The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS

DRAFT RECOMMENDED STANDARD

CCSDS 902.9-R-1

RED BOOK

September 2022

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS

DRAFT RECOMMENDED STANDARD

CCSDS 902.9-R-1

RED BOOK

September 2022

AUTHORITY

Issue:	Red Book, Issue 1
Date:	September 2022
Location:	Not Applicable

(WHEN THIS RECOMMENDED STANDARD 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 STANDARD 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 **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - -- The **standard** itself.
  - -- The anticipated date of initial operational capability.
  - -- The anticipated duration of operational service.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** 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 Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard 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:

http://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'Etudes 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.
- Federal Space Agency (FSA)/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 Standard. 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 organizations. As such, its technical contents are not stable, and several iterations of it 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 902.9-R-1	Service Management Utilization Request Formats, Draft Recommended Standard, Issue 1	September 2022	Current proposed draft

CONTENTS

CONTENTS (continued)

Figure Page

Table

CONTENTS (continued)

Table Page

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

1.1.1 PURPOSE

The Service Management Utilization Request Format (SMURF) Recommended Standard specifies the formats of the data entities that are used to submit requests in the scope of CCSDS Service Management.

These requests can be categorized into five types:

a) Planning Information Request—can be used to submit a query for various types of planning information.

b) Report Request—can be used to submit a request for a report to be returned. Currently one type of report is supported: Simple Schedule (see reference [4]).

c) Information Request—can be used to submit a request for one or more of the following types of information:

   – configuration profile(s);

   – event sequence(s);

   – service agreement(s);

   – service package(s);

   – trajectory(s);

   – Delta-Differential One-Way Ranging (DDOR) scan pattern(s);

   – service instance configuration file(s).

   The information that is being requested is specified by means of a cross reference to the identifiers of the data entities that were previously defined.

d) Service Package Request—can be used to submit requests relating to service packages. The supported types of service package requests are:

   – New Online Request—allows the submission of a new online service package request;

   – Replace Online Request—allows the replacement of a previously submitted online service package request;

   – New Offline Request—allows the submission of a new offline service package request;

   – Replace Offline Request—allows the replacement of a previously submitted offline service package request.

- Replace Service Package Request—allows the replacement of a service package resulting from a previously submitted request.
- Delete Service Package Request Request—allows the deletion of a previously submitted request.
- Delete Service Package Request—allows the deletion of some or all of the service packages resulting from a previously submitted request.

e) Submission Request—can be used to submit data entities that are used as part of Service Management. Currently 5 types of submission request are foreseen:

- Trajectory Submission Request—enables the submission of a trajectory prediction. If the ID of the Trajectory Submission request matches that of one previously submitted, then the previous trajectory data is overwritten with the newly specified data.
- Configuration Profile Submission Request—enables the submission of a configuration profile. If the ID of the Configuration Profile Submission request matches that of one previously submitted, then the previous Configuration Profile data is overwritten with the newly specified data.
- Event Sequence Submission Request—enables the submission of a event sequence. If the ID of the Event Sequence Submission request matches that of one previously submitted, then the previous Event Sequence data is overwritten with the newly specified data.
- DDOR Scan Pattern Submission Request—enables the submission of a DDOR Scan Pattern. If the ID of the DDOR Scan Pattern Submission request matches that of one previously submitted, then the previous DDOR Scan Pattern data is overwritten with the newly specified data.
- Service Instance Configuration File (SICF) Submission Request—enables the submission of a SICF. If the ID of the SICF Submission request matches that of one previously submitted, then the previous Event Sequence data is overwritten with the newly specified data.

1.1.2 SCOPE

The scope of this Recommended Standard is limited to the exchange of Service Management Utilization Requests required in the context of CCSDS Service Management. In figure 1-1, the SMURF Recommendation is put into context with the various standards that together form the Space Communication Cross Support Service Management.

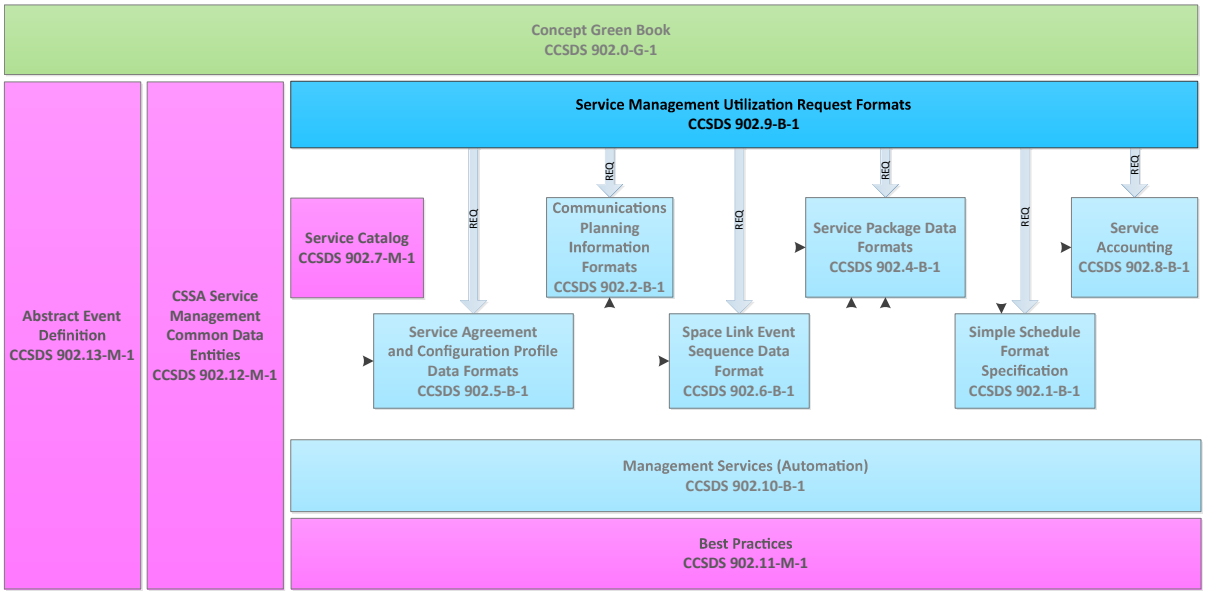

Figure 1-1: Service Management Utilization Requests Formats in the Context of Space Communication Cross Support Service Management

1.2 APPLICABILITY

This Recommended Standard is applicable only to the SMURF and contents, but not to its transmission. With respect to the transmission of the Planning Information between agencies and operators there are two scenarios:

a) The first is where the service management utilization request is exchanged within the context of Service management. Details of the transmission mechanism to be used in this case will be described in a forthcoming CCSDS recommendation.

b) The second deals with the case where the service management utilization request is exchanged outside the scope of Service Management. Here the mechanism by which the Planning Information is transmitted is outside of the scope of this document and should be specified in an ICD agreed by the parties involved.

1.3 RATIONALE

1.3.1 GENERAL

The primary goal of CCSDS is to increase the level of interoperability among Agencies. This Recommended Standard furthers that goal by establishing the means to exchange planning information relating to where most cross support activity occurs: between the tracking stations or ground data handling systems of various Agencies and the mission specific components of a mission ground system.

The use cases described in the following subsections were considered in deriving this Recommended Standard.

1.3.2 USE CASES

1.3.2.1 Overview

The Service Management Utilization Request Formats (SMURF) information entity serves a number of use cases. These can be categorized by the various types of requests defined by the SMURF and are discussed below.

1.3.2.2 Planning Information Request

The Planning Information Request support the submission of a query for various types of planning information as defined in reference [5].

The Planning Information Request allows the specification of constraints that can be used to determine the suitability of tracking assets, and also permits cross-referencing to trajectory and configure profiles so that the requested planning information can be accurately calculated. Planning information can be useful in the following cases:

1.3.2.3 Mission Design

During the mission design phase the planning information request can be used to obtain an idea of the feasibility of the required support, for instance by checking that there is sufficient coverage available from tracking assets to support the mission needs. It would also be possible, depending on the capabilities of the Provider Cross Support Service System (CSSS), to obtain estimates of the data rates that and be supported and check if there were any issues with RFI. In the longer term it would also be possible to obtain an estimate of the costs inherent in the tracking coverage being requested from the Provider CSSS.

1.3.2.4 Mission Planning

During the planning phase, the planning information request provides the mechanism for the User CSSS to obtain data about feasible tracking coverage that would then be fed into the detailed planning process for the mission.

The output from the User CSSS planning process would then be Service Package Requests (see 1.3.2.8).

1.3.2.5 Report Request

The Report Request enable a User CSSS to request a report from the Provider CSSS covering a specified time range. Currently two types of report are foreseen:

- Simple Schedule (see reference [4]);
- Accounting Report (see TBD).

Here the uses cases are that by obtaining a Simple Schedule report a user CSSS is able to see (depending on the capabilities of the Provider CSSS) what the schedule of the Provider CSSS tracking assets is and if there are any unallocated tracking opportunities that could be utilized to provide additional coverage.

The accounting report, when available, will enable a User CSSS to verify that the support provided was in line with what is in the service agreement with respect to data volume returned, lost TM frames, lost commands, etc.

1.3.2.6 Information Request

The Information Request is intended as a mechanism that enables the User CSSS to request various types of information from the Provider CSSS. Each Information Request can specify that one or more of the following types of information be provided:

- configuration profile(s);
- event sequence(s);
- service agreement(s);
- service package(s);
- trajectory(s);
- scenario set(s);
- DDOR scan pattern(s);
- service instance configuration file(s).

The information that is being requested is specified by means of a cross reference to the identifier(s) of the data entity(ies) that was previously defined.

Using the Information Request it is possible for the User CSSS to check that the information that the Provider CSSS has is correct.

1.3.2.7 Submission Request

In the scope of service management in order for the Provider CSSS to carry out the required calculations for scheduling tracking passes, configuring the equipment and carrying out activities during a pass it is necessary for the appropriate information to be provided from the User CSSS. Some of this information may be defined when the Service Agreement is set up (e.g., configuration profiles) but this may need to be adjusted during the lifetime of a mission. Other information, for example, trajectory information and event sequences, will need to be updated on a more regular basis.

With this in mind the Submission Request allows the User CSSS to submit data entities to the provider CSSS. Currently 5 types of submission request are foreseen:

- Trajectory Submission Request—enables the submission of a new trajectory prediction.
- Configuration Profile Submission Request—enables the submission of a new configuration profile.
- Event Sequence Submission Request—enables the submission of a new event sequence.
- DDOR Scan Pattern Submission Request—enables the submission of a new DDOR Scan Pattern.
- SICF Submission Request—enables the submission of a new SICF.

1.3.2.8 Service Package Request

Service Package requests allow the User CSSS to submit requests to the Provider CSSS that are related to service packages.

Typically Service Package Requests are issued during the mission planning process of the User CSSS and may be (but not mandatorily) based on plans that have been produced by the User CSSS using Planning Information obtained by means of prior Planning Information requests.

The Service Packages Requests allow the detailed specification of tracking constraints, equipment configuration and pass operations (via event sequences). The generation of Service Packages Requests may go through a number of cycles, starting months before the actual activities are expected to take place and being finalized a few days before execution,

hence the need for the Update Request. During these planning cycles the status of the Service Package Request can change with the initial submission being of a ‘provisional’ request that will probably change. The final request provided is typically considered as ‘operational’ and can be expected to result in the actual Service Package that will be executed, subject to unexpected events such as equipment failure or spacecraft.

1.4 DOCUMENT STRUCTURE

This document is organized as follows:

a) Section 1 provides the purpose, scope, applicability, and rationale of this Recommended Standard and identifies the conventions and references used throughout the document. This section also describes how this document is organized. A brief description is provided for each section and annex so that the reader will have an idea of where information can be found in the document. It also identifies terminology that is used in this document but is defined elsewhere.

b) Section 2 provides a brief overview of the CCSDS-recommended Service Management Utilization Request formats.

c) Section 3 provides details about the structure and content of the Service Management Utilization Request.

d) Annex A contains the Implementation Conformance Statement (ICS) proforma.

e) Annex B discusses security, SANA and patent considerations.

f) Annex C provides an informative description of the XML schema organization and packaging as well as the location of examples for the Service Management Utilization Request formats.

g) Annex D contains a list of Acronyms applicable to the Service Management Utilization Request formats.

1.5 DEFINITIONS

For the purposes of this document, the following definitions apply:

a) the word ‘agencies’ may also be construed as meaning ‘satellite operators’ or ‘satellite service providers’;

b) the notation ‘n/a’ signifies ‘not applicable’.

c) the term ‘Pass’ in this document may be construed as meaning a spacecraft being tracked by a one or more ground or space based apertures. During this tracking one or more services are provided between the spacecraft and the aperture(s). These services may be telemetry, telecommanding, tracking, DDOR, etc.

1.6 NOMENCLATURE

1.6.1 NORMATIVE TEXT

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

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 set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.7 CONVENTIONS—THE UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) diagrams used in the specification (including class diagrams, package diagrams, sequence diagrams, and activity diagrams) follow the notation, semantics, and conventions imposed by the Version 2.4.1 UML specification of the Object Management Group (OMG) (reference [2]).

Within the document use is made only of class diagrams. 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.

1.8 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] *Cross Support Service Management—Service Agreement and Configuration Profile Formats*. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.5-B-1. Proposed.

[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—Communications Planning Information Format*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.2-B-1. Washington, D.C.: CCSDS, March 2022.

[6] *Abstract Event Definition*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 902.13-M-1. Washington, D.C.: CCSDS, February 2021.

[7] *Cross Support Service Management—Common Data Entities*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 902.12-M-1. Washington, D.C.: CCSDS, February 2021.

[8] *Cross Support Service Management—Space Link Event Sequence Data Format*. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.6-B-1. Proposed.

2 OVERVIEW

2.1 GENERAL

This section provides a high-level overview of the CCSDS-recommended Service Management Utilization Request Formats, which is designed to facilitate standardized exchanges of service management utilization requests between space agencies.

2.2 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS

The service management utilization requests consists of files that are XML formatted. The formats of these files are suitable for automated interaction and/or (by means of a suitable XML viewer) human interaction.

Service management utilization requests information is either mandatory, in which case suitable values must be present, or optional, in which case values may be present or not. In addition it is possible to extend the contents of the various service management utilization formats by defining additional parameters. The content of any parameters defined is outside the scope of this document and should be documented in an ICD agreed by the involved parties.

2.3 MAPPING TO W3C XML SCHEMA

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

3 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS

3.1 OVERVIEW

The SMURF forms the basis from which various types of requests required by Service Management are derived.

3.2 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS CONTENT/STRUCTURE

3.2.1 OVERVIEW

Figure 3-1 presents the SMURF class diagram. The attributes of each class are described further in the following subsections and tables.

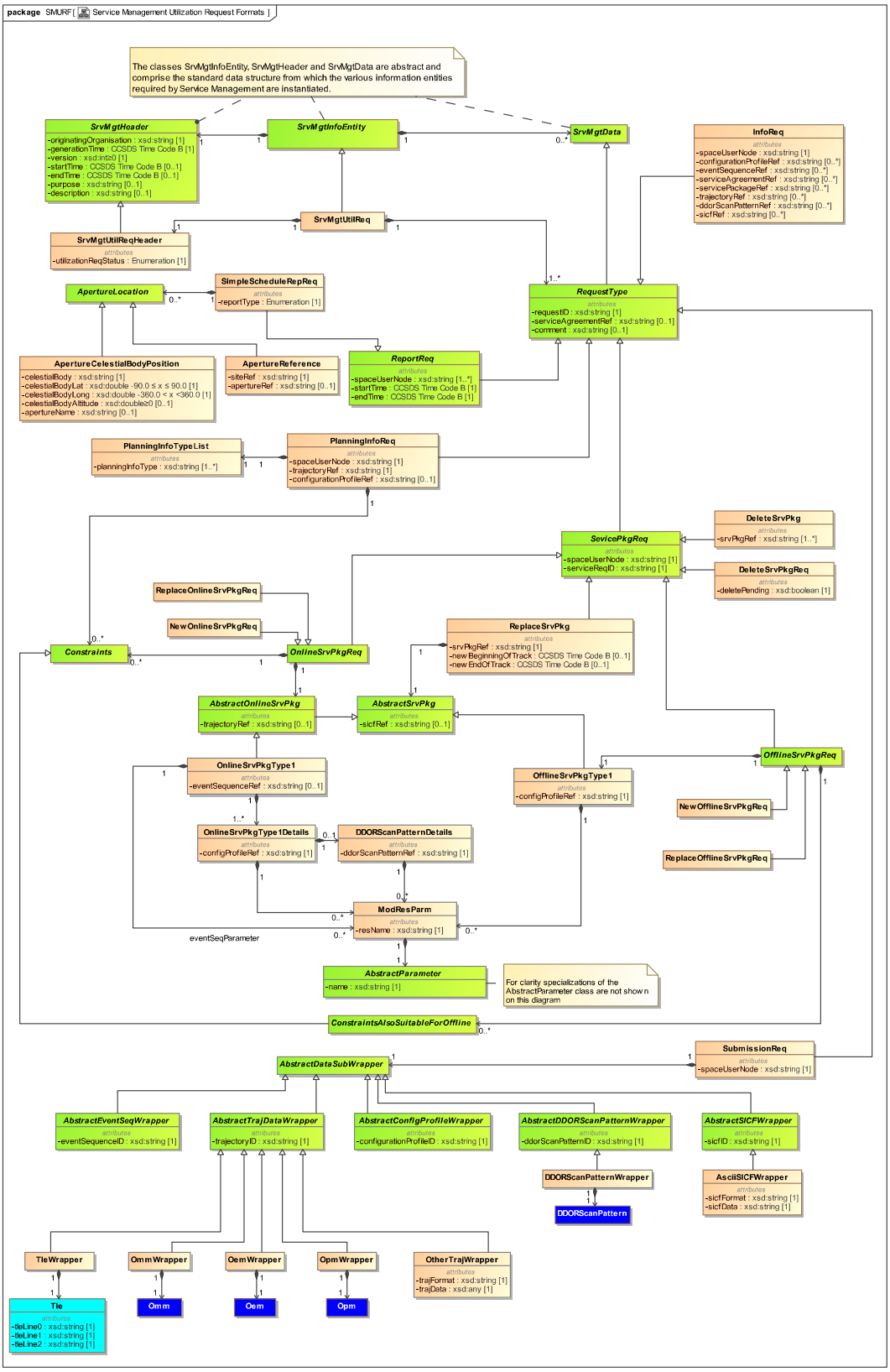

Figure 3-1: Service Management Utilization Request Formats Class Diagram

3.2.2 CLASS SrvMgtUtilReq

3.2.2.1 The SrvMgtUtilReq class is mandatory and shall constitute the Service Management Utilization Request information entity.

