CCSDS
The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

APPLICATION AND SUPPORT LAYER ARCHITECTURE

INFORMATIONAL REPORT

CCSDS 371.0-G-1

GREEN BOOK
November 2020

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

APPLICATION AND SUPPORT LAYER ARCHITECTURE

INFORMATIONAL REPORT

CCSDS 371.0-G-1

GREEN BOOK
November 2020

AUTHORITY

Issue:	Informational Report, Issue 1
Date:	November 2020
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

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 371.0-G-1	Application and Support Layer Architecture, Informational Report, Issue 1	November 2020	Original issue

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

Figure Page

Table

1 INTRODUCTION

1.1 PURPOSE

CCSDS develops standards that support interoperable interfaces between the various elements of space data systems that may be distributed between different space agencies or other institutions. CCSDS itself is organized into a number of Areas, each of which has responsibility for standardization of different elements of space data systems. These standards have been developed over decades, with the intention that the various standards developed by CCSDS can be deployed individually or aggregated together in actual space data systems. These standards must provide a stable base for developing ground systems and missions, but at the same time, CCSDS must support evolution as new technologies become available and are standardized for broad use in interoperation and cross-support situations.

CCSDS has defined a reference architecture for space data systems that shows how all of the CCSDS Recommended Standards are intended to be configured and deployed for use. This CCSDS reference architecture is being published in two parts, essentially distinguishing communications services that transport and deliver data from applications services that use these delivery services to carry out missions.

This Application and Support Layer (ASL) Reference Architecture specifically addresses both published standards and the current roadmap for development of standards for the second of these, the application and upper layers of the International Organization for Standardization (ISO) protocol stack (reference [43]). A separate document, the *Space Communications Cross Support—Architecture Description Document* (SCCS-ADD) (reference [50]), describes all of the defined underlying communications and networking services upon which the ASL depends.

This reference architecture is intended to inform users of CCSDS standards about the features that are available and how they may be effectively combined to create interoperable space data systems. This document, in particular, is intended for developers of mission operations systems and associated spacecraft. It provides descriptions of the primary features of a broad set of standards as well as pointers to where additional, in depth, information may be found.

1.2 SCOPE

The scope of the ASL Reference Architecture presented in this document covers Application Layer and support services, interfaces, and data exchange standards (Open Systems Interconnection [OSI] upper-layer functions, layers 5-7) that represent interoperability boundaries between agencies or systems. These are developed by the CCSDS Mission Operations and Information Management Services (MOIMS) and Spacecraft Onboard Interface Services (SOIS) Areas. It includes Application Layer services and protocols on ground and in flight and also addresses the syntax and semantics of data formats exchanged across such interfaces and how the standard information objects those data formats contain may be used or referenced across multiple service interfaces. The Spacecraft Onboard (SO) Area defines other spacecraft onboard services for subnet and wireless communications, which are also covered in this document. In principle, Application Layer interactions may be considered independent of the underlying communications architecture, but an actual deployment will need to consider the full protocol stack, including the underlying communications layers addressed in the SCCS and SOIS documents, to achieve end-to-end communication.

The scope of this document is specifically those CCSDS service and data exchange standards associated with the Mission Operations (MO) and SO functions and services. In this sense, it is a companion document to the SCCS-ADD (reference [50]) that addresses standards associated with the CCSDS Space Link Services (SLS), Space Internetworking Services (SIS), Cross Support Services (CSS), and Systems Engineering Area (SEA) Areas. All communications between Application Layer system elements, on the ground and in flight, will utilize these underlying Data Link Layer and/or Network Layer services and compatible terrestrial services when appropriate. All of the terrestrial service deployments are assumed to use the cross-support services to plan, schedule, configure, and utilize the space communications terminals that provide access to space. Some cross support standards are referenced directly within the ASL Reference Architecture, as they are themselves Application Layer services or directly call them.

With respect to the context of space link and network communications architecture, this document should be read in conjunction with the SCCS-ADD (reference [50]).

The scope of this reference architecture includes (for the MOIMS services and SOIS Areas):

– published CCSDS Recommended Standards;

– CCSDS Recommended Standards currently under development;

– identified future CCSDS Recommended Standards, forming part of the road map for standardization, as defined in published CCSDS Green Books or Working Group Charters;

– functional interactions for which there is currently no identified CCSDS Recommended Standard, but for which future standards may be identified.

The current status of standards development is clearly indicated within this reference architecture, but is also subject to change over time as the CCSDS Working Groups make progress on their planned development of new standards and updates to current ones. The CCSDS Project framework, https://cwe.ccsds.org/fm/Lists/Projects/AllItems.aspx, provides a view of the plans for updating them. Normal CCSDS processes require reviewing, and possibly updating, standards every five years. In the normal process of development, this document will be reviewed and updated in the same way, and will reflect additions, updates, and the maturation of standards that are presently in work. The CCSDS Web site, www.ccsds.org, is the best reference for the current state of any standards referenced in this document.

It is important to keep in mind that detailed definitions of the services and data exchange formats are not contained within this document, but are to be found within the referenced standards themselves. This reference architecture only models these to the extent required to identify the standards and describe the relationships among them.

1.3 RATIONALE

The complete CCSDS reference architecture can be used to:

- show how CCSDS Recommended Standards may be orchestrated together within actual space data systems;
- identify gaps in current and planned standardization at these layers;
- identify overlaps between current and planned standardization at these layers.

The first steps in this process defined the Space Communications Cross Support (SCCS) architecture (references [50] and [2]), which addresses the relationships among Physical to Transport Layer standards, security standards, and some Application Layer process and transport services, developed within four CCSDS Areas:

- SLS;
- SIS;
- CSS;
- SEA.

The purpose of this document is to provide the second half of this reference architecture and to cover ASL standards associated with the remaining CCSDS Areas:

- MOIMS;
- SOIS.

Taken together with the SCCS-ADD, these documents provide an understanding of how all CCSDS protocols, services, and data exchange standards may be used to work together in the context of a space data system.

As in most CCSDS technical documents, in this document these standards are be described based on their functional groupings and deployment locations and are not referenced by the Areas or Working groups that develop and maintain them. They are described as MO services and standards and Spacecraft Onboard (SO) services and standards.

The policy for CCSDS documents is that they are to be reviewed, and updated as needed, nominally every five years. This document, because of its design that touches on many other standards, is likely to become out of date during that time. The reader is cautioned that this may be the case and it is strongly recommended that the CCSDS Publications Web site be consulted for the latest versions of any standards identified herein. This is particularly the case for any standards that are marked [Future] or [Prospective].

In cases where there is no existing CCSDS Recommended Standard, but one is in process or in the plan for future CCSDS standardization, it is marked [Future]. If there is a described concept for a standard, but no concrete plan to produce it, it is marked as [Prospective]. There is no guarantee that [Prospective] standards will be produced or that they will look like what is described herein.

1.4 DOCUMENT STRUCTURE

This document is organized into the following main sections:

- ASL Architecture Concepts: provides a technical background and overview of the MO services and services.
- ASL Reference Architecture: defines the seven modeling viewpoints and the graphical notations used in this document.
- The remaining sections document each of the seven viewpoints in turn, extended from the RASDS (reference [1]) viewpoints:
  - Functional Viewpoint;
  - Information Viewpoint;
  - Service Viewpoint;
  - Communications (Protocol) Viewpoint;
  - Physical (Connectivity) Viewpoint;
  - (Functional) Deployment Viewpoint;
  - Implementation Viewpoint.
- Annex A: Acronyms.
- Annex B: Functional Viewpoint Alternative Style Diagrams.

1.5 DEFINITIONS

Any terms that are defined specifically in this document are defined in this section. There are also terms referenced from other documents that are shown for easy reference, along with their sources.

action: (MO Monitor and Control [M&C] Services) A single executable task of a service provider, a telecommand is an example of an action. (Reference [17].)

actuator: (SO) A component of a machine that is responsible for moving and controlling a mechanism or system.

aggregation: (MO M&C Services) A collection of parameters provided as a set by a service provider. (Reference [17].)

alert: (MO M&C Services) Any operationally significant event. (Reference [17].)

argument: A run-time parameter provided to various control items on invocation. Arguments apply to **actions**, **alerts**, **procedures**, **planning activities,** and **planning events**, among other items. [MO services, multiple references.]

Asynchronous Message Service, AMS: (SIS) The protocol procedures and data units that accomplish automatic configuration of Asynchronous Messaging Service (AMS) (reference [41]) communication relationships, dynamic reconfiguration of those relationships during operations, and the use of those relationships to accomplish the exchange of mission information among data system modules.

autocoding: An automated process for translating machine-readable interface specifications into executable or compilable code in some programming language.

check: (MO M&C Services) A check performed periodically on the value of a **parameter**, which may be of (but is not limited to) one of the following check types:

– limit check: the value lies within a specified range;

– constant check: the value is checked against a specified value or value of another parameter;

– delta check: the change in value is checked against a pair of thresholds. (Reference [17].)

COM activity: (MO Common Object Model) Anything that has a measurable period of time (a command, a remote procedure, a schedule, etc.). (Reference [16].)

Specifically, a COM activity is a compound pattern of **COM objects** representing any type of operation that is repeatable and extends over a measurable period of time. It comprises four elements, each of which is itself a COM object: **identity**, **definition**, **instance,** and **COM event**.

COM event: (MO Common Object Model) A specific object representing 'something that happens in the system at a given point in time'. (Reference [16].)

COM instance: (COM object pattern) A compound **information object** that can be dynamically instantiated. It can be used to represent operations that are repeatable and extend over a measurable period of time. During the lifetime of the operation, multiple attributes may be dynamically updated, not just its status. Examples are M&C procedures, **planning activities,** and **planning events**. It comprises four elements, each of which is itself a **COM object**: **identity**, **definition**, **instance,** and **update**.

COM object: (MO Common Object Model) A thing that is recognized as being capable of an independent existence and that can be uniquely identified. An object may be a physical object such as a spacecraft or a ground station, an event such as an eclipse, or a concept such as telemetry parameter. It forms the fundamental part of a service specification, for example, a parameter definition, a parameter value at a given point in time, a command. There are no requirements on what an object may be except that it must be possible to uniquely identify an instance of it. (Reference [16].)

Specifically, a COM object conforms to the structure of an MO Common Model Object defined in reference [16].

COM state: (COM object pattern) A compound **information object** representing a status: it is a persistent object for which there is only one status value at any given time (although this may not be known). Examples include M&C Parameters and **planning resources**. It comprises three elements, each of which is itself a **COM object**: **identity**, **definition,** and **update**.

COM static item: (COM object pattern) A compound **information object** comprising only statically declared information with no evolving status. Examples are M&C **checks** and **conversions**, and planning request templates. It comprises two elements, each of which is itself a **COM object**: **identity** and **definition**.

component: (SO) A logical element of a system accessed through defined interfaces. A component may be purely conceptual or realized in software or hardware (e.g., as a field-programmable gate array).

consumer, **service consumer**: A component that consumes or uses a service provided by another component. A component may be a provider of some services and a consumer of others. (Reference [17].)

convergence layer: (SO) Functions that can be inserted between SO subnetwork services and the agency-specific subnetwork software to provide uniform qualities of service across a variety of subnetwork technologies.

conversion: (MO M&C Services) Change in the representation of the value of a **parameter** or the **arguments** of an **action** or **alert**. This may, for example, be between the raw value and calibrated engineering units. (Reference [17].)

correspondence: (RASDS) The relationship between objects in the viewpoint in which they are defined and references to these from objects defined in other viewpoints.

definition: (COM object pattern) The statically declared information associated with an **information object**. This may, for example, include a description, set of defined **arguments**, or any other information that applies to all occurrences of the information object. There may be multiple definitions (versions) over the mission lifetime associated with the same **identity**, each with its own unique definition ID.

device: (SO) A physical element of a system accessed through subnetwork-layer interfaces.

deployment node: (RASDS) A target location for the deployment of **functions**. **Deployment nodes** may correspond to physical locations, computers or other devices, or virtual devices.

Deployment nodes are the building blocks of the Physical and Deployment Viewpoints of the reference architecture contained in this document.

Dictionary of Terms: (SO) Ontology of terms used to describe data in interfaces in electronic data sheets.

domain: A namespace that partitions separately addressable entities (e.g., actions, parameters, alerts [or any **information object**]) in the space system. The space system is decomposed into a hierarchy of domains within which entity identifiers are unique. (Reference [54].)

electronic data sheet, EDS: (SO) XML Specification, an electronic description of a device's metadata, device-specific functional and access interfaces, device-specific access protocol, and, optionally, device abstraction control procedure.

event: A time-stamped message, containing (changes in) information about **information objects**, that is exchanged across service interfaces and potentially stored in service history. (Reference [54].)

(See **COM event**.)

function: (RASDS) The set of actions or activities performed by some object to achieve a goal. The transformation of inputs to outputs that may include the creation, modification, monitoring, or destruction of elements (reference [1]).

Hierarchical **functions** are the building blocks of the Functional Viewpoint of the reference architecture contained in this document.

group: (MO M&C Services) A collection of COM objects of the same type. (Reference [17].)

identity: (COM object pattern) A combination of the **domain** and a unique name within the domain used to reference all occurrences of an information object (compound object) throughout the mission lifetime.

information object: (RASDS) The set of information about a real-world entity that is exchanged across an interface between **functions**. This may include static definitions, dynamic status, and metadata (reference [1]).

Hierarchical **information objects** are the building blocks of the Information Viewpoint of the reference architecture contained in this document.

instance: (COM object pattern) The current status associated with the **information object** or a specific occurrence of it. This may, for example, include a current value or set of values for defined arguments. Each occurrence of the information object has a separate instance, with its own unique instance ID.

onboard: (SO) Carried within or occurring aboard a vehicle (such as a spacecraft).

PackageFile: (SO) An element of the SO Electronic Data Sheets (SEDS) schema.

parameter: A single unit of data reported by a service provider. (Reference [17].)

plan: The output of the planning process. It contains a set of selected activities associated with time, position, or other event. A plan may contain additional related information. (Reference [62].)

planning activity: A meaningful unit of what can be planned. The granularity of a planning activity depends on the use case; it may be hierarchical. In other words, planning activities are the building block for planning. (Reference [62].)

planning database, PDB: A collection of all planning configuration data, including the definitions of **planning activities**, **planning events**, and **planning resources**, as well as templates for **planning requests**. (Reference [63].)

planning event: Event meeting a plan condition. The condition can be expressed in terms of time, location, or any other **planning resource**. (Reference [62].)

planning request, PRQ: An input to the planning process, which requests one or more **planning activities**. Each planning request contains all the information that the requester can provide. (Reference [62].)

planning resource: Any physical or virtual quantity that either impacts or is affected by the execution of the **planning activities**. (Reference [62].)

procedure: A single executable task of an Automation service (reference [57]) provider, which may have an extended duration. A procedure may correspond to a simple predefined sequence of **actions**, a complex procedure script or a software function that is executed automatically by the service provider.

provider: (see **service provider)** A component that offers a service to another by means of one of its provided service interfaces. A component may be a provider of some services and a consumer of others. (Reference [17].)

real-time: (SO) A computing paradigm in which computation is constrained to occur within deadlines prescribed by real events.

sensor: (SO) A device that responds to a physical stimulus (such as heat, light, sound, pressure, magnetism, or a particular motion) and transmits a resulting impulse (as for measurement or operating a control).

service: (RASDS) A set of capabilities that a component provides to another component via an interface. A service is defined in terms of the set of operations that can be invoked and performed through the service interface. Service specifications define the capabilities, behavior, and external interfaces, but do not define the implementation. (Reference [17].)

Services are the building blocks of the Service Viewpoint of the reference architecture contained in this document.

service provider: (SCCS) The role played by a physical, functional, or organizational entity that provides a cross-support service for a service user. (A single entity may play the roles of service provider and service user at the same time.) (Reference [2].)

(See also **provider** for the definition of the term used within MO services.)

service user: (SCCS) The role played by a physical, functional, or organizational entity that uses a cross-support service provided by a service provider. (A single entity may play the roles of service provider and service user at the same time.) (Reference [2].)

(See also **consumer** for the definition of the term used within MO services.)

statistic: (MO M&C Services) A defined statistical evaluation of the value of a **parameter**.

subnet: (SO) A separate and identifiable portion of a spacecraft's onboard communications network using a single technology.

update: (COM object pattern) The current state of an **information object** at a specific point in time, which can contain multiple dynamically changing attributes. Updates may be used to disseminate changing status and to record the detailed status history of the information object.

view: (RASDS) A representation of a whole system from the perspective of a set of concerns. Views are themselves modular and well formed, and each view is intended to correspond to exactly one viewpoint. A view may include representations or correspondences to elements defined in other viewpoints (reference [1]).

viewpoint: (RASDS) A form of abstraction achieved using a selected set of architectural concepts and structuring rules in order to focus on particular concerns within a space data system. A viewpoint specification defines a pattern or template from which to construct individual views, and it establishes the rules, techniques, and methods employed in constructing a view (reference [1]).

wireless: (SO) The transmission of data via electro-magnetic propagation, specifically via a digital packet communication network.

1.6 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] *Reference Architecture for Space Data Systems*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 311.0-M-1. Washington, D.C.: CCSDS, September 2008.

[2] *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.

[3] *Space Packet Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-1. Washington, D.C.: CCSDS, September 2003.

[4] *CCSDS File Delivery Protocol (CFDP)*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 727.0-B-4. Washington, D.C.: CCSDS, January 2007.

[5] *Spacecraft Onboard Interface Services—Subnetwork Packet Service*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 851.0-M-1. Washington, D.C.: CCSDS, December 2009.

[6] *Spacecraft Onboard Interface Services—Subnetwork Memory Access Service*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 852.0-M-1. Washington, D.C.: CCSDS, December 2009.

[7] *Spacecraft Onboard Interface Services—Subnetwork Device Discovery Service*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 854.0-M-1. Washington, D.C.: CCSDS, December 2009.

[8] *Spacecraft Onboard Interface Services—Subnetwork Synchronisation Service*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 853.0-M-1. Washington, D.C.: CCSDS, December 2009.

[9] *Spacecraft Onboard Interface Services—Subnetwork Test Service*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 855.0-M-1. Washington, D.C.: CCSDS, December 2009.

[10] *Spacecraft Onboard Interface Services—XML Specification for Electronic Data Sheets*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 876.0-B-1. Washington, D.C.: CCSDS, April 2019.

[11] *Spacecraft Onboard Interface Services—Specification for Dictionary of Terms for Electronic Data Sheets*. Issue 2. Draft Recommendation for Space Data System Practices (Red Book), CCSDS 876.1-R-2. Washington, D.C.: CCSDS, June 2016.

[12] *Spacecraft Onboard Interface Services—RFID-Based Inventory Management Systems*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 881.0-M-1. Washington, D.C.: CCSDS, May 2012.

[13] *Spacecraft Onboard Interface Systems—Low Data-Rate Wireless Communications for Spacecraft Monitoring and Control*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 882.0-M-1. Washington, D.C.: CCSDS, May 2013.

[14] *Spacecraft Onboard Interface Services—RFID Tag Encoding Specification*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 881.1-B-1. Washington, D.C.: CCSDS, October 2017.

[15] *Mission Operations Message Abstraction Layer*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 521.0-B-2. Washington, D.C.: CCSDS, March 2013.

[16] *Mission Operations Common Object Model*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 521.1-B-1. Washington, D.C.: CCSDS, February 2014.

[17] *Mission Operations Monitor & Control Services*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 522.1-B-1. Washington, D.C.: CCSDS, October 2017.

[18] *Mission Operations Message Abstraction Layer—JAVA API*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 523.1-M-1. Washington, D.C.: CCSDS, April 2013.

[19] *Mission Operations Message Abstraction Layer—C++ API*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 523.2-M-1. Washington, D.C.: CCSDS, December 2018.

[20] *Mission Operations—MAL Space Packet Transport Binding and Binary Encoding*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 524.1-B-1. Washington, D.C.: CCSDS, August 2015.

[21] *Mission Operations—Message Abstraction Layer Binding to TCP/IP Transport and Split Binary Encoding*. Issue 1. Recommendation for (Blue Book), CCSDS 524.2-B-1. Washington, D.C.: CCSDS, November 2017.

[22] *Mission Operations—Message Abstraction Layer Binding to HTTP Transport and XML Encoding*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 524.3-B-1. Washington, D.C.: CCSDS, June 2018.

[23] *XML Telemetric and Command Exchange—Version 1.2*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 660.0-B-2. Washington, D.C.: CCSDS, February 2020.

[24] *XML Specification for Navigation Data Messages*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 505.0-B-1. Washington, D.C.: CCSDS, December 2010.

[25] *Orbit Data Messages*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 502.0-B-2. Washington, D.C.: CCSDS, November 2009.

[26] *Tracking Data Message*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 503.0-B-1. Washington, D.C.: CCSDS, November 2007.

[27] *Attitude Data Messages*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 504.0-B-1. Washington, D.C.: CCSDS, May 2008.

[28] *Conjunction Data Message*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 508.0-B-1. Washington, D.C.: CCSDS, June 2013.

[29] *Pointing Request Message*. Issue 1. Recommendation for (Blue Book), CCSDS 509.0-B-1. Washington, D.C.: CCSDS, February 2018.

[30] *Reference Model for an Open Archival Information System (OAIS)*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 650.0-M-2. Washington, D.C.: CCSDS, June 2012.

[31] *Producer-Archive Interface Specification (PAIS)*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 651.1-B-1. Washington, D.C.: CCSDS, February 2014.

[32] *Parameter Value Language Specification (CCSD0006 and CCSD0008)*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 641.0-B-2. Washington, D.C.: CCSDS, June 2000.

[33] *The Data Description Language EAST Specification (CCSD0010)*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 644.0-B-3. Washington, D.C.: CCSDS, June 2010.

[34] *Data Entity Dictionary Specification Language (DEDSL)—Abstract Syntax (CCSD0011)*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 647.1-B-1. Washington, D.C.: CCSDS, June 2001.

[35] *Data Entity Dictionary Specification Language (DEDSL)—PVL Syntax (CCSD0012)*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 647.2-B-1. Washington, D.C.: CCSDS, June 2001.

[36] *Data Entity Dictionary Specification Language (DEDSL)—XML/DTD Syntax (CCSD0013)*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 647.3-B-1. Washington, D.C.: CCSDS, January 2002.

[37] *XML Formatted Data Unit (XFDU) Structure and Construction Rules*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 661.0-B-1. Washington, D.C.: CCSDS, September 2008.

[38] *Security Architecture for Space Data Systems*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 351.0-M-1. Washington, D.C.: CCSDS, November 2012.

[39] *CCSDS Cryptographic Algorithms*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 352.0-B-2. Washington, D.C.: CCSDS, August 2019.

[40] *Space Plug-and-Play Architecture Standard—Ontology*. AIAA S-133-7-2013. Reston, Virginia: AIAA, 2013.

[41] *Asynchronous Message Service*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 735.1-B-1. Washington, D.C.: CCSDS, September 2011.

[42] *International Vocabulary of Metrology—Basic and General Concepts and Associated Terms (VIM)*. 3rd ed. JCGM 200:2012. Sèvres, France: BIPM, 2012.

[43] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. 2nd ed. International Standard, ISO/IEC 7498-1:1994. Geneva: ISO, 1994.

[44] *Cross Support Transfer Services—Monitored Data Service*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 922.1-B-1. Washington, D.C.: CCSDS, April 2017.

[45] *Cross Support Transfer Service—Tracking Data Service*. Issue 1. Draft Recommendation for Space Data System Standards (Red Book), CCSDS 922.2-R-1. Washington, D.C.: CCSDS, August 2017.

[46] *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.

[47] *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.

[48] *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.

[49] *Space Data Link Security Protocol—Extended Procedures*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.1-B-1. Washington, D.C.: CCSDS, February 2019.

[50] *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.

[51] *Spacecraft Onboard Interface Services*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 850.0-G-2. Washington, D.C.: CCSDS, December 2013.

[52] *Electronic Data Sheets and Common Dictionary of Terms—Overview and Rationale*. Report Concerning Space Data System Standards (Green Book). Under Development.

[53] *Wireless Network Communications Overview for Space Mission Operations*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 880.0-G-3. Washington, D.C.: CCSDS, May 2017.

[54] *Mission Operations Services Concept*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 520.0-G-3. Washington, D.C.: CCSDS, December 2010.

[55] *Mission Operations—Common Services*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 522.0-B-1. Washington, D.C.: CCSDS, May 2020.

[56] *Mission Operations—Mission Data Product Distribution Services*. Issue 1. Draft Recommendation for Space Data System Standards (Red Book), CCSDS 522.2-R-1. Washington, D.C.: CCSDS, November 2018.

[57] *Mission Operations—Automation Service*. Proposed.

[58] *Mission Operations—Software Management Service*. Proposed.

[59] *Mission Operations—Time Services*. Proposed.

[60] *Mission Operations—Navigation Services*. Proposed.

[61] *Mission Operations—File Transfer and Management Services*. Proposed.

[62] *Mission Planning and Scheduling*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 529.0-G-1. Washington, D.C.: CCSDS, June 2018.

[63] *Mission Planning and Scheduling Services*. Proposed.

[64] *Navigation Events Message*. Proposed.

[65] *Re-entry Data Message*. Issue 1. Recommendation for (Blue Book), CCSDS 508.1-B-1. Washington, D.C.: CCSDS, November 2019.

[66] *Consumer-Archive Interface Specification*. Proposed.

[67] *Electronic Data Sheets and Dictionary of Terms for Onboard Devices and Components*. Report Concerning Space Data System Standards (Green Book). Under Development.

[68] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460. Reston, Virginia: ISOC, December 1998.

[69] "Appendix B—Classification Considerations for NASA Class A-D Payloads." In *Risk Classification for NASA Payloads*. Updated w/change 3. NPR 8705.4. Washington, DC: NASA, 2004.

[70] *Space Engineering—Test and Operations Procedure Language*. ECSS-E-ST-70-32C. Noordwijk, The Netherlands: ECSS Secretariat, 31 July 2008.

[71] *Security Threats against Space Missions*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 350.1-G-2. Washington, D.C.: CCSDS, December 2015.

[72] *System Engineering – Time Management Standards.* Proposed.

[73] *Mission Operations – Message Abstraction Layer to ZMTP Transport*. Recommendation for Space Data System Standards (Blue Book), CCSDS 524.4-B-1, Washington, D.C.: CCSDS, Oct 2019.

2 APPLICATION AND SUPPORT LAYER ARCHITECTURE CONCEPTS

2.1 BACKGROUND

Founded in 1982 by the major space agencies of the world, the CCSDS is a multinational forum for the development of communications and data systems standards for spaceflight, with the goal of enhancing governmental and commercial interoperability and cross support, while also reducing risk, development time, and project costs. Within CCSDS there are two sets of standards focused on application and support layer interoperability, specifically:

– MO standards, functions, and services, covering the interfaces between the ground mission control, planning and scheduling systems, and the spacecraft; over time some of these functions are being migrated onboard;

– SO standards, functions, and services, covering interfaces between the spacecraft and onboard functions, subnets and communications, and interfaces between electronic devices and applications.

For both groups of functions, the information they exchange, the data formats, syntax, semantics, and information content are important, and the data items exposed at application-level interfaces may reference other data items at the same level. The service definitions provide exposed interfaces and define the functions that produce these information objects, specification of the interactions between communicating entities, and how they may be assembled to define deployed applications. What these service interfaces, and the applications communications protocol stacks, look like must be specified for communicating a data format for file exchange or an interactive service based on message exchange. Underlying communications protocol stacks may either be as already defined in the Architecture Requirements Document (SCCS-ARD) (reference [2]) for space links or make use of alternative terrestrial or space-based Network or Data Link Layer technologies, depending on the deployment context.

This document models the mission operations aspects of a space system as a set of reference functions; identifies where the interactions between those functions create an interoperable boundary between agencies, organizations, or systems; maps these to existing or planned CCSDS Recommended Standards; describes the protocol stacks; identifies a set of representative deployment options; and identifies any key gaps in coverage.

2.2 APPLICATION SUPPORT LAYER DOMAINS

2.2.1 GENERAL

The flight domain is primarily addressed by the SO standards. The mission operations domain is primarily addressed by the MO standards and these are mostly located within the ground segment. The flight and ground domains are connected and secured using underlying data transport and other services provided by the other CCSDS communications, cross support, and systems standards. An aspect that is explored is a set of different cases describing the interaction of MO services on the ground with the onboard environment, and for the integration and potential transition of MO functions and services into the onboard environment. The basic case looks much like typical current deployments, with a straightforward command and telemetry connection from space to ground. The most extended case, with MO services onboard, is expected to primarily be of benefit to large, multi-agency, and relatively resource-rich deployments.

2.2.2 MISSION OPERATIONS AND INFORMATION MANAGEMENT

The MO standards are concerned with the Application Layer functionality required to perform mission operations and to manage mission information (both mission data products and configuration data), and specifically the interactions between its principal functions that may be exposed to interfaces between interoperating agencies, other institutions, or systems.

The MO standards cover interfaces associated with the following functional domains:

- Spacecraft (and Mission) Monitoring and Control;
- (Spacecraft) Navigation;
- Mission Planning and Scheduling;
- Data Archives.

MO standards do not address all Application Layer functionality of a typical space data system. In particular, they do not address the processing of mission payload or science data or typical science system functions.

The MO functional domains are often distributed across multiple agencies, geographical sites, or systems, exposing their interactions at interoperable interfaces. Traditionally, these functions have been deployed primarily within the ground segment of a space data system, but increasingly, certain MO functions are migrating onboard spacecraft, exposing application interactions across the space link.

The CCSDS communications, cross support, and systems standards are principally concerned with communications-layer protocols that provide services for how an information collection (message, file, or packet) is transferred to its destination, how large terrestrial space communication assets are to be planned and utilized, and in how to architect and secure these widely distributed systems. In contrast, MO standards are principally concerned with the format and meaning of the contained information that is exchanged between Application Layer functions. Two approaches have been taken in the standardization of MO information:

- specification of data formats that can be used for information exchange: only the data format is standardized;

– specification of a service that can be used for both information exchange and control: both the pattern of interaction between service provider and consumer and the format of exchanged information is standardized.

In the case of Navigation functions, interoperable data formats have been defined for the exchange of common Navigation data items, such as Orbit, Attitude, and Tracking Data.

In the case of Spacecraft Monitoring and Control (SM&C) functions, an abstract service framework has been developed that allows multiple MO services to be defined, and to allow mapping onto a variety of underlying communications architectures.

One of the challenges for MO standardization is that existing software solutions, whether commercial products or institutional frameworks, have their own internal architectures for inter-process communications, employing a range of data transfer technologies. This makes it easier to adopt a standard that just addresses data formats, but leaves the behavioral aspects of the interface or the functions unspecified. The MO services architecture has approached this differently, by defining a framework, consisting of a Message Abstraction Layer (MAL) and Common Object Model (COM), with multiple possible mappings to both underlying technologies (data representations and transfer protocols) and programming languages. This offers flexibility and adaptability as underlying technologies evolve and can enable bridging of application-level services between different communications infrastructures.

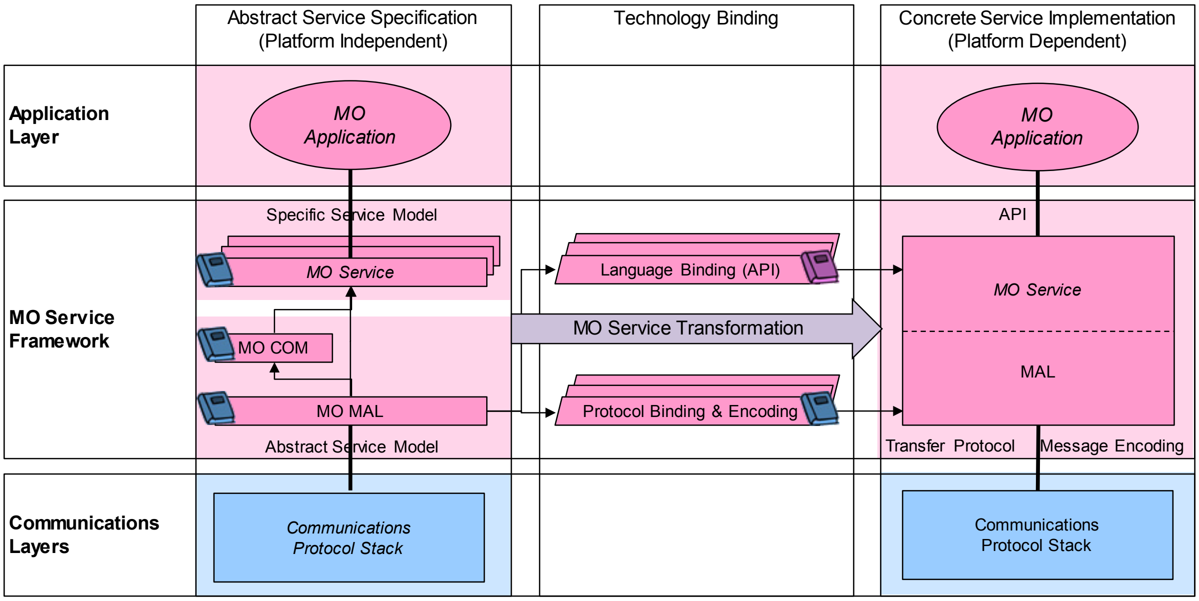


Figure 2-1: Mission Operations Services Framework

Each functional MO service represents a set of interactions relating to particular types of information. The information objects for the service extend the COM, while the service operations (invoked by the consumer application and executed by the provider application) are expressed in terms of generic message interaction patterns defined in the MAL and actualized using the protocol binding and encoding process.

Language bindings define how the abstract service specifications for any MO service can be transformed into a language-specific API.

Technology bindings define how the service messages are to be encoded and mapped to a technology-specific protocol for transfer across a service interface using a concrete communications protocol stack. To achieve interoperability, the same communications protocol stack and information bindings must be used on all instances of the interface. However, it is possible to use different language-specific APIs for each implementation of the interface.

The underlying communications protocol stack varies for different communications contexts, which may either be entirely terrestrial or use space links. When a communications link is only terrestrial, commonly used Internet protocols and links are typically employed to provide the underlying data transfer services. However, when communications involve a space link, use of one of the CCSDS-based protocol stacks defined in the SCCS-ADD is assumed, based on CCSDS communications, cross support, and systems standards. This is described in more detail in 7.3, but the diagram below shows a simplified case for a typical deployment, in which a spacecraft and terrestrial Mission Operations Center (MOC) belonging to Agency A, communicate via a Telemetry, Tracking, and Command (TT&C) ground station operated by Agency B, hence the 'ABA' communications deployment pattern.

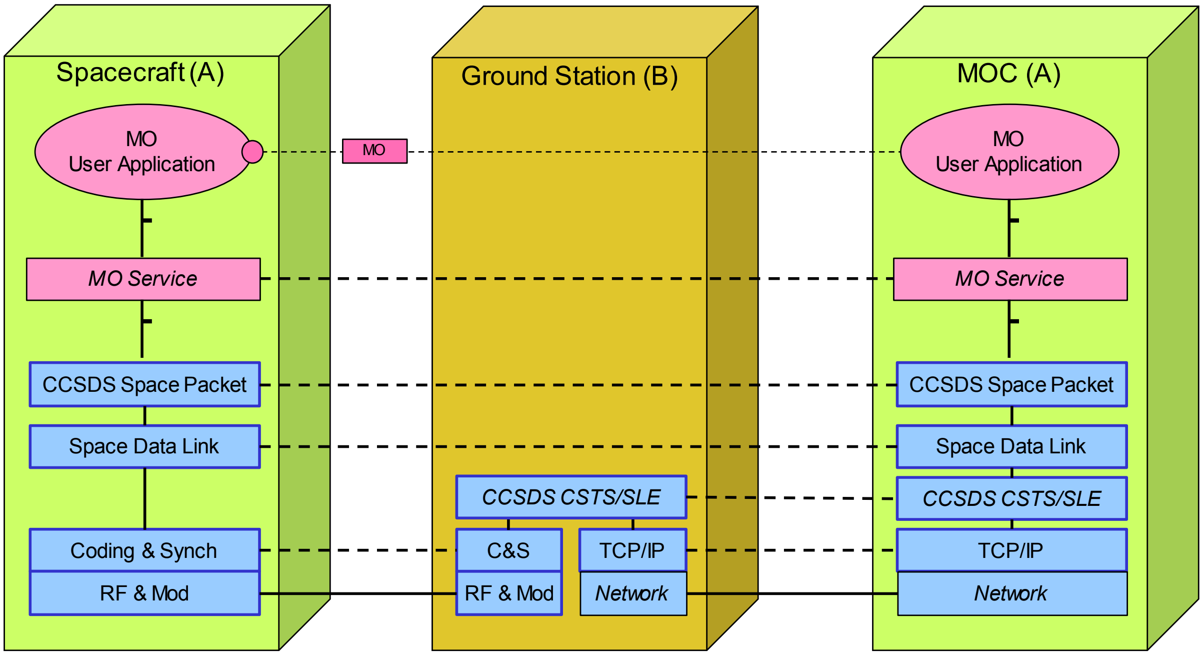


Figure 2-2: Simplified View of an ABA Space Link Deployment

CCSDS SLS standards define the space link interface between spacecraft and ground station; and CSS Space Link Extension (SLE) and Cross Support Transfer Service (CSTS) standards define how CCSDS Telemetry, Telecommand, Tracking, and monitoring interfaces are extended to the MOC. While the MOC receives Telemetry (TM) from and sends commands (TC)[1] to the Ground Station, the source and sink of TM and TC frames and space data packets (reference [3]) is actually the Spacecraft. The MO application-layer messages are encoded as data within the body of the space data packets. The SLE and CSTS cross-support protocols effectively provide a communications channel between Spacecraft and MOC that passes through the Ground Station.

In the MO functional model, the focus is on the exchange of information between mission operations Application Layer functions. This corresponds to the MO service level interaction shown at the top of the diagram (between items colored pink). This interaction may be enacted through TM/TC and SLE interfaces as shown above, or, in a terrestrial context, service-level messages may be exchanged directly across IP-based or Delay Tolerant Networking (DTN) networks. Functionality within the communications layers (colored blue) is intentionally not represented within the MO Area functional model.

Furthermore, Ground Station functionality is only represented when it is the source (or sink) of Application Layer data that MO functions operate upon. Specifically:

- Telemetry and Telecommand processing is not shown, as this is viewed as a communications-layer function, and the Ground Station is treated as a part of the communications path, but is neither the source of the telemetry nor the target of the telecommands.
- TT&C Ground Station networks do, however, contain Application Layer functions that directly interact with Mission Operations:
  - Spacecraft Tracking and Ranging (radiometric tracking and Delta Differential One-way Ranging [Delta-DOR]);
  - Recording Ground Reception Time (to support onboard clock correlation);
  - Planning and Scheduling of Ground Station service contacts with the spacecraft;
  - Monitoring and Control of the Ground Station services, including its configuration and status during support contacts with the spacecraft.

At time of writing, there was discussion on potential update to the MO service framework, with a focus on simplifying the COM and extending the set of data types supported by the MAL. The objective of this would be to remove implied internal architectural or implementation constraints on an MO-compliant system and to simplify the specification of MO services. This document describes the published MO service framework.

[1] The terms ‘TC’ and ‘TM’ are used as a convenient shorthand, but these space communication links may use any of the existing CCSDS space data link protocols, including TC, TM, AOS, USLP, or Proximity-1. Deployments may also use network protocols such as DTN or IP and Application Layer data transfer protocols such as CFDP or AMS.

2.2.3 SPACECRAFT ONBOARD INTERFACES

The SO standard services provide support for development activities intended to radically improve the spacecraft flight segment data systems design and development process. It defines generic services that will simplify the way flight software interacts with flight hardware and permits interoperability and reusability.

The SO standard interfaces are associated with the following functional domains:

- Application Support Services;
- Subnetwork Services;
- Onboard Wireless Services.

SO standards cover more than just application and subnetwork support services. Most platforms provide Application Programming Interface (API) calls for communication with services provided by the platform. Some platforms provide a software message bus for inter-process message exchange. Other platforms may be Time and Space Partitioned (TSP), with messages passed between partitions using features that may be provided by an RT operating system. SO standards recognize that the space environment is a challenging one and that there are a significant variety of real-time, fault-tolerant, architectures in use using different RT operating systems. There is no attempt to standardize these environments, but to develop features in the SO standards that allow all these different approaches to be used as needed.

Because of the degree of variability in real-time platform architectures that are typical in a resource-constrained space environment, SO standards would have difficulty in meeting the CCSDS goal of interoperability if they were just to present static recommendations. Forcing adherence to a single static recommendation would likely be thought of as a restriction of innovation and probably an unnecessary cost to implement. Accordingly, the SO standards accommodate the evolution of agency flight systems in a way that facilitates interoperability and portability across platforms.

- The SEDS (reference [10]) is an extensible and adaptable framework to describe components and deployments that use an extensible Dictionary of Terms (DoT) (reference [11]) that may be adopted and extended as needed.
- The application support service interfaces identified by the SOIS Green Book (reference [51]) is described in one or more SEDS files.
- Agencies can develop implementations of application support services and describe their interfaces in one or more SEDS, using the standard and/or extended DoT.
- Each agency's tool chain can adapt their implementation to the agency's platform through at least three mechanisms:
  - The tool chain generates device access services from device SEDS & DoT.

    - The tool chain adapts application services to communicate in the style of the platform, which may include use of an API or a message bus service, such as AMS.

    - The tool chain adapts application services to mission-specific design parameters and constraints specified in SEDS.

- There are two forms of abstraction for the interface between onboard subnetworks and the application support services:

    - Five SO Magenta Books define abstract service access points through which application support services may find a uniform interface for any subnetwork technology (references [5], [6], [7], [8], and [9]). An agency's tool chain uses these abstractions to generate device access services that directly use the concrete agency-specific subnetwork software.

    - The SOIS Green Book (reference [51]) defines a set of abstract convergence functions that can be inserted between the agency-specific subnetwork software and application support services to provide uniform qualities of service across a variety of subnetwork technologies.

- There are recommended standards and practices for the use of onboard wireless subnetworks:

    - One application of wireless technology is in tracking onboard items tagged with RFID. There is a recommended standard for tag encoding and a Recommended Practice for sensing those tags (references [12] and [14]).

    - Another application of wireless technology is a Recommended Practice for replacement of typical massive wiring harnesses by onboard sensors and actuators connected by a wireless subnetwork (reference [13]).

2.3 RELATIONSHIP WITH OTHER CCSDS ARCHITECTURE DOCUMENTS AND STANDARDS

The ASL architecture described in this document uses underlying communications services and data exchange standards provided by CCSDS. The ASL data exchanges and services are intended to be used over these underlying space data link and networking services, but they may also be deployed over standard terrestrial networks (ground to ground), or a variety of different, bespoke, onboard communications architectures.

The MO standards relate to the exchange of information or services between Application Layer functions, primarily on the ground but also, increasingly, onboard, and the SO services and standards relate to the exchange of information and services among processors and other devices in the onboard environment. These are referenced as elements within the Functional, Information, and Service Viewpoints of the ASL Reference Architecture described in this document. How these elements relate to specific CCSDS books and standards is defined within the Service Viewpoint tables in section 6.

2.4 ASL ARCHITECTURE: ASSUMPTIONS, GOALS, AND CHALLENGES

A key goal is to describe how the various MO and SO capabilities may be used and integrated to provide functioning systems. A fundamental assumption is that many of these ASL services live on top of underlying CCSDS data transfer protocols and services (and terrestrial ones as needed).

A secondary goal is to describe how and where the MO functions and even the service framework could be deployed in space as well as on the ground.

The two areas of standardization addressed in this book have separate foci. SO is focused on the spacecraft onboard environment, while MO services are focused on mission operations that are today primarily carried out on the ground. While it is necessary to describe how the MO services interface with the spacecraft environment, there is the possibility of migrating some of the MO application-layer functions and services into the onboard environment. One aspect that is to be explored is that MO services may be implemented and deployed in different ways, ranging from an implementation that is strictly terrestrial to one that is extended into the space environment.

Three possible cases of MO function and service migration onboard have been identified and is described further in section 9, Deployment Viewpoints. In all of these cases, the core spacecraft, onboard, real-time, fault-tolerant, environment is expected to be provided by the same kinds of real-time deployments that are in use today, or their natural evolution. Software that runs in these environments is expected to comply with Class A/B requirements (reference [69]). In some cases, application software that is run in a TSP operating system, or in a separate processor, may not need to meet these stringent requirements.

> Case 1: Traditional case with SO supplying the interfaces between spacecraft and onboard devices while MO functions and services are only on ground. Interfaces between flight and ground are 'traditional' TT&C with mapping from MO functions to TT&C done on the ground.
>
> Case 2: Intermediate integrated case with MO service interfaces exposed by the spacecraft at its interface to the ground, extending across the space link, utilizing space data packets to transfer its messages. This may be supported by a Proxy interface used onboard to map internally to traditional TT&C, or alternatively, to onboard Application Layer functions that communicate directly using MO service interfaces to the ground and SO services to access onboard devices. In both cases, SO functions continue to supply the same interfaces onboard as before.
>
> Case 3: Onboard integrated case with MO based services and frameworks migrated onboard and adapted to the onboard environment. In this case, some onboard mission operations applications may use MO services to communicate among each other, but these would still use SO services and functions to access onboard devices. The same SO spacecraft and onboard subnetwork services are used, along with the same real-time spacecraft operational services and functions. Some onboard devices could even integrate directly with the MO service framework.

This may be summarized as:

1. MO functions and services deployed only within the Ground Segment.

2. MO functions and services deployed across the space link to the Spacecraft, but not within the Spacecraft.

3. MO functions and services deployed across the space link and onboard the Spacecraft to support communication between certain mission operations onboard functions.

There are, of course, many different possible deployments, but these three cases provide adequate coverage of the option space to permit evaluation of the potential benefits and limitations.

Another goal is to explore the potential for use of the extensible SO SEDS and DoT approaches for describing components, their interfaces, and configuration into deployments in space and ground environments and in support of operations preparation. Many of the needed constructs are already available within these standards, but all of the necessary extensions have yet to be explored and documented.

3 APPLICATION AND SUPPORT LAYER REFERENCE ARCHITECTURE

3.1 INTRODUCTION

This section provides an overview of how the CCSDS ASL Reference Architecture is presented in the remainder of this document.

The ASL Reference Architecture has been modelled using the methodology defined in the CCSDS Reference Architecture for Space Data Systems (RASDS) (reference [1]), modified and extended to enable a concise representation of the ASL from seven modelling viewpoints:

a) Functional;

b) Information;

c) Services;

d) Communications (protocol stack);

e) Physical (deployment nodes and connectivity);

f) Deployment;

g) Implementation.

These viewpoints are described in 3.2 below. The following sections (4 to 10) of this document each describe the ASL Reference Architecture from one of these seven viewpoints in turn. Each of these subsections contains sections addressing the following topics:

- overview of the viewpoint;
- MO services aspects;
- SO services aspects;
- security concepts;
- additional topics specific to the viewpoint.

RASDS provides a conceptual framework for describing system architectures and a simple notation for the representation of Enterprise, Functional, Communications, Physical Deployment, and Information Viewpoints of a space-domain reference architecture. Specific extensions to RASDS have been used to improve the representation of service interfaces and the data that are exchanged in the Functional Viewpoint, and a tabular presentation is used to identify end-to-end services. The limited representational methods that RASDS used for the Information Viewpoint have been extended through the use of simplified UML class diagrams to represent the associated information models. A summary of all of the graphical notations used for the representation of the ASL Reference Architecture in this document is given in 3.3.

The ASL Reference Architecture is focused on the representation of the ASL, rather than on the lower layers of the communications protocol stack. When communications are deployed over space links, the Communications Viewpoint builds upon the RASDS models contained in the CCSDS SCCS-ARD (reference [2]) that define the services, information models, communication protocol stacks, and Deployment Viewpoints for point-to-point 'ABA' space links and Space System Internet (SSI) deployment cases. The focus in that document is on interoperable and cross-supported communications, end-to-end, and the applications details are typically abstracted away.

From an ASL perspective, communications are also employed within a spacecraft and over terrestrial networks. The onboard environment is often highly resource constrained and typically requires specialized Data Link Layer protocols and real-time services, while in the terrestrial context, widespread, general purpose, Internet, and Data Link Layer protocols are used. More detail on these deployment use cases from a communications perspective is given in 7.3.

The kinds of service level agreements and access management arrangements that might be needed in multi-mission and multi-agency cross-support and interoperability environments is discussed in 9.4.

Strategies that might be employed in the transition from ABA to SSI style deployments and from ground-only MO services to those that may be deployed in flight, together with the issues that may be encountered are discussed in 2.4 and 9.5.

3.2 SEVEN VIEWS OF SYSTEM ARCHITECTURE

3.2.1 OVERVIEW

This subsection provides a brief description of each of the views used to describe the ASL Reference Architecture, explaining their scope and what each shows.

3.2.2 FUNCTIONAL VIEWPOINT

The Functional Viewpoint describes the functional composition of a space data system, its interfaces, actions, and behaviors, and how functions interact with each other. The Functional Viewpoint is used to address these abstract functional aspects of space data systems. A Functional Object may be composed of other Functional Objects. A formal or informal group of Functional Objects that provides some services in a space data system, such as a related set of navigation or data processing services, may be modeled as a higher-level element in a Functional View.

The functional scope of the ASL model is broken down hierarchically into a set of functions corresponding to recognizable areas of functionality within space systems, which are often associated with a particular type of information. While functions may correspond to identifiable subsystems, they may also be arbitrarily grouped within existing systems.

The ASL Reference Architecture Functional Viewpoint corresponds to core functionality (sometimes referred to as ‘business logic’) at the Application Layer and not to supporting functions at lower communications layers. For this reason, functions such as TT&C only appear when they are the source or sink of Application Layer information, and not when they have a role in transporting telemetry and telecommands between spacecraft and ground-segment facilities.

CCSDS usually standardizes the interfaces between functions, rather than the functions themselves. For these purposes, a functional decomposition is only needed to a level that identifies those interfaces potentially exposed as interoperability points between organizations, physical locations, various communications capabilities, or systems.

This viewpoint shows the interfaces between functions in terms of the information exchanged; and the services used to manage that information exchange, or to allow one function to control the processing performed by another. It is concerned with application-level information and not with the communications-layer data formats (such as telemetry packets, frames, files, or network traffic) that are addressed in the SCCS-ADD and ARD.

The information (or data) associated with an interface between functions is identified on the interface and is specified as an information object defined in the Information Viewpoint.

Each interface may also expose a service that is defined in the Service Viewpoint. The representation of the interface identifies which function acts as the provider of the service, and which the consumer. This is a different concept from data flow in that the information objects associated with the interface do not necessarily flow only from provider to consumer. This can be illustrated through the real-world example of banking services: the banks are the providers of services, the householder the consumer; deposits flow from consumer to provider, while withdrawals flow from provider to consumer (or often to other payees).

It is important to note that the interfaces between functions in the Functional Viewpoint are logical interfaces corresponding to the end-to-end interactions at application level between functions, wherever they are deployed. In an actual deployment, a logical interface between two nominally ‘adjacent’ functions may actually be transferred through lower-level communications functions in multiple deployment nodes. This is not shown in the Functional Viewpoint, but is represented in the Communications, Physical, and Deployment Viewpoints.

The Functional Viewpoint shows interfaces between different functions, but does not show the interfaces between similar functions that may exist if a function is itself distributed (whether for redundancy or other reasons) across several deployment nodes. This is represented in the Deployment Viewpoint.

