTERFACE-TEST: test exchange purposes only; – OPERATIONAL: operational purposes; – OTHER: other uses.	Enumeration	N/A

3.3.1.3 Class ServicePkgBody

This class contains the 'body' of the Service Package data. It contains 0 or more instances of the ServicePkg class (see section 3.3.1.4) and the parameters described in table 3-11.

Table 3-11: Class ServicePkgBody Parameters

Parameter	Description	Data Type	Data Units
spaceUserNode	User of requested services. Must be spacecraft names as registered in SANA. NOTE – Must match the spaceUserNode specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	N/A
serviceReqRef	Reference to service request identifier. For SMURF Requests, matches the serviceReqID specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	N/A

3.3.1.4 Class ServicePkg

This class contains the information for an individual service package included within the ServicePkgBody (see section 3.3.1.3). It contains the following classes:

– One instance of the ApertureReference class (see section 3.3.1.5);

– One instance of the class OnlineSrvPkgType1 (see section 3.3.1.6).

Additionally, it contains the parameters described in table 3-12.

Table 3-12: Class ServicePkg Parameters

Parameter	Description	Data Type	Data Units
servicePackageID	Unique identifier of the service package, assigned by the provider CSSS.	xsd:string	N/A
beginningOfTrack	Scheduled start time for the service package.	CCSDS ASCII Time Code B (reference [1])	N/A
endOfTrack	Scheduled stop time for the service package.	CCSDS ASCII Time Code B (reference [1])	N/A

Parameter	Description	Data Type	Data Units
servicePackageStatus	Optional status indicator for service package (e.g., state machine tracking). Recommended values: CREATED, SCHEDULED, DELETED, ARCHIVED, EXECUTING, ABORTED. ***SM CS-1 does not report state transitions for requests.***	xsd:string	N/A

3.3.1.5 Class ApertureReference

See section 3.2.1.8 above.

3.3.1.6 Class OnlineSrvPkgType1

See section 3.2.1.9 above.

3.3.1.7 Class OnlineSrvPkgType1Details

See section 3.2.1.10 above.

3.4 SIMPLE SCHEDULE DATA FORMAT

3.4.1 OVERVIEW

The following class diagram shows the classes of the Simple Schedule Format (SSF) that are relevant to SM CS-1.