NOTE – The SrvMgtUtilReq class is a specialization and of class SrvMgtInfoEntity described in reference [7], and the generic description of the parameters can be found there.

3.2.2.2 A SrvMgtUtilReq class shall contain the following:

a) one, and only one, instance of the SrvMgtUtilReqHeader class;

b) one or more instances of a class derived from the abstract RequestType class.

3.2.2.3 The SrvMgtUtilReq class shall contain no additional parameters.

3.2.3 CLASS SRVMGTUTILREQHEADER

3.2.3.1 The SrvMgtUtilReqHeader class is mandatory and shall constitute the 'header' of the Service Management Utilization Request Formats.

NOTE – The SrvMgtUtilReqHeader class is a specialization of class SrvMgtHeader described in reference [7], and the generic description of the parameters can be found there.

3.2.3.2 Each instance of the SrvMgtUtilReqHeader class may have zero or more instances of parameter classes derived from the abstract SrvMgtAbstractEvent class described in reference [7]. Usage of any additional parameters so specified is not within the scope of this document and should be specified in an ICD between by the relevant parties.

3.2.3.3 The use of the parameters in the SrvMgtHeader class in the scope of the SrvMgtUtilReqHeader class shall be as per reference [7], with the following exceptions:

a) `startTime`—usage of this parameter is request-type dependent and shall be as specified in the subsections on the various request types;

b) `endTime`—usage of this parameter is request-type dependent and shall be as specified in the subsections on the various request types.

3.2.3.4 The SrvMgtUtilReqHeader class shall contain the additional parameter specified in table 3-1.

Table 3-1: Class SrvMgtUtilReqHeader Additional Parameters

Parameter	Description	Data Type	Data Units
`utilizationReqStatus`	The status of the service management utilization request.	Enumeration 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.	n/a

3.2.4 CLASS RequestType (Abstract)

3.2.4.1 The RequestType is an abstract class that shall be used to instantiate the various classes of request required by service management in the context of planning and scheduling requests.

NOTE – The RequestType class is a specialization and of class SrvMgtData described in reference [7].

3.2.4.2 The RequestType class shall contain parameters as specified in table 3-2.

Table 3-2: Class RequestType Parameters

Parameter	Description	Data Type	Data Units
`requestID`	A unique request ID assigned by the requestor.	xsd:string	n/a
`serviceAgreementRef`	Optional Parameter. This can be used to specify a reference to the service agreement under which the requested services are to be provided.	xsd:string	n/a
`comment`	Optional Parameter. May be used for the provision of ad hoc information.	xsd:string	n/a

3.2.5 CLASS ServicePkgReq (Abstract)

3.2.5.1 The ServicePkgReq is an abstract class that shall be used to instantiate the various classes of request required by service management for manipulating service packages.

NOTE – The ServicePkgReq class is a specialization and of class RequestType described in 3.2.4.

3.2.5.2 In addition to the parameters of the abstract RequestType class defined in 3.2.4, the ServicePkgReq class shall contain parameters as specified in table 3-3.

Table 3-3: Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`spaceUserNode`	The user of the requested services. These must be spacecraft names as specified in SANA.	xsd:string—Permitted values registered in SANA (see B2.3 for further information).	n/a
`serviceReqID`	Service request identifier.	xsd:string	n/a

3.3 PLANNING INFORMATION REQUEST

3.3.1 OVERVIEW

Figure 3-2 shows the subset of classes that are applicable to the Planning Information Request.

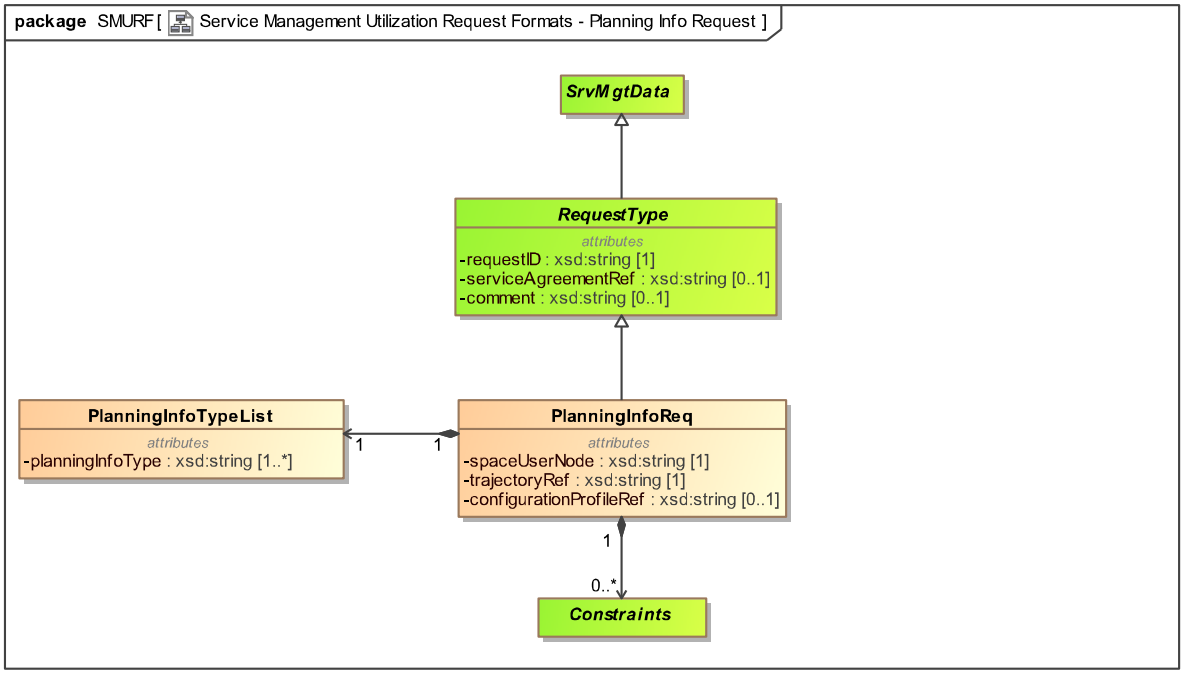

Figure 3-2: Service Management Utilization Request Formats Planning Information Request Class Diagram

3.3.2 USAGE OF SrvMgtUtilReqHeader `startTIME` AND `endTIME` PARAMETERS

In the context of the Planning Information Request the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are optional. If either `startTime` or `endTime` is `specified` they shall be interpreted as follows:

a) `startTime`—if specified, no planning information should be returned that has a time earlier than the time specified by this parameter;

b) `endTime`—if specified no planning information should be returned that has a time later than the time specified by this parameter.

3.3.3 CLASS PlanningInfoReq

3.3.3.1 The PlanningInfoReq class is optional and may be used to submit a query that can request various type of planning information.

NOTE – The PlanningInfoReq class is a specialization of class RequestType described in 3.2.4, and the generic description of the parameters is given in that subsection.

3.3.3.2 Each instance of the PlanningInfoHeader class must contain one and only one instance of the PlanningInfoTypeList as described in reference [7].

3.3.3.3 In addition to the parameters of the abstract RequestType class defined in 3.2.4, the PlanningInfoReq class shall contain parameters as specified in table 3-4.

Table 3-4: Class PlanningInfoReq Parameters

Parameter	Description	Data Type	Data Units
`spaceUserNode`	The user of the requested services. These must be spacecraft names as specified in SANA.	xsd:string—Permitted values registered in SANA (see B2.3 for further information).	n/a
`trajectoryRef`	This is used to specify a reference to the trajectory that the `spaceUserNode` wants the planning information for.	xsd:string	n/a
`configurationProfileRef`	Optional Parameter. This is used to specify a reference to the configuration profile that the `user` wants the information for. NOTE – This parameter is mandatory for the following `requestedInfo` types: – DATARATE – RFI	xsd:strings The format of the configuration reference is defined in reference [3]	n/a

3.3.4 CLASS PlanningInfoTypeList

The PlanningInfoTypeList class allows the specification of a list of Planning Info Types and is fully described in reference [7].

3.3.5 CLASS Constraints (ABSTRACT)

(See 3.10.2 for the definition Constraints class.)

3.4 REPORT REQUEST

3.4.1 OVERVIEW

The UML diagram in figure 3-3 shows the subset of classes that are applicable to the Report Request.

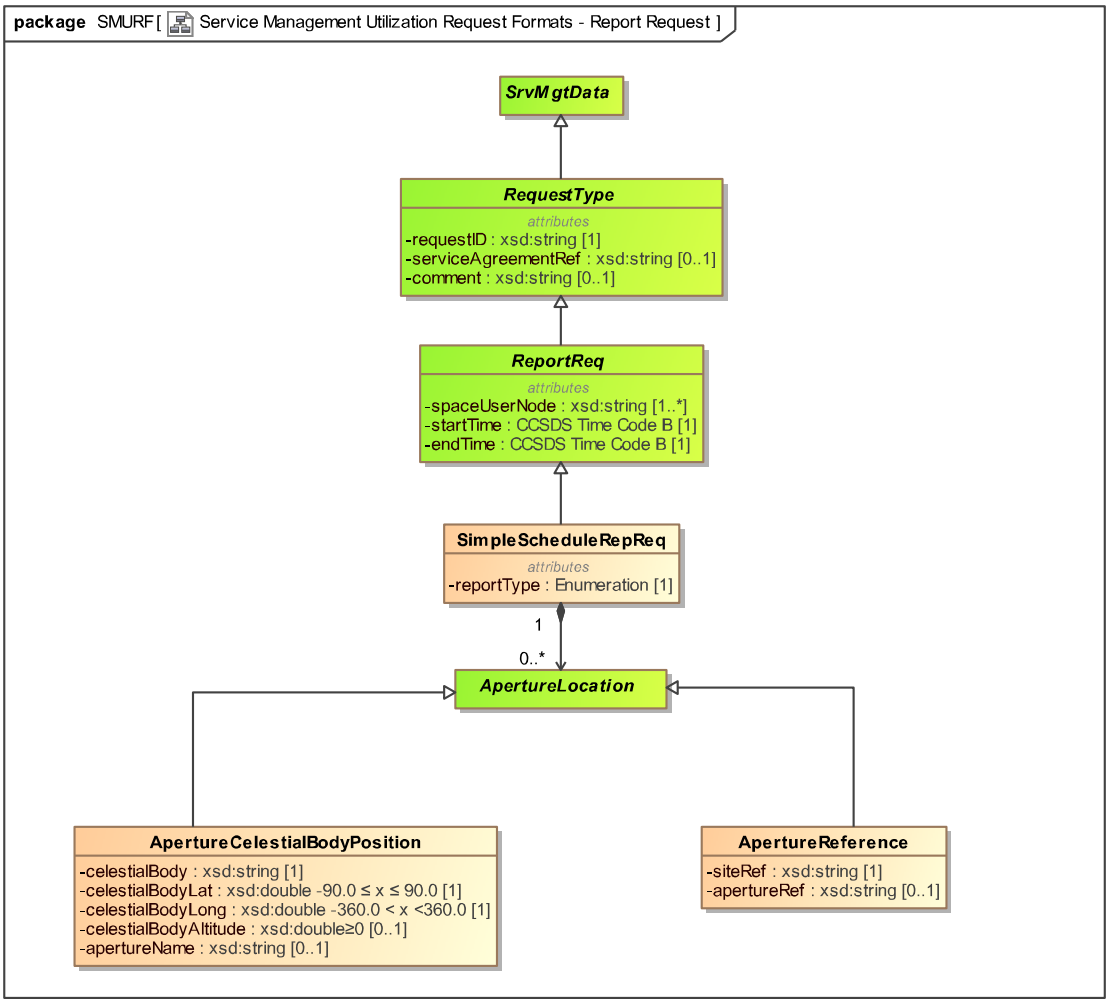

Figure 3-3: Service Management Utilization Request Formats Report Request Class Diagram

3.4.2 USAGE OF SrvMgtUtilReqHeader `startTime` AND `endTime` PARAMETERS

In the context of the Report Request the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are not required and should not be specified. If specified they should be ignored.

3.4.3 CLASS ReportReq (Abstract)

3.4.3.1 The ReportReq class is an abstract class that may be used to instantiate concrete report request classes.

NOTE – The ReportReq class is a specialization of class RequestType described in 3.2.4, and the generic description of the parameters is given in that subsection.

3.4.3.2 In addition to the parameters of the abstract RequestType class defined in 3.2.4, the ReportReq class shall contain parameters as specified in table 3-5.

Table 3-5: Class ReportReq Parameters

Parameter	Description	Data Type	Data Units
`spaceUserNode`	The user of the requested services. These must be spacecraft names as specified in SANA. At least one `spaceUserNode` must be specified.	Array of xsd:strings — Permitted values registered in SANA (see B2.3 for further information).	n/a
`startTime`	The start time to be used for the requested report.	CCSDS ASCII Time Code B (reference [1])	UTC
`endTime`	The end time to be used for the requested report.	CCSDS ASCII Time Code B (reference [1])	UTC

3.4.4 CLASS SimpleScheduleRepReq

3.4.4.1 The SimpleScheduleRepReq class is an optional class that may be used to request Simple Schedule reports.

NOTE – The SimpleScheduleRepReq class is a specialization of class ReportReq described in 3.4.2, and the generic description of the parameters is given in that subsection.

3.4.4.2 Each instance of the SimpleScheduleRepReq class may contain 0 or more instances of the ApertureReference class that is derived from the abstract ApertureLocation class.

NOTE – In the returned Simple Schedule the following is expected:

- Service Packages that start before the startTime but end after the startTime will be included in the schedule information.
- Service Packages that start before the endTime but end after the endTime will be included in the schedule information.

3.4.4.3 In addition to the parameters of the abstract RequestType class defined in 3.2.4, the SimpleScheduleRepReq class shall contain parameters as specified in table 3-6.

Table 3-6: Class SimpleScheduleRepReq Additional Parameters

Parameter	Description	Data Type	Data Units
`reportType`	This is used to specify the type of report required.	Enumeration—Currently supported values for simple schedule reports are: – SIM_SCH_SPECIFIC results in a Simple Schedule (see reference [4]) containing only information directly related to the requested `spaceUserNode(s)`; – SIM_SCH_FREE results in a Simple Schedule (see reference [4]) containing a schedule of all aperture free times insofar as this is permitted by the agencies policies. For this type of report, there must be exactly one `spaceUserNode` specified, and the value for this shall be; – UNALLOCATED.	n/a

3.4.5 CLASS ApertureLocation (ABSTRACT)

ApertureLocation is an abstract class that shall be used to instantiate the various classes of aperture location required by service management and is fully described in reference [7].

3.4.6 CLASS ApertureReference

3.4.6.1 The ApertureReference class allows the specification of the location of an aperture by the site name where it is located and the name of the aperture and is fully described in reference [7].

NOTE – The ApertureReference class is a specialization of class ApertureLocation mentioned in 3.4.5 and fully defined in reference [7].

3.4.7 CLASS ApertureCelestialBodyPosition

3.4.7.1 The ApertureCelestialBodyPosition class allows the specification of the location of an aperture by the site name where it is located and the name of the aperture and is fully described in reference [7].

NOTE – The ApertureCelestialBodyPosition class is a specialization of class ApertureLocation mentioned in 3.4.5 and fully defined in reference [7].

3.5 INFORMATION REQUEST

3.5.1 OVERVIEW

The UML diagram in figure 3-4 shows the subset of classes that are applicable to the Information Request.

