The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

CROSS SUPPORT TRANSFER SERVICE SPECIFICATION FRAMEWORK CONCEPT

INFORMATIONAL REPORT

CCSDS 920.0-G-1

Green Book
May 2023

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

CROSS SUPPORT TRANSFER SERVICE SPECIFICATION FRAMEWORK CONCEPT

INFORMATIONAL REPORT

CCSDS 920.0-G-1

Green Book
May 2023

AUTHORITY

Issue:	Informational Report, Issue 1
Date:	May 2023
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

This document is published and maintained by:

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

FOREWORD

This document is a technical **Report** for use in developing ground systems for space missions and has been prepared by the **Consultative Committee for Space Data Systems** (CCSDS). The Cross Support Transfer Service Specification Framework described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Report is therefore subject to CCSDS document management and change control procedures, which are defined in *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.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 920.0-G-1	Cross Support Transfer Service Specification Framework Concept, Informational Report, Issue 1	May 2023	Original issue

CONTENTS

Section Page

CONTENTS (continued)

CONTENTS (continued)

Figure Page

Table

1 INTRODUCTION

1.1 PURPOSE

This Report is an adjunct document to the CCSDS Recommended Standards for the Cross Support Transfer Service (CSTS) Specification Framework Recommended Standard (reference [1]) and for the Guidelines for the Specification of Cross Support Transfer Services Recommended Practice (reference [2]) and it contains material that will be helpful in understanding the primary documents.

1.2 SCOPE

This Report provides supporting descriptive and tutorial material with regard to the framework for all of the Cross Support Transfer Services used in exchanging data between spacecraft mission operations centers and ground stations. This document is not part of the Recommended Standards. In the event of conflicts between this Report and the Recommended Standards, the Recommended Standards are the controlling specifications.

1.3 DOCUMENT STRUCTURE

1.3.1 ORGANIZATION OF THIS REPORT

This Report is organized as follows:

a) section 1 defines the purpose and scope of this document, provides the definition of terms, and provides the references used in this Report;

b) section 2 provides the background and objectives of the development of the CSTS Specification Framework and explains the relationship to the SLE service specifications;

c) section 3 provides an overview of the CSTS Specification Framework;

d) section 4 summarizes concepts from the reference model on which the definition of Cross Support Transfer Services is based;

e) section 5 explains the approach to extensibility, which is a core concept of the CSTS Specification Framework;

f) section 6 describes the common operations defined by the CSTS Specification Framework;

g) section 7 describes the common procedures defined by the CSTS Specification Framework;

h) section 8 describes the role of a service instance and its relationship to procedure instances within the CSTS Specification Framework;

i) section 9 explains how data communications services are addressed within the CSTS Specification Framework;

j) section 10 summarizes the Guidelines for the Specification of CSTSes on the basis of the CSTS Specification Framework;

k) annex A lists acronyms and abbreviations used in this Report;

l) annex B outlines examples of CSTS specifications.

1.3.2 CROSS SUPPORT TRANSFER SERVICES DOCUMENTATION

The basic organization of the Cross Support Services documentation and the relationship to CSTS documentation is shown in figure 1-1.

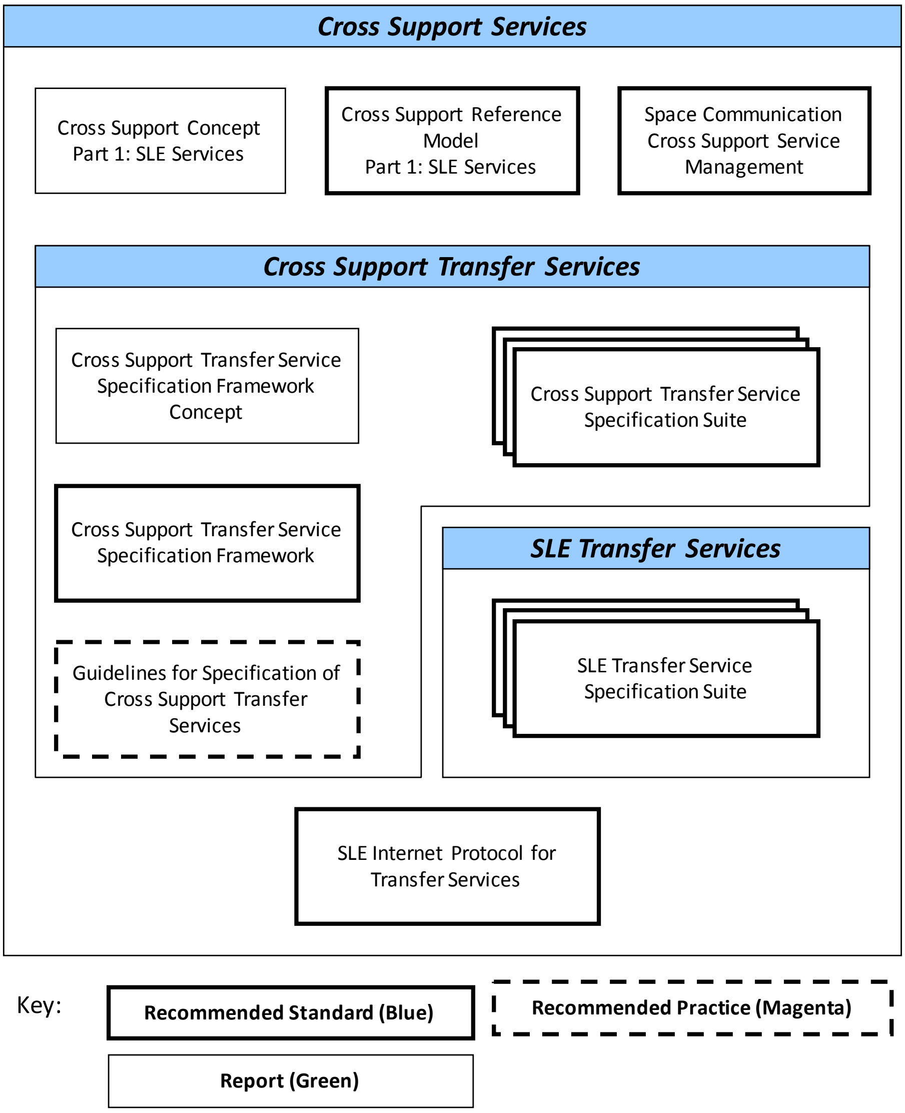
Cross Support Services
Cross Support Concept Part 1: SLE Services
Cross Support Reference Model Part 1: SLE Services
Space Communication Cross Support Service Management
Cross Support Transfer Services
Cross Support Transfer Service Specification Framework Concept
Cross Support Transfer Service Specification Suite
Cross Support Transfer Service Specification Framework
SLE Transfer Services
Guidelines for Specification of Cross Support Transfer Services
SLE Transfer Service Specification Suite
SLE Internet Protocol for Transfer Services
Key:
Recommended Standard (Blue)
Recommended Practice (Magenta)
Report (Green)

Figure 1-1: Cross Support Service Documentation

The Cross Support Service documents that are related to Cross Support Transfer Services are:

Pertaining to Cross Support Architecture:

a) Space Communications Cross Support—Architecture Description Document (reference [21]): An Informational Report describing an architecture in terms of CCSDS-recommended configurations for secure space communications cross support. This architecture is intended to be used as a common framework when CCSDS Agencies a) provide and use space communications cross support services and b) develop systems that provide interoperable space communications cross support.

b) Space Communications Cross Support—Architecture Requirements Document (reference [22]): A Recommended Practice defining a set of requirements for CCSDS-recommended configurations for secure Space Communications Cross Support architectures.

Common to all Cross Support Services:

c) *Space Communication Cross Support Service Management suite* (references [7], [6], and [20]). Data-format Recommended Standards that specify the Service Management Information Entities that are used to configure and schedule cross support services, which include transfer services.

Common to the Transfer Services, that is, SLE Transfer Services and Cross Support Transfer Services:

d) *Space Link Extension—Internet Protocol for Transfer Services* (reference [3]): A Recommended Standard that defines a protocol for transfer of PDUs defined in the Cross Support Transfer Services. This Recommended Standard was originally developed to support SLE transfer services (hence the title), but it is also applicable to use by Cross Support Transfer Services.

Pertaining to the concept, the reference model, and the SLE Transfer Services:

e) *Cross Support Concept—Part 1: Space Link Extension Services* (reference [4]): A report introducing the concepts of cross support and the SLE services. Many of the concepts for the SLE transfer services have been adopted for the CSTSes (see k) below).

f) *Cross Support Reference Model—Part 1: Space Link Extension Services* (reference [5]): A Recommended Standard that defines the framework and terminology for the specification of SLE services. Much of the framework and terminology of this reference model has been adopted or adapted for CSTSes (see 1.4.1 and 4.2).

g) The *SLE Transfer Services* suite: A group of Cross Support Services used to transfer specific telecommand and telemetry protocol data units. The SLE Transfer Services are closely related to the CSTS suite in that they collectively define the set of operations that are the basis for the CSTS Specification Framework. However, as the

SLE Transfer Services were already specified and implemented prior to development of the CSTS Specification Framework, the SLE Transfer Services are separated from CSTSes.

Specific to Cross Support Transfer Services:

h) *Cross Support Transfer Services Specification Framework* (reference [1]): A Recommended Standard that defines the specification of the Cross Support Transfer Service procedures.

i) *Guidelines for Specification of Cross Support Transfer Services* (reference [2]): A Recommended Practice that defines the guidelines for construction of a Cross Support Transfer Service based on the CSTS Specification Framework.

j) *Cross Support Transfer Services Concept* (this Report): A Report that provides tutorial material on the objectives and concepts of the CSTS specification.

k) *Cross Support Transfer Services Suite*: The set of specifications for actual CSTSes built from the procedures in the CSTS Specification Framework and in accordance with the CSTS Guidelines.

1.4 DEFINITIONS

1.4.1 TERMS

1.4.1.1 abstract CSTS: A CSTS that is intended to serve as the basis of multiple concrete CSTSes that are to be created through refinement or extension of that abstract service.

1.4.1.2 acknowledgement: A confirmation that the invocation of a three-phase operation sent by the invoker was accepted by the performer and will now be acted upon.

1.4.1.3 association: A cooperative relationship between a service-providing application entity and a service-using application entity. An association is formed by exchanging service protocol data units using an underlying communications service.

1.4.1.4 binding: The act of establishing an association. Before operations can be invoked, an association has to be established between the ports associated with the specific CSTS instance.

1.4.1.5 concrete CSTS: A CSTS that is intended to be directly implemented as specified. A concrete CSTS may also serve as the basis of further derived CSTSes by refinement and/or extension.

1.4.1.6 confirmed operation: An operation that requires the performer to return a report of the operation outcome to the invoker. Confirmed operations are further classified as two-phase operations and three-phase operations.

1.4.1.7 cross support: The term applied when one Agency uses part of another Agency's resources to complement its own for the purpose of operating a spacecraft.

1.4.1.8 Cross Support Service: A set of capabilities that an object that belongs to one Space Agency provides to objects that belong to other Space Agencies by means of one or more ports, in support of spacecraft operations; a Cross Support Service can also be used within a given Space Agency.

1.4.1.9 Cross Support Service provision: Exposure of the operations necessary so that the service user can obtain the service; provision involves the interface between the service user and the service provider.

1.4.1.10 Cross Support Transfer Service, CSTS: A subclass of Cross Support Service that provides reliable, access-controlled transfer of spaceflight mission-related data between ground element entities, realized through the invocation and performance of defined operations in accordance with defined procedures. A CSTS is qualified by the kind of data it transfers (e.g., telemetry frames, tracking data). A CSTS may optionally have capabilities to coordinate and monitor the behavior of the Cross Support Service production with which this service is associated.

1.4.1.11 Cross Support Transfer Service instance: An instance of a specific Cross Support Transfer Service type by means of which a Provider Cross Support Service System (CSSS) provides the capability to the service user to transfer the service-type specific spaceflight mission-related data.

1.4.1.12 Cross Support Transfer Service production: Performance of the data acquisition process and/or the data transformation as necessary for the given type of Cross Support Transfer Service.

1.4.1.13 derivation: A mechanism that allows extending or refining an operation or the behavior of a procedure. This mechanism can also be applied to a Cross Support Transfer Service.

1.4.1.14 extension: The act of extending operations or procedures. Operations are extended by (a) adding new parameters to the invocation or response message, (b) extending the range of values for already existing parameters, or (c) changing an unconfirmed operation into a confirmed operation. Procedures are extended by (a) adding new operations or (b) extending the used operations.

1.4.1.15 Framework: A set of generic behaviors (called common procedures) and common operations used to simplify the specification of systems providing or using CSTSes.

1.4.1.16 initiator: The entity that issues the request to bind to another entity (the responder).

1.4.1.17 invocation: The making of a request by an entity (the invoker) to another entity (the performer) to carry out the invoked operation.

1.4.1.18 operation: A task that the invoker requests the performer to execute. Depending on the type of operation, the performer may or may not report the result of the operation to the invoker. Service user and service provider interact by invoking and performing operations.

1.4.1.19 parameter: In the context of an operation, data that may accompany the operation's invocation, acknowledgement, or return. The term parameter is also used to refer to mission-dependent configuration information used in the production or provision of a Cross Support Service.

1.4.1.20 performance: In the context of an operation, the execution of the operation by an object (the performer).

1.4.1.21 production status: The aggregate status of the Cross Support Service production processes that generate or process the data that is transferred by the transfer service instance.

1.4.1.22 real-time data: Data that can be accessed by the service user as soon as it is collected or generated by the service provider.

1.4.1.23 recorded data: Data that has been collected and stored by the service provider for access by the service user at some later time.

1.4.1.24 refinement: The act of refining operations or procedures. Operations are refined by constraining the values of parameters or by detailing the parameter semantics. Procedures are refined by modifying (e.g., narrowing) their semantics or defining their behavior or states in more detail.

1.4.1.25 responder: (See **initiator**.)

1.4.1.26 response: A report, from the performer to the invoker, in reaction to the invocation of an operation. A response may be a return or, in the case of a three-phase operation, an acknowledgement or a return.

1.4.1.27 return: A report, from the performer to the invoker, of the outcome of the performance of the operation.

1.4.1.28 service instance provision period: The time during which a service is scheduled to be provided by this instance.

1.4.1.29 service provider: An entity that offers a service to another by means of one or more of its ports.

1.4.1.30 service user: An entity that uses a service that is provided by another entity by means of one or more of its ports.

1.4.1.31 three-phase operation: An operation that requires the performer to return to the invoker an initial acknowledgement of the acceptance of the invocation in addition to the report of the execution of the operation, the return.

1.4.1.32 unconfirmed operation: An operation for which the performer does not return a report of its outcome to the invoker.

1.4.2 CONVENTIONS

Whenever useful to aid the understanding of the basic concept of the Cross Support Transfer Service Specification Framework (reference [1]) and how CSTSes are specified, these concepts are compared to the SLE concepts, and differences and similarities are discussed. The format of these comparison is shown below.

SLE Reference

Throughout this Report, hints presented in this format point to similarities and differences between SLE service specifications and the CSTS Specification Framework.

1.5 REFERENCES

The following publications are referenced in 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] *Cross Support Transfer Service—Specification Framework*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 921.1-B-2. Washington, D.C.: CCSDS, February 2021.

[2] *Guidelines for the Specification of Cross Support Transfer Services*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 921.2-M-1. Washington, D.C.: CCSDS, March 2019.

[3] *Space Link Extension—Internet Protocol for Transfer Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 913.1-B-2. Washington, D.C.: CCSDS, September 2015.

[4] *Cross Support Concept—Part 1: Space Link Extension Services*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 910.3-G-3. Washington, D.C.: CCSDS, March 2006.

[5] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 910.4-B-2. Washington, D.C.: CCSDS, October 2005.

[6] *Cross Support Service Management—Service Management Utilization Request Formats*. Issue 1. Draft Recommendation for Space Data System Standards (Red Book), CCSDS 902.9-R-1. Washington, D.C.: CCSDS, September 2022.

[7] *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.

[8] *Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*. 5th ed. International Standard, ISO/IEC 8824-1:2015. Geneva: ISO, 2015.

[9] *Cross Support Transfer Services—Monitored Data Service*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 922.1-B-2. Washington, D.C.: CCSDS, September 2022.

[10] *Space Link Extension—Return All Frames Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 911.1-B-4. Washington, D.C.: CCSDS, August 2016.

[11] *Space Link Extension—Return Channel Frames Service Specification*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 911.2-B-3. Washington, D.C.: CCSDS, August 2016.

[12] *Space Link Extension—Return Operational Control Fields Service Specification*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 911.5-B-3. Washington, D.C.: CCSDS, August 2016.

[13] *Space Link Extension—Forward CLTU Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 912.1-B-4. Washington, D.C.: CCSDS, August 2016.

[14] *Space Link Extension—Forward Space Packet Service Specification*. Issue 3-S. Recommendation for Space Data System Standards (Historical), CCSDS 912.3-B-3-S. Washington, D.C.: CCSDS, (August 2016) September 2022.

[15] *TM Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-3. Washington, D.C.: CCSDS, October 2021.

[16] *TC Synchronization and Channel Coding*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-4. Washington, D.C.: CCSDS, July 2021.

[17] *TC Space Data Link Protocol*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-4. Washington, D.C.: CCSDS, October 2021.

[18] *Communications Operation Procedure-1*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.1-B-2. Washington, D.C.: CCSDS, September 2010.

[19] *Cross Support Transfer Service—Tracking Data Service*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 922.2-B-2. Washington, D.C.: CCSDS, February 2023.

[20] *Extensible Space Communication Cross Support—Service Management—Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1. Washington, D.C.: CCSDS, September 2014.

[21] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1. Washington, D.C.: CCSDS, November 2013.

[22] *Space Communications Cross Support—Architecture Requirements Document*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1. Washington, D.C.: CCSDS, May 2015.

[23] *Cross Support Service Management—Service Package Data Formats*. Issue 1. Draft Recommendation for Space Data System Standards (Red Book), CCSDS 902.4-R-1. Washington, D.C.: CCSDS, September 2022.