Two formulations of the Functional Viewpoint diagrams are provided. The standard diagram is function-oriented and shows functions connected by logical interfaces. These are contained in the body of the document. A set of alternative diagrams that more clearly show all of the functions that share a common interface type are provided in Annex B. These show color-coded horizontal lines corresponding to the services, with vertical lines connecting to the functions that provide or consume each service. The diagrams are topologically equivalent, but show the interrelationships from different perspectives.

3.2.3 INFORMATION VIEWPOINT

The Information Viewpoint of space data systems describes how data objects and their structures, relationships, metadata, and constraints are defined and configured within the system.

The Information Viewpoint looks at space data systems from the perspective of the Information Objects that are defined and managed. It includes descriptions of Information Objects (their structure and syntax), information about the meaning and use of these Objects (contents and semantics), the relationships among Information Objects (the data model), rules that define constraints on their use transformation and retention, and policies on access.

The ASL Information Viewpoint, models the principal information objects and the relationships between them, including:

- inheritance;
- composition;
- aggregation;
- other associations.

The purpose of the information model within the ASL Reference Architecture is to identify the types of information objects exchanged across individual interfaces or services that are the subject of specific CCSDS Recommended Standards, but not to define them in detail. There is no attempt to expand the structure of these information objects in terms of their subordinate objects or attributes; this level of detail is contained in the individual standards that define them.

At the higher levels of the functional decomposition hierarchy, information objects may be aggregated to a more abstract representation of the information relating to a wider functional area.

The information model uses a simplified UML class diagram to show the relationships between information objects.

3.2.4 SERVICE VIEWPOINT

The Service Viewpoint defines service interfaces (space and ground), methods for using services, and behavior for planning, scheduling, configuring, requesting, using, and reporting on services. The Service Viewpoint also addresses the means for locating and binding to services at Service Access Points (SAPs), including the roles that protocols play in the binding process. These service interfaces and binding methods are a primary compliance point for any service architecture.

The ASL Service Viewpoint identifies standard application-level interfaces between functions. A standardized service is specified in terms of the information exchanged and/or the behavior of the interface, rather than to a particular specific pair of interfaced functions. The same service specification can be used to support multiple interfaces between different interfaced functions, providing they relate to the same type(s) of information and share a common interface behavior.

The information exchanged using a service corresponds to one or more information objects in the Information Viewpoint.

The behavior of a service corresponds to the pattern of interaction between functions across an interface. This may be to support the exchange of information objects or the control of a function. These service interactions may be implemented as simple exchange of data, typically as a file transfer, or by a more complex interaction involving message exchanges between service consumer and provider functions. It should be noted that when messages are exchanged across a space link, there may be significant delays.

An interactive service specification defines the set of operations that the service consumer can invoke on the service provider and the bidirectional pattern of message exchanges required to achieve this. Bidirectional message exchanges across a space link may be particularly challenging and must be carefully engineered because of light-time delays. For the ASL Reference Architecture, this is limited to identification of high-level capabilities (groups of operations) supported by the interface.

Not all interfaces are supported by an interactive service standard. In some cases, only the format of the exchanged information object is currently standardized by CCSDS. These information objects may be exchanged by a variety of means, including standard file transfers and bespoke interfaces.

The Service Viewpoint identifies the services and for each service:

- the functions that act as provider or consumer of the service;
- the information objects that are exchanged across the service interface;
- the behavior exposed at the interface, in terms of the high-level capabilities or operations supported by the service;
- references to CCSDS Recommended Standards defining the service and/or associated data formats.

3.2.5 COMMUNICATIONS VIEWPOINT

The Communications Viewpoint is used to describe the layered sets of communications protocols that support communications among the nodes in the system. These protocols must meet the requirements imposed by the actual physical connectivity in the deployed system and the operational challenges of communicating in space. The communications protocol data units are sometimes called the ‘wire protocol’. The Communications Viewpoint is used to address these protocol aspects of space data systems.

The Communications Viewpoint Specification describes the mechanisms that support information transfer among system elements (i.e., hardware or software) in a space data system. These protocol ‘stacks’ are directly associated with the physical links that exist between the nodes of a system, and they have responsibility for transporting data across these links.

The ASL Communications Viewpoint shows how the application-level services are supported by underlying communications protocol stacks, depending on their deployment context. This viewpoint should be read in conjunction with the SCCS-ADD, which identifies two principal deployment contexts for space links (ABA and SSI) together with their associated communications protocol stacks.

The ASL Reference Architecture builds upon the SCCS-ADD by identifying three contexts for the deployment of Application Layer services:

– **Terrestrial Link**: a ground to ground communications channel;

– **Space Link**: a space-to/from-ground or space-to-space communications channel based on the ABA and SSI deployment cases defined in the SCCS-ADD;

– **Onboard Link**: a communications channel supported between functions onboard the same spacecraft.

Within each context, two principal approaches for service deployment are considered:

– service interaction using bidirectional message exchanges or streaming message transfers;

– file transfer.

For each context and deployment approach, the Communications Viewpoint provides a representative layered communications protocol stack. In the case of the Space Link context, this is by reference to the CCSDS recommendations defined in the SCCS-ADD. In the case of the Terrestrial Link context, this assumes use of typical internet and Data Link Layer protocols, although alternative protocol stacks could also be used. For the Onboard Link context, this assumes the use of a SO onboard architecture, as described in this document.

The same service specification at the Application Layer may utilize several different protocol stack deployments. The use of communications gateway functions or protocol bridges may allow different communication protocol stacks to be used for different segments of the same end-to-end service interface. This may be thought of as Application Layer protocol bridging in a similar way to IP or DTN providing a common Network Layer across different underlying Data Link Layer interfaces. Managing the ‘impedance matching’ at such Application Layer bridges may require significant design effort if end-to-end interoperability and timing is to be preserved.

3.2.6 PHYSICAL (CONNECTIVITY) VIEWPOINT

The Physical Viewpoint describes how space data systems are composed of physical elements that must operate in space, and the connections between elements, the physics of motion, and external environmental forces that must be considered. The primary elements in the Physical Viewpoint are nodes, links, and deployed components. A node is an abstract model of a physical entity that is used in a space data system and is operating in a physical environment. A node is a configuration of component forming a single unit for the purpose of location in space and embodying a set of processing, storage, and communication functions. A node represents a system entity (such as a spacecraft, a tracking system, or a control system) or an individual physical entity of a system (such as an instrument, a computer, or a piece of communications equipment). A node may be composed of other nodes.

A link is a physical connection between or among nodes. A link represents an RF or optical link, a hardwired link, or a network of some kind (such as the Internet, a LAN, or a bus).

The Physical Viewpoint identifies potential deployment nodes and the types of communication interfaces supported between them.

This document extends the limited set of deployment node classes identified in the SCCS-ADD to cater to representative ground segment scenarios by identifying a realistic set of example ground segment deployment nodes. All of these identified ground segment nodes are purely examples for use in the presentation of representative deployment scenarios in the Deployment Viewpoint. Other types of deployment nodes, with different sets of deployed functions, may exist in actual mission deployments, and nodes with these same behaviors may appear with different names.

This viewpoint shows the example set of deployment nodes together with physical communications links between the nodes representing a range of physical deployment architectures, including both ABA and SSI space links, onboard links, and terrestrial networks. Each of the physical communications links between nodes in the diagrams representing these physical deployment architectures is classified in terms of the deployment contexts identified in the Communications Viewpoint as a Space Link, an Onboard Link, or a Terrestrial Link (together with their associated communications protocol stacks).

The physical deployment architectures shown are only representative examples, and many other physical deployment architectures may exist in actual mission deployments.

3.2.7 DEPLOYMENT VIEWPOINT

The Deployment Viewpoint describes how a set of physical deployment nodes, defined in the Physical Viewpoint, may be connected to construct a set of representative mission operations configurations. This viewpoint also shows how different sets of functions from the Functional Viewpoint may be deployed across these nodes. This viewpoint also shows the resultant application-level interactions between functions in terms of services (Service

Viewpoint) and information objects (Information Viewpoint) exposed to the potential interoperability boundaries between deployment nodes. Functional interactions between functions co-located on the same node are omitted for clarity.

The selected set of deployment scenarios are just examples. The objective is to show how the services identified in the Service Viewpoint may be allocated to different systems and thereby expose interoperability interfaces between systems (and potentially organizations) in realistic mission deployments.

It is stressed that these are only example deployment scenarios, provided to cover a usefully broad set of options, and actual missions may have a wide variety of alternative architectures.

3.2.8 IMPLEMENTATION VIEWPOINT

In most of the CCSDS Recommended Standards, the focus is on defining a single protocol or information structure that may be directly implemented to provide interoperability and cross support. The focus is on clear specification for interoperability and cross support, and not on implementation details. Some of the standards covered in this document have those same properties, but others have specific properties that are intended more for portability or implementation convenience.

The Implementation Viewpoint provides a way to showcase these features so as to make them more visible. The objective is to show how some of these standards, such as the MAL or the SEDS, may be used to transform electronically rendered specifications of interfaces and behaviors into operational code, or at least into compilable interface specifications.

The objective of the Implementation Viewpoint is to describe how the ASL specifications may be extracted and used in specialized tool chains to transform specifications and adaptation rules into executable code, or at least into code fragments that may then be used to develop Application Layer functionality. These standards provide features that a variety of different toolchains might leverage, but the standards do not specify either the toolchains or these implementation processes. This viewpoint consists largely of examples of how these features of the standards may be employed to assist in their implementation, but they are only examples.

3.3 GRAPHICAL CONVENTIONS

3.3.1 GENERAL

CCSDS RASDS (reference [1]) defines a graphical notation for use in the modelling of space data systems. The following figure summarizes the RASDS notation, the additional color coding used within the SCCS-ADD, and the ASL extensions for the service interfaces.

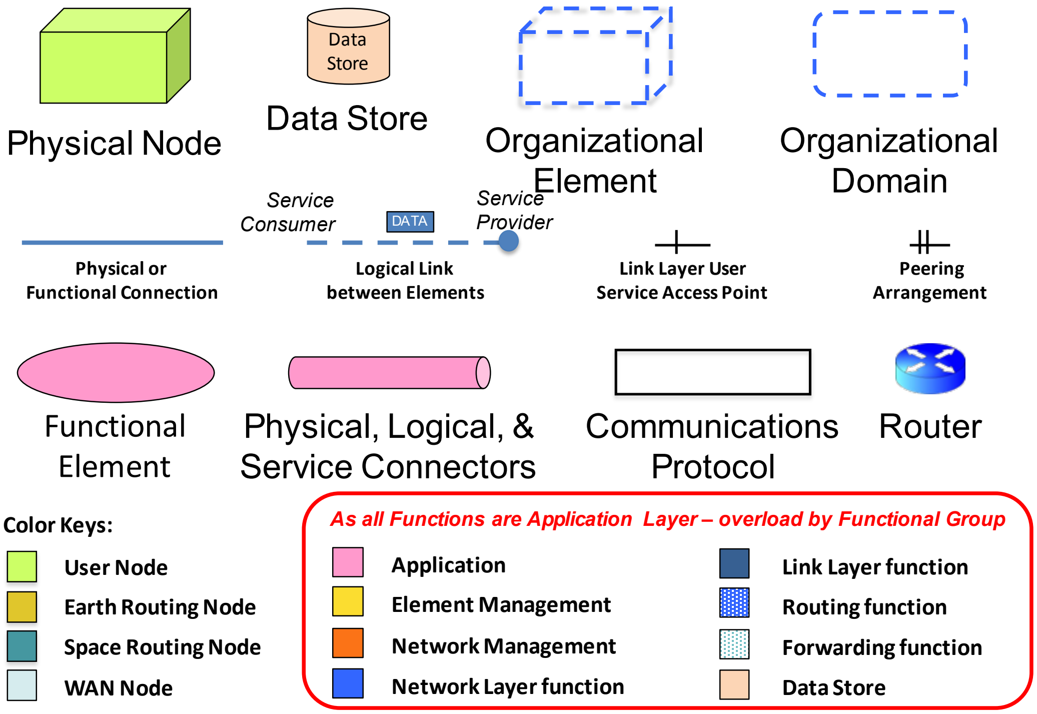


Figure 3-1: Adapted RASDS Graphical Notation as Used in SCCS-ADD

Since this document builds upon the communications architecture described in the SCCS-ADD, this common notation has been used, and this has been extended to enable representation of both the information model and services that are central to the ASL.

Distinct from the SCCS-ADD, this document is primarily concerned with Application Layer functions, it is not useful in most modelling views (Functional, Information and Service) to use color to represent different communications-layer functions. Color in these views is instead used to represent different high-level functional groups within the Application Layer.

The following diagram shows the specific MO application-layer color coding used to distinguish different functional groups within these diagrams.

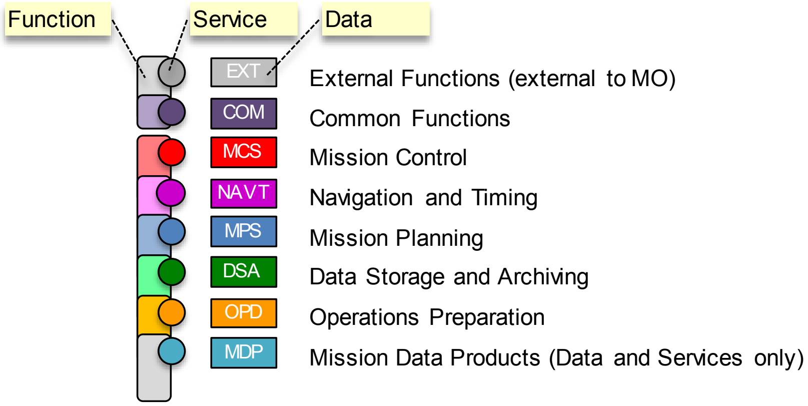


Figure 3-2: Color Coding of MO Functional Groups

For MO Communications Viewpoint diagrams, the standard pink color is used to represent Application Layer functions, while all functions of the communications layers are shown in light blue.

The following diagram shows the color coding used to distinguish different functional areas within the SO function diagrams.

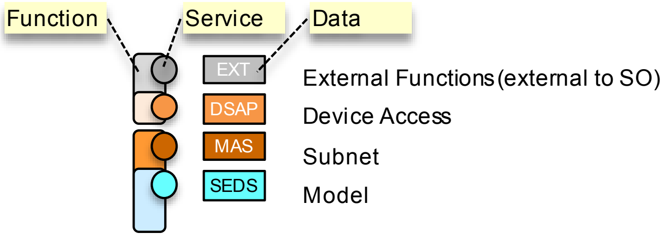


Figure 3-3: Color Coding of SO Functional Groups

The following subsections summarize the notation used for each of the modelling views in this document. It should be noted that in these example diagrams, where functions and data are generic and not specific to any of the functional groups detailed above, a neutral color is used to represent a function, service, or data that may belong to any functional group.

3.3.2 FUNCTIONAL VIEWPOINT NOTATION

The Functional Viewpoint diagrams show functions (functional elements) and the associated logical links between them. At the Application Layer, these logical links represent either simple file-based information exchange between functions or application-level service interfaces that may be interactive or delayed due to the distance between deployments. The underlying communications functions and links are not shown in this viewpoint, as the diagrams are independent of deployment architecture.

The basic RASDS notation for Functional Viewpoint is used and has been extended to allow the explicit representation of service providers and exchanged information (or data).

Information objects exchanged across a logical interface are shown as rectangular boxes attached to the connector, annotated with their abbreviated names. Conventions detailed in the diagram below enable differentiation between information objects that are specific or generic and those contained within another information object.

To enable differentiation between the provider and consumer of a service, the provider end is annotated with a small circle. When standardization of the interface concerned is limited to the format of the exchanged information object, then the 'provider' corresponds to the function that provides that service or that generates that information.

The current standardization status for specific information objects and services is also encoded in the border style of the corresponding shape, as detailed in the diagram. This correlates to the marking of standards in the document text as published, [Future] or [Prospective], as introduced in 1.3.

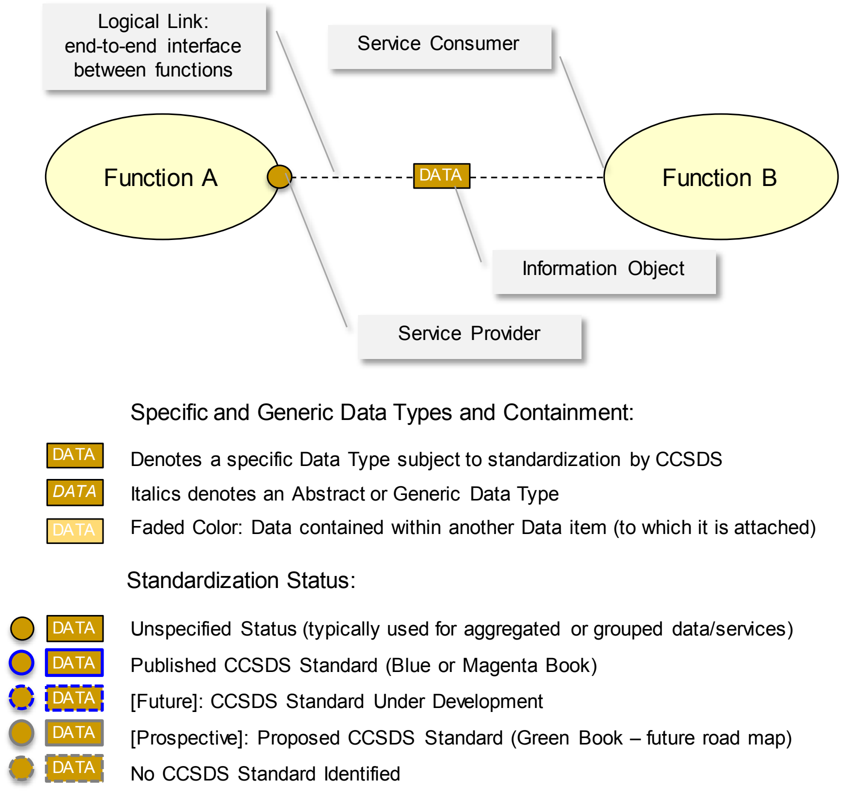

Logical Link:
end-to-end interface
between functions
Service Consumer
Function A
DATA
Function B
Information Object
Service Provider
Specific and Generic Data Types and Containment:
DATA
Denotes a specific Data Type subject to standardization by CCSDS
DATA
Italics denotes an Abstract or Generic Data Type
DATA
Faded Color: Data contained within another Data item (to which it is attached)
Standardization Status:
DATA
Unspecified Status (typically used for aggregated or grouped data/services)
DATA
Published CCSDS Standard (Blue or Magenta Book)
DATA
[Future]: CCSDS Standard Under Development
DATA
[Prospective]: Proposed CCSDS Standard (Green Book – future road map)
DATA
No CCSDS Standard Identified

Figure 3-4: Graphical Notation for Functional Viewpoint Diagrams

In the SCCS-ADD, color coding of functions was used, as a local convention, to differentiate between different communications layers and associated function types, with all Application Layer functions having the same color. In the context of the Application Layer, this is not particularly useful. For this document, the ASL has adopted the color coding of related Application Layer functions and associated information objects and services to indicate specific functional areas or groups of related functions. The color coding used is summarized in figure 3-2.

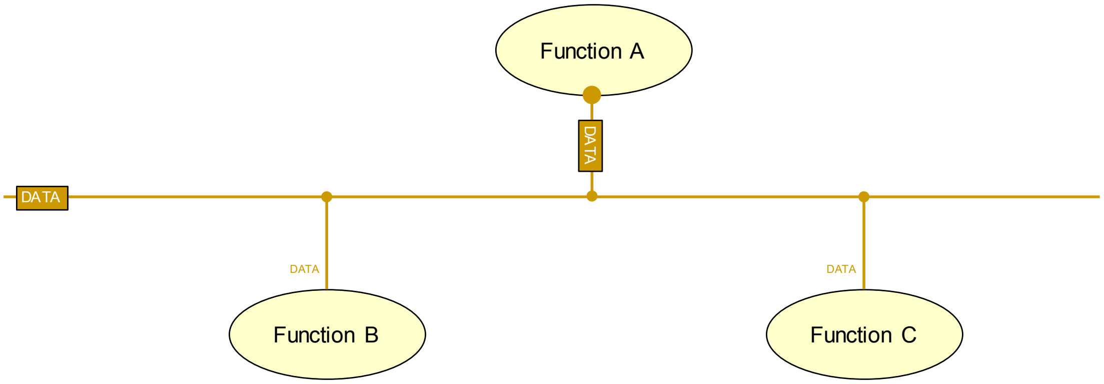


Figure 3-5: Alternative Representation for Functional Viewpoint Diagrams

An alternative representation of the Functional Viewpoint diagrams is also provided in annex B of this document. This puts the service connections at the center of the picture as a set of horizontal color-coded ‘tramlines’. Functions are then attached to these with vertical lines, distinguishing between service provider and service consumer, as before. This approach makes it easier to see which functions are the providers and consumers of each service.

3.3.3 INFORMATION VIEWPOINT NOTATION

The Information Viewpoint shows information objects and the relationships between them. The information objects in this view correspond to those shown in the Functional Viewpoint.

RASDS does not supply a recommended representation for the Information Viewpoint, so in this document, a form of entity relationship diagram is used based on simplified UML class diagram notation. In the context of the ASL ADD, the types or classes of information object are identified, but their detailed structure is not elaborated. Information classes are represented as a simple rounded rectangle containing the name of the information class (or an abbreviation of it), connected by standard UML relationships to other information classes.

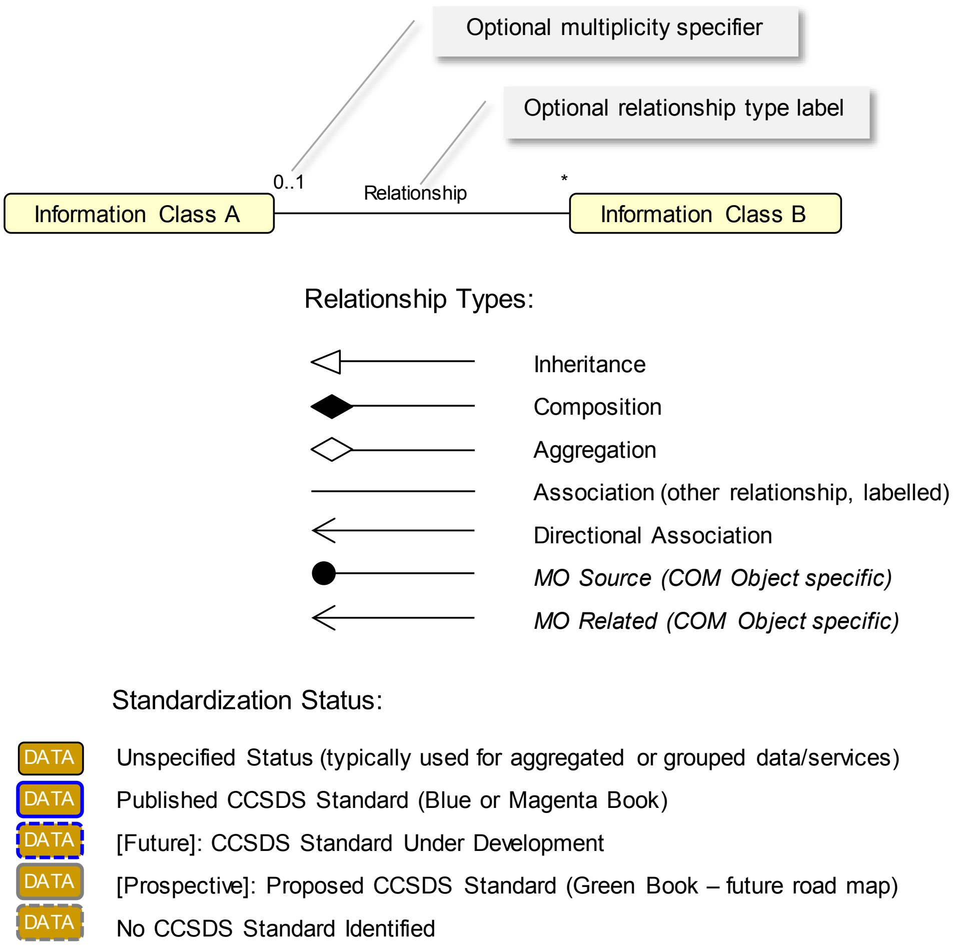

Optional multiplicity specifier
Optional relationship type label
0..1
Relationship
*
Information Class A
Information Class B
Relationship Types:
Inheritance
Composition
Aggregation
Association (other relationship, labelled)
Directional Association
MO Source (COM Object specific)
MO Related (COM Object specific)
Standardization Status:
DATA
Unspecified Status (typically used for aggregated or grouped data/services)
DATA
Published CCSDS Standard (Blue or Magenta Book)
DATA
[Future]: CCSDS Standard Under Development
DATA
[Prospective]: Proposed CCSDS Standard (Green Book – future road map)
DATA
No CCSDS Standard Identified

Figure 3-6: Graphical Notation for Information Viewpoint Diagrams

Standard UML notation is used for the following common relationship types as shown in figure 3-6:

– **Inheritance**: Information Class B is derived from Information Class A, which is typically an abstract class, extending and specializing its detailed definition.

– **Composition**: Information Class B forms part of Information Class A.

– **Aggregation**: Information Class B is included in a collection of objects represented by Information Class A.

– **Association**: a general-purpose relationship between Information Classes A and B. This is typically labelled to indicate the nature of the relationship, and may be non-directional or directional.

Specific notation is used to represent two specialized relationship types defined by the MO COM.

– **MO Source**: Information Class A is the source of Information Class B. This is typically used to provide a link back to the information object responsible for the creation of another. This may be used to represent a parent-child relationship, and can provide an audit trail of control responsibility.

- **MO Related**: a secondary generic relationship supported by MO COM, represented as a directed association. This can be used to represent the relationship between an information object instance and its definition.

The status of CCSDS standardization is shown by the border style of the information objects, using the same color and line-style coding as the Functional Viewpoint.

As a shorthand to represent a contained set of information classes, the contained objects can be placed within a large rounded rectangle with a dashed border, with a composition or aggregation relationship between this and the container information class.

Color coding (for functional area) and other graphical conventions (for abstract classes, non-standardized classes, etc.) are consistent with those defined for the Functional Viewpoint.

3.3.4 SERVICE VIEWPOINT NOTATION

The Service Viewpoint is represented as a set of tables listing the identified service interfaces or data formats for a given functional area. These tables have the following columns:

Area Functional Area represented by color coding consistent with that defined for the Functional Viewpoint.

Group Name or acronym for a group of related services or data formats.

Service Name of service or data format.

Functions List of functions from the Functional Viewpoint that act as provider or consumer of the service. For data formats, this corresponds to the functions that output (provider) and input (consumer) the data format.

Operations List of capabilities of the service, corresponding to high-level groups of related operations that the consumer may invoke on the provider.

Data List of information objects from the Information Viewpoint that are transferred across the service interface.

Description Description of the purpose of the service and its dependencies on other services.

Standards References to the CCSDS Recommended Standards relevant to the service. This may include the full-service specification, other specifications it is dependent on, or informational reports that identify the service as part of the future roadmap of CCSDS standardization.

S Status of Service Specification.

D Status of Data Format Specification.

The last two columns indicate the current status of service specification by CCSDS. It is divided into two columns to indicate service- and data-format specification, as these may be separately defined. The status is color-coded consistently with the Functional and Information Viewpoints to show standardization status:

🟦 Blue — Published CCSDS Recommended Standard or CCSDS Recommended Practice (Blue Book or Magenta Book).

🟦 Blue/Light Blue — Published CCSDS Recommended Standard providing a partial solution.

▨ Blue/White — [Future]: CCSDS Recommended Standard under development.

◼ Grey — [Prospective]: Identified CCSDS Recommended Standard (mentioned in a Green Book or future road map).

□ White — No CCSDS Recommended Standard Identified.

3.3.5 COMMUNICATION VIEWPOINT NOTATION

The Communications Viewpoint shows how the application-level services are supported by underlying communications protocol stacks, depending on their deployment context. RASDS notation and color coding consistent with the SCCS-ADD (see 3.1) is used to support this viewpoint.

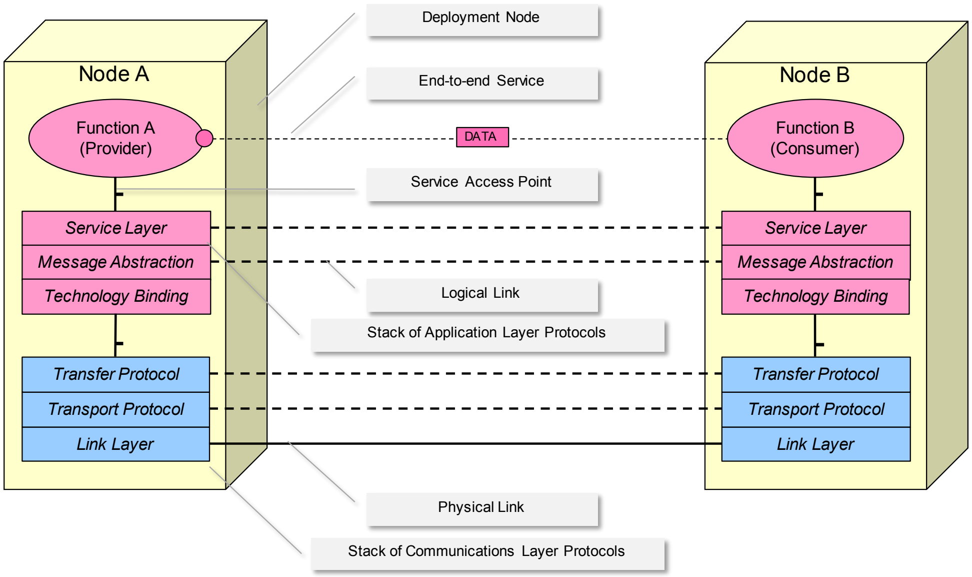


Figure 3-7: Graphical Notation for Communications Viewpoint Diagrams

The figure above illustrates this for a generic example of service protocol layering at both communication and Application Layers. The protocol stacks must match on both sides of a deployed interface to enable communication at the Application Layer. Communications-layer protocols (Data Link and possibly Network) correspond to those defined in the SCCS-ADD and are shown in blue. Application Layer protocols identified in that document were shown in pink.

In order to give prominence to the typical MO *Service Layer* bindings, this diagram differs from the normal ISO OSI Basic Reference Model (BRM) (reference [43]) stack layering. The *Service Layer* is used to distinguish the Application Layer protocol. The *Transfer Protocol Layer* is used to distinguish any standard message exchange protocol, such as Java Message Service (JMS), AMS, or ZeroMQ, that is adopted as the means to transfer *Service Layer* messages. The *Transport Protocol Layer* might be a typical network or transport protocol pairing, like TCP/IP, or the DTN/Licklider Transmission Protocol (LTP) pairing used over space links, or it might be a simple Space Packet (reference [3]) transport mapping.

As in other viewpoints, italics are used to indicate abstraction; in this case, italics indicate a generic protocol that must be replaced by a specific protocol in an actual deployment.

3.3.6 PHYSICAL VIEWPOINT NOTATION

The Physical Viewpoint identifies potential deployment nodes and the types of communication interface supported between them. RASDS notation and color coding consistent with the SCCS-ADD (see 3.3) is used to support this viewpoint.

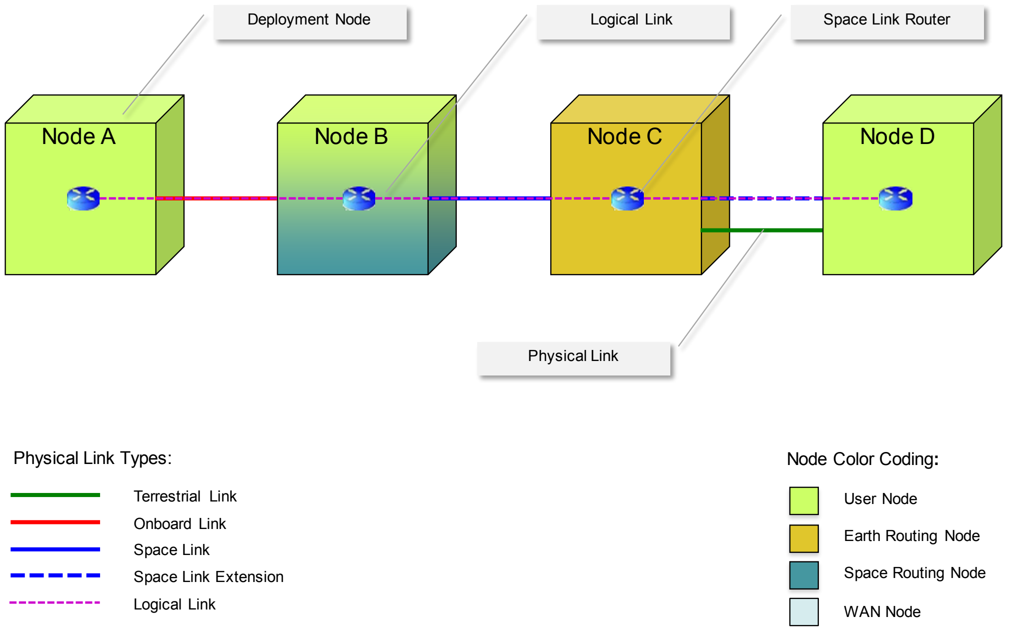


Figure 3-8: Graphical Notation for Physical Viewpoint Diagrams

Each type of example deployment node is represented as a 3D physical node element, color coded to indicate whether it is a user node (space or ground), or a space or ground segment routing node. Color coding of deployment nodes is consistent with the SCCS-ADD as shown in figure 3-1. Shaded color is used when a node acts as both user and routing node.

Physical links between nodes are shown as solid lines. As an extension to the SCCS-ADD notation, these are color-coded to indicate which Communications Viewpoint link context applies: Space, Onboard, or Terrestrial Link. A dashed blue line may be used to show use of the CCSDS Space Link Extension standard to tunnel space data link flows between a ground station and other ground segment facilities.

The router symbol is only used when there is a Space Internet routing function using SSI protocols. Standard terrestrial internet routers are omitted for to avoid clutter. In this view, where Space Internet routing is shown, the logical link between nodes is shown by a dashed magenta line, from a source router to a destination router, running over the corresponding physical links.

3.3.7 DEPLOYMENT VIEWPOINT NOTATION

The Deployment Viewpoint shows potential functional deployment for a number of physical deployment scenarios.

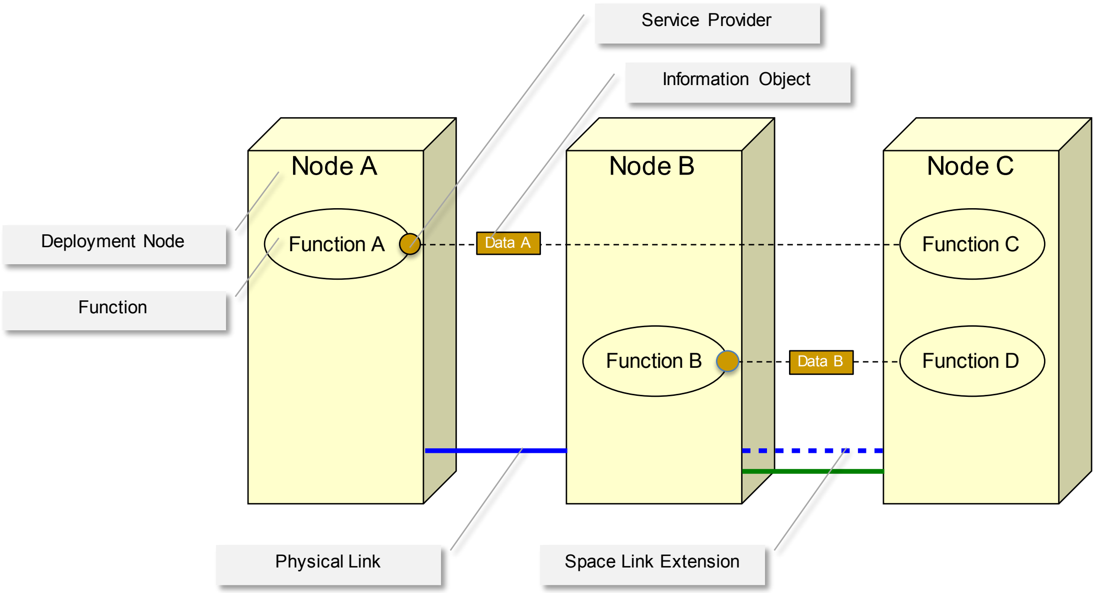


Figure 3-9: Graphical Notation for Deployment Viewpoint Diagrams

The diagrams combine the Physical Viewpoint notation for deployment nodes and physical links, with the Functional Viewpoint notation for deployed functions and the associated services and information objects that are exposed to the interfaces between deployment nodes.

The diagram above shows neutral colors for both deployment nodes and functions, services, and information objects. In the Deployment Viewpoint diagrams used later in this document, the color coding for deployment nodes and physical links is consistent with both the Physical Viewpoint and the SCCS-ADD (see 3.3.6).

Color coding for functions, services, and information objects is consistent with that of the Functional Viewpoint (see 3.3.2).

To avoid clutter, typically only one function bubble is shown per higher-level functional group on each node. If only a subset of functionality is deployed, then the deployed subfunctions are listed in the text within the bubble.

For clarity, only those logical interfaces between functions exposed at physical node boundaries are shown on the diagrams. Logical interfaces between functions deployed on the same node may also exist, consistent with the Functional Viewpoint. When a logical interface crosses a node without terminating at a function deployed there, this indicates that the lower level (physical and other) communication path is routed through that node, but that it is opaque to the node because the node only looks at the Data Link or Network Layer and not the contents of those protocol data units.

3.3.8 IMPLEMENTATION VIEWPOINT NOTATION

The Implementation Viewpoint shows how the standards may be used to generate actual software components that implement the interfaces addressed. The generation process itself may be manual or use autocoding techniques.

The viewpoint comprises three views:

– Information Transformation View;

– Component Creation View;

– Implementation Process View.

The Information Transformation View shows the relationships between standards, or standardized objects, and uses the same representation as the Information Viewpoint. The following additional conventions may be used:

– When the information object corresponds to a published (or proposed) standard, this is indicated by a book symbol 📖, color-coded to indicate a CCSDS Blue or Magenta book, or white to indicate an external standard.

– Directional relationships may be used to indicate specific relationships between information objects or standards, such as representing the transformation of an abstract standard into a concrete implementation, or a binding to an underlying communications protocol or encoding.

– Information objects may follow the color-coding introduced in figure 3-1 to represent whether they relate to the Application Layer or communications layers.

The Component Creation View uses similar graphical notation to the Communication Viewpoint, but is augmented with a representation (based on UML component) for deployable software components. The ‘implements’ relationship is used to show the correspondence between deployable software components and the protocol layers that it implements (it should be noted that a single component may implement multiple protocol layers).

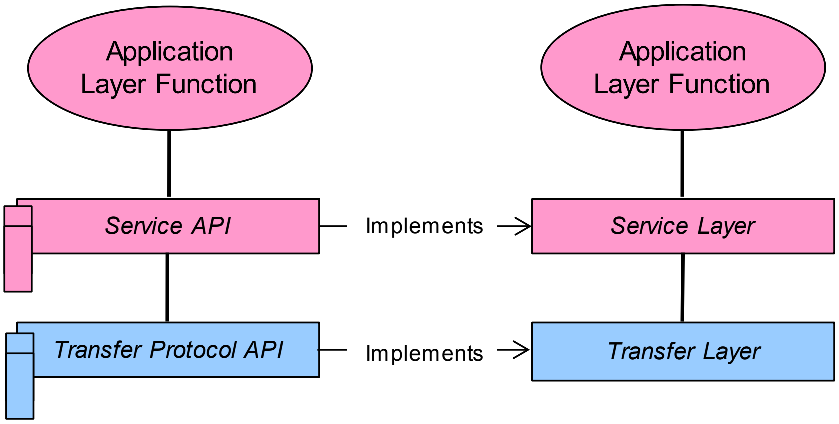


Figure 3-10: Graphical Notation for Implementation Viewpoint Component Creation View

The Implementation Process View represents a code generation function (manual or automatic) or other processes that are not themselves part of the deployed system, but are used to generate software components that implement a standard interface. The process function is annotated with a ⚙ symbol.

Inputs to the process are abstract standards or information objects identified in the Information Transformation View. Other standards or information objects may be referenced by the generation process. These typically define how to transform an abstract standard into a concrete implementation. In practice, for automated code generation, the transformation is implemented within the generation process itself.

The Outputs of the process are deployable software components identified in the Component Creation View that implement the abstract standard for a given deployment context.

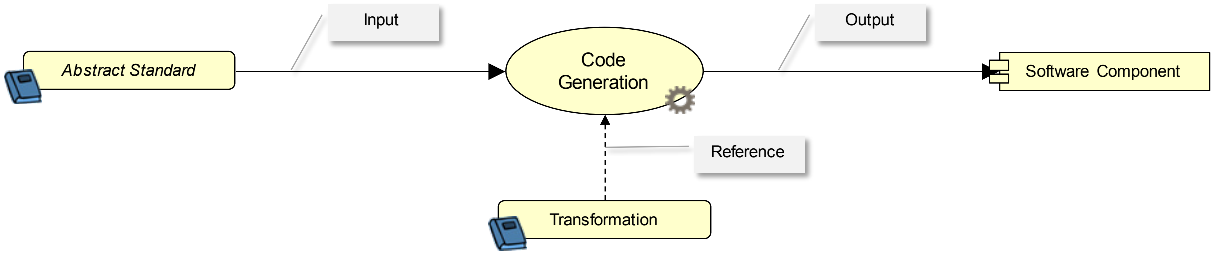


Figure 3-11: Graphical Notation for Implementation Viewpoint Process View

4 FUNCTIONAL VIEWPOINT

4.1 OVERVIEW

The Functional Viewpoint diagrams show functions (functional elements) and the associated logical links between them. For both MO functions and SO functions each section is introduced with an overall functional viewpoint context diagram, showing the major functional groups, and then a series of diagrams that drills into the details of each functional group.

4.2 MO FUNCTIONS

4.2.1 INTRODUCTION

This section addresses MO functionality. The Functional Viewpoint presented here is of Application Layer functionality, independent of its deployment location. These are typically deployed on the ground, but options for ground and flight deployments are treated in 9.5.2.

There is a wide range of MO functions and these are presented hierarchically with two levels of decomposition. First the MO capabilities are decomposed into five main functional groups:

– Mission Control;

– Navigation and Timing;

– Mission Planning and Scheduling;

– Operations Preparation;

– Data Storage and Archiving.

Each of these is then in turn decomposed into a set of high-level functions. The same approach is taken for the services and information objects representing the Application Layer interfaces between functions that support information exchange and other operations. This is distinct from the communications layer protocol stack used to implement the interface, which is described in the Communications Viewpoint (see 7.3).

The use of color coding to differentiate these functional groups was introduced in 3.3.1.

The remainder of the section is structured as follows:

a) **MO Context**: scope of the MO capabilities;

b) **MO Functional Groups**: top-level decomposition of MO capabilities;

c) **Common Functions and Services**: applicable across multiple functional groups;

d) **<Functional Group>**: a set of subsections, one per MO Functional Group, providing their second-level decomposition into functions.

Each of the above subsections comprises a Functional Viewpoint diagram and a description of each of the functions it contains. Descriptions for external and higher-level functions introduced in earlier diagrams are not repeated.

For more detailed descriptions of the information objects and services that are also shown in the diagrams, reference should be made to the Information and Service Viewpoints, respectively.

CCSDS does not intend to standardize the internal design of the MO functions, only the associated provided and required interfaces and the protocol behaviors that are exposed at these interfaces. The standardization status of the interfaces is indicated in the diagrams using the style conventions outlined in figure 3-4. In the function descriptions that follow each diagram, there is a list of the function's provided and required interfaces. When this corresponds to a specific interface, references to existing CCSDS Recommended Standards are provided. If there is no existing CCSDS Recommended Standard, then it is indicated whether it is currently under development [Future], on the roadmap for future CCSDS standardization [Prospective], or if there is no CCSDS Recommended Standard planned. In the higher-level diagrams (Context and Functional Groups), interfaces may be shown aggregated at the functional group level (see figure 3-7); in this case, the standardization status is not shown for the aggregated group. The reader is referred to the corresponding functional-group-level diagram for a detailed status of individual interfaces.

For both existing and future CCSDS Recommended Standards, these may address only the data format, or both the data format and service (associated interface behavior). This is clarified in the Service Viewpoint provided in section 6.

The functional viewpoint identifies the MO functions, but it is not implied that in a given context there is only one instance (or copy) of that function. For example, in the context of a fleet or constellation MO system, some functions may be instantiated for each spacecraft, or a single function may be responsible for multiple spacecraft. MO information objects (data) and services may similarly refer to or support single or multiple spacecraft. Within the MO framework, this is supported by the concept of *domain* (see 5.2.4.2), which allows for a hierarchical namespace, that can be used to represent mission, spacecraft, subsystem or payload. An example of how MO services and data can be deployed for a constellation mission is described in 9.2.6.

4.2.2 MO FUNCTIONAL VIEWPOINT CONTEXT

4.2.2.1 Overview

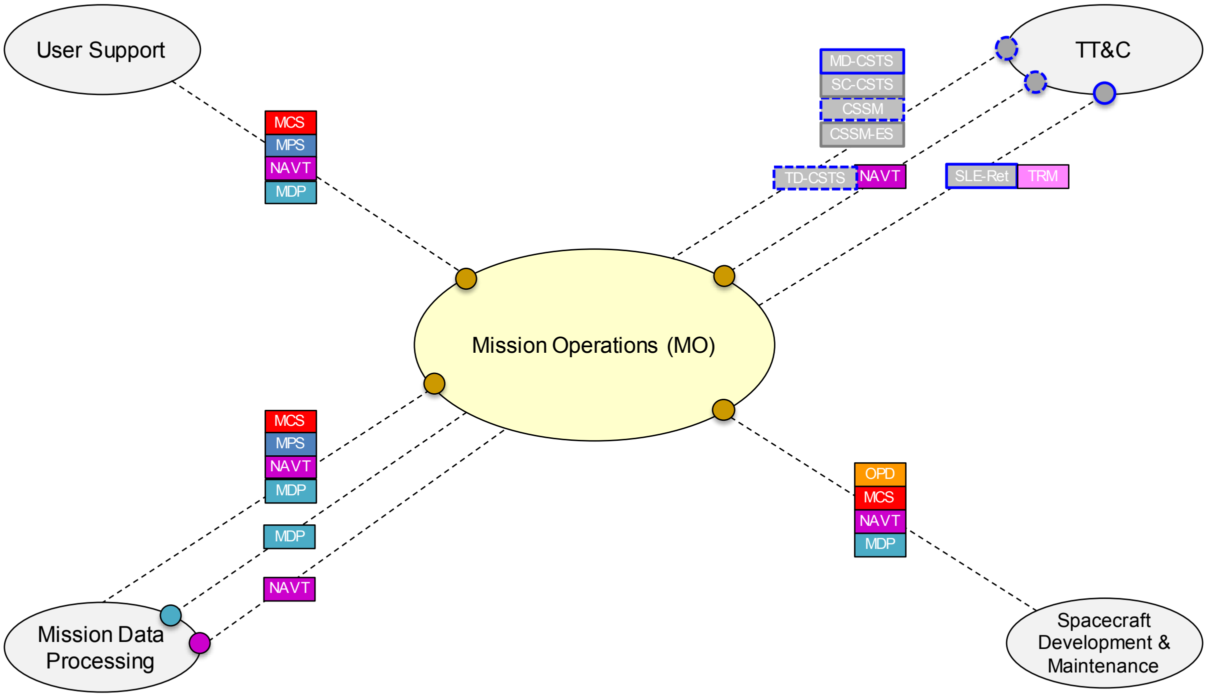


Figure 4-1: MO Functional Viewpoint Context

The MO standards cover a wide range of functions associated with the support of mission operations and mission data archiving. This functionality is associated with Mission Operations Centers (MOC) or Payload Operations Centers (POCs) or mission data archiving facilities and includes:

– Mission Control;

– Navigation and Timing;

– Mission Planning and Scheduling;

– Operations Preparation;

– Data Storage and Archiving.

The diagram above shows the context of the MO functions, and the external functions and the logical interfaces with them in terms of data (information objects) and services.

The diagram does not show the spacecraft or space segment as an external function. This is because the functional model allows for deployment of MO functions within the space segment. Logical interfaces with the spacecraft for these kinds of deployments are seen as peer-to-peer application-level interfaces between mission operations functions deployed on the ground and onboard, and are therefore treated as *internal* to MO. MO functions would otherwise connect to the spacecraft functions using traditional command and telemetry TT&C communications path. Example physical deployment scenarios showing these different options are provided in the Deployment Viewpoint.

While most current missions effectively have representative mission operations functions onboard the spacecraft, their Application Layer interaction with the ground is either proprietary/bespoke or follows non-CCSDS standards. However, MO standards are intentionally open to deployment across space links, and the Functional Viewpoint has been modelled in a way that permits this to be shown. As previously introduced in 2.4, a staged approach to MO function deployment in the onboard environment is possible, treated in more detail in section 9:

a) within the ground segment;

b) across the space link to the spacecraft;

c) within the spacecraft.

Integration of MO functions with SO functions is modelled in the Protocol Viewpoint and in the Deployment Viewpoint.

At the Application Layer, MO functions do not interact with the TT&C function (ground stations) for the acquisition of telemetry or the uplink of telecommands. This interaction does occur at lower communications layers, but from the Application Layer Functional Viewpoint presented here, this interaction utilizes the communications-layer functions located in the TT&C node, as is depicted in the later Protocol and Deployment Viewpoints.

MO functions do interact at the Application Layer with the TT&C function for the management of communications and tracking contacts with spacecraft, for which a range of cross-support services is used, as detailed below.

At the communications layer, MO functions access the space link via CSS SLE and CSTSes. Telecommands are sent as Command Link Transmission Units (CLTU) via the SLE Forward CLTU (FCLTU) service (reference [48]). Telemetry is received either via the SLE Return All Frames (RAF) or Return Channel Frames (RCF) services (references [46] and [47]).

Communications details for TM, TC, AOS, and Unified Space Data Link Protocol (USLP), which are communications Data Link Layer protocols, are already covered in the SCCS-ADD (reference [2]), and are not treated as Application Layer information. The CCSDS Packet (reference [3]) is often used as an Application Layer data transfer structure and mappings from MO SM&C to that standard are available, as are others. The MO application-layer data (or information objects) refer to the meaningful information contained within those packets.

The Application Layer data exchanged between MO and external functions can be grouped into types of information as follows:

- Mission Control Services (MCS);

- Navigation and Timing (NAVT);
- Mission Planning and Scheduling (MPS);
- Operations Preparation Data (OPD) (onboard configuration: software, procedures);
- Mission Data Products (MDP).

The standardization status of these groups can be summarized as follows:

MCS: The MO M&C Service is a published standard, but other services in this group are [Future] or [Prospective].

NAVT: Most navigation data messages have been published as interoperable data formats (rather than services); some are under development. Navigation Services are [Prospective] with no current plans for development; Timing services are [Future].

MPS: Mission Planning and Scheduling Services are [Future], currently under development.

OPD: Operations Preparation Data is mostly not standardized or described at present. The XML Telemetric and Command Exchange (XTCE) Recommended Standard (reference [23]) provides a [Partial] data format solution for exchange of TM and TC definitions; standard formats for Procedure and Mission Planning definitions are [Prospective]. Onboard software is [Not planned for standardization].