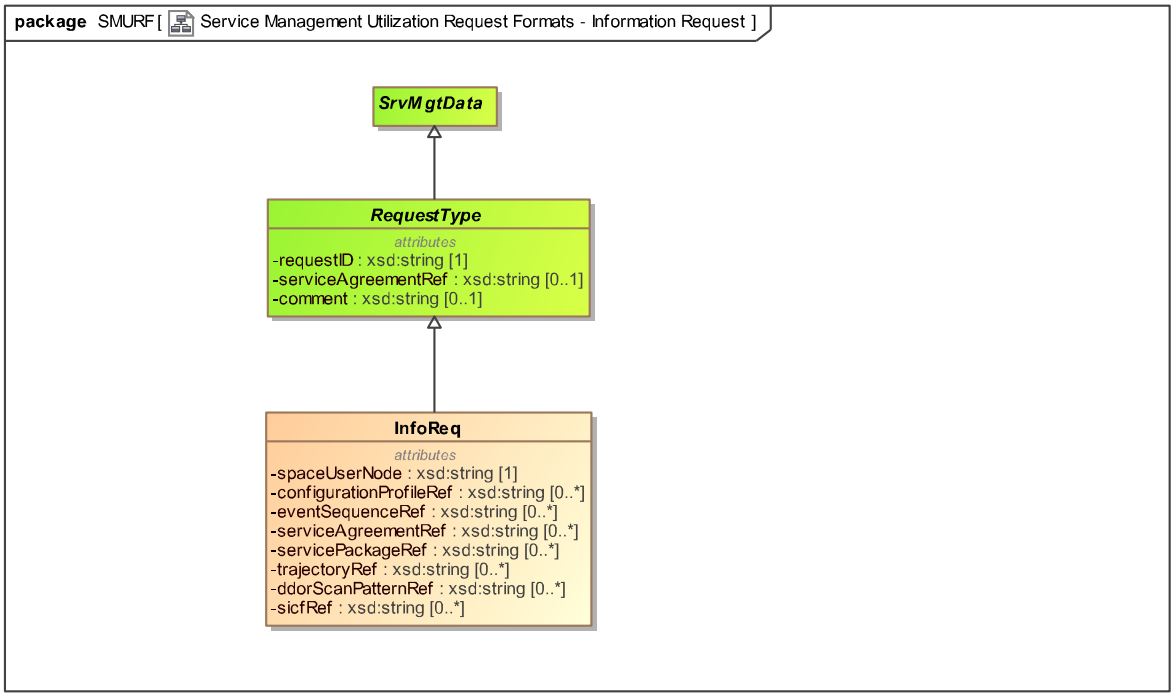

Figure 3-4: Service Management Utilization Request Formats Information Request Class Diagram

3.5.2 USAGE OF SrvMgtUtilReqHeader `startTime` AND `endTime` PARAMETERS

In the context of the Information Request the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are not required and should not be specified. If specified they should be ignored.

3.5.3 CLASS InfoReq

3.5.3.1 The InfoReq class is optional and may be used to submit a request for information about one or more of the following:

a) configuration profile(s);

b) event sequence(s);

c) service agreement(s);

d) service package(s);

e) trajectory(ies);

f) DDOR scan pattern(s);

g) service instance configuration file(s).

NOTE – The InfoReq class is a specialization of class RequestType described in 3.2.4, and the generic description of the parameters is given in that subsection.

3.5.3.2 It is expected that the Information Request will result in those data entities matching the request criteria being returned to the requestor.

3.5.3.3 In addition to the parameters of the abstract RequestType class defined in 3.2.4, the InfoReq class shall contain at least one of the parameters specified in table 3-7.

Table 3-7: Class InfoReq Parameters

Parameter	Description	Data Type	Data Units
`spaceUserNode`	The user of the requested services. These must be spacecraft names as specified in SANA.	xsd:string Permitted values registered in SANA (see B2.3 for further information).	n/a
`configurationProfileRef`	Optional Parameter.This is used to specify a reference or references to the configuration profile(s) for which the `spaceUserNode` wants the information.	Array of xsd:strings The format of the configuration cross-reference is defined in reference [3].	n/a
`eventSequenceRef`	Optional Parameter. This is used to specify a reference or references to the event sequence(s) for which the `spaceUserNode` wants the information.	Array of xsd:strings	n/a

Parameter	Description	Data Type	Data Units
serviceAgreementRef	Optional Parameter. This is used to specify a reference or references to the service agreement(s) for which the spaceUserNode wants the information.	Array of xsd:strings	n/a
servicePackageRef	Optional Parameter. This is used to specify a reference or references to the service package(s) for which the spaceUserNode wants the information.	Array of xsd:strings	n/a
trajectoryRef	Optional Parameter. This is used to specify a reference or references to the trajectory(ies) that the spaceUserNode wants the information for.	Array of xsd:strings	n/a
ddorScanPatternRef	Optional Parameter. This is used to specify a reference or references to the DDOR scan pattern(s) for which the spaceUserNode wants the information.	Array of xsd:strings	n/a
sicfRef	Optional Parameter. This is used to specify a reference or references to the Service Instance Configuration File(s) for which the spaceUserNode wants the information.	Array of xsd:strings	n/a

3.6 ONLINE NEW AND REPLACE SERVICE PACKAGE REQUESTS

3.6.1 OVERVIEW

The UML diagram in figure 3-5 shows the subset of classes that are applicable to the Online New and Replace Service Package Requests.

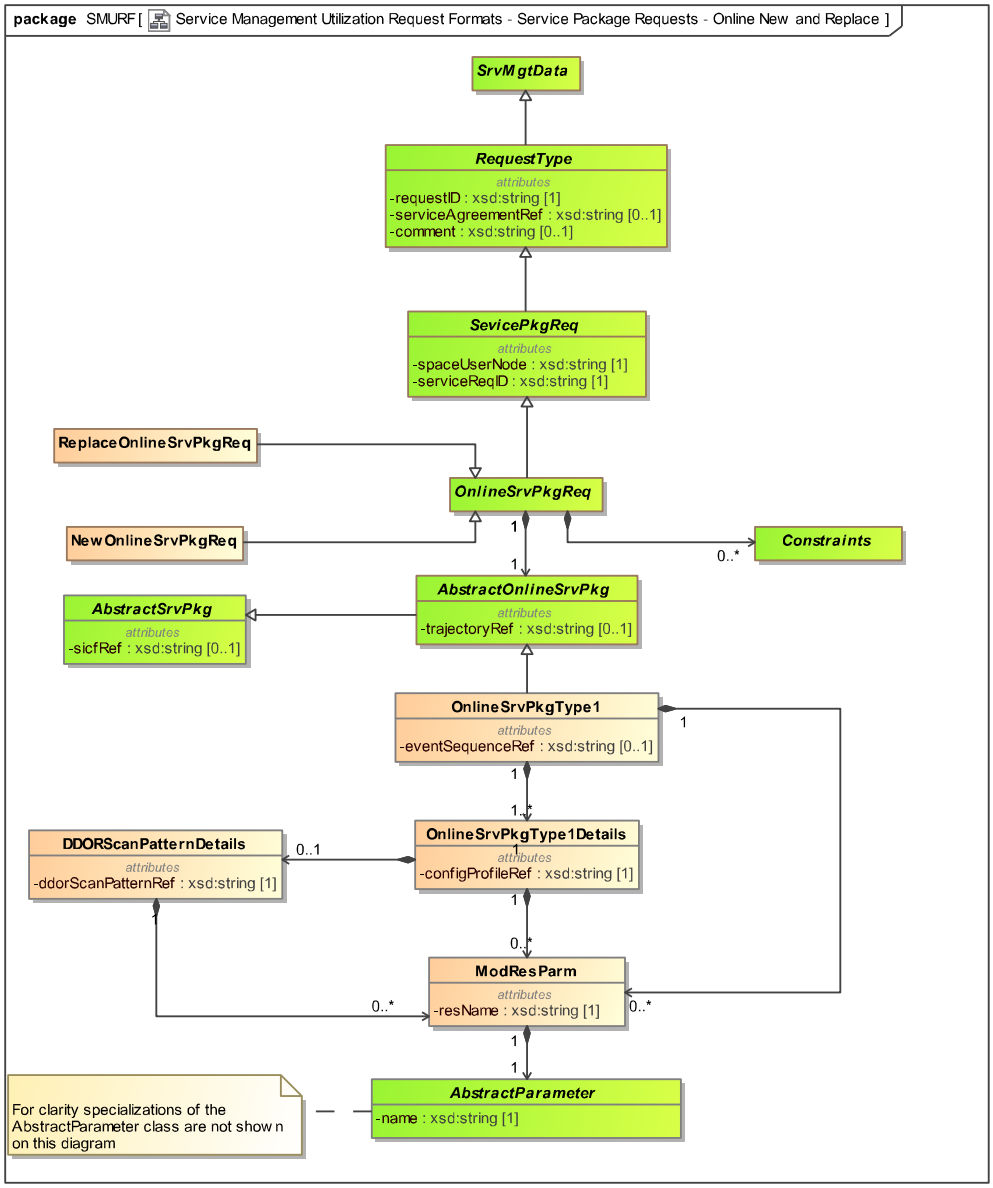

Figure 3-5: Service Management Utilization Request Formats Online New and Replace Service Package Requests Class Diagram

3.6.2 USAGE OF SrvMgtUtilReqHeader `startTime` AND `endTime` PARAMETERS

In the context of the Online New and Replace Service Package Request the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are not required and should not be specified. If specified they should be ignored. Any timing constraints required shall be defined in one or more of the classes derived from the abstract Constraints class (see 3.11).

3.6.3 CLASS OnlineSrvPkgReq (Abstract)

3.6.3.1 The OnlineSrvPkgReq class is an abstract class and shall be used to derive the classes required to submit a new online service package request or replace an existing online service package request.

NOTE – The OnlineSrvPkgReq class is a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.6.3.2 Each instance of a class derived from the OnlineSrvPkgReq class shall contain 1, and only 1, instance of a class derived from the AbstractOnlineSrvPkg class defined in 3.6.7.

3.6.3.3 Each instance of a class derived from the OnlineSrvPkgReq class may contain zero or more instances of classes derived from the Constraints class defined in 3.6.13.

3.6.3.4 The use of the parameters in the ServicePkgReq class in the scope of the OnlineSrvPkgReq class shall be as per tables 3-2 and 3-3.

3.6.4 CLASS NewOnlineSrvPkgReq

3.6.4.1 The NewOnlineSrvPkgReq class is optional and may be used to submit a new online service package request.

NOTES

1 The NewOnlineSrvPkgReq class is a specialization of class OnlineSrvPkgReq described in 3.6.2, and the generic description of the parameters is given in that subsection.

2 The OnlineSrvPkgReq class is itself a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.6.4.2 The NewOnlineSrvPkgReq class shall contain the parameters in the abstract OnlineSrvPkgReq class defined in 3.6.2.

3.6.4.3 In the context of the NewOnlineSrvPkgReq class, usage of the OnlineSrvPkgReq and ServicePkgReq parameters defined in table 3-3 shall be as specified in table 3-8.

Table 3-8: Class NewOnlineSrvPkgReq use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the NewOnlineSrvPkgReq class, this parameter shall specify a unique identifier (note).	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.			

3.6.5 CLASS ReplaceOnlineSrvPkgReq

3.6.5.1 The ReplaceOnlineSrvPkgReq class is optional and may be used to replace an existing online service package request.

NOTES

1 The ReplaceOnlineSrvPkgReq class is a specialization of class OnlineSrvPkgReq described in 3.6.2, and the generic description of the parameters is given in that subsection.

2 The OnlineSrvPkgReq class is itself a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.6.5.2 The ReplaceOnlineSrvPkgReq class contains the parameters in the abstract OnlineSrvPkgReq class defined in 3.6.2.

3.6.5.3 In the context of the ReplaceOnlineSrvPkgReq class, usage of the OnlineSrvPkgReq and ServicePkgReq parameters defined in table 3-3 shall be as specified in table 3-9.

Table 3-9: Class ReplaceOnlineSrvPkgReq use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the ReplaceOnlineSrvPkgReq class, this parameter must refer to the identifier of the previously submitted request that is to be updated.	xsd:string	n/a

3.6.6 CLASS AbstractSrvPkg (ABSTRACT)

The AbstractSrvPkg class is abstract and shall be used to derive the various service package classes required in service management.

NOTE – The AbstractSrvPkg class is fully described in reference [7].

3.6.7 CLASS AbstractOnlineSrvPkg (ABSTRACT)

The AbstractOnlineSrvPkg class is abstract and shall be used to derive the various online service package classes required in service management.

NOTES

1 The AbstractOnlineSrvPkg class is a specialization of class AbstractSrvPkg described in 3.6.6.

2 The AbstractSrvPkg class is fully described in reference [7].

3.6.8 CLASS OnlineSrvPkyType1

The OnlineSrvPkyType1 class shall be used to specify the contents of an online service package.

NOTES

1 The OnlineSrvPkyType1 class is a specialization of class AbstractOnlineSrvPkg described in 3.6.7.

2 The OnlineSrvPkyType1 class is fully described in reference [7].

3.6.9 CLASS OnlineSrvPkyType1Details

The OnlineSrvPkyType1Details class shall be used to specify details relevant to an online service package.

NOTE – The OnlineSrvPkyType1Details class is fully described in reference [7].

3.6.10 CLASS DDORScanPatternDetails

The DDORScanPatternDetails class shall be used to specify a reference to an existing DDOR Scan Pattern.

NOTE – The DDORScanPatternDetails class is fully described in reference [7].

3.6.11 CLASS ModResParm

The ModResParm class may be used to modify the value of a parameter of a resource. The Resource for which the parameter value is to be modified shall be specified by the resource name.

NOTE – The ModResParm class is fully described in reference [7].

3.6.12 CLASS AbstractParameter (ABSTRACT)

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

NOTE – The AbstractParameter class is fully described in reference [6].

3.6.13 CLASS CONSTRAINTS (ABSTRACT)

(See 3.10.2 for the definition of this class.)

3.7 OFFLINE NEW AND REPLACE SERVICE PACKAGE REQUESTS

3.7.1 OVERVIEW

The UML diagram in figure 3-6 shows the subset of classes that are applicable to the Offline New and Replace Service Package Requests.

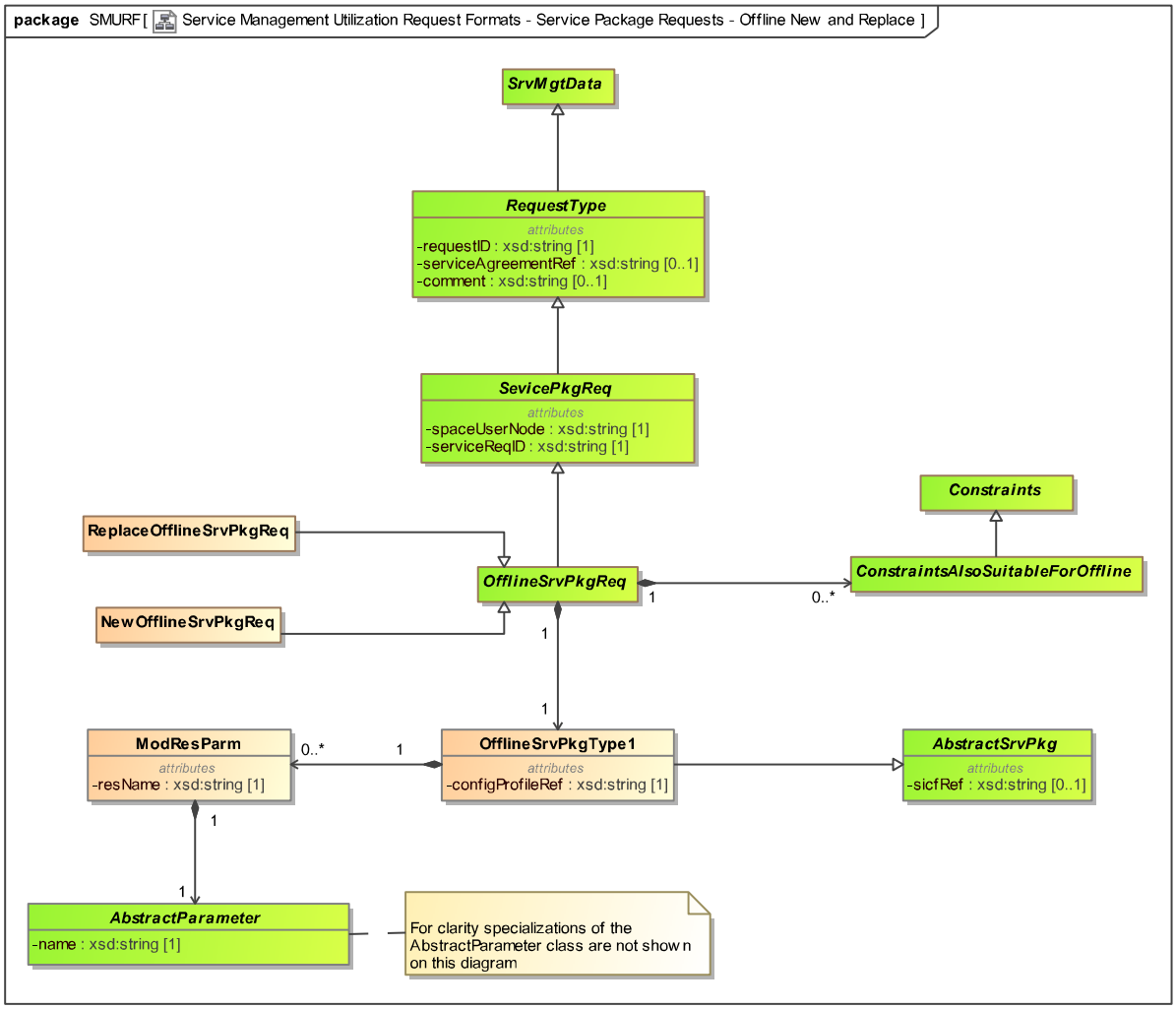

Figure 3-6: Service Management Utilization Request Formats Offline New and Replace Service Package Requests Class Diagram

3.7.2 USAGE OF SrvMgtUtilReqHeader `startTime` AND `endTime` PARAMETERS

In the context of the Offline New and Replace Service Package Request the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are not required and should not be specified. If specified they should be ignored. Any timing constraints required shall be defined in one or more of the classes derived from the abstract Constraints class (see 3.11).