[24] *Cross Support Transfer Service—Forward Frame Service*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 922.3-B-1. Washington, D.C.: CCSDS, April 2021.

2 BACKGROUND AND OBJECTIVES

2.1 SPACE LINK EXTENSION SERVICES

In 1991, CCSDS embarked on the specification of Space Link Extension (SLE) services for cross support between ground facilities operated by different Space Agencies. As the name suggests, SLE services extend and complement the space link services previously specified by CCSDS and support exchange of telemetry and telecommand data between TT&C networks and mission user facilities. Reference [5] specifies a formal reference model for the definition of SLE services; an introduction to SLE services of more tutorial nature can be found in reference [4].

While the Cross Support Reference Model (reference [5]) identifies all SLE services that are considered technically feasible, the services that have actually been specified to be detailed and implemented include only the following:

a) for the Return Link:

   1) Return All Frames (RAF, reference [10]) to transmit to the SLE service user all telemetry frames received on a physical space link;

   2) Return Channel Frames (RCF, reference [11]) to transmit to the SLE service user all frames received on a Master Channel or on a selected Virtual Channel as defined by the Telemetry Space Data Link Protocol (reference [15]);

   3) Return Operational Control Fields (ROCF, reference [12]) to transmit to the SLE service user Operational Control Fields extracted from frames conforming to the Telemetry Space Data Link Protocol (reference [15]).

b) for the Forward Link:

   1) Forward CLTU (reference [13]) to receive from the SLE service user Communications Link Transmission Units (CLTU) as specified by the Recommended Standard for Telecommand Synchronization and Channel Coding (reference [16]) and to radiate them on the space link;

   NOTE – This service allows as well facilities to receive from an SLE service user a non-CCSDS data structure (any octet aligned bit pattern) and to radiate it via the space link.

   2) Forward Space Packet (FSP, reference [14]) to receive from the SLE service user telecommand packets and transmit them to the spacecraft as specified by the TC Space Data Link Protocol (reference [17]) and the Communications Operation Procedure-1 (reference [18]).

These services are referred to as SLE Transfer Services because they are used to transfer data between a service user and a service provider. Beside the Transfer Services, CCSDS has developed Recommended Standards for the management of Cross Support Services (references [6], [7], and [20]).

SLE services have first been used operationally in 2002, and since then, their use has been steadily increasing not only for cross support between Space Agencies but also to replace proprietary intra-Agency telemetry and telecommand ground data transfer services. Today SLE services have become the standard way of interfacing ground facilities for the exchange of telemetry and telecommand data.

2.2 NEXT GENERATION CROSS SUPPORT TRANSFER SERVICES

The importance of cross support in space missions keeps increasing and with it the demand for further standardized Cross Support Services, not only for the exchange of space link data but also for other data such as radiometric measurements or space link monitoring data.[1]

Because of the success of SLE services, and because of the investment already made in the development and deployment of SLE services, it was clear that the new Cross Support Services would have to apply the same concepts as the SLE Transfer Services and would have to be compatible with Cross Support Service Management. However, CCSDS Member Agencies also demanded that the effort required for specification and for implementation of new Cross Support Services be significantly reduced. In order to meet these requirements, the responsible CCSDS Working Group analyzed the existing SLE service specifications and obtained the results presented in what follows.

Although SLE services are all based on the same concepts and share a significant amount of common specifications, each of the SLE service specifications has been developed as a self-contained document. While the availability of a self-contained specification has its advantages, the approach also implies some drawbacks:

a) a large amount of information is duplicated between the specifications, and maintaining consistency of the overall SLE service documentation is work intensive and error prone;

b) although operations defined by different specifications are typically similar, they frequently differ in certain details, which impedes reusable implementation;

c) the data types have been specified specifically for each service and therefore often have different names despite referring to more or less the same information item, which again limits reusability.

Because of these issues, developing such a specification ‘from scratch’ requires a large effort, and implementation, testing, and deployment is expensive.

[1] The Cross Support Reference Model has anticipated extension of ground Cross Support Services to other domains, which it refers to as ‘ground domain services’. However, the contents and structure of such services were not itemized.

On the other hand, all specifications apply the same basic communication patterns. By generalizing those communication patterns, it is possible to specify reusable building blocks that can be called up by service specifications, specializing them when necessary. This method is expected to simplify the standard development process and to produce smaller standards that are easier to review. Finally, this approach has the potential to foster the development of tested reusable software components in order to reduce development and validation cost.

2.3 CSTS SPECIFICATION FRAMEWORK

Because of the considerations presented in the previous section, the next generation Cross Support Transfer Services will be based on a specification framework that provides a set of reusable specifications as building blocks and the rules for construction of Cross Support Transfer Services with these building blocks.

The CSTS Specification Framework clearly builds on SLE concepts, taking into account lessons learned in implementation and operation of these services, when applicable. In order to be reusable for a wide range of transfer services, SLE specifications cannot be adopted directly but must be generalized and abstracted. In order to be useful for specific services, there must be mechanisms by which such abstract building blocks can be specialized to fit the requirements of the service. This process of specializing a generic specification provided by the CSTS Specification Framework is referred to as ‘derivation’.

The CSTS Specification Framework has been created to develop new services, for example, the MD-CSTS (reference [9]) and the TD-CSTS (reference [19]), while the existing SLE telemetry and telecommand services will be retained for some time. However, as will be demonstrated in annex B, it would be possible to rework the existing SLE Service Specifications to conform to the CSTS Specification Framework. CSTS Transfer Services are built on specifications written in ASN.1, a proven technology to create interoperable implementations.

3 OVERVIEW

CSTSes provide for reliable, access-controlled transfer of spaceflight mission related data between ground element entities. A Cross Support Transfer Service is characterized by the kind of data it transfers (e.g., telemetry data, tracking data, service production monitored data) and therefore different CSTSes need to respond to specific requirements that may demand specific solutions. On the other hand, all CSTSes defined by CCSDS apply the same basic communication patterns in order to simplify specification, implementation, and operation of these services. These basic communication patterns are specified by the CSTS Specification Framework.

The CSTS Specification Framework is defined by two specifications:

a) *Cross Support Transfer Service—Specification Framework* (reference [1]) defines reusable building blocks that can be called up in a CSTS specification; and

b) *Guidelines for the Specification of Cross Support Transfer Services* (reference [2]) defines the rules for constructing CSTS specifications with these building blocks.

The objectives of these specifications are to

– maximize the commonality between CCSDS Cross Support Transfer Services;

– simplify specification of new CCSDS Cross Support Transfer Services; and

– enable the design and implementation of reusable software components with the potential of simplifying the implementation of CSTSes.

While the CCSDS specifications have been prepared with the explicit intention to support development of reusable software components, CCSDS does not provide a specification of such components.

The basic approach applied to CSTSes is that they are realized through invocation and performance of operations in accordance with well-defined procedures. The building blocks defined in the CSTS Specification Framework therefore are:

a) data types for information exchanged between the service user and the service provider;

b) common operations that are invoked by one entity and performed by the other entity and as such implement the basic elements of interaction between the service user and the service provider; and

c) common procedures that define the behavior and protocol for the invocation and performance of a set of operations to achieve a well-defined objective.

A CSTS may be defined by combining a set of procedures specified in the CSTS Specification Framework in a manner that best suits the objective of the service. The elements that make up the CSTS Specification Framework and their use by CSTS specifications are illustrated by figure 3-1.

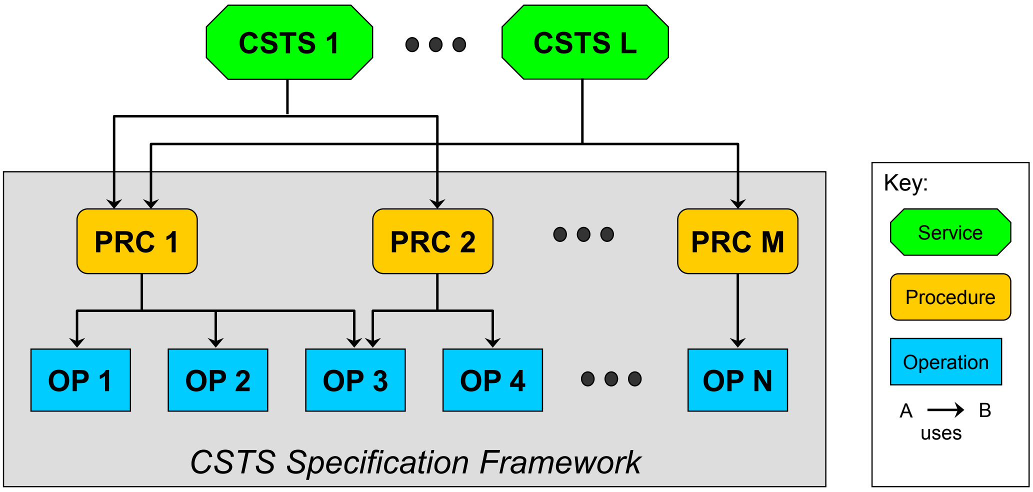

Figure 3-1: Elements of the CSTS Specification Framework

SLE Reference

The concept of data types and operations is the same as in SLE Services. In fact, most of the operations specified in the CSTS Specification Framework present a direct generalization of operations found in SLE Services. The concept of procedures that encapsulate specific parts of the protocol or behavior of a service is a new concept introduced by the CSTS Specification Framework.

To be generally applicable, parameters, operations, and procedures defined by the CSTS Specification Framework are sufficiently abstract such that they may or may not be directly usable for real services. For cases in which the building blocks are not directly usable, new operations or procedures may be derived from those defined in the CSTS Specification Framework. The Guidelines (reference [2]) describe mechanisms by which

a) procedures can be extended by adding operations or can be refined by detailing the behavior or narrowing the semantics associated with the procedure;

b) new procedures can be specified by combining existing operations or new operations derived from those defined in the CSTS Specification Framework;

c) within new or derived procedures, operations can be extended by adding new parameters to the invocation or response PDU, by extending the range of values for already existing parameters, or by changing an unconfirmed operation to a confirmed operation. Operations can be refined by constraining the values of parameters or by detailing the parameter semantics.

Taking the concept of generic definitions one step further, the CSTS Specification Framework supports the concept of abstract services that specify the common operations and behavior of CSTSes but are not directly implementable because service-specific specifications are still missing. Real CSTSes can be derived from such abstract services. An example of an abstract service could be a CSTS Return Link Service from which real services such as RAF-CSTS, RCF-CSTS, and ROCF-CSTS could be derived. These CSTSes would be equivalent to SLE RAF, SLE RCF, and SLE ROCF.

Likewise, the CSTS Specification Framework supports the concept of abstract procedures. An abstract procedure cannot be implemented directly because its specification is incomplete. An example of an abstract procedure is the Data Processing procedure, the purpose of which is to provide for the transfer of to-be-processed data units from the service user to the service provider. This procedure does neither define the data units being transferred nor the processing to be performed by the service provider. It is assumed, however, that processing is part of Cross Support Service production and will be specified by the service using this procedure.

The Buffered Data Processing procedure extends the Data Processing procedure, and in particular, it provides specifications for the handling of Input Queue overflow conditions that are left unspecified in the Data Processing procedure. The specification of this behavior makes the Buffered Data Processing procedure implementable.

The Sequence-Controlled Data Processing procedure is also an extension of the Data Processing procedure. This procedure is intended for use in Cross Support Transfer Services that involve sequence-controlled data transfer and processing and that need confirmation of data transfer and status reports on the ongoing Cross Support Service production process.

The extensibility mechanisms provided by the CSTS Specification Framework are discussed in more detail in section 5, and their use by CSTS specifications is explained in section 10.

SLE Reference

Extensibility based on the concept of derivation of new, more specialized operations and procedures from those specified in the CSTS is a core concept of the CSTS Specification Framework that does not exist in SLE Service specifications. In contrast, each SLE Service has been specified by a self-contained Recommended Standard. As explained in 2.2, this approach has some problems, which the CSTS Specification Framework intends to solve.

In order not to restrict the applicability of service specifications based on the CSTS Specification Framework to a specific communication technology, as few assumptions as possible are made about the characteristics of the underlying communications service (see section 9). While communicating parties can in principle agree on any communications service that meets the requirements, the default communications service foreseen by the CSTS Specification Framework is the Internet SLE Protocol One (ISP1, reference [3]). ISP1 is based on the Internet protocol suite (IP and TCP) and uses the Abstract Syntax Notation One (ASN.1, reference [8]) Basic Encoding Rules as transfer syntax.

The CSTS Specification Framework specifies data types used in the common operations in the ASN.1. If ISP1 is used as communications service, then these data types are encoded using the ASN.1 Basic Encoding Rules. This does not, however, imply that services based on the CSTS Specification Framework must use ASN.1. When deriving new operations, service specifications may use any data representation for parameters added to the operations.

SLE Reference

With respect to data communication and data encoding, the CSTS Specification Framework takes essentially the same approach as has been applied to SLE Services. The CSTS Specification Framework goes one step further in that it foresees use of ISP1 as the default communications service.

Support for non-ASN.1 data representation in operation extensions is a new concept introduced by the CSTS Specification Framework (see annex F of the CSTS Specification Framework for details).

4 CROSS SUPPORT REFERENCE MODEL

4.1 INTRODUCTION

CCSDS Cross Support Transfer Services are defined within a CSTS Specification Framework that is a logical extension of the Cross Support Reference Model (reference [5]) that was originally defined for Space Link Extension services. The following subsections summarize concepts from the reference model that are supported by the CSTS Specification Framework and to which later sections in this Report refer. Tutorial material on the concepts defined by the Reference Model may be found in reference [4].

NOTE – The formal definition of the terms used in this Report is provided in 1.4.1. For convenience, the definition of some of these terms is repeated and, in some cases, extended in this section.

SLE Reference

All aspects of the Cross Support Reference Model that are not directly related to processing of space link data units have been adopted for the new generation Cross Support Transfer Services. Those parts of the model that deal with space link data units and Space Link Extension have been generalized and formulated in a more abstract manner.

4.2 BASIC DEFINITIONS

A **Cross Support Service** is defined as a set of capabilities that an entity of one Space Agency provides to entities that belong to other Space Agencies by means of one or more ports, in support of spacecraft operations.

NOTE – Cross Support Services can also be used between entities belonging to the same Agency.

Cross Support Transfer Service is a subclass of Cross Support Service that provides reliable, access-controlled transfer of spaceflight mission-related data between ground element entities, realized through the invocation and performance of defined operations in accordance with defined procedures. A CSTS is qualified by the kind of data that it transfers (e.g., telemetry frames, tracking data). A CSTS may have capabilities to coordinate and monitor the behavior of the Cross Support Service production with which this service is associated.

Cross Support Service production refers to the common processes performed by service production of an Earth Space Link Terminal (ESLT) associated with the provision of one or more Cross Support Services.

A **Cross Support Transfer Service instance** is an instance of a specific Cross Support Transfer Service type by means of which an ESLT provides the capability to the service user to transfer the service-type specific spaceflight mission-related data. There may be CSTSes that operate in other Provider CSSS nodes than an ESLT. In such cases, ‘Element Management’ would refer to the management functions that have local purview over the provision and production of those CSTSes.

The **service instance provision period** is the time during which a service is scheduled to be provided by this instance.

An entity that offers a service to another by means of one or more of its ports is called a **service provider**. An ESLT is said to **provide** a service when it makes available to the service user the capability to obtain the service via one or more of its ports. Provision involves the interface between the service user and the service provider and is characterized by the type(s) of data transferred and the quality of service with which they are transferred (e.g., completely, reliably).

An entity that uses a service that is provided by another entity by means of one or more of its ports is called a **service user**.

A service user and a service provider interact by invoking and performing operations. An **operation** is a task that the **invoker** requests the **performer** to execute. Depending on the type of operation, the performer may or may not report the result of the operation to the invoker.

Before operations can be invoked, an association has to be established between the ports associated with the specific CSTS instance. The act of establishing such an association is called **binding**.

A set of resources that are operated by a single management authority to provide Cross Support Services to a space flight mission is called a **Provider CSSS**.

4.3 SERVICE PRODUCTION AND PROVISION

The Cross Support Reference Model (reference [5]) distinguishes between service production and service provision.

Cross Support Service production is the performing of the data acquisition process and/or the data transformation, as necessary for the given type of Cross Support Transfer Service.

A Provider CSSS is said to *provide* a service when it makes available to the service user the capability to obtain the service via one or more of its ports. Cross Support Service provision involves the interface between the service user and the service provider, and is characterized by the type(s) of data transferred, when the data are transferred, and the quality of service with which they are transferred.

The CSTS Specification Framework deals primarily with Cross Support Service provision. Cross Support Service production is generally very specific for specific services or a class of services and therefore the CSTS Specification Framework only makes very general assumptions on Cross Support Service production.

Service production status is the aggregate status of the production processes that generate, transform, or otherwise process the data that is transferred by the Cross Support Service instance.

The status of the service production resource is reported by means of the `production-status` parameter and is made available to all service instances accessing the same production resource. The permissible values of `production-status` and the transitions between these values, as defined by the CSTS Specification Framework, are shown in figure 4-1. Service specifications based on the CSTS Specification Framework may refine this diagram and may also define sub-states for one or more of the `production-status` values as necessary.

Production status changes are notified to the service user via the Notification procedure (see 7.10) or via associated NOTIFY operations included in other procedures. The current value of the `production-status` may be obtained using the Cyclic Report procedure (see 7.4) or the Information Query procedure (see 7.8) if that is supported by the service.