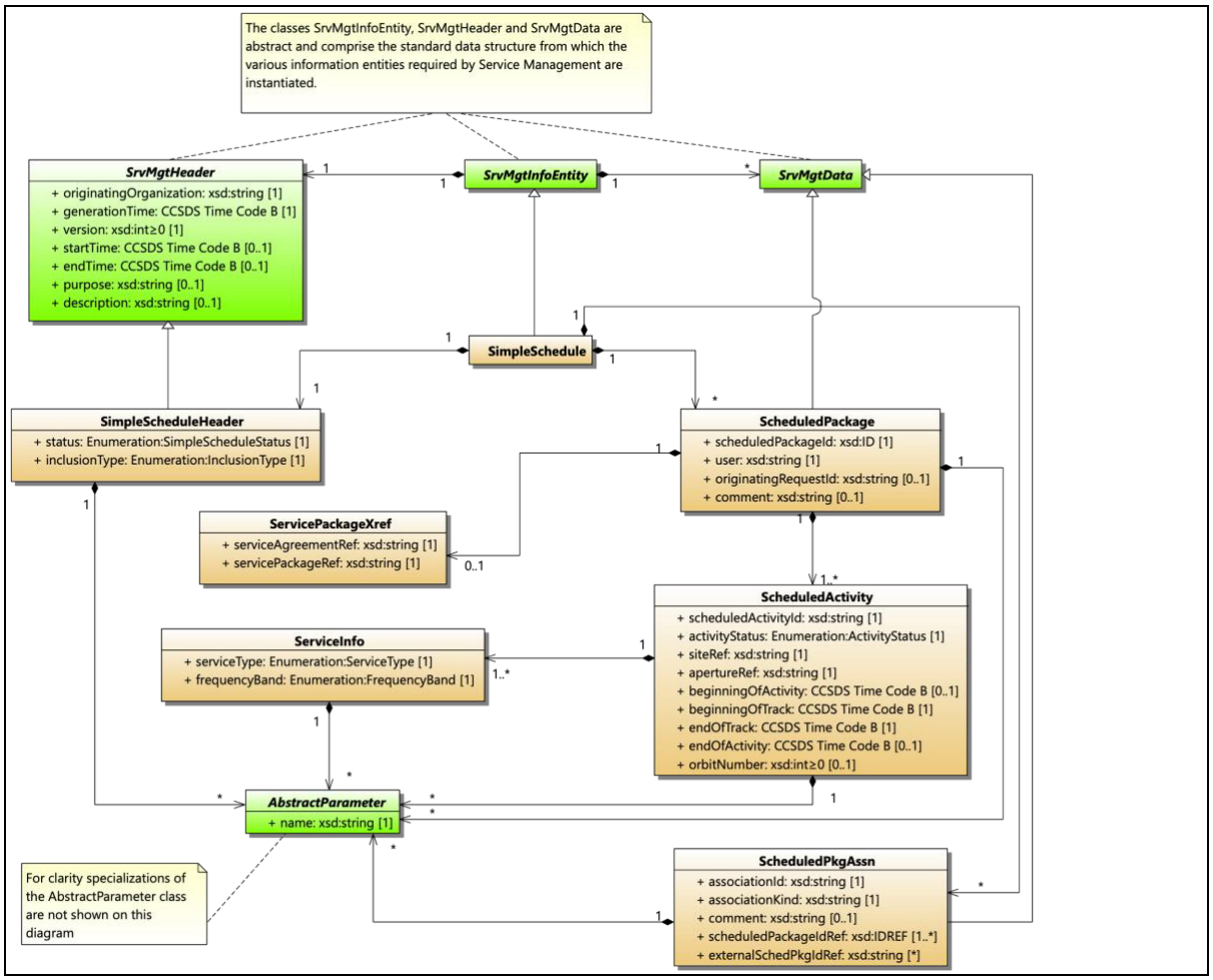

Figure 3-3: Service Management Capability Set-1 – Simple Schedule Data Format Class Diagram

The concrete classes shown in the above figure are described in the following sections. Full descriptions of all SSF classes and parameters can be found in Ref. [6].

3.4.1.1 Class SimpleSchedule

This class is effectively the ‘wrapper’ for the Simple Schedule Format. It does not have any parameters but contains the following classes:

– One instance of the SimpleScheduleHeader class (see section 3.4.1.2);

– Zero or more instances of the ScheduledPackage class (see section 3.4.1.3);

– Zero or more instances of the ScheduledPkgAssn class (see section 3.4.1.7).

3.4.1.2 Class SimpleScheduleHeader

This class contains the 'header' for the Simple Schedule Format and the following:

– Zero or more instances of parameter classes derived from the abstract AbstractParameter (see section 3.4.1.8) as well as parameters described in table 3-13. These parameters may be used to extend SimpleScheduleHeader class. Usage of any additional parameters is not within the scope of this document and should be specified in an ICD between the relevant parties.

Table 3-13: Class SimpleScheduleHeader Parameters

Parameter	Description	Data Type	Data Units
originatingOrganization	Organization that generated the information entity. Permitted values registered in SANA.	xsd:string	n/a
generationTime	Generation time of information entity in CCSDS ASCII Time Code B format (reference [1])	xsd:string	UTC
version	Optional version number of the information entity. Increments when regenerated with the same concrete type, status, startTime, and endTime. Must increase, but not necessarily by 1.	xsd:positiveInteger	n/a
startTime	Start time of the simple schedule in CCSDS ASCII Time Code B format (reference [1]).	xsd:string	UTC
endTime	End time of the simple schedule, in CCSDS ASCII Time Code B format (reference [1]).	xsd:string	UTC
purpose	Optional purpose of the information entity, specified in an ICD between relevant parties.	xsd:string	n/a
description	Optional description of the information entity contents, specified in an ICD between relevant parties.	xsd:string	n/a
status	Status of the information entity. Permitted values: – TEST: for test purposes only; – PROVISIONAL—provisional schedule; – OPERATIONAL—operational schedule; – OTHER—for other purposes that can be specified by purpose.	Enumeration	n/a

Parameter	Description	Data Type	Data Units
inclusionType	Inclusion type rule applied to the simple schedule. Permitted values are: – OVERLAP_INCLUSION – START_INCLUSION (see Ref. [6] for the description of both).	Enumeration	n/a

3.4.1.3 Class ScheduledPackage

This class contains the 'body' of the Simple Schedule data with the following classes:

– One or more instances of the ScheduledActivity class (see section 3.4.1.4);

– Zero or one instances of the class ServicePackageXRef (see section 3.4.1.6);

– Zero or more instances of parameter classes derived from the abstract AbstractParameter (see section 3.4.1.8). These parameters may be used to extend ScheduledPackage class. Usage of any additional parameters is not within the scope of this document and should be specified in an ICD between the relevant parties.