3.7.3 CLASS OfflineSrvPkgReq (Abstract)

3.7.3.1 The OfflineSrvPkgReq class is an abstract class and shall be used to derive the classes required to submit a new online service package request or replace an existing offline service package request.

NOTE – The OfflineSrvPkgReq class is a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.7.3.2 Each instance of a class derived from the OfflineSrvPkgReq class shall contain 1, and only 1, instance of the OfflineSrvPkg class defined in 3.7.6.

3.7.3.3 Each instance of a class derived from the OfflineSrvPkgReq class may contain zero or more instances of classes derived from the ConstraintsAlsoSuitableForOffline class defined in 3.7.11.

3.7.3.4 The OfflineSrvPkgReq class shall contain the parameters in the abstract ServicePkgReq class described in 3.2.5.

3.7.4 CLASS NewOfflineSrvPkgReq

3.7.4.1 The NewOfflineSrvPkgReq class is optional and may be used to submit a new offline service package request.

NOTES

1 The NewOfflineSrvPkgReq class is a specialization of class OfflineSrvPkgReq described in 3.7.2, and the generic description of the parameters is given in that subsection.

2 The OfflineSrvPkgReq class is itself a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.7.4.2 The NewOfflineSrvPkgReq class shall contain the parameters in the abstract OfflineSrvPkg class described in 3.7.2.

3.7.4.3 In the context of the NewOfflineSrvPkgReq class, usage of the OfflineSrvPkgReq and ServicePkgReq parameters defined in table 3-3 shall be as specified in table 3-10.

Table 3-10: Class NewOfflineSrvPkgReq use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the NewOfflineSrvPkgReq class, this parameter must be an identifier that has not previously been used.	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 `spaceUserNode`, `serviceAgreementRef`, and `serviceReqID` constitutes a unique identifier for a service request.			

3.7.5 CLASS ReplaceOfflineSrvPkgReq

3.7.5.1 The ReplaceOfflineSrvPkgReq class is optional and may be used to replace an existing offline service package request.

NOTES

1 The ReplaceOfflineSrvPkgReq class is a specialization of class OfflineSrvPkgReq described in 3.7.2, and the generic description of the parameters is given in that subsection.

2 The OfflineSrvPkgReq class is itself a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.7.5.2 The ReplaceOfflineSrvPkgReq class shall contain the parameters in the abstract OfflineSrvPkg class defined in 3.7.2.

3.7.5.3 In the context of the ReplaceOfflineSrvPkgReq class, usage of the OfflineSrvPkgReq and ServicePkgReq parameters defined in table 3-3 shall be as specified in table 3-11.

Table 3-11: Class ReplaceOfflineSrvPkgReq use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the ReplaceOfflineSrvPkgReq class this parameter must refer to the identifier of the previously submitted request that is to be updated.	Xsd:string	n/a

3.7.6 CLASS AbstractSrvPkg (ABSTRACT)

The AbstractSrvPkg class is abstract and shall be used to derive the various service package classes required in service management.

NOTE – The AbstractSrvPkg class is fully described in reference [7].

3.7.7 CLASS OfflineSrvPkgType1

The OfflineSrvPkgType1 class shall be used to specify the contents of an offline service package.

NOTES

1 The OfflineSrvPkgType1 class is a specialization of class AbstractSrvPkg described in 3.7.6.

2 The OfflineSrvPkgType1 class is fully described in reference [7].

3.7.8 CLASS ModResParm

The ModResParm class may be used to modify the value of a parameter of a resource. The Resource for which the parameter value is to be modified shall be specified by the resource name.

NOTE – The ModResParm class is fully described in reference [7].

3.7.9 CLASS AbstractParameter (ABSTRACT)

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

NOTE – The AbstractParameter class is specified in reference [6].

3.7.10 CLASS Constraints (Abstract)

(See 3.10.2 for the definition of this class.)

3.7.11 CLASS ConstraintsAlsoSuitableForOffline (Abstract)

(See 3.10.3 for the definition of this class.)

3.8 DELETE SERVICE PACKAGE REQUEST, DELETE SERVICE PACKAGE, AND REPLACE SERVICE PACKAGE

3.8.1 OVERVIEW

The UML diagram in figure 3-7 shows the subset of classes that are applicable to the Delete Service Package Request, Delete Service Package, Replace Service Package and Set Active Scenario.

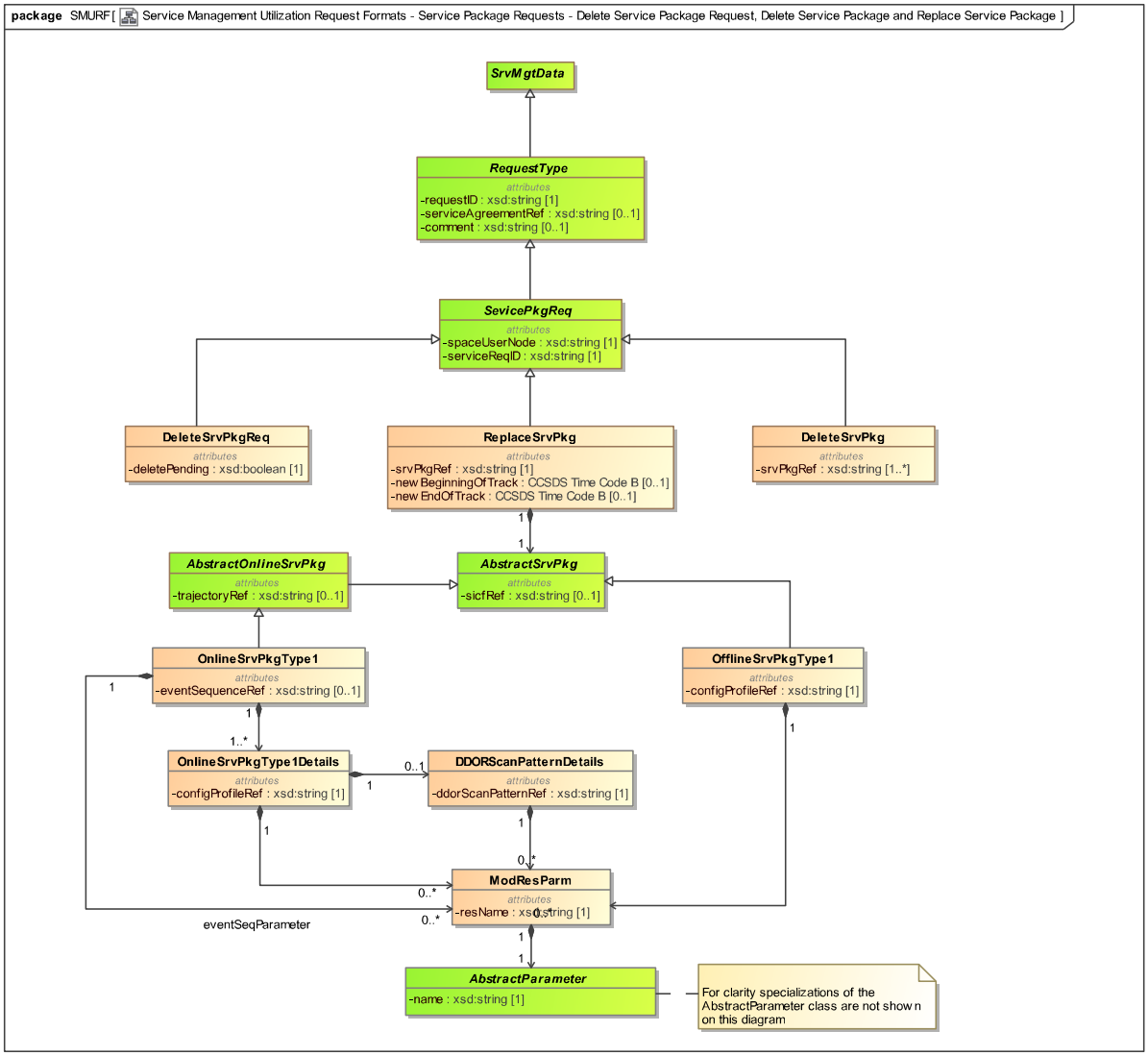

Figure 3-7: Service Management Utilization Request Formats Delete Service Package Request, Delete Service Package and Replace Service Package Class Diagram

3.8.2 USAGE OF SrvMgtUtilReqHeader `startTime` AND `endTime` PARAMETERS

In the context of the Delete Service Package Request, Delete Service Package and Replace Service Package Requests the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are not required and should not be specified. If specified they should be ignored.

3.8.3 CLASS DeleteSrvPkgReq

3.8.3.1 The DeleteSrvPkgReq class is optional and may be used to delete a previously submitted service request.

NOTE – The DeleteSrvPkgReq class is a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.8.3.2 The use of the DeleteSrvPkgReq shall cause the specified Service Package request to be deleted and may also specify if all service packages that have been generated as a result of the deleted service package request should be deleted.

3.8.3.3 In addition to the parameters of the abstract RequestType and ServicePkgReq classes defined in 3.2.4 and 3.2.5, the DeleteSrvPkg class shall contain the parameter specified in table 3-12.

Table 3-12: Class DeleteSrvPkgReq Parameters

Parameter	Description	Data Type	Data Units
`deletePending`	This is a Boolean flag that is used to specify whether pending Service Packages (i.e., Service packages that have been generated but not yet executed) resulting from the service package request specified for deletion should also be deleted. – If the flag is set TRUE all pending Service Packages that resulted from the service package request are deleted. – If the flag is set FALSE any pending service packages are not deleted.	xsd:boolean	n/a

3.8.3.4 In the context of the DeleteSrvPkgReq class, usage of the ServicePkgReq parameters defined in table 3-3 shall be as specified in table 3-13.

Table 3-13: Class DeleteSrvPkgReq use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the DeleteSrvPkgReq class, this parameter must refer to the identifier of the previously submitted request that is to be deleted.	xsd:string	n/a

3.8.4 CLASS DeleteSrvPkg

3.8.4.1 The DeleteSrvPkg class is optional and may be used to delete a service package that was generated by a previously submitted service request.

NOTES

1 The DeleteSrvPkg class is a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

2 DeleteSrvPkg will only result in the deletion of service packages that have not been, or are not currently being, executed.

3.8.4.2 In addition to the parameters of the abstract RequestType and ServicePkgReq classes defined in 3.2.4 and 3.2.5, the DeleteSrvPkg class shall contain the parameter specified in table 3-14.

Table 3-14: Class DeleteSrvPkg Parameters

Parameter	Description	Data Type	Data Units
`srvPkgRef`	This is used to specify the references(s) of the service package(s) that is (are) to be deleted.	xsd:string	n/a

3.8.4.3 In the context of the DeleteSrvPkg class, usage of the ServicePkgReq parameters defined in table 3-3 shall be as described in table 3-15.

Table 3-15: Class DeleteSrvPkg Use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the DeleteSrvPkg class, this parameter must refer to the identifier of the request that resulted in the generation of the service package(s) that is(are) now to be deleted.	xsd:string	n/a

3.8.5 CLASS ReplaceSrvPkg

3.8.5.1 The ReplaceSrvPkg class is optional and may be used to replace a service package that was generated by a previously submitted service request.

NOTE – The ReplaceSrvPkg class is a specialization of class ServicePkgReq described in 3.2.5, and the generic description of the parameters is given in that subsection.

3.8.5.2 Submitting a ReplaceSrvPkg request does not guarantee that the Provider CSSS will actually be able to carry out the replacement. Depending on the elements of the service package that are changed, there may be conflicts with existing scheduled service package(s) for other users which preclude the replace operation being carried out.

3.8.5.3 Each instance of the ReplaceSrvPkg class must contain exactly one instance of a class derived from the AbstractSrvPkg class (see 3.8.6) which fully describes the service package to be used as a replacement.

3.8.5.4 In addition to the parameters of the abstract RequestType and ServicePkgReq classes defined in 3.2.4 and 3.2.5, the ReplaceSrvPkg class shall contain the parameter specified in table 3-16.

Table 3-16: Class ReplaceSrvPkg Parameters

Parameter	Description	Data Type	Data Units
`srvPkgRef`	This is used to specify the reference of the service package that is to be replaced.	xsd:string	n/a

3.8.5.5 In the context of the ReplaceSrvPkg class, usage of the ServicePkgReq parameters defined in table 3-3 shall be as described in table 3-17.

Table 3-17: Class ReplaceSrvPkg use of Class ServicePkgReq Parameters

Parameter	Description	Data Type	Data Units
`serviceReqID`	In the context of the ReplaceSrvPkg class, this parameter must refer to the identifier of the request that resulted in the generation of the service package that is now to be replaced.	xsd:string	n/a

3.8.6 CLASS AbstractSrvPkg (ABSTRACT)

The AbstractSrvPkg class is abstract and shall be used to derive the various service package classes required in service management.

NOTE – The AbstractSrvPkg class is fully described in reference [7].

3.8.7 CLASS AbstractOnlineSrvPkg (ABSTRACT)

The AbstractOnlineSrvPkg class is abstract and shall be used to derive the various online service package classes required in service management.

1 The AbstractOnlineSrvPkg class is a specialization of class AbstractSrvPkg described in 3.8.6.

2 The AbstractSrvPkg class is fully described in reference [7].

3.8.8 CLASS OnlineSrvPkyType1

3.8.8.1 The OnlineSrvPkyType1 class shall be used to specify the contents of an online service package.

NOTES

1 The OnlineSrvPkyType1 class is a specialization of class AbstractOnlineSrvPkg described in 3.8.7.

2 The OnlineSrvPkyType1 class is fully described in reference [7].

3.8.9 CLASS OnlineSrvPkyType1Details

The OnlineSrvPkyType1Details class shall be used to specify details relevant to an online service package.

NOTE – The OnlineSrvPkyType1Details class is fully described in reference [7].

3.8.10 CLASS OfflineSrvPkgType1

The OfflineSrvPkgType1 class shall be used to specify the contents of an offline service package.

NOTES

1 The OfflineSrvPkgType1 class is a specialization of class AbstractSrvPkg described in 3.8.6.

2 The OfflineSrvPkg class is fully described in reference [7].

3.8.11 CLASS DDORScanPatternDetails

DDORScanPatternDetails class shall be used to specify a reference to an existing DDOR Scan Pattern.

NOTE – The DDORScanPatternDetails class is fully described in reference [7].

3.8.12 CLASS ModResParm

The ModResParm class may be used to modify the value of a parameter of a resource. The Resource for which the parameter value is to be modified shall be specified by the resource name.

NOTE – The ModResParm class is fully described in reference [7].

3.8.13 CLASS AbstractParameter (ABSTRACT)

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

NOTE – The AbstractParameter class is fully described in reference [6].

3.9 SUBMISSION REQUESTS

3.9.1 OVERVIEW

The UML diagram in figure 3-8 shows the subset of classes that are applicable to the Submission Requests.