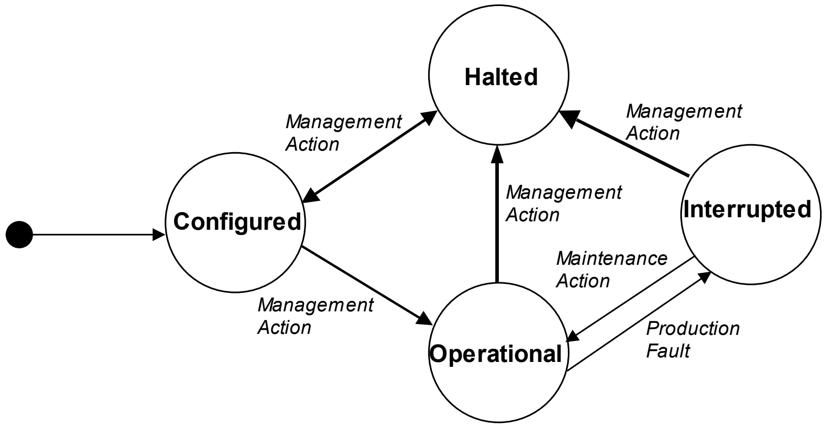

Figure 4-1: Production Status

SLE Reference

The concept of the production status is adopted from the SLE service specifications. In these specifications, different names are used for the values in the forward service specifications and in the return service specifications. For the CSTS Specification Framework, the terminology of the forward service specifications has been adopted.

4.4 SERVICE MANAGEMENT

All Cross Support Transfer Services are subject to Service Management, which facilitates the collection and exchange of all information required to agree to, schedule, prepare, and access

Cross Support Transfer Services. Cross Support Service Management is the subject of separate CCSDS Recommended Standards (references [6], [7], [20], and [23]).

For all CSTSes, Service Management determines the number and schedule of service instances to be provided, the resources required to enable those service instances, and the initial configuration of all service instances and their supporting resources.

The set of Cross Support Transfer Service instances, which may be of different service types, together with the specification of the characteristics of the production of those Cross Support Transfer Service instances are referred to as **service package.** A service package is provided by one ESLT to one or more Earth User CSSS as agreed between the Utilization Management (UM) function of the Earth User CSSS and the Provision Management (PM) function of the Provider CSSS.

Provision Management is the authority of a Provider CSSS that negotiates the provision of service packages with Utilization Management of an Earth User CSSS and controls and monitors the production and provision of the Cross Support Transfer Service instances by the Functional Resources belonging to the ESLT via ESLT Element Management (EM).

Utilization Management is the abstraction of the entities within a real Earth User CSSS that, on behalf of the cross-supported mission, interacts with the Provision Management of the Provider CSSS to arrange for the required Cross Support Services.

Configuration parameters to be agreed upon by Service Management may refer to Cross Support Service provision or to Cross Support Service production. The CSTS Specification Framework defines only a small set of Cross Support Service provision parameters that shall be agreed by Service Management, including but not limited to:

a) identification of the service provider and the service user;

b) identification of the port at which the service is made available;

c) the service instance provision period.

For some common operations and procedures, the CSTS Specification Framework identifies further configuration parameters that are known to the service user and the service provider but delegates the definition of how their values are determined to the service specification. The service specification may specify these values or delegate definition of the values to Service Management.

5 EXTENSIBILITY

Extensibility is a core concept of the CSTS Specification Framework. The CSTS Specification Framework (reference [1]) provides a set of reusable definitions of common procedures and operations for the specification of CSTSes. The Guidelines for the Specification of CSTSes (reference [2]) define general rules by which new procedures and operations can be derived from these definitions by extension and refinement of the common procedures and operations to match the specific requirements of a CSTS.

Derivation of operations and procedures from the building blocks defined in the CSTS Specification Framework encompasses two dimensions:

a) extension; and

b) refinement.

Operations may be extended in the following ways:

a) new parameters may be added to an operation invocation or response;

b) in some cases, new values may be added to a parameter in order to represent service-specific information (e.g., additional `diagnostic` values for negative responses and/or additional `event-name` and `event-value` parameter values for notifications);

c) for specific operations, parameters may be specified without associated syntax structure. The actual format and content may have to be specified when defining the service using the corresponding operation;

d) an unconfirmed operation may be converted to a confirmed operation by adding a return PDU or a two-phase (i.e., a confirmed operation) may be changed to a three-phase operation by adding an acknowledgement PDU (see section 6).

In principle, all operations defined in the CSTS Specification Framework can be extended. However, the specification of a procedure using or extending an operation might not permit further extension by procedures derived from it.

Operations may be refined by narrowing the value range of a parameter or by specializing the meaning of a parameter. For instance, the `generation-time` parameter of the TRANSFER-DATA operation in the Buffered Data Delivery procedure may be refined for a procedure dealing with transfer of telemetry by stating that the `generation-time` is the Earth Receive Time of the telemetry frame being transferred.

Procedures may be extended in the following ways:

a) extension of the operations used by a procedure as described above;

b) addition of operations defined in the CSTS Specification Framework or derived from operations defined in the CSTS Specification Framework.

Procedures may be refined, for example, by narrowing the specification of the semantic associated with the procedure or by adding more detail to the behavior description or the state table.

As an example, a Telemetry Delivery procedure could be derived from the Buffered Data Delivery procedure by extension and refinement of the operations START and TRANSFER-DATA and by refinement of the semantics associated with the data to be transferred and the parameters of the operations.

The rules for derivation of operations and procedures have been rigorously applied to the specification of the CSTS Specification Framework itself:

a) the common operations specified in the CSTS Specification Framework present the most abstract and general version;

b) several procedures defined within the CSTS Specification Framework derive new operations, which may be further derived by services using these procedures;

c) a procedure defined within the CSTS Specification Framework may be derived from another procedure also defined in the CSTS Specification Framework. For example, the Cyclic Report procedure is derived from the Unbuffered Data Delivery procedure.

6 OPERATIONS AND DATA TYPES

6.1 OVERVIEW

An operation is a task that the invoker requests the performer to execute and as such presents the basic interaction pattern between the service user and the service provider. The invocation of an operation may include parameters that specify further details of the task to be performed.

Operations are further classified by the number of interactions required to complete the operation:

a) An **unconfirmed operation** is invoked by the invoker and performed by the performer, but there is no report on the outcome of the operation.

b) For a **confirmed operation**, the performer provides a report (the '**response**') to the invoker on success or failure of the operation. Operation responses comprise an indication of success (positive result) or failure (negative result) and, in case of failure, a diagnostic that further specifies the reason for the failure. Confirmed operations are further classified as:

   1) **two-phase operations**, for which the performer returns a single report to the invoker, the **return**;

   2) **three-phase operations**, for which the performer provides an initial **acknowledgement** when receiving the invocation and subsequently issues the **return** when the operation has completed. This type of operation is typically used for operations where the task to be performed requires non-negligible time to complete. The initial acknowledgement confirms that the invocation has been received and understood and that the general preconditions for the operation are fulfilled.

Confirmed operations are further classified as 'blocking' or 'non-blocking' with respect to the procedure that uses the operation. When a blocking operation has been invoked, no further operations of the same procedure instance can be invoked before the return (two-phase operation) or acknowledgement (three-phase operation) to the invocation has been received. Non-blocking operations do not impose any constraints with respect to invocation of further operations.

NOTES

1 Sending of an invocation to a procedure instance when the response of a previously invoked blocking operation has not been sent yet by that procedure is a protocol error and will therefore result in the aborting of the association.

2 Given that all PDUs of a given service instance exchanged between service user and service provider share the same communications channel (reference [3]), backpressure used by one procedure for achieving the required flow control affects the communication of all procedure instances of the given service instance. If, for

example, a service user stops reading incoming TRANSFER-DATA invocations sent by the Buffered Data Delivery procedure, this also blocks the TRANSFER-DATA invocations issued by the Cyclic Report procedure of the same service instance.

3 Network congestion always affects the exchange of all PDUs regardless of the behavior of individual procedures.

SLE Reference

The SLE concept of operations has been fully adopted for the CSTS Specification Framework. Three-phase operations have been added to simplify handling of cases in which performance of an operation can take considerable time, and therefore an initial acknowledgement of the request is advisable. In the SLE specification, such cases make use of an additional ASYNC-NOTIFY operation to report the outcome of the operation. That ASYNC-NOTIFY operation must be linked to the original invocation by a special sequence number to report. The notion of 'blocking' and 'non-blocking' operations exists in SLE as well, but there refers to the complete service instance; that is, when a blocking operation has been invoked, the invoker may not invoke any other operation within the complete service instance until receiving the return of the blocking operation. The only blocking operations defined for SLE services are the BIND and the UNBIND operations.

The CSTS Specification Framework defines standard operation headers for unconfirmed and confirmed operations and a set of common operations for use within CSTSes. To be of general use, the operations of the CSTS Specification Framework are rather abstract and might not all be directly usable within a procedure used by a specific CSTS. However, all operations used in a procedure of a CSTS that conforms to the CSTS Specification Framework must be derived directly or indirectly from the common operations specified in the Recommended Standard.

Figure 6-1 depicts the 'operation layer' within the CSTS Specification Framework, and table 6-1 lists the common operations defined by the CSTS Specification Framework, providing more detail. A brief summary of each operation is provided by the following subsections. The parameters included in the standard operation headers are identified in tables 6-2 and 6-3, respectively.

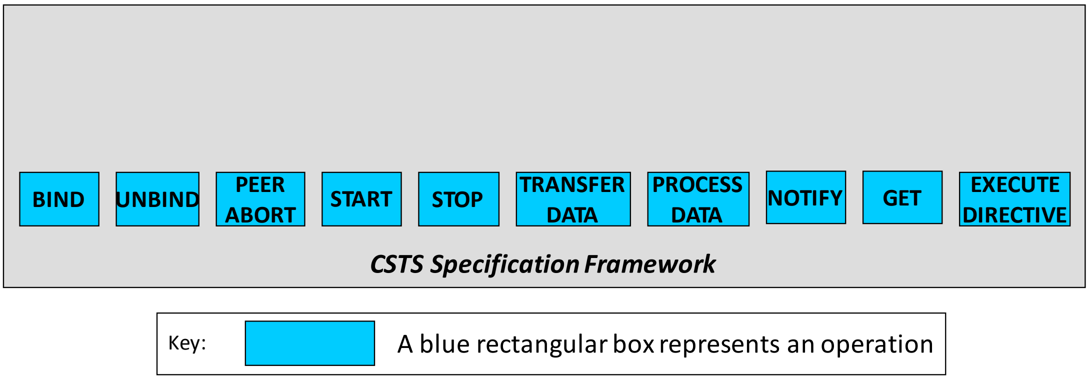

Figure 6-1: Common Operations Defined by the CSTS Specification Framework

Table 6-1: Common Operations Defined by the CSTS Specification Framework

Operation	Invoked By	Purpose	Confirmed	
			Ack.	Ret
BIND	service user	To establish an association with the peer	No	Yes
UNBIND	service user	To release an association previously established by a BIND operation	No	Yes
PEER-ABORT	service user or service provider	To notify the peer that the local application detected an error that requires the association to be terminated	No	No
START	service user	To request that the service provider start performing activities associated with the procedure using the operation	No	Yes
STOP	service user	To request that the service provider stop performing activities associated with the procedure using the operation	No	Yes
TRANSFER-DATA	service provider	To transfer a data unit to the service user	No	No
PROCESS-DATA	service user	To request that the service provider process the data contained in the invocation	No	No / Yes[2]
NOTIFY	service provider	To send a notification of an event to a service user	No	No
GET	service user	To ascertain the value of one or more service parameters	No	Yes
EXECUTE-DIRECTIVE	service user	To request the service provider to perform a predefined action.	Yes	Yes

Table 6-2: Standard Unconfirmed Operation Header Parameters

Parameter	Invocation
`invoker-credentials`	M
`invoke-id`	M
`procedure-instance-identifier`	M

[2] The Specification Framework specifies both the unconfirmed version of the PROCESS-DATA operation as well as the confirmed version that is derived from the unconfirmed version by means of extension.

Table 6-3: Standard Confirmed Operation Header Parameters

Parameters	Invocation	Acknowledgement	Return
`invoker-credentials`	M		
`performer-credentials`		M	M
`invoke-id`	M	M	M
`procedure-instance-identifier`	M		
`result`		M	M
`diagnostic`		C	C

Tables 6-2 and 6-3 also show how parameters of an operation are specified. An 'M' indicates that the parameter is mandatory for the invocation, acknowledgement, or return, whereas a 'C' indicates that the presence of the parameter is conditional. For instance, a parameter `diagnostic` with a 'C' in the Acknowledgement and Return column would only be included in an acknowledgement or return if the parameter `result` were 'negative'. An empty cell in table 6-3 indicates that this parameter is absent in the invocation, acknowledgement, or return. The meaning and purpose of the other parameters displayed in the tables will be explained by later sections in this report.

Each operation has a corresponding formal ASN.1 definition in an annex to the CSTS Specification Framework. Figure 6-2 shows, as an example, the ASN.1 specification of the standard invocation and return headers. Although the data structure is referred to as 'standard return header', it is actually used both for returns and acknowledgements.

This example also illustrates how extension of operations is handled within ASN.1. To make a data type extensible, an item of the type `Extended` is added to the type. The type `Extended` can be used to embed any externally specified data, which do not need to be specified in ASN.1 or be encoded using any of the ASN.1 encoding rules.

In the example, both the positive result and the negative result can be extended. In case of the negative result, the specification ensures that the `diagnostic` parameter is always included.

```
StandardInvocationHeader      ::= SEQUENCE
{     invokerCredentials      Credentials
,     invokeId                InvokeId
,     procedureInstanceId     ProcedureInstanceId
}

StandardReturnHeader          ::= SEQUENCE
{     performercredentials    Credentials
,     invokeId                InvokeId
,     result                  CHOICE
      {     positive    [0]   Extended
      ,     negative    [1]   SEQUENCE
                              {     diagnostic  Diagnostic
                              ,     extended    Extended
                              }
      }
}
```

Figure 6-2: ASN.1 Specification of Standard Invocation and Return Headers

6.2 BIND, UNBIND, AND PEER-ABORT

The operations BIND and UNBIND are used for orderly establishment and release of an association between the service user and the service provider. BIND and UNBIND can only be invoked by the service user.

The PEER-ABORT operation may be invoked by either side at any time to abruptly terminate an association in case of a problem.

SLE Reference

The operations for association establishment and release have been adopted from the SLE service specifications without modification except for the standard operation header, which adds the procedure instance identifier. Since this standard operation header is used for all operation invocations, the target procedure instance is always identified by an invocation. The service instance with which the service user wishes to interact only needs to be identified in the BIND invocation because all further interactions will take place via the association that is established by means of the BIND operation. Therefore, as in the SLE service specifications, the `service-instance-identifier` *is a parameter of the BIND invocation PDU, but not part of the standard operation header.*

6.3 START AND STOP

The operation START is invoked by the service user to request that the service provider start an activity that is associated with the procedure that defines the operation. Likewise, STOP is used to request termination of the activity. Both START and STOP can only be invoked by the service user. The START and STOP operations, if used, control the state of the procedure as explained in 7.1.

SLE Reference

In SLE services, START and STOP are used to request the start and stop of the main activity of a service, for example, transfer of telemetry for the SLE return services. In order to start other activities, for example, status reporting, SLE service specifications use special operations, for example, SCHEDULE-STATUS-REPORT. The CSTS Specification Framework applies a uniform approach to all procedures within a service; that is, transfer of telemetry and status reporting would each be controlled by START and STOP operations in which the target procedure that shall be started or stopped in the context of the given association is identified by the procedure instance identifier in the standard operation header of the START or STOP operation invocation.

6.4 TRANSFER-DATA AND PROCESS-DATA

TRANSFER-DATA is an unconfirmed operation used to transmit a data unit from the service provider to the service user. In the most abstract version, the operation invocation includes the time at which the data unit was generated and a sequence counter that can be used to check continuity of the data stream. The `data` parameter itself is of the special type AbstractChoice, which offers two choices: the type can either be used directly without extension if the `data` parameter shall be simply represented as an octet string, or it has to be specified by defining a suitable extension.

For data units that shall be transmitted from the service user to the service provider for processing by the provider, the CSTS Specification Framework defines the unconfirmed operation PROCESS-DATA, which is invoked by the service user. Derived procedures may refine the PROCESS-DATA operation to be confirmed. As is the case for TRANSFER-DATA, the operation invocation includes an unspecified `data` parameter of type `AbstractChoice` and a `data-unit-id` parameter.

SLE Reference

The TRANSFER-DATA operation corresponds to the TRANSFER-DATA operation of an SLE return service, whereas PROCESS-DATA corresponds to the TRANSFER-DATA operation in an SLE forward service.

6.5 NOTIFY

The unconfirmed operation NOTIFY is invoked by the service provider to notify the service user of the occurrence of an event of interest to the service user. Typically, events that are useful for understanding specific service provider behavior, such as an interruption in data delivery, are considered of interest and should be notified. The basic version of this operation includes the time when the event occurred, the identification (name) of the event, and an event value that may provide additional information related to the event.

The CSTS Specification Framework defines a small set of basic events reporting about changes of the production status (see 4.3) and/or about the progress in data processing.

SLE Reference

SLE service specifications include the operations SYNC-NOTIFY (return services) and ASYNC-NOTIFY (forward services) to specify whether or not notifications are synchronized with the main data flow, that is, the TRANSFER-DATA invocations. In the CSTS Specification Framework, this distinction has been dropped in the specification of the operations. Whether a notification is transmitted synchronously or asynchronously must be defined by the specification of the procedure using the NOTIFY operation.

6.6 GET

The confirmed operation GET is invoked by the service user to query the values of one or more parameters. The query is specified by a `list-of-parameters` parameter. This parameter contains one of the following: (a) a ‘null’ value (i.e., to be left empty), (b) a list of individual Parameter Names, (c) a list of individual Parameter Labels, (d) the name of a list, (e) a Functional Resource Name, (f) a Functional Resource Type, (g) a procedure instance identifier, or (h) a procedure type.

NOTES

1 Functional Resources are functions that are being performed to provide space communications services to the supported spaceflight mission. Functional Resources are *service-oriented.* A Functional Resource Type is a logical function or related set of functions that characterizes a service provider or production capability. A Functional Resource Type is identified by means of a Published Identifier. A Functional Resource Name is formed by the concatenation of a Functional Resource Type identifier and a Functional Resource Instance Number. A Functional Resource Instance Number is an integer index used to identify an instance of a Functional Resource Type.

2 A Parameter Name is formed by the concatenation of the Functional Resource Name of the resource the parameter is associated with and a Parameter Identifier that represents the individual parameter type. A Parameter Label is formed by the concatenation of a Functional Resource Type and a Parameter Identifier.