It also contains the parameters described in table 3-14.

Table 3-14: Class ScheduledPackage Parameters

Parameter	Description	Data Type	Data Units
scheduledPackageId	Unique identifier for every ScheduledPackage in a schedule. May change if a schedule is regenerated.	xsd:ID	n/a
user	User of the scheduled package, typically a spacecraft name registered in SANA. The following strings are also permitted: – UNALLOCATED: the time is unallocated – PROVIDER-CSSS: the time is allocated to the Provider CSSS (e.g., for tests, maintenance, upgrades, etc.). NOTE – Must match the spaceUserNode specified in the NewOnlineSrvPkgReq that created the request.	xsd:string	n/a
comment	Optionally provides ad-hoc information. Use specified in an ICD between relevant parties.	xsd:string	n/a
originatingRequestId	Optionally echoes the identifier from the request that generated the scheduled package (the serviceReqID specified in the NewOnlineSrvPkgReq that created the request).	xsd:string	n/a

3.4.1.4 Class ScheduledActivity

This class contains the activities that are scheduled in a scheduled package. It includes the following classes:

– One or more instances of the ServiceInfo class.

– Zero or more instances of parameter classes derived from the abstract AbstractParameter (see section 3.4.1.8). These parameters may be used to extend ScheduledActivity class if required. Usage of any additional parameters is not within the scope of this document and should be specified in an ICD between the relevant parties.

Additionally, it contains the parameters described in table 3-15.

Table 3-15: Class ScheduledActivity Parameters

Parameter	Description	Data Type	Data Units
`scheduledActivityId`	Unique identifier for every scheduled activity. May change if a schedule is regenerated.	xsd:string	n/a
`activityStatus`	Activity commitment level. Permitted values are: – COMMITTED: supported barring unforeseen circumstances, the activity will be supported; – TENTATIVE: scheduled but subject to change. When the ScheduledPackage—`user` is UNALLOCATED and the ServiceInfo—`serviceType` is UNUSED, one of the following must be used: – AVAILABLE: aperture can be scheduled during the UNUSED time. – UNAVAILABLE: aperture unavailable during the UNUSED time.	Enumeration	n/a
`siteRef`	Site for the scheduled activity. Permitted values registered in SANA.	xsd:string	n/a
`apertureRef`	Aperture identifier. Must be unique per site (ground station/relay satellite). Permitted values registered in SANA.	xsd:string	n/a
`beginningOfActivity`	Allocation start time, including setup activities. Optional but required if `endOfActivity` is present. Specified when available per agency policy, otherwise omitted.	CCSDS ASCII Time Code B (reference [1])	UTC
`beginningOfTrack`	Scheduled start time for spacecraft tracking.	CCSDS ASCII Time Code B (reference [1])	UTC
`endOfTrack`	Scheduled end time for spacecraft tracking.	CCSDS ASCII Time Code B (reference [1])	UTC
`endOfActivity`	Allocation end time, including teardown activities. Optional but required if `beginningOfActivity` is present. Specified when available per agency policy, otherwise omitted.	CCSDS ASCII Time Code B (reference [1])	UTC
`orbitNumber`	Optional orbit number on which `beginningOfTrack` occurs. Omitted if not relevant (e.g., deep space missions) or unavailable.	xsd:integer ≥ 0	n/a

3.4.1.5 Class ServiceInfo

This class specifies the type of scheduled activities. It contains the following:

– Zero or more instances of parameter classes derived from the abstract AbstractParameter (see section 3.4.1.8) and parameters described in table 3-16. These parameters may be used to extend ServiceInfo class. Usage of any additional parameters is not within the scope of this document and should be specified in an ICD between the relevant parties.