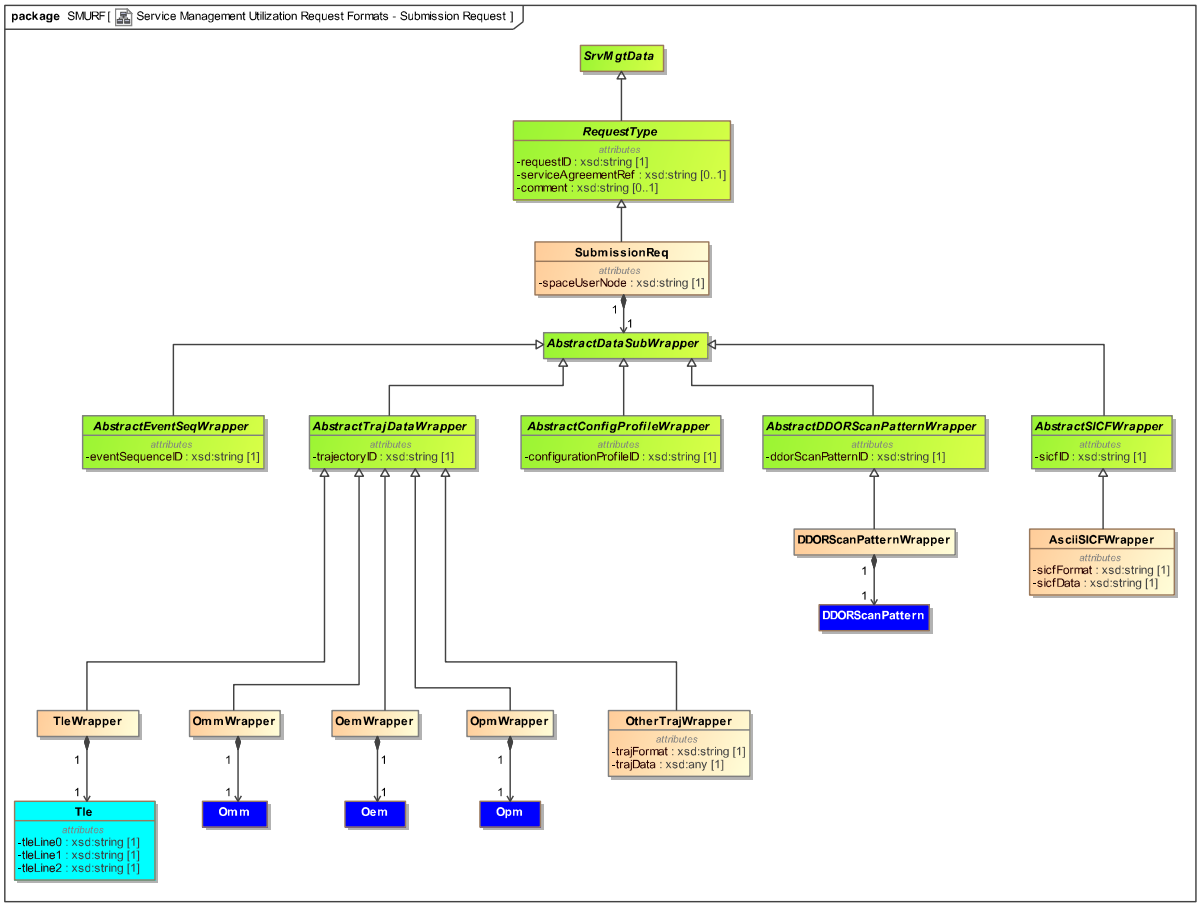

Figure 3-8: Service Management Utilization Request Formats Submission Requests Class Diagram

3.9.2 USAGE OF SrvMgtUtilReqHeader `startTime` AND `endTime` PARAMETERS.

In the context of the Submission Request the SrvMgtUtilReqHeader `startTime` and `endTime` parameters are not required and should not be specified. If specified they should be ignored.

3.9.3 CLASS SubmissionReq

3.9.3.1 The SubmissionReq is a class that may be used to submit new information entities, that is, trajectory predictions, configuration profiles, event sequences, DDOR scan patterns and SICF.

NOTE – The SubmissionReq class is a specialization and of class RequestType described in 3.2.4.

3.9.3.2 The SubmissionReq class shall contain 1, and only 1, instance of a class derived from the AbstractDataSubWrapper described in 3.9.4.

3.9.3.3 In addition to the parameters of the abstract RequestType class described in 3.2.4, the SubmissionReq class shall contain the parameter specified in table 3-18.

Table 3-18: Class SubmissionReq Parameters

Parameter	Description	Data Type	Data Units
`spaceUserNode`	The user of the requested services. These must be spacecraft names as specified in SANA.	xsd:string—Permitted values registered in SANA (see B2.3 for further information).	n/a

3.9.4 CLASS AbstractDataSubWrapper (ABSTRACT)

3.9.4.1 AbstractDataSubWrapper is an abstract class that shall be used to instantiate the wrapper classes required to encapsulate data entities that may be used in submission requests.

NOTES

1 The AbstractDataSubWrapper class is a specialization of class AbstractDataWrapper described in reference [7], and the generic description of the parameters is given in that document.

2 The AbstractDataSubWrapper class is fully described in reference [7].

3.9.5 CLASS AbstractEventSeqWrapper (ABSTRACT)

3.9.5.1 AbstractEventSeqWrapper is an abstract class that shall be used to instantiate the wrapper classes required for Event Sequence data entities.

NOTES

1 The AbstractEventSeqWrapper class is a specialization of class AbstractDataSubWrapper described in 3.9.4, and the generic description of the parameters is given in that subsection.

2 The AbstractEventSeqWrapper class is fully described in reference [7].

3.9.5.2 For a fully compliant CCSDS Submission Request for an Event Sequence,

a) the concrete Event Sequence Wrapper shall be derived from the AbstractEventSeqWrapper class as specified in reference [8];

b) this Event Sequence Wrapper shall contain one, and only one, instance of the Event Sequence class defined in reference [8].

3.9.6 CLASS AbstractTrajDataWrapper (ABSTRACT)

AbstractTrajDataWrapper is an abstract class that shall be used to instantiate the wrapper classes required for Trajectory data entities.

NOTES

1 The AbstractTrajDataWrapper class is a specialization of class AbstractDataSubWrapper described in 3.9.5, and the generic description of the parameters is given in that subsection.

2 The AbstractTrajDataWrapper class is fully described in reference [7].

3.9.7 CLASS OemWrapper

The OemWrapper is a class that shall be used to wrap an OEM data entity.

NOTES

1 The OemWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.9.6, and the generic description of the parameters is given in that subsection.

2 The OemWrapper class is fully described in reference [7].

3.9.8 CLASS OmmWrapper

The OmmWrapper is a class that shall be used to wrap an OMM data entity**.**

NOTES

1 The OmmWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.9.6, and the generic description of the parameters is given in that subsection.

2 The OmmWrapper class is fully described in reference [7].

3.9.9 CLASS OpmWrapper

The OpmWrapper is a class that shall be used to wrap an OPM data entity.

NOTES

1 The OpmWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.9.6, and the generic description of the parameters is given in that subsection.

2 The OpmWrapper class is fully described in reference [7].

3.9.10 CLASS TleWrapper

The TleWrapper is a class that shall be used to wrap a TLE data entity.

NOTES

1 The TleWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.9.6, and the generic description of the parameters is given in that subsection.

2 The TleWrapper class is fully described in reference [7].

3.9.11 CLASS Tle

Tle class shall be used to specify trajectory data in the TLE format.

NOTES

1 The TLE format is not a CCSDS Recommended Standard. It is therefore strongly recommended that, if at all feasible, that one of the OEM, OMM or OPM formats be used instead.

2 The Tle class is fully described in reference [7].

3.9.12 CLASS OtherTrajWrapper

The OtherTrajWrapper is a class that shall be used to wrap an arbitrary format trajectory data entity. This is done by using the type xsd:any to hold the required trajectory data; because of this an additional schema must be defined to specify the data that is stored in the xsd:any element.

NOTES

1 The OtherTrajWrapper class is a specialization of class AbstractTrajDataWrapper described in 3.9.6, and the generic description of the parameters is given in that subsection.

2 The OtherTrajWrapper class is fully described in reference [7].

3.9.13 CLASS AbstractConfigProfileWrapper (ABSTRACT)

AbstractConfigProfileWrapper is an abstract class that shall be used to instantiate the wrapper classes required for Configuration Profile data entities.

NOTES

1 The AbstractConfigProfileWrapper class is a specialization of class AbstractDataSubWrapper described in 3.9.4, and the generic description of the parameters is given in that subsection.

2 The AbstractConfigProfileWrapper class is fully described in reference [7].

3 For a fully compliant CCSDS Submission Request for a Configuration Profile it is required that the concrete Configuration Profile Wrapper derived from the AbstractConfigProfileWrapper class is as specified in reference [3]. It is further required that this Configuration Profile Wrapper contains one, and only one, instance of the Configuration Profile class defined in reference [3].

3.9.14 CLASS AbstractDDORScanPatternWrapper (ABSTRACT)

AbstractDDORScanPatternWrapper is an abstract class that shall be used to instantiate the wrapper classes required for DDOR Scan Pattern data entities.

NOTES

1 The AbstractDDORScanPatternWrapper class is a specialization of class AbstractDataSubWrapper described in 3.9.4, and the generic description of the parameters is given in that subsection.

2 The AbstractDDORScanPatternWrapper class is fully described in reference [7].

3.9.15 CLASS DDORScanPatternWrapper

The DDORScanPatternWrapper is a class that shall be used to wrap a DDOR Scan Pattern data entity.

NOTES

1 The DDORScanPatternWrapper class is a specialization of class AbstractDDORScanPatternWrapper described in 3.9.14, and the generic description of the parameters is given in that subsection.

2 The DDORScanPatternWrapper class is fully described in reference [7].

3.9.16 CLASS AbstractSICFWrapper (ABSTRACT)

AbstractSICFWrapper is an abstract class that shall be used to instantiate the wrapper classes required for SICF data entities.

NOTES

1 As there is no standard specification for SICF it is necessary to derive an Agency (or mission) specific SICF Wrapper class from this class.

2 The AbstractSICFWrapper class is a specialization of class AbstractDataSubWrapper described in 3.9.4, and the generic description of the parameters is given in that subsection.

3 The AbstractSICFWrapper class is fully described in reference [7].

3.9.17 CLASS AsciiSICFWrapper

Whilst, as noted above, there is no standard format for SICFs, the vast majority of these are at least expressed in ASCII. The AsciiSICFWrapper is a class that may used to wrap an arbitrary format SICF that is in ASCII format.

NOTES

1 The AsciiSICFWrapper class is a specialization of class AbstractSICFWrapper described in 3.9.16, and the generic description of the parameters is given in that subsection.

2 The AsciiSICFWrapper class is fully described in reference [7].

3.10 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS BASIC CONSTRAINTS

Basic constraints are intended to be used to request a single tracking pass of a spacecraft. This single tracking pass may involve the use of one or more ground stations and/or relay satellites apertures to provide the required coverage. It is therefore possible to specify handover constraints between apertures.

3.10.1 OVERVIEW

Figure 3-9 shows the UML Class diagram for the Service Management Utilization Request Formats Basic Constraints.

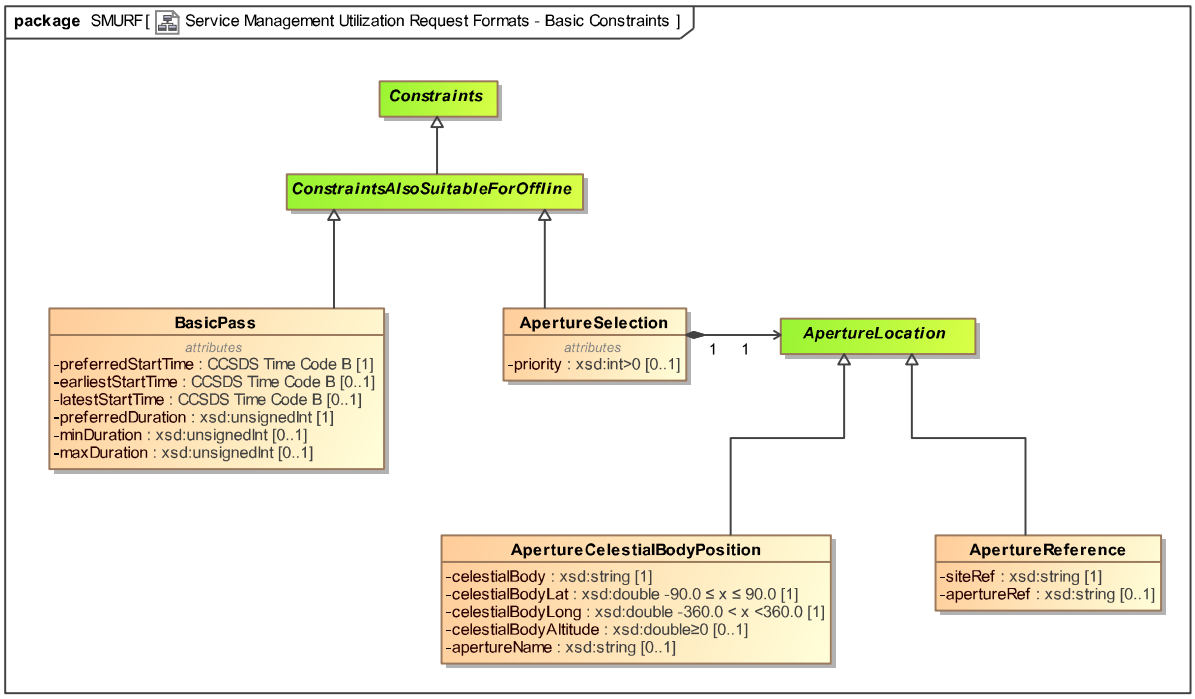

Figure 3-9: Service Management Utilization Request Formats Basic Constraints Class Diagram

The attributes of each class are described further in the following subsections and tables.

3.10.2 CLASS Constraints (Abstract)

3.10.2.1 The Constraints class is abstract and shall be used to instantiate the various constraint classes that apply to the Service Management Utilization Request Formats.

3.10.2.2 The abstract Constraint class does not contain any parameters.

3.10.3 CLASS ConstraintsAlsoSuitableForOffline (Abstract)

3.10.3.1 The ConstraintsAlsoSuitableForOffline class is abstract and shall be used to instantiate the various constraint classes that can be used for offline as well as other Service Management Utilization Request Formats.

NOTE – The ConstraintsAlsoSuitableForOffline class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.10.3.2 The abstract ConstraintsAlsoSuitableForOffline class does not contain any parameters.

3.10.4 CLASS BasicPass

3.10.4.1 The BasicPass class is optional and 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.

NOTE – The BasicPass class is a specialization of class ConstraintsAlsoSuitableForOffline described in 3.10.3, and the generic description of the parameters can be found there.

3.10.4.2 The parameters of the BasicPass class shall be as specified in table 3-19.

Table 3-19: Class BasicPass Parameters

Parameter	Description	Data Type	Data Units
`PreferredStartTime`	Used to specify (in UTC) the preferred time at which a pass starts. NOTES 1 If an earliestStartTime is specified, then PreferredStartTime ≥ earliestStartTime. 2 If a latestStartTime is specified, then PreferredStartTime ≤ latestStartTime.	CCSDS ASCII Time Code B (reference [1])	UTC
`earliestStartTime`	Optional parameter. Used to specify (in UTC) the earliest time at which a pass can start.	CCSDS ASCII Time Code B (reference [1])	UTC

Parameter	Description	Data Type	Data Units
latestStartTime	Optional parameter. Used to specify (in UTC) the latest time at which a pass can start. NOTE – If both earliestStartTime and latestStartTime are specified, then latestStartTime ≥ earliestStartTime.	CCSDS ASCII Time Code B (reference [1])	UTC
preferredDuration	This is used to specify the preferred duration of a pass. NOTES 1 If a minimumDuration is specified then preferredDuration ≥ minimumDuration. 2 If a maximumDuration is specified, then preferredDuration ≤ maximumDuration.	xsd:unsignedInt	Secs
minimumDuration	Optional parameter. This is used to specify the minimum duration required for a pass.	xsd:unsignedInt Default value if omitted is 0	Secs
maximumDuration	Optional parameter. This is used to specify the maximum duration required for a pass. NOTE – If both minimumDuration and maximumDuration are specified, then maximumDuration ≥ minimumDuration.	xsd:unsignedInt Default value if omitted is ∞	Secs

3.10.5 CLASS ApertureSelection

3.10.5.1 The ApertureSelection class is optional and may be used to specify an aperture with which it is desired that the service is provided.

NOTE – The ApertureSelection class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.10.5.2 Each instance of the ApertureSelection class, in this context, must contain one, and only one, instance of the ApertureReference class that is derived from the abstract ApertureLocation class.

3.10.5.3 The parameters of the ApertureSelection class shall be as specified in table 3-20.

Table 3-20: Class ApertureSelection Parameters

Parameter	Description	Data Type	Data Units
priority	Optional Parameter. This can be used to specify a preference for which aperture is used if more than one ApertureSelection class is instantiated. The lower the value specified here the higher the priority, with 1 being the highest priority.	xsd:integer >0 Default value if omitted is 1	n/a

3.10.6 CLASS ApertureLocation (ABSTRACT)

3.10.6.1 ApertureLocation is an abstract class that shall be used to instantiate the various classes of aperture location required by service management and is fully described in reference [7].

3.10.7 CLASS ApertureReference

3.10.7.1 The ApertureReference class allows the specification of the location of an aperture by the site name where it is located and the name of the aperture and is fully described in reference [7].

NOTE – The ApertureReference class is a specialization of class ApertureLocation mentioned in 3.10.6 and fully defined in reference [7].

3.10.8 CLASS ApertureCelestialBodyPosition

3.10.8.1 The ApertureCelestialBodyPosition class allows the specification of the location of an aperture by the site name where it is located and the name of the aperture and is fully described in reference [7].

NOTE – The ApertureCelestialBodyPosition class is a specialization of class ApertureLocation mentioned in 3.10.6 and fully defined in reference [7].

3.11 SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS ENHANCED CONSTRAINTS

The enhanced constraints are intended to provide additional flexibility when requesting tracking passes of a spacecraft and allow the possibility for 'standing orders' to be defined that request of a number of passes within a specified time range subject to the same constraints. It should be noted that the use of basic and enhanced constraints is not mutually exclusive as the enhanced constraints effectively extends the functionality provided by the basic constraints.