The CSTS Specification Framework does not specify how the Parameter Names and the parameter lists are defined and agreed on, but it delegates these specifications to service specifications based on the CSTS Specification Framework. Service specifications may further delegate this to Service Management when applicable.

The operation return includes, for each parameter, the following information:

a) the Parameter Name;

b) the parameter data type, selected from a set of supported types defined in the CSTS Specification Framework;

c) the parameter value; and

d) the parameter qualifier reporting the validity of its value as ‘valid’, ‘unavailable’, ‘undefined’ or ‘error’.

SLE Reference

The GET-PARAMETER operation used for SLE services only supports a query for a single parameter value. As the specification of a SLE GET-PARAMETER operation includes the specification of all ‘gettable’ parameters, the data types of the parameters are included in the definition of the GET operation. The CSTS Specification Framework generalizes the specification to refer to a set of standard data types and extends it to allow queries for multiple parameters in a more flexible manner.

6.7 EXECUTE DIRECTIVE

The operation EXECUTE-DIRECTIVE can be used to request execution of an activity that may take a non-negligible time to complete. It is the only three-phase operation defined in the CSTS Specification Framework. EXECUTE-DIRECTIVE is invoked by the service user, and the service provider provides intermediate feedback to the service user to acknowledge that the invocation has been received and is valid, as well as a final outcome of the operation that is returned some time following the acknowledgement.

In the CSTS Specification Framework, the Sequence-Controlled Data Processing procedure refines the EXECUTE-DIRECTIVE operation by adding the value ‘reset’ to the set of Object Identifiers possible for the `directive-identifier` parameter.

SLE Reference

EXECUTE-DIRECTIVE is conceived as an abstraction of the operations THROW-EVENT defined in the CLTU and FSP specifications and of EXECUTE-DIRECTIVE defined in the FSP specification to invoke a Command Operations Procedure (COP) Directive. In both cases, the operation return is an intermediate acknowledgment, whereas the operation result is reported with an ASYNC-NOTIFY invocation. This has been replaced by a three-phase operation.

The specific semantics of the EXECUTE-DIRECTIVE operation are specified by the Throw Event procedure in the CSTS Specification Framework (see 7.11).

7 PROCEDURES

7.1 OVERVIEW

Procedures define the protocol and the expected behavior for the invocation and performance of a set of operations to achieve a well-defined objective. A CSTS may be specified by assembling a given set of procedures to handle specific aspects of the service.

For example, a service that forwards telemetry received on the space link can be broken down into a number of procedures that handle association establishment and release, actual transfer of telemetry data units, status reporting, and querying of the service configuration parameters. All of these procedures except for telemetry data transfer can also be used for a service that supports reception and uplinking of telecommands, in which only a subset of the parameters in the reports differs.

The specification of a procedure in the CSTS Specification Framework includes:

a) a description of the objective of the procedure, including all assumptions that are made with respect to the type of data to be transferred and the Cross Support Service production process (descriptive);

b) a definition of the behavior of the procedure (prescriptive);

c) a definition of the operations that constitute the procedure (prescriptive);

d) a description of behavior expected of the service provider and the service user supported by a service provider side state matrix, where applicable (prescriptive).

The CSTS Specification Framework classifies procedures as **stateful** or **stateless**. A stateful procedure assumes the states ‘inactive’ and ‘active’ with transitions between these states triggered by well-defined operation invocations or responses. Typically, the transition from ‘inactive’ to ‘active’ is triggered by a START invocation, and the transition from ‘active’ to ‘inactive’ is triggered by a STOP invocation. A simplified state transition diagram for stateful procedures using START and STOP operations is shown in figure 7-1 below.

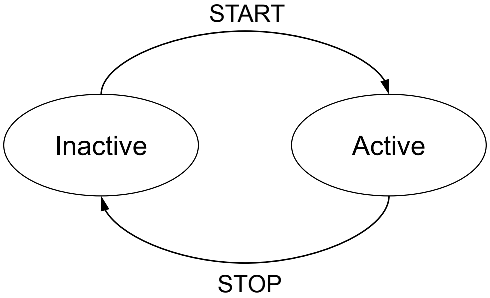

Figure 7-1: Procedure States

A CSTS will typically use more than one procedure in addition to the Association Control procedure and may require more than one procedure of the same type to be active at the same time, for example, to transfer different data streams concurrently. Therefore the CSTS Specification Framework distinguishes between the **procedure type** and the **procedure instance**. The procedure type corresponds to the specification of the procedure or the supporting program that implements the procedure specification. A procedure of a given type can be instantiated once or several times within a CSTS instance. Different instances of the procedure are distinguished by a 'procedure instance identifier' that is assigned by Service Management and included in all operation invocation headers.

The CSTS Specification Framework defines a set of common procedures that cover typical tasks of a CSTS. In order to be commonly usable, the procedure definition and in particular the purpose and semantics specification are reasonably abstract, such that the specifications as provided by the CSTS Specification Framework may sometimes not be directly usable. For those cases, the Guidelines (reference [2]) specify methods to derive more specific procedures from these common procedures.

The Association Control procedure is a special case. It is provided for establishment and release of the association between the service user and the service provider, and as such does not belong to any of the classes (stateful, stateless). As explained in more detail in section 8, the Association Control procedure must interface with all other procedure instances to achieve the desired effects.

Figure 7-2 depicts the 'procedure layer' within the CSTS Specification Framework and shows how the common procedures use the common operations. Table 7-1 lists the common procedures defined by the CSTS Specification Framework, providing more detail, and table 7-2 identifies the operations that are used by the procedures. A brief summary of each procedure is provided by the following subsections.

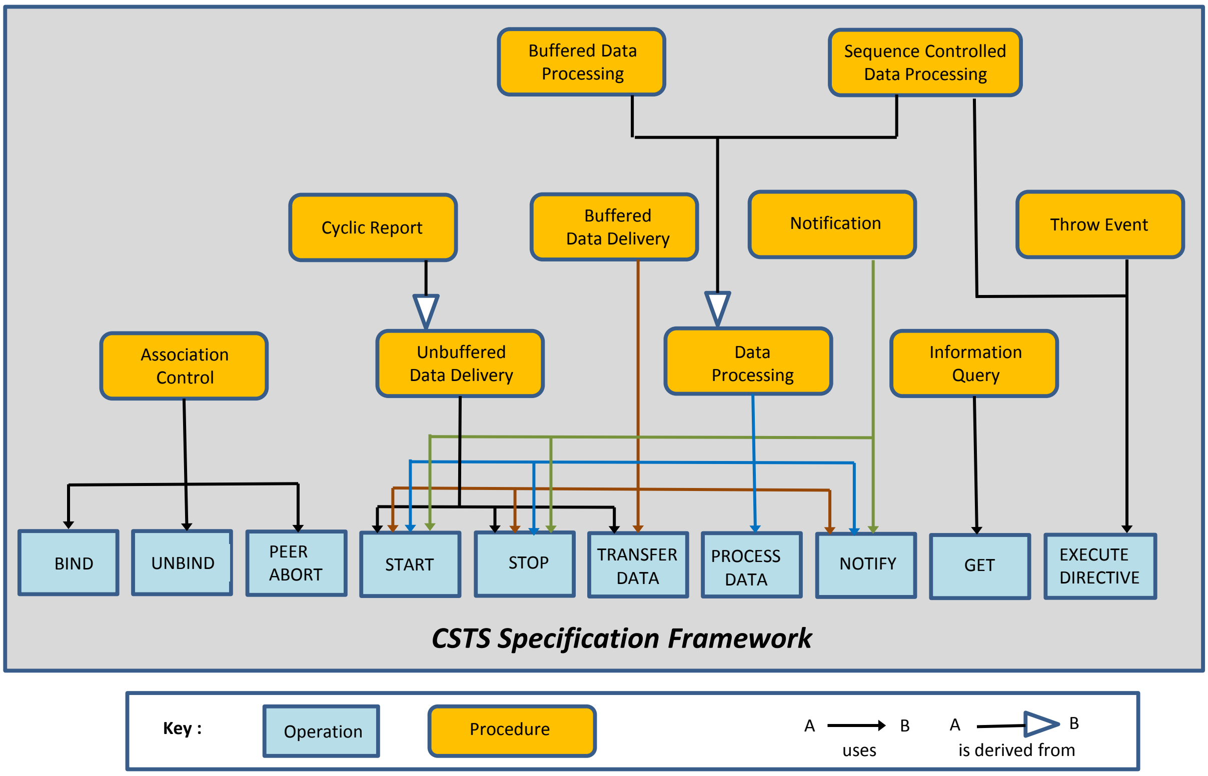

Figure 7-2: Common Operations and Procedures Defined by the CSTS Specification Framework

Table 7-1: Common Procedures Defined by the CSTS Specification Framework

Procedure	Purpose	Class
Association Control	Establishment and release of an association between a service user and a service provider.	N/A
Buffered Data Delivery	In-sequence transfer of bulk data, structured into data units, from the service provider to the service user. In real-time mode, low latency is given priority over data completeness. In complete mode, data completeness is given priority over low latency.	SF
Unbuffered Data Delivery	Best-effort transfer of bulk data, structured into data units, from the service provider to the service user in real-time delivery mode.	SF
Cyclic Report	Cyclic reporting of parameter values from the service provider to the service user.	SF
Information Query	Query by the service user of the value of one or more parameters controlling the service provider behavior.	SL
Notification	Notification of the service user by the service provider of events of interest to the service user.	SF
Data Processing	Processing of data units in the sequence as sent by the service user to the service provider and reporting of processing progress. This is an abstract procedure and is only foreseen for inheritance.	SF

Procedure	Purpose	Class
Sequence-Controlled Data Processing	Processing of data units strictly in the sequence as sent by the service user to the service provider, and enabling the service user to resynchronize transfer and processing in case of a problem detected during processing.	SF
Buffered Data Processing	Transfer of bulk data by the service user to the service provider, and reporting of processing progress.	SF
Throw Event	Signaling to the service provider that execution of a configuration change action is required, and report of the result of the action back to the service user.	SL

In table 7-1, 'SF' indicates stateful procedure while 'SL' flags a stateless procedure.

Table 7-2: Use of Operations by Common Procedures

	Association Control	Unbuffered Data Delivery	Buffered Data Delivery	Data Processing	Buffered Data Processing	Sequence-controlled Data Processing	Information Query	Cyclic Report	Notification	Throw Event
BIND	B									
UNBIND	B									
PEER-ABORT	NB									
START		B	B	B	B	B		B	B	
STOP		B	B	B	B	B		B	B	
TRANSFER-DATA		NB	NB					NB		
PROCESS-DATA				NB	NB	NB				
GET							NB			
NOTIFY			NB	NB	NB	NB			NB	
EXECUTE-DIRECTIVE						NB				NB

NOTE – 'B' and 'NB' indicate that the operation is 'Blocking' or 'Non-Blocking', respectively. An empty cell indicates that the procedure does not use the operation.

7.2 ASSOCIATION CONTROL

The Association Control procedure establishes and releases an association between a service user and a service provider for a given service instance.

The initiator, who within the scope of CSTSes is always the service user, initiates the association by sending a BIND invocation. On reception of the BIND invocation, the responder, who within the scope of CSTS is always the service provider, verifies that

a) the service user is known to the service provider and can be authenticated if use of authentication has been agreed on;

b) the service type and version identified by the BIND invocation is supported by the service provider;

c) the service instance to which the service user wants to bind is scheduled for the service user invoking the BIND operation; and

d) any service-specific conditions that might apply are fulfilled.

If all tests are passed, the service provider responds with a positive BIND return to indicate that the association is established. Otherwise, the service provider rejects the BIND invocation with a negative return unless authentication of the service user has failed. If the service user cannot be authenticated, the service provider aborts the association, revealing as little information about the reason as possible.

To orderly terminate the association, the service user invokes the UNBIND operation. On reception of the UNBIND invocation, the service provider confirms to the service user that the association will be released. In order to enable this confirmation, UNBIND is a confirmed operation. However, an UNBIND invocation received in a legal state is not rejected by the service provider. A service instance can be unbound by the service user only while the service provider is in the state ‘bound.ready’, that is, while the prime procedure is ‘inactive’. Service instances with a stateless prime procedure can be unbound at any time.

The association can be aborted by either service user or service provider using the PEER-ABORT operation to inform the peer system that the local system detected an error that requires the association to be terminated.

In order to ensure consistent behavior of all procedure instances within a service instance, the Association Control procedure must interact with all other procedure instances. This is further discussed in section 8, which deals with the role of service instances in the CSTS Specification Framework.

SLE Reference

On the level of a service instance, association control with the operations BIND, UNBIND, and PEER-ABORT is essentially the same as in the SLE service specifications. Differences are related to maintaining consistency between procedure instance states and the service instance states, which are subject to section 8.

7.3 UNBUFFERED DATA DELIVERY

This procedure can be used to accomplish the transfer of data from the service provider to the service user in a 'best effort' manner. Data are delivered as soon as they have been generated, if possible, and are discarded individually if the underlying communications service generates backpressure due to congestion of the connection between service provider and service user, or due to the service user's reading the data at a rate that is lower than the rate at which the service provider attempts to transfer them. Loss of data can be detected by the service user by inspection of the sequence number in the TRANSFER-DATA invocation used to transfer the data.

The operations defined in this procedure allow a service user to interact with a service provider to

a) request the start of the data transfer specifying the selection criteria of the data to be transferred (START operation);

b) receive the specified data units (TRANSFER-DATA operation); and

c) terminate (STOP operation) and optionally later re-start the delivery of data units applying the same or a different selection.

Table 7-3 identifies the operations used by the Unbuffered Data Delivery procedure. As this table shows, the procedure uses the common operations unmodified.

Table 7-3: Operations Used by the Unbuffered Data Delivery Procedure

Operations	Source	Extended	Refined
START	Common	No	No
STOP	Common	No	No
TRANSFER-DATA	Common	No	No

SLE Reference

There is no concept in the SLE service specifications that corresponds directly to the Unbuffered Data Delivery procedure. However, the Cyclic Report Procedure derived from the Unbuffered Data Delivery procedure (see 7.4) is an abstraction of status reporting as defined for SLE services.

7.4 BUFFERED DATA DELIVERY

The Buffered Data Delivery procedure supports transfer of bulk data, structured into data units, from the service provider to the service user. The operations used by this procedure allow a service user to interact with a service provider to

a) request start of the data transfer, specifying the criteria to apply when selecting the data to transfer (START operation);

b) receive the specified data units (TRANSFER-DATA operation);

c) receive, synchronized with transfer of data units, notifications of events that have a direct impact on the Cross Support Service production and delivery of data units (NOTIFY operation);

d) terminate (STOP operation) and optionally later re-start the delivery of data units applying the same or a different selection.

The Buffered Data Delivery procedure is suitable for transfer of data with the following characteristics:

a) data are structured into well-defined service data units;

b) data are generated by a service-provider process and may have to be transmitted as soon as possible after their generation.

The procedure supports scenarios in which the data generation rate exceeds the bandwidth of the communications link between the service provider and the service user using one of the following delivery modes:

a) **real-time**, in which the service provider transmits real-time data and ensures that a minimum number of the most recent consecutive data units is transferred with a specified maximum latency. If the capacity of the ground link does not allow transmitting all data, then data units are discarded and the service user is notified. Transmission of real-time data implies that data units are transmitted as soon as possible after their production by an ongoing Cross Support Service production session.

b) **complete,** in which the service provider transmits recorded data accepting high latency if the capacity of the ground link does not allow transmitting the data in a timely manner. The service provider ensures that no data are lost. Transmission of recorded data implies that transmitted data units are retrieved from an ongoing Cross Support Service production session or from terminated Cross Support Service production sessions.

The delivery mode to be applied for a given service instance must be agreed upon in advance by Service Management.

In order to support these delivery modes, data must be buffered. Conceptually, the procedure makes use of two different buffers, as illustrated in figure 7-3: the 'return buffer' and the 'recording buffer'.

In real-time delivery mode, the data is buffered within the return buffer. When the buffer has been filled or when a configurable latency timer expires, whichever comes first, the return buffer is transferred to the service user. In the case of a failure to transfer the data, the data in the return buffer is discarded, and a notification is inserted into the stream informing the service user that data has been discarded.

In complete delivery mode, a data buffer, called recording buffer, stores the data. The data is then passed to the return buffer. When the return buffer has been filled or when a configurable latency timer expires, whichever comes first, it is transferred to the service user. If transfer of the data is not possible, the procedure will retry until it succeeds. In the meantime, data is stored in the recording buffer.

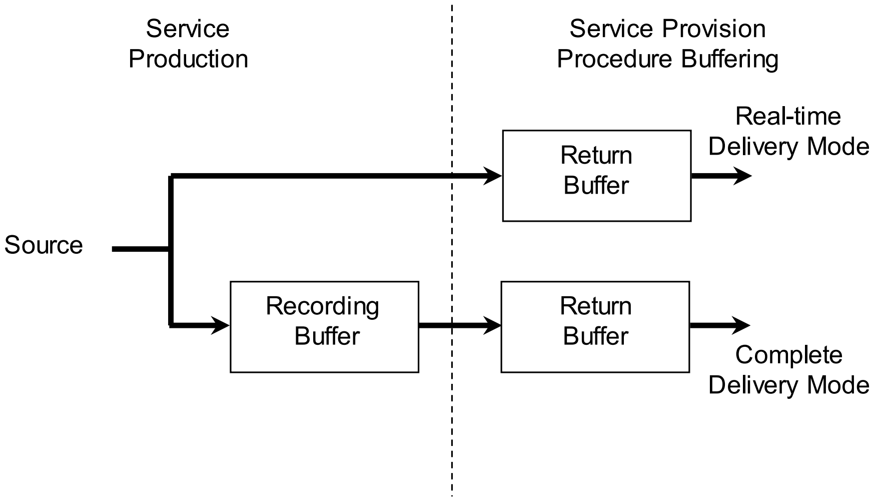

Figure 7-3: Buffers and Delivery Modes

Transfer of data units to the service user is complemented by notification of events that have an impact on the sequence of the transmitted data. Events are reported in a synchronous manner; that is, a notification is transmitted to the service user after the last data unit generated before the event occurred and before the first data unit generated after the event. This implies that notifications must be buffered in the same manner as data units.