MDP: The Mission Data Product distribution service is [Future], currently under development.

DSA: Data Storage and Archiving Services comprise three main elements:

- the general-purpose MO COM Archiving service that can be used in conjunction with any COM-compliant service (reference [16]);
- MO File Transfer and Management services [Future] (reference [61]);
- Data Archive Interfaces to Producers and consumers for which abstract process specifications exist but full-service definitions are [Future].

MO application-layer interactions with the TT&C functions are partially addressed by services standardized by the CCSDS CSS Area. These include:

- Negotiation of the provision of TT&C services with a TT&C network provider. This is a specific Application Layer planning and scheduling service covered by the CSS Service Management (SM) standard, shown on the diagram as CSSM.
  - Provision of detailed events relating to the provision of cross support services by a TT&C network provider. The identified CSS Service Management Event Sequence (SMES) is shown on the diagram as CSSM-ES.

- Provision of Tracking and Ranging Data acquired by the TT&C function. This is provided through the CSS Tracking Data standard, which is shown on the diagram as TD-CSTS[NAVT]. The data carried on this service is compliant with the MO Navigation Tracking Data Message (TDM).
- Monitoring of the TT&C function itself, in terms of the cross-support services it provides and events occurring on the link. This is covered by the CSS Monitored Data Service (MDS) standard (reference [44]), shown on the diagram as MD-CSTS.
- Control of the TT&C function itself, in terms of the cross-support services it provides. This is covered by the identified CSS Service Control (CS) standard, which is shown on the diagram as SC-CSTS.
- Provision of Earth reception time (Time Reception Message [TRM]): accurate timestamping of the reception time of messages received from the spacecraft, which is needed to support onboard time correlation. This is currently performed at telemetry-frame or packet level, with the Earth Reception Time being additional information added as part of the corresponding CSS SLE RAF or RCF transfer services, shown in the diagram as SLE-Ret TRM.

The following subsections describe each of the External Functions shown in the MO Area context diagram.

4.2.2.2 Mission Data Processing

Function

Acquisition and processing of payload or mission data performed systematically within the mission data system. The nature of any such processing is specific to mission type, but may include:

- science data processing;
- image data processing (raw, calibrated, derived);
- navigation system data processing;
- telecommunications system network management.

Provided Interfaces

- **MDP**: Mission Data Products (for distributing payload data, including to archives) [Future] (reference [56]).
- **NAVT**: Navigation and Timing Data (accurate spacecraft position may be derived from image data).

Required Interfaces

- **MCS**: Monitoring & Control Data.
- **MPS**: Mission Planning & Scheduling Data.
- **NAVT**: Navigation and Timing Data.
- **MDP**: Mission Data Products (payload data may be routed via Mission Operations; archived Mission Data Products may be retrieved) [Future] (reference [56]).
- **PAIS**: Producer Archive Interface (reference [31]).
- **CAIS**: Consumer Archive Interface [Future] (reference [66]).

4.2.2.3 Spacecraft Maintenance and Development

Function

The manufacturer of a spacecraft (or payload) provides configuration data that Mission Operations requires to configure its systems. This includes the onboard software, spacecraft database, onboard control procedures, and operational procedures. The manufacturer may retain or be delegated responsibility for maintenance of any or all of these for the duration of the mission, or alternatively, they may be provided initially and handed over to Mission Operations for maintenance.

The manufacturer may also provide analysis and support in the event of spacecraft anomalies and may perform long-term performance monitoring of the spacecraft or payload systems. Following such analysis, updates to onboard software, spacecraft database, onboard operations procedures, or operational procedures may be provided.

Required Interfaces

- **OPD**: Submission/Retrieval of Operations Preparation Data, including onboard software, spacecraft database, and procedures.
- **MCS**: Monitoring & Control Data (for analysis).
- **NAVT**: Navigation and Timing Data (for analysis).
- **MDP**: Mission Data Products (for analysis) [Future] (reference [56]).
- **CAIS**: Consumer Archive Interface [Future] (reference [66]).

4.2.2.4 Telemetry, Tracking, and Commanding

Function

The TT&C function represents a network of ground stations providing telemetry acquisition, telecommand uplink, and spacecraft tracking and ranging services. At network level it is also responsible for planning and scheduling TT&C resources based on requests for the provision of TT&C services.

As explained previously, telemetry acquisition and telecommand uplink are communications-layer functions that do not directly interact at the Application Layer with MO functions. MO data interchanges are, however, carried across space links provided by these functions.

Some MO functions do, however, have Application Layer interactions with the TT&C function:

- to negotiate the provision of TT&C services with a TT&C network provider;
- to obtain spacecraft tracking and ranging data acquired by the TT&C station;
- to provide orbit vectors or predicted orbital events to the TT&C station to enable it to acquire and track a spacecraft;
- to obtain accurate Earth-reception timestamps associated with messages received from the spacecraft (this is needed to support onboard time correlation);
- to obtain monitoring data on the status of TT&C services provided.

Provided Interfaces

- **CSSM**: Service Management interface for negotiation of the provision of cross-support (TT&C) services.
- **CSSM-ES**: Service Management Event Sequence interface for the provision of detailed timings of cross-support (TT&C) services.
- **MD-CSTS**: Monitored Data Service giving status of provided TT&C services.
- **SC-CSTS**: Service Control enabling control of provided TT&C services.
- **SLE-Ret[TRM]**: Provision of Earth reception timestamps as part of the CSS SLE Return (RAF or RCF) transfer services.
- **TD-CSTS[NAVT]**: Provision of Tracking and Ranging Data (also via file transfer including D-DOR). The tracking data itself follows the MO Navigation TDM format.

Required Interfaces

- **NAVT**: Provision of Orbit and Event Data messages (included as part of Service Management as far as CSS is concerned).

4.2.2.5 User Support

Function

The User Support function represents any external user of a space system, including Principal Investigators (PIs) for many science missions and those requesting specific observations in astronomy, Earth observation, or planetary missions. Two principal subfunctions are included:

- tasking the mission to perform particular operations, typically payload operations supporting scientific experiments or observations;
- analysis of mission data products.

Tasking is primarily supported through Mission Planning and Scheduling interfaces that allow the submission and tracking of Planning Requests (PRQs). However, this may also require supporting information, such as: spacecraft orbit and predicted orbital events, and spacecraft or payload status.

Analysis requires the provision of Mission Data Products, but may also need data such as spacecraft orbit and attitude, and spacecraft or payload status to support interpretation.

Provided Interfaces

None.

Required Interfaces

- **MPS**: Mission Planning & Scheduling.
- **MCS**: Monitoring & Control Data (for analysis).
- **NAVT**: Navigation and Timing Data (for analysis).
- **MDP**: Mission Data Products (for analysis) [Future] (reference [56]).
- **CAIS**: Consumer Archive Interface [Future] (reference [66]).

4.2.3 MO FUNCTIONAL GROUPS

4.2.3.1 Overview

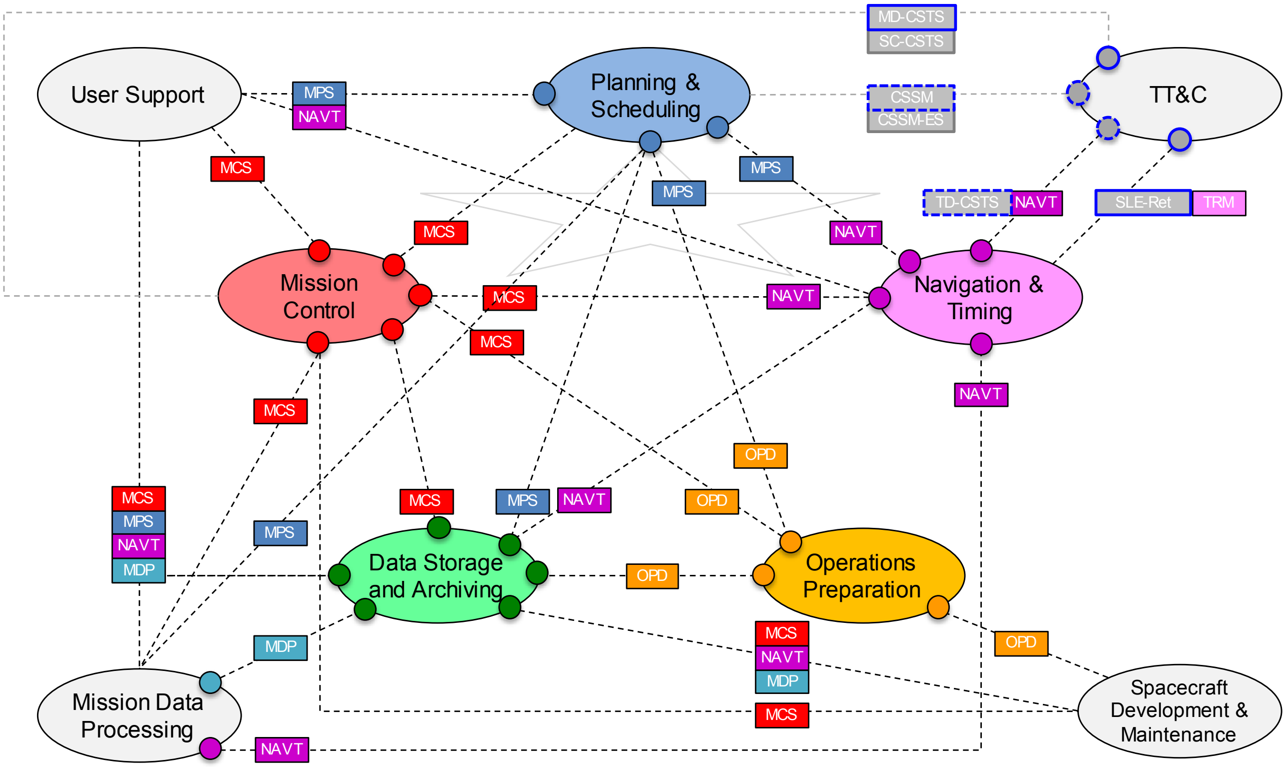


Figure 4-2: MO Level 1: Functional Groups

The first-level decomposition of the MO capabilities identifies five main functional groups, color coded as defined in the introduction to this section:

- Mission Control;
- Mission Planning and Scheduling;
- Navigation and Timing;
- Operations Preparation;
- Data Storage and Archiving.

In addition to these five main functional groups, there is a set of common functions and services that may be used by all MO functions. These are described in 4.2.4, below.

At this level, all data (information objects) exchanged between functions can also be grouped by the principal functional group with which they are identified. In addition, MDP are identified as a type of information that is generated externally to the defined MO functions.

The same acronyms are used as in the MO Context (4.2.2, above). Potential service interfaces are indicated by a color-coded circle representing the service provider. Data formats may be separately defined (Navigation and Mission Planning data) or specified in the context of the service (MO M&C and Mission Planning).

Data Storage and Archiving differs slightly, in that it is principally concerned with the temporary storage, archiving, and retrieval of the information objects generated by other functional groups.

4.2.3.2 Mission Control

Function

Mission Control encompasses all functions associated with the execution of mission operations, both in terms of spacecraft control and that of the wider mission system. It includes:

- Monitoring of Spacecraft, Payload, and Mission Status;
- Provision of Status Displays to the operations team;
- Manual Commanding;
- Execution of Automated Operations;
- Onboard Configuration Management.

Provided Interfaces

- **MCS**: Mission Control Data.

Required Interfaces

- **OPD**: Operations Preparation Data (onboard software, spacecraft database, and automated procedure definitions).
- **NAVT:** Navigation and Timing Data.
- **DSA**: Data Storage and Archiving services.
- **MD-CSTS**: Monitored Data Service giving status of provided TT&C services.
- **SC-CSTS**: Service Control enabling control of provided TT&C services [Future].

4.2.3.3 Mission Planning & Scheduling

Function

Mission Planning and Scheduling encompasses both: the generation of mission plans, based on received PRQs and defined planning constraints, and the realization of those plans, using underlying Mission Control services to execute the planned activities.

Provided Interfaces

– **MPS**: Mission Planning & Scheduling [Future] (reference [63]).

Required Interfaces

– **MCS**: Mission Control Services.

– **NAVT**: Navigation and Timing Data (including predicted orbital events).

– **OPD**: Operations Preparation Data (planning database) [Prospective].

– **DSA**: Data Storage and Archiving services.

– **CSSM**: Service Management interface for negotiation of the provision of cross-support (TT&C) services.

– **CSSM-ES**: Service Management Event Sequence interface for the provision of detailed timings of cross-support (TT&C) services.

4.2.3.4 Navigation and Timing

Function

Navigation is concerned with the management of spacecraft orbital dynamics and spacecraft attitude; Timing is concerned with the management of onboard clocks. It includes:

– Position and/or Time Determination;

– Orbit Determination and Propagation;

– Attitude Determination;

– Maneuver Planning;

– Conjunction Assessment.

– Time Correlation;

Provided Interfaces

Navigation and timing currently provides interoperable data exchange standards, but not service interfaces.

- **NAVT**: Navigation and Timing Data:
  - Orbit Data;
  - Attitude Data;
  - Predicted Orbital Events;
  - Conjunction Data (collision warnings);
  - Spacecraft Maneuvers [Prospective];
  - Time Correlation Data [Future][2].

Required Interfaces

- **TD-CSTS**: CSS Tracking Data Service (carries NAVT Tracking Data Messages) (reference [45]).
- **NAVT**: Navigation and Timing Data:
  - Tracking Data;
  - Attitude Data;
  - Pointing Requests.
- **MPS**: Mission Planning & Scheduling (maneuver PRQ) [Future] (reference [63]).
- **DSA**: Data Storage and Archiving services.
- **CSS-Ret[TRM]**: Earth reception timestamps as part of the CSS SLE Return (RAF or RCF) transfer services.

4.2.3.5 Operations Preparation

Function

Operations Preparation is an offline function concerned with the preparation, maintenance, configuration management, and distribution of mission operations configuration data. This configuration data includes:

- onboard software [Not planned for standardization];
- spacecraft databases: defining available monitoring data and commands [Partial];
- automated procedure definitions: scripts for both onboard procedures and those automated on ground [Prospective];

[2] Currently under development as an SEA Time Management product.

- Planning Database (PDB): defining planning activities, events, and resources, together with static planning constraints [Prospective].

Configuration Management and Distribution functions are common to all configuration data types. The definition of the configuration data (and associated editing tools) is specific to the configuration data type.

Provided Interfaces

- **OPD**: Operations Preparation Data.

Required Interfaces

- **DSA**: Data Storage and Archiving services.

4.2.3.6 Data Storage and Archiving

Function

The Data Storage and Archiving function supports the storage and archiving of mission data associated with any of the other Functional Groups or Mission Data Products. This includes the following:

- management of onboard file store;
- mission operations archive;
- long-term mission data archive.

Provided Interfaces

- **DSA**: Data Storage and Archiving services.

Required Interfaces

- **MCS**: Mission Control Data.
- **MDP**: Mission Data Products [Future] (reference [56]).
- **MPS**: Mission Planning Data [Future] (reference [63]).
- **NAVT**: Navigation and Timing Data.
- **OPD**: Operations Preparation Data.

4.2.4 COMMON FUNCTIONS AND SERVICES

4.2.4.1 Overview

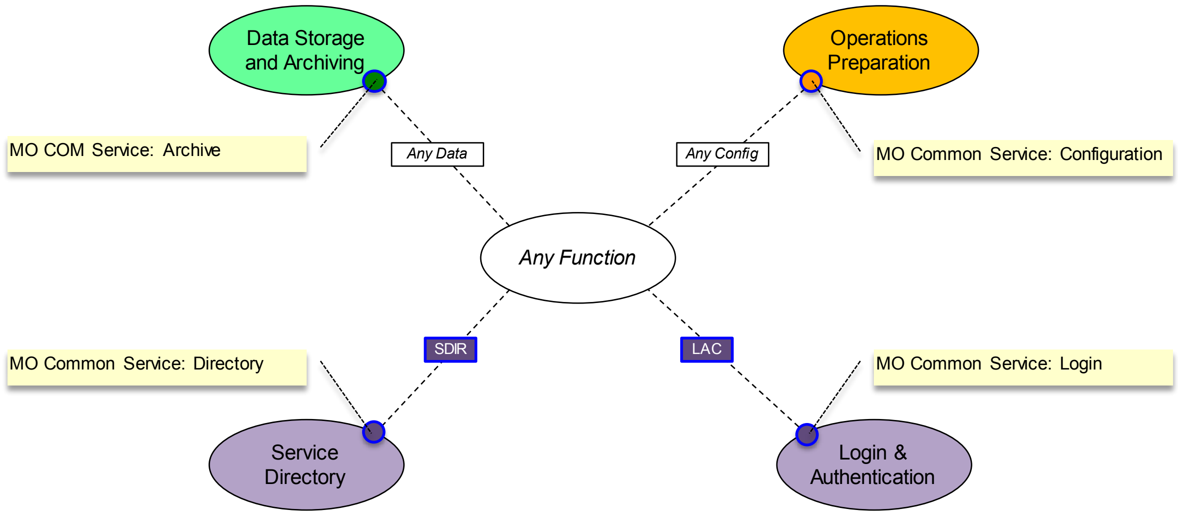


Figure 4-3: MO Level 2: Common Functions and Services

Common Functions and Services are those that can be used by any other MO function. The associated interfaces have been omitted from the other Functional Viewpoint diagrams to reduce clutter in the interests of clarity.

The associated services have been defined in the context of the MO service framework as MO Common Services or as part of the MO COM.

4.2.4.2 Data Storage and Archiving

This corresponds to the function identified in 4.2.3 above, and specifically its Operations Archive subfunction. The COM Archive service (reference [16]) is a generic service specification for the archiving and retrieval of any data whose structure is compliant with the MO Common Object Model.

4.2.4.3 Login and Authentication

Login and Authentication provides a common Login service (reference [55]) to all other functions to support user login and authentication, including access to user access rights or privileges. The information object associated with the service is the user’s Login and Authentication Credentials (LAC). Existing security functions described in the CCSDS Security Architecture (reference [38]) may be used to provide these credentials.

4.2.4.4 Operations Preparation

This corresponds to the function identified in 4.2.3 above, and specifically its Configuration Management and Distribution subfunction. The common Configuration service (reference [55]) can be applied to any configuration data and supports both the configuration management operations required by functions that maintain the configuration data and the configuration distribution operations required by those that use it.

4.2.4.5 Service Directory

The Service Directory (reference [55]) maintains a catalogue of available services and their providers that can be interrogated by any function in order to locate a required service.

4.2.5 MISSION CONTROL

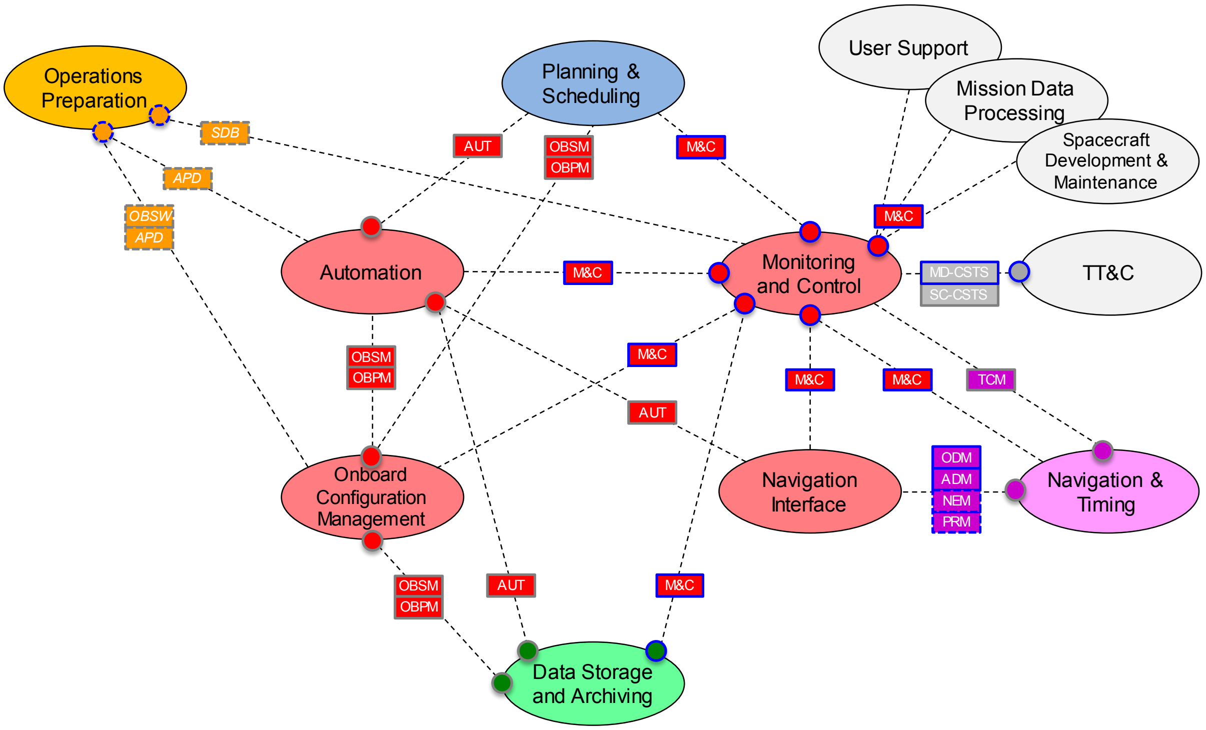


Figure 4-4: MO Level 2: Mission Control Functions

The Mission Control functional group is broken down into the following Level 2 functions:

- Monitoring and Control;
- Onboard Configuration Management;
- Automation;
- Navigation Interface.

Similarly, MCS data comprises the following information objects:

- **M&C**: Monitoring and Control Data (parameters, actions, and alerts) (reference [17]);
- **OBSM**: Onboard Software Management Data [Prospective] (reference [58]);
- **OBPM**: Onboard Procedure Management Data [Prospective] (reference [58]);
- **AUT**: Automation Data (procedure level control and monitoring) [Prospective] (reference [57]).

Any function can expose a Monitoring & Control interface for the purposes of supporting automation, it is not restricted to specialist Mission Control functions. Examples of other functions which may expose an M&C interface are Mission Planning and Navigation functions to allow their operations to be integrated within an overall mission planning and automation context.

4.2.5.1 Monitoring and Control

Function

Monitoring and Control corresponds to the basic functionality required to perform manual mission operations, both in terms of spacecraft control and that of the wider mission system. It includes:

- Monitoring of Spacecraft, Payload, and Mission Status;
- Provision of Status Displays to the operations team;
- Manual Commanding.

Provided Interfaces

- **M&C**: Monitoring and Control (parameters, actions, and alerts) (reference [17]).

Required Interfaces

- **SDB**: Spacecraft Database (SDB) defining available parameters, actions, and alerts [Partial] (reference [23]) [Nonstandard].
- **TCM**: Onboard Time Correlation Data [Future] (reference [59]).
- **MD-CSTS**: Monitored Data Service giving status of provided TT&C services.
- **SC-CSTS**: Service Control enabling control of provided TT&C services. [Future].
- **DSA**: Data Storage and Archiving services.

4.2.5.2 Onboard Configuration Management

Function

This function manages onboard configuration data, including onboard software and onboard procedures. It is responsible for the uplink and activation of new versions of configuration data, as well as the selective dump to ground of onboard configuration data (memory images or procedure scripts) for comparison to configuration controlled versions on the ground. It also maintains a record of what is currently installed on the spacecraft.

Provided Interfaces

– **OBSM**: Onboard Software Management [Prospective] (reference [58]).

– **OBPM**: Onboard Procedure Management [Prospective] (reference [58]).

Required Interfaces

– **OBSW**: Onboard Software Images [Not planned for standardization].

– **APD**: Onboard Procedure Scripts [Prospective].

– **M&C**: Monitoring & Control (reference [17]).

– **DSA**: Data Storage and Archiving Services.

4.2.5.3 Automation

Function

This function provides automation of pre-defined operations procedures. It supports execution of procedures when invoked to do so via the provided Automation interface, providing feedback on execution status.

It uses the M&C service to invoke lower-level actions and monitor status through receipt of Parameter status. It may also respond to received alerts, and raise alerts. If available, it may also integrate with other services to support their automation: onboard configuration management and the iterative invocation of automated procedures (or other automated functions) via the Automation service.

Automation can be deployed onboard the spacecraft as well as within the ground segment, although the implementation and available service set may differ significantly.

There is currently no standard for Automated Procedure Definitions (APDs) although their configuration management and distribution can make use of standard common services.

Provided Interfaces

– **AUT**: Automation [Prospective] (reference [57]).

Required Interfaces

- **APD**: Automated Procedure Scripts [Prospective].
- **M&C**: Monitoring & Control (parameters, actions, and alerts) (reference [17]).
- **OBSM**: Onboard Software Management [Prospective] (reference [58]).
- **OBPM**: Onboard Procedure Management [Prospective] (reference [58]).
- **AUT**: Automation (of other procedures or functions) [Prospective] (reference [57]).
- **DSA**: Data Storage and Archiving Services.

4.2.5.4 Navigation Interface

Function

This function manages the translation of data and services between Mission Control functions that are nominally MO framework compliant and Navigation functions, in order to translate specific Navigation Data formats for other Mission Control functions.

In principle, the Navigation Interface can receive any Navigation Data format provided by the Navigation and Timing functions and use Automation or M&C services to apply it. For example, a Pointing Request Message (PRM) could be used to invoke an automated procedure or discrete telecommand (M&C action) to perform the pointing, with specific fields from the PRM being mapped to arguments of the action or Procedure. What mapping is performed is mission specific.

Required Interfaces

- **M&C**: Monitoring & Control (reference [17]).
- **AUT:** Automation [Prospective] (reference [57]).
- **NAV:** Navigation Data (Orbit Data Messages [ODM], reference [25]; Attitude Data Messages [ADM], reference [27]; Navigation Events Message [NEM] [Future], reference [64]; PRM, reference [29]).

4.2.6 NAVIGATION AND TIMING

4.2.6.1 Overview

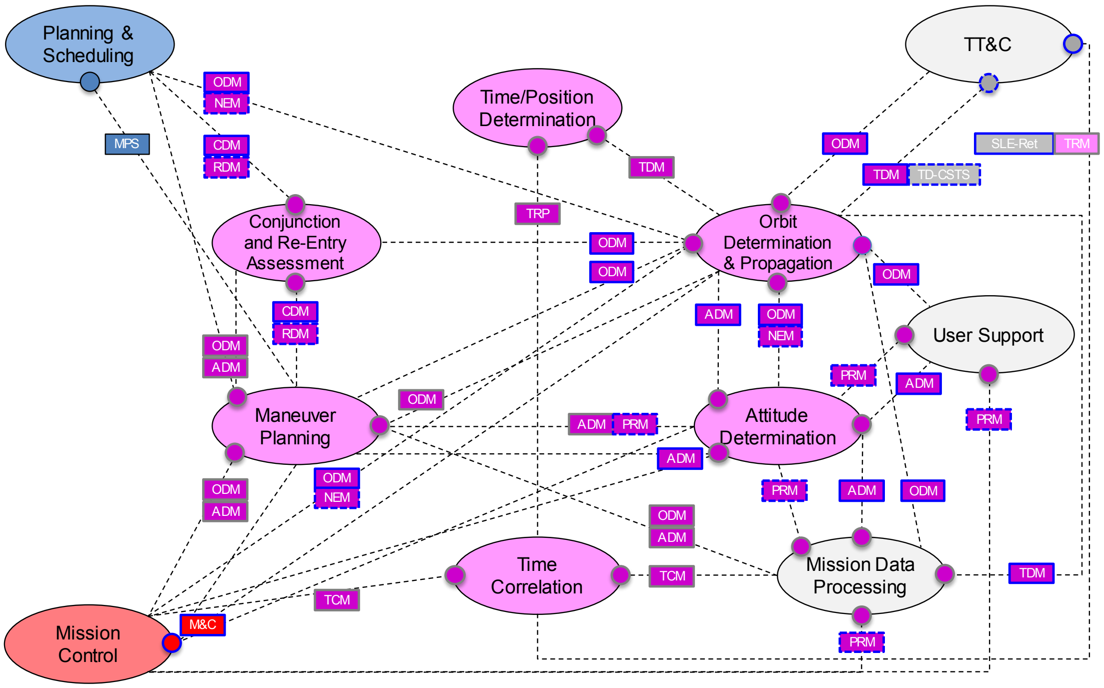


Figure 4-5: MO Level 2: Navigation and Timing Functions

The Navigation and Timing functional group is broken down into the following Level 2 functions:

- Time/Position Determination;
- Orbit Determination and Propagation;
- Attitude Determination;
- Maneuver Planning;
- Conjunction and Re-Entry Assessment;
- Time Correlation.

Because of the complexity and variety of mission-specific configurations of these navigation functions, there is no expectation that a single, discrete set of functions and interfaces will be standardized.

Similarly, Navigation and Timing Data comprises the following information objects grouped as Navigation Data and Timing Data, respectively:

– Navigation Data:

  - **ADM**: Attitude Data Messages (reference [27]),
  - **CDM**: Conjunction Data Message (reference [28]),
  - **NEM**: Predicted Navigation Events Message [Future] (reference [64]),
  - **ODM**: Orbit Data Messages (reference [25]),
  - **PRM**: Pointing Request Message (reference [29]),
  - **RDM**: Re-Entry Data Message (reference [65]),
  - **TDM**: Tracking Data Message (reference [26]);

NOTE – A Spacecraft Maneuver Message had been identified by CCSDS, but its requirements are instead to be satisfied by future extended capabilities within the ODM and ADM.

– Timing Data:

  - **TRP**: Time Report [Future] (reference [59]),
  - **TRM**: Time Reception Message (Earth Reception Time, from SLE services) [Future] (reference [59]),
  - **TCM**: Time Correlation Message [Future] (reference [59]).

4.2.6.2 Time/Position Determination

<u>**Function**</u>

The function concerns the determination of a measurement of spacecraft position at a particular point in time. There are various ways in which information relating to spacecraft position can be determined:

– through onboard reception of Global Navigation Satellite System (GNSS) signals;

– ground station tracking and ranging;

– processing of spacecraft observation data.

Time and Position determination are combined as they are closely associated and may be determined using the same method. However, specific deployments of this function may only provide either Position or Time measurements.

Provided Interfaces

- **TDM**: Tracking Data Message (reference [26]).
- **TRP**: Time Report [Future] (reference [59]).
- **TD-CSTS**: CSS Tracking Data Service (delivers TDM from a TT&C Ground Station).

Required Interfaces

None.

4.2.6.3 Orbit Determination and Propagation

Function

Measurements of spacecraft position over a period of time are used to determine the characteristics of the spacecraft orbit (the orbit vectors). An orbit vector is then propagated forward in time to predict the spacecraft's future position as a set of orbital ephemerides, and also to predict required orbital events, such as periods of ground station visibility, or sensor blindings.

Other data may be used to augment Tracking Data in determining the future orbit and/or events. This includes the state of the spacecraft (M&C), its attitude, and notification of planned maneuvers (using augmented Orbit Data Messages and Attitude Data Messages).

Provided Interfaces

- **ODM**: Orbit Data Messages (reference [25]).
- **NEM**: Predicted Navigation Events Message [Future] (reference [64]).

Required Interfaces

- **TDM**: Tracking Data Message (reference [26]).
- **ADM**: Attitude Data Messages (reference [27]).
- **ODM**: Orbit Data Messages (for maneuvers and otherwise) (reference [25]).
- **M&C**: Monitoring & Control (to access status of navigation hardware) (reference [17]).
- **TD-CSTS**: CSS Tracking Data Service (carries TDM from a TT&C Ground Station).

4.2.6.4 Attitude Determination

Function

Measurements from onboard sensors over a period of time combined with the latest orbit vector are used to determine the characteristics of the spacecraft attitude (the attitude vector). This is then propagated forward in time to predict the spacecraft's future attitude. Information on spacecraft attitude may also be derived from the processing of image data acquired by the spacecraft.

Other data may be used to augment onboard determining of the future attitude. This includes Pointing Requests and notification of planned maneuvers (using augmented Attitude Data Message).

Provided Interfaces

– **ADM**: Attitude Data Messages (reference [27]).

Required Interfaces

– **M&C**: Monitoring & Control (to access status of navigation hardware) (reference [17]).

– **ODM:** Orbit Data Messages (reference [25]).

– **NEM**: Predicted Navigation Events Message [Future] (reference [64]).

– **ADM**: Attitude Data Messages (in event of a maneuver) (reference [27]).

– **ADM**: Attitude Data Message (from Mission Data Processing) (reference [27]).

4.2.6.5 Maneuver Planning

Function

The Maneuver Planning function supports the derivation of the parameters for required orbital corrections, whether for station keeping or to change trajectory or attitude. When possible, maneuvers are optimized to minimize fuel consumption. The requirements for maneuvers may be linked to the overall mission plan, linked to the result from station keeping policy, or be in response to notification of a potential collision.

Once a planned maneuver is approved, Mission Planning & Scheduling services can be used to request its inclusion in the mission plan.

Provided Interfaces

– **ODM**: Orbit Data Messages (augmented to support translational maneuvers) (reference [25]).

– **ADM**: Attitude Data Messages (augmented to support rotational maneuvers) (reference [27]).

Required Interfaces

- **ODM**: Orbit Data Messages (reference [25]).
- **ADM**: Attitude Data Messages (reference [27]).
- **M&C**: Monitoring & Control (to access status of navigation hardware) (reference [17]).
- **CDM**: Conjunction Data Message (collision warning) (reference [28]).
- **RDM**: Re-entry Data Message (reference [65]).
- **MPS**: Mission Planning & Scheduling (maneuver PRQ) [Future] (reference [63]).

4.2.6.6 Conjunction and Re-entry Assessment

Function

Conjunction and Re-entry Assessment may be implemented as distinct functions or supported as part of a wider Space Situational Awareness (SSA) or Space Surveillance and Tracking (SST) function.

Conjunction Assessment compares the orbits of multiple spacecraft, solar-system objects, and space debris to identify predicted conjunctions (or potential collisions). This requires input of the current trajectory for each spacecraft to be considered and details of any planned maneuvers.

Re-entry Assessment models the trajectory of a spacecraft as its orbit decays and predicts the time and position of re-entry. Re-entry data includes remaining orbital lifetime, start and end of the re-entry and impact windows, impact location, and object physical properties.

Provided Interfaces

- **CDM**: Conjunction Data Message (reference [28]).
- **RDM**: Re-entry Data Message (reference [65]).

Required Interfaces

- **ODM**: Orbit Data Messages (reference [25]).
- **ADM**: Attitude Data Messages (reference [27]).

4.2.6.7 Time Correlation

Function

Time Correlation derives time-correlation coefficients between the standard reference timeframe and onboard clocks, to enable timestamps specified in terms of the onboard clock to be accurately converted to an absolute time (and vice versa for uplinked time tags).

Each correlation measurement requires two elements: a time report from the onboard clock and a precise timestamp in terms of the standard reference timeframe that can be associated with it. The time report can be any timestamp in a downlinked message, although it is common to use a specific time report that gives greater resolution. The reference time is usually provided by accurately recording the Earth Reception Time of this message, which, in conjunction with knowledge of the orbit vector and location of the ground station, can be used to derive the generation time of the time report. For spacecraft in Earth orbit, the reference time may be generated onboard from received GNSS signals.

Provided Interfaces

– **TCM**: Time Correlation Message [Future] (reference [72]).

Required Interfaces

– **TRP**: Time Report [Future] (reference [72]).

– **TRM**: Time Reception Message [Future] (reference [72]).

– **ODM**: Orbit Data Messages (reference [25]).

– **SLE-Ret**: SLE RAF or RCF transfer service includes Earth Reception Time of received telemetry (equivalent to a Time Reception Message).

4.2.7 MISSION PLANNING AND SCHEDULING

4.2.7.1 Overview

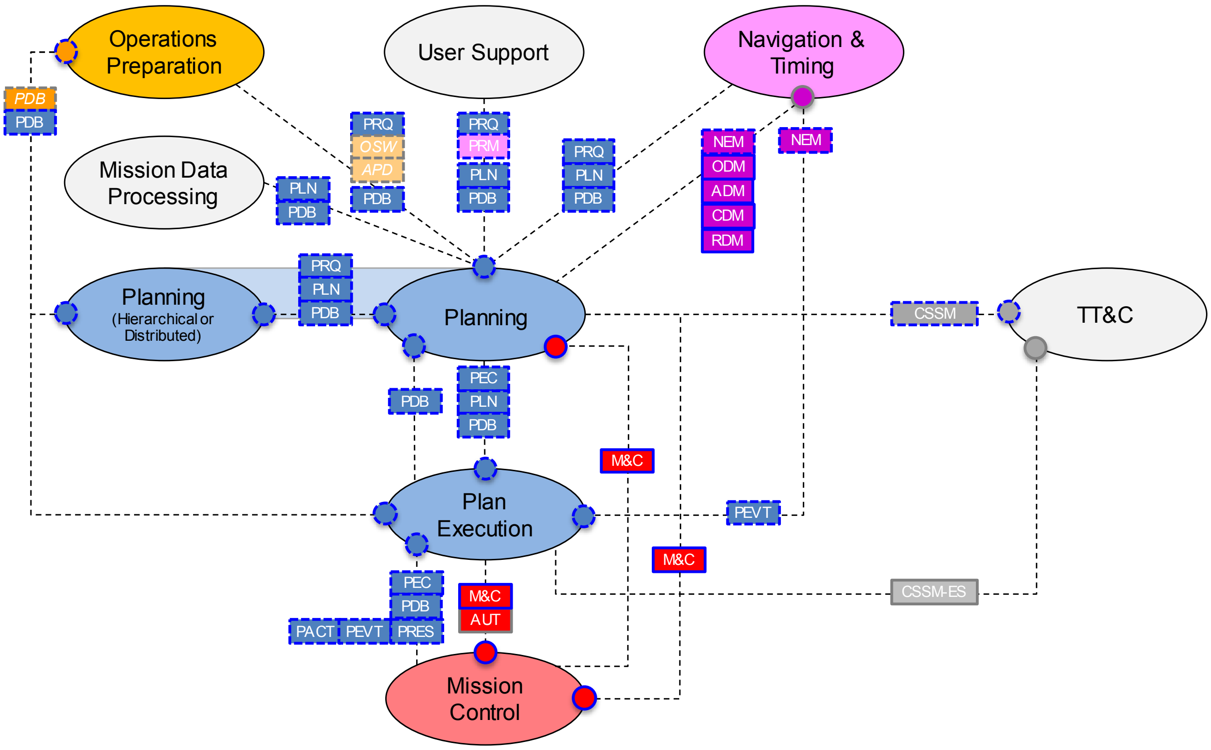


Figure 4-6: MO Level 2: Mission Planning and Scheduling Functions

The Mission Planning & Scheduling functional group is broken down into the following Level 2 functions:

- Planning;
- Plan Execution.

Planning itself may be hierarchical and distributed, while there may be multiple Plan Execution functions within the same system.

Mission Planning & Scheduling data comprises the following information objects:

- **PRQ**: planning requests [Future] (reference [63]);
- **PLN**: plans [Future] (reference [63]);
- **PDB**: planning database (definitions) [Future];
- **PEC**: Plan Execution Control Data [Future] (reference [63]);
- **PACT**: Planning Activities [Future] (reference [63]);

- **PEVT**: Planning Events [Future] (reference [63]);
- **PRES**: Planning Resources [Future] (reference [63]).

The Planning Database (PDB) contains the definitions of planning information objects specific to a given mission deployment. This includes the definitions of Planning request templates, planning activities, planning events, and planning resources.

Plan Execution Control is concerned with the monitoring and control of the Plan Execution function. Some associated information objects and services could therefore be expressed as specializations of generic MO Monitoring & Control, but there are dedicated operations associated with plans. Similarly, monitoring & control of the plan function may be exposed to enable automation using a specialization of the MO M&C service.

Planning activities, planning events, and planning resources are defined in the PDB and contained or referenced within planning requests and plans.

The following Mission Planning & Scheduling services are identified:

- **PRS**: Planning Request Service [Future] (reference [63]);
- **PDS**: Plan Distribution Service [Future] (reference [63]);
- **PIM**: Plan Information Management Service [Future] (reference [63]);
- **PEC**: Plan Execution Control Service [Future] (reference [63]);
- **PED**: Plan Edit Service [Future] (reference [63]).

The Planning Request Service (PRS) is concerned with the submission, update, and monitoring of planning requests.

The Plan Distribution Service is concerned with distribution and monitoring of plans.

The Plan Information Management Service is concerned with access to and update of planning definitions contained in a PDB.

The Plan Execution Control Service is concerned with control of the execution of plans.

The Plan Edit Service is concerned with direct manipulation of the planning activities, planning events, and planning resources within the currently executing plan(s).

4.2.7.2 Planning

Function

Planning is the function responsible for performing Mission Planning and Scheduling. Internally, it may be hierarchically organized and/or distributed. Planning requests are received from multiple users of planning services and feedback on their status provided. Other functions may also perform high-level control of the planning processes supported by the planning function. The output of the planning function is plans, which may be retrieved by planning users and distributed to Plan Execution functions. Planning may also control the execution of plans via the Plan Execution functions. Planning is itself a user of predicted orbital events and negotiates the scheduling of ground station support via CSS Service Management.

Provided Interfaces

- **PRS**: Planning Request Service (planning requests) [Future] (reference [63]).
- **PDS**: Plan Distribution Service (plans) [Future] (reference [63]).
- **PIM**: Plan Information Management Service (PDB) [Future] (reference [63]).
- **M&C**: Monitoring & Control Data (optional to enable automation of the planning process) (reference [17]).

Required Interfaces

- **PDB**: Planning Configuration Data [Future].
- **NEM**: Predicted Navigation Events Message [Future] (reference [64]).
- **ODM**: Orbit Data Messages (reference [25]).
- **ADM**: Attitude Data Messages (reference [27]).
- **CDM**: Conjunction Data Message (collision warning) (reference [28]).
- **RDM**: Re-entry Data Message (reference [65]).
- **CSSM**: CSS Service Management.
- **M&C**: Monitoring & Control Data (current mission status) (reference [17]).
- **PRM**: Pointing Request Message (reference [29])

It is noted that while it is possible for implementations of a Planning function to receive a PRM directly, pointing constraints derived from and compatible with the PRM may also be embedded within a generic Planning Request message of the PRS.

And in the context of hierarchical/distributed mission planning:

- **PRS**: Planning Request Service (planning requests) [Future] (reference [63]).

- **PDS**: Plan Distribution Service (plans) [Future] (reference [63]).
- **PIM**: Plan Information Management Service (PDB) [Future] (reference [63]).

Plan Information Management Service could be used by Operations Preparation to perform direct editing of the operational PDB for a Planning Function.

4.2.7.3 Plan Execution

Function

Plan Execution is the function responsible for executing a plan (or part of it). There may be multiple Plan Execution functions distributed between space and ground segments. It is not a planning function itself, but it does support a common model of the plan in its interface with planning. It receives or retrieves distributed plans, allows external control of the Plan Execution process, and provides execution status of the plan to Planning. Plan Execution may use underlying Mission Control Services to effect planned activities.

Provided Interfaces

- **PEC**: Plan Execution Control Service [Future] (reference [63]).
- **PIM**: Plan Information Management Service (PDB) [Future] (reference [63]).
- **PED**: Plan Edit Service (planning activities, planning events, and planning resources) [Future] (reference [63]).

Required Interfaces

- **PDB**: Planning Configuration Data (distributed by Operations Preparation) [Future].
- **PIM:** Plan Information Management Service (PDB) [Future] (reference [63]).
- **M&C**: Monitoring & Control Data (reference [17]).
- **AUT**: Automation [Future] (reference [57]).
- **NEM**: Predicted Navigation Events Message [Future] (reference [64]).
- **CSSM-ES**: CSS Service Management Event Sequence.

Plan Information Management Service could be used by Operations Preparation to perform direct editing of the operational PDB for a Plan Execution Function.

Similarly, Navigation & Timing could use the Plan Edit Service to directly update planning events.

4.2.8 OPERATIONS PREPARATION

4.2.8.1 Overview

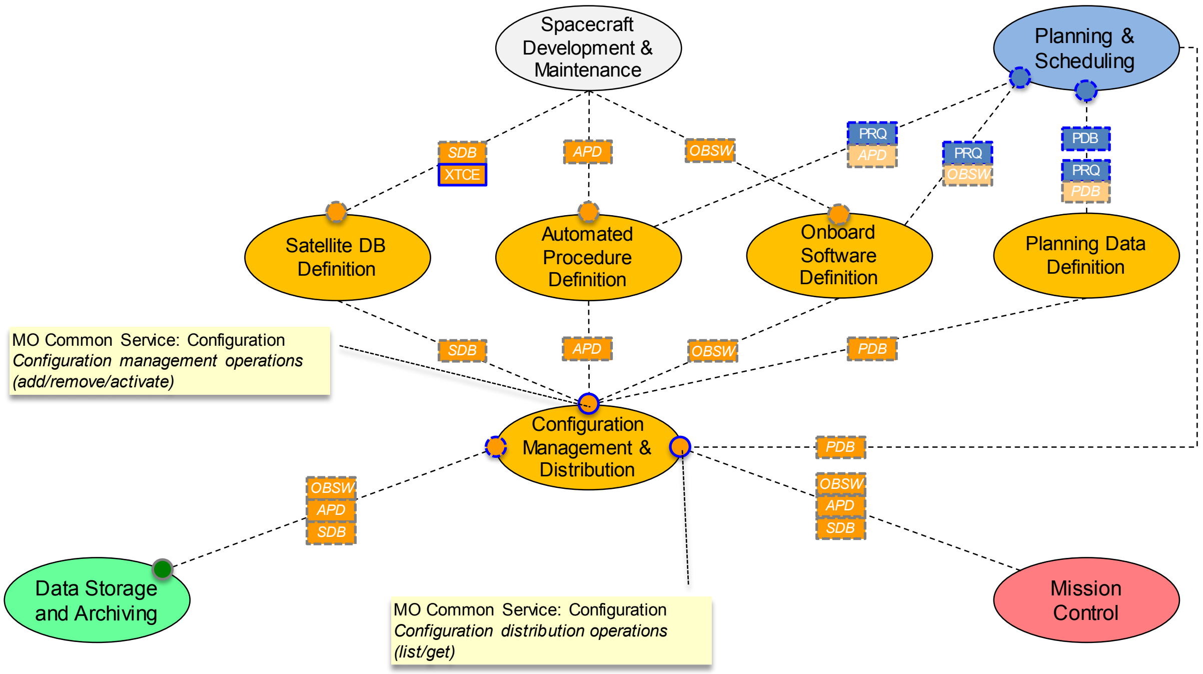

Spacecraft Development & Maintenance
Planning & Scheduling
SDB
XTCE
APD
OBSW
PRQ
APD
PRQ
OBSW
PDB
PRQ
PDB
Satellite DB Definition
Automated Procedure Definition
Onboard Software Definition
Planning Data Definition
MO Common Service: Configuration
Configuration management operations (add/remove/activate)
SDB
APD
OBSW
PDB
Configuration Management & Distribution
PDB
OBSW
APD
SDB
OBSW
APD
SDB
Data Storage and Archiving
MO Common Service: Configuration
Configuration distribution operations (list/get)
Mission Control

Figure 4-7: MO Level 2: Operations Preparation Functions

The Operations Preparation functional group is broken down at Level 2 into a Configuration Management and Distribution function plus a set of editor functions for defining each class of configuration data.

Operations Preparation Data comprises the following information objects:

- **APD**: Automated Procedure Definition [Prospective];
- **OBSW**: onboard software [Not planned for standardization];
- **PDB**: planning database (planning data definitions) [Prospective];
- **SDB**: Spacecraft Database (monitoring & control data definitions) [Partial] (reference [23]) [Prospective].

None of these information objects are currently standardized by CCSDS; however, they may be embedded in or referenced by standard services.

XTCE provides a standardized data representation for the exchange of telemetry and command definitions, which is a subset of the Spacecraft Database.

There is potential for the SO Electronic Data Sheets (reference [10]) to be integrated with, or provide input to, the SDB, although there are no current plans for standardization of how these relate to each other. In the above diagram, one possible approach would be for the Spacecraft Development and Maintenance function to supply EDS to the Satellite DB Definition, which then outputs an augmented SDB for use by Mission Control. Alternatively, SEDS could be used to support the exchange of the SDB.

4.2.8.2 Configuration Data Definition Editors

<u>**Function**</u>

Each class of mission operations configuration data has a dedicated function for its development and maintenance, which may be referred to as an Editor. These include:

- Spacecraft Database Definition (SDB) [Partial] (reference [23]) [Prospective];
- APD [Prospective];
- OnBoard SoftWare (OBSW) definition [Not planned for standardization];
- Planning Data Definition (PDD) [Prospective].

All of these interact with the Configuration Management and Distribution function to place versions of their data under configuration control. The configuration management operations of the MO Common Service for Configuration (reference [55]) can be used to support this.

The spacecraft (or payload) manufacturer typically provides at least initial versions of the spacecraft database, onboard automated procedure definitions, and onboard software. These may not be in a format compatible with the mission operations system and so require ingestion by the associated Operations Preparation Definition Editor. There is currently no CCSDS Recommended Standard for the exchange of data between the Spacecraft Development and Maintenance function and the Operations Preparation function, with the exception of XTCE.

Definition Editors that are responsible for configuration data that is to be deployed onboard the spacecraft (onboard software; automated procedures) may use the Mission planning requests or associated PRS to plan the upload of new versions of onboard configuration data.

For the specific case of the PDB, a Planning Data Definition Editor could make use of the MPS Planning Information Management Service (PIM) [Future] (reference [63]) to directly update the operational PDB within the Mission Planning and Scheduling function.

<u>**Provided Interfaces**</u>

- **SDB, APD, or OBSW** Ingestion [Partial], [Prospective], and [Not planned for standardization].
- **XTCE:** Telemetry and Command Exchange (partial SDB ingestion) (reference [23]).

Required Interfaces

- **MO Common: Configuration Service** (configuration management operations) (reference [55]) applied to **SDB, APD, OBSW,** or **PDB**.
- **PRQ:** planning request [Future] (reference [63]).

4.2.8.3 Configuration Management and Distribution

Function

The Configuration Management and Distribution function provides a configuration-controlled repository of Mission Operations configuration data, including version control and history for individual configuration data items, together with their current validation status and compatibility with versions of other configuration data items.

The function is also responsible for the distribution of configuration data versions to their target functions. For **SDB**, **APD,** and **OBSW** this is to the Mission Control function (which is itself responsible for managing onboard configuration of **APD** and **OBSW**); and for **PDB** this is the to the Mission Planning & Scheduling function. The configuration distribution operations of the MO Common Service for Configuration can be used to support this.

The storage of (historical) configuration data versions may be delegated to an external Operations Data Archive function.

Provided Interfaces

- **MO Common: Configuration Service** (configuration management and distribution operations) (reference [55]) applied to **SDB, APD, OBSW,** or **PDB**.

Required Interfaces

- **DSA**: Data Storage and Archiving services.