3.11.1 OVERVIEW

Figure 3-10 shows the UML Class diagram for the Service Management Utilization Request Formats Enhanced Constraints.

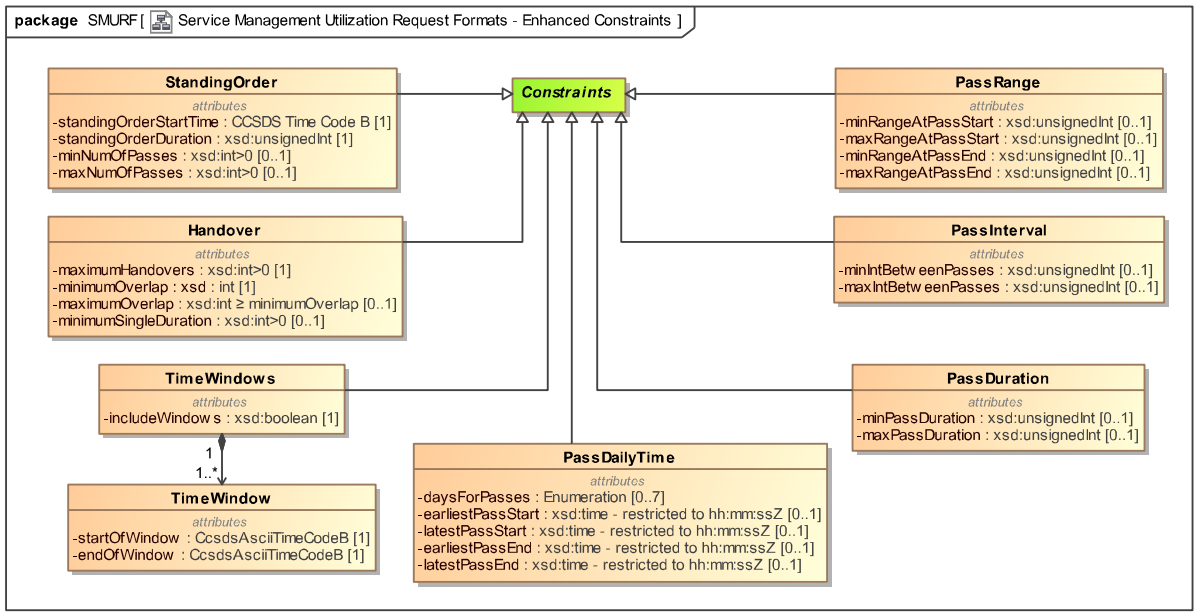

Figure 3-10: Service Management Utilization Request Formats Enhanced Constraints Class Diagram

The attributes of each class are described further in the following subsections and tables.

3.11.2 CLASS Constraints (Abstract)

(See 3.10.2 for the definition of this class.)

3.11.3 CLASS StandingOrder

3.11.3.1 The StandingOrder class is optional and may be used to submit 'repeat' requests.

NOTES

1 The StandingOrder class allows the specification of the number of passes that are required within a specified time interval.

2 The StandingOrder class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.3.2 The StandingOrder class shall include parameters as specified in table 3-21.

Table 3-21: Class StandingOrder Parameters

Parameter	Description	Data Type	Data Units
`standingOrderStartTime`	Used to specify (in UTC) the start time of the standing order. That is the earliest time at which a service package generated as a result of the standing order may start.	CCSDS ASCII Time Code B (reference [1])	UTC
`standingOrderDuration`	Used to specify the duration of the standing order.	xsd:unsignedInt	Secs
`minNumOfPasses`	Optional parameter. This may be used to specify the minimum number of passes required in the specified time interval.	xsd:integer > 0	n/a
`maxNumOfPasses`	Optional parameter. This may be used to specify the maximum number of passes required in the specified time interval. NOTE – If both `minNumOfPasses` and `maxNumOfPasses` are specified, then maxNumOfPasses ≥ `minNumOfPasses`	xsd:integer ≥ 0	n/a

3.11.4 CLASS Handover

3.11.4.1 The Handover class is optional and may be used to indicate if a station handover is permitted; if not specified for a request it shall be assumed that a station handover is not permitted.

NOTE – The Handover class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.4.2 The Handover class shall contain the details of the constraints applying to station handovers and shall include parameters as specified in table 3-22.

Table 3-22: Class Handover Parameters

Parameter	Description	Data Type	Data Units
`maximumHandovers`	The maximum number of station handovers that are permitted during the requested tracking time.	xsd:integer >0	n/a
`mimimumOverlap`	Used to specify the minimum overlap between stations during a handover. NOTE – If this is negative, then there will be a gap in the coverage provided.	xsd:integer	Secs
`maximumOverlap`	Optional Parameter. This can be used to specify the maximum overlap between stations during a handover.	xsd:integer ≥ `mimimumOverlap`	Secs
`minimunSingleDuration`	Optional Parameter. This can be used to specify the acceptable minimum duration of contact on any one of the apertures used during the pass.	Xsd:integer>0	Secs

3.11.5 CLASS TimeWindows

3.11.5.1 The TimeWindows class is optional and may be used to submit a set of time windows.

3.11.5.2 Each instance of the TimeWindows class must contain one or more instances of the TimeWindow class described in 3.11.6.

NOTE – The TimeWindows class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.5.3 The use of the TimeWindows class allows the specification of a set of one or more time windows that must either all be included or all excluded in the resulting schedule.

3.11.5.4 Use of the parameters in the TimeWindows class shall be as specified in table 3-23.

Table 3-23: Class TimeWindows Parameters

Parameter	Description	Data Type	Data Units
`includeWindow`	If `true`, indicates that the time range specified by the window is considered to be included. If `false`, indicates that the time range specified by the window is considered to be excluded.	xsd:Boolean	n/a

3.11.6 CLASS TimeWindow

3.11.6.1 The TimeWindow class shall be used to specify time windows that can be included in a set of time windows represented by the TimeWindows class described in 3.11.4.

3.11.6.2 Use of the parameters in the TimeWindow class shall be as specifies in table 3-24.

Table 3-24: Class TimeWindow Parameters

Parameter	Description	Data Type	Data Units
`startOfWindow`	Used to specify (in UTC) the start time of the window.	CCSDS ASCII Time Code B (reference [1])	UTC
`endOfWindow`	Used to specify (in UTC) the end time of the window.	CCSDS ASCII Time Code B (reference [1])	UTC

3.11.7 CLASS PassRange

3.11.7.1 The PassRange class is optional and may be used to permit the specification of the minimum and/or maximum range between the spacecraft and aperture for the start and/or end of a pass.

NOTE – The PassRange class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.7.2 The parameters of the PassRange class shall be as specified in table 3-25.

Table 3-25: Class PassRange Parameters

Parameter	Description	Data Type	Data Units
`minRangeAtPassStart`	Optional parameter. Used to specify the minimum range required at the pass start.	xsd:unsignedInt	Kilometres
`maxRangeAtPassStart`	Optional parameter. Used to specify the maximum range required at the pass start. NOTE – If both `minRangeAtPassStart` and `maxRangeAtPassStart` are specified, then `maxRangeAtPassStart` ≥ `minRangeAtPassStart`.	xsd:unsignedInt	Kilometres
`minRangeAtPassEnd`	Optional parameter. Used to specify the minimum range required at the pass end.	xsd:unsignedInt	Kilometres
`maxRangeAtPassEnd`	Optional parameter. Used to specify the maximum range required at the pass start. NOTE – If both `minRangeAtPassEnd` and `maxRangeAtPassEnd` are specified, then `maxRangeAtPassEnd` ≥ `minRangeAtPassEnd`.	xsd:unsignedInt	Kilometres

3.11.8 CLASS PassInterval

3.11.8.1 The PassInterval class is optional and may be used to permit the specification of the minimum and/or maximum intervals between passes.

NOTE – The PassInterval class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.8.2 The parameters of the PassInterval class shall be as specified in table 3-26.

Table 3-26: Class PassInterval Parameters

Parameter	Description	Data Type	Data Units
`minIntBetweenPasses`	Optional parameter. This is used to specify the minimum interval required between passes.	xsd:unsignedInt Default value if omitted is 0	Secs
`maxIntBetweenPasses`	Optional parameter. This is used to specify the maximum interval required between passes. NOTE – If both minimumIntervalBetweenPasses and maximumIntervalBetweenPasses are specified, then maximumIntervalBetweenPasses ≥ minimumIntervalBetweenPasses.	xsd:unsignedInt Default value if omitted is ∞	Secs

3.11.9 CLASS PassDuration

3.11.9.1 The PassDuration class is optional and may be used to permit the specification of the minimum and/or maximum duration of passes.

NOTE – The PassDuration class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.9.2 The parameters of the PassDuration class shall be as specified in table 3-27.

Table 3-27: Class PassDuration Parameters

Parameter	Description	Data Type	Data Units
`minPassDuration`	Optional parameter. This is used to specify the minimum duration of a pass.	xsd:unsignedInt Default value if omitted is 0	Secs
`maxPassDuration`	Optional parameter. This is used to specify the maximum duration of a pass. NOTE – If both minPassDuration and maxPassDuration are specified, then minPassDuration ≥ maxPassDuration.	xsd:unsignedInt Default value if omitted is ∞	Secs

3.11.10 CLASS PassDailyTime

3.11.10.1 The PassDailyTime class is optional and may be used to permit the specification of the earliest and/or latest absolute (UTC) times for passes. These constraints may be used to limit passes to being only during normal working hours, etc.

NOTE – The PassDailyTime class is a specialization of class Constraints described in 3.10.2, and the generic description of the parameters can be found there.

3.11.10.2 The parameters of the PassDailyTime class shall be as specified in table 3-28.

Table 3-28: Class PassDailyTime Parameters

Parameter	Description	Data Type	Data Units
`daysForPasses`	Optional parameter. Used to specify the day(s) on which a pass can be accepted.	Enumeration – MONDAY – TUESDAY – WEDNESDAY – THURSDAY – FRIDAY – SATURDAY – SUNDAY	n/a
`earliestPassStart`	Optional parameter. Used to specify (in UTC) the earliest time during the day at which a pass can start.	xsd:time restricted to hh:mm:ssZ	n/a
`latestPassStart`	Optional parameter. Used to specify (in UTC) the latest time during the day at which a pass can start. NOTE – If both `earliestPassStart` and `latestPassStart` are specified, then `latestPassStart` ≥ `earliestPassStart`.	xsd:time restricted to hh:mm:ssZ	n/a
`earliestPassEnd`	Optional parameter. Used to specify (in UTC) the earliest time during the day at which a pass can end.	xsd:time restricted to hh:mm:ssZ	n/a
`latestPassEnd`	Optional parameter. Used to specify (in UTC) the latest time during the day at which a pass can end. NOTE – If both `earliestPassEnd` and `latestPassEnd` are specified, then `latestPassEnd` ≥ `earliestPassEnd`.	xsd:time restricted to hh:mm:ssZ	n/a

ANNEX A

IMPLEMENTATION CONFORMANCE STATEMENT (ICS) PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of the *Service Management Utilization Request Formats Specification* (CCSDS 902.2-B-1). The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements referenced in the RL.

The RL in this annex is blank. An implementation's completed RL is called the ICS. The ICS states which capabilities and options have been implemented. The following can use the ICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- a supplier or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard ICS proforma;
- a user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (it should be noted that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible ICSes);
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 ABBREVIATIONS AND CONVENTIONS

A1.2.1 General

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

A1.2.2 Item Column

The item column contains sequential numbers for items in the table.

A1.2.3 Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means 'Is this feature supported by the implementation?'

NOTE – The features itemized in the RL are elements of the Service Management Utilization Request Formats. Therefore support for a mandatory feature indicates that a generated file will include that feature, and support for an optional feature indicates that generated files can include that feature.

A1.2.4 Class Column/Parameters

The Class/Parameters column contains, where applicable, the Service Management Utilization Request Formats class associated with the feature.

A1.2.5 Reference Column

The reference column indicates the relevant subsection or table in the *Service Management Utilization Request Formats Specification* (CCSDS 902.2-B-1) (this document).

A1.2.6 Status Column

The status column uses the following notations:

M mandatory.

O optional.

C<n> conditional as defined in corresponding expression below the table.

N/A not applicable.

It should be noted that a parameter may be marked as M, while the class that contains it is marked O. This should be interpreted to mean that while the class is optional if it is present, the parameter must be present.

A1.2.7 Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

Y Yes, supported by the implementation.

N No, not supported by the implementation.

N/A Not applicable.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference *Xi*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 ICS PROFORMA FOR SERVICE MANAGEMENT UTILIZATION REQUEST FORMAT

A2.1 GENERAL INFORMATION

A2.1.1 Identification of ICS

Date of Statement (DD/MM/YYYY)	
ICS serial number	
System Conformance statement cross-reference	

A2.1.2 Identification of Implementation Under Test (IUT)

Implementation name	
Implementation version	
Special Configuration	
Other Information	

A2.1.3 Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other Information necessary for full identification, for example, names(s) and version(s) for machines and/or operating systems.	

A2.1.4 Document Version

CCSDS 920.9-R-1	
Have any exceptions been required? Note – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming.	Yes _____ No _____

A2.1.5 Requirements List

A2.1.5.1 Service Management Utilization Request Format Classes

A2.1.5.1.1 Class SrvMgtUtilReq

A2.1.5.1.1.1.1 General

Item	Description	Ref.	Status	Support
1.	`SrvMgtUtilReq`	3.2.2	M	

A2.1.5.1.1.2 Class SrvMgtUtilReq Parameters

N/A.

A2.1.5.1.2 Class SrvMgtUtilReqHeader

A2.1.5.1.2.1 General

Item	Description	Ref.	Status	Support
2.	`SrvMgtUtilReqHeader`	3.2.3	M	

A2.1.5.1.2.2 Class SrvMgtUtilReqHeader Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
2.1.	`originatingOrganization`	reference [7], table 3-1	M	
2.2.	`generationTime`	reference [7], table 3-1	M	
2.3.	`version`	reference [7], table 3-1	M	
2.4.	`startTime`	reference [7], table 3-1	O	
2.5.	`endTime`	reference [7], table 3-1	O	
2.6.	`purpose`	reference [7], table 3-1	O	
2.7.	`description`	reference [7], table 3-1	O	
2.8.	`utilizationReqStatus`	table 3-1	M	

A2.1.5.1.3 Class RequestType (Abstract)

A2.1.5.1.3.1 General

Item	Description	Ref.	Status	Support
3.	RequestType	3.2.4	M	

A2.1.5.1.3.2 Class RequestType Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
3.1.	requestID	table 3-2	M	
3.2.	serviceAgreementRef	table 3-2	O	
3.3.	comment	table 3-2	O	

A2.1.5.1.4 Class ServicePkgReq (Abstract)

A2.1.5.1.4.1 General

Item	Description	Ref.	Status	Support
4.	ServicePkgReq	3.2.5	M	

A2.1.5.1.4.2 Class ServicePkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
4.1.	requestID	table 3-2	M	
4.2.	serviceAgreementRef	table 3-2	O	
4.3.	comment	table 3-2	O	
4.4.	spaceUserNode	table 3-3	M	
4.5.	serviceReqID	table 3-3	M	

A2.1.5.2 Planning Information Request Classes

A2.1.5.2.1 Class PlanningInfoReq

A2.1.5.2.1.1 General

Item	Description	Ref.	Status	Support
5.	`PlanningInfoReq`	3.3.2	O	

A2.1.5.2.1.2 Class PlanningInfoReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
5.1.	`requestID`	table 3-2	M	
5.2.	`serviceAgreementRef`	table 3-2	O	
5.3.	`comment`	table 3-2	O	
5.4.	`spaceUserNode`	table 3-4	M	
5.5.	`trajectoryRef`	table 3-4	M	
5.6.	`configurationProfileRef`	table 3-4	O	

A2.1.5.2.2 Class PlanningInfoTypeList

A2.1.5.2.2.1 General

Item	Description	Ref.	Status	Support
6.	`PlanningInfoTypeList`	3.2.2	M	

A2.1.5.2.2.2 Class PlanningInfoTypeList Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
6.1.	`planningInfoType`	reference [7], table 3-7	M	

A2.1.5.2.3 Class Constraints (Abstract)

A2.1.5.2.3.1 General

Item	Description	Ref.	Status	Support
7.	`Constraints`	3.3.5	O	

A2.1.5.2.3.2 Class Constraints Parameters

N/A.

A2.1.5.3 Report Request Classes

A2.1.5.3.1 Class ReportReq (Abstract)

A2.1.5.3.1.1 General

Item	Description	Ref.	Status	Support
8.	ReportReq	3.4.2	O	

A2.1.5.3.1.2 Class ReportReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
8.1.	requestID	table 3-2	M	
8.2.	serviceAgreementRef	table 3-2	O	
8.3.	comment	table 3-2	O	
8.4.	spaceUserNode	table 3-5	M	
8.5.	startTime	table 3-5	M	
8.6.	endTime	table 3-5	M	

A2.1.5.3.2 Class SimpleScheduleRepReq

A2.1.5.3.2.1 General

Item	Description	Ref.	Status	Support
9.	SimpleScheduleRepReq	3.4.4	O	

A2.1.5.3.2.2 Class SimpleScheduleRepReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
9.1.	requestID	table 3-2	M	
9.2.	serviceAgreementRef	table 3-2	O	
9.3.	Comment	table 3-2	O	
9.4.	spaceUserNode	table 3-5	M	
9.5.	startTime	table 3-5	M	
9.6.	endTime	table 3-5	M	
9.7.	reportType	table 3-6	M	

A2.1.5.3.3 Class ApertureLocation (Abstract)

A2.1.5.3.3.1 General

Item	Description	Ref.	Status	Support
10.	ApertureLocation	3.4.5	O	

A2.1.5.3.3.2 Class ApertureLocation Parameters

N/A.