With respect to handling of overflow conditions in the real-time delivery mode, the Buffered Data Delivery procedure further distinguishes between

a) discardable notifications, which are discarded in the same way as data units; and

b) non-discardable notifications, which are retained when a return buffer is discarded and for which the procedure makes sure that they are transferred to the service user as soon as possible.

Table 7-4 identifies the operations used by the Buffered Data Delivery procedure and whether the source is extended or refined.

Table 7-4: Operations Used by the Buffered Data Delivery Procedure

Operations	Source	Extended	Refined	Procedure Blocking/Non-Blocking
START	Common	Y	N	Blocking
STOP	Common	N	N	Blocking
TRANSFER-DATA	Common	N	N	Non-Blocking
NOTIFY	Common	Y	N	Non-Blocking

The START operation is extended to allow the selection of a generation time interval for which the data units shall be delivered. The NOTIFY operation is extended to add events that signal loss of data due to excessive backlog and 'end of data' when the service provider has no more data to transmit, when an overflow of the recording buffer occurs, or when the return buffer size or latency limit has been changed.

SLE Reference

The Buffered Data Delivery procedure is an abstraction of the telemetry transfer used in SLE return services and can be applied to any data generated by a process in the Provider CSSS. The main conceptual difference is that the 'complete online' delivery mode and the 'offline' delivery mode defined in the SLE service specifications have been combined into a single delivery mode, called the 'complete' delivery mode. As a consequence of this, the 'timely online' delivery mode has been renamed the 'real-time' delivery mode. Besides this, the special handling applied to the notification 'data discarded due to excessive backlog' has been generalized by the concept of 'discardable' and 'non-discardable' events.

7.5 DATA PROCESSING

The Data Processing procedure provides transfer of data units from the service user to the service provider for processing of the data units by the service provider. This procedure does neither define the data units being transferred nor the processing to be performed by the service provider. It is assumed, however, that processing is part of Cross Support Service production and will be specified by the Cross Support Service using this procedure.

This procedure is abstract and cannot be implemented directly because the specification is incomplete. To be implementable, derived procedures must provide the missing specifications.

The operations used by this procedure allow a service user to interact with a service provider to

a) request the service provider start accepting data units sent by the service user and processing them (START operation);

b) transfer to the service provider data units to be processed (PROCESS-DATA operation);

c) receive notifications of events that have an impact on data processing, of errors in processing of data units, and, optionally, of successful completion of the processing of individual data units (NOTIFY operation); and

d) terminate (STOP operation) and optionally later re-start the transfer and processing of data units.

The procedure enables monitoring of the completeness of the data processing performed by reporting the completion of the processing of a given data unit if such notification has been requested in the PROCESS-DATA invocation that has been used to transfer the data unit. Errors, or events that prevent processing to start or complete, are always reported. Table 7-5 identifies the operations used by the Data Processing procedure and whether the source is extended or refined.

Table 7-5: Operations Used by the Data Processing Procedure

Operations	Source	Extended	Refined
START	Common	No	No
STOP	Common	No	No
PROCESS-DATA	Common	Yes	No
NOTIFY	Common	Yes	No

The PROCESS-DATA is extended by the parameter by means of which the service user can select if the completion of the processing of the given data unit shall be reported or not. The NOTIFY operation is extended to add events for successful completion of data unit processing and for specific error conditions.

7.6 BUFFERED DATA PROCESSING

The purpose of the Buffered Data Processing procedure is to support transfer of large volumes of data at high data rates from the service user to the service provider for processing by the service provider in situations in which the maximum latency of data units may have to be limited. To that end, the procedure allows a service to select one of the following transfer modes:

a) complete transfer mode, in which data are transmitted to the service provider completely based on the flow control capabilities of the underlying data communications service, accepting any latency that might be implied;

b) timely transfer mode, in which the service provider ensures that data units will not be queued longer than a configurable latency limit before being processed and will discard the oldest unprocessed data units if either the capacity of the configurable Input Queue or the latency limit would be exceeded.

The Buffered Data Processing procedure extends the Data Processing procedure and in particular provides specifications for the handling of Input Queue overflow conditions that are left unspecified in the Data Processing procedure. Given that the Buffered Data Processing procedure is derived from the Data Processing procedure, the extensions of the PROCESS-DATA and NOTIFY operations specified for the Data Processing procedure are inherited by the Buffered Data Processing procedure and therefore not shown as extension in table 7-6.

Table 7-6: Buffered Data Processing Operations

Operations	Source	Extended	Refined
START	Data Processing	No	No
STOP	Data Processing	No	No
PROCESS-DATA	Data Processing	No	No
NOTIFY	Data Processing	Yes	No

NOTE – The NOTIFY operation is extended (see table 7-6) by a modified specification of the `event-value` parameter such that it can report the dynamically modifiable parameters of the Buffered Data Processing procedure as part of the 'data processing configuration change' event.

7.7 SEQUENCE-CONTROLLED DATA PROCESSING

The purpose of the Sequence-Controlled Data Processing procedure is

a) to provide strict sequential transfer and processing of data units in the sequence defined by the service user; and

b) to enable the service user to resynchronize transfer and processing of data units in case a problem is detected during processing of a data unit.

The procedure is intended for use in CSTSes that involve sequence-controlled data transfer and processing and that need confirmation of data transfer and status reports on the ongoing Cross Support Service production process.

The Sequence-Controlled Data Processing procedure extends the parent Data Processing procedure by the following features:

a) use of a confirmed PROCESS-DATA operation;

b) start of processing of the data units not earlier than an earliest and not later than a latest processing start time defined by the service user;

c) transition to a 'locked' state and notification of the service user in case processing of a data unit fails; in such cases, the service provider is blocked and does not accept any further PROCESS-DATA invocations, as the strict sequential processing can no longer be guaranteed;

d) resynchronization of data unit processing by the service user in case of a problem detected during data unit processing.

Table 7-7: Sequence Controlled Data Processing Operations

Operations	Source	Extended	Refined
START	Data Processing	Yes	No
STOP	Common	No	No
PROCESS-DATA	Data Processing	Yes	No
NOTIFY	Data Processing	Yes	No
EXECUTE-DIRECTIVE	Common	Yes	No

The operations are extended as shown in table 7-7.

The START operation is extended through the addition of the `first-data-unit-id` parameter to the invocation.

This procedure extends the PROCESS-DATA operation through changing the operation to be a confirmed operation and through the addition of the `earliest-data-process-start-time` and `latest-data-process-start-time` parameters and `diagnostic` values.

The NOTIFY operation is extended through the addition of the 'expired' and 'locked' events and through extension of the `data-processing-status` parameter with additional values. The `event-value` parameter is extended such that it can carry the procedure-specific values associated with the 'data processing configuration change' event.

The EXECUTE-DIRECTIVE operation is extended through the definition of one value possible for the `directive-identifier` parameter and the specification of the associated `directive-qualifier` parameter.

SLE Reference

The Sequence-Controlled Data Processing procedure is an abstraction of telecommand transfer and processing as defined for the SLE services CLTU and FSP. Conceptual differences include explicit modeling of the state 'locked', which is considered a sub-state of 'active', and the possibility to clear the Input Queue and restart processing with the directive 'reset'. The only way to recover from locking in the SLE services is to send STOP followed by START.

7.8 INFORMATION QUERY

The Information Query procedure enables the service user to request from the service provider the provision of a standard set of parameters reflecting

a) the configuration of the CSTS instance using this procedure or the values of other service provider parameters related to the provisioning of the service; and

b) the configuration and status of the Functional Resources involved in the Cross Support Service production associated with the Transfer Service instance.

In essence, this procedure is a wrapper around the GET operation described in 6.6, and this is also the only operation it uses without modification. Because the procedure only supports querying of parameters using the GET operation, it is classified as stateless.

The existence of this procedure allows a designer to add information for querying a service without further specifying the procedure to be used for this purpose.

SLE Reference

The Information Query procedure builds on the concept of 'gettable configuration parameters' in SLE service specifications. In the context of the CSTS Specification Framework, it has been generalized and specified as an independent procedure. Additional flexibility is provided through the enhanced features of the new GET operation.

7.9 CYCLIC REPORT

The Cyclic Report procedure extends the Unbuffered Data Delivery procedure with the following capabilities:

a) the Cyclic Report procedure defines the structure of the data unit as a set of parameters and supports selection of the parameters that shall be delivered in the START operation;

b) the selected parameters are delivered periodically.

The service user requests periodic reporting by invoking the START operation and specifying the following criteria:

a) the delivery cycle to use for periodic delivery;

b) the parameters to be delivered, in the same manner as for the GET operation, in one of the following ways:

   1) a ‘null’ value (corresponds to the default list);

   2) a list of individual Parameter Names;

   3) a list of individual Parameter Labels;

   4) the name of a list of Parameter Labels;

   5) the name of a Functional Resource Instance;

   6) a Functional Resource Type;

   7) a procedure type; or

   8) a procedure instance identifier.

NOTE – Most procedures can be configured in accordance with the needs of a given service type and instance. Most of these configuration parameters will have constant values throughout the lifetime of the associated service instance, but some can be modified while the service instance is bound. By using options 7) and 8) on the list above, a service user can retrieve the procedure configuration parameters.

Annex E of the CSTS Specification Framework specifies how the Parameter Names and the parameter lists are defined.

Table 7-8 identifies the operations used by the Cyclic Report procedure and whether the source is extended or refined.

Table 7-8: Operations Used by the Cyclic Report Procedure

Operations	Source	Extended	Refined
START	Unbuffered Data Delivery	Yes	No
STOP	Unbuffered Data Delivery	No	No
TRANSFER-DATA	Unbuffered Data Delivery	No	Yes

The START operation is extended by the parameters used to specify the parameter selection and the delivery cycle. The specification of the `data` parameter in the TRANSFER-DATA operation is refined by specification of the data format as a sequence of parameters, including for each parameter:

a) the Functional Resource Instance or procedure instance the parameter is associated with and the Published Identifier of the parameter;

b) the parameter type, selected from a set of supported types defined in the CSTS Specification Framework, and the parameter value, provided the parameter qualifier is ‘valid’ (see c), below);

c) the parameter qualifier, reporting the validity of its value as ‘valid’, ‘unavailable’, ‘undefined’, or ‘error’.

SLE Reference

As mentioned earlier in 7.3, the Cyclic Report procedure is an abstraction of status reporting in SLE services. In contrast to SLE status reporting, it is, however, permissible to discard data in case of link congestion, which is considered acceptable because the data are delivered periodically. Instead of using a special operation to start and stop status reporting as in SLE services, the cyclic report is started and terminated by the START and STOP operation, just as any other activity.

As the specification of the SLE STATUS-REPORT operation includes the specification of all status parameters, the data types of the parameters are included in the definition of that operation. The CSTS Specification Framework generalizes the specification to refer to a set of standard data types and extends it to allow selection of the parameters to be reported in a more flexible manner.

7.10 NOTIFICATION

The Notification procedure provides means by which a service user is able to select from a set of pre-identified events and subsequently receive notifications of the occurrence of any of those selected events while the Notification procedure is active.

The operations defined in this procedure allow a service user to interact with a service provider to

a) request the start of notification of a set of pre-identified notifiable events (START operation);

b) receive notifications of the occurrence of the selected events (NOTIFY operation); and

c) terminate (STOP operation) and optionally later re-start the delivery of event notifications, applying the same or a different selection of notifiable events.

When starting the Notification procedure with the START operation, the service user subscribes to the particular events that are to be reported by that procedure in one of the following ways:

a) a ‘null’ value (corresponds to the default list);

b) a list of individual Event Names;

c) a list of individual Event Labels;

d) the name of a list of Event Labels;

e) the name of a Functional Resource Instance;

f) a Functional Resource Type;

g) a procedure instance identifier; or

h) a procedure type.

Annex E of the CSTS Specification Framework specifies how the Event Names and the event lists are defined.

Events may be notified by other procedures as well, but these will typically be events that are specific for that procedure. Examples include notification of data loss by the Buffered Data Delivery procedure (see 7.4) or processing completion reports in the Data Processing procedure (see 7.5). The Notification procedure can be used in a CSTS in combination with other event notifying procedures if the events that are reported by each procedure are clearly specified and overlap is avoided.

Table 7-9 identifies the operations used by the Notification procedure and whether the source is extended or refined. The START operation is extended by the parameters used to specify the event selection.

Table 7-9: Operations Used by the Notification Procedure

Operations	Source	Extended	Refined
START	Common	Yes	No
STOP	Common	No	No
NOTIFY	Common	No	No

SLE Reference

The Notification procedure builds on the concept of notifications in SLE service specifications. In the context of the CSTS Specification Framework, it has been generalized and specified as an independent procedure. Additional flexibility is provided as the service user can specify the set of events that should be reported.

7.11 THROW EVENT

The Throw Event procedure provides the capability for a service user to request the service provider to initiate predefined actions to be performed by the Provider CSSS, typically by EM of an ESLT, to receive acknowledgements of successful receipt of the requests, and to receive reports of the final outcomes of the actions back from the service provider. The service provider checks the conditions under which it can properly and safely execute the actions and executes the actions if these tests are passed.

The Throw Event procedure is intended for use in Cross Support Transfer Services that involve the modification of operating parameters of a Provider CSSS, typically an ESLT, during the execution of a service package. The following types of applications are envisaged:

a) the procedure may be used to modify Cross Support Service production parameters associated with the data being transferred by that CSTS. For example, a service whose primary purpose is to deliver commands to the space element may use the Throw Event procedure to request modification of the uplink parameters used to radiate those commands;

b) the procedure may be used to modify Cross Support Service production parameters of the Provider CSSS independent of any data transfer via that CSTS. For example, a service whose purpose is to control parameters of Cross Support Service production functions that support multiple CSTS instances, and for which no single service instance has the authority to modify those parameters, may use the Throw Event procedure to modify these parameters on behalf of all of the service users.

The CSTS Specification Framework does not specify any specific events, actions, or guard conditions but delegates these definitions to the specification of the service using this procedure. The service specification may further delegate these specifications to bilateral agreements, when this is considered appropriate.

This procedure uses only one operation, the EXECUTE-DIRECTIVE, without any extension or refinement, and is consequently stateless.

SLE Reference

The Throw Event procedure is equivalent to the THROW-EVENT operation defined in the SLE service specifications for F-CLTU and FSP. Within the CSTS Specification Framework, it is defined as a procedure using the EXECUTE-DIRECTIVE operation in order to capture the specific behavior associated with the SLE THROW-EVENT operation.

8 SERVICES

The CSTS Specification Framework defines building blocks for the construction of services, not the services themselves. The CSTS Specification Framework therefore has a reasonably high level and abstract notion of 'service' that can be characterized by the following statements:

a) a Cross Support Transfer Service is composed of a set of procedures;

b) the actual provisioning of a transfer service by a given service provider for a specific service user is referred to as service instance;

c) the service instance is subject to scheduling and configuration, and it is the service instance that must be bound; that is, the association between service user and service provider needs to be established before any further interaction can take place;

d) the service instance can be seen as a container for the instantiations of the procedures defined in the service specification.

Because binding of a service instance is a prerequisite for interactions between the service user and the service provider, one of the procedure instances in a service instance must be an instance of the Association Control procedure, which is responsible for binding and unbinding and as such controls the service instance state.

From the perspective of service specification and implementation, it is convenient to view a service instance as a collection of procedure instances, each of which having its own state. However, from an operations perspective, a service has a specific purpose, for example, to transfer telemetry from an ESLT to an Earth User CSSS. From this viewpoint, it seems natural to say that the service instance is 'active' if telemetry transfer has been started, and that it is 'ready' if the service instance has been successfully bound but telemetry transfer has not yet been started, even if status reporting has already been switched on.

This operations viewpoint is acknowledged by the CSTS Specification Framework by saying that each service must identify a single procedure instance as the 'prime procedure', and that this prime procedure determines the service instance state. It is permissible, however, that the type of procedure that serves as prime procedure is chosen by configuration of the service instance. If the prime procedure is stateful, the service instance state 'bound' is structured into two sub-states: 'bound.ready', in which the prime procedure is in the state 'inactive'; and 'bound.active', in which the prime procedure is in the state 'active'. Consequently, an UNBIND invocation is allowed only in the state 'bound.ready', that is, when the prime procedure is not active.

Other procedure instances in the same service instance have no influence on the service instance state. This implies that the UNBIND operation can be invoked in any state of the procedure instance and therefore has the same effect as PEER-ABORT for a non-prime procedure instance.

Figure 8-1 illustrates this concept.

NOTE – The diagrams in figures 8-1 and 8-2 do not present state diagrams in a formal sense, as they present different state machines in a synoptic view.