4.2.9 DATA STORAGE AND ARCHIVING

4.2.9.1 Overview

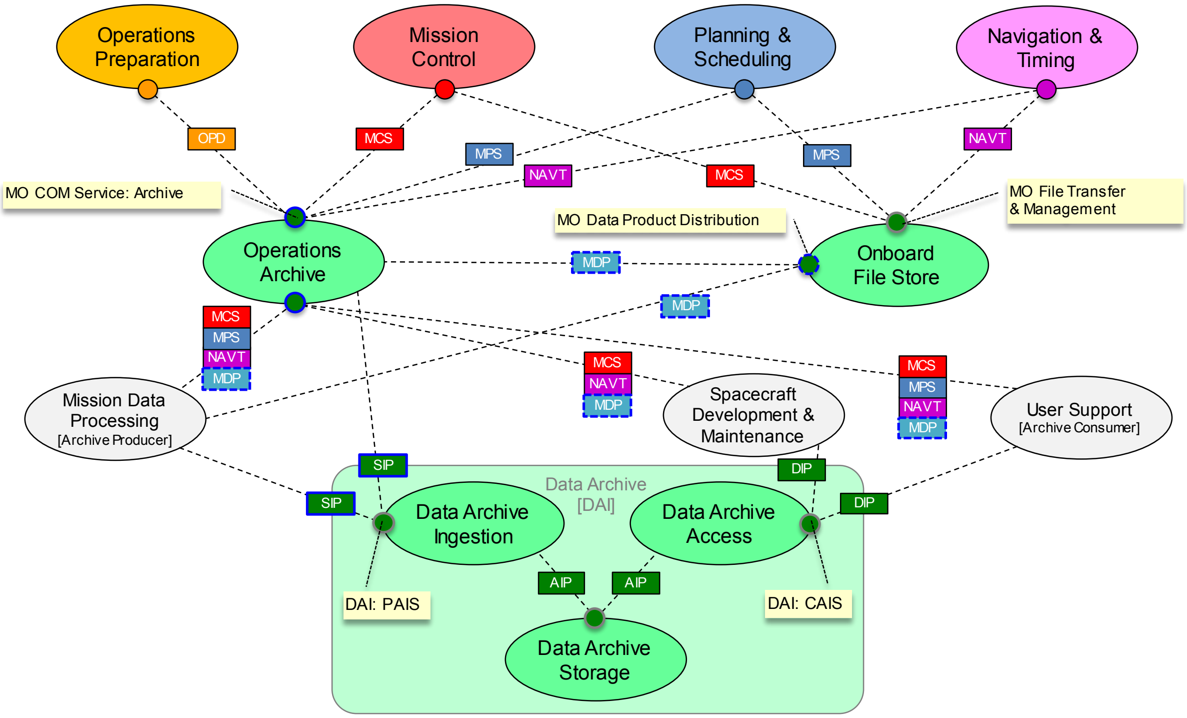


Figure 4-8: MO Level 2: Data Storage and Archiving

The Data Storage and Archiving functional group comprises three separate functions related to the storage and archiving of mission data associated with any of the other Functional Groups or Mission Data Products:

- Onboard File Store;
- Operations Archive;
- Data Archive.

The data subject to Data Storage and Archiving is that generated by other Functional Groups, but specific information objects are identified for the packaging of data associated with the Data Archive function:

- **AIP**: Archival Information Package;
- **DIP**: Dissemination Information Package;
- **SIP**: Submission Information Package.

Services are identified for the submission and dissemination of data to/from the Data Archive function:

- **CAIS**: Consumer Archive Interface Specification [Future] (reference [66]);
- **PAIS**: Producer Archive Interface Specification (reference [31]).

The current archive service specifications are of abstract services that are not directly implementable. Interoperable specifications are [Prospective].

4.2.9.2 Onboard File Store

Function

The onboard file store is a temporary onboard repository for files containing mission control, mission planning, navigation and operations preparation data, or mission data products, which may be generated onboard or uploaded from the ground. There are special services for File Management and File Transfer (the latter delegated to a lower level protocol, such as CFDP or FTP), that can be used by MO functions to interact with the file store. There is an existing CCSDS File Store Model contained within the CFDP Standard (reference [4]).

The CCSDS MO Data Product Distribution service may also be provided to disseminate Mission Data Products, either directly to an external Mission Data Processing function or to the Operations Archive.

Provided Interfaces

- MO File Transfer [Future] (reference [61]).
- MO File Management [Future] (reference [61]).
- MO Data Product Distribution [Future] (reference [56]).

Required Interfaces

- **MCS**: Mission Control Data (files) (reference [17]).
- **NAVT**: Navigation and Timing Data (files) (references [25], [26], [27], [28], [29], [Future] [64], and [65]).
- **MPS**: Mission Planning & Scheduling Data (files) [Future] (reference [63]).
- **MDP**: Mission Data Products (files) [Future] (reference [56]).

4.2.9.3 Mission Operations Archive

Function

The Mission Operations Archive is a ground-based repository of historical mission operations data of all types. This is used for rapid storage and retrieval by Mission Operations functions. It may persist for the mission lifetime.

The MO Data Product Distribution Service may be used for dissemination of Mission Data Products.

Provided Interfaces

- **Archiving and Retrieval** of MCS, MPS, NAVT, OPD, and MDP data. The **MO COM Archive** service (reference [16]) may be used for data compliant with the MO Common Object Model.
- **MO DPD**: Data Product Distribution [Future] (reference [56]).

Required Interfaces

- **MCS**: Mission Control Data (reference [17]).
- **MPS**: Mission Planning & Scheduling Data [Future] (reference [63]).
- **NAVT**: Navigation & Timing Data (references [25], [26], [27], [28], [29], [Future] [64], and [65]).
- **OPD**: Operations Preparation Data (see 4.2.8 for details).
- **MDP**: Mission Data Products [Future] (reference [56]).
- **PAIS**: Producer Archive Interface (reference [31]).

4.2.9.4 Mission Data Archive

Function

The (long-term) Mission Data Archive is a ground-based repository of mission data products with ancillary mission operations data intended to ensure long-term preservation of data and to provide access to external users.

It may be further decomposed into three subfunctions:

- Data Archive Ingestion;
- Data Archive Storage;
- Data Archive Access.

Data to be archived is submitted to the Data Archive Ingestion function as a Submission Information Package (SIP), using a service conforming to the Producer Archive Interface Specification (PAIS). SIPs are restructured (typically collated into larger data items) as Archival Information Packages (AIPs), that are stored in the Data Archive Storage function.

AIPs may be subsequently retrieved by the Data Archive Access function, which restructures them for dissemination to users as Dissemination Information Packages (DIPs). Users interact with the Data Archive Access function through a service conforming to the Consumer Archive Interface Specification (CAIS).

Provided Interfaces

– **PAIS**: Producer Archive Interface (reference [31]).

– **CAIS**: Consumer Archive Interface [Future] (reference [66]).

The current archive service specifications are of abstract services and processes that are not directly implementable. Interoperable specifications are [Future].

4.3 SO FUNCTIONS

4.3.1 OVERVIEW

This subsection addresses functionality within the scope of the SO functions.

Functional groups are differentiated by color coding as introduced in 3.3.1.

The remainder of the subsection is structured as follows:

a) **SO Context**: scope of the SO capabilities;

b) **SO Functions**: functional decomposition of SO capabilities;

c) **Protocol Convergence Functions**: future convergence functions for subnetworks;

d) **Management Functions**: future functions to expose Management Information Bases (MIBs) as services.

For descriptions of the information objects and services that are also shown in the diagrams, reference should be made to the Information and Service Viewpoints, respectively.

4.3.2 SO CONTEXT

Figure 4-9 is a summary of the SO functions. Subsequent diagrams throughout this publication tease apart the details into more comprehensible units of exposition. SO services consist of application support, subnetwork, and wireless services in an onboard computing platform.

The 'Applications' function in figure 4-9 includes any onboard application functions that use spacecraft onboard services, information, communications, and services of onboard devices. Any MO functions that get migrated onboard are also included in this set. The figure arranges the onboard functions in four layers, which may be related to identified layers in the OSI model. The application support functions are a part of what is normally considered the Application Layer, and the subnet functions are aligned with the ISO Data Link Layer. There are a wide variety of protocol capabilities of onboard subnetworks, and these functions are designed to expose a uniform set of functions to onboard applications and support functions. Some of these subnet functions may be integrated into the onboard computing platform, as is explained in 7.4.

In figure 4-9, SO standardized functions are green or orange, with interface circles and usage lines in the same color. SO support of standardized functions consists of the following tools, in addition to those of non-standardized functions.

- SEDS can describe the interfaces and internal implementation of those functions.
- An agency tool chain can generate software from SEDS that implements those functions.

The SO architecture supports exploitation both of non-standardized functions and of functions standardized outside the scope of SO in the protocol stack between devices and applications. In addition, the SO architecture includes non-standardized application support functions outside the protocol stack. Such functions are called 'non-SO' here. In figure 4-9, these non-SO functions are grey. Non-SO functions have the following characteristics.

- They may be pre-existing functions whose replacement by standardized functions would be too expensive.
- They may be newly written functions whose innovation would be stifled by standardization.
- They may implement standards outside the scope of SO.

Support for non-SO functions in the onboard environment consists of the following approach.

- SEDS can describe the interfaces of those functions: The SEDS-described interfaces provided by non-SO functions are blue circles in figure 4-9.
- An agency tool chain can generate software artifacts from SEDS to enable other functions to use those functions: A function's use of an interface of a non-SO function is represented in figure 4-9 by a blue dashed line from the using function to the interface of the providing function.

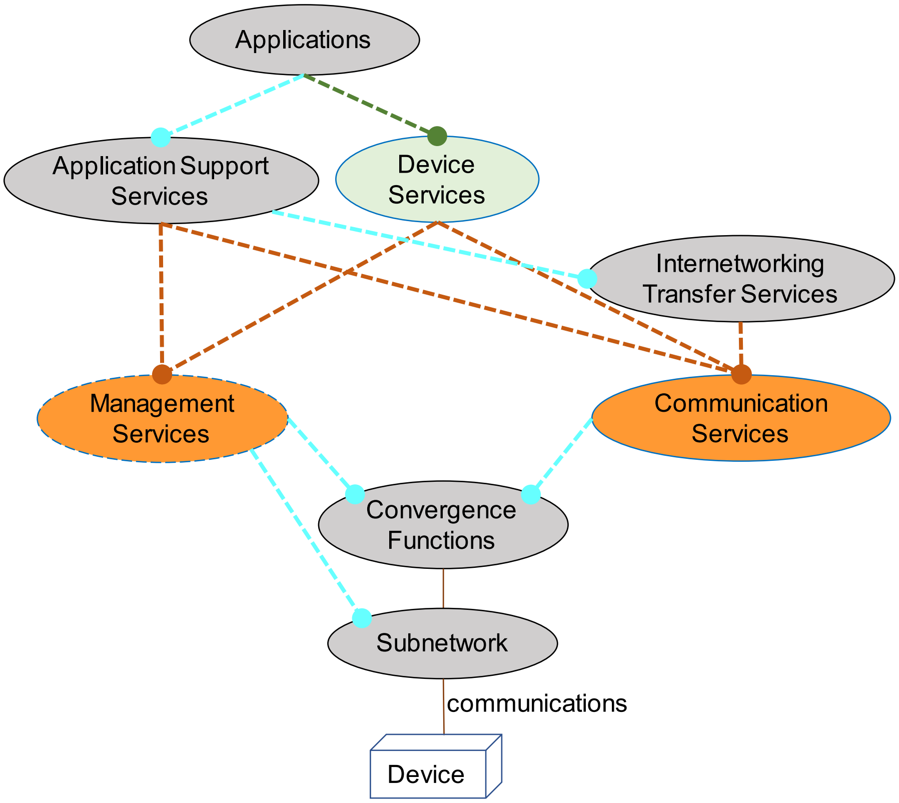


Figure 4-9: Summary of SO Functions

Network and Transport Layer protocols may be utilized in some onboard environments as shown by the non-SO function Internetworking Transfer Services in figure 4-9, in which case they will typically be layered, as is usual, above the subnet protocols and below any applications. For example, a message transfer service, which could be a subset of Asynchronous Message Service (reference [41]), can use internetworking transfer services to provide a software message bus onboard.

The Subnetwork functions support a variety of onboard Data Link Layer technologies, including SpaceWire, 1553 varieties, CAN bus, and wireless. There is a set of standards that correspond to the ISO OSI Physical Layer and the Medium Access Control (MAC) sublayer of the Data Link Layer (reference [13]).

A future objective is to integrate the wireless monitoring and control subnetwork with SO subnetwork functions, so wireless instruments are accessible by flight software applications. [Future].

4.3.3 SO SUBNETWORK FUNCTIONS

The SOIS Green Book (reference [51]) describes services that facilitate communication between spacecraft applications and onboard devices. Figure 4-10 summarizes those services.

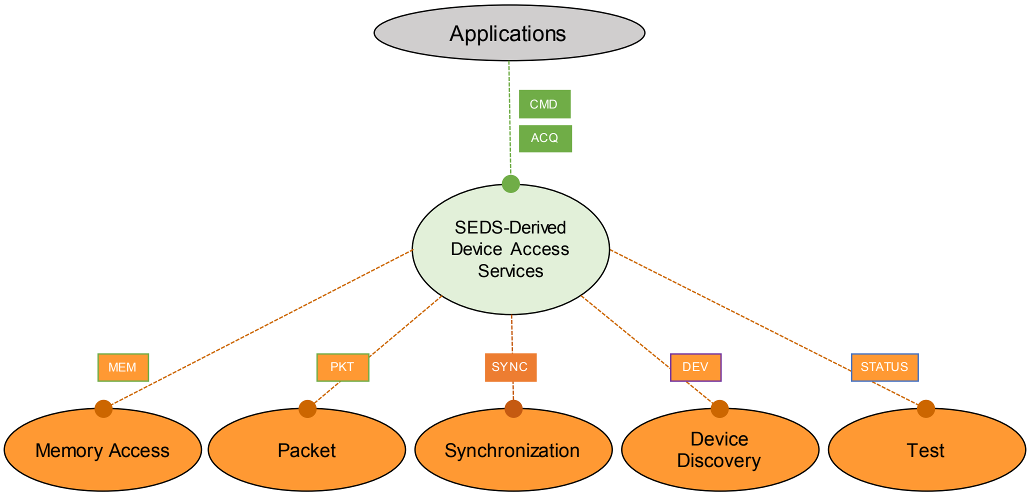


Figure 4-10: Functional Summary of SO

The diagram of SO subnetwork functions in figure 4-10 has the following three layers.

- The spacecraft applications that use SO services appear as a generic function at the top of figure 4-10.
- A set of device services, which are derived from SEDS, provide Command and Data Handling (C&DH) functions for onboard devices.
- The device services utilize SO subnet functions to communicate with devices connected via a variety of different onboard subnetwork technologies.

The SEDS description of device access services includes a provided interface, which is used by applications, and one or more required interfaces, which identify subnetwork functions.

4.3.4 SO APPLICATION SUPPORT FUNCTIONS

4.3.4.1 General

The SOIS Green Book (reference [51]) defines a set of application support functions (shown as dashed boxes in figure 4-11), which were temporarily retired during the development of SEDS. The intention now is to revive those application support functions by describing them with SEDS, accompanied by Recommended Practices for implementation. The interfaces described abstractly in the Recommended Practices are augmented with SEDS descriptions. The Recommended Practices are for SO implementations written by hand without SEDS, and the SEDS descriptions are for SO implementations using SEDS and a tool chain to generate software interfaces [Future].

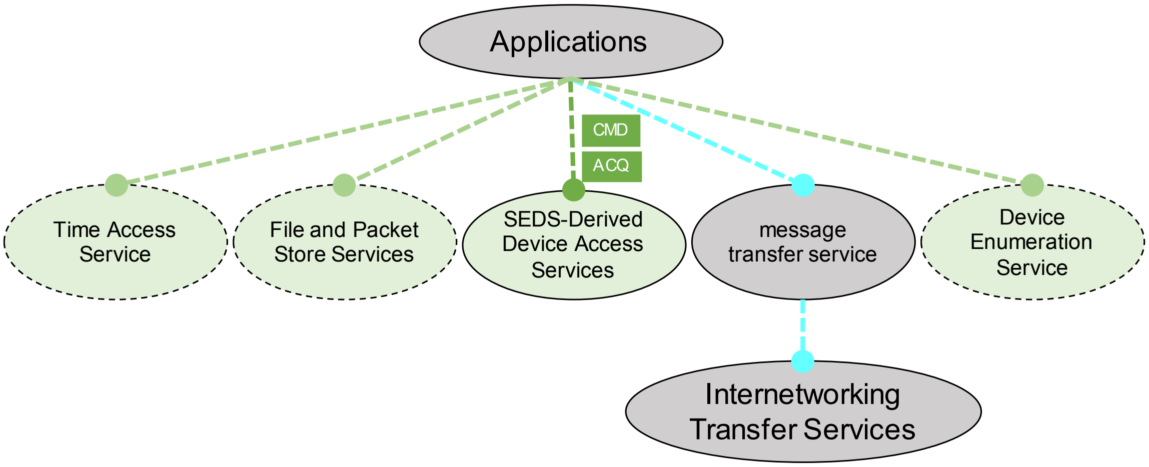

Applications
CMD
ACQ
Time Access Service
File and Packet Store Services
SEDS-Derived Device Access Services
message transfer service
Device Enumeration Service
Internetworking Transfer Services

Figure 4-11: SO Application Support Functions

The application support functions include the following capabilities.

- Command and Data Acquisition Services: This has been revived, and it is the current scope of SEDS. It appears in figure 4-11 as 'SEDS-Derived Device Access Services'. More on this topic can be found in 4.3.4.3.
- Time Access Service [Future]: This service will provide an interface for setting and indicating time-correlated events for synchronizing applications. Time synchronization is expected to be accessible for most clock devices through command and data acquisition services derived from SEDS.
- File and Packet Store Services [Future]: These services will provide an interface required by MO functions. (See 7.3.6.)
- message transfer service: The name of this function is all lowercase because it represents a generic concept that has multiple possible implementations. A simple function that routes space packets, using the Application Process Identifier (APID) to designate a logical data path, is an example implementation. This function has been described as a subset of AMS. The subset of AMS for this function is described by SEDS [Future], but an accompanying Recommended Practice is unnecessary because of the existence of AMS books.
- Device Enumeration Service [Future]: This service will provide an onboard searchable database of devices managed by SO. The subnetwork Device Discovery Service populates the database.

4.3.4.2 Prospective Application Support Services

An additional set of application support services that are in consideration for description by SEDS are the following examples. All of these are [Prospective] and there are not yet firm plans for producing them. The names may change when any of these prospective functions are adopted for publication. (See 4.3.2 and 10.3.3 for description of using SEDS to integrate non-SO application support functions._

- CCSDS File Delivery Protocol transfers files between vehicle and ground.
- Checksum functions checks data integrity of memory, data, and files.
- Data storage records housekeeping, engineering, and science data for later download.
- File manager provides interfaces to the ground for managing files onboard.
- Housekeeping collects and packages telemetry from onboard applications.
- Health and safety ensures that critical tasks check in, services watchdog, detects CPU hogging, and calculates CPU utilization. Management of platform resources by SO applies to the processor(s) in which real-time activities occur.
- Limit checker monitors values and takes action when they cross thresholds.
- Diagnostic telemetry allows ground to telemeter the contents of memory locations, generally for debugging.
- Memory manager loads and dumps memory, on request.
- Scheduler schedules onboard activities, such as housekeeping requests.
- Onboard control procedure executes small script-like sequences of commands at absolute or relative times with flow control logic.
- Event-action service allows to trigger onboard action based on an event generated onboard.
- Critical event log allows generation and retrieval of critical onboard events.

4.3.4.3 SEDS-Derived Device Access Services

The SEDS-Derived Device Access Services consist of two functions, as shown in figure 4-12. There is one pair per device.

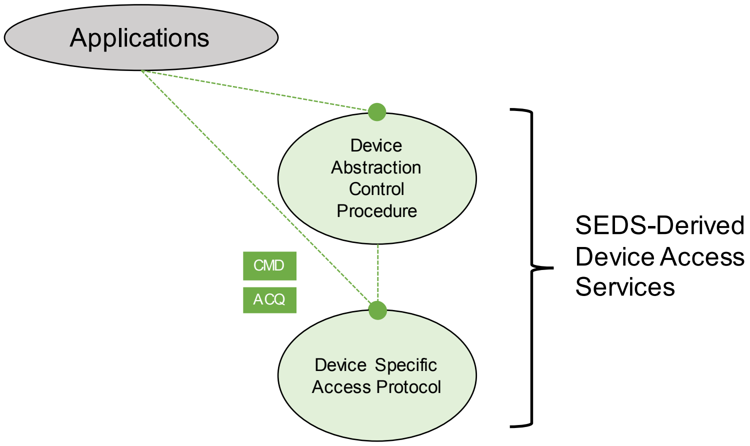

Applications
Device Abstraction Control Procedure
CMD
ACQ
Device Specific Access Protocol
SEDS-Derived Device Access Services

Figure 4-12: Functional Decomposition of SO Device Service

Each Device Abstraction Control Procedure (DACP) provides the functional interface for a device and hides the device specifics by mapping the functional interface onto the device-specific interface. The mapping may include transformations like the following.

– conversion between engineering units of measure and raw counts of analog-to-digital converters;

– implementation of the simple commands of a functional interface by procedures that consist of more than one device-specific command.

Each Device-Specific Access Protocol (DSAP) provides the device-specific interface for a device and hides the subnetwork specifics by mapping the device-specific interface onto the subnetwork communications services interfaces. The mapping may include sequential activities and event-driven state machine logic.

Applications may use the interface provided by the DACP or the interface provided by the DSAP. The DACP is optional onboard, while the greater efficiency of direct device-specific protocols is typically necessary. When the DACP is absent onboard, any applications have to use the DSAP directly. However, when a mission operations proxy is present onboard, the DACP is needed to translate between the functional interface required by the mission operations proxy and the device-specific interface provided by the DSAP.

NOTE – The aggregation of onboard DACP instances was also known as ‘Device Virtualization Service’ in older SOIS publications. Similarly, the aggregation of onboard DSAP instances was also known as ‘Device Access Service’ in older SOIS publications.

4.3.5 PROTOCOL CONVERGENCE FUNCTIONS

The preceding sections have described the application support functions and the subnetwork functions. Some 'glue code' is often needed between the SO subnetwork functions and the Data Link Layer. The subnets that are typically used onboard have a wide range of different features and interaction modes. Some have a highly controlled master/slave hierarchy; others are first come/first served; others may embed Network Layer or timing functions within what is normally considered a Data Link Layer protocol. In order to deal with this diversity, one of the features of SO is the capability to compose a stack of 'convergence' functions to provide uniform types of services across a variety of subnetworks that individually provide different levels of service. The convergence functions that are described in the SOIS Green Book (reference [51]) appear in figure 4-13. The grey color indicates that these are non-standardized functions, so only their interfaces are described by SEDS instances [Future].

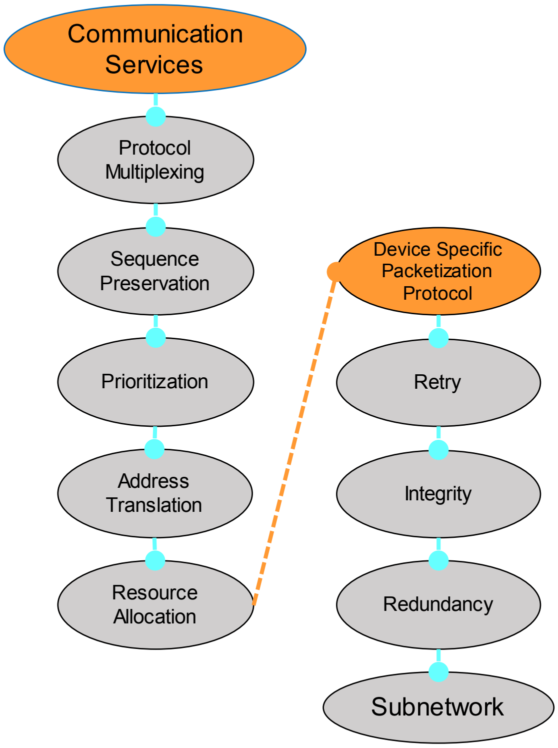


Figure 4-13: SO Convergence Functions

Depending on the characteristics of the particular subnet, the convergence layer may include, or possibly eliminate, some of these convergence functions. For instance, multiplexing, sequence preservation, and segmentation may already be present in the subnet design, so they do not need to be included in the convergence layer for that subnet. Other subnets may require a very different set of convergence layer features because their native service set is so different.

The Device Specific Packetization Protocol (DSPP) is shown in color because it can be generated from a SEDS instance. For example, this function may provide specialized framing of packets in a MIL-STD-1553 subnetwork.

4.3.6 MANAGEMENT FUNCTIONS

The ‘Management Services’ shown in figure 4-9 contain a collection of functions that are not directly in the protocol stack for data communications, but which define and control the paths for data communications. Figure 4-14 shows the ‘management services’ inside a dashed box, which indicates that they are outside the protocol stack.

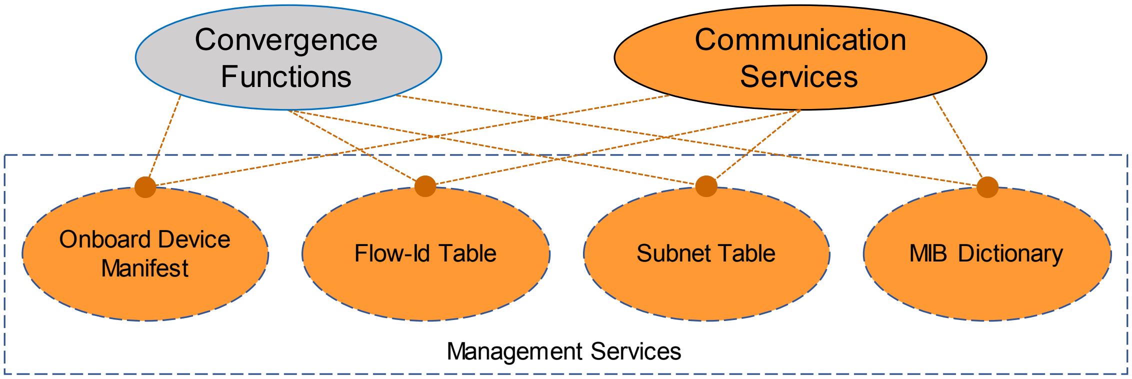


Figure 4-14: SO Management Services

The following additional management functions have been identified:

- onboard device manifest [Future];
- FLOW-ID collection [Future];
- subnet collection [Future];
- a set of single-valued management parameters have that can be specified at design time in a mission-parametric SEDS and can be stored in a ‘MIB Dictionary’. (See 5.3.7.)

The purpose of identifying management services above is to facilitate design of access to data needed to define and to control data communications onboard a spacecraft. The description of these as services identifies the most flexible and demanding implementation, but not necessarily the most efficient and reliable way to satisfy the requirements of a mission.

The choice of implementation depends upon the phase in development of a mission when the management data assumes fixed values. The following phases affect the choice.

- Design is the phase in which the choices of implementations are made.
- During compile-time, the flight software is encoded for execution in an onboard computer. Management parameters whose values will not change subsequently can be encoded as definitions that compile directly into the flight software as parts of instructions or as static data.

- At the time that an onboard computer boots, which may occur during integration or just after launch or after recovery from an abnormal condition such as safe-mode, the computer may read some management parameters from an onboard memory device, such as one-time programmable non-volatile memory. This implementation allows for choices of values for management parameters up to the time of delivery of the onboard computer and memory before integration.
- During flight, spare devices may be activated to replace failed devices, and FLOW-IDs may be composed to accommodate flexible Quality of Service (QoS) during different mission tasks. On vehicles designed to be serviced in flight, onboard devices may be replaced or added during a mission. The corresponding values of management data can be implemented through API or message-passing interfaces to management services or by updating values in programmable shared memory by management services.

Summarizing the choices above: The management services may be implemented as actual services if changes in management data are expected in flight. The management services may be implemented as static arrays or as compile-time definitions if changes to their content do not occur in flight. Designers must review the requirements for access to data in each management service to determine an appropriate implementation for each service.

4.4 SECURITY CONCEPTS FOR FUNCTIONAL VIEWPOINT

There is not an extensive suite of MO or SO security features at a functional level. The assumption is that terrestrial service interfaces will be secured, and that is addressed in the Service Viewpoint. The MO functions do include an optional access control function that supports control over user access to a set of functions and/or domains. Some of the SO functions identify an ‘authorization’ data item in the interface for memory access, which is an authentication key; authorization would be done outside the interface. Space and time partitioned RT operating systems may offer the needed features if they are employed.

There are some specific security features that may appear in Functional Viewpoints of ASL systems:

a) Access control: The process of granting access to the resources of a system only to authorized users, programs, processes, or other systems.

b) Authentication: The process of verifying the authorization of a user, process, or device, usually as a prerequisite for granting access to resources in an IT system.

c) Authorization: The granting of access rights to a user, program, or process.

d) Encryption: The cryptographic transformation of data (see cryptography) to produce ciphertext.

e) Cryptography: The discipline that embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification, and/or prevent its unauthorized use.

The MO TT&C interfaces that are used to communicate with spacecraft typically use access control at the function/service interfaces and often also utilize Physical and Network Layer security mechanisms. (See the relevant sections of the SCCS-ADD, reference [50], for further descriptions of these security practices.) The operational facilities, and the systems themselves, running the MO MPS, MCS, NAVT, and OPD functions are also typically secured, both physically and electronically.

Many spacecraft are not directly secured in the sense of using any kind of overt access control or authentication mechanisms, and the SO functions do not define any such mechanisms. For the underlying communications and cross support services a variety of security protocols, including authentication and encryption have been defined. These are all available for use terrestrially, over the space link, and onboard, but the SO and MO functions have not defined any particular usage of these functions.

5 INFORMATION VIEWPOINT (INFORMATION OBJECTS)

5.1 OVERVIEW

The details of the information exchanged across interfaces between functions are the subject of the Information Viewpoint, which models the principal information objects and the relationships between them, including:

- inheritance;
- composition;
- aggregation;
- other associations.

5.2 MO INFORMATION OBJECTS

5.2.1 GENERAL

This section addresses data or information objects within the scope of the MO capabilities. This information model is only introduced at a relatively high level, sufficient to identify the information exchanged between functions and any relationships that exist between information objects exchanged across multiple interfaces. For a full and detailed specification of the referenced information objects, the reader is directed to the relevant CCSDS Recommended Standards.

Some information objects correspond to complex file-based schemas that are often relatively self-contained and self-documenting. Other information objects are simpler in data structure but exposed at service-based interfaces in which they have associated static or dynamic behavior: including reporting status and being subject to discrete operations that may affect their state.

The remainder of the section is structured as follows:

a) **MO Information Groups**: top-level decomposition of MO related Information;

b) **MO Common Object Model**: introduction to the generic information model for MO functions;

c) **Common Services Data**: applicable across multiple functional groups;

d) **<Functional Group>**: a set of subsections, one per MO Functional Group, providing the second level decomposition into information objects.

Each subsection comprises an Information Viewpoint diagram and a description of each of the information objects it contains. It should be noted that some smaller information groups are expanded within the top-level composition diagram.

5.2.2 MO SERVICE CONFIGURATION DATA

The use of MO services requires applications to have access to registers of object types, catalogues of available services and mission-specific configuration data that define the available set of objects to be referenced within the service.

This configuration data can be considered to be organized into three levels:

1. A registry of the identities of MO object types defined in MO service specifications, corresponding to the object numbers and other fixed identifiers used by that standard. This may include service defined Events and their associated text description.

2. A directory of available services within a specific system: this catalogues the services available with their service provider, access and binding details, supported version and capability sets and scope in terms of domain. This information is deployment specific.

3. Configuration data required to use a specific instance of a service. This information is mission specific and includes, for example, the definitions of available parameters, actions and events for an instance of the MO M&C service supporting a specific spacecraft.

The Space Assigned Numbers Authority (SANA) at https://sanaregistry.org is the registrar for all registries created under CCSDS, and hosts the registry for the first level of MO service configuration data. The MO Service Directory [Future], one of the MO Common Services, provides the second level (see 5.2.5.2. below).

The configuration data specific to a service instance is represented here as Operations Preparation Data (see 5.2.10 below), and is managed by the Operations Preparation function.

5.2.3 MO INFORMATION GROUPS

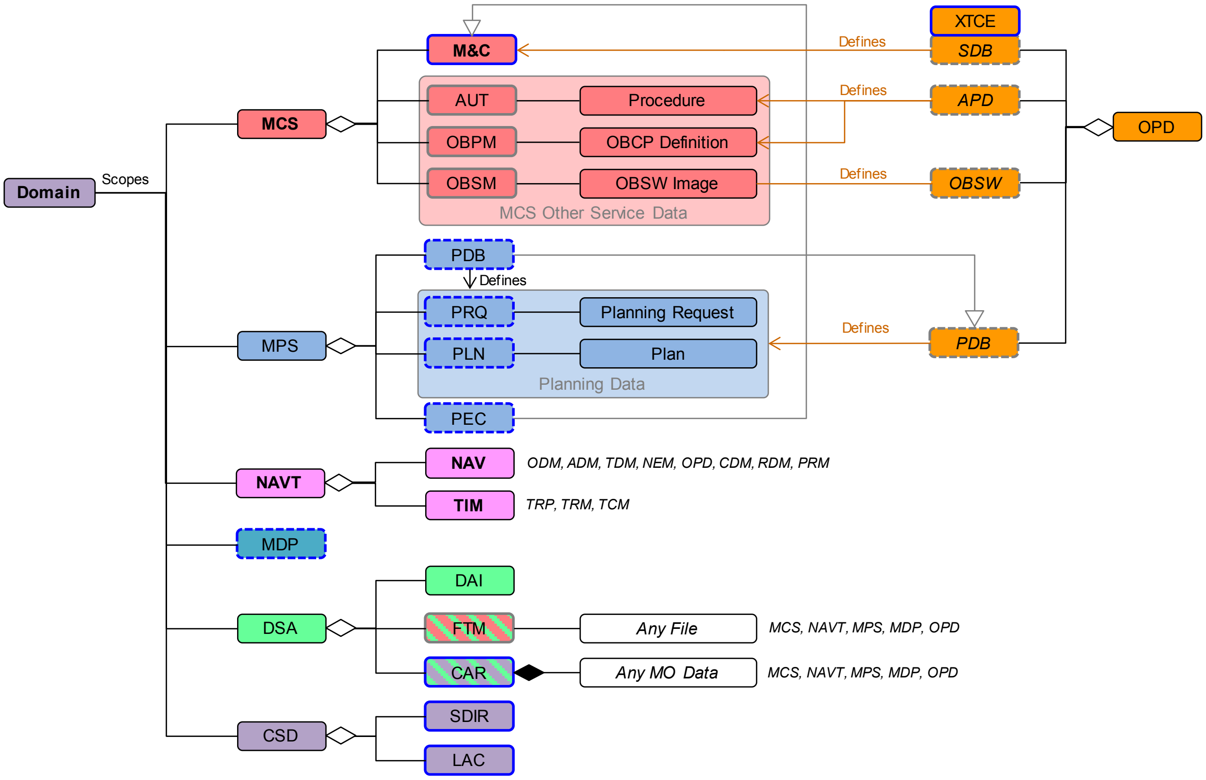


Figure 5-1: MO Information Groups

MO Information is grouped into the same functional groups identified in the Functional Viewpoint, but with an additional group relating to Mission Data Products that originate outside the defined MO functions, but are exchanged with MO functions:

- MCS;
- NAVT, which is further decomposed into:
  - Navigation (NAV),
  - Timing (TIM);
- MPS;
- MDP;
- OPD;
- Data Storage and Archiving (DSA), which comprises:
  - DAI,
  - File Transfer and Management (FTM), MO service for management of remote file stores,
  - Common ARchive (CAR), part of the MO COM;
- Common Services Data (CSD).

The top-level information object associated with each of these groups is essentially an aggregation of lower-level information objects that are defined within specific CCSDS Recommended Standards. The current standardization status of these lower level information objects is identified in the following subsections that describe each area in turn.

The MO service framework includes both a COM and Common Services, for which more detail is provided in the following two subsections.

OPD relates to the configuration data required for other functional groups and defines the information objects that exist within those groups. For this reason, it has been fully expanded in the top-level diagram above to show how its information objects relate to those of other groups. This applies primarily to Mission Control and Mission Planning for which configuration data has been identified, but could be extended to cover Navigation and other information groups if specific configuration data is identified for those groups.

NAVT comprises a set of defined message formats, listed in the diagram but detailed in the dedicated section below.

5.2.4 MO COMMON OBJECT MODEL

5.2.4.1 Overview

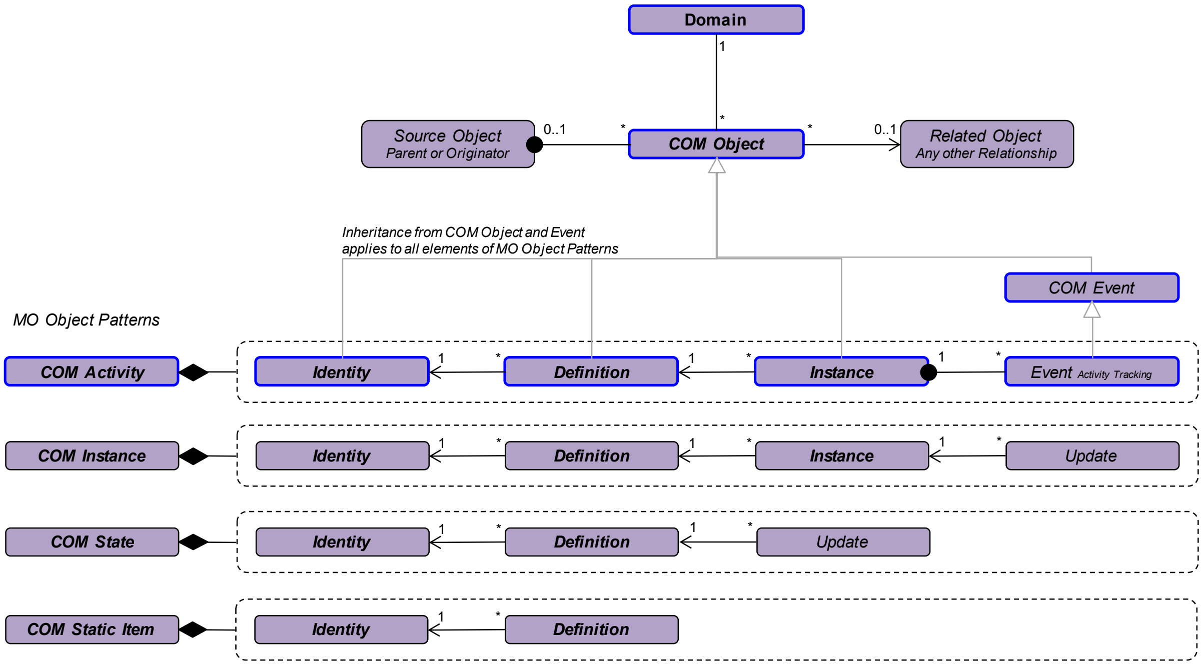


Figure 5-2: MO Common Object Model and Object Patterns

The MO service framework includes a COM from which the information objects associated with an MO-compliant service specification must be derived.

5.2.4.2 Domain

The majority of MO Information (whether derived from the COM or not) may be scoped by the concept of domain, which provides a namespace context for the identity of an information object. Domains may be hierarchical and are intended to correspond to real-world entities such as a space mission, a spacecraft (or ground system) within that mission, or its component subsystems and equipment. An example usage for a constellation mission would be to use domain to identify a specific spacecraft within the constellation.

5.2.4.3 MO COM Object

At the core of the MO COM is a root abstract class, the COM object (see figure 5-2), from which MO service information objects may be derived.

COM objects have:

- An **identity**, including its domain.
- A reference to a **Source** object that may be the originator or parent of the object. This can be used to establish an audit trail of operations, for example, from planning request > planned activity > automated procedure > M&C activity (command).
- A reference to a **Related** object, the meaning of which is specific to the derived information object. For example, an instance of an object may use this to point to its definition.

Any information objects defined in terms of the COM can be packaged as Common Archive (CAR) data using the generic MO COM Archive service (reference [16]) (see 6.2.2). Since much mission data may not be in the COM natively, the MO architecture allows packaging of any data within a top-level COM object, but if specialized indices and search criteria are desired, it is necessary to define a new custom service.

5.2.4.4 MO COM Event

The COM event is itself derived from the COM object but additionally has a Time attribute that represents its occurrence at a specific point in time. It is defined as a specific object representing ‘something that happens in the system at a given point in time’. Any information objects derived from the COM event can also make use of the generic COM event service. Each MO service can define the specific events that it supports.

A registry of assigned COM Event identifiers and text messages is used to associate specific service events with their associated text messages.

5.2.4.5 MO COM Object Patterns

5.2.4.5.1 General

The information objects associated with MO-compliant services are often implemented as a compound set of objects, each of which is derived from the MO COM object. These compound objects often follow one of a number of common object patterns. Some of these patterns are themselves standardized within the MO Common Object Model, while others are inferred here from repeated patterns found in standard MO services.

Four such COM object patterns are illustrated in figure 5-2:

- COM activity;
- COM instance;
- COM state;
- COM static item.

These share the following common elements:

- A unique identity that may be used to reference all occurrences of the information object (compound object) throughout the mission lifetime. This is a combination of the domain and a unique name within the domain.
- A definition that comprises the statically declared information associated with the information object. This may, for example, include a description, set of defined arguments, or any other information that applies to all occurrences of the information object. There may be multiple definitions (versions) over the mission lifetime associated with the same identity, each with its own unique definition ID. Definitions are typically contained in configuration databases that are maintained offline under version control and deployed for use in the online environment.

  The **Related** object reference of a definition points to its identity.
- An instance that maintains the current status associated with the information object or a specific occurrence of it. This may, for example, include a current value or set of values for defined arguments. Each occurrence of the information object has a separate instance, with its own unique instance ID.

  The **Related** object reference of an instance points to its definition.
- An Update that represents the current state of an information object at a specific point in time and can contain multiple dynamically changing attributes. Updates may be used to disseminate changing status and to record the detailed status history of the information object.

The **Related** object reference of an Update points either to its definition or to an instance, depending on whether the information object is dynamically instantiated or not.

- In the specific case of the COM activity pattern, an activity event (derived from COM event) represents the occurrence of one of a standard set of events. This represents changes in state of an activity instance. Activity events may be used to disseminate changing status and to record the detailed status history of the Activity.

  The **Source** object reference of the event points to the associated instance.

For specific information objects, according to which pattern it follows, there is a specialized object derived from the COM object associated with each of these elements.

5.2.4.5.2 COM Activity Object Pattern

A COM activity is a compound information object representing any type of operation that is repeatable and extends over a measurable period of time. An example is M&C actions.

COM activity comprises four of the elements described above: identity, definition, instance, and event. As a COM activity may be invoked multiple times, there may be multiple instance objects created from the same definition object.

The COM activity is formally specified within the MO Common Object Model, together with an associated generic COM activity tracking service that enables distribution of the evolving status of the COM activity.

5.2.4.5.3 COM Instance Object Pattern

A COM instance is a compound information object that can be dynamically instantiated. Like the COM activity, this can be used to represent operations that are repeatable and extend over a measurable period of time. During the lifetime of the operation, multiple attributes may be dynamically updated, not just its status. Examples are M&C procedures, MPS activities, and MPS events.

COM instance comprises four of the elements described above: identity, definition, instance, and update. Updates reference the instance to which they relate.

5.2.4.5.4 COM State Object Pattern

A COM state is a compound information object representing a status: it is a persistent object for which there is only one status value at any given time (although this may not be known). Examples include M&C parameters and MPS planning resources.

COM state comprises three of the elements described above: identity, definition, and update. As it is persistent rather than instantiated, Updates reference the definition directly; there is no instance object.

5.2.4.5.5 COM Static Item

A COM static item is a compound information object that only comprises statically declared information with no evolving status. Examples are M&C checks and conversions, and MPS PRQ Templates.

COM static items therefore only comprise the identity and definition elements described above.

5.2.5 MO COMMON SERVICES DATA

5.2.5.1 General

MO Common Services Data is applicable across multiple functional groups as it relates to the MO Common Services (Directory, Login, and Configuration) that can be used in conjunction with any other MO service. Configuration data is closely associated with the Operations Preparation functional group and detailed in 5.2.10 below.

The following information objects (shown in figure 5-1) are specific to Common Services:

5.2.5.2 Service Directory (SDIR)

The Service Directory (SDIR) (reference [55]) is a list of all MO services available within a given system, together with the following information:

- service ID and version;
- description;
- service scope (by domain);
- supported capability sets/operations;
- service provider(s).

The Service Directory may be updated and interrogated using the MO Common Directory service.

5.2.5.3 Login and Authentication Credentials

Login and Authentication Credentials (LAC) (reference [55]) are the information associated with user access control and include:

– user ID;

– password;

– access rights (by service, domain, and capability set/operation).

The information is used in conjunction with the MO Common Login service. This login service is the only security-related service that is presently offered as an MO service. It may use CCSDS credentials as described in (reference [38]) or other means. Security, including encryption, may also be applied at other layers, such as Data Link Layer or Network Layer security in terrestrial or space deployments. It is also possible to use Application Layer encryption of service data, in which one or more fields of the service message are provided as encrypted blobs of binary data.

5.2.6 MISSION CONTROL DATA

5.2.6.1 Overview

Mission Control Data is closely associated with the Mission Control functional group and its subfunctions. As the information model associated with the M&C function is relatively complex, this is addressed separately below.

5.2.6.2 Monitoring and Control Data

5.2.6.2.1 General

The following diagram shows the top-level information objects associated with the Monitoring & Control function and the CCSDS MO Monitoring & Control services (reference [17]).

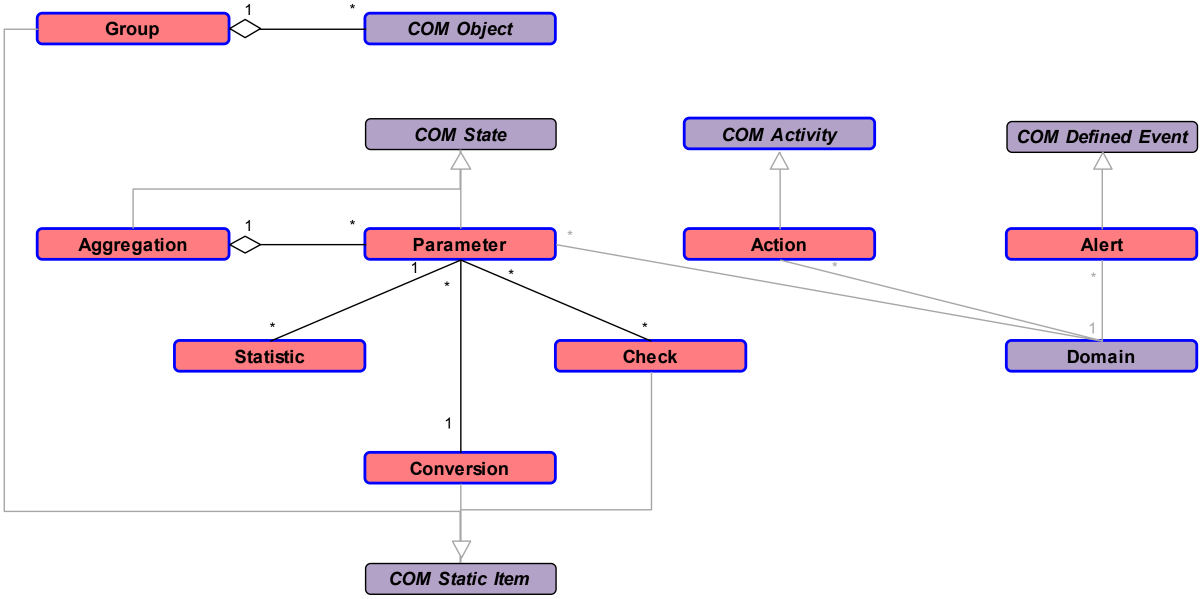


Figure 5-3: Mission Control Data (Monitoring & Control)

The core information objects of Monitoring & Control are parameter, action, and alert. Parameters can also have associated monitoring checks, statistics, and raw-to-engineering-value conversions. Sets of parameters can also be defined as parameter aggregations.

5.2.6.2.2 Action

An action is a single executable task of an M&C service provider, for example, a telecommand, but it can also be a ground-system command or any other discrete operation.

Actions follow the MO COM activity object pattern, comprising separate identity, definition, instance, and event objects. A new instance object is created each time the action is invoked. The state of progress of an action is reflected in a series of event objects associated with each action instance.

Actions are typically defined with an associated set of arguments that can be used to parameterize the associated executable task. In this case, the argument definitions are contained within the action definition, and the argument values within the action instance.

5.2.6.2.3 Alert

An alert corresponds to any operationally significant event that is raised asynchronously by an M&C service provider.

Alerts follow the MO COM-defined event object pattern, comprising identity, definition, and event objects. A new event object is created each time the alert is raised.

Alerts may be defined with an associated set of arguments that can be used to provide more detailed information about the alert event. In this case, the argument definitions are contained within the alert definition, and the argument values within the alert event.

5.2.6.2.4 Parameter

A parameter is a single unit of data reported by an M&C service provider, for example, a telemetry parameter, but it can be any discrete item of monitoring data. Parameters can be of any supported data type.

Parameters follow the MO COM state object pattern, comprising identity, definition, Value instance, and event objects. There is a single value instance object associated with each parameter definition, which holds the current status of the parameter.

Parameters may have both raw and engineering values. Raw values are unconverted, while engineering values have been calibrated to have a meaningful value in defined engineering units. The parameter definition can reference a conditional list of conversions to be applied under different circumstances.

Parameters may also have an associated validity condition that indicates whether the value is meaningful or not (for example, a reported parameter may be invalid if the equipment generating it is not powered).

Parameters may also have associated conversions, checks, and statistics (see below).

Sets of parameters may also be defined as aggregations.

5.2.6.2.5 Aggregation

An aggregation is a collection of parameters provided as a set by a service provider.

Aggregations follow the same MO COM state object pattern as parameters, comprising identity, definition, instance, and event objects.

5.2.6.2.6 Check

A check may be defined and applied to parameters by an M&C service provider that then reports the check results. Check types are extensible, but include:

- limit check: the parameter value lies within a specified range;
- constant check: the parameter value is checked against a specified value or the value of another parameter;
- delta check: the change in value is checked against a pair of thresholds.

The structure of checks follows the MO COM Static item object pattern, comprising identity and definition objects (specific to check type). This is then extended by check link and check link definition objects to associate the check with one or more parameters. Check transition events are generated by the M&C service provider to report changes in check result for a given check-parameter pair.

5.2.6.2.7 Conversion

A conversion may be defined and applied to parameters by an M&C service provider to convert raw parameter values to engineering values.

Conversions follow the MO COM Static item object pattern, comprising identity and definition objects (specific to conversion type). The linkage to parameters is provided within the parameter definition object, the result of conversion (engineering value) stored in the parameter instance object and reported through parameter event objects.

5.2.6.2.8 Statistic

A statistic is a defined statistical evaluation (for example, min, max, mean, standard deviation) associated with parameters that is evaluated and reported by an M&C service provider.

Statistics have a bespoke structure of MO COM objects. Supported statistical evaluations are defined as statistic function objects; the linkage to parameters is provided through statistic link and statistic link definition objects; the result of the statistical evaluation is stored in a statistic value instance object.

5.2.6.2.9 Group

A group is a collection of MO COM objects of the same type. The MO M&C group service provides a mechanism for other services to reference sets of their own objects using a single group reference. Groups follow the MO COM static item object pattern, comprising identity and definition objects.

5.2.6.3 Other Mission Control Data

5.2.6.3.1 Overview

The following diagram shows the top-level information objects associated with the Automation and Onboard Configuration Management functions and how they relate to the Monitoring and Control information objects described above.