A2.1.5.3.4 Class ApertureReference

A2.1.5.3.4.1 General

Item	Description	Ref.	Status	Support
11.	ApertureReference	3.4.6	C1	

C1 - Mandatory if ApertureCelestialBodyPosition (A2.1.5.3.5) class is not included; shall not be present if ApertureCelelstiaBodyPosition is present.

A2.1.5.3.4.2 Class ApertureReference Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
11.1.	apertureRef	reference [7], table 3-5	M	
11.2.	siteRef	reference [7], table 3-5	M	

A2.1.5.3.5 Class ApertureCelestialBodyPosition

A2.1.5.3.5.1 General

Item	Description	Ref.	Status	Support
12.	ApertureCelestialBodyPosition	3.4.7	C2	

C2 - Mandatory if ApertureReference (A2.1.5.3.4) class is not present; shall not be present if ApertureReference class is present.

A2.1.5.3.5.2 Class ApertureCelestialBodyPosition Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
12.1.	`celestialBody`	reference [7], table 3-6	M	
12.2.	`celestialBodyLat`	reference [7], table 3-6	M	
12.3.	`celestialBodylong`	reference [7], table 3-6	M	
12.4.	`celestialBodyAltitude`	reference [7], table 3-6	O	
12.5.	`apertureName`	reference [7], table 3-6	O	

A2.1.5.4 Information Request Classes

A2.1.5.4.1 Class InfoReq

A2.1.5.4.1.1 General

Item	Description	Ref.	Status	Support
13.	InfoReq	3.5.2	O	

A2.1.5.4.1.2 Class InfoReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
13.1.	requestID	table 3-2	M	
13.2.	serviceAgreementRef	table 3-2	O	
13.3.	comment	table 3-2	O	
13.4.	spaceUserNode	table 3-7	M	
13.5.	configurationProfileRef	table 3-7	O	
13.6.	eventSequenceRef	table 3-7	O	
13.7.	serviceAgreementRef	table 3-7	O	
13.8.	servicePackageRef	table 3-7	O	
13.9.	trajectoryRef	table 3-7	O	
13.10.	ddorScanPatternref	table 3-7	O	
13.11.	sicfRef	table 3-7	O	

A2.1.5.5 Online New and Replace Service Package Requests Classes

A2.1.5.5.1 Class OnlineSrvPkgReq (Abstract)

A2.1.5.5.1.1 General

Item	Description	Ref.	Status	Support
14.	OnlineSrvPkgReq	3.6.2	O	

A2.1.5.5.1.2 Class OnlineSrvPkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
14.1.	requestID	table 3-2	M	
14.2.	serviceAgreementRef	table 3-2	O	
14.3.	comment	table 3-2	O	
14.4.	spaceUserNode	table 3-3	M	
14.5.	serviceReqID	table 3-3	M	

A2.1.5.5.2 Class NewOnlineSrvPkgReq

A2.1.5.5.2.1 General

Item	Description	Ref.	Status	Support
15.	NewOnlineSrvPkgReq	3.6.4	C3	

C3 - Mandatory if ReplaceOnlineSrvPkgReq (A2.1.5.5.3) class is not present; shall not be present if ReplaceOnlineSrvPkgReq class is present.

A2.1.5.5.2.2 Class NewOnlineSrvPkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
15.1.	requestID	table 3-2	M	
15.2.	serviceAgreementRef	table 3-2	O	
15.3.	comment	table 3-2	O	
15.4.	spaceUserNode	table 3-3	M	
15.5.	serviceReqID	table 3-8	M	

A2.1.5.5.3 Class ReplaceOnlineSrvPkgReq

A2.1.5.5.3.1 General

Item	Description	Ref.	Status	Support
16.	ReplaceOnlineSrvPkgReq	3.6.5	C4	

C4 - Mandatory if NewOnlineSrvPkgReq (A2.1.5.5.2) class is not present; shall not be present if NewOnlineSrvPkgReq class is present.

A2.1.5.5.3.2 Class ReplaceOnlineSrvPkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
16.1.	requestID	table 3-2	M	
16.2.	serviceAgreementRef	table 3-2	O	
16.3.	comment	table 3-2	O	
16.4.	spaceUserNode	table 3-3	M	
16.5.	serviceReqID	table 3-9	M	

A2.1.5.5.4 Class AbstractSrvPkg (ABSTRACT)

A2.1.5.5.4.1 General

Item	Description	Ref.	Status	Support
17.	AbstractSrvPkg	3.6.6	M	

A2.1.5.5.4.2 Class AbstractSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
17.1.	sicfRef	reference [7], table 3-20	O	

A2.1.5.5.5 Class AbstractOnlineSrvPkg (ABSTRACT)

A2.1.5.5.5.1 General

Item	Description	Ref.	Status	Support
18.	AbstractOnlineSrvPkg	3.6.7	M	

A2.1.5.5.5.2 Class AbstractOnlineSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
18.1.	trajectoryRef	reference [7], table 3-21	O	

A2.1.5.5.6 Class OnlineSrvPkyType1

A2.1.5.5.6.1 General

Item	Description	Ref.	Status	Support
19.	OnlineSrvPkyType1	3.6.8	M	

A2.1.5.5.6.2 Class OnlineSrvPkyType1 Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
19.1.	eventSequenceRef	reference [7], table 3-22	M	

A2.1.5.5.7 Class OnlineSrvPkyType1Details

A2.1.5.5.7.1 General

Item	Description	Ref.	Status	Support
20.	OnlineSrvPkyType1Details	3.6.9	M	

A2.1.5.5.7.2 Class OnlineSrvPkyType1Details Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
20.1.	configProfileRef	reference [7], table 3-23	M	

A2.1.5.5.8 Class DDORScanPatternDetails

A2.1.5.5.8.1 General

Item	Description	Ref.	Status	Support
21.	DDORScanPatternDetails	3.6.10	O	

A2.1.5.5.8.2 Class DDORScanPatternDetails Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
21.1.	DDORScanPatternRef	reference [7], table 3-24	M	

A2.1.5.5.9 Class ModResParm

A2.1.5.5.9.1 General

Item	Description	Ref.	Status	Support
22.	ModResParm	3.6.11	O	

A2.1.5.5.9.2 Class ModResParm Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
22.1.	resName	reference [7], table 3-8	O	

A2.1.5.5.10 Class AbstractParameter (ABSTRACT)

A2.1.5.5.10.1 General

Item	Description	Ref.	Status	Support
23.	AbstractParameter	3.6.12	C5	

C5—Mandatory if ModResParm (A2.1.5.5.9) is present.

A2.1.5.5.10.2 Class AbstractParameter Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
23.1.	name	reference [6], table 3-2	M	

A2.1.5.5.11 Class Constraints (Abstract)

A2.1.5.5.11.1 General

Item	Description	Ref.	Status	Support
24.	Constraints	3.6.13	O	

A2.1.5.5.11.2 Class Constraints Parameters

N/A.

A2.1.5.6 Offline New and Replace Service Package Requests Classes

A2.1.5.6.1 Class OfflineSrvPkgType1Req (Abstract)

A2.1.5.6.1.1 General

Item	Description	Ref.	Status	Support
25.	OfflineSrvPkgType1Req	3.7.2	M	

A2.1.5.6.1.2 Class OfflineSrvPkgType1Req Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
25.1.	requestID	table 3-2	M	
25.2.	serviceAgreementRef	table 3-2	O	
25.3.	comment	table 3-2	O	
25.4.	spaceUserNode	table 3-3	M	
25.5.	serviceReqID	table 3-3	M	

A2.1.5.6.2 Class NewOfflineSrvPkgReq

A2.1.5.6.2.1 General

Item	Description	Ref.	Status	Support
26.	NewOfflineSrvPkgReq	3.7.4	C6	

C6 - Mandatory if ReplaceOfflineSrvPkgReq (A2.1.5.6.3) class is not present; shall not be present if ReplaceOfflineSrvPkgReq class is present.

A2.1.5.6.2.2 Class NewOfflineSrvPkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
26.1.	requestID	table 3-2	M	
26.2.	serviceAgreementRef	table 3-2	O	
26.3.	comment	table 3-2	O	
26.4.	spaceUserNode	table 3-3	M	
26.5.	serviceReqID	table 3-10	M	

A2.1.5.6.3 Class ReplaceOfflineSrvPkgReq

A2.1.5.6.3.1 General

Item	Description	Ref.	Status	Support
27.	`ReplaceOfflineSrvPkgReq`	3.7.5	C7	

C7 - Mandatory if NewOfflineSrvPkgReq (A2.1.5.6.2) class is not present; shall not be present if NewOfflineSrvPkgReq class is present.

A2.1.5.6.3.2 Class ReplaceOfflineSrvPkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
27.1.	requestID	table 3-2	M	
27.2.	serviceAgreementRef	table 3-2	O	
27.3.	comment	table 3-2	O	
27.4.	spaceUserNode	table 3-3	M	
27.5.	serviceReqID	table 3-11	M	

A2.1.5.6.4 Class AbstractSrvPkg

A2.1.5.6.4.1 General

Item	Description	Ref.	Status	Support
28.	AbstractSrvPkg	3.7.6	M	

A2.1.5.6.4.2 Class AbstractSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
28.1.	sicfRef	reference [7] Table 3-20	O	

A2.1.5.6.5 Class OfflineSrvPkgType1

A2.1.5.6.5.1 General

Item	Description	Ref.	Status	Support
29.	OfflineSrvPkgType1	3.7.7	M	

A2.1.5.6.5.2 Class OfflineSrvPkgType1 Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
29.1.	configProfileRef	reference [7] Table 3-25	M	

A2.1.5.6.6 Class ModResParm

A2.1.5.6.6.1 General

Item	Description	Ref.	Status	Support
30.	ModResParm	3.7.8	O	

A2.1.5.6.6.2 Class ModResParm Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
30.1.	resName	reference [7] Table 3-8	M	

A2.1.5.6.7 Class AbstractParameter (ABSTRACT)

A2.1.5.6.7.1 General

Item	Description	Ref.	Status	Support
31.	AbstractParameter	3.7.9	C8	

C8—Mandatory if ModResParm (A2.1.5.6.6) is present.

A2.1.5.6.7.2 Class AbstractParameter Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
31.1.	name	reference [6], table 3-2	M	

A2.1.5.6.8 Class Constraints (Abstract)

A2.1.5.6.8.1 General

Item	Description	Ref.	Status	Support
32.	Constraints	3.7.10	M	

A2.1.5.6.8.2 Class Constraints Parameters

N/A.

A2.1.5.6.9 Class ConstraintsAlsoSuitableForOffline (Abstract)

A2.1.5.6.9.1 General

Item	Description	Ref.	Status	Support
33.	`ConstraintsAlsoSuitableForOffline`	3.7.11	M	

A2.1.5.6.9.2 Class ConstraintsAlsoSuitableForOffline Parameters

N/A.

A2.1.5.7 Delete Service Package request, Delete Service Package and Replace Service Package Classes

A2.1.5.7.1 Class DeleteSrvPkgReq

A2.1.5.7.1.1 General

Item	Description	Ref.	Status	Support
34.	DeleteSrvPkgReq	3.8.2	O	

A2.1.5.7.1.2 Class DeleteSrvPkgReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
34.1.	requestID	table 3-2	M	
34.2.	serviceAgreementRef	table 3-2	O	
34.3.	comment	table 3-2	O	
34.4.	spaceUserNode	table 3-3	M	
34.5.	serviceReqID	table 3-13	M	
34.6.	deletePending	table 3-12	M	

A2.1.5.7.2 Class DeleteSrvPkg

A2.1.5.7.2.1 General

Item	Description	Ref.	Status	Support
35.	DeleteSrvPkg	3.8.4	O	

A2.1.5.7.2.2 Class DeleteSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
35.1.	requestID	table 3-2	M	
35.2.	serviceAgreementRef	table 3-2	O	
35.3.	comment	table 3-2	O	
35.4.	spaceUserNode	table 3-3	M	
35.5.	serviceReqID	table 3-15	M	
35.6.	srvPkgRef	table 3-14	M	

A2.1.5.7.3 Class ReplaceSrvPkg

A2.1.5.7.3.1 General

Item	Description	Ref.	Status	Support
36.	ReplaceSrvPkg	3.8.5	O	

A2.1.5.7.3.2 Class ReplaceSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
36.1.	requestID	table 3-2	M	
36.2.	serviceAgreementRef	table 3-2	O	
36.3.	comment	table 3-2	O	
36.4.	spaceUserNode	table 3-3	M	
36.5.	serviceReqID	table 3-17	M	
36.6.	srvPkgRef	table 3-16	M	

A2.1.5.7.4 Class AbstractSrvPkg (ABSTRACT)

A2.1.5.7.4.1 General

Item	Description	Ref.	Status	Support
37.	AbstractSrvPkg	3.8.6	M	

A2.1.5.7.4.2 Class AbstractSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
37.1.	sicfRef	reference [7], table 3-20	O	

A2.1.5.7.5 Class AbstractOnlineSrvPkg (ABSTRACT)

A2.1.5.7.5.1 General

Item	Description	Ref.	Status	Support
38.	AbstractOnlineSrvPkg	3.8.7	M	

A2.1.5.7.5.2 Class AbstractOnlineSrvPkg Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
38.1.	trajectoryRef	reference [7], table 3-21	O	

A2.1.5.7.6 Class OnlineSrvPkyType1

A2.1.5.7.6.1 General

Item	Description	Ref.	Status	Support
39.	OnlineSrvPkyType1	3.8.8	M	

A2.1.5.7.6.2 Class OnlineSrvPkyType1 Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
39.1.	eventSequenceRef	reference [7], table 3-22	M	

A2.1.5.7.7 Class OnlineSrvPkyType1Details

A2.1.5.7.7.1 General

Item	Description	Ref.	Status	Support
40.	OnlineSrvPkyType1Details	3.8.9	M	

A2.1.5.7.7.2 Class OnlineSrvPkyType1Details Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
40.1.	configProfileRef	reference [7], table 3-23	M	

A2.1.5.7.8 Class OfflineSrvPkgType1

A2.1.5.7.8.1 General

Item	Description	Ref.	Status	Support
41.	OfflineSrvPkgType1	3.8.10	M	

A2.1.5.7.8.2 Class OfflineSrvPkgType1 Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
41.1.	configProfileRef	reference [7], table 3-25	M	

A2.1.5.7.9 Class DDORScanPatternDetails

A2.1.5.7.9.1 General

Item	Description	Ref.	Status	Support
42.	DDORScanPatternDetails	3.8.11	O	

A2.1.5.7.9.2 Class DDORScanPatternDetails Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
42.1.	DDORScanPatternRef	reference [7], table 3-24	M	

A2.1.5.7.10 Class ModResParm

A2.1.5.7.10.1 General

Item	Description	Ref.	Status	Support
43.	ModResParm	3.8.12	O	

A2.1.5.7.10.2 Class ModResParm Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
43.1.	resName	reference [7], table 3-8	O	

A2.1.5.7.11 Class AbstractParameter (ABSTRACT)

A2.1.5.7.11.1 General

Item	Description	Ref.	Status	Support
44.	`AbstractParameter`	3.8.13	C9	

C9—Mandatory if ModResParm (A2.1.5.7.10) is present.

A2.1.5.7.11.2 Class AbstractParameter Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
44.1.	`name`	reference [6], table 3-2	M	

A2.1.5.8 Submission Requests Classes

A2.1.5.8.1 Class SubmissionReq (ABSTRACT)

A2.1.5.8.1.1 General

Item	Description	Ref.	Status	Support
45.	SubmissionReq	3.9.2	O	

A2.1.5.8.1.2 Class SubmissionReq Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
45.1.	requestID	table 3-2	M	
45.2.	serviceAgreementRef	table 3-2	O	
45.3.	spaceUserNode	table 3-18	M	
45.4.	comment	table 3-2	O	

A2.1.5.8.2 Class AbstractDataSubWrapper (ABSTRACT)

A2.1.5.8.2.1 General

Item	Description	Ref.	Status	Support
46.	AbstractDataSubWrapper	3.9.4	M	

A2.1.5.8.2.2 Class AbstractDataSubWrapper Parameters

N/A.