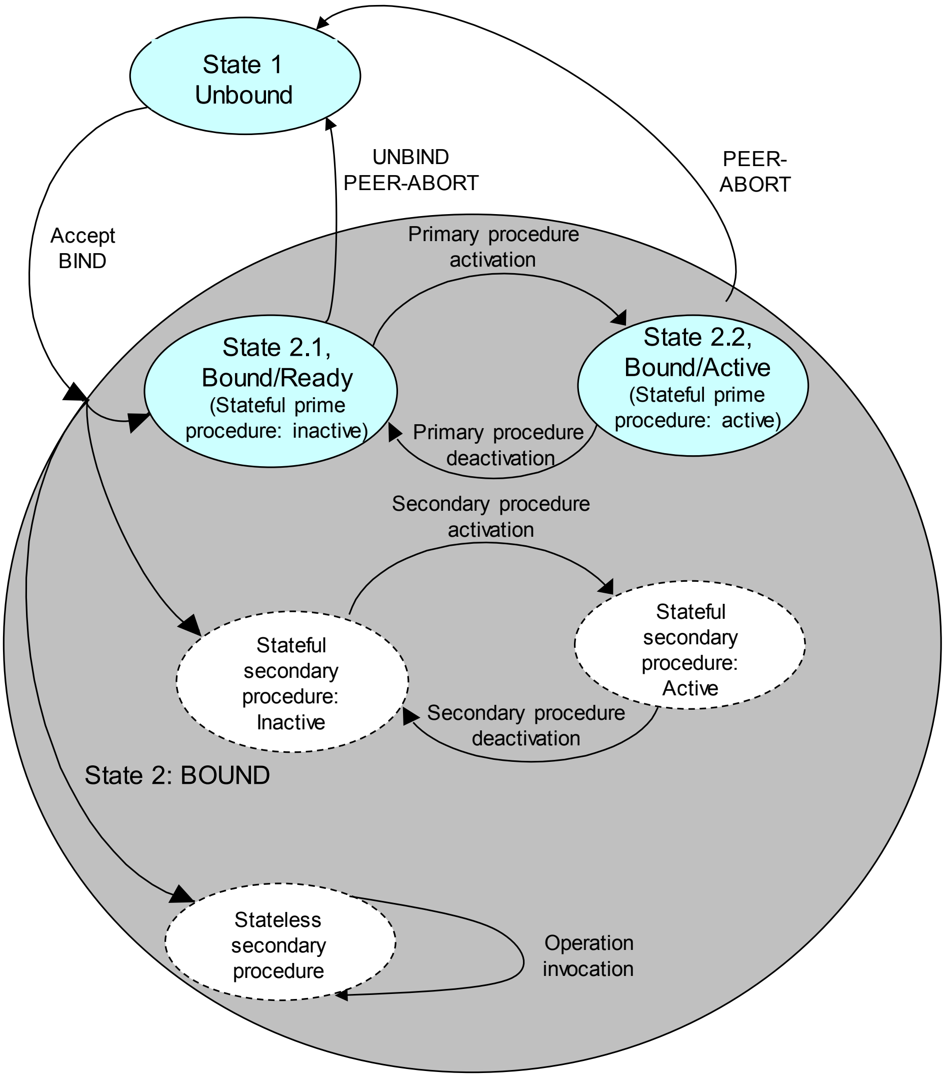

Figure 8-1: Service and Procedure States (Stateful Prime Procedure)

A Cross Support Transfer Service specification can identify any procedure instance as the prime procedure, which can be an instance of a stateless procedure. In this case, the prime procedure has no state, and consequently, the service instance state ‘bound’ has no sub-states, as indicated in figure 8-2.

The concept of a prime procedure is quite natural in most of the cases, but sometimes selection of the prime procedure is not obvious. For example, the Monitored Data CSTS includes a set of Cyclic Report procedure instances that can be active concurrently and monitor different sets of parameters. On the level of the service specification, there is no criterion for selection of the prime procedure, and therefore the specification delegates this selection to Service Management.

The association between the service user and the service provider is handled by the Association Control procedure, but the effects of association management have an impact on other procedure instances. In order to control the procedure state according to the specified rules, the Association Control procedure also needs to know the state of the designated prime procedure instance.

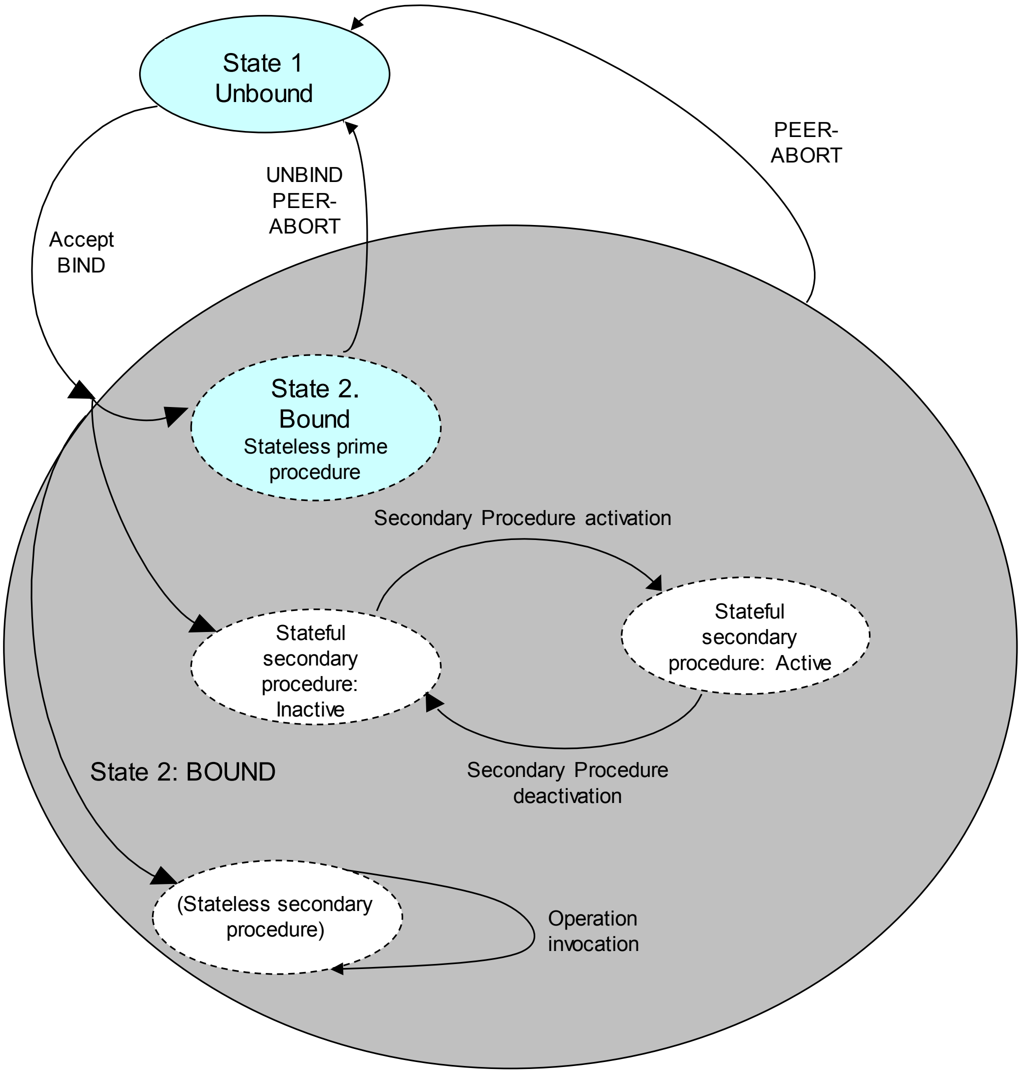

Figure 8-2: Service and Procedure States (Stateless Prime Procedure)

The CSTS Specification Framework considers necessary interactions between procedure instances a local implementation matter. However, for the purpose of specifying state tables, it models the cooperation of procedure instances as follows:

a) only the Association Control procedure handles the operations BIND, UNBIND, and PEER-ABORT;

b) when receiving a PEER-ABORT invocation or accepting an UNBIND invocation, the Association Control procedure generates a local ‘terminate’ event on which all procedure instances shall close down;

c) when a procedure needs to abort, it requests invocation of PEER-ABORT from the Association Control procedure, which subsequently generates the local ‘terminate’ event to signal the end of the association to all procedure instances.

9 DATA COMMUNICATIONS SERVICES

It is the specific intent of the CSTS Specification Framework to define building blocks for a CSTS independently of any particular communications services, protocols, technologies, or formatting of the data content. However, the CSTS Specification Framework provides a basis for the specification and development of a Cross Support Transfer Service that is intended to be used for development of real systems that implement the service, and this obviously requires the availability of a communications service to convey invocations and responses of the CSTS operations between the service user and the service provider.

The only requirement that the CSTS Specification Framework imposes on such a communications service is that it provides a reliable protocol, that is, to ensure that invocations and responses of operations are transferred:

a) in sequence;

b) completely and with integrity;

c) without duplication;

d) with flow control that notifies the application layer in the event of backpressure; and

e) with notification to the application layer in the event that communication between the service user and the service provider is disrupted, possibly resulting in a loss of data.

The communications service specified in reference [3] meets all of these requirements and, as pointed out in the CSTS Specification Framework, is assumed to be used by default for CSTSes.

In order to specify interactions between the service user and the service provider in a technology independent manner, the CSTS Specification Framework uses an abstract data communications model, sketched in figure 9-1 and briefly described below.

The CSTS interface between the service user and the service provider is specified in terms of the operations that the service implements. These operations are realized by mapping the service operation invocations, acknowledgements, and returns to protocol data units that can be conveyed by means of the underlying communications service.

Typically, one CSTS PDU corresponds to the invocation, acknowledgement, or return of an operation. From the point of view of the service provider or service user application, the interaction between the service user and service provider is in terms of operations, but from the point of view of the application entities that implement the CSTS protocol, it is in terms of the exchange of CSTS PDUs.

The mapping of CSTS PDUs to an underlying communications service is intentionally outside the scope of the CSTS Specification Framework. A suitable mapping is defined in reference [3].

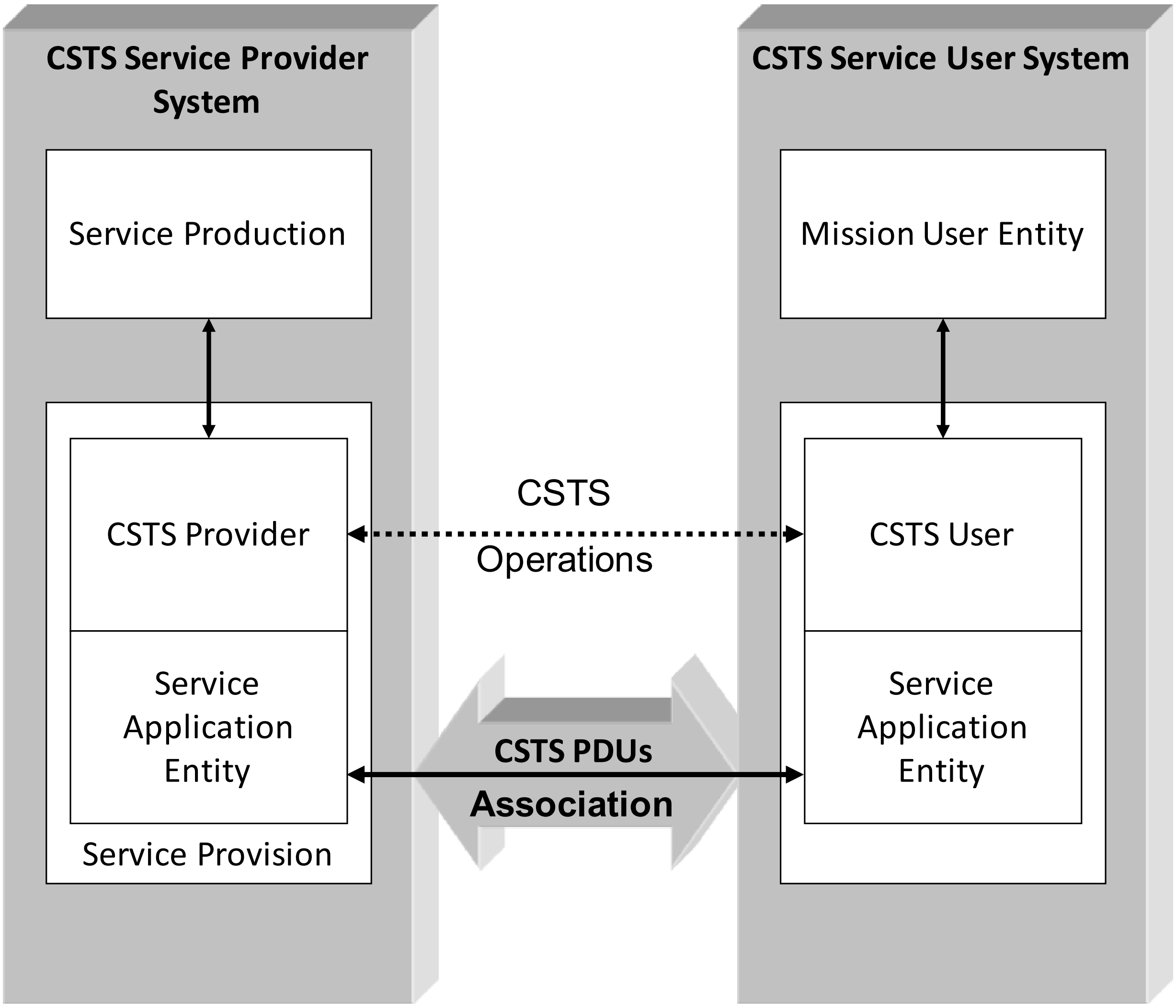

Figure 9-1: Communication between Service User and Service Provider

SLE Reference

The general approach in specifying data communications services, including the requirements for a reliable service and the abstract communications model, has been fully adopted from the SLE service specifications.

10 HOW TO SPECIFY A CROSS SUPPORT TRANSFER SERVICE

10.1 INTRODUCTION

The second normative component of the CSTS Specification Framework, the Guidelines for Specification of CSTSes (reference [2]), defines the requirements and constraints that govern the way in which a legal CSTS can be developed in accordance with the CSTS Specification Framework and provides guidance on the structure and content of CSTS specifications.

Any CSTS specification should follow the Guidelines for Specification of CSTSes (reference [2]) to the greatest extent possible and practical, but if a service has circumstances that are not addressed by that document, the service specification can go beyond these guidelines as long as those extensions are explicitly identified in the service specification.

The Guidelines for Specification of CSTSes support four major approaches for developing CSTSes from the common procedures defined in the CSTS Specification Framework:

a) direct adoption of procedures defined in the CSTS Specification Framework;

b) derivation of new procedures from those defined in the CSTS Specification Framework;

c) specification of new procedures using the operations defined in the CSTS Specification Framework;

d) derivation from an existing Cross Support Transfer Service specification.

A CSTS specification may apply any or all of these approaches in any combination.

10.2 RULES FOR COMPOSITION OF A CSTS

The most important rules that must be applied for all service specifications include the following:

a) Every service must include one and only one instance of the Association Control procedure, and the Guidelines for Specification of CSTSes strongly recommend that the Association Control procedure be adopted unmodified.

b) For every procedure type except the Association Control procedure, the specification may define that a service instance includes exactly one instance or that a service instance may include multiple instances. In the latter case, the specification must define the minimum number and may also fix the maximum number. Alternatively, a specification may delegate definition of the maximum number to Service Management, but at the time a service instance is bound, the number of procedure instances must have been fixed.

c) A service specification may declare a procedure type to be optional, which is equivalent to setting the minimum number of procedure instances to zero. A service provider is not required to implement an optional procedure type, but it must at least be able to identify invocation of operations of the unsupported procedure and reject them using the appropriate `diagnostic` value in the response. A service user and a service provider may also agree not to use an optional procedure by setting the maximum number to zero via Service Management if that option is provided by the service specification.

d) The service must identify one procedure instance as the prime procedure instance that controls the service instance state (see section 8). A service specification may permit more than one procedure type to be selected as the prime procedure type; however, at the time the service instance is created, the type of the prime procedure must have been selected by configuration. Obviously, the procedure type of the prime procedure instance must not be optional. However, a service specification may allow multiple instances of the procedure and define the rules by which the prime procedure instance is determined.

e) The specification of an implementable service must ensure that all definitions delegated to the service by the CSTS Specification Framework are fixed. This applies in particular to all data types. For other specifications, the service specification may further delegate the final definition to Service Management, but must include a complete specification of requirements on Service Management. As mentioned earlier in section 3, a CSTS may be declared abstract, and in this case, the specification may delegate any specification to services derived from it.

10.3 COMPOSING A SERVICE ONLY FROM COMMON PROCEDURES

The first, and simplest, approach is to develop a service through direct adoption of one or more common procedures defined in the CSTS Specification Framework without modification. Figure 10-1 depicts a new hypothetical service (‘Service 1’) that is composed of the unmodified CSTS Specification Framework procedures Association Control, Cyclic Report, and Sequence-Controlled Data Processing.

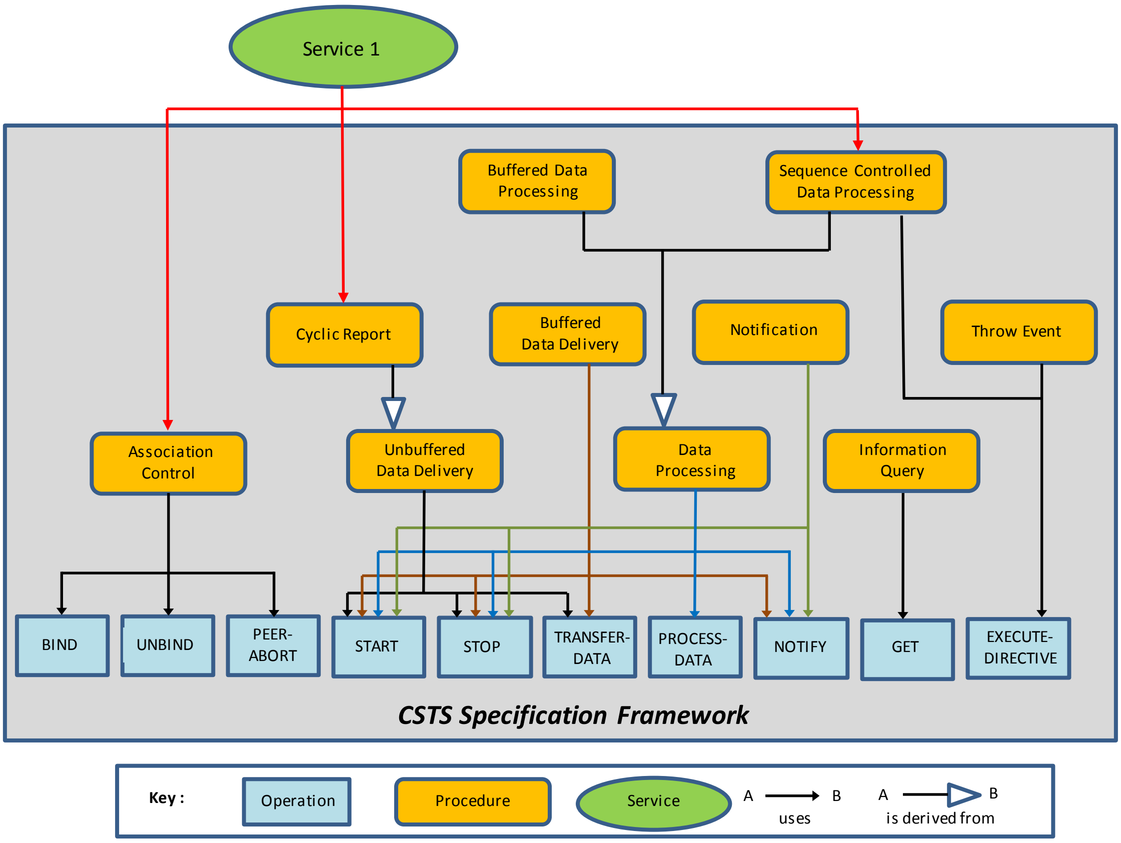

Figure 10-1: CSTS Directly Composed of Common Procedures

The composition of the CSTS in this case is essentially confined to the identification of the component common procedures. This is the recommended approach if it can be made to work in satisfaction of the requirements of the application addressed by the CSTS.