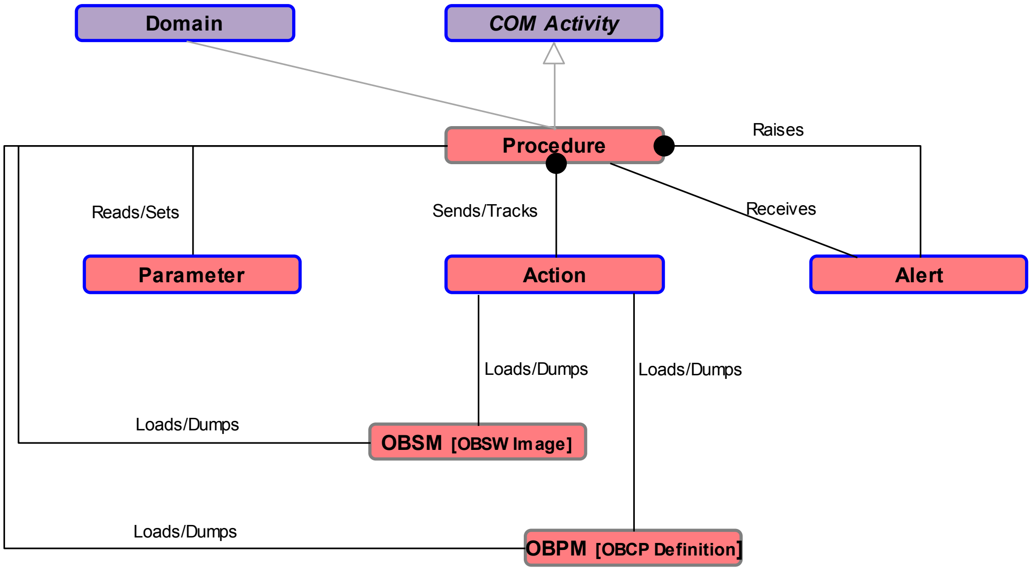


Figure 5-4: Mission Control Data (Other Services)

5.2.6.3.2 Procedure [Prospective]

A procedure (reference [57]) is a single executable task of an automation service provider, which may have an extended duration. A procedure may correspond to a simple predefined sequence of actions, a complex procedure script, or a software function that is executed automatically by the service provider.

Procedures may be defined in terms of other Mission Control information objects:

- referencing and/or setting parameter values;
- initiating (sending) and tracking actions;
- receiving and raising alerts;
- loading and/or dumping OBSW images;
- loading and/or dumping OnBoard Control Procedures (OBCP) definitions.

Procedures, like simple actions, follow the MO COM activity object pattern, comprising separate identity, definition, instance, and event objects. A new instance object is created each time the procedure is invoked. The state of progress of a procedure is reflected in a series of event objects associated with each procedure instance.

Procedures are typically defined with an associated set of arguments that can be used to parameterize the associated executable task. In this case, the argument definitions are contained within the procedure definition, and the argument values within the procedure instance.

5.2.6.3.3 Onboard Software Management: OBSW Image [Not Planned for Standardization]

Onboard Software Management (OBSM) data (reference [58]) is associated with the onboard configuration management function and its management of Onboard Software (OBSW). This includes OBSW Images that may be transferred to or from a spacecraft.

5.2.6.3.4 Onboard Procedure Management: OBCP Definition [Prospective]

OnBoard Procedure Management (OBPM) data (reference [58]) is associated with the onboard configuration management function and its management of OBCP. This includes OBCP definitions that may be transferred to or from a spacecraft.

5.2.7 NAVIGATION AND TIMING DATA

5.2.7.1 Overview

The information objects associated with the Navigation and Timing functional group relate to one or more domains: spacecraft, ground stations, and celestial bodies, as illustrated in the following diagram. They fall into two main subgroups: Navigation Data (NAV) and Timing Data (TIM).

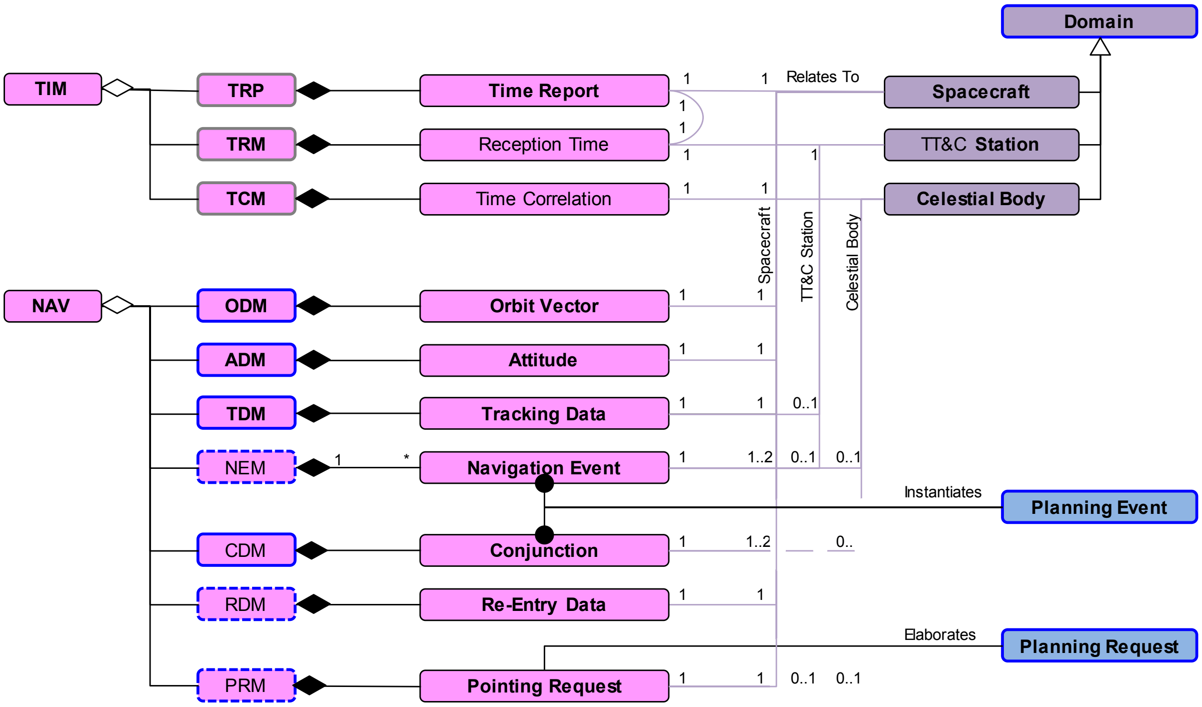


Figure 5-5: Navigation and Timing Data

Navigation events and conjunctions contain the predicted timing of events that are significant to Mission Planning, and may be instantiated as planning events.

Similarly, PRQs associated with the pointing of a spacecraft, instrument, or antenna, may be elaborated by a Pointing Request Message.

5.2.7.2 NAV: Navigation Data

5.2.7.2.1 General

Navigation Data comprises a set of standardized messages relating to the Navigation functions. All have currently been defined as XML schema and are typically exchanged as files.

5.2.7.2.2 ADM: Attitude Data Messages

The Attitude Data Message (ADM) (reference [27]) contain information that defines the attitude state of a spacecraft at one or more times. The ADM support two message formats:

- Attitude Parameter Message (APM);
- Attitude Ephemeris Message (AEM);
- Attitude Comprehensive Message (ACM) [Future]

The APM consists of an instantaneous attitude state and optional additional information, including planned maneuvers, that enable a consumer with attitude modelling capability to propagate the spacecraft attitude over time.

The AEM comprises a history or forecast of the spacecraft's attitude as a series of attitude states at specific points in time. A consumer can use interpolation techniques to determine the attitude states at arbitrary times within the span of the ephemeris.

Augmentation of the ADM is proposed to support the exchange of information on planned rotational maneuvers.

The ACM will provide more comprehensive attitude related information than is contained in the earlier two messages. One planned augmentation of the ADM is proposed via the ACM to support the exchange of information on planned rotational maneuvers.

5.2.7.2.3 CDM: Conjunction Data Message

The Conjunction Data Message (CDM) (reference [28]) contains information that defines the relationship between the orbit states of different space objects at a single point in time: the time of closest approach.

The CDM is the final product of Conjunction Assessment and can be used to provide spacecraft operators with the information they need to assess the risk of collision and plan collision avoidance maneuvers, if necessary. The CDM notifies the spacecraft operator(s) of possible conjunctions with another space object and enables consistent warning by different organizations employing diverse CA techniques. It comprises the identity of the affected objects, miss distance, probability of collision, and the relative position and velocities of the objects at the time of closest approach.

5.2.7.2.4 NEM: Navigation Events Message [Future]

The Navigation Event Message (NEM) (reference [64]) contains the predicted timings of orbital events, such as ground station visibilities, sensor blinding events, and eclipses. The NEM is associated with a single object (usually a spacecraft) and contains all the predicted geometric events occurring within a specified time window. For each event, the following information is provided:

- event type;
- unique event ID;
- predicted time of the event;
- related object (e.g., celestial body or ground station ID) if relevant for the event type;
- duration of event if relevant.

The NEM is an output of Orbit Propagation and is a key input to Mission Planning.

5.2.7.2.5 ODMs: Orbit Data Messages

Orbit Data Messages (ODM) (reference [25]) contain information that defines the orbit state of a spacecraft at one or more times. The ODM supports three message formats:

- Orbit Parameter Message (OPM);
- Orbit Mean-Elements Message (OMM);
- Orbit Ephemeris Message (OEM);
- Orbit Comprehensive Message (OCM) [Future].

The OPM specifies the orbital state (single position and velocity in Cartesian coordinates) or osculating Keplerian elements of a spacecraft at an instant of time, while the OMM specifies the characteristics of the spacecraft orbit expressed in mean Keplerian elements at a specified epoch. Neither the OPM nor OMM is designed for higher fidelity propagation. However, the OPM allows the user to specify simple parameters related to finite and instantaneous maneuvers and provides simple parameters for the modelling of solar radiation pressure and atmospheric drag.

The OEM comprises a history or forecast (prediction) of the spacecraft’s orbit as a series of orbital state vectors at specific points in time and allows for the modelling of any number of gravitational and non-gravitational accelerations. The consumer can use interpolation to obtain the spacecraft position and velocity state at times other than those explicitly contained in the message.

Augmentation of the ODM is proposed to support the exchange of information on planned translational maneuvers.

The OCM will provide more comprehensive orbit related information than is contained in the earlier three messages. One planned augmentation of the ODM is proposed via the OCM to support the exchange of information on planned translational maneuvers.

5.2.7.2.6 PRM: Pointing Request Message

The Pointing Request Message (PRM) (reference [29]) requests that spacecraft, instrument or antenna be pointed at a specified position external to the spacecraft at one or more times, which indirectly implies an attitude state. It provides a common and standardized format for the exchange of pointing requests between the requestor and spacecraft operators.

5.2.7.2.7 RDM: Re-entry Data Message

The Re-entry Data Message (RDM) (reference [65]) contains information about a single re-entry event:

- information about the message itself (creation date, originator, etc.);
- identification of the re-entering object (name, id);
- basic re-entry information (mandatory): remaining orbital lifetime, whether the re-entry is controlled or not, and which celestial body the object is orbiting;
- more complex re-entry information (optional): re-entry and impact windows, impact location and probabilities, state vector, object properties, the OD process, and observations used to predict the re-entry.

The information is used by satellite operators, civil protection, or aviation authorities to assess the re-entry risk and plan any needed mitigation measures.

The RDM is not limited to man-made objects re-entering the Earth’s atmosphere. It could be used for any entry/impact event (e.g., a space probe landing on Venus, or an asteroid impacting Earth).

5.2.7.2.8 TDM: Tracking Data Message

The Tracking Data Message (TDM) (reference [26]) contains information that can be used to determine the orbit state of a spacecraft. It specifies a standard format for a single message type used in the exchange of spacecraft tracking data. Currently, the following tracking data types are supported:

- ground-based radiometric tracking data:
  - uplink and downlink frequencies,
  - range, differenced range, and range rate,

  - delta-DOR,
  - Doppler (1-, 2-, 3-, and 4-way) and differenced Doppler,
  - antenna angles (azimuth and elevation),
  - interferometric types,
  - optical data (planned);
- spacecraft-to-spacecraft Doppler and range;
- ancillary information required to calculate measurement residuals: meteorological data (weather), media delays/correction, and clock bias/drift measurements.

The TDM does not currently support direct measurements of position, such as may be acquired using satellite navigation systems or image processing.

5.2.7.3 TIM: Timing Data [Future]

5.2.7.3.1 General

Three key information exchanges associated with timing have been identified, but CCSDS has not yet defined standards to support these, with the partial exception of Time Reception (see 5.2.7.3.3 below).[3]

5.2.7.3.2 TRP: Time Report [Future]

A Time Report contains a full resolution time generated by a spacecraft onboard clock and transmitted to ground immediately, or with a known onboard processing delay. The report is required, in conjunction with an associated Time Reception Message, to support correlation of the onboard clock to the mission time reference.

Other messages generated by the spacecraft may also contain a timestamp generated from the onboard clock, but these timestamps may not be at full resolution and may have variable onboard processing delays.

There is currently no CCSDS Recommended Standard for the Time Report, aside from the optional use of the packet secondary header in the Space Packet (reference [3]) to transmit a time code.[3]

[3] CCSDS has begun work on standards for time exchange, time correlation, and time synchronization [Future]. These will be integrated into MO Timing services once they become available.

5.2.7.3.3 TRM: Time Reception Message [Partial]

A Time Reception Message is associated with a Time Report and provides an accurate timestamp (in terms of the mission time reference) of the ground reception time of that Time Report message.

While there is no current CCSDS Recommended Standard for a dedicated Time Reception Message,[3] the functionality is supported through the CCSDS CSS Space Link Extension Transfer Services, where Earth Receive Time is provided as an annotation parameter to the transfer data. Assuming the Time Report itself is carried in a discrete frame or packet, the Earth Receive Time of the container frame or packet can be associated with it.

5.2.7.3.4 TCM: Time Correlation Message [Future]

The onboard clock correlation function takes Time Reports and associated Time Reception Messages, together with any required ancillary information, such as:

- one-way light time between spacecraft and ground reception station, which can be derived from ranging data or from the spacecraft's orbit vector and the known location of the ground station;
- statically defined processing delays: onboard the spacecraft in generating and transmitting the Time Report, and within the ground station in terms of generating the Time Reception Message.

to derive onboard clock coefficients that specify the relationship between onboard clock and the mission reference time at a given point in time. This enables conversions between onboard time and reference time to be performed.

The Time Correlation Message (TCM) contains one or a series of onboard clock correlation coefficients, together with their reference times.

There is currently no CCSDS Recommended Standard for the TCM.[3]

5.2.8 MISSION PLANNING DATA [FUTURE]

5.2.8.1 Overview

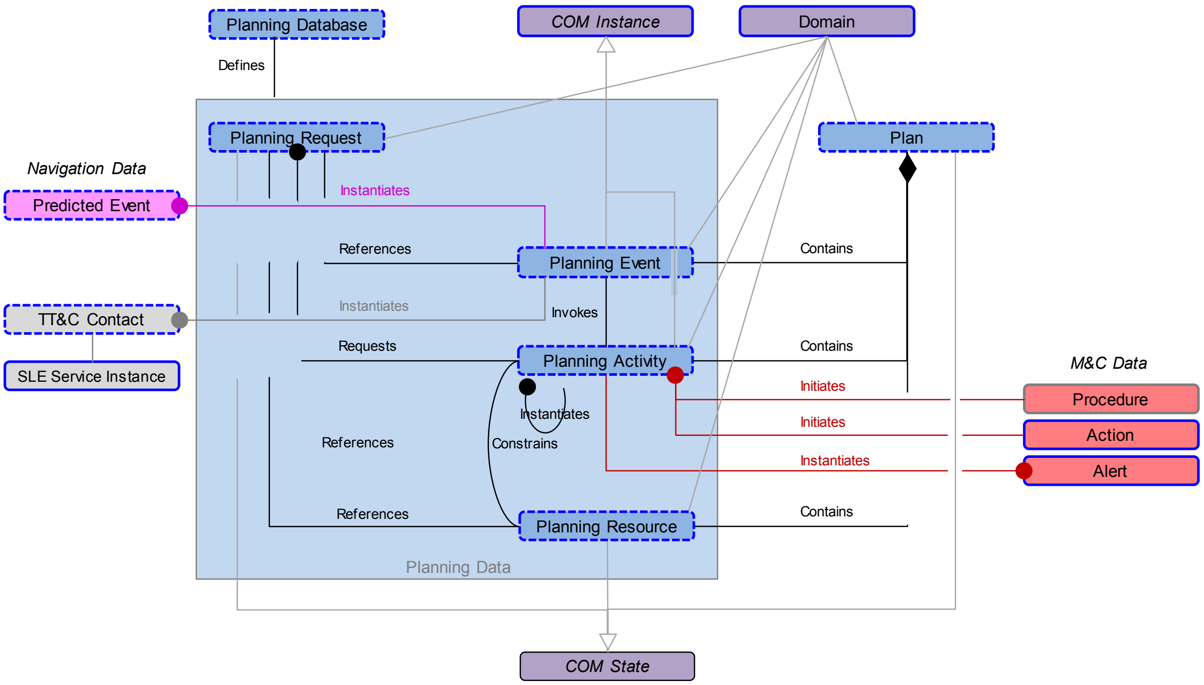


Figure 5-6: Mission Planning Data

The top-level information objects associated with Mission Planning and Scheduling functional group and how they relate to those of other functional areas are shown in figure 5-6, above. These are defined in (reference [63]).

Planning requests and plans are essentially container structures used to represent the input and output, respectively, of the planning process.

Planning activities, planning events, and planning resources are compound objects with associated definitions and instances that represent the items within a plan or planning request.

Planning database comprises all planning configuration data, including the definitions of planning activities, planning events, and planning resources, as well as templates for planning requests.

Planning constraints are rules or conditions applicable to the planning process that can be defined in advance and contained within the definitions of planning activities. They can also be defined at run-time in the context of a planning request or plan.

5.2.8.2 Plan

A Plan (PLN) is the output of the planning process. It contains a set of selected planning activities associated with time, position, or other planning event. A plan may contain additional related information, including:

- planning events;
- planning resource vectors.

A plan also contains information relating to the plan itself, including:

- source and generation time;
- time period and domains to which it relates;
- predecessor plan, if any; relevant if plans are iterative.

In the context of the Mission Planning and Scheduling standardization activity, there is no distinction between the terms ‘plan’ and ‘schedule’, and only the term plan is used.

5.2.8.3 Planning Activity

A planning activity is a meaningful unit of what can be planned: the building blocks from which plans are constructed. They may be hierarchical: planning activities may be composed of other planning activities, but the leaf nodes of the hierarchy must be executable (either automatically or manually) and will typically correlate to Monitoring & Control actions or procedures that are to be initiated when the plan is executed.

Planning activities are instantiated within a plan in response to:

- an explicit planning request;
- the inclusion of an associated planning event in the plan;
- the occurrence of an associated M&C alert.

Planning activities follow the MO COM activity object pattern, comprising separate identity, definition, instance, and event objects. A new instance object is created each time the activity is invoked. The state of progress of an activity is reflected in a series of event objects associated with each activity instance.

Planning activities are typically defined with an associated set of arguments that can be used to parameterize the associated executable task. In this case, the argument definitions are contained within the activity definition, and the argument values within the activity instance.

5.2.8.4 Planning Constraint

A planning constraint is something that limits or restricts the scheduling of planning activities. A planning constraint could be based on planning resources, with their specific allocation and consumption, but other types of constraint exist, including: time constraints, sequencing constraints, position or other geometric constraints, and exclusion constraints that restrict which activities can be executed in parallel.

Planning constraints can be defined or occur in various parts of the Mission Planning information model:

- within a planning request;
- within a plan if it was defined in the context of a planning request;
- within the definition of a planning activity.

5.2.8.5 Planning Event

A planning event marks when or where something of significance to planning is predicted to occur (or exceptionally during plan execution, has occurred). The execution of planning activities may be linked to planning events.

Predicted events may be internally generated by the planning function, but typically originate from an external function, such as a Navigation function or the scheduling of TT&C contacts, and a planning event is created within the plan to reference the external event.

Planning events follow the MO COM activity object pattern, comprising separate identity, definition, instance, and event objects. A new instance object is created each time the planning event is invoked within a plan. The state of progress of a planning event is reflected in a series of event objects associated with each planning event instance.

Planning events are typically defined with an associated set of arguments that can be used to parameterize the associated planning activity to be executed. In this case, the argument definitions are contained within the planning event definition, and the argument values within the planning event instance.

5.2.8.6 Planning Resource

A planning resource is an abstract status modelling the state of the system being planned. It may be necessary to model some aspects of system state in order to:

- trigger the execution of a planning activity;
- constrain the execution of a planning activity.

While the modelling of planning resources is internal to the planning system, planning resources may be referenced in constraints associated with a planning request. It may also be necessary in a distributed planning system to coordinate the value of planning resources between planning functions, in which case the resource values may be contained within a plan.

Planning resources follow the MO COM state object pattern, comprising identity, definition, Value instance, and event objects. There is a single value instance object associated with each resource definition, which holds the current status of the resource.

5.2.8.7 Planning Request

Planning Requests (PRQ) are the main input to the planning function and are containers for the information needed to be exchanged between the requester and the planner. It is envisaged that this will support the specification of different types of request:

– request to plan a planning activity or a set of activities;

– request to achieve a goal;

– request to use a plan as an input to the planning process;

– request to modify the content of a plan.

The main characteristic of the planning request is that, being a container, it needs to hold references to, or instances of, the constituent information items that are required by the planner and agreed upon by the interacting parties for exchange at interface level. It has one or more planning activities as the basis of the request, optionally referencing planning events.

Information about constraints on when a requested activity can or shall be planned may also be exchanged as part of the planning request, by referencing constraints on the time, on the position, on the state of planning resources, or other planning activities.

5.2.8.8 Plan Execution Control Data

Plan Execution Control Data (see figure 5-1) is specific data required to monitor and control the Plan Execution function. Some aspects of this may be realized as a specialization of the MO Monitor & Control service and its associated information objects. However, there are special operations on plans and their execution which require dedicated data.

5.2.9 MISSION DATA PRODUCTS [FUTURE]

Mission Data Products (MDP) (reference [56]) are mission data sets of potentially large size and varied internal structure that are transferred between distributed elements of a space system.

The MDP concept abstracts the structure, content, and format in which diverse space mission data products can be persisted, requested, and provisioned. This allows the specification of a generic set of services for managing, requesting, and provisioning space mission data products, without making assumptions about the implementations of the underlying mission data product distribution systems.

An MDP may contain bulk data relating to Mission Operations stored onboard a spacecraft or historical data retrieved from a Data Archive. Examples of this include parameter value evolution in a given time period and actions history.

An MDP may also contain science or other mission data acquired onboard a spacecraft or generated by an external Mission Data Processing function that is stored onboard a spacecraft or in a Data Archive. Although the content of this data is not itself meaningful to Mission Operations functions, they are responsible for its transfer and storage.

The MDP information comprises:

– Mission Data Product Catalogue;

– Mission Data Product:

  - Product Type and associated properties that describe the product,
  - Product Source (the stored data product),
  - Product Specification (optional; specifies the internal format of the product).

A limited set of standard product specifications are provided for product types that are typically involved in interoperable mission-operation scenarios. For other product types, it is possible to add customized product specifications.

5.2.10 OPERATIONS PREPARATION DATA [PARTIAL]

5.2.10.1 General

The information objects associated with the Operations Preparation functional area are shown in the top-level diagram of MO Information Groups (see figure 5-1) as they constitute the configuration data for other MO functions, primarily Mission Control and Mission Planning.

In general, these information objects have not yet been standardized by CCSDS, although they are typically exchanged between spacecraft and instrument manufacturers and spacecraft operators. An exception is the XTCE Recommended Standard (reference [23]).

They may, however, be transferred between functions as opaque data structures using standard services, including:

- MO Common: Configuration Service (any operations preparation data);
- MO OPM Service (APD only);
- MO OSM Service (OBSW only).

5.2.10.2 SDB: Spacecraft Database [Partial]

The SDB is the configuration data required by the Mission Control: Monitoring & Control function (see 5.2.6.2) and includes:

- parameter definitions together with associated aggregation, conversion, check, and statistics definitions;
- action definitions;
- alert definitions.

It may also include information on how the above are encoded within CCSDS Space Data Packets or Frames, although this is not required if the standard MO M&C services are used.

The current XTCE Recommended Standard (reference [23]) provides a partial solution for the SDB, supporting the definition of parameters and commands (actions) and their encoding in CCSDS Space Data Packets.

The SEDS Recommended Standard (reference [10]) may in the future also be applicable for the exchange of the SDB.

5.2.10.3 APD: Automated Procedure Definition [Prospective]

Automated Procedure Definitions (APD) are required to configure the Mission Control: Automation function. In the event that Automation is onboard a spacecraft, this is also associated with the Mission Control: Onboard Configuration Management OBPM function for the definition of OBCP.

APDs are not currently subject to standardization by CCSDS but may be transported by the OBPM service and referenced or contained in a PRQ and associated planning activities.

5.2.10.4 OBSW: Onboard Software Image [Not Planned for Standardization]

Onboard Software (OBSW) is typically exchanged as onboard software images both between spacecraft or instrument manufacturers and the spacecraft operator, and between the spacecraft operator and the spacecraft. The latter is supported by the Mission Control: Onboard Configuration Management function (OBSM) function.

OBSW is not currently subject to standardization by CCSDS, but may be transported by the OBSM service and referenced or contained in a planning request and associated planning activities.

5.2.10.5 PDB: Mission Planning Database [Prospective]

The Mission Planning Database (PDB) is the configuration data required by the Mission Planning functional area and includes:

- planning activity definitions;
- planning event definitions;
- planning resource definitions;
- planning constraint definitions.

The PDB is not currently subject to standardization by CCSDS. However, the information model defined for Mission Planning and Scheduling Services (see 5.2.8) (reference [63]) identifies the key elements required within it.

5.2.11 DATA STORAGE AND ARCHIVING DATA

5.2.11.1 General

The Data Storage and Archiving functional group comprises three distinct functions with associated information objects:

- Onboard File Store [Future];
- Operations Archive;
- Data Archive.

The Onboard File Store is strictly an SO function, but is managed through interaction with MO functions.

The CCSDS Data Archive Information (DAI) standards do not define interoperable interfaces, only procedures and information content, and are not considered applicable to mission operations archives.

5.2.11.2 Onboard File Store [Future]

The Onboard File Store is inherently an SO function, but it subject to management by MO functions. File Transfer and Management (FTM) (reference [61]) concerns the management of remote (onboard) file stores. While grouped within Data Storage and Archiving, the corresponding services are typically associated with Mission Control, while the files themselves can contain data relating to any of the information groups (Mission Control, Mission Planning, Navigation, Operations Preparation, and Mission Data Products). In addition to the file content, the FTM data includes the drive and directory structure of the file store and the messages associated with File Transfer and File Management operations, including:

- file upload/download;
- file create/rename/delete/move/copy;
- directory list/create/rename/delete/move.

5.2.11.3 Operations Archive

An Operations Archive is typically integrated within a Mission Control function to support rapid retrieval, display, and analysis of mission operations data. This is distinct from the long-term Data Archive discussed in 4.2.9.4 above. An Operations Archive typically contains a record of the evolving status of many mission operations information objects (for example, parameters, actions, alerts, automated procedures, planning requests, planning activities, planning events, planning resources, and navigation data messages). This is typically stored as a time series, indexed to allow cross-referencing between different data items, and is organized to support rapid retrieval in operational use.

Common Archive (CAR) data is a generic term for any information object defined in terms of the MO COM (see 5.2.4) (reference [16]) that can make use of the generic COM Archive service.

Using the COM Archive approach, each change to a COM object that represents the information object (identity, definition, instance, or update, depending on the applicable COM object pattern) is stored as a record in the archive.

The same mission operations information may also be stored in the long-term data archive, but in this context, it will normally be stored as a large aggregate data package covering a period of time (e.g., a day’s telemetry or command history), together with the associated DAI metadata describing its structure and preservation data.

5.2.11.4 DAI: Data Archive Information

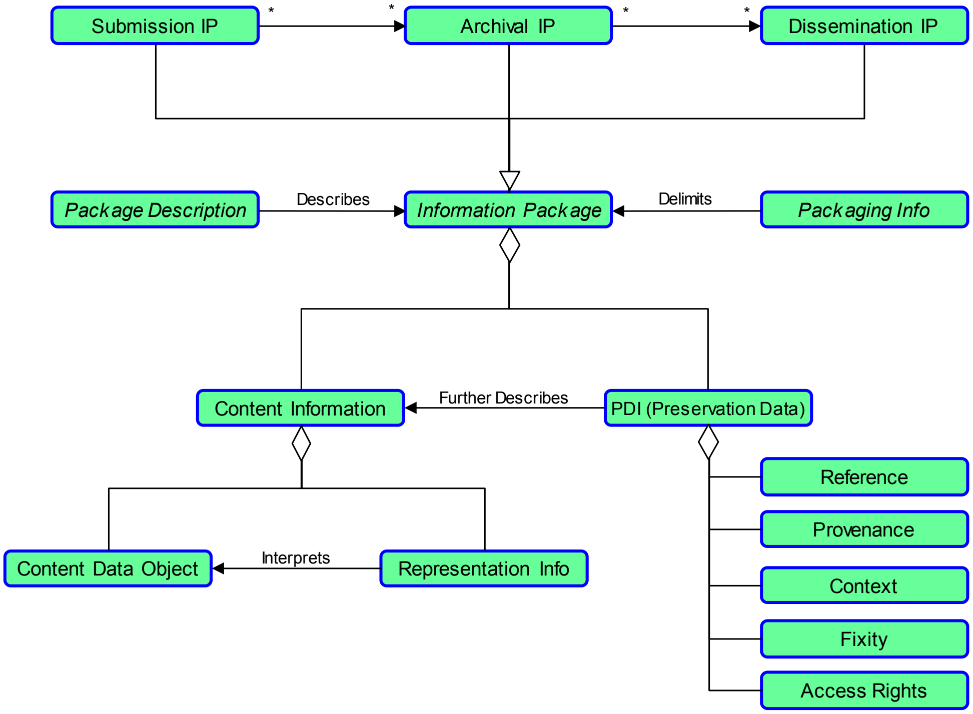


Figure 5-7: Data Archive Information

Abstract information objects associated with the long-term archiving and retrieval of mission data have been defined in the context of the CCSDS Open Archival Information System (OAIS) standard (reference [30]).

This standard defines the concept of an abstract Information Package (IP), which has three derived information classes:

- Submission Information Package (SIP): generated by an archive producer;
- Archival Information Package (AIP): stored in an archive;
- Dissemination Information Package (DIP): disseminated to an archive consumer.

There is not a one-to-one relationship between these classes. Multiple SIPs may be constituted into a single AIP during archive ingestion, and AIPs may be decomposed into multiple DIPs during dissemination.

Each IP has three principal elements:

- Package Description: the definition of the structure of an IP;
- Information Package: the IP itself;

- Packaging Information: information that actually or logically binds or relates the components of the package into an identifiable entity on specific media.

Package Descriptions may be expressed using the following CCSDS Recommended Standard data description languages or other widely supported methods:

- Parameter Value Language (PVL) Specification (reference [32]);
- Data Description Language EAST Specification (reference [33]);
- Data Entity Dictionary Specification Language (DEDSL) (Abstract, reference [34]; PVL, reference [35]); XML/DTD, reference [36]);

Information Packages comprise:

- **Content Information**: comprising Content Objects together with Representation Information that supports their interpretation;
- **Preservation Description Information**: information that supports trust in, access to, and context of the Content Information over a long and indefinite period of time; this includes:
  - **Reference**: unambiguously identifies the content,
  - **Provenance**: documents the source and history of any changes to the content,
  - **Context**: describes the relationship of the content to its environment,
  - **Fixity**: checks data integrity or validation/verification keys to ensure content has not been altered in an undocumented manner,
  - **Access Rights**: restricts access to the content, including legal framework, licensing terms, and access control.

5.3 SO INFORMATION VIEWS

5.3.1 OVERVIEW

This subsection addresses data or information objects within the scope of the SO functions. This Information model is only introduced at a relatively high-level, sufficient to identify the information exchanged between functions and any relationships between information objects exchanged across multiple interfaces. For a full and detailed specification of the referenced information objects, the reader is directed to the relevant CCSDS Recommended Standards.

The remainder of the subsection is structured as follows:

a) **SO Information Model**: top-level decomposition of SO Information;

b) **SO Electronic Data Sheet Model**: summary of the generic information model for EDS;

c) **SO Dictionary of Terms**: summary of the mechanism for consistent interpretation of terms in the EDS and SO information model;

d) **SO Functional Interface Data**: summary of information objects that pass through SO service interfaces;

e) **SO Management Information**: summary of management information that might be defined in the future.

Each subsection comprises an Information Viewpoint diagram and a description of each of the information objects it contains.

5.3.2 SO INFORMATION MODEL

The SO information model consists of the following parts:

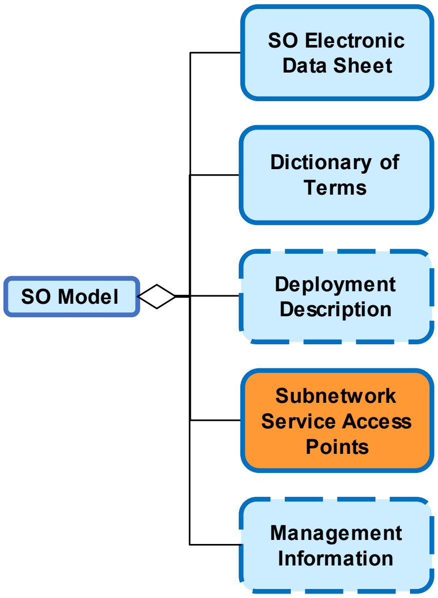


Figure 5-8: SO Information Model

The SEDS provides the means to describe:

– spacecraft components;

– interfaces (physical, electrical, service);

– behavior;

– configurations and deployments of components [Future].

The SOIS Electronic Data Sheets use a set of schemas that control the SEDS models. These may be extended using a defined procedure. When component descriptions complying with SEDS are constructed, they may then be used as controlled artefacts that guide a chain of software tools at each agency to compose a vehicle. The tool chain is a set of software tools that assist in designing the composition of a spacecraft.

The SO DoT is an ontological model of terms used in the SO information model to assure consistent interpretation by algorithms. The DoT is designed to be extensible. It defines a core set of terms that may then be extended as needed to describe new components, interfaces, and features.

The SEDS may be used to create a Deployment Description [Future] that describes the configuration of devices on the subnetwork(s) in a vehicle.

The subnetwork concepts describe the elements of data that flow between functions in the SO subnetwork service access point.

The management information contains data about the configuration of subnetworks. This information may be implicit within the compiled software of a vehicle when it is static throughout the vehicle's mission, or it may be present and accessible where it can be updated during the vehicle's mission.

5.3.3 SO ELECTRONIC DATA SHEET MODEL

5.3.3.1 General

In 10.3, figure 10-5 identifies the roles of SO EDSs and the information model in the overall SO context. This subsection explains the details of the SO information model itself.

The base structure of an instance of SEDS appears in figure 5-9. The trunk of the tree structure is a Datasheet element or a PackageFile element. The primary purpose of the Datasheet element is to hold in its branches the description of a device. A Datasheet is a 'fully resolved' EDS document that contains the full description of all entities, types, and interfaces referenced within that document (either directly or via 'include' directives). The purpose of a PackageFile is to describe a software service onboard the vehicle, or to provide metadata or to define shared types. A PackageFile is a modular EDS component, and typically not directly usable on its own. A Datasheet may be a collection of PackageFiles combined with the logic and information of how they work together.. Either kind of SO EDS can contain Packages or Metadata. A PackageFile can consist entirely of metadata that is shared by other SEDS instances; for example, parameters of a computing platform, such as word size, can appear in a PackageFile and be referenced by other PackageFiles that describe services.

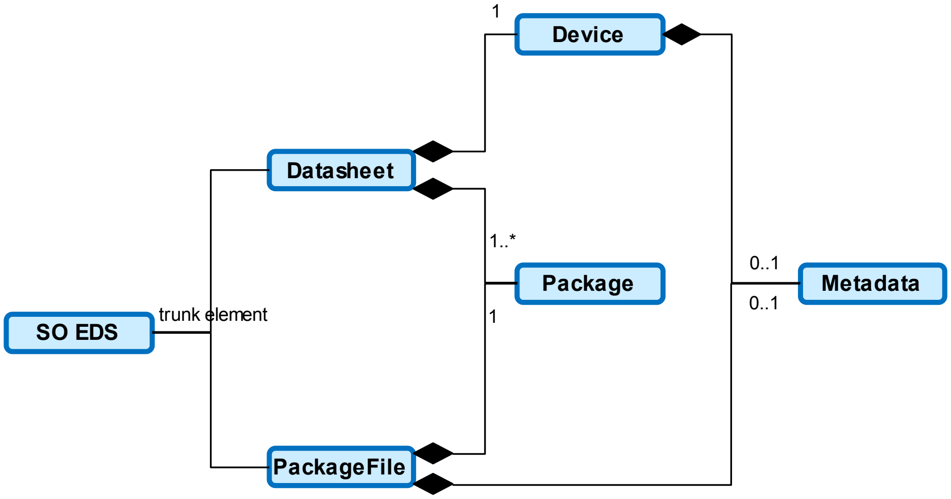


Figure 5-9: Base Structure of a SO EDS

There are two kinds of elements that form a Datasheet element. One element, a Package element, may appear in any quantity, including zero. The purpose of a Package element is to describe types of data, data interfaces, and behaviors that are peculiar to the composable part that is the subject of the data sheet. The elements in a Package that describe types of data make a model of the syntactic structure of a data type, which may be elaborated with semantic tags defined in the SO DoT.

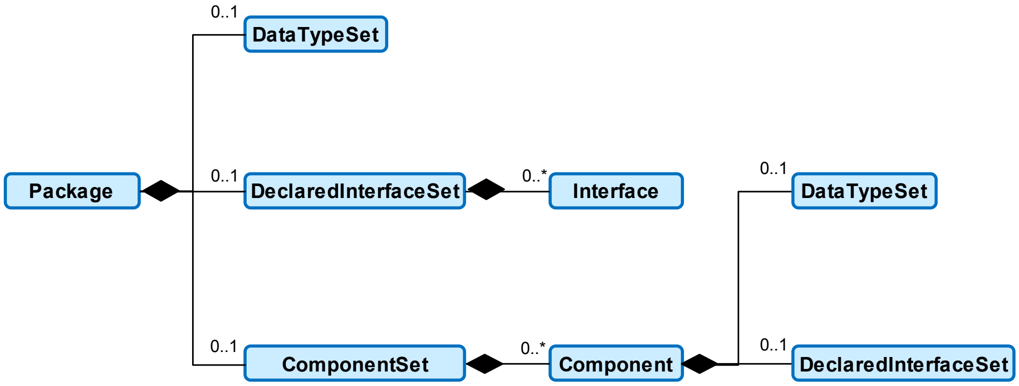


Figure 5-10: Structure of Package Elements in SO EDS

The other kind of element in a Datasheet, a device element, must be present as a single element. The purpose of a device element is to provide metadata about the composable part described by the Package element. In the future, a device element can be expanded to include specification of physical features of a device, including the following:

– Ports [Future]: Devices may have more than one network adapter. A concentrator will have multiple network adapters. Some devices, such as clocks, may have synchronization ports for periodic pulses, which may be accompanied by time-at-tone messages.

– Mounting [Future]: The mounting surface of a device includes a transformation of coordinates between vehicle structure (in Deployment Description) and device coordinates. This transformation is part of a series of rotations that convert between device coordinates and vehicle coordinates, used in attitude control.

– Mass Properties [Future]: The mass properties of a device are needed for computing torques in attitude control.

The content of a device element provides essential information for a vehicle database in mission control and for onboard control systems that maintain homeostasis in flight.

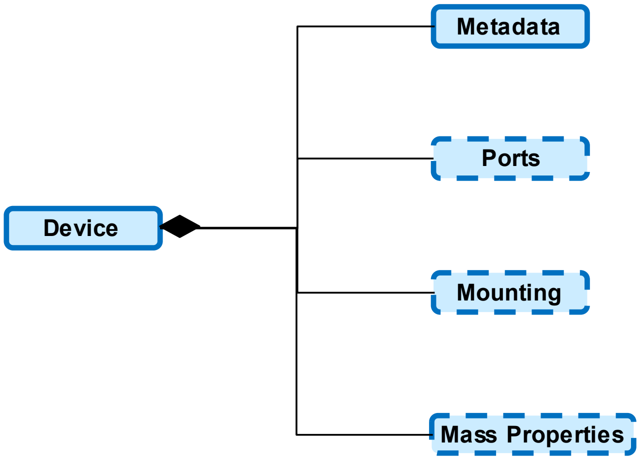


Figure 5-11: Structure of Device Element in SO EDS

The Metadata element includes configuration management information, such as manufacturer's model and serial number, and a model of operation. The elements in a Metadata element may be decorated with semantic tags defined in the DoT, which serve to clarify the intent of categories and values.

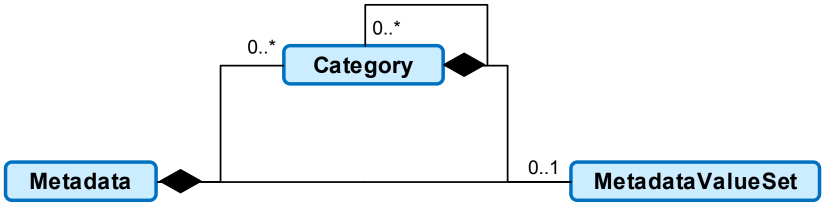


Figure 5-12: Structure of a SO EDS Metadata Element

A PackageFile element may contain a Package element that describes a composable software part. Additionally, a PackageFile element may contain a Metadata element that includes configuration management data about the composable software part. A PackageFile element may contain only a Metadata element that contains mission or platform metadata values.

The EDS and DoT are the parts of the System Model in figure 10-5 that have been defined formally. The third part of the System Model is a Deployment Description [Future]. All three parts appear in figure 5-13, figure 5-14, and figure 5-15 to describe how to compose a vehicle from its parts.

5.3.3.2 Application Service Interfaces

The SO application services appear in figure 5-13, near the left side, in the context of its related concepts. The onboard platform architecture has a collection of application services and a collection of subnet services. The Application Services has Mission Applications, Application Support Services, and Device Services. A SEDS Datasheet may describe a device, and a SEDS PackageFile may describe a software object. For a device, the SEDS Datasheet may also specify the behavior of the Device Service that stands as a proxy for the device in an onboard computer.

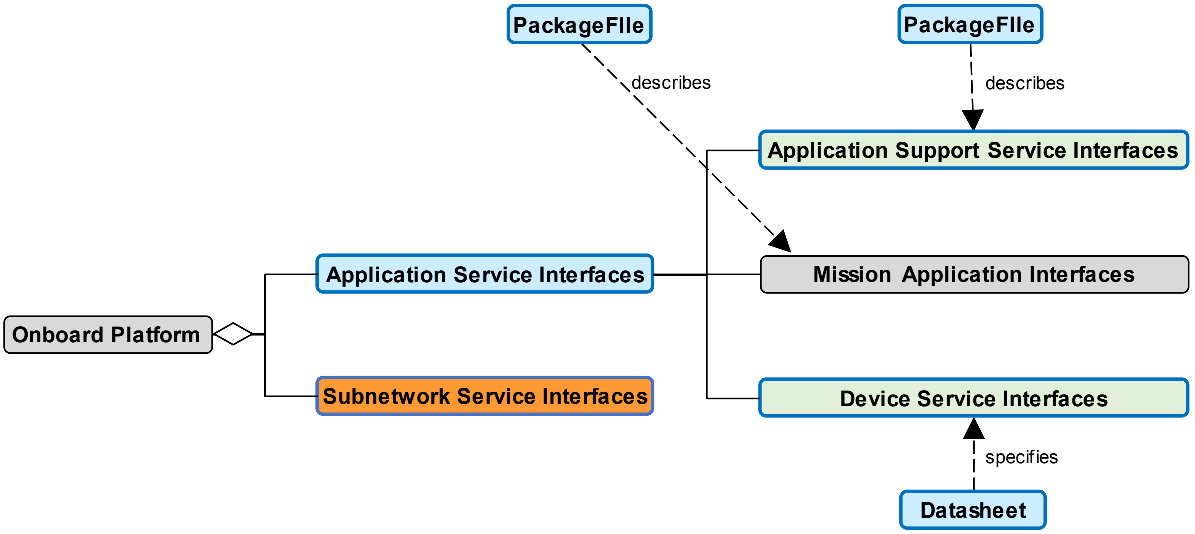

PackageFIle
PackageFIle
describes
describes
Application Support Service Interfaces
Application Service Interfaces
Mission Application Interfaces
Onboard Platform
Subnetwork Service Interfaces
Device Service Interfaces
specifies
Datasheet

Figure 5-13: SO Application Support Concepts

5.3.3.3 Subnetwork Service Interfaces

The subnetwork services collection appears in figure 5-14 near the left side, in the context of its relationships with other concepts. The Subnet Services have Communications Services, Convergence Functions, External Protocols, and Management Services. A Deployment Description [Future] (see 5.3.5) describes the Subnet Services collection, including features such as topology and schedule. Describing those features requires reference to the device Datasheets that describe devices attached to the subnetwork. The Deployment Description also specifies the behavior and interfaces of management services for the subnetwork.

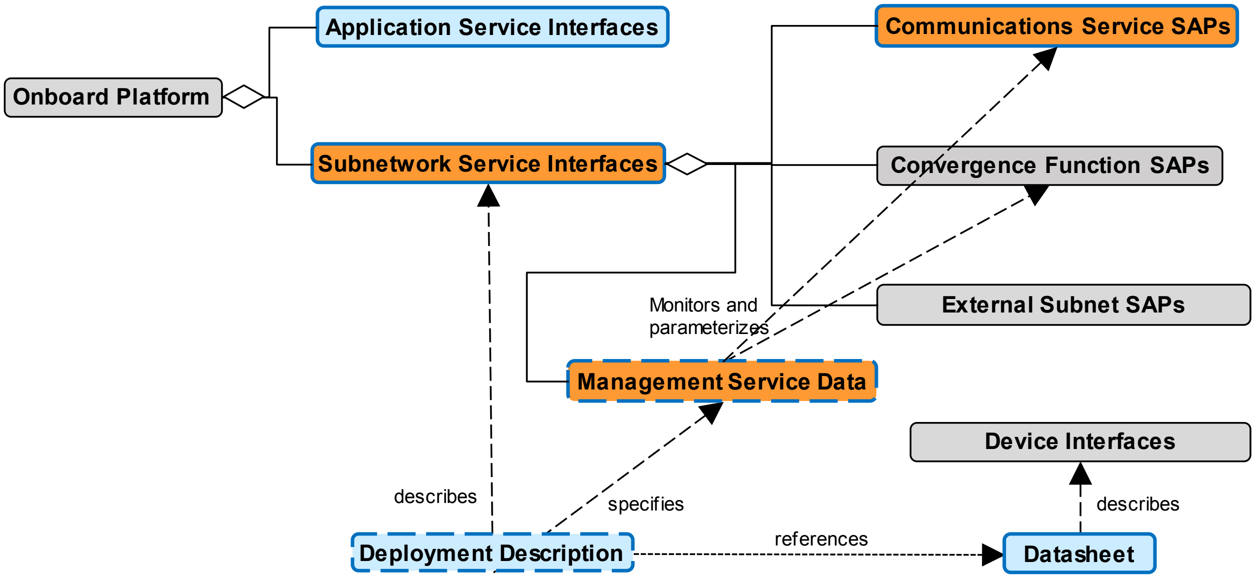


Figure 5-14: SO Subnet Layer Concepts

5.3.3.4 SO System Model

The SO System Model appears in figure 5-15 at the left side, in the context of its relations to other concepts. The System Model has PackageFiles, Datasheets, Deployment Description, and a Dictionary of Terms. The PackageFiles, Datasheets, and Deployment Description all reference the Dictionary of Terms. The Deployment Description [Future] (see 5.3.5) refers to Datasheets that describe devices and that specify Device Services. The Deployment Description specifies Management Services and describes the Subnetwork. Some of the PackageFiles describe externally provided Spacecraft Application software objects. Some of the PackageFiles will describe Application Support Services identified in the SOIS Green Book (reference [51]); examples of package files appear in the SOIS EDS and DoT Green Book (reference [52]).

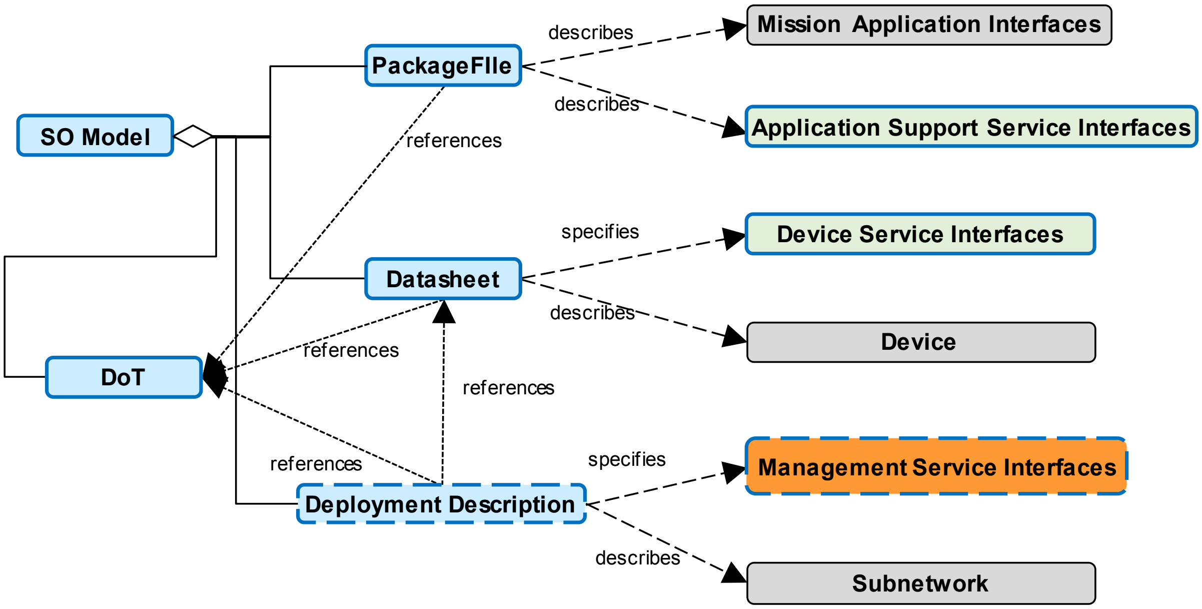


Figure 5-15: SO System Model Concepts

5.3.4 DICTIONARY OF TERMS

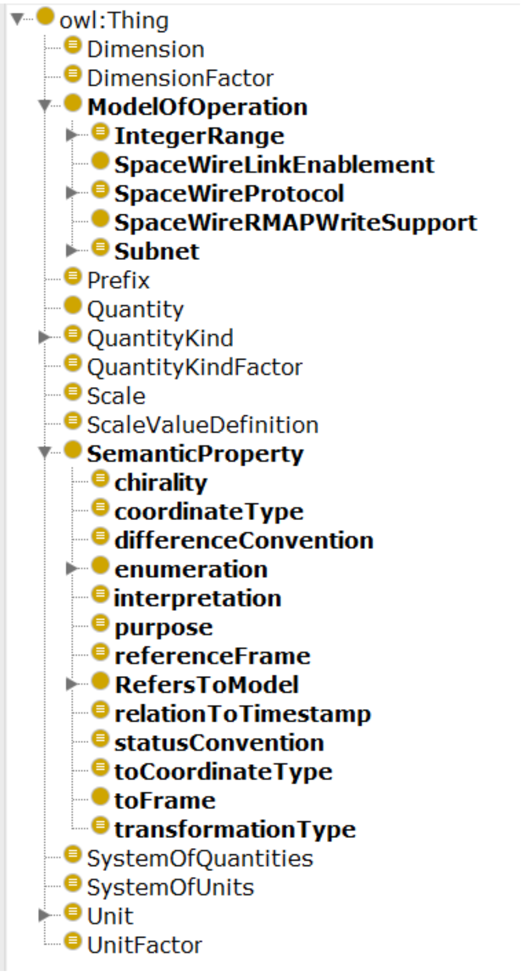
The DoT provides an extensible set of terms that may be used in the SO System Model for interpretation by algorithms in tool chains that assemble the flight software.