Table 3-16: Class ServiceInfo Parameters

Parameter	Description	Data Type	Data Units
`serviceType`	Service type for the activity. Permitted values: – APA-AZ/EL (Antenna Pointing Angles, Azimuth/Elevation); – APA-X/Y (Antenna Pointing Angles, X/Y); – DELTADOR; – DOPPLER; – OFFLINE-TM-RECORDING; – OFFLINE-TM-PROVISION; – RF-ONLY; – RANGING; – RESERVED; – TBD; – TELECOMMAND; – TELEMETRY; – TEST; – UNUSED; – VLBI (Very-Long-Baseline Interferometry).	Enumeration	n/a
`frequencyBand`	Frequency band for the service. Values per Frequency Band Designator (FBD) in table 2-1 of reference [7] and: – N/A (not applicable); – OTHER (band not defined in in table 2-1); – ALL (see ServiceInfo class usage).	Enumeration	n/a

3.4.1.6 Class ServicePackageXRef

This class is optional and may be used to map the contents of the Simple Schedule to the appropriate SCCS SM service packages (see section 3.3). It does not contain any other classes, but contains the parameters described in table 3-17.

Table 3-17: Class ServicePackageXRef Parameters

Parameter	Description	Data Type	Data Units
`serviceAgreementRef`	Service Agreement to which servicePackageRef applies.	xsd:string	n/a
`servicePackageRef`	Cross-reference to the service package containing the aperture allocation if obtained via SCCS SM. Several ScheduledActivity instances may have the same servicePackageRef since one service package can contain multiple aperture allocations.	xsd:string	n/a

3.4.1.7 Class ScheduledPkgAssn

This optional class may be used to specify associations between scheduled packages. This may be useful when there may be dependencies between scheduled packages on two or more antennas (e.g., Delta-Differential One-way Range [Delta-DOR] measurements) or when there are two or more spacecraft using the same aperture at the same time (Multiple Spacecraft per Aperture [MSPA]). It contains:

– Zero or more instances of parameter classes derived from the abstract AbstractParameter (see section 3.4.1.8) and the parameters described in table 3-18. These parameters may be used to extend ScheduledPkgAssn class. Usage of any additional parameters is not within the scope of this document and should be specified in an ICD between the relevant parties.

Table 3-18: Class ScheduledPkgAssn Parameters

Parameter	Description	Data Type	Data Units
`associationId`	Unique identifier for the association, assigned by the Provider CSSS. May change if schedule is regenerated.	xsd:string	n/a
`associationKind`	Association type. Permitted values registered in SANA	xsd:string	n/a
`comment`	Optionally provides ad hoc information. Use specified in an ICD between relevant parties.	xsd:string	n/a
`scheduledPackageIdRef`	List of the `scheduledPackageIds` (see table 3-14) for the associated scheduled packages.	Array of xsd:IDREF	n/a
`externalSchedPkgIdRef`	Optional association to scheduled packages not contained in the current file.	Array of xsd:string	n/a

NOTES

1 The xsd:IDREF type shall be used for an attribute that references an ID:

a) all attributes of type xsd:IDREF must reference an xsd:ID in the same XML document;

b) as with ID, an xsd:IDREF value must be an NCName.

2 For a ScheduledPkgAssn to be valid, it must refer to a least one scheduled package that is in the same xml document (file).

3 When a ScheduledPkgAssn must reference scheduled packages outside the current XML document (e.g., different time range or, in the case for Delta-DOR, different Provider CSSS), the `externalSchedPkgIdRef` parameter shall contain the references.

3.4.1.8 CLASS AbstractParameter (ABSTRACT)

The AbstractParameter abstract class is optional and may be used to instantiate parameters.

NOTE – The AbstractParameter abstract class is fully described in reference [8].

3.5 USAGE NOTES

This section provides notes on how the various aspects of SM should be used for SM CS-1.

3.5.1 LIFECYCLE

The lifecycle of the various data entities in SM CS-1 is described below:

1. The User CSSS submits a SMURF request of type `NewOnlineSrvPkgReq`. This request contains the information that the Provider CSSS needs to process it and find a suitable time for generating a service package compatible with specified constraints.