10.4 DERIVING NEW PROCEDURES FROM THE CSTS SPECIFICATION FRAMEWORK

The second approach is to derive one or more procedures from existing procedures, in order to add to or modify the capabilities of those procedures. As explained earlier in section 5, derivation of a procedure may occur through *refining* or *extending* the procedure (or both). Figure 10-2 depicts a new service (‘Service 2’) that is composed of the common Association Control, Cyclic Report, and Sequence Controlled Data Processing procedures defined in the CSTS Specification Framework plus a ‘Configuration’ procedure that is derived from the Throw Event procedure defined in the CSTS Specification Framework. The Configuration procedure might specify how the service user can request change of the Cross Support Service production engine configuration and for that purpose specify specific Directive Identifiers and directive qualifiers that are left unspecified in the Throw Event procedure specification.

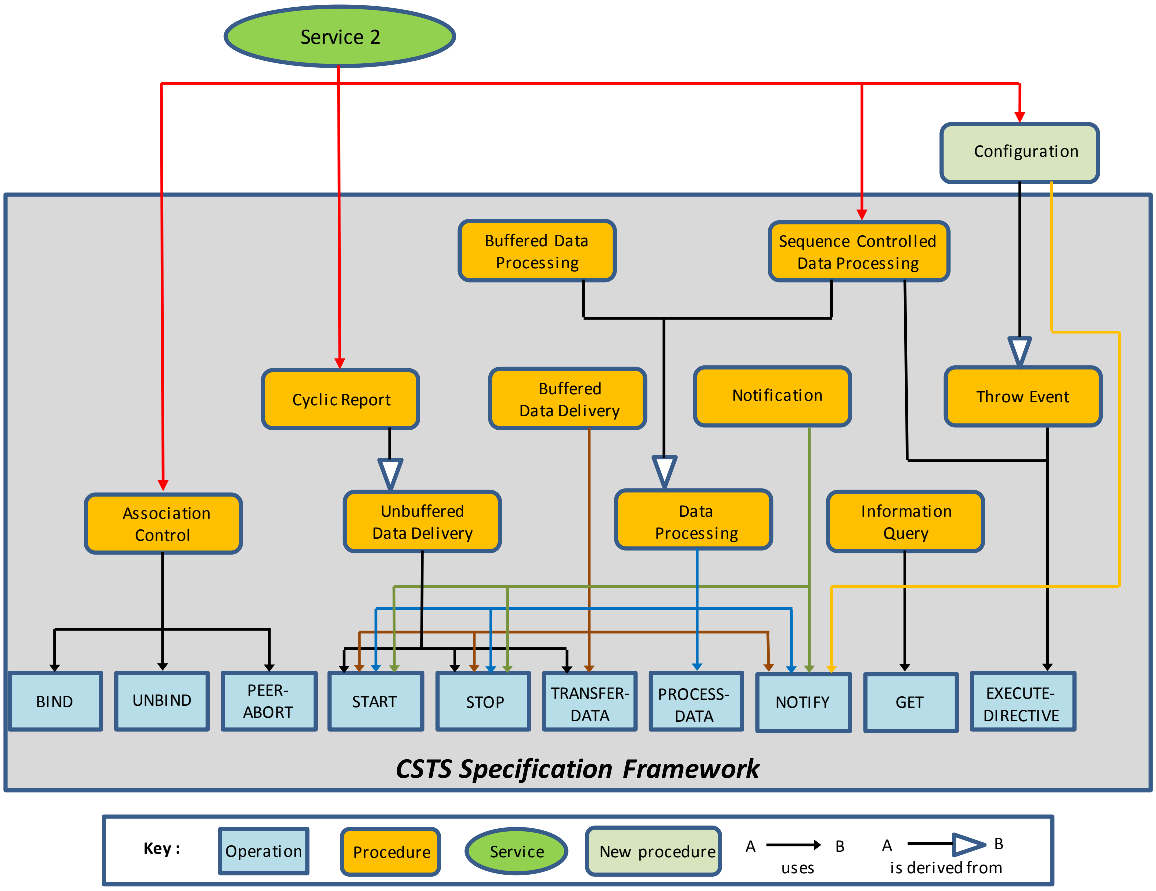

Figure 10-2: CSTS Composed of Common and Derived Procedures

The specification of a CSTS that includes one or more derived procedures requires more information to be provided for each of the derived procedures. Specifically, for each derived procedure, the CSTS specification must (a) define how the behavior of the procedure differs from its parent procedure, (b) identify which of the operations of the procedure are extended, refined, or new to the procedure, and (c) define the procedure state table.

The CSTS Specification Framework makes use of such derivation in that, for example, the Sequence-Controlled Data Processing and Buffered Data Processing procedures are derived from the Data Processing procedure. Likewise, the Cyclic Report procedure is derived from the Unbuffered Data Delivery procedure. A real example of such a service being specified using such an approach is the Monitored Data CSTS (reference [9]), for which annex B summarizes the specification.

10.5 DEFINING NEW SERVICE SPECIFIC PROCEDURES

The third approach to creating a new CSTS is to create one or more service-specific procedures from the common operations defined in the CSTS Specification Framework. These operations can be extended or refined to meet the needs of the service. Figure 10-3 depicts a new service ('Service 3') that is composed of the common Association Control, Cyclic Report, and Sequence Controlled Data Processing procedures, plus a service-specific procedure 'Notify & Get' that is composed of the START, STOP, GET, and NOTIFY operations.

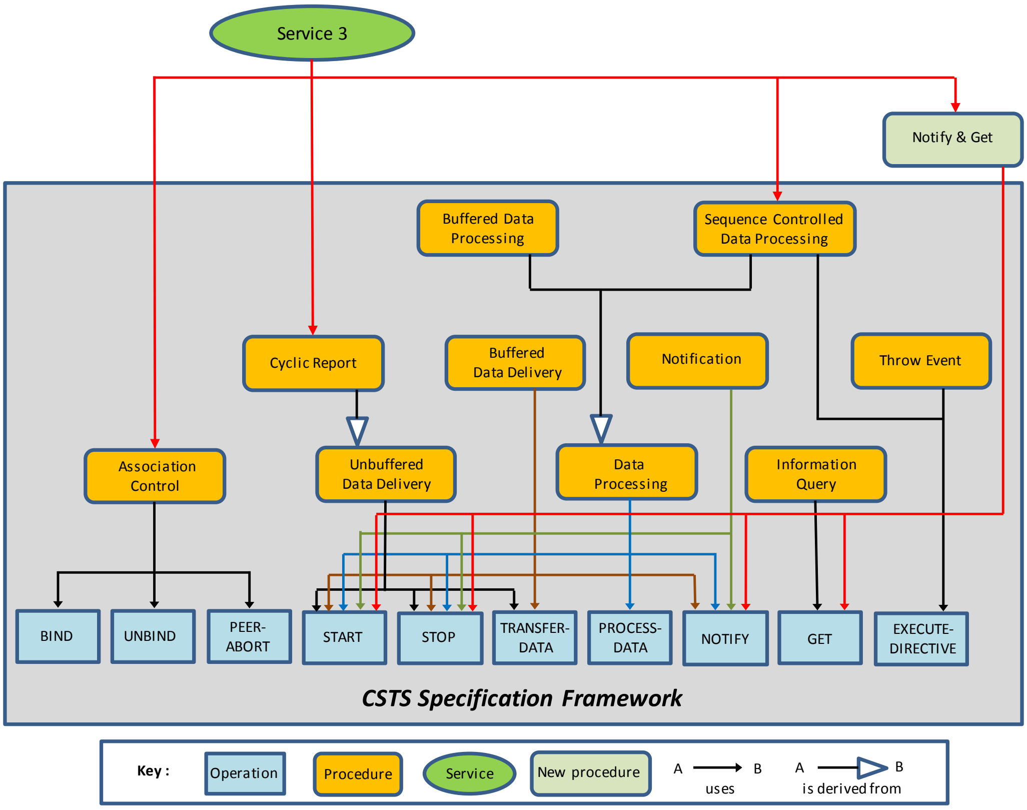

Figure 10-3: CSTS Composed of Common and Service Specific Procedures

The specification of a CSTS that includes one or more service-specific procedures requires that for each new service specific procedure, the CSTS specification must (a) define the complete behavior of the procedure, (b) identify which of the operations used by the procedure are extended and/or refined, and (c) define the procedure state table.

This approach is not recommended by the Guidelines for Specification of CSTSes if it is possible to derive the procedure from one of the common procedures defined in the CSTS Specification Framework. In the example, it might have been possible to extend the common Notification procedure by adding the GET operation, or to use the Notification and the Information Query procedures without modifications, which might simplify the service specification and increase the chance of reusing existing software components for its implementation.

10.6 DERIVING FROM AN EXISTING CSTS

The fourth approach to creating a new CSTS is to derive it from an existing CSTS. A derived CSTS can be based on a *concrete* CSTS, that is, a CSTS that is fully capable of performing a useful service on its own, or on an *abstract* CSTS, that is, a service that does not perform a useful service on its own, but rather is intended to serve as a common basis for a family of services that could be derived from it. Figure 10-4 depicts a new service ('Service 4') that is derived from an abstract 'Data Processing' service, which itself is composed of the common Association Control, Cyclic Report, and Sequence Controlled Data Processing procedures defined in the CSTS Specification Framework and a Configuration procedure (see 10.4) derived from the common Throw Event procedure.

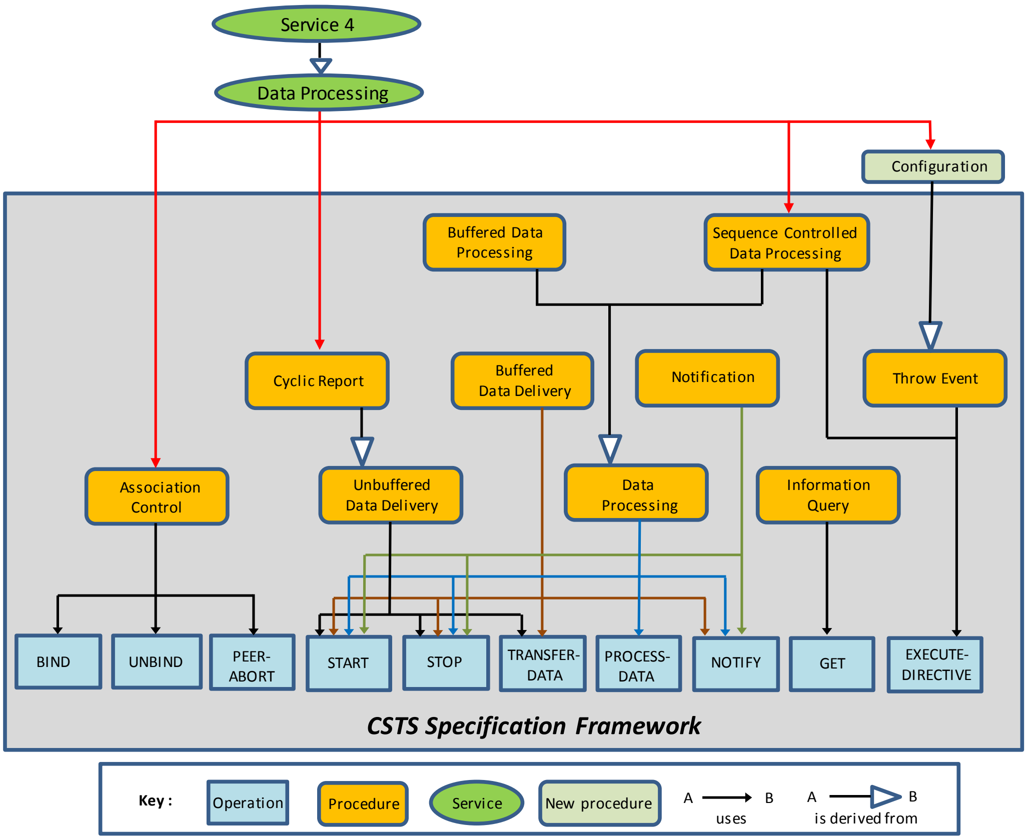

Figure 10-4: CSTS Derived from an Existing CSTS

The specification of a CSTS that is derived from an existing CSTS can potentially involve all of the information in the other three methods of CSTS creation: (a) direct inheritance of standard procedures, (b) derivation of any of the procedures inherited from the base CSTS or included from the CSTS Specification Framework, and (c) addition of service-specific procedures.

ANNEX A

ACRONYMS AND ABBREVIATIONS

Term	Meaning
ASN.1	Abstract Syntax Notation One
B	Blocking (CSTS SFW operation)
BDD	Buffered Data Delivery (CSTS SFW procedure)
CLTU	Communications Link Transmission Unit
COP	Command Operations Procedure
CSSS	Cross Support Service System
CSTS	Cross Support Transfer Service
CSTS SFW	CSTS Specification Framework
EM	Element Management
ESLT	Earth Space Link Terminal
FSP	Forward Space Packet (SLE service)
HRAF	Hypothetical RAF service (see annex B)
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
ISP1	Internet SLE Protocol One
NB	Non-Blocking (CSTS SFW Operation)
PDU	protocol data unit
PM	Provision Management
RAF	Return All Frames (SLE service)
RCF	Return Channel Frames (SLE service)
ROCF	Return Operational Control Fields (SLE service)
RSLD	Return Space Link Data

Term	Meaning
SF	Stateful (CSTS SFW procedure)
SFW	Specification Framework (reference [1])
SL	Stateless (CSTS SFW procedure)
SLE	Space Link Extension
TC	Telecommand
TCP	Transmission Control Protocol
TM	Telemetry
TT&C	Telemetry, Tracking, and Command
UM	Utilization Management

ANNEX B

OUTLINE OF CSTS SPECIFICATION EXAMPLES

B1 INTRODUCTION

This annex outlines examples for the specification of Cross Support Transfer Services in order to illustrate how the CSTS Specification Framework (reference [1]) and the Guidelines for Specification of CSTSes (reference [2]) can be applied, and to demonstrate that they lead to smaller specifications that are easier to develop and to review. The examples addressed include

a) the Monitored Data Cross Support Transfer Service (reference [9]), which is composed completely of common procedures defined in the CSTS Specification Framework; and

b) a hypothetical Return All Frames CSTS, that is, the SLE RAF service re-specified based on the CSTS Specification Framework in order to illustrate use of an abstract service and derivation from an existing CSTS.

The examples in this annex are only outlined to show the essentials, and the presentation clearly does not conform to the documentation rules set forth in the Guidelines for Specification of CSTSes (reference [2]). For a CSTS specification that conforms to reference [2], the reader is asked to refer to reference [9].

B2 MONITORED DATA CSTS

The Monitored Data Cross Support Transfer Service is a real CSTS specified in reference [9]. It allows a spaceflight mission to receive cyclic reports on, and optionally to query the current values of, parameters that are pertinent to Cross Support Services being provided by a Provider CSSS. The Monitored Data service also allows a spaceflight mission to optionally receive notifications of the occurrence of events of interest associated with the services that are being provided by a Provider CSSS. For this purpose, the service uses the following procedures defined in the CSTS Specification Framework:

a) the Association Control procedure, to establish and release the association between the service user and the service provider;

b) the On-Change-Option Cyclic Report procedure derived from the CSTS SFW Cyclic Report procedure, for periodic or on-change delivery of monitored parameter values;

c) the refined Information Query procedure, to allow the service user to query the values of monitored parameters; and

d) the refined Notification procedure, to notify events to the service user.

Table B-1, copied from reference [9], shows how the service is composed in accordance with the requirements set forth in the Guidelines for Specifications of CSTSes (reference [2]):

The first row ('Procedure') identifies the procedure types used; in this row the table also specifies that an instance of the On-Change-Option Cyclic Report procedure shall be the prime procedure instance.

Table B-1: Composition of the Monitored Data CSTS

Procedure	Association Control	On-Change-Option Cyclic Report [P]	Information Query	Notification
Version	-	1	1	1
No. of Instances	1..1	1..*	0..1	0..*
Specification Approach	adopted	refined-and-extended	refined	refined
Source	CSTS SFW [1]: Association Control	CSTS SFW [1]: Cyclic Report	CSTS SFW [1]: Information Query	CSTS SFW [1]: Notification

The second row ('Version') identifies the versions of the procedure specifications. In this case, the Association Control procedure is the same as the version of the adopted procedure and therefore is denoted by '-'. All extended and/or refined procedures have the version number '1'.

The third row ('No. of Instances') identifies how many instances of the applicable procedure type can be used in a service instance by specifying the minimum number and the maximum number of instances:

a) One and only one instance of the Association Control Instance may be used, as mandated by the Guidelines for Specification of CSTSes.

b) At least one instance of the On-Change-Option Cyclic Report procedure must be present, and this would then also be the prime procedure instance. However, the service user and service provider may agree to use more than one instance in order to allow cyclic monitoring of different sets of monitored parameters concurrently and independently.

c) Both the Information Query and the Notification procedures are optional because the minimum number of instances is specified as zero. If used, there shall only be a single instance of the Information Query procedure, whereas a service instance might make use of multiple Notification procedures to notify multiple sets of events concurrently and independently.