The development format for DoT is a Protégé ontology, and its operational form is an XML schema derived from that ontology. Either form may be used in a tool chain, but it is expected that the XML schema will be preferred. The terminology is extensible by adding new individuals to classes, and by defining new classes. Formal relationships among classes can help to clarify the intended meanings of classes in the ontology, and for human users, the meanings are expressed by natural language descriptions.

SEDS instances (that is, Datasheets and PackageFiles) use the terms in the DoT by means of attributes in the XML schema. Each attribute represents a class in the ontology. The possible values of attributes are either individuals in the ontology, or they are names of elements in SEDS instances.

The XML schema generated from the DoT extends the base schema for SEDS. For generation of device services from a Datasheet, the base schema for SEDS may be sufficient. The extensions provided by the DoT allow for sophisticated, automated checking of compatibility between provided and required interfaces. By using the DoT extensions throughout the description of the chain of commands and telemetry, costly mistakes can be avoided, such as commanding thrusters with the wrong units of measure. The attributes provided by the DoT also can guide generation of ‘shims’ between required and provided interfaces, such as selecting relevant elements of a container. For example, software that controls rotational rates could be shimmed to use telemetry packets that carry rotational rates among other elements for attitude, position, and velocity. Composable software components become even more interoperable with this capability to generate adaptations to the semantics of the data that is available onboard, allowing human programmers to concentrate on the difficult coding issues instead of data conversions.

Human readable descriptions of the terms can provide connections to the actual conversational vocabulary used by engineers in various organizations, and these descriptions can be formatted into glossaries for use in those organizations. The formal terms in the DoT are used in EDS and in deployment descriptions because those are intended to be interpreted by tool-chain algorithms that are not equipped to handle natural language.



- owl:Thing
  - Dimension
  - DimensionFactor
  - **ModelOfOperation**
    - **IntegerRange**
    - **SpaceWireLinkEnablement**
    - **SpaceWireProtocol**
    - **SpaceWireRMAPWriteSupport**
    - **Subnet**
  - Prefix
  - Quantity
  - QuantityKind
  - QuantityKindFactor
  - Scale
  - ScaleValueDefinition
  - **SemanticProperty**
    - **chirality**
    - **coordinateType**
    - **differenceConvention**
    - **enumeration**
    - **interpretation**
    - **purpose**
    - **referenceFrame**
    - **RefersToModel**
    - **relationToTimestamp**
    - **statusConvention**
    - **toCoordinateType**
    - **toFrame**
    - **transformationType**
  - SystemOfQuantities
  - SystemOfUnits
  - Unit
  - UnitFactor

Figure 5-16: High-Level Summary of Dictionary of Terms

A summary of the initial content of the DoT appears in figure 5-16. Concepts like ‘QuantityKind’ and ‘Unit’ come from the International Bureau of Weights and Measures VIM publication (reference [42]). The ‘SemanticProperty’ elements were derived from terms developed in a self-organizing spacecraft experiment in the United States Air Force Research Laboratory (reference [40]). The ModelOfOperation includes terms for describing subnetwork properties, as well as terms for describing major features of the operation of a device.

The DoT may be extended to include terms for ad-hoc use within projects, for which the delay for integration into a standard cannot be tolerated. The extension terms cannot be relied upon for general interoperability but may be used within a mission or an identified community. In addition, communication of these terms to CCSDS will enable their integration into the standard DoT, possibly with changes to agree with similar terms. The distribution of the SO EDS schema in SANA (https://sanaregistry.org/r/sois) includes an empty extension schema which may be modified for use within a project (references [10] and [11]).

5.3.5 DEPLOYMENT DESCRIPTION [FUTURE]

The SO EDS concept is extended to include a description of spacecraft subnetwork topology and any related configuration descriptions that are needed to interpret data in interfaces described by SO EDS. This extension is called a deployment description. It is an abstract description of a spacecraft, which may refer to detailed external documents such as manufacturer-specific devices or subnetwork schedules.

The content of a deployment description is expected to include the information elements depicted in figure 5-17.

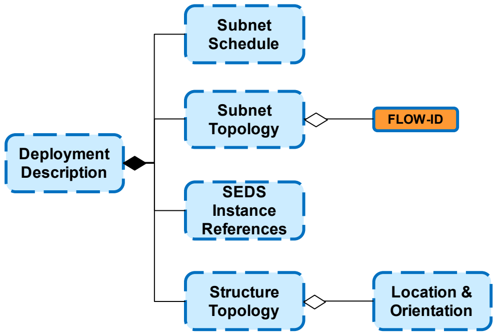


Figure 5-17: Deployment Description

A deployment description is expected to include the following information:

- schedule of subnetwork(s);
- topology of subnetwork(s) as built;
- topology of structure as built;
- reference to SEDS instances in topologies;
- FLOW-IDs in subnetwork topology of instruments represented by SEDS instances, to permit addressing, routing, and QoS for those devices;
- locations and orientations in structural topology of instruments represented by SEDS instances, to permit transformation between device and vehicle reference frames.

The structural topology is expected to contain locations and orientations of mounting surfaces in vehicle coordinates, and the SEDS instances for devices will contain location and orientation of the device mounting surface in device coordinates. By combining the rotations and translations, data can be converted between device coordinates and vehicle coordinates.

The FLOW-IDs resemble the flow labels of Internet Protocol V6 (reference [68]) in purpose, but the two concepts differ in the timing of their usage. The flow labels travel with messages in the headers, but the FLOW-IDs do not travel with messages. The reason for this difference is that Internet Protocol has a Network Layer that must resolve flows as messages move from processor to processor, while FLOW-IDs are resolved at the time of designing a subnetwork. A Network Layer is optional in subnetwork design for a spacecraft for which resources are limited.

5.3.6 DATA EXCHANGED ACROSS SO SUBNETWORK FUNCTIONAL INTERFACES

5.3.6.1 General

The categories of data exchanged across the SO Subnetwork functional interfaces appear in figure 5-18.

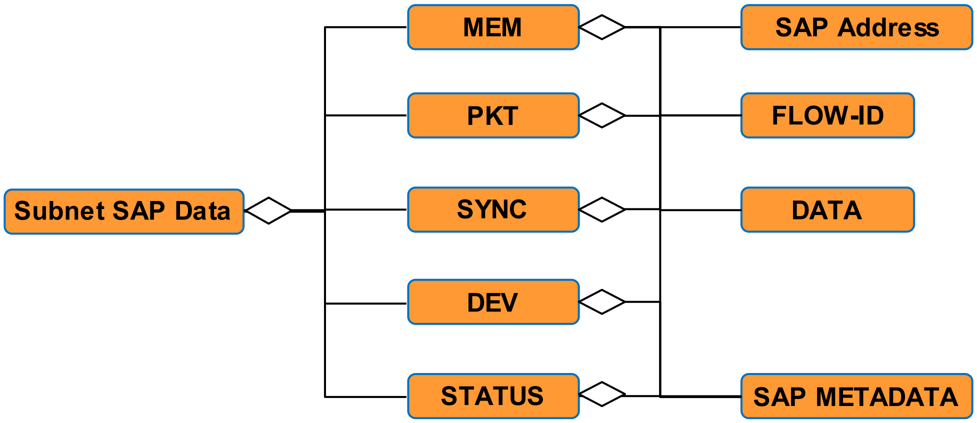


Figure 5-18: Data at Subnetwork Service Access Point Interfaces

Each subnetwork interface has a SAP. The middle column in figure 5-18 corresponds to the interface data items in figure 4-10. The right column in figure 5-18 identifies major categories of data that may appear in the interface data items.

A 'SAP address' is a term used in the early editions of SO subnetwork books, which was intended to mean 'subnetwork identifier and address within subnetwork'. A similar term in the ISO OSI Basic Reference Manual is (N)-connection-endpoint. The information elements named by the 'SAP address' term are being replaced by information elements named by the 'FLOW-ID' term as the subnetwork magenta books undergo their five-year reviews. The FLOW-ID is not placed into messages; it crosses the interfaces of SO subnetwork services in the form of an index, which the services can use to identify management information that includes SAP address, QoS, and path information. (See 5.3.6.3 for an explanation of the FLOW-ID concept.) The process of replacement is not yet complete, so both concepts appear in the diagram above. The Data constitutes the application Service Data Unit (SDU) that crosses the interface. The SAP Metadata is information about a particular request or indication that crosses the interface.

5.3.6.2 Subnetwork Addresses

The subnetwork magenta books use a variety of terms for the usage of subnetwork addresses in the various subnetwork functional interfaces. These terms appear with their relationships in figure 5-19. The term ‘Service Access Point’, or ‘SAP’, appears frequently here. This term should be understood as an endpoint in a subnetwork where a device or a software entity can send and receive messages.

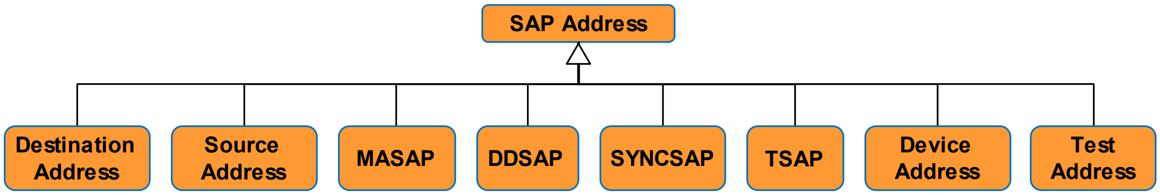


Figure 5-19: SO Subnetwork Addresses

An address in a subnetwork is an address that is defined by the standard of the subnetwork technology:

- SAP address: service access point address;
- a destination address identifies a device that is target of data transfer;
- a source address identifies a device that is the origin of data transfer;
- MASAP: memory access SAP address;
- DDSAP: device discovery SAP address;
- SYNCSAP: synchronization SAP address;
- TSAP: test SAP address;
- a device address identifies a device discovered or lost;
- a test address identifies a device that is the subject of a test operation.

5.3.6.3 Flow Identifiers

The concept of FLOW-ID was recently introduced. Instead of simply depicting a single endpoint of a data transfer, as in SAP addresses in the preceding subsection, a FLOW-ID depicts a pre-planned path between an origin SAP address and at least one destination SAP address. This change represents the system-level planning that goes into a network that must deliver some of its messages within a certain deadline.

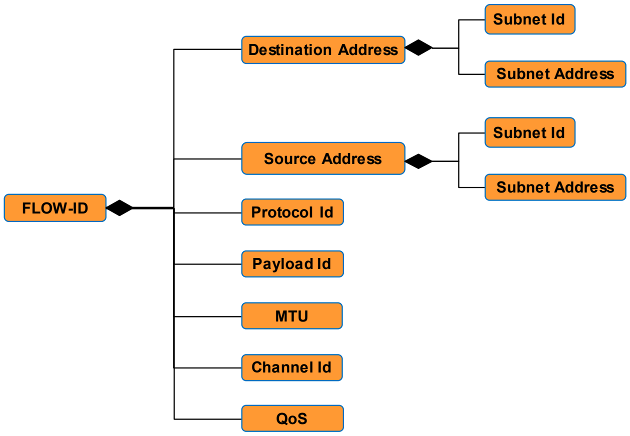


Figure 5-20: The Composition of a SO Flow Identifier

The parts of a FLOW-ID appear in figure 5-20.

- The destination and source addresses identify the endpoints of the flow; for multicast, the destination address may consist of multiple addresses. A subnetwork identifier denotes the technology of a subnetwork and discriminates when there is more than one subnetwork of the same technology onboard.

- The protocol identifier identifies the protocol that defines the format of the message. The protocol ID applies to the message and may span multiple subnetworks. In SpaceWire subnetworks, the protocol ID is explicit in messages and often establishes the purpose of a message within the subnetwork; however, the concept of protocol ID can be generalized to extend a system-level purpose of a message across multiple subnetworks.

- The payload identifier serves to identify packets or other application Protocol Data Units (PDUs) that do not carry clues to their own types.

- The Maximum Transmission Unit (MTU) limits the units of transfer along the path.

- The channel identifier is a designation across all subnetworks that reserves resources for delivery of data.

- The QoS contains additional qualifiers that apply across all subnetworks that specify a level of service, such as priority.

A FLOW-ID may be implemented as a row in a collection, indexed by a number. For multicast, multiple rows may be needed. In casual discourse, the term ‘FLOW-ID’ may refer to the index number or to the content of the row. When it is necessary to distinguish between index and content in this document, the words ‘index’ or ‘content’ are stated explicitly after ‘FLOW-ID’.

A FLOW-ID content is everything that is needed to deliver a message to an endpoint, or to recognize a message received from an endpoint, so it includes the SAP address (defined in the ISO BRM, reference [43]). The FLOW-ID contents for a given subnet layer reside in a MIB. Each processor has its own MIB with its own collection of FLOW-ID contents. Flows often run between a device and a processor, so there is often no device message traffic across multiple subnetworks. Flows between processors are typically between applications on a message bus, and do not require a Network Layer to span multiple subnetworks; instead, the flows are scheduled at system level during the design of a spacecraft and typically cross a single subnetwork. When it is necessary for a flow to cross multiple subnetworks, potentially of different technologies, then a designer might consider the trade between adding a Network Layer for adaptive routing and pre-configuring flows across processors for high QoS.

A FLOW-ID index is an index or some other arbitrary key to a collection of FLOW-ID contents. The value of the identifier may differ between a protocol stack in one processor and a protocol stack in another processor. Its value is fixed only while generating the flight software for a given processor so that flight software can use the index to access its MIB and find the flow information for communication with an endpoint represented by the FLOW-ID. The FLOW-ID index is the only part of a FLOW-ID that crosses the interface between a device service in the application support layer and a communication service in the subnet layer, such as the SO Packet Service. The FLOW-ID index does not travel with a message, because it is meaningless at the destination.

The FLOW-ID content does not necessarily travel with a message, although parts of it may appear in message headers, depending on the protocol layer below the SO communication service. For example, the SAP address could appear in a header to indicate the destination of a message in a protocol that could have routers along the path; this possibility depends on the subnet technology because the Data Link Layer may lack routers. The SAP address in the MIB would be expressed or translated as an address on the subnetwork. For SpaceWire, it would show up in a SpaceWire header as a logical address to be interpreted by routers or as a path string through routers.

The collection of FLOW-ID content is a part of the Deployment Description [Future]. The collection may be present onboard a spacecraft if the flight software passes FLOW-ID indexes across service interfaces when sending and receiving messages; typically, this design option is taken when there is an expectation to modify FLOW-ID content in flight through a management function. When FLOW-IDs will not be changed in flight, the collection may only exist statically on the ground during generation of device services.

5.3.6.4 Subnetwork Data

When an SO device service interacts with a subnet communication service of the SO subnetwork layer, an SDU passes between the two services. In the device service, the SDU is the PDU that passes virtually between the actual device and the device service. The device service is at ISO Layer 7 (Application Layer), and the subnet communication service is a shim that is just above ISO Layer 2 (Data Link Layer). ISO Layers 3 through 6 are often absent. The subnet communication service can be viewed as a reduced form of ISO layer 3, which provides a unified view of addressing and Data Link Layer functionality in one or more subnetworks. In the communication service, the SDU passes between the communication service and the Data Link Layer for the subnetwork. Along with the SDU, additional data passes between the device service and the communication service. This subsection identifies that additional data.

The additional data is loosely classified as plain data and as metadata in figures 5-21 and 5-22. The paragraphs that follow explain how that additional data may be used. (See 6.3 for how these data apply as parameters of SO service calls.)

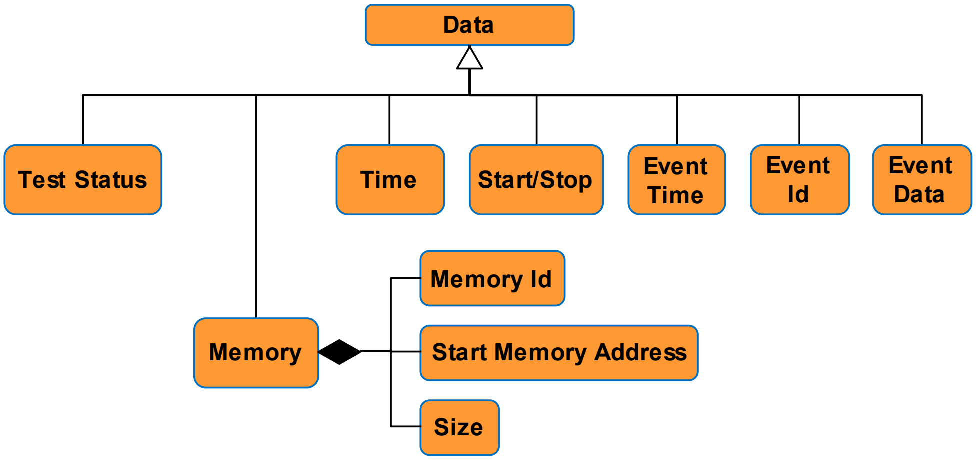


Figure 5-21: Subnetwork Data Payloads

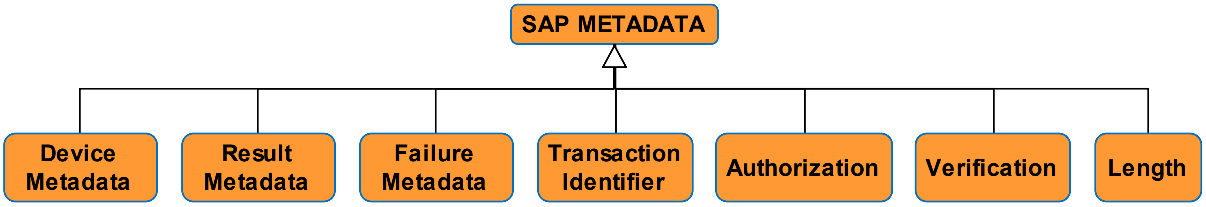


Figure 5-22: Subnetwork Metadata

For all services, the subnetwork addresses and FLOW-IDs are a part of the additional data, but have been described in 5.3.6.2 and 5.3.6.3.

When using the Packet Service to send a packet to a device, an application support service may provide a transaction identifier in order to associate the request with failure metadata in case a PACKET_FAILURE.indication is returned. The application support service may be written to refer the failure to its client, or the failure may be referred to a Fault Detection, Isolation, and Recovery (FDIR) service instead of the client. The treatment of the failure metadata is a mission-specific implementation option for a device service when it is generated by the agency tool chain.

When using the Memory Access Service, a device service may provide an internal transaction identifier in order to associate the request with result metadata returned asynchronously, similar to the Packet Service. To specify the memory to be accessed, the FLOW-ID destination address and source address are augmented with an identifier of a memory space, a start address in that space, and a size of the memory to be accessed in that space. Depending upon the QoS specified in the FLOW-ID, the device service may generate data to request acknowledgement (in a MEMORY_ACCESS_RESULT.indication), verification of writing, and authorization of request. In a revision [Future] to the Memory Access Service magenta book, the FLOW-ID may carry the acknowledgement, verification, and authorization data.

When using the Synchronization Service, the Time at which a Time.indication was generated may be requested. Notification of an identified event may be started or stopped and associated with data.

When using the Test Service, the Test Status may appear in a TEST.indication.

When using the Device Discovery Service, Device Metadata (describing a device) may be returned to a Device Enumeration Service [Future] when a device joins or leaves the subnetwork.

5.3.7 SO MANAGEMENT INFORMATION

The SO Management Information [Future] is defined in conjunction with the definition of the Deployment Description. The following categories of data are under consideration for inclusion in the management information for SO services. These categories also appear in figure 5-23.

- onboard device manifest [Future];
- FLOW-ID collection [Future];
- subnet collection [Future];
- MIB dictionary [Future].

The Onboard Device Manifest provides an inventory of devices that are onboard. For each device, the status and capabilities are available. The Device Discovery Service, when present, keeps this collection up to date.

The FLOW-ID collection associates FLOW-ID indexes with FLOW-ID content. (See figure 5-20 for the content of a FLOW-ID.)

The Subnet collection associates subnetwork properties with subnetwork identifiers. The maximum frame size is an example of such a property.

The MIB Dictionary is a set of single-valued management parameters. These parameters can be specified at design time in a mission-parametric SEDS and can be stored in a 'MIB Dictionary':

- out-of-sequence buffer size for SO sequence preservation convergence function [Future];
- unacknowledged buffer size for SO retry convergence function [Future].

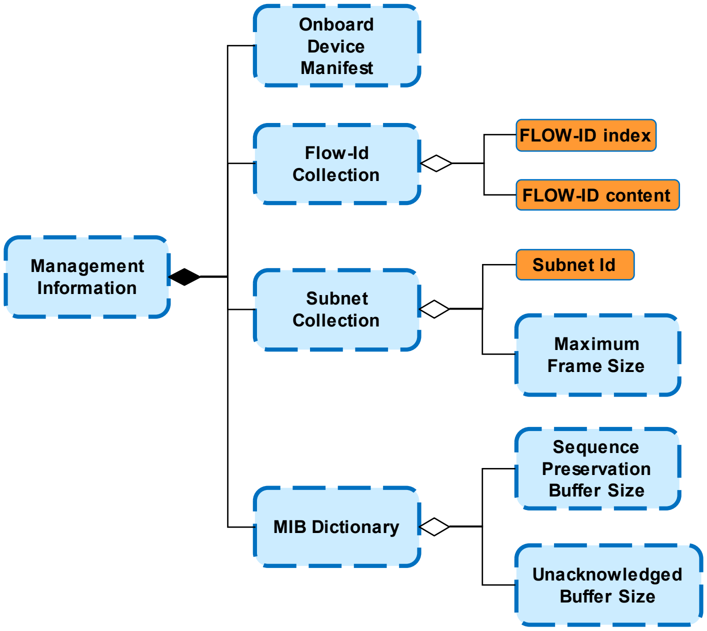


Figure 5-23: SO Management Information

5.4 SECURITY CONCEPTS FOR INFORMATION VIEWPOINT

In the Information Viewpoint, security topics may appear in two different forms, one is the nature of the kinds of data that may be secured, the other is the ways in which it may be secured. Some of these security mechanisms may be directly applied to the information objects themselves, such as using digital signatures or other means to authenticate the source of the data. Other mechanisms, such as encryption, may be used when personally identifiable data or data that is otherwise sensitive is being exchanged. The SCCS-ARD (reference [2]) and the CCSDS Security Architecture (reference [38]) describe the kinds of security mechanisms that CCSDS provides and how these may be employed.

These mechanisms may be employed whether information is ‘at rest’, in some repository, or in transit using communications protocols. The ‘in transit’ aspects are addressed in the Communications Viewpoint, section 7.

Neither MO nor SO functions make direct use of the available encryption or authentication capabilities for data that is at rest or for data that is in transit. As such, these capabilities are available to designers of systems that use these standards, but the mechanisms to implement them, use them, or exchange such data must rely upon these data being carried as an ‘opaque’ data payload. Any key management functionality must also be managed separately, and there is an SDLS protocol (reference [49]) being published that describes these functions.

There are some specific security terms that may be used in an Information Viewpoint:

a) Login Credentials: Typically some form of ‘username’ and a matching ‘password’. In practice, modern secure systems also often require a second factor for extra security.

b) Encryption algorithm: The algorithmic description of the cryptographic transformation of data (see cryptography) to produce ciphertext.

c) Digital signature: Data appended to, or a cryptographic transformation (see cryptography) of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, for example, by the recipient.

d) Cryptographic algorithm: The algorithmic description of the methods for the transformation of data in order to hide its information content, prevent its undetected modification and/or prevent its unauthorized use.

6 SERVICE VIEWPOINT

6.1 OVERVIEW

The Service Viewpoint identifies standard application-level interfaces between functions. A standardized service is specified in terms of the information exchanged and/or behavior of the interface, rather than to a specific pair of interfaced functions.

Services have already been identified in the Functional Viewpoint (section 4) as connections between functions, annotated with the information objects exchanged and identifying the service provider by a color-coded circle at one end. It is these services that are described in this viewpoint, in terms of the service operations supported and the information objects referenced by them. It also lists the functions that provide or consume each service. The protocols used at service interfaces to achieve interoperability are described in the Communications Viewpoint (section 7).

The Service Viewpoint is represented as a set of tables listing the identified service interfaces or data formats for each functional group. For each service, the table lists the functions interfaced, the information objects exchanged, operations/capabilities of the service/data exchanged, and references the CCSDS Recommended Standards that relate to this. The following is a summary of the table columns and color coding used (refer to 3.3.4 for the full definition).

Area Functional Area: color coding consistent with Functional Viewpoint.

Group Name or acronym for a group of related services or data formats.

Service Name of service or data format.

Functions List of provider/user functions from the Functional Viewpoint.

Operations List of service capabilities (groups of operations) that may be invoked.

Data List of information objects from the Information Viewpoint used by the service.

Description Description of the purpose of the service and its dependencies on other services.

Standards References to the CCSDS Recommended Standards relevant to the service.

S Status of Service Specification.

D Status of Data Format Specification.

NOTE – The last two columns in the table indicate the current status of service specification by CCSDS. It is divided into two columns to indicate whether there is a service and/or data format specification as these may be separately defined. The standardization status is color-coded consistent with the Functional and Information Viewpoints:

🟦	Blue	Published CCSDS Recommended Standard or CCSDS Recommended Practice (Blue Book or Magenta Book)
🟦	Blue/Light Blue	Published CCSDS Recommended Standard providing a partial solution
▨	Blue/White	CCSDS Recommended Standard under development
◼	Grey	Identified CCSDS Recommended Standard (Green Book or future road map)
□	White	No CCSDS Recommended Standard Identified

6.2 MO SERVICES

6.2.1 GENERAL

The MO services previously identified in the Functional Viewpoint (section 4) and described here fall into two main categories:

a) defined services with formalized interfaces and behavior;

b) interoperable data formats.

Defined services utilize online interactions between functions and define the associated behavior as a set of service operations that the service consumer may invoke upon the service provider. Interoperability is achieved by using the same set of communications-layer protocols to implement the service interface.

Interoperable data formats may be transferred as files or messages, but there is no defined standard interface behavior or protocol. Exchange of these data format standards may be implemented in a variety of ways, and the only assumptions made about the behavior at the ends of such an exchange is that the provider and user both understand the format and how to interpret it.

Several of the MO services have been, or are planned to be, defined in terms of the MO service framework. This framework is a set of CCSDS Recommended Standards that provide a way of specifying abstract MO services that can be mapped onto a variety of communications protocols and technologies. The framework includes:

- MAL;
- COM;
- Common Services;
- bindings for various communications technologies for encoding and transport of service messages;

– bindings that define one or more APIs for MO-compliant services in different programming languages.[4]

The remainder of the section is structured as follows:

a) **MO COM and Common Services**: lists generic and common services defined as part of the MO service framework, either in the MO COM or MO Common Services standards.

b) **Mission Control Services**: lists services associated with the Mission Control functional area. These services are defined in terms of the MO service framework.

c) **Navigation and Timing Services**: lists message formats and services associated with the Navigation and Timing functional areas. For Navigation, the majority of standards are currently defined as interoperable message formats. There are no current plans to develop Navigation or Timing Services. However Navigation services [Prospective] have been identified that could be based on the MO service framework and existing Navigation message formats.

d) **Mission Planning and Scheduling Services**: services supporting the Mission Planning and Scheduling functional area and based on the MO service framework are currently under development.

e) **Operations Preparation Services**: corresponding to the Operations Preparation functional area, this section identifies potential services and/or data formats to support the exchanges of configuration data for Mission Operations functions. Currently CCSDS only provides the XTCE Recommended Standard for exchange of telemetry and command definitions.

f) **Data Archiving Services**: services relating to the Mission Data Archive function and exchange of associated DAI.

g) **Other MOS**: lists additional services based on the MO service framework that are not associated with specific MO functional areas. These include:

- MDP distribution;
- file transfer and management.

[4] These APIs are an optional enabler for application portability and re-use. They do not, by themselves, ensure interoperability. That comes only from selection of compatible interface binding communication technologies for both provider and consumer. Provider and consumer may however use different programming languages, each using the corresponding API.

6.2.2 MO COM AND COMMON SERVICES

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟩	**MO COM**	***Archiving***	***Providers:*** Ops. Archive ***Consumers:*** <any function>	Store Update Query	<any COM obj>	Generic archive service for MO service s defined in terms of the Common Object Model (COM).	Reference [16], *Mission Operations Common Object Model*	🟦	🟦
🟪		***Activity Tracking***	***Providers:*** <any function> ***Consumers:*** <any function>	Publish/Subscribe	<any Activity>	Activities are COM objects that have a limited duration. The service provides a mechanism to report progress/status and uses the COM event service.			
🟪		***Event***	***Providers:*** <any function> ***Consumers:*** <any function>	Publish/Subscribe	<any Event>	Events are COM objects that represent an occurrence at a point in time. Each service can define the Events it supports.			
🟪	**MO Common**	**Directory**	***Providers:*** Service Directory ***Consumers:*** <any function>	Publish Provider Withdraw Provider Lookup Provider Get Service XML	Service Descriptor	Allows Providers to publish information about the services they provide; and Consumers to query the Service Directory and retrieve Service XML descriptors.	Reference [55], MO Common Services	🟦	🟦
🟪		**Login**	***Providers:*** Login and Authentication ***Consumers:*** <any function>	Log in Log out Report Available Roles Handover to other users	Authentication Credentials	Common login service for submission of authentication details to a deployment specific security system. Integrated with Access Control aspect of MAL.			
🟨		**Configuration**	***Providers:*** Configuration Management & Distribution ***Consumers:*** <any function>	Activate List Get Current Get XML Add Remove Store Current Store XML	Configurations XML Configurations <any Config Data>	Configurations can be hard-coded, use bespoke configuration data, or a standard COM service configuration. Service consumers can activate predefined configurations of a service provider and list, get, add, remove, and store current configurations. It also defines a standardized XML representation for configurations.			

6.2.3 MISSION CONTROL SERVICES

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟥	**MO M&C**	**Action**	***Providers:*** Monitoring & Control ***Consumers:*** Automation Onboard Configuration Management Navigation Interface Planning Plan Execution Operations Preparation Operations Archive User Support Mission Data Processing Spacecraft Development & Maintenance	Submit Action preCheck Action Manage Action Definitions	Action	Allows control directives (e.g., a spacecraft telecommand) to be invoked and their evolving status to be monitored. Uses COM Services for Action Tracking and Archiving.	Reference [17], *Mission Operations Monitor & Control Services*	🟦	🟦
		Parameter		Monitor Value Get Value Set Value Enable Generation Manage Parameter Definitions	Parameter	Provides the capability to monitor and set parameter values. Uses COM Archiving service for parameter archiving.			
		Alert		Enable Generation Manage Alert Definitions	Alert	Provides a mechanism for asynchronous notification of operationally significant events or anomalies. Uses COM event and archiving services to publish/subscribe to alerts and to archive them.			
		Check		Get Current Transition List Get Summary Report Enable Service Get Service Status Enable Check Trigger Check Manage Check Definitions	Parameter Check Check Link Check Transition Event	Provides online checking of parameter values against defined checks (Limit, Constant, Delta) and notification of check violations. Uses COM event service to publish/subscribe to check status transition events.			

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟥		**Statistics**		Get Statistics Reset Evaluation Monitor Statistics Enable Service Get Service Status Enable Generation Add Parameter Evaluation Update Parameter Evaluation Remove Parameter Evaluation	Parameter Statistic Link Statistic Value	Provides online statistical evaluation of parameter values. Uses COM Archive service.		🟦	🟦
🟥		**Aggregation**		Monitor Value Get Value Enable Generation Enable Filter Manage Aggregate Definitions	Aggregation of Parameters	Provides aggregation of separate parameter values into coherent sets. Uses COM Archive service.		🟦	🟦
🟥		**Conversion**		none	Parameter Conversion	Provides conversion of raw parameter values into engineering units. Uses COM Archive service		🟦	🟦
🟥		**Group**		none	Group of <any COM object>s	Provides the ability to define groupings of objects to simplify the operations of other services. Uses COM Archive service		🟦	🟦
🟥	**MO AUT**	**Automation** [Prospective]	***Providers:*** Automation ***Consumers:*** Monitoring and Control Navigation Interface Plan Execution Operations Archive	Start Procedure Stop Procedure Suspend/Resume Procedure Manual Control Manage Procedure Definitions [TBD]	Procedure	Provides support for automation of mission operations. The service allows automated procedures or autonomous functions to be invoked, controlled, and their evolving status to be monitored. Uses COM Services for Procedure Tracking and Archiving.	Reference [54], MO service s Concept Reference [57], *Mission Operations—Automation Service*	▨	▨
🟥	**MO OBSM**	**Software Management** [Prospective]	***Providers:*** Onboard Configuration Management ***Consumers:*** Automation Plan Execution	Load Software Image Dump Software Image Check Software Image	Onboard Software Image	Supports the management of software loaded into the remote system (spacecraft).	Reference [54], MO service *s Concept* Reference [58], *Mission Operations—Software Management Service*	▨	▨
🟥	**MO OBPM**	**Procedure Management** [Prospective]	***Providers:*** Onboard Configuration Management ***Consumers:*** Automation Plan Execution	Load Procedure Definition Dump Procedure Definition Check Procedure Definition Manage Procedure Definitions [TBD]	Onboard Procedure Definition	Supports the management of automated procedure definitions loaded into the remote system (spacecraft).	Reference [58], *Mission Operations—Software Management Service*		

6.2.4 NAVIGATION AND TIMING SERVICES

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
	TIM	**Time** [Future]	***Providers:*** Time/Position Determination ***Consumers:*** Time Correlation	Report Time Set Time Configure rate of Time Report generation.	Time Report	Provides accurate reporting of onboard time. For an unsynchronized onboard clock this may require correlation with the system reference time	Reference [54], MO service *s Concept* Reference [59], *Mission Operations—Time Services*		
		Time Correlation [Future]	***Providers:*** Time Correlation ***Consumers:*** Mission Control Mission Data Processing	Correlate Time	Time Correlation	Supports Time correlation between onboard clocks and the system reference time.	Reference [54], MO service *s Concept* Reference [59], *Mission Operations—Time Services*		
		Time Reception [Partial]	***Providers:*** TT&C ***Consumers:*** Time Correlation	Report Reception Time	Reception Time	Provides accurate ground reception time reporting that can be associated with a Time Report. This is currently supported by the CSS Space Link Extension Transfer Services, where Earth Receive Time is provided as an annotation parameter to the transfer data. An alternative service may be required where SLE is not used.	Reference [46], *Space Link Extension—Return All Frames Service Specification* *And related CSS SLE Transfer Services*		
	NAV	**Navigation Services** [Prospective]	***Providers:*** Navigation Functions ***Consumers:*** Mission Control Mission Planning & Scheduling Mission Data Processing Data Storage & Archiving User Support TT&C Satellite Dev. & Maintenance	Request Navigation Message Retrieve Navigation Message Subscribe to Navigation Message	Orbit Vector Attitude Tracking Data Predicted Orbital Events Conjunction Data Re-entry Data Pointing Request	Supports the provision of spacecraft positioning information such as: — Position reports (e.g., from onboard GPS) — Spacecraft ranging and range-rate measurements — Antenna tracking azimuth and elevation — Orbit vectors — Attitude vectors — Trajectory requests — Predicted orbital events (including ground station visibilities) The services will use the following data message formats defined by the CCSDS Navigation working group, but wrap these as service specifications based on the MO framework.	Reference [54], MO service *s Concept* Reference [60], *Mission Operations—Navigation Services*		

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
		ODM (data format only)	***Providers:*** Orbit Determination and Propagation ***Consumers:*** Maneuver Planning Attitude Determination Conjunction Assessment Mission Control Mission Planning & Scheduling Mission Data Processing TT&C User Support	-	Orbit Vector	The ODM contain information that defines the orbit state of a spacecraft at one or more times.	Reference [25], *Orbit Data Messages*		
		ADM (data format only)	***Providers:*** Attitude Determination ***Consumers:*** Orbit Determination and Propagation Maneuver Planning Mission Control Mission Data Processing User Support	-	Attitude	The ADM contain information that defines the attitude state of a spacecraft at one or more times.	Reference [27], *Attitude Data Messages*		
		TDM (data format only)	***Providers:*** Time/Position Determination TT&C Mission Data Processing ***Consumers:*** Orbit Determination and Propagation	-	Tracking Data	The TDM contains information that can be used to determine the orbit state of a spacecraft.	Reference [26], *Tracking Data Message*		
		NEM [Future]	***Providers:*** Orbit Determination and Propagation ***Consumers:*** Attitude Determination Maneuver Planning Mission Planning & Scheduling Mission Control	-	Predicted Orbital Events	The NEM contains the predicted timings of orbital events, such as ground station visibilities, sensor blindings, and eclipses.	Reference [64] *,Navigation Events Message*		

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
■		**CDM** (data format only)	***Providers:*** Conjunction Assessment ***Consumers:*** Maneuver Planning Mission Planning & Scheduling	-	Conjunction Data	The CDM contains information that defines the relationship between the orbit states of different space objects at different times.	Reference [28], *Conjunction Data Message*	■	■
		RDM (Data Format only)	***Providers:*** Re-entry Assessment ***Consumers:*** Maneuver Planning Mission Planning & Scheduling	-	Re-entry Data	The RDM contains information about a single re-entry event of a natural or man-made object entering the atmosphere of the Earth or another planet.	Reference [65], *Re-entry Data Message*	■	▨
		PRM (Data Format only)	***Providers:*** User Support Mission Data Processing ***Consumers:*** Attitude Determination Maneuver Planning Mission Control	-	Pointing Request	The PRM contains information on the pointing of a spacecraft or instrument desired by a mission user at one or more times.	Reference [29], *Pointing Request Message*	■	▨

6.2.5 MISSION PLANNING AND SCHEDULING SERVICES

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
	MO MPS	**PRS** [Future]	***Providers:*** Planning ***Consumers:*** User Support Operations Preparation Navigation & Timing Planning [Distributed]	Submit Request Update or cancel Requests Provide Request Status feedback	Planning Request Plan Planning Activity Planning Event Planning Resource	Asynchronous submission of PRQs, associated responses and their subsequent management, and status feedback. A PRQ may reference a Plan (output from an earlier planning process), when the provided feedback includes the status of the plan and its contained activities and other items.	Reference [54], MO service *s Concept* Reference [62], *Mission Planning and Scheduling* Reference [63], *Mission Planning and Scheduling Services*		
		PDS [Future]	***Providers:*** Planning ***Consumers:*** User Support Navigation & Timing Mission Data Processing Planning [Distributed]	Retrieve Plan or Plan Status Subscribe to Plan or Plan Status	Plan Planning Activity Planning Event Planning Resource	Provides distribution and access to plans generated by the planning function and their status.			
		PIM [Future]	***Providers:*** Planning Plan Execution ***Consumers:*** Plan Execution Mission Control Navigation & Timing User Support Mission Data Processing Operations Preparation Planning [Distributed]	List Planning Definitions Retrieve Planning Definitions Add, Update, and Remove Planning Definitions	PDB PRQ Template Planning Activity Planning Event Planning Resource	Access to and Management of Mission Planning configuration data, specifically the definitions of Planning Activities, Planning Events, Planning Resources, and PRQ Templates.			
		Plan Execution Control Service (PEC) [Future]	***Providers:*** Plan Execution ***Consumers:*** Planning Mission Control	Submit, Activate, Deactivate, Monitor, and Control Plan execution	Plan Plan Execution Control Data	Control and management of the execution of a plan, including actions to Submit Plans and Activate/Deactivate their execution. Provide execution status updates at the level of Plans and their contained Planning Activities, Events, and Resources.			
		Plan Edit Service (PED) [Future]	***Providers:*** Plan Execution ***Consumers:*** Mission Control Navigation & Timing	Insert, Update, and Delete Planning Activities and Planning Events Update Planning Resources	Plan Planning Activity Planning Event Planning Resource	Edit content of a currently executing Plan at the level of its contained Planning Activities, Events, and Resources.			

6.2.6 OPERATIONS PREPARATION SERVICES

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟨	**OPD**	**SDB** [Partial]	***Providers:*** Satellite DB Definition Spacecraft Development & Maintenance ***Consumers:*** Mission Control Spacecraft Development & Maintenance	-	Satellite DB	Contains definition of Telemetry Data, Telecommands, and Events present in the TM/TC interface with the spacecraft and represented within the Mission Control System. XTCE provides an exchange format for TM/TC data between systems.	Reference [23], *XML Telemetric and Command Exchange—Version 1.2*		🟦
		APD [Prospective]	***Providers:*** Automated Procedure Definition Spacecraft Development & Maintenance ***Consumers:*** Mission Control Spacecraft Development & Maintenance	-	Automated Procedure	Definition of an operational procedure that can be automatically executed within a space system (either onboard a spacecraft, or within the mission control system).	**No CCSDS Recommended Standard.** Reference [70] defines a standard model for a procedure, but not a normative representation.		
		PDDs [Prospective]	***Providers:*** Planning DB Definition ***Consumers:*** Mission Planning & Scheduling	-	PDDs	Definitions of Planning Data (Activities, Events, Resources, and Constraints) and potentially of Planning Rules.	**No CCSDS Recommended Standard** Reference [62], *Mission Planning and Scheduling* introduces Planning Data Model.		
		OBSW [No Standard]	***Providers:*** Onboard Software Definition Spacecraft Development & Maintenance ***Consumers:*** Mission Control Spacecraft Development & Maintenance	-	Onboard Software	Onboard Software Image	**No CCSDS Recommended Standard**		

6.2.7 DATA ARCHIVING SERVICES

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟩	**DAI**	**PAIS** [Prospective (service)]	***Providers:*** Data Archive Ingestion ***Consumers:*** Mission Data Processing Operations Archive	[TBD]	SIP SIP Sequencing Constraint Transfer Object Collection Descriptor	The current PAIS standard provides the abstract syntax and an XML implementation of descriptions of data to be sent to an archive. It addresses how these data will be aggregated into packages for transmission and one concrete implementation for the packages based on the XML Formatted Data Unit (XFDU) standard. A service specification is proposed but has not yet been developed.	Reference [30], *Reference Model for an Open Archival Information System (OAIS)* Reference [31], *Producer-Archive Interface Specification (PAIS)* Reference [37], *XML Formatted Data Unit (XFDU) Structure and Construction Rules*	▨	🟦
🟩		**CAIS** [Future] [Prospective (service)]	***Providers:*** Data Archive Access ***Consumers:*** User Support Mission Data Processing Spacecraft Dev. & Maint.	[TBD]	Dissemination Information Package (DIP)	Delivery of digital sources from the Archive.	Reference [30], *Reference Model for an Open Archival Information System (OAIS)*	▨	🟦
🟩		**Archive Storage and Retrieval** (data format only) (no standard service)	***Providers:*** Data Archive Storage ***Consumers:*** Data Archive Ingestion Data Archive Access	[TBD]	AIP	Storage and Retrieval of standard AIPs. Abstract specification of an AIP may be expressed using PVL, EAST, or DEDSL languages.	Reference [30], *Reference Model for an Open Archival Information System (OAIS)* Reference [32], *Parameter Value Language Specification (CCSD0006 and CCSD0008)* Reference [33], *The Data Description Language EAST Specification (CCSD0010)* References [34], [35], and [36], *Data Entity Dictionary Specification Language (DEDSL)*		🟦

6.2.8 OTHER MO SERVICE S

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟦	**MO MDP**	**Data Product Distribution** [Future]	***Providers:*** Operations Archive Onboard File Store ***Consumers:*** Mission Data Processing User Support Spacecraft Development & Maintenance	Subscribe Online Subscribe Batch Retrieve [TBD]	Mission Data Product	Mission Data Product Distribution Service is used for the distribution of historical archived data and online 'live' data. It provides two delivery modes, batch mode and stream mode.	Reference [56], *Mission Operations—Mission Data Product Distribution Services*	🟦	🟦
🟩	**MO FTM**	**File Transfer & Management** [Future]	***Providers:*** Onboard File Store ***Consumers:*** Mission Control Planning & Scheduling Navigation and Timing	List, Rename, Move, Copy, and Delete Files and Directories Add Directory Get Drive Information Upload File Download File	Directory File Drive	Supports the management of a remote (onboard) file store and the initiation of transfers of files between local and remote file stores.	Reference [54], MO service *s Concept* Reference [61], *Mission Operations—File Transfer and Management Services*	▨	▨

6.3 SO SERVICES

The table of services for SO appears below. The colors in the leftmost column correspond to the colors in figure 3-3. The colors in the rightmost two columns are defined in 3.3.4.

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟨	**Subnet**	**Packet Service**	***Providers:*** Packet Service ***Consumers:*** EDS-Derived Device Services	PACKET_SEND, PACKET_RECEIVE, PACKET_FAILURE	FLOW-ID, Data, Length, Transaction-Id, Failure Metadata	Provides means to read or to write packets from or to devices.	Reference [5], *Spacecraft Onboard Interface Services—Subnetwork Packet Service*	🟦	
🟨		**Memory Access Service**	***Providers:*** Memory Access Service ***Consumers:*** EDS-Derived Device Services	READ, WRITE, READ/MODIFY/WRITE, MEMORY_ACCESS_RESULT	MASAP, Destination Address, Transaction Id, Memory Id, Result Metadata	Provides means to read or to write data from or to memory of a device.	Reference [6], *Spacecraft Onboard Interface Services—Subnetwork Memory Access Service*	🟦	
🟨		**Device Discovery Service**	***Providers:*** Device Discovery Service ***Consumers:*** EDS-Derived Device Services	DEVICE_DISCOVERY, DEVICE_DISCOVERY_LOSS	DDSAP, Device Address, Device Metadata	Detects devices attached to subnetworks; detects connection and disconnection of devices to and from subnetworks; notifies management functions.	Reference [7], *Spacecraft Onboard Interface Services—Subnetwork Device Discovery Service*	🟦	
🟨		**Synchronization Service**	***Providers:*** Synchronization Service ***Consumers:*** EDS-Derived Device Services	TIME, EVENT	SYNCSAP, Time, Event Id, Event Time, Event Data, Start/Stop,	Provides means to maintain knowledge of time that is common across a single subnetwork.	Reference [8], *Spacecraft Onboard Interface Services—Subnetwork Synchronisation Service*	🟦	
🟨		**Test Service**	***Providers:*** Test Service ***Consumers:*** EDS-Derived Device Services	TEST	TSAP, Test Address, Test Status	Provides basic capability to request tests of devices and to receive the results of the tests.	Reference [9], *Spacecraft Onboard Interface Services—Subnetwork Test Service*	🟦	
🟨		**DSPP**	***Providers:*** DSPP ***Consumers:*** Packet Service	<As specified in SEDS>	<As specified in SEDS>	Provides specialized framing for devices, such as on 1553 subnetworks.	Reference [10], *Spacecraft Onboard Interface Services—XML Specification for Electronic Data Sheets*	▨	
🟨		**Management Information Service [Future]**	[TBD]	[TBD]	[TBD]	Provides management information for spacecraft onboard information services.		▨	

A	Group	Service	Functions	Operations	Data	Description	Standards	S	D
🟩	**App**	**DSAP**	***Providers:*** DSAP ***Consumers:*** Applications	<As specified in SEDS>	<As specified in SEDS>	Provides access between devices and applications for raw data.	Reference [10], *Spacecraft Onboard Interface Services—XML Specification for Electronic Data Sheets*		🟦
🟩		**DACP**	***Providers:*** DACP ***Consumers:*** Applications	<As specified in SEDS>	<As specified in SEDS>	Provides access between devices and applications for functional data.	Reference [10], *Spacecraft Onboard Interface Services—XML Specification for Electronic Data Sheets*		🟦
🟩		**Device Data Pooling Service [Prospective]**	***Providers:*** DDPS ***Consumers:*** Applications	[TBD]	[TBD]	Provides a current value table.	[TBD]		🟦
🟩		**Time Access Service [Prospective]**	***Providers:*** TAS ***Consumers:*** Applications	[TBD]	[TBD]	Provides access to clocks onboard for time correlation and synchronization.	[TBD]		🟦
🟩		**File and Packet Store Services [Prospective]**	***Providers:*** FPSS ***Consumers:*** Applications	[TBD]	[TBD]	Provides access to onboard file systems and packet stores.	[TBD]		🟦
🟩		**message transfer service [Prospective]**	***Providers:*** message transfer service ***Consumers:*** Applications	[TBD]	[TBD]	Provides subset of Asynchronous Message Service, or another message routing implementation.	[TBD]		🟦
🟩		**Device Enumeration Service [Prospective]**	***Providers:*** DES ***Consumers:*** Applications	[TBD]	[TBD]	Provides discovery of devices onboard.	[TBD]		🟦
🟦		**Deployment Description [Future]**		<access, update>	<subnetwork topology, device addresses within subnetworks>	Defines topology, traffic flows, and schedules of subnetworks.			▨

6.4 SECURITY CONCEPTS FOR SERVICE VIEWPOINT

The system elements that provide user services will typically be secured in a number of ways, both physically and functionally. The following security methods are likely to be employed in the implementation of service systems, but only a few of these are provided as actual services:

- operational staff will be required to log in to the operational systems to access and control services;
- operational staff will have assigned roles and access controls, as appropriate;
- all service interfaces will be secured and require some sort of access credentials;
- users will be required to log in to the system management interfaces in order to plan and schedule services;
- users will be required to log in to the system management interfaces in order to request, monitor, and control services;
- users will be required to log in to the service execution interfaces in order to send and receive data;
- different users may have different roles and access credentials;
- users will be required to establish a service contract with the service provider before services may be accessed.
- on-orbit servicing and various kinds of cross support will require their own service agreements, but this is not the norm for typical missions.

The security sections of the MO and SO documentation and the CCSDS Security Architecture and Cryptographic Algorithms documents (references [38] and [39]) provide more details about service security interfaces and appropriate authentication approaches.

Physical security is addressed in 8.4; other types of communications link security, such as link and network encryption, are addressed in section 7.

7 COMMUNICATIONS VIEWPOINT (PROTOCOL STACKS)

7.1 OVERVIEW

The Communications Viewpoint shows how the interfaces of application-level services are defined by service protocols and by underlying communications protocol stacks, which may differ depending on their deployment context. This viewpoint should be read in conjunction with the SCCS-ADD (reference [50]), which identifies two principal kinds of deployment contexts for space links (ABA and SSI) together with their associated lower-layer communications protocol stacks. The space link and network protocols, and service interfaces like SLE and CSTS, provide the underlying communication 'fabric' that supports communications between terrestrial systems and spacecraft of all types.