A2.1.5.8.3 Class AbstractEventSeqWrapper (ABSTRACT)

A2.1.5.8.3.1 General

Item	Description	Ref.	Status	Support
47.	AbstractEventSeqWrapper	3.9.5	M	

A2.1.5.8.3.2 Class AbstractEventSeqWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
47.1.	eventSequenceID	reference [7], table 3-10	M	

A2.1.5.8.4 Class AbstractTrajDataWrapper (ABSTRACT)

A2.1.5.8.4.1 General

Item	Description	Ref.	Status	Support
48.	AbstractTrajDataWrapper	3.9.6	M	

A2.1.5.8.4.2 Class AbstractTrajDataWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
48.1.	trajectoryID	reference [7], table 3-11	M	

A2.1.5.8.5 Class OemWrapper

A2.1.5.8.5.1 General

Item	Description	Ref.	Status	Support
49.	OemWrapper	3.9.7	M	

A2.1.5.8.5.2 Class OemWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
49.1.	trajectoryID	reference [7], table 3-11	M	

A2.1.5.8.6 Class OmmWrapper

A2.1.5.8.6.1 General

Item	Description	Ref.	Status	Support
50.	OmmWrapper	3.9.8	M	

A2.1.5.8.6.2 Class OmmWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
50.1.	`trajectoryID`	reference [7], table 3-11	M	

A2.1.5.8.7 Class OpmWrapper

A2.1.5.8.7.1 General

Item	Description	Ref.	Status	Support
51.	`OpmWrapper`	3.9.9	M	

A2.1.5.8.7.2 Class OpmWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
51.1.	`trajectoryID`	reference [7], table 3-11	M	

A2.1.5.8.8 Class TleWrapper

A2.1.5.8.8.1 General

Item	Description	Ref.	Status	Support
52.	`TleWrapper`	3.9.10	M	

A2.1.5.8.8.2 Class TleWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
52.1.	`trajectoryID`	reference [7], table 3-11	M	

A2.1.5.8.9 Class Tle

A2.1.5.8.9.1 General

Item	Description	Ref.	Status	Support
53.	`Tle`	3.9.11	M	

A2.1.5.8.9.2 Class Tle Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
53.1.	tleLine0	reference [7], table 3-12	M	
53.2.	tleLine1	reference [7], table 3-12	M	
53.3.	tleLine2	reference [7], table 3-12	M	

A2.1.5.8.10 Class OtherTrajWrapper

A2.1.5.8.10.1 General

Item	Description	Ref.	Status	Support
54.	OtherTrajWrapper	3.9.12	M	

A2.1.5.8.10.2 Class OtherTrajWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
54.1.	trajectoryID	reference [7], table 3-11	M	
54.2.	trajFormat	reference [7], table 3-15	M	
54.3.	trajData	reference [7], table 3-15	M	

A2.1.5.8.11 Class AbstractConfigProfileWrapper (ABSTRACT)

A2.1.5.8.11.1 General

Item	Description	Ref.	Status	Support
55.	AbstractConfigProfileWrapper	3.9.13	M	

A2.1.5.8.11.2 Class AbstractConfigProfileWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
55.1.	configurationProfileID	reference [7], table 3-16	M	

A2.1.5.8.12 Class AbstractDDORScanPatternWrapper (ABSTRACT)

A2.1.5.8.12.1 General

Item	Description	Ref.	Status	Support
56.	`AbstractDDORScanPatternWrapper`	3.9.14	M	

A2.1.5.8.12.2 Class AbstractDDORScanPatternWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
56.1.	`ddorScanPatternID`	reference [7], table 3-17	M	

A2.1.5.8.13 Class DDORScanPatternWrapper

A2.1.5.8.13.1 General

Item	Description	Ref.	Status	Support
57.	`DDORScanPatternWrapper`	3.9.15	M	

A2.1.5.8.13.2 Class DDORScanPatternWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
57.1.	ddorScanPatternID	reference [7], table 3-17	M	

A2.1.5.8.14 Class AbstractSICFWrapper (ABSTRACT)

A2.1.5.8.14.1 General

Item	Description	Ref.	Status	Support
58.	AbstractSICFWrapper	3.9.16	M	

A2.1.5.8.14.2 Class AbstractSICFWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
58.1.	sicfID	reference [7], table 3-18	M	

A2.1.5.8.15 Class AsciiSICFWrapper

A2.1.5.8.15.1 General

Item	Description	Ref.	Status	Support
59.	AsciiSICFWrapper	3.9.17	M	

A2.1.5.8.15.2 Class AsciiSICFWrapper Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
59.1.	sicfID	reference [7], table 3-18	M	
59.2.	sicfFormat	reference [7], table 3-19	M	
59.3.	sicfData	reference [7], table 3-19	M	

A2.1.5.9 Service Management Utilization Request Formats Basic Constraints Classes

A2.1.5.9.1 Class Constraints (Abstract)

A2.1.5.9.1.1 General

Item	Description	Ref.	Status	Support
60.	`Constraints`	3.10.2	M	

A2.1.5.9.1.2 Class Constraints Parameters

N/A.

A2.1.5.9.2 Class ConstraintsAlsoSuitableForOffline (Abstract)

A2.1.5.9.2.1 General

Item	Description	Ref.	Status	Support
61.	`ConstraintsAlsoSuitableForOffline`	3.10.3	M	

A2.1.5.9.2.2 Class ConstraintsAlsoSuitableForOffline Parameters

N/A.

A2.1.5.9.3 Class BasicPass

A2.1.5.9.3.1 General

Item	Description	Ref.	Status	Support
62.	`BasicPass`	3.10.4	M	

A2.1.5.9.3.2 Class BasicPass Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
62.1.	`preferredStartTime`	table 3-19	M	
62.2.	`earliestStartTime`	table 3-19	O	
62.3.	`latestStartTime`	table 3-19	O	
62.4.	`preferredDuration`	table 3-19	M	
62.5.	`minimumDuration`	table 3-19	O	
62.6.	`maximumDuration`	table 3-19	O	

A2.1.5.9.4 Class ApertureSelection

A2.1.5.9.4.1 General

Item	Description	Ref.	Status	Support
63.	ApertureSelection	3.10.5	M	

A2.1.5.9.4.2 Class ApertureSelection Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
63.1.	priority	table 3-20	O	

A2.1.5.9.5 Class ApertureLocation (Abstract)

A2.1.5.9.5.1 General

Item	Description	Ref.	Status	Support
64.	ApertureLocation	3.10.6	M	

A2.1.5.9.5.2 Class ApertureLocation Parameters

N/A.

A2.1.5.10 Class ApertureReference

A2.1.5.10.1.1 General

Item	Description	Ref.	Status	Support
65.	ApertureReference	3.10.7	M	

A2.1.5.10.1.2 Class ApertureReference Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
65.1.	apertureRef	reference [7], table 3-5	M	
65.2.	siteRef	reference [7], table 3-5	M	

A2.1.5.10.2 Class ApertureCelestialBodyPosition

A2.1.5.10.2.1 General

Item	Description	Ref.	Status	Support
66.	ApertureCelestialBodyPosition	3.10.8	M	

A2.1.5.10.2.2 Class ApertureCelestialBodyPosition Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
66.1.	celestialBody	reference [7], table 3-6	M	
66.2.	celestialBodyLat	reference [7] table 3-6	M	
66.3.	celestialBodylong	reference [7], table 3-6	M	
66.4.	celestialBodyAltitude	reference [7], table 3-6	O	
66.5.	apertureName	reference [7], table 3-6	O	

A2.1.5.11 Service Management Utilization Request Formats Enhanced Constraints Classes

A2.1.5.11.1 Class Constraints (Abstract)

A2.1.5.11.1.1 General

Item	Description	Ref.	Status	Support
67.	Constraints	3.11.2	M	

A2.1.5.11.1.2 Class Constraints Parameters

N/A.

A2.1.5.11.2 Class StandingOrder

A2.1.5.11.2.1 General

Item	Description	Ref.	Status	Support
68.	StandingOrder	3.11.3	O	

A2.1.5.11.2.2 Class StandingOrder Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
68.1.	standingOrderStartTime	table 3-21	M	
68.2.	standingOrderDuration	table 3-21	M	
68.3.	minNumOfPasses	table 3-21	O	
68.4.	maxNumOfPasses	table 3-21	O	

A2.1.5.11.3 Class Handover

A2.1.5.11.3.1 General

Item	Description	Ref.	Status	Support
69.	Handover	3.11.4	M	

A2.1.5.11.3.2 Class Handover Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
69.1.	maximumHandovers	table 3-22	M	
69.2.	mimimumOverlap	table 3-22	M	
69.3.	maximumOverlap	table 3-22	O	
69.4.	minimunSingleDuration	table 3-22	O	

A2.1.5.11.4 Class TimeWindows

A2.1.5.11.4.1 General

Item	Description	Ref.	Status	Support
70.	TimeWindows	3.11.4	M	

A2.1.5.11.4.2 Class TimeWindows Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
70.1.	includeWindow	table 3-23	M	

A2.1.5.11.5 Class TimeWindow

A2.1.5.11.5.1 General

Item	Description	Ref.	Status	Support
71.	TimeWindow	3.11.6	M	

A2.1.5.11.5.2 Class TimeWindow Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
71.1.	startOfWindow	table 3-24	M	
71.2.	endOfWindow	table 3-24	M	

A2.1.5.11.6 Class PassRange

A2.1.5.11.6.1 General

Item	Description	Ref.	Status	Support
72.	PassRange	3.11.7	M	

A2.1.5.11.6.2 Class PassRange Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
72.1.	minRangeAtPassStart	table 3-25	O	
72.2.	maxRangeAtPassStart	table 3-25	O	
72.3.	minRangeAtPassEnd	table 3-25	O	
72.4.	maxRangeAtPassEnd	table 3-25	O	

A2.1.5.11.7 Class PassInterval

A2.1.5.11.7.1 General

Item	Description	Ref.	Status	Support
73.	PassInterval	3.11.8	M	

A2.1.5.11.7.2 Class PassInterval Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
73.1.	minIntBetweenPasses	table 3-26	O	
73.2.	maxIntBetweenPasses	table 3-26	O	

A2.1.5.11.8 Class PassDuration

A2.1.5.11.8.1 General

Item	Description	Ref.	Status	Support
74.	PassDuration	3.11.9	M	

A2.1.5.11.8.2 Class PassDuration Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
74.1.	minPassDuration	table 3-27	O	
74.2.	maxPassDuration	table 3-27	O	

A2.1.5.11.9 Class PassDailyTime

A2.1.5.11.9.1 General

Item	Description	Ref.	Status	Support
75.	PassDailyTime	3.11.10	M	

A2.1.5.11.9.2 Class PassDailyTime Parameters

Item	Parameter	Ref.	Status	Item Support or Values Supported
75.1.	daysForPasses	table 3-28	O	
75.2.	earliestPassStart	table 3-28	O	

Item	Parameter	Ref.	Status	Item Support or Values Supported
75.3.	latestPassStart	table 3-28	O	
75.4.	earliestPassEnd	table 3-28	O	
75.5.	latestPassEnd	table 3-28	O	

ANNEX B

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

B1 SECURITY CONSIDERATIONS

B1.1 OVERVIEW

This annex presents the results of an analysis of security considerations applied to the technologies specified in this Recommended Standard.

B1.2 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include potential loss, corruption, and theft of data. Since it is possible to utilize these messages in preparing and disseminating schedules relating to the availability of communications and tracking resources for spacecraft, the consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include compromise or loss of the mission if malicious tampering of a particularly severe nature occurs.

B1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

Potential threats or attack scenarios include, but are not limited to, (a) unauthorized access to the programs/processes that generate and interpret the messages, and (b) unauthorized access to the messages during transmission between exchange partners. Protection from unauthorized access during transmission is especially important if the mission utilizes open ground networks such as the Internet to provide ground station connectivity for the exchange of data formatted in compliance with this Recommended Standard. It is strongly recommended that potential threats or attack scenarios applicable to the systems and networks on which this Recommended Standard is implemented be addressed by the management of those systems and networks and the utilization of adequate authentication, suitable protocols, and secured interfaces for the exchange of this information.

B1.4 SECURITY CONCERNS RELATED TO THIS RECOMMENDED STANDARD

B1.4.1 Data Privacy

Privacy of data formatted in compliance with the specifications of this Recommended Standard should be assured by the systems and networks on which this Recommended Standard is implemented.

B1.4.2 Data Integrity

Integrity of data formatted in compliance with the specifications of this Recommended Standard should be assured by the systems and networks on which this Recommended Standard is implemented.

B1.4.3 Authentication of Communicating Entities

Authentication of communicating entities involved in the transport of data which complies with the specifications of this Recommended Standard should be provided by the systems and networks on which this Recommended Standard is implemented.

B1.4.4 Data Transfer between Communicating Entities

The transfer of data formatted in compliance with this Recommended Standard between communicating entities should be accomplished via secure mechanisms approved by the Information Technology Security functionaries of exchange participants.

B1.4.5 Control of Access to Resources

Control of access to resources should be managed by the systems upon which provider formatting and recipient processing are performed.

B1.4.6 Auditing of Resource Usage

Auditing of resource usage should be handled by the management of systems and networks on which this Recommended Standard is implemented.

B1.5 UNAUTHORIZED ACCESS

Unauthorized access to the programs/processes that generate and interpret the messages should be prohibited in order to minimize potential threats and attack scenarios.

B1.6 DATA SECURITY IMPLEMENTATION SPECIFICS

Specific information-security interoperability provisions that apply between agencies and other independent users involved in an exchange of data formatted in compliance with this Recommended Standard should be specified in an ICD.

B2 SANA CONSIDERATIONS

B2.1 GENERAL

The recommendations of this document rely on the SANA registries described below. New assignments in these registries, in conformance with the policies identified, will be available at the SANA registry Web site: http://sanaregistry.org. Therefore the reader shall look at the SANA Web site for all the assignments contained in these registries.

B2.2 REGISTRY FOR ORIGINATINGORGANIZATION

(See annex subsection A2.2 of reference [7].)

B2.3 REGISTRY FOR SPACEUSERNODE

(For `spaceUserNode` see annex subsection A2.3 of reference [7]**.)**

In addition to the values for the '`spaceUserNode`' parameter contained in the CCSDS Spacecraft Identifiers registry, the following values need to be recognized by implementations (see table 3-6):

`user`	**Description**
UNALLOCATED	Indicates that the time is unallocated.

B2.4 REGISTRY FOR SITEREF AND APERTUREREF

(See annex subsection A2.4 of reference [7].)

B2.5 USE OF UNREGISTERED VALUES

Only values that have been registered should be used for the `originatingOrganization`, `spaceUserNode`, `siteRef` and `apertureRef` parameters. Unregistered values for the `originatingOrganization`, `spaceUserNode`, `siteRef` and `apertureRef` parameters may be used. If unregistered values are used they should be prefixed with the string 'UNR::'.

NOTES

1 'UNR::' indicates an unregistered value;

2 this helps eliminate potential confusion in a multi-agency cross support context;

3 use of unregistered values is not recommended and should be avoided if possible.

B3 PATENT CONSIDERATIONS

No patent rights are known to apply to any of the specifications of the Recommended Standard.

ANNEX C

XML SCHEMA ORGANIZATION, PACKAGING AND EXAMPLES FOR THE SERVICE MANAGEMENT UTILIZATION REQUEST FORMATS.

(INFORMATIVE)

C1 PURPOSE

This annex provides an informative information regarding the XML Schema Organization and packaging.

C2 SCHEMA ORGANIZATION AND PACKAGING

The normative Service Management Utilization Request schema types and global elements are contained in the file '902x09w0_19-Smurf.xsd'.

The Service Management Utilization Request types and global elements are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

The Service Management Utilization Request schema includes the following schemas:

a) 902x12m1-SmCmnEnt-InfEntHdr.xsd

   – Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

b) 902x12m1-SmCmnEnt-AbsEvnt.xsd

   – Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

c) 902x12m1-SmCmnEnt-CmnClss.xsd

   – Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

d) 902x12w1_04-SmCmnEnt-DataWrp.xsd

   – Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

e) 902x12w1_04-SmCmnEnt-SrvPkg.xsd

   – Types and global elements in this schema are registered in the 'urn:ccsds:schema:cssm:1.0.0' name space.

f) 902x09w0_19-Smurf-EnhancedCons.xsd

   – Types and global elements in this schema are registered in the ‘urn:ccsds:schema:cssm:1.0.0’ name space.

The source of the following schema files,

– 902x09w0_19-Smurf.xsd

– 902x12m1-SmCmnEnt-InfEntHdr.xsd

– 902x12m1-SmCmnEnt-AbsEvnt.xsd

– 902x12m1-SmCmnEnt-CmnClss.xsd

– 902x12w1_04-SmCmnEnt-DataWrp.xsd

– 902x12w1_04-SmCmnEnt-SrvPkg.xsd

– 902x09w0_19-Smurf-EnhancedCons.xsd,

is the SANA SCCS-SM Information Entity XML Schemas registry:

http://sanaregistry.org/r/service_management_xml_schemas

C3 EXAMPLES

Various examples of the use of SMURF are available. These are maintained in a git repository, the location of which is:
https://github.com/cssAreaGH/CSSM-XML-InstanceExamples

ANNEX D

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee on Space Data Systems
DDOR	Delta-Differential One-Way Ranging
ICD	interface control document
N/A	not applicable
OMG	Object Management Group
SANA	Space Assigned Numbers Authority
SICF	Service Instance Configuration File
SMURF	Service Management Utilization Request Format
TBD	to be decided
UML	Unified Modelling Language
UTC	Coordinated Universal Time
W3C	World Wide Web Consortium
XML	eXtensible Markup Language