2. After processing a syntactically correct request, the Provider CSSS **accepts** the service package if it satisfies the constraints specified in the request or **rejects** it, if the constraints cannot be satisfied. A Service Package data entity is generated in either case:

    a. **Accepted**: The `ServicePkgBody` contains the details of the generated service package. The `servicePackageStatus` parameter in the `ServicePkg` is set to 'SCHEDULED' (recommended but not mandatory).

    b. **Rejected**: The `ServicePkgBody` is empty except for the `spaceUserNode` and `serviceReqRef` parameters.

3. Once a Service Package has been generated, the User CSSS may submit two additional SMURF requests to act on this. For SM CS-1, SM does not directly notify whether these requests were accepted or rejected, but the results may appear in the Simple Schedule:

    a. `ReplaceSrvPkg` enables the User CSSS to modify a Service Package generated by the Provider CSSS. For SM CS-1, this method can modify track timing (beginning and end) and configuration profile. Accepting these changes is at the discretion of the Provider CSSS, so they may not be implemented.

    b. `DeleteSrvPkg` enables the User CSSS to delete a Service package generated by the Provider CSSS.

4. The Simple Schedule Format (SSF) specifies a standard format for transferring scheduling information related to apertures at ground stations and/or relay satellites between space agencies and commercial or governmental spacecraft operators. It may be generated periodically by the Provider CSSS or on request. For SM CS-1, no standard mechanism exists for requesting to generate a Simple Schedule. Any such mechanism is subject to agreement between participating parties. It is expected that the provider will publish the SSF and make the current relevant SSF available to users. The Simple Schedule essentially reports for one or more Space Users Nodes (i.e., spacecraft) the following:

a. Status of an assigned tracking pass;

b. Aperture assigned at a site;

c. Beginning of track;

d. End of track;

e. Services to be supported during the tracking pass;

f. Other optional information may also be provided:

   i) Beginning of activity;

   ii) End of activity;

   iii) Orbit number;

   iv) Service agreement covering the activity;

   v) Reference of the service package containing the activity.

The Simple Schedule may also contain information about unallocated time or time allocated to the Provider CSSS for testing, maintenance, etc. Whether this information is provided depends on the Provider CSSS policies.

ANNEX A

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

As this recommended practice does not specify any new standards but rather describes how to use those specified in other CCSDS books, there are no explicit security, SANA or patent considerations, but rather, the appropriate sections in the following references should be consulted for the relevant considerations:

- Ref . [4] Annex B Security, SANA and Security Considerations;
- Ref . [5] Annex B Security, SANA and Security Considerations;
- Ref . [6] Annex B Security, SANA and Security Considerations.

ANNEX B

XML SCHEMA ORGANIZATION AND PACKAGING FOR THE VARIOUS INFORMATION ENTITY CLASSES

(INFORMATIVE)

As this recommended practice does not specify any new XML schemas but rather describes how to use those specified in other CCSDS books the appropriate sections in the following references should be consulted for the relevant information:

- Ref. [4] Annex C XML Schema Organization and Packaging for the Cross Support Service Management—Service Management Utilization Request Formats. Classes.

- Ref. [5] Annex C XML Schema Organization and Packaging for the Cross Support Service Management—Service Package Data Formats Classes.

- Ref. [6] Annex C XML Schema Organization and Packaging for the Simple Schedule Format Specification Classes.

B1 EXAMPLES

Several XML examples illustrating the usage of the SM CS described in this Recommended Practice can be found in GitHub at the following URL along with the required schemas:

https://github.com/CCSDS-CSSM/902Releases/releases/tag/CESG_Review-20250922

ANNEX C

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

Term	Meaning
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CS	Capability Set
CSSS	Cross Support Service System
DDOR	Delta-Differential One-Way Ranging
DOR	Differential One-Way Ranging
FBD	Frequency Band Designator
ICD	Interface Control Document
ID	Identifier
MSPA	Multiple Spacecraft per Aperture
n/a	not applicable
OMG	Object Management Group
SANA	Space Assigned Numbers Authority
SCCS SM	Space Communication Cross Support Service Management
SICF	Service Instance Configuration File
SM	Service Management
SM CS-1	Service Management Capability Set 1
SMURF	Service Management Utilization Request Formats
SPDF	Service Package Data Formats
SSF	Simple Schedule Format
UML	Unified Modelling Language
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
XSD	XML Schema Definition