For purely terrestrial deployments, the availability of TCP/IP and the rest of the Internet Protocol Suite is assumed, including typical Data Link Layer protocols such as Ethernet, Wi-Fi, USB, and LTE. In the spacecraft environment, approaches have been documented for onboard Data Link Layer protocols and subnets, and wireless approaches such as Wi-Fi and LTE are also in development.

7.2 ISO PROTOCOL STACK AND LAYER DEFINITIONS

7.2.1 GENERAL

Protocol configurations are usually described as a 'stack' of protocols, showing how the functions at the various layers, from Physical Layer through Application Layer, are provided. Figure 7-1 shows an abstract view of the nominal ISO protocol stack as defined within the CCSDS Recommended Standards suite. As in many terrestrial applications, some of the layers in the full ISO stack have been left out (Session, Presentation), and some new functions derived from the Physical Layer (radiometric, time) have been added. Not shown explicitly is the further distinction that is usually made for space-link communications that separately treats coding and synchronization as the sublayer at the 'bottom' of the Data Link Layer, and modulation as the 'top' part of the RF Physical Layer signaling, along with the relevant RF (or future optical) frequency spectrum (the rest of the Physical Layer). The protocol data types associated with each layer are shown in dashed boxes.

The definitions of the layers used throughout this document are derived directly from the ISO Basic Reference Model (reference [43]), modified slightly to align with CCSDS terminology. The Session and Presentation Layers are left out, although they often do have an identifiable role within systems.

For many terrestrial applications, and even some in space, an intermediate 'Application Support' layer between Layer 7 (Application) and Layer 4 (Transport) may be adopted, providing data transfer services for messages and files. These may be generic, data-content neutral services separate from the Application Layer, 'business logic', specific exchanges of requests and responses among distributed systems elements. The ISO BRM predates these kinds of considerations, but clearly identifying these in modern distributed systems approaches has become essential.

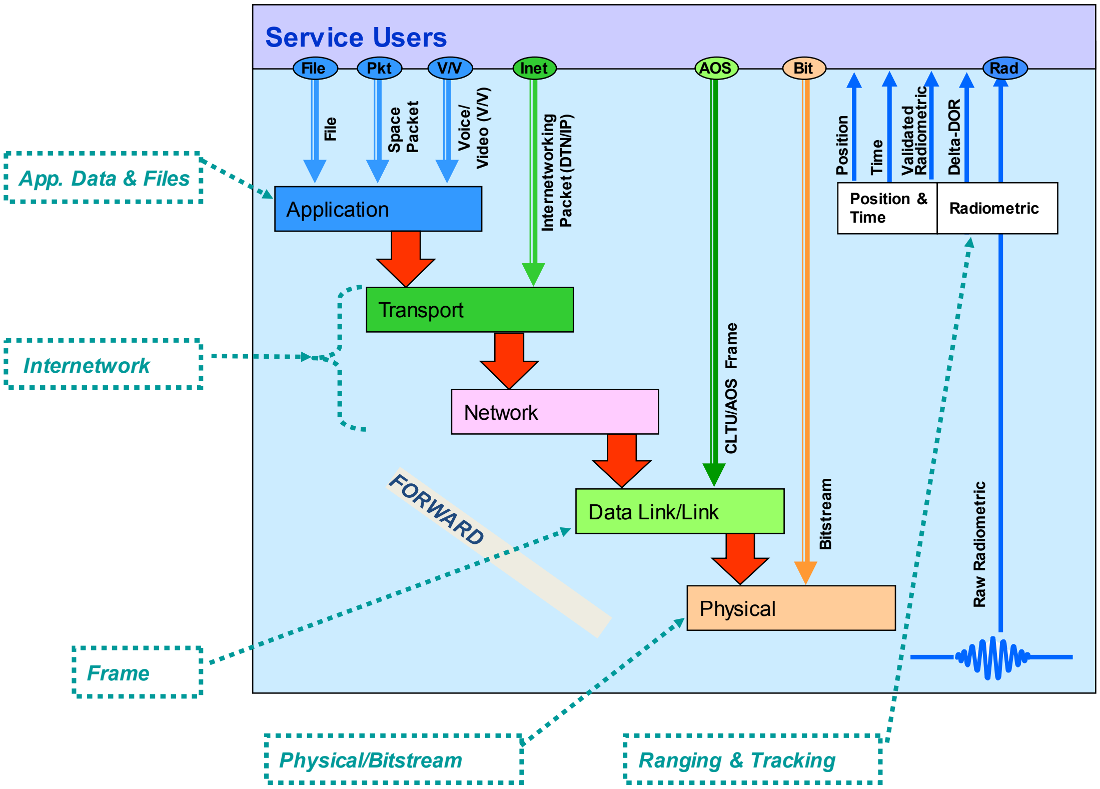


Figure 7-1: ISO Protocol Stack, with Notes Showing Data at Each Layer

The following subsections define the functions at each layer and then use these to provide protocol stack building-block diagrams showing use of these protocol layers. For the basic ABA Data Link Layer deployments, there is usually not a Transport or Network Layer, since user applications interface directly either to application services (file or message) transfer, or to the Data Link Layer itself. For the SSI configurations, the Transport and Network Layers will be used to do routing and end-to-end delivery. Message transfer layers may exist separately on the ground (MO MAL) and in space (AMS or onboard message bus). For cases in which some of the MO service s have migrated into the flight environment, message transfer layers may extend end-to-end, across the space link, both in ABA configurations and in SSI configurations using network protocols. SO uses Application Layer, Data Link (subnet) Layer, and Physical Layer, but calls the merged Transport/Network Layers ‘transfer’ when it is present. This is not substantially different from figure 7-1, which is all about the relationships among OSI layers, the functions they provide and the PDUs they use.

7.2.2 APPLICATION LAYER

The OSI Application Layer (Layer 7) contains all those functions that imply communication between open systems that are not already performed by the lower layers. These include functions performed by programs as well as functions performed by human beings. An application entity can be structured internally into Application Layer objects representing groups of functions. The Application Layer may also provide security services such as authentication, integrity, and confidentiality, either in addition to any such services provided at lower layers in the protocol stack.

For our purposes the OSI Application Layer may be thought of as having two sublayers:

a) application functions that provide ‘business logic’ services;

b) application support functions that provide applications with content-neutral data transfer services.

The features normally associated with the OSI Presentation and Session Layers are subsumed into the data encodings that are adopted, and any authentication features built into the Application Layer implementations.

7.2.3 TRANSPORT LAYER

All protocols defined in the Transport Layer (Layer 4) have end-to-end significance, where the ends are defined as transport entities. The Transport Layer is relieved of any concern with routing and relaying since the Network Layer provides data transfer from one network node to any other across one or more subnetworks. The Transport Layer provides transparent transfer of data between applications and relieves them from any concern with the details of how reliable and complete transfer of data is achieved. SO subnetworks often omit a Transport Layer when reliability and congestion is managed by the Data Link Layer.

7.2.4 NETWORK LAYER

The Network Layer (Layer 3) provides transport entities independence from routing and relay considerations across successive network nodes. It provides the means to establish, maintain, and terminate network connections between open systems containing communicating applications and the functional and procedural means to exchange network service data units between transport entities over network connections. The Network Layer may also provide security services such as authentication, integrity, and confidentiality. SO subnetworks often omit a Network Layer when addressing is known to applications and traffic is managed by the Data Link Layer.

7.2.5 DATA LINK LAYER

7.2.5.1 General

The Data Link Layer (or simply ‘link layer’) (Layer 2) provides functional and procedural means for a connectionless or connection-oriented mode for the establishment, maintenance, and release of data link connections between pairs of Network Layer entities and for the transfer of data link service data units. A data link connection is built upon one or several physical connections. The Data Link Layer detects and possibly corrects errors that may occur in the Physical Layer (see ‘Coding and Synchronization Sublayer’). SO data links often have their own built-in properties for reliability and timeliness that control scheduled communications between endpoints. The Data Link Layer may also provide security services such as authentication, integrity, and confidentiality.

7.2.5.2 Coding and Synchronization Sublayer

CCSDS specifies a Coding and Synchronization Sublayer (of the Data Link Layer) and defines methods for synchronization and channel coding for successfully sending transfer frames over a space link. For the Space Data Link Layer protocols it provides error detection and correction and deals with noisy, low SNR, space link physical channel characteristics. For SO subnet protocols it provides error detection, possible error correction, and deals with reliability concerns.

7.2.6 PHYSICAL LAYER

The Physical Layer (Layer 1) provides the mechanical, electrical, functional, and procedural means to activate, maintain, and de-activate physical connections for bit transmission between Data Link Layer entities. The Physical Layer provides for the transparent transmission of bit streams between data link entities across physical connections. The services provided by the Physical Layer are determined by the characteristics of the underlying medium and are too diverse to allow categorization. The Physical Layer may also provide security services such as authentication, integrity, and confidentiality.

7.3 SPECIFIC PROTOCOLS FOR MO SERVICE INTERFACE BINDING

7.3.1 GENERAL

MO service standards provide end-to-end application-level information exchange and are defined in terms of two principal information exchange paradigms at layer 7:

– Message Based Interaction (bidirectional message exchange);

– File Transfer (unidirectional message transfer).

For message-based interaction, MO services are defined in a way that is abstracted from the underlying communications technology. This allows them to be deployed in different contexts, but interoperability depends upon selection of appropriate technology bindings.

In the case of file transfer, MO standards use defined message formats that can be encoded in a file. A specific means of effecting the file transfer is not imposed. The application is expected to use an existing standard (or bespoke) protocol for achieving this, such as FTP (terrestrial) or CFDP (over a space link).

The description of the specific protocols for MO service interface binding in the remainder of this section is structured as follows:

a) communications deployment contexts;

b) generic protocol stack;

c) space link context;

d) ground context;

e) onboard context;

f) context bridging.

7.3.2 COMMUNICATIONS DEPLOYMENT CONTEXTS

In terms of the underlying communications architecture, MO services, and message formats may be deployed in three principal contexts (illustrated in figures 7-2, 7-3, and 7-4, respectively):

– **Space Link**: using CCSDS communications protocol stacks compliant with those described in the SCCS-ARD. The ARD describes the communications architecture for two primary cases:

  - ABA: a point-to-point space communications configuration that involves a single direct link from spacecraft to ground. This includes the use of CCSDS CSS SLE services to extend the space link from a terrestrial ground station to a MOC or other ground facility. The term ABA derives from the case of a spacecraft and

MOC owned by Agency A using a TT&C ground station owned by Agency B, and hence relying upon interoperability between different Agency's systems.

- SSI: a networked space communications configuration that involves using SSI protocols deployed over potentially multiple nodes, space links, and space or terrestrial networks. It assumes a loose confederation of independent space communications networks, each often owned and administered by a different space agency, that all share a single Network Layer protocol to allow them to interoperate and exchange Network Layer messages (PDUs).

– **Ground**: across a Terrestrial Network, using industry standard communications protocol stacks such as TCP/IP, Ethernet, Wi-Fi, and LTE.

– **Onboard**: within a Spacecraft, using a SO-compliant architecture.

Initially, MO services may only be deployed in the Ground context, but as a standard technology binding to CCSDS Space Packet Protocols [3] has already been defined[5], there is also potential for deployment in space link and onboard contexts. (See 9.5 for an extended discussion of this topic.)

Bridging between these separate communications contexts is performed at application level. However, when the MAL and fully compliant technology mappings are used, a generic MAL bridge may be able to provide this as an application-independent service. Depending on the nature of the specific technology mappings, additional conversions and out-of-band 'managed parameters' may be needed to make this work. This is discussed further in 7.3.7 below.

NOTE – Space agencies may own their own ground stations and may choose to use bespoke approaches to control and transfer data through their ground stations. Only CCSDS-compliant deployments are directly addressed.

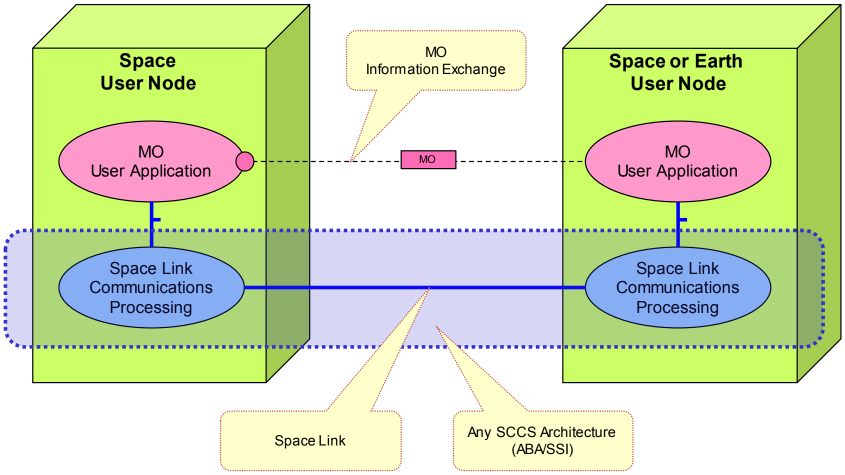


Figure 7-2: MO Space Link Communications Context

[5] There is no sensible binding of the MAL to the Encapsulation Packet Protocol, but the latter could be used in a protocol stack relative to CFDP or DTN.

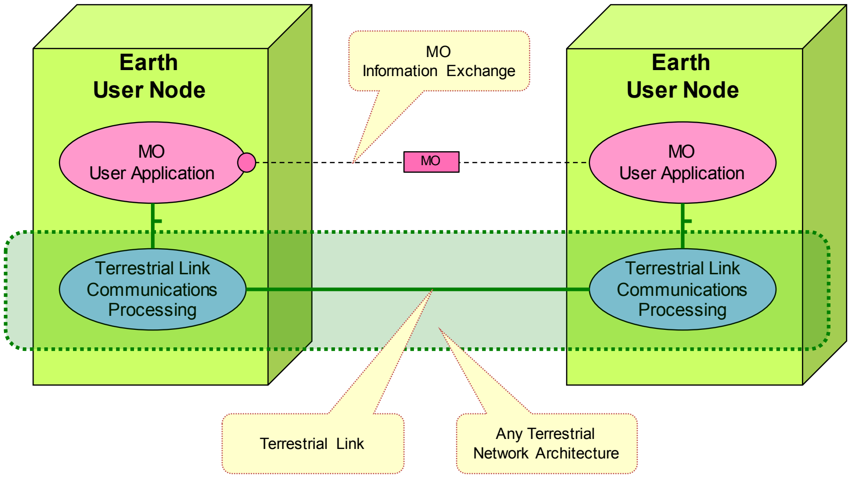


Figure 7-3: MO Terrestrial Link Communications Context

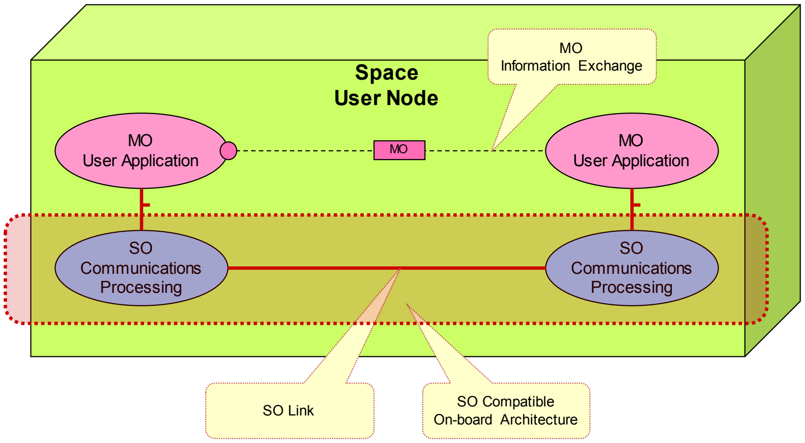


Figure 7-4: MO Onboard Link Communications Context

7.3.3 GENERIC PROTOCOL STACK

The following diagram shows the generic protocol stack applicable to all communications deployment contexts for both MAL-compliant services and MO file exchange.

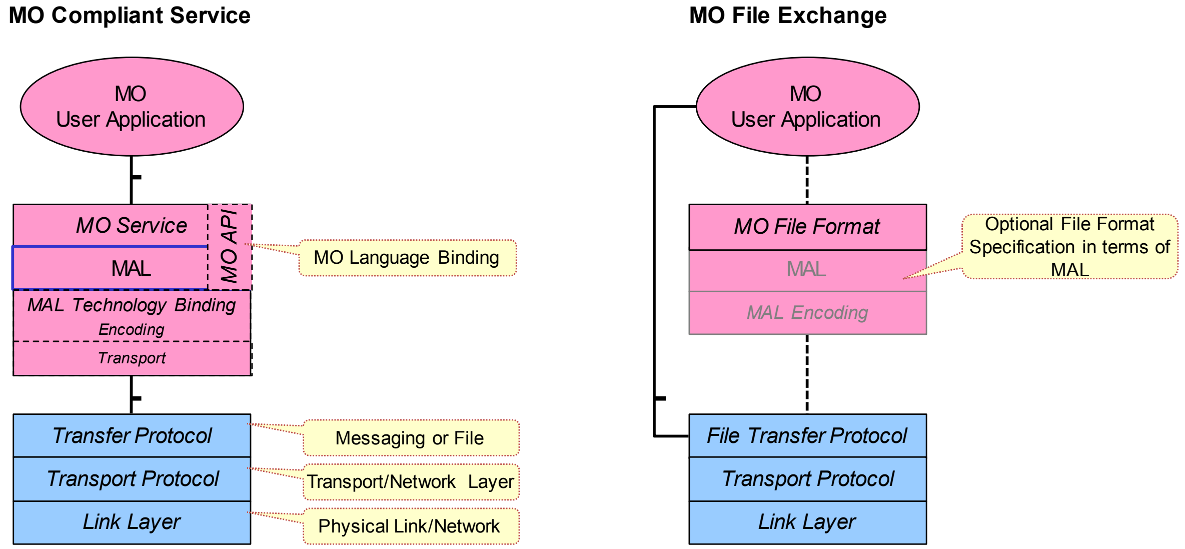


Figure 7-5: MO Services Generic Protocol Stack

The blue layers correspond to those of the underlying communications architecture:

- ‘Transfer Protocol’ (Application sublayer) (e.g., FTP, HTTP, or ZeroMQ) providing message exchange or file transfer.

- ‘Transport Protocol’ (typically the Transport and Network Layers combined (e.g., TCP/IP) providing end-to-end reliable, complete, in-order delivery of Transfer Protocol PDUs, unreliable, best effort (UDP/IP), or DTN.

- ‘Link Layer’ (Physical link or ‘Network’ (e.g., RF Space Link, WAN, LAN, or WLAN) providing hop-by-hop delivery of Transport Protocol PDUs among intermediate nodes.

NOTE – Because each of these protocol layers define their own ‘on the wire’ formats, all parties to a data exchange must agree to use the same protocol stack or support protocol matching bridges as required for interoperability.

The pink layers correspond to those of the MO application-level services themselves. In the case of an MO-compliant service, these comprise:

- The MO service specification itself (one of many), which can be any service defined in terms of the underlying MO/MAL Service Framework.

- The *Mission Operations Message Abstraction Layer* (reference [15]) that provides a syntax for the representation of MO service operations and messages that is abstracted from the underlying communications transfer layers.
- The MAL Technology binding (one of several available) that defines how the MAL is mapped to the underlying communications transfer layers. There are two aspects of this which may be separate or combined in a single standard:
  - Encoding: how the MAL service messages are encoded within a transfer message;
  - Transport: how the encoded MAL transfer message is bound to those of the underlying transfer services.
- An extensible set of standard MAL Technology bindings is available, including:
  - *Mission Operations—MAL Space Packet Transport Binding and Binary Encoding* (reference [20]);
  - *Mission Operations—Message Abstraction Layer Binding to TCP/IP Transport and Split Binary Encoding* (reference [21]);
  - *Mission Operations—Message Abstraction Layer Binding to HTTP Transport and XML Encoding* (reference [22]);
  - MO MAL ZeroMQ Transport (ZMTP) Binding (reference [73]) that can be used with any encoding.

  NOTE – All of these transport protocols and encodings are intended to be able to be mixed, such that any encoding, such as the Binary encoding, can be used with another transport binding, such as HTTP.
- The MO language binding (one of several available), which defines how the MAL and any MO service expressed in terms of the MAL, is presented to the application as a language-specific API. An extensible set of standard MO Language API Bindings is available, including:
  - *Mission Operations Message Abstraction Layer—JAVA API* (reference [18]);
  - *Mission Operations Message Abstraction Layer—C++ API* (reference [19]).

The use of a compliant technology binding allows interoperability among systems components based on the same bindings, and potentially with other bindings, by use of a bridge. Use of an API does not, in and of itself, carry such guarantees if the underlying implementation of the API does not also adopt compliant and interoperable technology bindings. APIs are useful for application portability, but do not, in and of themselves, provide interoperability.

For MO File Formats, in which only the message format is defined, a specific file transfer protocol is not imposed. The communicating applications are expected to make use of an existing standard (or bespoke) protocol to effect the file transfer. Message formats may optionally be defined in terms of the MAL, which brings the benefit that the different message encodings available through the MAL Technology bindings can be used.

The following diagram provides a generic deployment example for two MO user applications communicating using some MAL-compliant service.

Protocol stacks at both Communications and MO application levels must match between communicating user applications. The MO language binding can, however, differ between communicating user applications. In all of the following examples, the assumption is made that the APIs under discussion are compliant with both the API spec and a suitable MAL technology binding.

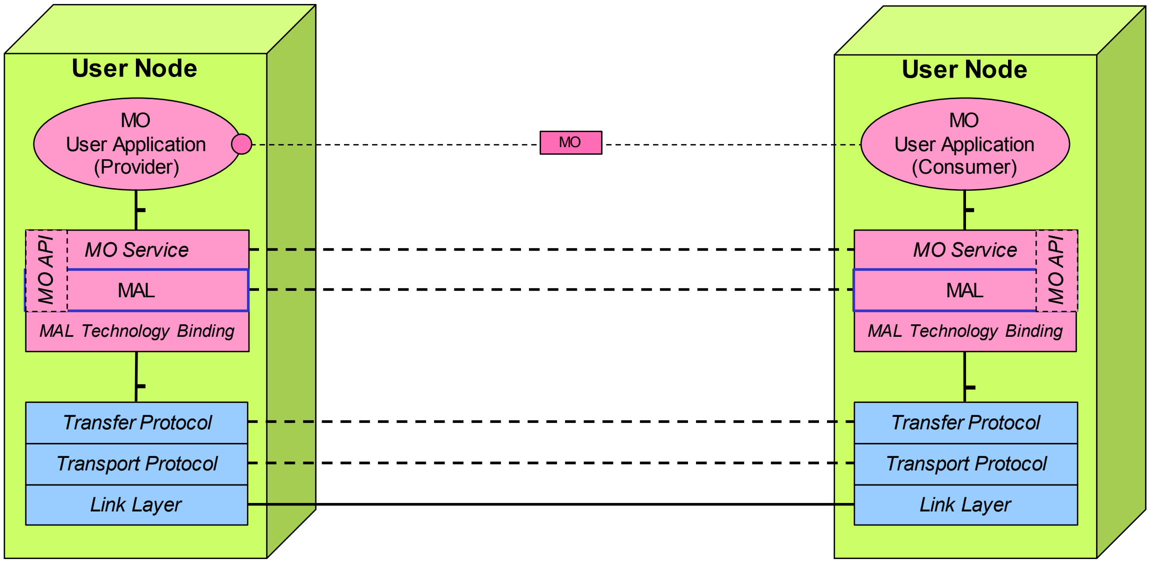


Figure 7-6: MO service Generic Service Deployment Example

The following diagram shows the same deployment case for a specific MO service: the MO Monitoring & Control service. It should be noted that this is still a generic deployment diagram. In an actual deployment, a specific MAL Technology binding (Encoding and Transport) must be selected and used by both communicating applications, but the MO language binding can be selected according to the programming language used for each application.

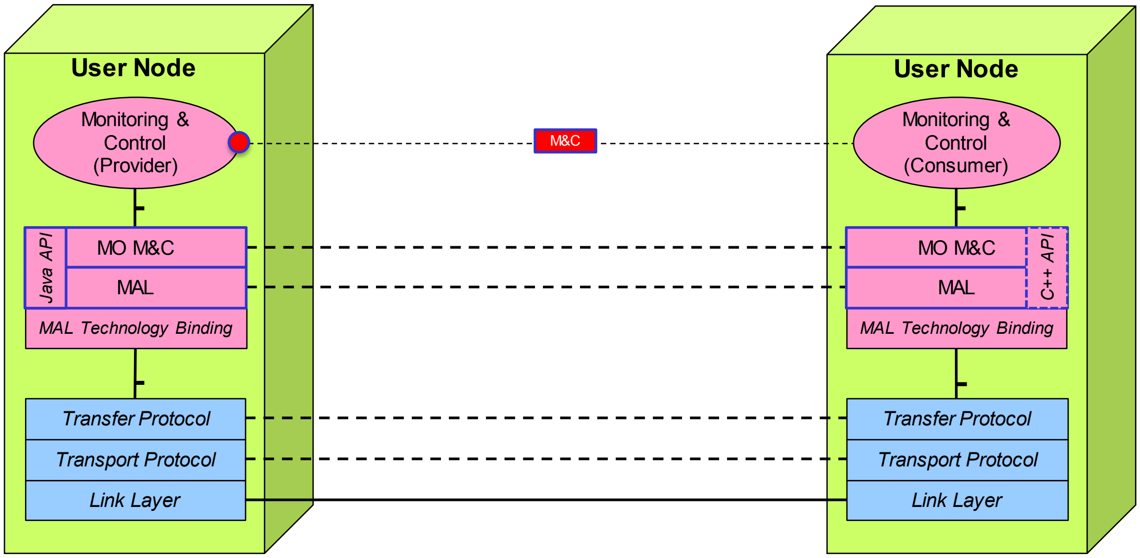


Figure 7-7: MO Specific Service Deployment Example

The following three subsections elaborate the specific protocol stacks for each of the identified communications deployment contexts.

7.3.4 SPACE LINK CONTEXT

In the Space Link Context, the communications transfer layer is assumed to be either CCSDS Space Packets or the CCSDS CFDP File Transfer protocol carried within CCSDS Space Packets.

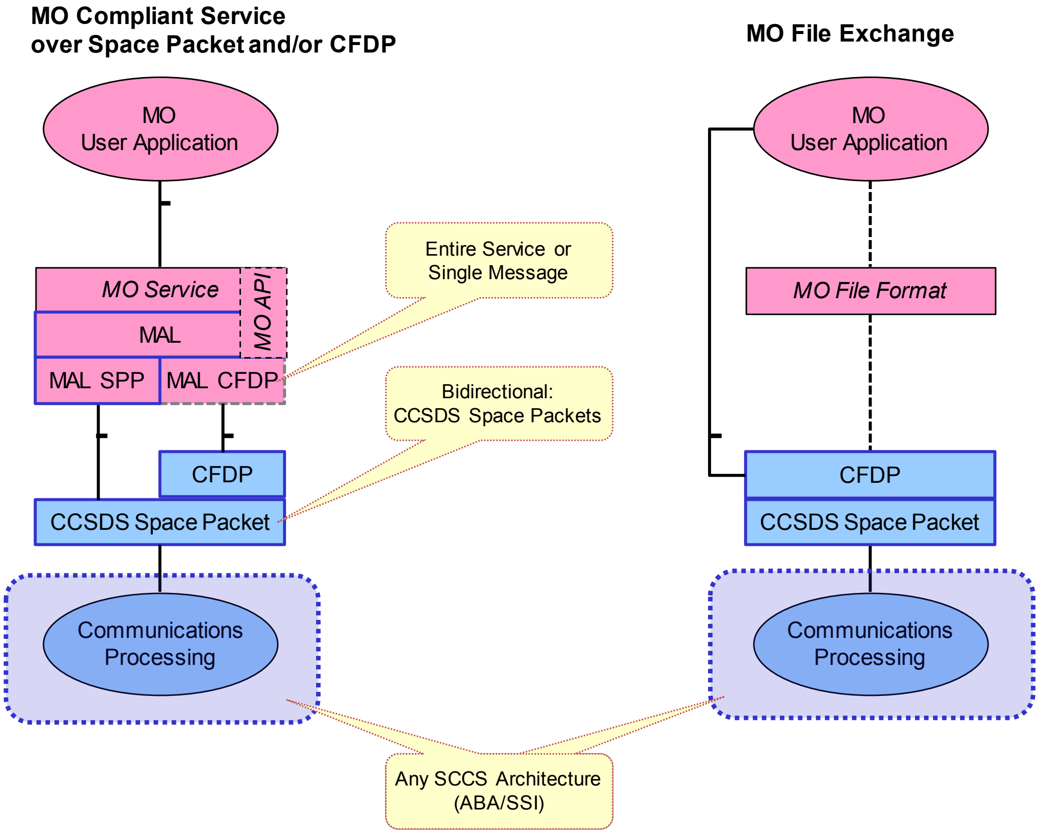


Figure 7-8: MO Services Space Link Protocol Stack

The full communications-layer protocol stacks, and the use of SLE and CSTS for accessing the ground-station services, are defined in the CCSDS SCCS-ARD for both the ABA and SSI communications architecture cases.

In this instance the MAL Technology binding is one or a combination of MAL to Space Packet (reference [20]) and/or MAL to CFDP. The is no MAL to CFDP binding currently available, although existing encodings may be used to generate files). The option of using file transfer is suggested for larger MO service messages, or for the bulk transfer of many smaller MO service messages (e.g., recorded onboard the spacecraft), but this has not yet been the subject of standardization.

For MO File Exchange, CFDP may be used across the Space Link, but the CFDP file transfer must be initiated directly by some onboard application.

The following diagram shows an example deployment of any MO service across a Space Link that follows the defined ABA deployment case, comprising a Space User Node (SUN), TT&C Ground Station, and Earth User Node (EUN). In this case, the MO service provider is onboard the spacecraft, and the MO Service Consumer is within the EUN. MO service messages are encoded within Space Packets and transferred over the extended Space Link. The application-level software does not need to be aware of how the encoding or transfer in terms of Space Packets is achieved; at this level, the exchange is in terms of MO service messages. The actual encoding and transfer is specified through the selected MAL Technology binding and underlying communications protocol stack.

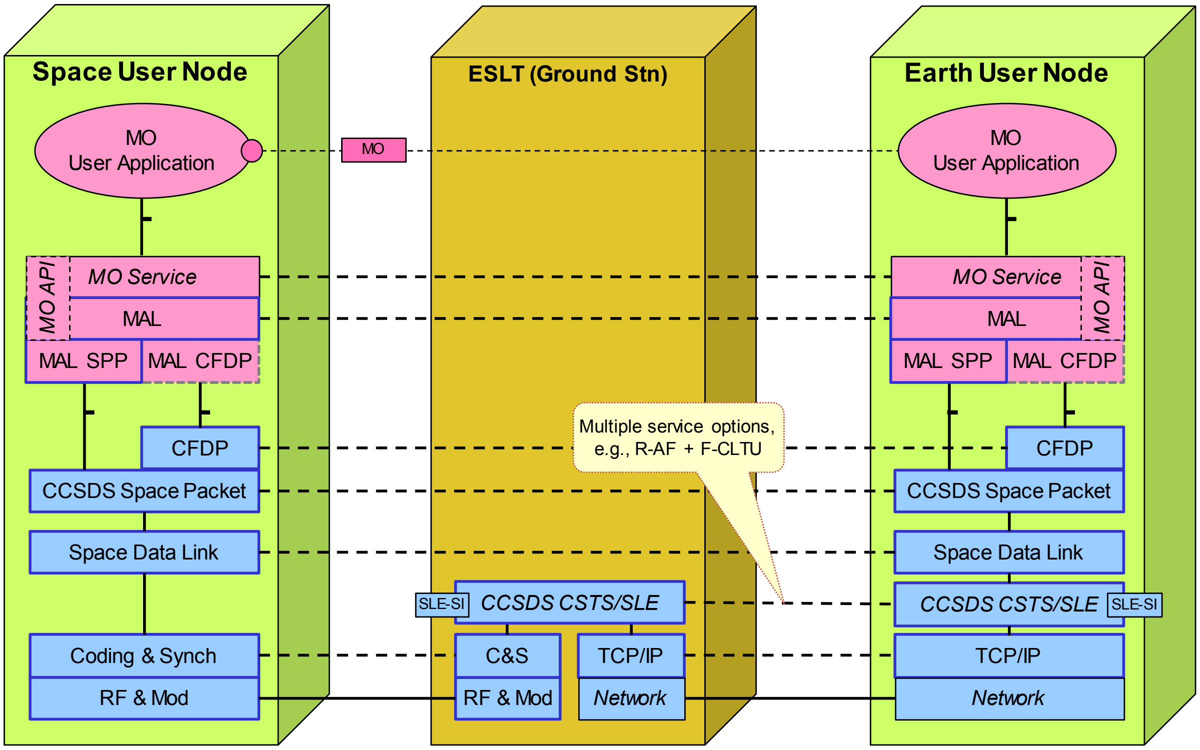


Figure 7-9: MO Services Deployment over Space Link (ABA Example)

This example extends the Space Data Link from the SUN to the EUN through the use of CCSDS CSS, specifically the SLE and CSTS. These cross-support services provide interoperable access to the physical space link (coding and synchronization, RF and modulation layers) between the spacecraft and ground station, and TCP/IP and a terrestrial link between the Ground Station and the EUN. SLE and CSTS offer multiple service options for both forward and return links. A typical deployment case is shown using the SLE RAF and FCLTU services. The link between Ground Station and EUN is established according to a previously negotiated Service Agreement (SA). The scheduling and configuration of the link are managed through the use of Service Management (SM) standards.

7.3.5 GROUND CONTEXT

In the Ground Context, the underlying communications layers are assumed to be based on TCP/IP, but a range of options are available for the Transport Layer. Figure 7-10 illustrates an example protocol stack using HTTP and FTP, but other transfer protocols may be used, providing the corresponding MAL technology binding is available.

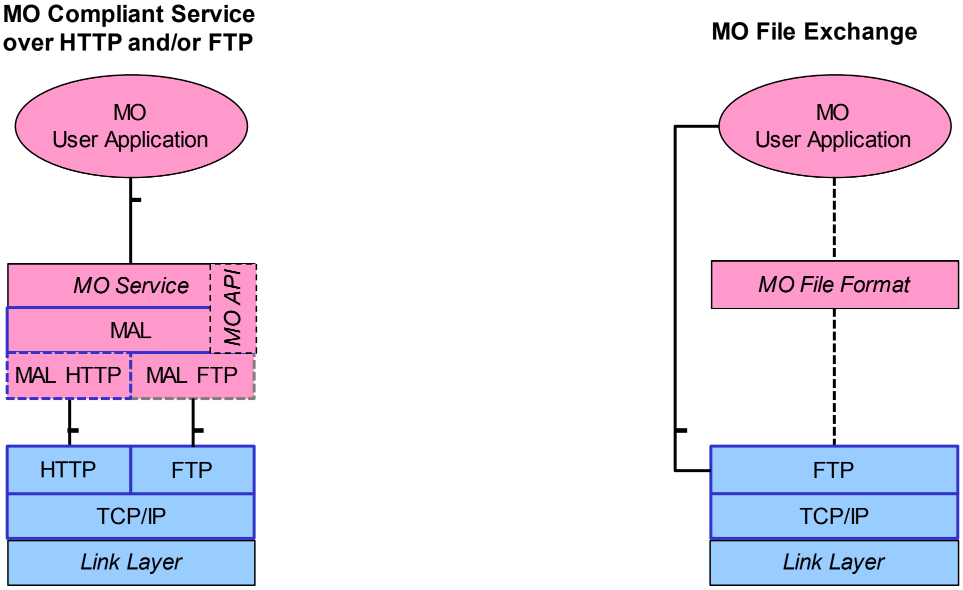


Figure 7-10: Example MO Services Terrestrial Link Protocol Stack

For message exchange, technology bindings are currently available for both HTTP and ZeroMQ. CCSDS may develop standard bindings to other messaging technologies in the future, as demand and resources permit. This approach of separating the abstract services from the underlying message encodings and transfer protocols provides a degree of future-proofing for MAL-compliant applications. It is also possible to use a proprietary or bespoke protocol, but only if a bespoke technology binding is developed and the same binding is deployed on both sides of the interface.

As for the Space Link context, there is also the potential for using file transfer for bulk message exchange, but there is currently no standard MAL-FTP technology binding to support this.

For MO file exchange, no specific file transfer protocol is imposed. FTP, CFDP, other standards, or bespoke protocols may be used by the communicating applications (or a third party) to effect the file transfer.

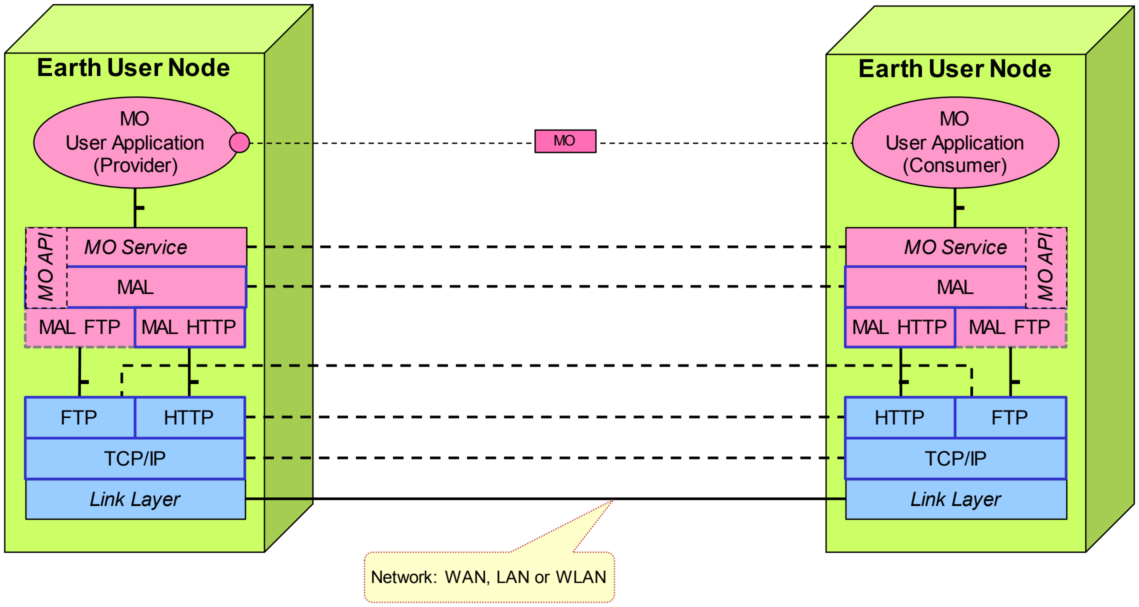


Figure 7-11: Example MO Services Deployment over Terrestrial Link

7.3.6 ONBOARD CONTEXT

Where MO functions are deployed in an onboard context they will require an exposed File Access Service (FAS) to interact with onboard file systems, and a Messaging Service to support interaction with other onboard applications.

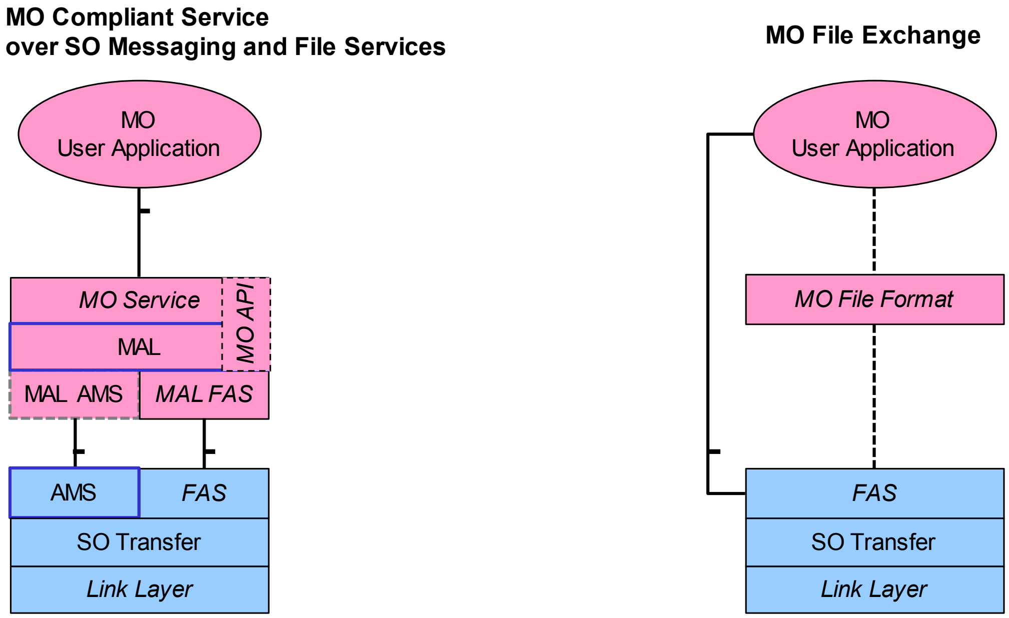


Figure 7-12: Example MO Services Onboard Link Protocol Stack

Although Message Transfer, Packet Store, and File Services, are identified there are not presently specifications for such standardized application support services. In practice, such services are specific to the onboard software architecture implemented for a given mission, which may be based on a manufacturer's proprietary infrastructure. A bespoke MAL Technology binding is therefore required for each onboard infrastructure solution.

AMS may be used to provide a standardized messaging service to onboard applications, but other/bespoke services may also be used, requiring a dedicated MAL transport binding. The AMS example is illustrated in figure 7-13.

If there is a distinct onboard Packet Store (rather than storing packets in files), then an additional binding may be required between the MAL and the Packet Store Service. This packet store service is often associated with access to the Space Link from an application perspective.

For File Exchange, the transfer must be independently initiated at Communications Protocol level, either directly by the User Application or by a third party.

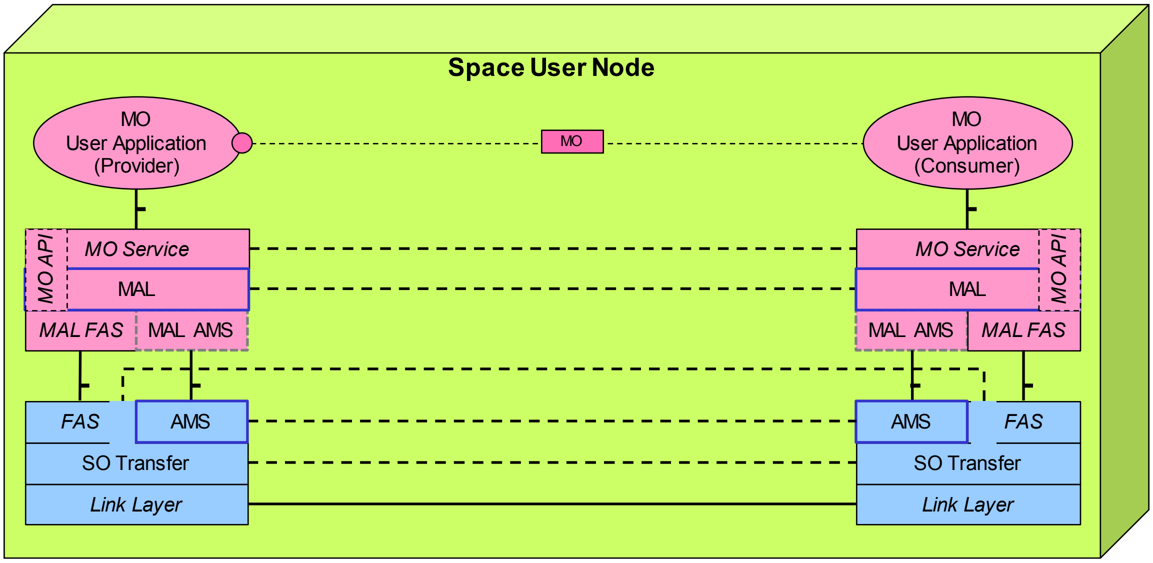


Figure 7-13: Example MO Services Deployment over Onboard Link

7.3.7 CONTEXT BRIDGING

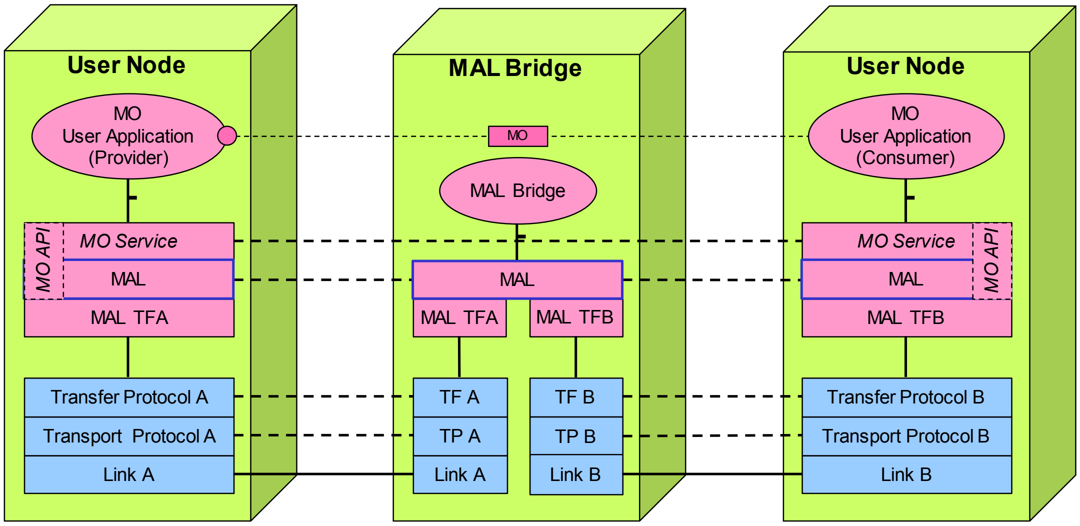


Figure 7-14: MAL Bridging Concept

Since MO services are specified in terms of the MAL, it is possible to implement a bridge between two separate communications contexts. If the MAL Bridge supports two technology bindings and translates between them, this can enable all MO services to be carried across the boundary transparently to the communicating applications.

This is illustrated in figure 7-14 above, which shows an example case for two different messaging technologies (Transfer Protocols A and B). All MO services defined in terms of the MAL can be transferred across the same generic MAL Bridge as long as the selected technology bindings are fully interoperable.

This bridging approach can also be used to link different communications contexts, such as Space-Ground and Ground-Ground, enabling end-to-end application-level communication across a heterogeneous network.

7.4 SPECIFIC PROTOCOLS FOR SO SERVICE INTERFACE BINDING

7.4.1 GENERAL

SO standards are about collection and distribution of information among onboard endpoints, which are in the Application Layer. The endpoints are connected by one or more networks belonging to a vehicle, called ‘subnetworks’. In the case of wireless subnetworks or tethered deployments, the subnetwork may extend a short distance outside the vehicle. The endpoints are typically devices, their software proxies in onboard computers, or other software components in onboard computers.

– For the onboard sensors and actuators, SO protocols are simple, and exploit the protocols provided by the subnetwork.

- When the onboard devices are gateways to other networks, an SO service may optionally have some transport and networking functions as needed to represent the endpoints in the other networks.

The description of the specific protocols for SO services in the remainder of this section is structured as follows:

a) simple subnetwork;

b) software message bus;

c) multiple processors separating application and instrument;

d) multiple processors separating applications;

e) connection to external network.

7.4.2 SIMPLE SUBNETWORK

Figure 7-15 shows a connection between an onboard device and an application, which is one of the basic purposes of SO services and protocols. The application executes in an onboard computer, which is connected to an onboard subnetwork. The device is connected to the same subnetwork. The EDS for the device describes the interfaces presented by the Device Service on behalf of the device. Conversion between D-PDUs (in the syntax of the device) and A-PDUs (in the syntax of the application) occurs in the device service, which consists of a DSAP and an optional DACP. The effect is as if a PDU flowed between the application and the device, with virtual translation.

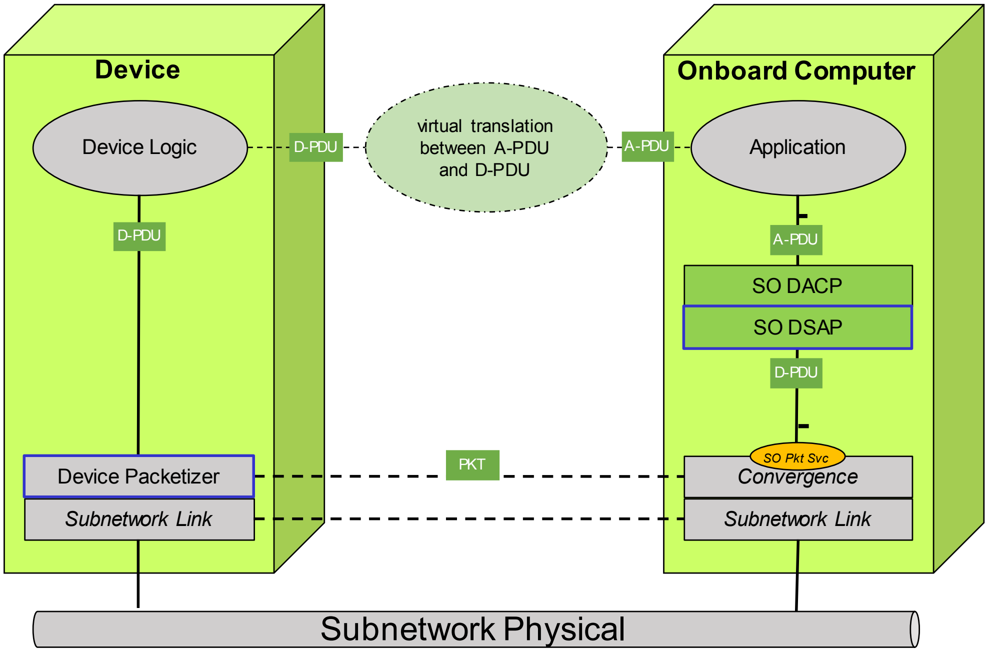


Figure 7-15: Connecting a Device to an Application

NOTES

1 The device is shown communicating through the SO packet service, but depending on the device, memory access could also be used.

2 The Device Packetizer is shown at the same protocol level as the SO Packet Service and the Convergence Functions. This is because the SO Packet Service has no discrete protocol functions, its function is to provide a defined service access point for the convergence layer functions.

3 The SO Wireless Green Book (reference [53]), subsection 5.2.1, considers the location of QoS provisions in the SO architecture. In figure 7-15, the one-sided SO Convergence functions would provide QoS.

The basic pattern of flow of messages in this example resembles a tree, in which the leaves are the devices, and the trunk is the command and data handling application function at the onboard computer. The messages flow between leaves and trunk; they do not flow between leaves.

7.4.3 SOFTWARE MESSAGE BUS

Figure 7-16 shows a connection between an application and an onboard device that offers some flexibility for reuse of both the device and the application in other contexts. The software bus is shown here as a service in the application support layer, such as AMS. The adapter for the device will be generated from the SEDS for the device. Conversion between D-PDUs (in the syntax of the device) and A-PDUs (in the syntax of the application) occurs in the device service, which consists of a DSAP and an optional DACP. The adapter for the application can be generated from the SEDS for the application. The effect is as if an A-PDU flowed between the application and the device, with virtual translation.