The actual number of instances of On-Change-Option Cyclic Report, Notification, and Information Query procedures for any given instance of the Monitored Data CSTS must be fixed by Service Management for a service instance.

The fourth row ('Specification Approach') says either that a procedure is adopted from the source specification without modification or that it is derived from the source specification by extension, refinement, or both.

The fifth and last row ('Source') identifies the procedure specification that is adopted or from which the applicable procedure is derived.

B3 HYPOTHETICAL RETURN ALL FRAMES CSTS

B3.1 INTRODUCTION

This subsection outlines how an existing SLE return link service, the RAF service (reference [10]) could be re-specified using the CSTS Specification Framework. At the time of writing, CCSDS has no plans to re-specify any of the existing SLE services as Cross Support Transfer Services based on the CSTS Specification Framework, and therefore the approach outlined here is purely hypothetical. In order to avoid any confusion with the SLE RAF Recommended Standard, this example uses the abbreviation HRAF for Hypothetical RAF.

Because all SLE Return services share a number of common specifications and concepts, this example specifies a Return Space Link Data (RSLD) CSTS to capture those communalities in a manner that the specifications can be reused in any derived return link service. The RSLD-CSTS contains incomplete specifications and is not intended for implementation; it is therefore an **abstract** service (see 10.6). In particular, the RSLD-CSTS does not specify the data that are transferred from the service provider to the service user. The Hypothetical RAF service is then derived from the RSLD-CSTS. Using the same approach, it would also be possible to derive hypothetical CSTS specifications for the SLE services RCF, ROCF, and others, and that would justify development of an additional abstract service specification.

The outline description of the RSLD-CSTS and the HRAF-CSTS in the following subsection focuses on specification of a Cross Support Service provision in order to demonstrate how the CSTS Specification Framework can be used. A real specification must of course also address Cross Support Service production and define how the service fits into the overall Cross Support Reference Model. Moreover, the description in the following subsections strives to keep as close to the original SLE RAF specification as possible, without consideration whether or not that would technically be the best approach.

NOTE – The following subsections make use of service composition tables, the contents of which has been explained in B2.

B3.2 ABSTRACT RETURN SPACE LINK DATA CSTS

B3.2.1 Purpose

The purpose of the Return Space Link Data CSTS is to capture common specifications for all services that deal with delivery of space link data units received from the space link to a service user. The RSLD-CSTS is an abstract service that needs to be specialized by a derived service before the specification can be implemented.

B3.2.2 Service Composition

The composition of the RSLD-CSTS is shown in table B-2. An instance of the RSLD service includes

a) one instance of an Association Control procedure adopted from the CSTS Specification Framework without modification;

b) one instance of a new RSLD Delivery procedure derived from the Buffered Data Delivery (BDD) procedure defined in the CSTS Specification Framework for the delivery of Return Space Link Data from the service provider to the service user;

c) one instance of a new RSLD Status Report procedure derived from the Cyclic Report procedure defined in the CSTS Specification Framework for cyclic reporting of service status parameters to the service user; and

d) one instance of a new RSLD Configuration Query procedure derived from the Information Query procedure in the CSTS Specification Framework, by which the service user can retrieve the current values of one or more configuration parameters.

Table B-2: Composition of the Return Space Link Data CSTS

Procedure	Association Control	Delivery [P]	Status Report	Configuration Query
Version	-	1	1	1
No. of Instances	1..1	1..1	1..1	1..1
Specification Approach	adopted	extended-and-refined	refined	refined
Source	CSTS SFW [1]: Association Control	CSTS SFW [1]: Buffered Data Delivery	CSTS SFW [1]: Cyclic Report	CSTS SFW [1]: Information Query

B3.2.3 RSLD Delivery Procedure

The purpose of the RSLD Delivery procedure is to transfer return space link data (e.g., telemetry frames or operational control fields extracted from telemetry frames) from the service provider to the service user. Through its derivation from the Buffered Data Delivery procedure, it supports the real-time delivery mode and the complete delivery mode as well as the reporting of events synchronized with the data transfer.

The RSLD Delivery procedure does not add new operations to the Buffered Data Delivery procedure and does not extend the behavior of the Buffered Data Delivery procedure, but it does extend the operations and narrow the semantics of inherited parameters. The operations of the procedure are identified in table B-3, which conforms to documentation requirements in reference [2].

Table B-3: RSLD Delivery Required Operations

Operation	Source	Extended	Refined	Procedure Blocking/Non-Blocking
START	BDD	N	Y	Blocking
STOP	BDD	N	N	Blocking
TRANSFER-DATA	BDD	Y	Y	Non-Blocking
NOTIFY	BDD	Y	N	Non-Blocking

START Operation

The meaning of the inherited parameters `start-generation-time` and `stop-generation-time` are refined to refer to the Earth receive time of the frame from which a space data unit delivered by a derived service is extracted.

TRANSFER-DATA Operation

The meaning of the inherited parameter `generation-time` is refined to refer to the Earth receive time of the frame from which the space data unit delivered by the TRANSFER-DATA invocation is extracted.

Furthermore, the RSLD specification extends the TRANSFER-DATA operation by the parameters identified table B-4. (For the meaning of the parameters, see reference [10].)

Table B-4: RSLD Delivery TRANSFER-DATA Extension Parameters

Extension Parameters	Invocation
`antenna-id`	M
`data-link-continuity`	M
`private-annotation`	M

The RSLD specification leaves the `data` parameter of the TRANSFER-DATA operation undefined and delegates its specification to derived services.

NOTIFY Operation

The RSLD specification adds the new event 'loss of frame synchronization' and specifies that the following information shall be included in the notification that reports this event (see reference [10]):

a) the UTC time when the frame synchronizer transitioned from 'in-lock' to 'out-of-lock';

b) the current status of the carrier demodulation process, which shall be 'in-lock', 'out-of-lock', or 'unknown';

c) the current status of the subcarrier demodulation process, which shall be 'in-lock', 'out-of-lock', 'not in use', or 'unknown';

d) the current status of the symbol synchronization process, which shall be 'in-lock', 'out-of-lock', or 'unknown'.

This new notification shall be delivered synchronously and may be discarded together with data in the real-time delivery mode in the same way as the notifications inherited from the Buffered Data Delivery procedure.

B3.2.4 RSLD Status Report Procedure

The RSLD Status Report procedure is derived from the Cyclic Report procedure and used to report to the service user the current values of a set of parameters that present the status of the Cross Support Service production and provision. The RSLD specification does not extend or refine any operations, but adopts the behavior specification unmodified. It does, however, define the parameters that can be included into the status report, and therefore the RSLD Status Report procedure is a derivation of the Cyclic Report procedure.

The RSLD specification identifies the following status parameters that can be included in the report (for the meaning of these parameters, see reference [10]), including the name and the data type specifications, as required by the Cyclic Report procedure:

a) `production-status` (named integer supporting the values 'configured', 'operational', 'interrupted', and 'halted');

b) `number-of-data units-delivered` (unsigned integer);

c) `frame-sync-lock-status` (named integer supporting the values 'in lock', 'out of lock', and 'unknown');

d) `symbol-sync-lock-status` (named integer supporting the values 'in lock', 'out of lock', and 'unknown');

e) `subcarrier-lock-status` (named integer supporting the values 'in lock', 'out of lock', 'not in use', and 'unknown');

f) `carrier-lock-status` (named integer supporting the values 'in lock', 'out of lock', and 'unknown').

Enumerated values are represented as 'named integers' rather than using the simple built-in ASN.1 type ENUMERATED because the latter is not extensible. More details on this topic may be found in F3 of the CSTS Specification Framework.

The RSLD specification additionally states that derived services may extend the list of status parameters.

B3.2.5 RSLD Configuration Query Procedure

The RSLD Configuration Query procedure is derived from the Information Query procedure and may be used by the service user to query the current value of one or more 'gettable' parameters. The RSLD specification does not extend or refine any operations, but adopts the behavior specification unmodified. It does, however, define the parameters that can be queried, and therefore the RSLD Configuration Query procedure is a derivation of the Cyclic Report procedure.

The RSLD specification identifies the following parameters that can be queried (for the meaning of these parameters, see reference [10]), including the name and the data type specifications, as required by the Information Query procedure:

a) `delivery-mode` (named integer supporting the values 'real-time' and 'complete');

b) `latency-limit` (unsigned integer);

c) `reporting-cycle` (unsigned integer);

d) `return-timeout-period` (unsigned integer);

e) `transfer-buffer-size` (unsigned integer).

The RSLD specification additionally states that derived services may extend the list of parameters that may be queried.

B3.3 HYPOTHETICAL RETURN ALL FRAMES CSTS

B3.3.1 Purpose

The HRAF service is a Cross Support Transfer Service based on the CSTS Specification Framework that delivers to a mission user all telemetry frames from one space link physical channel.

B3.3.2 Service Composition

The HRAF service is derived from the Return Space Link Data CSTS and as such inherits all procedures included in the RSLD-CSTS. The HRAF specification does not add any procedures to the set defined by the RSLD specification. The resulting service composition is shown in table B-5. In this table, '[55]' is assumed to be the reference of the hypothetical RSLD-CSTS Specification.

Table B-5: Composition of the Hypothetical Return All Frames CSTS

Procedure	Association Control	HRAF Delivery [P]	HRAF Status Report	HRAF Configuration Query
Version	-	1	1	1
No. of Instances	1..1	1..1	1..1	1..1
Specification Approach	adopted	extended	extended	extended
Source	CSTS SFW [1]: Association Control	RSLD [55]: Delivery	RSLD [55]: Status Report	RSLD [55]: Configuration Query

An instance of the HRAF service includes

a) one instance of Association Control procedure, adopted without modification;

b) one instance of a new HRAF Delivery procedure derived from the RSLD Delivery procedure for the delivery of frames from the service provider to the service user;

c) one instance of a new RSLD Status Report procedure derived from the RSLD Status Report procedure; and

d) one instance of a new HRAF Configuration Query procedure derived from the RSLD Configuration Query procedure.

B3.3.3 HRAF Delivery Procedure

The purpose of the HRAF Delivery procedure is to transfer frames received from the space link from the service provider to the service user. Through its derivation from the RSLD Delivery procedure, it inherits basic selection criteria (start and stop Earth receive time), real-time and complete delivery modes, and reporting of events synchronously with data transfer.

In order to provide the same service as the SLE RAF service, the following must be added:

a) detailed specification of the data to be transferred;

b) selection of the frame quality in the START operation;

c) reporting of the delivered frame quality in the TRANSFER-DATA operation.

The operations used by the procedure and their inheritance are shown in table B-6. As can be seen from the table, the list of reported events is the same as defined by the RSLD-CSTS, and the NOTIFY operation can therefore be adopted unmodified.

Table B-6: HRAF Delivery Required Operations

Operation	Source	Extended	Refined	Procedure Blocking/Non-Blocking
START	RSLD	Y	N	Blocking
STOP	RSLD	N	N	Blocking
TRANSFER-DATA	RSLD	Y	N	Non-Blocking
NOTIFY	RSLD	N	N	Non-Blocking

START Operation

The HRAF specification extends the START operation as shown in table B-7.

Table B-7: HRAF Delivery START Extension Parameters

Extension Parameters	Invocation
`requested-frame-quality`	M

As specified in reference [10], the parameter `requested-frame-quality` allows the service user to request all frames, only good frames, or only erred frames.

TRANSFER-DATA Operation

The inherited parameter `data` is extended to specify the content of the `data` parameter. It varies depending on the applicable coding scheme and on the frame quality.

Furthermore, the HRAF specification extends the TRANSFER-DATA operation as shown in table B-8.

Table B-8: HRAF Delivery TRANSFER-DATA Extension Parameters

Extension Parameters	Invocation
`delivered-frame-quality`	M

The parameter `delivered-frame-quality` identifies the quality of the delivered frame as ‘good’, ‘erred’, or ‘undetermined’.

SLE Reference

The HRAF Delivery procedure corresponds to the combination of the RAF operations RAF-START, RAF-STOP, RAF-TRANSFER-DATA, and RAF-SYNC-NOTIFY. The HRAF service based on the CSTS Specification Framework differs from the original SLE RAF service in that the complete online delivery mode and the offline delivery mode have been combined into a single complete delivery mode. The timely online delivery mode has been renamed ‘real-time’ delivery mode, but its specification is the same as in reference [10].

B3.3.4 HRAF Status Report Procedure

The only modification of the inherited RSLD Status Report procedure is the addition of the parameter `number-of-error-free-frames-delivered` to the set of parameters that can be included in the status report.

SLE Reference

The HRAF Status Report procedure corresponds to the combination of the SLE RAF operations RAF-SCHEDULE-STATUS-REPORT and RAF-STATUS-REPORT. RAF-SCHEDULE-STATUS-REPORT has been replaced by the START and STOP operations. The option to request only one report has been discarded; the service user will have to explicitly STOP reporting after the first report.

In contrast to the specification in reference [10], the service user can specify the parameters that shall be included in the report. The same approach as in reference [10] can be achieved by leaving the selection empty and defining the complete list of parameters as the ‘default list’ for the service.

B3.3.5 HRAF Configuration Query Procedure

The only modification of the inherited RSLD Configuration Query procedure is the addition of the parameter `requested-frame-quality` to the set of parameters that can be queried.

SLE Reference

The HRAF Configuration Query procedure corresponds to the SLE RAF operation RAF-GET-PARAMETER. The service user may specify one or more parameters in the invocation, whereas reference [10] only allows querying a single parameter at a time.

B4 FORWARD FRAME CSTS

The Forward Frame Cross Support Transfer Service is a real CSTS specified in reference [24]. It allows forward space link PDUs to be encoded and radiated to a Space User Node (e.g., a Mission spacecraft), depending on the procedure type selected by configuration as prime procedure either in asynchronous mode for the transfer of variable-length frames or at constant frame rate for the transfer of fixed-length frames. The Forward Frame CSTS is applicable in situations in which there is a need for multiple users to concurrently send data on the same space link physical channel to a Space User Node. Transfer frames from various users are multiplexed into a single stream of frames and transmitted by a single service production entity.

For this purpose, the service uses the following procedures defined in the CSTS Specification Framework:

a) the Association Control procedure to establish and release the association between the service user and the service provider;

b) the Sequence Controlled Frame Data Processing procedure derived from the CSTS SFW Sequence-Controlled Data Processing procedure for processing variable-length transfer frames if this procedure is configured to act as prime procedure;

c) the Buffered Frame Data Processing procedure derived from the CSTS SFW Buffered Data Processing procedure for processing fixed length transfer frames if this procedure is configured to act as prime procedure;

d) optionally, the Cyclic Report procedure to periodically deliver monitor data to the service user;

e) optionally, the Information Query procedure to allow the service user to query the values of monitored parameters;

f) optionally, the Notification procedure to notify events to the service user; and

g) optionally, the Master Throw Event procedure derived from the CSTS SFW Throw Event procedure to allow a service user to reconfigure the values of dynamically modifiable configuration parameters of the Functional Resource instances that comprise the production processes that directly support that instance of the Forward Frame service.

Table B-9 shows how the service is composed in accordance with the requirements set forth in the Guidelines for Specifications of CSTSes (reference [2]):

The first row ('Procedure') identifies the procedure types used; in this row, in combination with the NOTE, the table also specifies that either the Sequence-Controlled Frame Data Processing procedure or the Buffered Frame Data Processing procedure shall be the prime procedure instance.

Table B-9: Composition of the Forward Frame CSTS

Procedure	Association Control	Sequence-Controlled Frame Data Processing [P] (See NOTE)	Buffered Frame Data Processing [P] (See NOTE)	Cyclic Report	Information Query	Notification	Master Throw Event
Version	-	1	1	-	-	-	1
No. of Instances	1..1	0..1	0..1	0..*	0..1	0..*	0..1
Specification Approach	adopted	refined-and-extended	refined-and-extended	adopted	adopted	adopted	refined-and-extended
Source	Reference [1]: Association Control (4.3)	Reference [1]: Sequence-Controlled Data Processing (4.8)	Reference [1]: Sequence-Controlled Data Processing (4.8)	Reference [1]: Cyclic Report (4.10)	Reference [1]: Information Query (4.9)	Reference [1]: Notification (4.11)	Reference [1]: Throw Event (4.12

NOTE – If the Sequence-Controlled Frame Data Processing procedure is configured as the prime procedure, then the Buffered Frame Data Processing procedure is not instantiated. Likewise, if the Buffered Frame Data Processing procedure is configured as Prime procedure, then the Sequence-Controlled Frame Data Processing procedure is not instantiated.

The second row ('Version') identifies the versions of the procedure specifications. In this case, the Association Control procedure is the same as the version of the adopted procedures and therefore is denoted by '-'. All extended and/or refined procedures have the version number '1'.

The third row ('No. of Instances') identifies how many instances of the applicable procedure type can be used in a service instance by specifying the minimum number and the maximum number of instances:

a) one and only one instance of the Association Control Instance may be used as mandated by the Guidelines for Specification of CSTSes;

b) exactly one instance of either the Sequence-Controlled Frame Data Processing procedure or the Buffered Frame Data Processing procedure must be present, and this would then also be the prime procedure instance;

c) Cyclic Report, the Information Query, and the Notification procedures are optional because the minimum number of instances is specified as zero. If used, there shall only be a single instance of the Information Query procedure, whereas a service instance might make use of multiple Cyclic Report and Notification procedures to obtain multiple sets of monitored data notifications for multiple sets of events concurrently and independently;

d) The Master Throw Event procedure is optional because the minimum number of instances is set to zero. If used, there shall only be a single instance of the Master Throw Event procedure.

The actual number of instances of Cyclic Report and Notification procedures for any given instance of the Forward Frame CSTS must be fixed by Service Management.

The fourth row (‘Specification Approach’) says either that a procedure is adopted from the source specification without modification, in this case the Association Control, the Cyclic Report, the Information Query, and the Notification procedures, or that it is derived from the source specification by extension, refinement, or both, in this case the Sequence-Controlled Frame Data Processing, the Buffered Frame Data Processing, and the Master Throw Event procedures.

The fifth and last row (‘Source’) identifies the procedure specification that is adopted or from which the applicable procedure is derived.