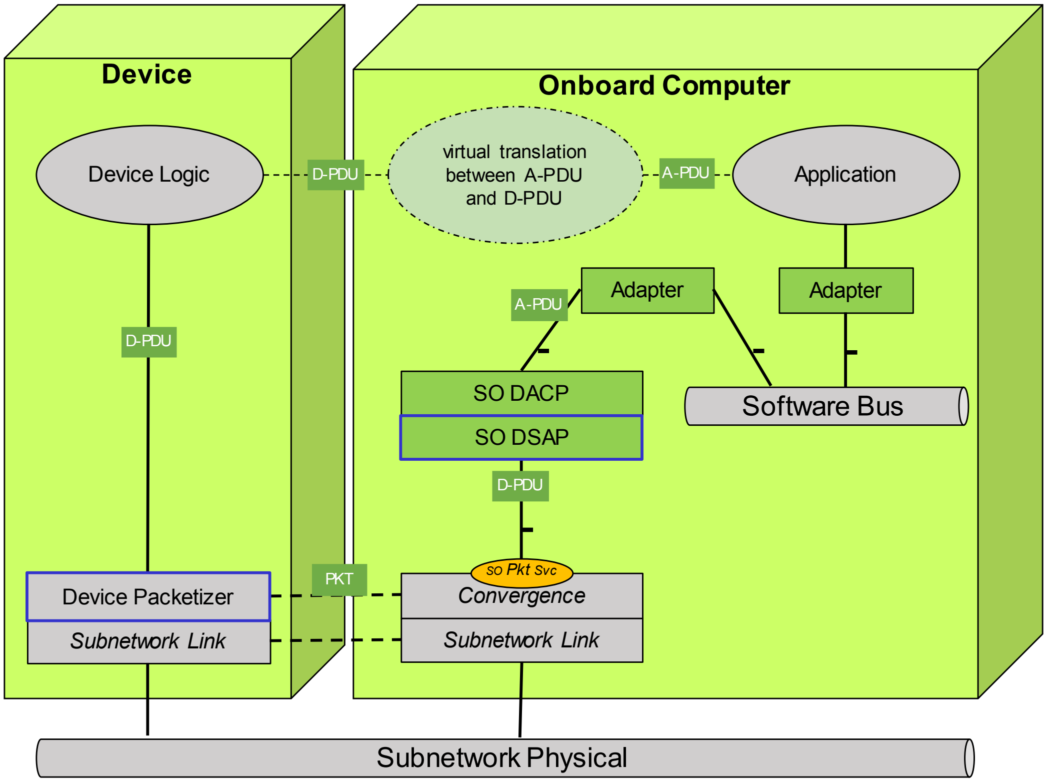


Figure 7-16: Connection between Onboard Device and Application through Software Bus

NOTE – The device is shown communicating through the SO packet service, but memory access could be used, depending on the device.

Although a software bus could provide arbitrary connectivity between endpoints, the pattern of flow of messages in this example is tree-like, as described in 7.4.2.

7.4.4 MULTIPLE PROCESSORS SEPARATING APPLICATION AND INSTRUMENT

Figure 7-17 shows a connection between an application and an onboard device, in which the device connects to one processor, and the application is on a different processor connected through a subnetwork. The processor containing the EDS-Derived Device Access Service might be any of the following:

- a physical adapter for the device;
- a remote interface unit;
- a processor peer of the processor containing the application.

The 'Software Bus' is optional, providing publish/subscribe capability as an alternative to pre-planned routing. The internetworking Layers (Transport and Network) in the diagram represent non-CCSDS protocol standards, such as TCP/IP, which support addressing and end-to-end connections across multiple, possibly different, subnetworks.

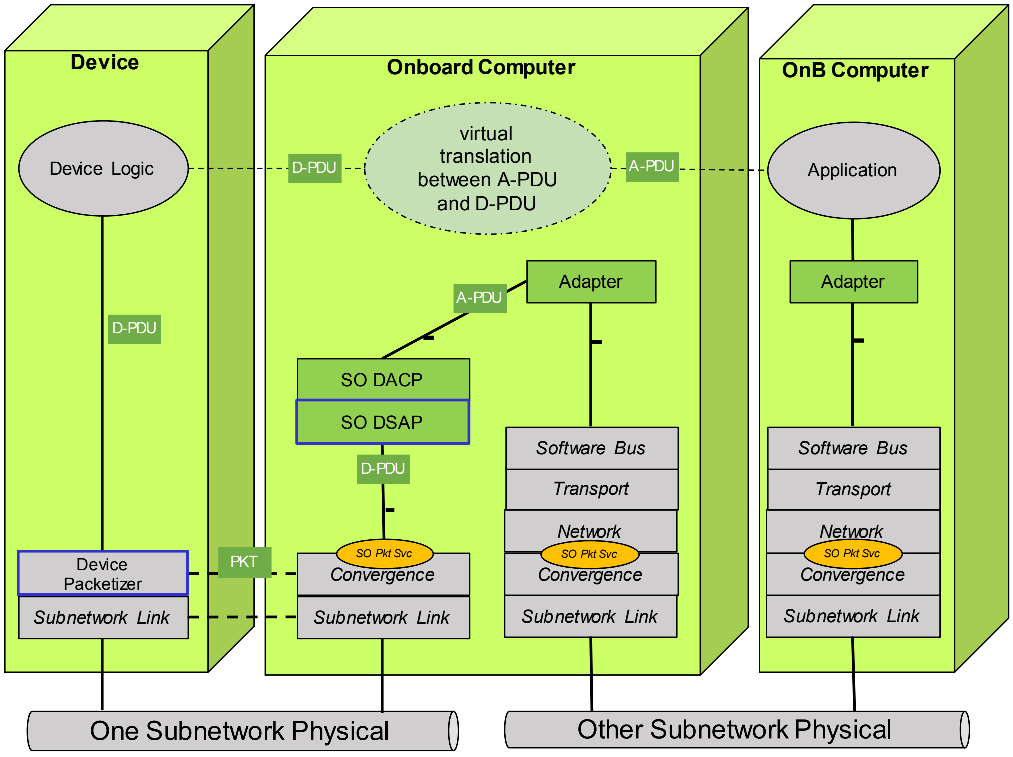


Figure 7-17: Connection Between Onboard Device and an Application through a Subnetwork

NOTE – The device is shown communicating through the SO packet service, but memory access could be used, depending on the device.

Although the software bus and the Transport and Network Layers could provide arbitrary connectivity between endpoints, the pattern of flow of messages in this example is tree-like, as described in 7.4.2.

7.4.5 MULTIPLE PROCESSORS SEPARATING APPLICATIONS

Figure 7-18 shows a connection between two applications on different onboard processors. The processors containing the applications might be any of the following:

- time/space partitions (the software bus could then be an inter-partition communication channel, and the internetworking layers could be unnecessary);
- peer processors.

The 'Software Bus' is optional, providing publish/subscribe capability as an alternative to pre-planned routing. The internetworking Layers (Transport and Network) in the diagram represent protocol standards outside of CCSDS, such as TCP/IP, which support addressing and end-to-end connections on the subnetwork that provides the hardware message bus.

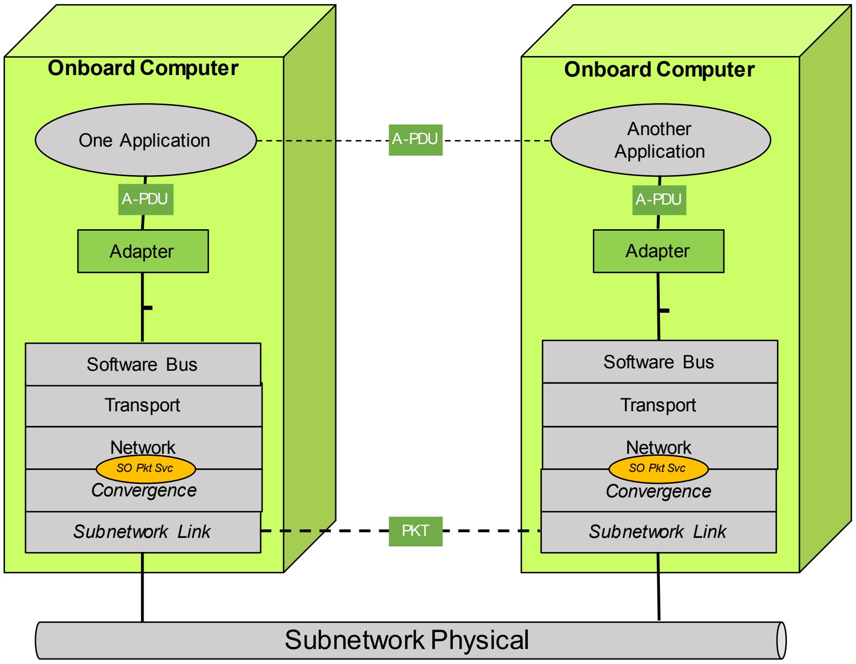


Figure 7-18: Connection between Applications on Separate Onboard Processors through Subnetwork

The pattern of flow of messages in this example differ from the tree-like pattern in the preceding peer sections. In this case, the flow of messages between application endpoints forms an arbitrary graph, supported by the software bus and the Transport and Network Layers.

7.4.6 CONNECTION TO EXTERNAL NETWORK

In this example, SO services are deployed at one end of a space data link, which is one ‘A’ leg of an ABA configuration.

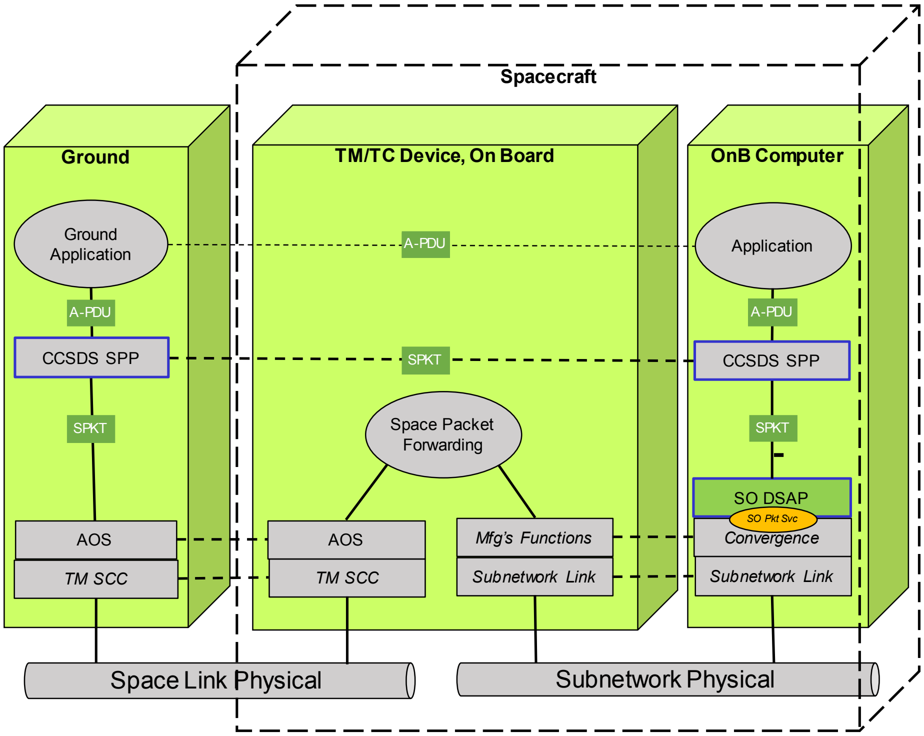


Figure 7-19: SO Configuration for Telemetry and Telecommands

In figure 7-19, the vehicle’s antenna and radio (the TM/TC device) connect through a Space Packet forwarding function and SO protocols to onboard applications. The interfaces for the TM/TC radio and antenna(s) and the OnBoard Computer are described by SEDS. The SEDS descriptions are used at design time to configure data paths, including generation of SO device service (DSAP) interface, but during the mission, the SEDS are no longer involved in the movement of data. (See 10.3 for the design-time function of SEDS.) SO device services perform two tasks for radio interfaces: (1) They provide command and data handling for the radio as a device, and (2) they pass TM/TC packets to and from the data link for the spacecraft as a whole. The latter task may use a separate hardware interface that completely bypasses SO layers.

Figure 7-20 shows a similar diagram to figure 7-19 but is an SSI configuration using DTN networking protocols and SO protocols to connect onboard and terrestrial elements. Connection to the distributed elements in the Solar System Internet is through one or more high-capacity channels, such as an antenna and radio or an optical emitter and detector, as shown below. The same kinds of SO device access and subnet services defined in earlier examples would also be deployed in these cases, as desired.

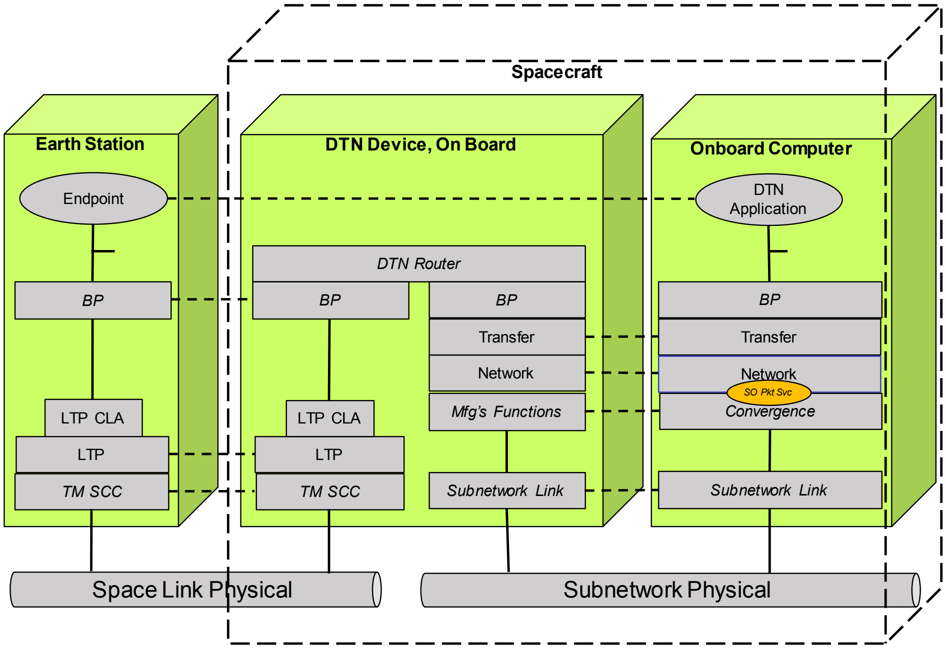


Figure 7-20: Solar System Internet

7.5 REMAINING CHALLENGES TO PROTOCOL DEPLOYMENT FOR SO AND MO

The MO suite of technology bindings is already quite broad, and is likely to grow over time. This will especially be the case if MO is deployed in onboard environments. There it will be necessary to provide very efficient mappings from MAL abstract messages to the typical resource limited spacecraft environment. Furthermore, some of the MO mappings require what are characterized as ‘managed parameters’, in which not all of the data present in a given message, or message header, get translated to the new technology binding. These managed parameters, and any efficient mappings must be carefully tested to demonstrate interoperability in the nominal case.

For SO services there is a substantial set of possible onboard subnets that may be adopted. There are not yet any concrete mappings from these various subnets to any sort of ‘subnet convergence layer’, nor of the SO Packet Service to any sort of convergence layer. This is all [Future] ‘work to go’. In a similar vein, the DACP and DSAP functions are only defined in the abstract at this point, and implementations will be highly context dependent and provide portability of devices and services into diverse contexts powered by tool chain generation of software artifacts. If the same device or onboard application is used in multiple spacecraft, the same EDS describing that device would also be used in all those systems.

An important function of the SEDS in this context is to provide a description of the ‘glue’ needed to hook up such components to whatever onboard operating system and application framework might exist in a given deployment. The interpretation of SEDS instances by a tool chain is viewed as the point at which interoperability and portability are generated in the context of an agency-specific flight platform.

7.6 SECURITY CONCEPTS FOR PROTOCOL VIEWPOINT

7.6.1 GENERAL

Security for the Protocol Viewpoint may be applied at the Physical Layer, Data Link Layer, Network Layer, or Application Layer. Table 7-1 identifies the current security standards defined for each layer.

Table 7-1: Security Standards

ISO Layer	Security Standards	Reference
Physical Layer	No CCSDS Recommended Standards defined; various methods in local use	N/A
Link Layer— Space-to-Ground Links	Space Data Link Security Protocol	CCSDS 355.0-B-1, dated Sept 2015
	CCSDS Cryptographic Algorithms	CCSDS 352.0-B-2 dated Aug 2019 (reference [39])
	Symmetric Key Management	CCSDS Green Book CCSDS 350.6-G-1, dated Nov 2011
	Space Data Link Security Protocols—Extended Procedures	CCSDS Red Book CCSDS 355.1-R-1, dated Apr 2018
Network Layer—DTN Internetworking Protocols	Bundle Protocol Specification	Internet RFC 5050 (also CCSDS 734.2-B-1, dated Sept 2015)
	Bundle Security Protocol Specification	Internet RFC 6257
	Network Layer Security Adaptation Profile	CCSDS 356.0-B-1 dated Jun 2018
Network Layer—IP Internetworking Protocols	Security Architecture for the Internet Protocol	Internet RFC 4301
	IP Encapsulating Security Payload	Internet RFC 4303
Application Layer	HTTP Over TLS	Internet RFC 2818
	Hypertext Transfer Protocol—HTTP/1.1	Internet RFC 2068
	FTP Security Extensions	Internet RFC 2228
	File Transfer Protocol (FTP)	Internet RFC 959
	CCSDS Cryptographic Algorithms	CCSDS 352.0-B-2 dated Aug 2019 (reference [39])
	CCSDS Authentication Credentials	CCSDS 357.0-B-1 dated July 2019

7.6.2 SECURING CONNECTIONS AMONG SERVICE USERS & PROVIDERS

As was noted in the Services Viewpoint, there are not a lot of security features included in the existing SO and MO services, nor functions. That said, it is possible to employ existing CCSDS Data Link Layer, Network Layer, and/or Application Layer security mechanisms such as those featured in table 7-1. Link-layer communications, from the MOS user to the spacecraft, may be secured using the Space Data Link Security protocol extension and any of the CCSDS Space Data Links. These provide for authentication, encryption, or authenticated encryption. For ABA missions, this will provide a high level of command and even telemetry security. It is vulnerable to traffic analysis, but this is not usually an issue with civilian space missions.

For SSI/network deployments, DTN and the DTN bundle security protocol may be employed. This provides secure end-to-end networked communications. For missions that are close to Earth, up to GEO distances, missions may choose to adopt TCP/IP and the related IP Protocol Suite services. These are not suitable for deep space, but work fine close to Earth. The SCCS-ARD (reference [2]) provides more information on link and Network Layer security deployments.

For secure access to terrestrial services, the secure HTTPS or Secure FTP Internet protocols may be employed. This secures the interfaces between remote users and the terrestrial interfaces of service-providing systems, using encryption protocols. Service interfaces that offer web services or file transfer functionality should secured. They may also implement stronger security in the form of HTTPS or Secured File Transfer Protocol (SFTP), encrypted interfaces that both authenticate the user and shield the traffic by sending it in an encrypted ‘tunnel’.

These approaches do require the host service-providing systems to invest in security infrastructure that they may not otherwise adopt.

For all of the terrestrial service interfaces, access control and authentication should be applied, usually during connection establishment or binding to the service. The SLE, CSTS, and service management service interfaces include access control mechanisms that will require user authentication and management of port assignments. These may use one of the types of credentials described in the cryptographic algorithms document (reference [39]) or other methods, as required by the service provider. Access to space service interfaces may also be secured, as noted above, but this is not the norm.

8 PHYSICAL (CONNECTIVITY) VIEWPOINT

8.1 OVERVIEW

The deployment cases that appear in this section are just examples that were selected to illustrate a wide enough set of cases in which interfaces may be exposed to an interoperability boundary. Generic examples have intentionally been used, and they may bear no relationship to any existing or planned deployments.

8.2 MO PHYSICAL ELEMENTS

8.2.1 GENERAL

MO functions may be distributed over a wide network of Earth (and space) User Nodes, and as a result, MO services and MO file exchange may be deployed across the boundaries between these distributed User Nodes, supported by communications links that fall into one of three communications contexts, as identified in the Communications Viewpoint (space link, ground, and onboard).

There is no definitive set of standard names or User Nodes. Each space mission system has its own deployment architecture and identifies its own set of physical nodes, each of which are named in that mission context and may be owned or operated by a different agency or organization. The remainder of this section provides some illustrative, but realistic, examples of potential physical deployment architectures that can be used to identify where interfaces between MO functions could be exposed to an interoperability boundary. Node types are defined in an attempt to be representative, and generally meaningful, but not in any way prescriptive.

It is structured as follows:

a) identification of deployment nodes;

b) physical deployment architecture examples;

c) potential functional deployment.

8.2.2 IDENTIFICATION OF DEPLOYMENT NODES

As indicated above, the SCCS-ADD only identified three classes of physical deployment nodes: SUN, EUN, and Earth-Space Link Terminal (ESLT). For the purposes of modelling distributed mission operations networks with multiple deployment nodes in both the Space and Ground segments, a representative set of deployment nodes is identified below. These deployment nodes are only examples, the names are intentionally generic, and actual space system physical architectures may identify and name other classes of deployment node.

– SUNs:

  - Spacecraft (Orbiter/Relay, Lander/Rover),
  - Habitat (Station, Base, or Suit),
  - Payload (or Instrument, hosted in a Spacecraft or Habitat);
- EUNs:
  - MOC,
  - POC or Rover Operations Center (ROC),
  - Navigation Services Center (NSC),
  - Data Processing Center (DPC),
  - Data Archive Center (DAC),
  - PI/user,
  - Spacecraft Manufacturer (SCM).

There may be multiple Spacecraft nodes within a given space system. This may be the case for an interplanetary mission that includes an Orbiter that also acts as a communications relay, and a Lander or Rover deployed to the surface. In the case of constellation missions, there may be a large number of spacecraft nodes.

Habitats are crewed space system nodes, which may range from a space station, a Lunar or Mars base, or an individual astronaut's suit.

Payload nodes are instruments or other equipment that is hosted onboard a Spacecraft (or Habitat), which is owned or operated by a different authority from the host.

Most space systems include a MOC that is responsible for mission operations functions, including: Mission Control, Mission Planning, Navigation, and Operations Preparation.

Many missions also have a dedicated POC for the operation of the payload instrument(s). These are usually more closely associated with mission scientific objectives and interact more closely with PIs and the user community. This is sometimes termed a Science Operations Center (SOC). In the case of a deployed surface rover (or other probe), there may also be a dedicated Rover Operations Center (ROC).

Some Navigation functions may be centralized at a dedicated NSC, which provides specialized support to multiple missions. This is commonly the case for functions such as Conjunction Assessment, which provides collision warnings, but many low-cost CubeSat missions also use centralized services to perform their Orbit Determination.

The processing of acquired mission data may be performed at a separate DPC. Although Mission Data Processing is currently not an MO service, Mission Operations functions frequently interact with such facilities.

In a similar way, the long-term archiving of mission or science data may be delegated to a dedicated DAC that stores both the raw data acquired and processed mission data products, possibly from multiple space missions associated with a specific science discipline.

The MOC, POC, DPC, and DAC may all be co-located at a single site or distributed in different configurations across multiple sites operated by different agencies, but they represent functional groups found in many space system architectures. Even when co-located, they typically correspond to distinct systems. There is also the potential for some of these (DPC and DAC) to be multi-mission facilities.

Principal Investigators, or users, correspond to external entities that are responsible for tasking and utilizing the space system. They may be academic teams responsible for a science payload, astronomers using an observatory mission, or the users of an Earth observation system. They are typically widely distributed and will interact with the space system primarily through Mission Planning and the distribution of mission data.

SCM**s** have a responsibility to provide initial information required to support Operations Preparation for the configuration of Mission Operations. This includes the provision of the spacecraft database (telemetry and telecommand definitions), operations procedures, and onboard software. Manufacturers often also have an on-going responsibility to support anomaly resolution and performance monitoring of the spacecraft or payload. This requires access to Mission Control data for investigative purposes.

Other EUNs could include functions required that are outside the normal scope of Mission Operations and are therefore not discussed further here. These include:

- pre-launch spacecraft Assembly, Integration, and Verification (AIV) (also known as checkout) facilities;
- launch facilities;
- spacecraft operations simulators;
- training facilities.

Such additional EUNs normally contain variations on the set of functions already identified in this document, potentially extended by specialized functions.

8.2.3 PHYSICAL DEPLOYMENT ARCHITECTURE EXAMPLES

8.2.3.1 Overview

This subsection provides examples of typical physical deployment architectures for space systems. The examples have been chosen to illustrate deployment of the different types of communications contexts (as identified in the Communications Viewpoint) for typical space systems. There is no intention to standardize the deployment architectures themselves, and many alternative physical deployment architectures are possible for actual space missions.

Each example shows a set of deployment nodes (based on the example set identified in the previous section) and the topology of the communications links between those nodes, using the notation defined in 3.3.6. Communications links are color coded to indicate the type of link in terms of the communications deployment context (space link, ground, or onboard).

The following example physical deployment architectures are provided:

– ABA space link;

– SSI;

– hosted payload.

Each example builds on its predecessor to introduce additional communications deployment contexts. The following points are noted:

– At application level, the ABA Space Link case is effectively a subset of the SSI case.

– The routing function is only indicated for SSI routing nodes. In practice, all ground deployment nodes will have some sort of routing function for the terrestrial network.

– Habitats and deployed rovers/landers may have similar network topology (from a perspective of some distance) and have been represented as a single deployment node.

– There may be multiple instances of all types of deployment nodes within a single system.

8.2.3.2 ABA Space Link Example

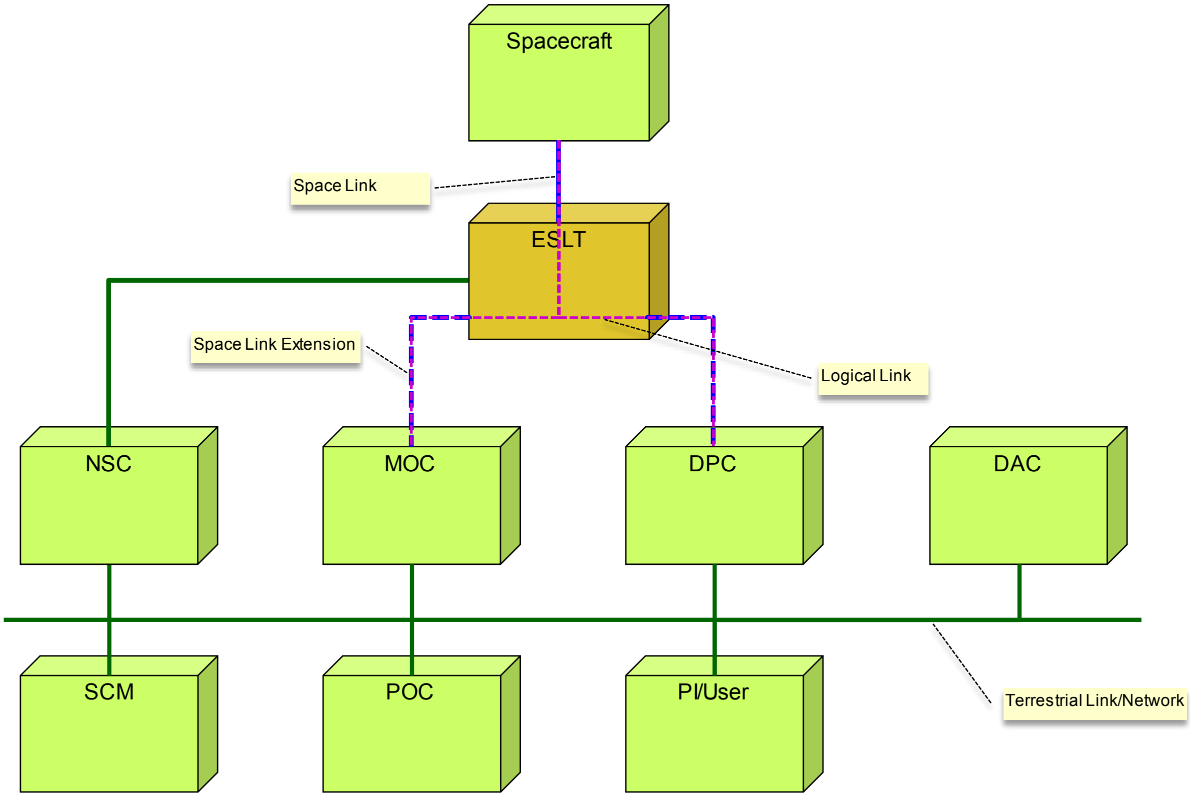


Figure 8-1: Physical Deployment Architecture: ABA Space Link Example

This example shows a relatively simple mission comprising a single Spacecraft deployment node, an ESLT (ground station, although this could equally be a network of ground stations), and a full set of example ground segment deployment nodes.

CCSDS CSTSes are used to extend the Space Link from the ESLT to both Mission Operations and Data Processing Centers. It is not unusual in Earth Observation and certain Relay missions to have separate communications channels for TT&C communications and mission data downlink. In some cases, separate ESLT ground stations are used for TT&C and mission data acquisition. This is not usually the case for space exploration missions.

Terrestrial networks are used to link the distributed physical deployment nodes of the ground segment. SLE and CSTS usually run as 'tunneled' services over terrestrial networks as well.

An NSC may use the CSTS Tracking Data Service or be linked through terrestrial networks to the ESLT ground stations for the acquisition of satellite tracking data.

8.2.3.3 SSI Example

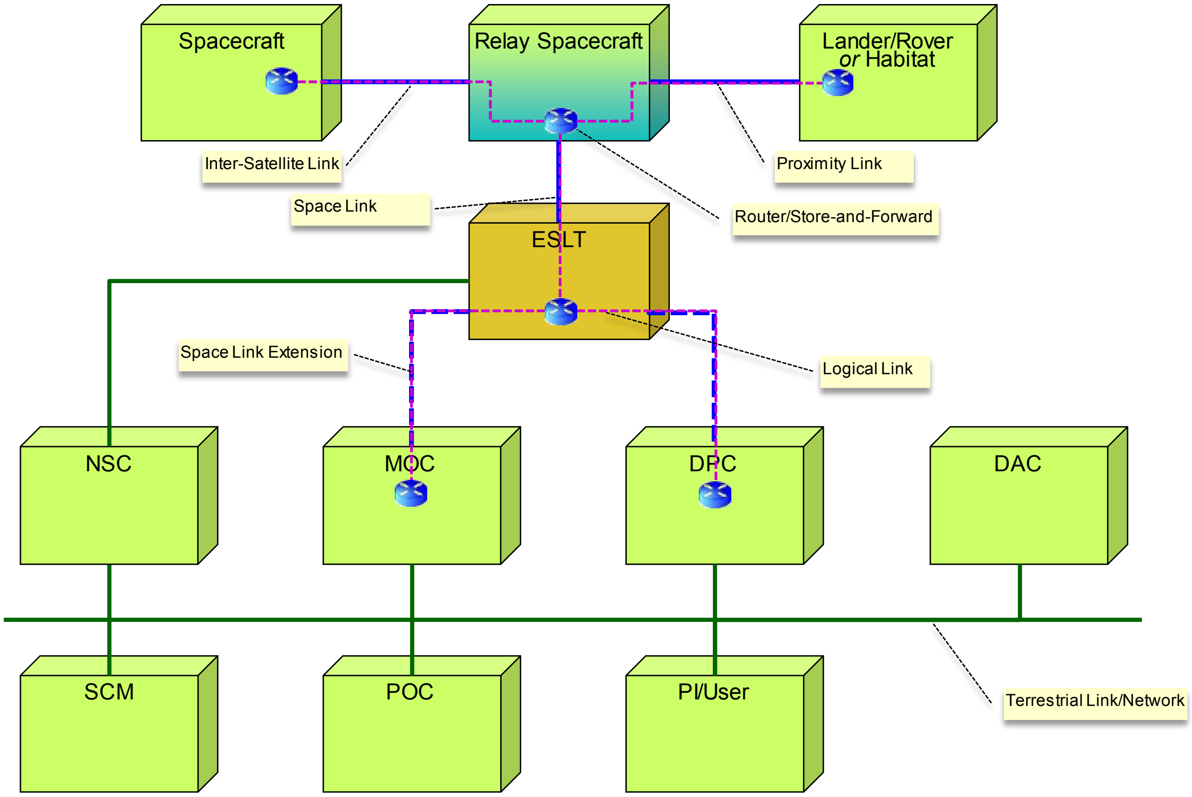


Figure 8-2: Physical Deployment Architecture: SSI Example

In this SSI example, the space segment comprises three separate deployment nodes. One Spacecraft is acting as a space routing node, with an inter-satellite link to a second Spacecraft node, and a proximity link to a surface Lander, Rover, or Habitat.

In the case of a space system using the SSI, all deployment nodes forming part of the SSI contain an SSI router function supporting data storage and forwarding. All deployment nodes on the terrestrial network will also include normal network routing capability, but this is omitted for clarity.

8.2.3.4 Hosted Payload Example

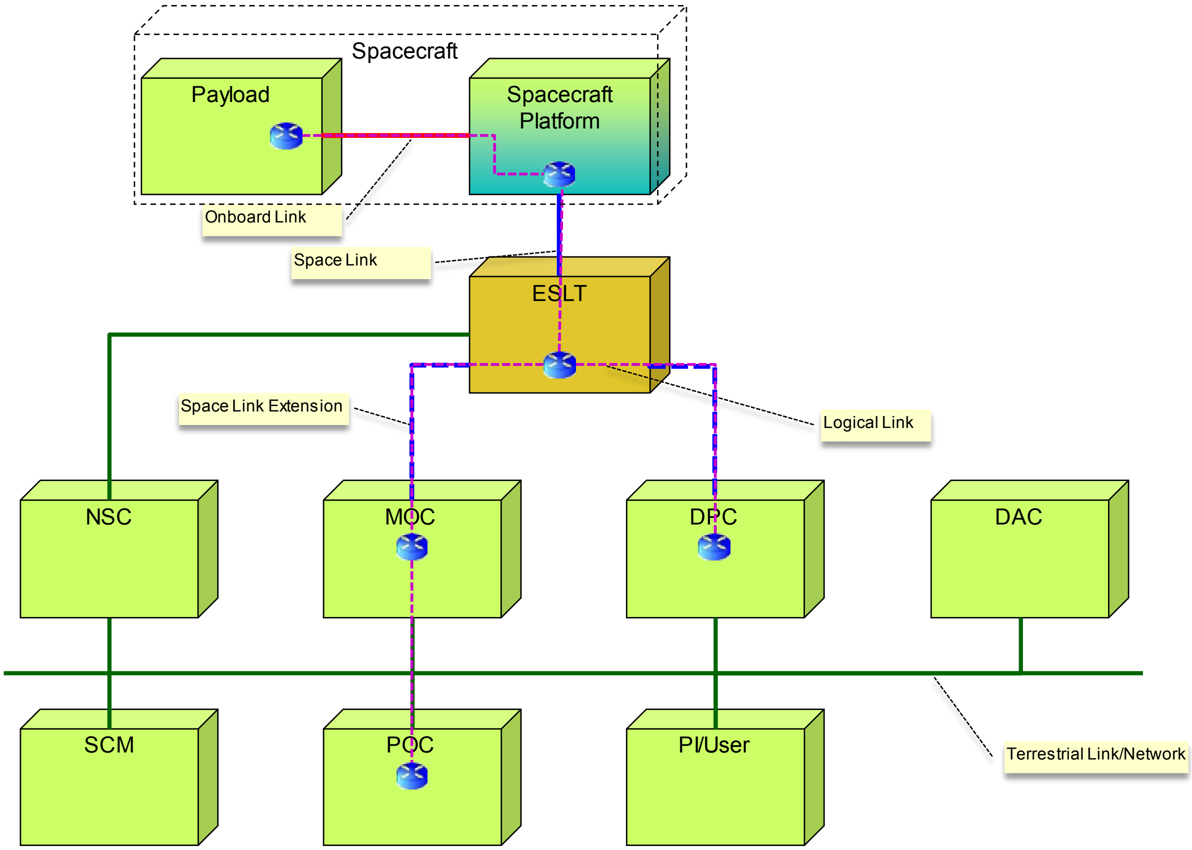


Figure 8-3: Physical Deployment Architecture: Hosted Payload Example

In this Hosted Payload example, the Spacecraft Platform and the hosted Payload are considered separate deployment nodes, each capable of hosting MO functions. Communications between them are via an onboard subnet and the distribution of application data using, for instance, DTN.

In the ground segment, the SSI has been extended across the terrestrial network from the MOC to the POC, enabling direct end-to-end communications between POC and the Payload it controls.

8.2.4 POTENTIAL MO FUNCTIONAL DEPLOYMENT

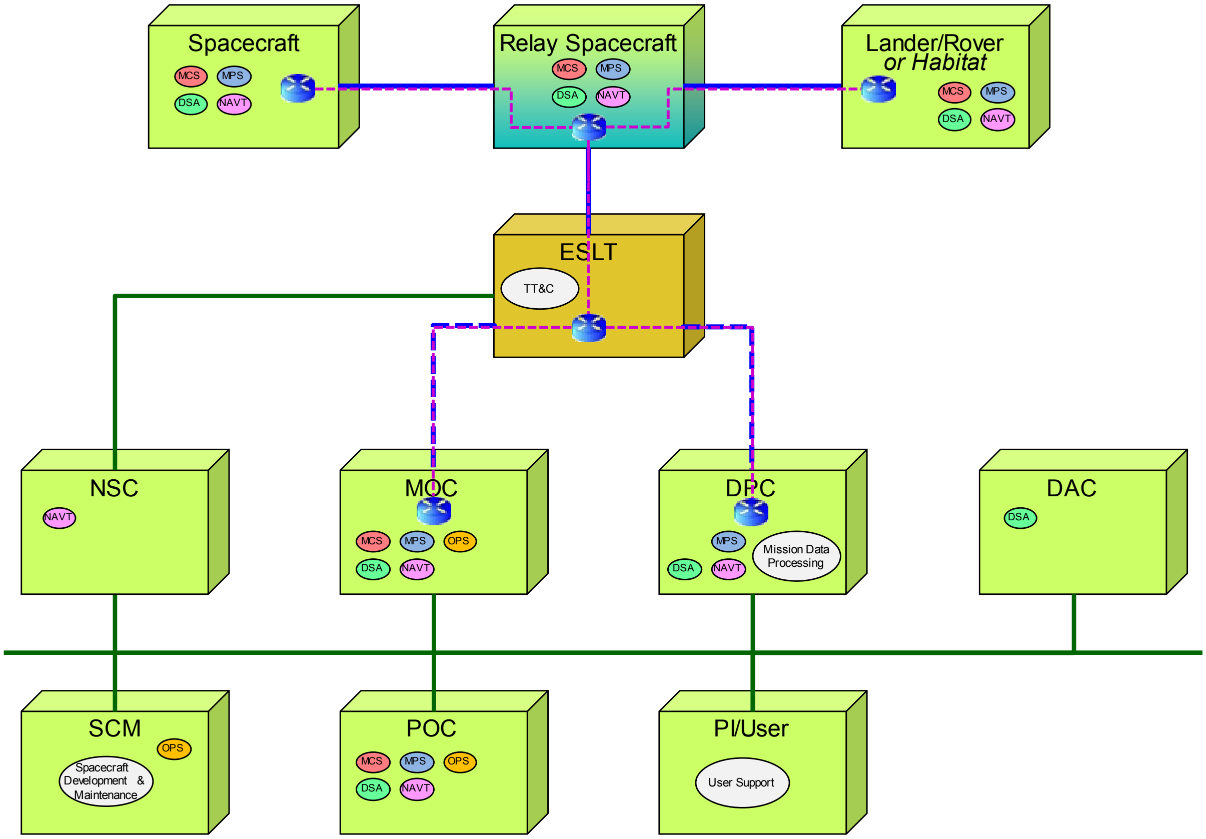


Figure 8-4: Potential Functional Deployment

The various physical deployment architectures presented in the previous section contain no information about the deployment of application-level MO functions. It is only when Mission Operations functions are placed within the deployment nodes that it becomes possible to identify which MO-level information exchanges are exposed to the potentially interoperable boundaries between those nodes. The representation of MO information exchanges at interoperable boundaries is the primary purpose of the Deployment Viewpoint contained in the next section.

Figure 8-4 gives a high-level summary of the potential deployment of functions, based on what might be a typical deployment of functions among the identified example classes of deployment nodes given in 8.2.2, above. The colored ellipses correspond to the top-level MO functional groups identified in the Functional Viewpoint.

All Space Segment nodes could potentially host a subset of the following Mission Operations functions: Mission Control, Mission Planning, Navigation and Timing, and Onboard Data Storage.

ESLT nodes host the external TT&C function including the SLE, CSTS, and Service Management functions required to actually plan for, configure, and exchange data over space links and to provide ancillary data such as monitor and radiometric data types.

The MOC and POC may host all MO functions applicable to their respective domain of interest.

The DPC primarily hosts the external Mission Data Processing function, but may also host local Data Archiving and a subset of Mission Planning and Navigation functions. The last of these is when spacecraft position data is generated as a by-product of image or other mission data processing.

The DAC hosts long-term Data Archiving functions.

A dedicated NSC hosts a subset of Navigation functions, potentially including Orbit Determination, Conjunction Assessment, and Re-entry Assessment.

PI/user deployment nodes host the external User Support function.

SCM nodes host the external Spacecraft Development and Maintenance function, and may also contribute to the Operations Preparation function.

8.3 SO PHYSICAL ELEMENTS

The present statement of scope is as follows:

> *The basic context of SO services is that of a single spacecraft within a single mission.*

As such, the SO physical elements are a just the space subset of those listed for MO:

- SUNs:
    - Spacecraft (Orbiter/Relay, Lander/Rover);
    - Habitat (Station, Base, or Suit);
    - Payload (or Instrument, hosted in a Spacecraft or Habitat).

The SO physical elements may correspond to ABA or SSI types, and they may be crewed or uncrewed/robotic.

Recent discussion has asked whether the SO charter might be extended. A draft of this extension [Future]:

- begins with the current scope, in which SO subnet and SO app covers the communication onboard a spacecraft between application software and devices that provide the sensory and actuation capabilities of the vehicle; and

- generalizes the idea of ‘spacecraft’ to include a variety of ‘vehicles’ in which a local network connects sensors and actuators to processing elements onboard the vehicle for the management of the functions of the vehicle; this generalization might include the following possible kinds of SUNs:
  - spacecraft (current scope), including Earth observation, exploration, and relay missions,
  - human excursion vehicles, including spacesuits,
  - uncrewed excursion vehicles, including probes and rovers,
  - habitats,
  - all of the possible space segments shown in the these examples are candidates for use of SO subnets, other SO services, and description by EDS and DoT.

The external communications between vehicles, and between vehicles and mission control centers, use other CCSDS standards:

- How to treat docked vehicles that may share the control network.
- The SO wireless standards cover one of the areas of communication between vehicles. [Future]
- The boundary between SO and extra-vehicular communication devices consists of a spacecraft side and an external side.
  - The spacecraft side is described and managed like any other onboard device interface.
  - The external side is described by other CCSDS communications standards.

8.4 SECURITY CONCEPTS FOR PHYSICAL VIEWPOINT

The system elements that provide user services typically will be secured physically. The following physical security methods are likely to be employed at service system boundaries:

- operational systems will be within a secure physical perimeter;
- only approved and trained staff will be allowed physical access to operational systems;
- appropriate credentials and vetting will be required to gain access to operational facilities;
- isolated LANs and firewalls are likely to be used to secure the operational systems;
- operational systems may be configured to only be accessible via proxy agents or a ‘DeMilitarized Zone’ (DMZ).

Physical and communications security approaches among space vehicles, including those that are tethered or attached at mating surfaces, and those that are operating in proximity with one another, are a matter for negotiation among the agencies engaged in such activities. All of the available terrestrial mechanisms, specialized as needed for such remote/disconnected uses, may be employed.

The security documents (references [38] and [39]) provide more details about security and threat analysis. Elements located in space typically will be secured by various encryption algorithms and protocols that have been described in more detail in 7.6.

9 DEPLOYMENT VIEWPOINT

9.1 OVERVIEW

The Deployment Viewpoint provides illustrative examples of how functions from the Functional Viewpoint may be deployed across the set of physical deployment nodes identified in the Physical Viewpoint. This viewpoint also shows the resultant application-level interactions between functions in terms of services (Service Viewpoint) and information objects (Information Viewpoint) exposed to the potential interoperability boundaries between deployment nodes. Functional interactions between functions co-located on the same node are omitted for clarity.

An intentionally limited, but useful, set of example deployment views are provided, but this cannot be exhaustive because there is a very large number of possible deployments.

9.2 MO FUNCTIONAL DEPLOYMENT

9.2.1 GENERAL

The Physical Viewpoint showed that there are many possible physical deployment architectures for a space system comprising multiple deployment nodes. In this Deployment Viewpoint, the distribution of MO functions across these deployment nodes is shown. For any given physical architecture, there is a wide range of possible functional deployments.

An objective of the Deployment Viewpoint is to show which CCSDS Recommended Standard information exchanges and services are exposed to the potentially interoperable interfaces that may occur at the boundaries between physical deployment nodes. It is these interfaces for which standardization is most appropriate. Within the MO Functional Viewpoint, information exchanges and services are only fully resolved at the level of decomposition of MO functions, not at the level of individual Application Layer data exchanges. Protocol details are handled in section 4. Some of these MO deployments involve onboard elements, and these are understood to use SO onboard services when required.

With many possible physical deployment architectures, and many possible Level 2 functional deployments for each physical architecture, it is not possible to cover every possibility within this document. For this reason, a realistic and representative set of deployment examples has been selected in order to demonstrate coverage of:

- all MO functions [Functional Viewpoint];
- all MO information objects [Information Viewpoint];
- all MO services [Service Viewpoint];
- all identified classes of communications context (Link type) [Communications Viewpoint];
- all identified example MO deployment nodes [Physical Viewpoint];
- a range of Mission types.

Representation of common and generic services is omitted from this viewpoint to reduce complexity, but in principal these can be applied across any of the identified boundaries.

Showing all MO Level 2 functions within a single diagram is not practical. The functional breakdown shown within each deployment node is therefore limited to MO Level 1 functions. If only a subset of Level 2 functions is deployed on a given node, then these are listed within the corresponding color-coded function 'bubble'. If all Level 2 functions are present, then the Level 1 functional group name is used.

Only those interfaces exposed at a boundary between deployment nodes are shown. Interfaces between functions within a deployment node are omitted to reduce clutter. The set of internal interfaces present can be derived from the Functional Viewpoint.

The example Functional Deployments given represent the following use cases or scenarios:

a) Communications Satellite (ABA);

b) Earth Observation Mission (ABA);

c) Deep Space Mission with Orbiter/Relay and Rover (SSI);

d) Constellation Mission (SSI);

e) Hosted Payload (SSI);

f) Crewed Mission (SSI).

The selected use cases are only examples; many other mission deployment scenarios are possible. The functional deployment illustrated for each scenario is also only an example, other physical architectures and functional deployments are also possible for real space missions.

The examples show existing CCSDS Recommended Standards, those currently under-development, and those identified on the future CCSDS roadmap. Alternative standards and bespoke solutions may also be used to implement the identified interactions, but these are not addressed.

The deployment scenarios show where interoperable boundaries could lie within a mission operations system. It is not implied that all interfaces shown in these example deployment scenarios would necessarily employ CCSDS MO standards. They may only be used only at an 'edge-to-edge' interoperability boundary between agencies, missions, or systems. In this context, several deployment nodes and functions may be considered internal to a wider system. These non-interoperable interfaces may utilize other standards and proprietary or bespoke interface specifications. The CCSDS MO services may be seen as a set of options available for selection to support interoperable interfaces at the boundary of a system, with minimal impact on existing spacecraft or ground systems. Actual deployments have the choice of whether to apply MO standards only at the interoperable boundaries of a system, or to use them for the internal boundaries shown within a system.

The remainder of this section presents each of these deployment scenarios in turn, preceded by a discussion of the representation of distributed functions. The descriptions provided are incremental, describing differences and additions to the preceding examples.

9.2.2 DISTRIBUTED FUNCTIONS

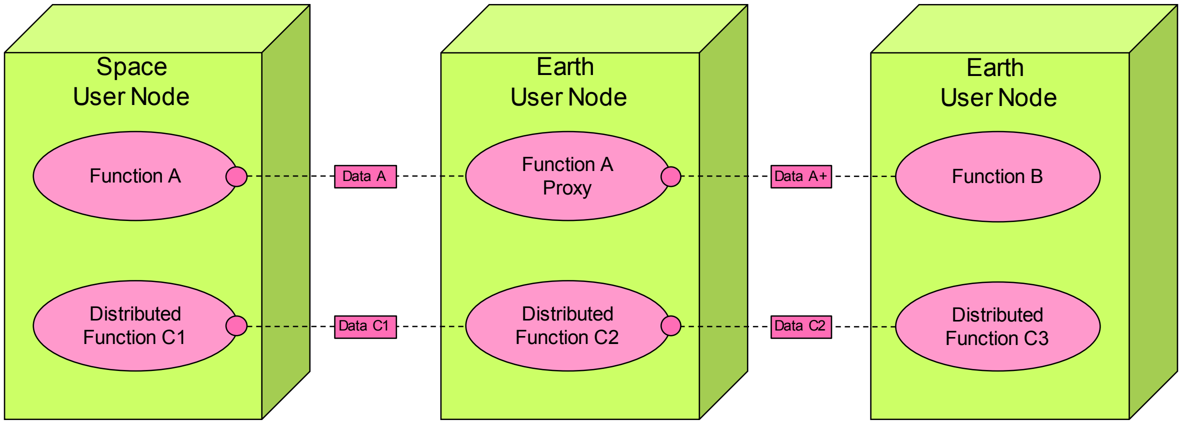


Figure 9-1: Distributed and Proxy Functions

The Functional Viewpoint in general only shows application-level interfaces (exchanges of information objects or services) between heterogeneous functions. In a distributed system, however, it is common for individual functions themselves to be distributed, resulting in exposed application-level interfaces between different nodes.

Figure 9-1 shows two ways in which a function may be distributed across multiple nodes.

Proxy Functions

MO functions may be distributed between space and ground deployment nodes, with the MOC (an Earth User Node) or other nodes acting as ground proxies for the Spacecraft, for example, Monitoring and Control. In figure 9-1, Function A is deployed onboard the spacecraft, while Function A proxy is deployed on ground. Other ground-based functions [Function B] then effectively interact with the spacecraft via the ground proxy. Where physical communication with the spacecraft is intermittent, this has the additional benefit of providing a permanent entity on the ground representing the last known state of the spacecraft function and queuing interactions with it. Mission Control systems typically act as a proxy for the spacecraft in this way. The proxy function may translate among different data representations or communications modes, such as translating from rather verbose terrestrial MO service mappings to efficient space link communications. The proxy function may also augment the data provided by the onboard function. For example, deriving additional status information and statistics or performing checks.

Distributed Functions

MO functions may be distributed across multiple Nodes, requiring peer-to-peer application interfaces between functionally similar applications, as well as those between functionally dissimilar applications identified in the Functional Viewpoint. In the figure 9-1, Function C is distributed between the spacecraft and two ground deployment nodes. This could be the case for functions such as Mission Planning and Scheduling: both POC and MOC may be involved in Mission Planning and deliver plans (schedules) to the spacecraft for execution. In this case, the same standard interface may be used between the distributed elements, although the content may differ across each interface. A similar situation may occur in data processing contexts as data are converted from raw observations onboard the spacecraft to calibrated data to higher-level data products and thence to science conclusions on the ground.

9.2.3 COMMUNICATIONS SATELLITE (ABA) EXAMPLE

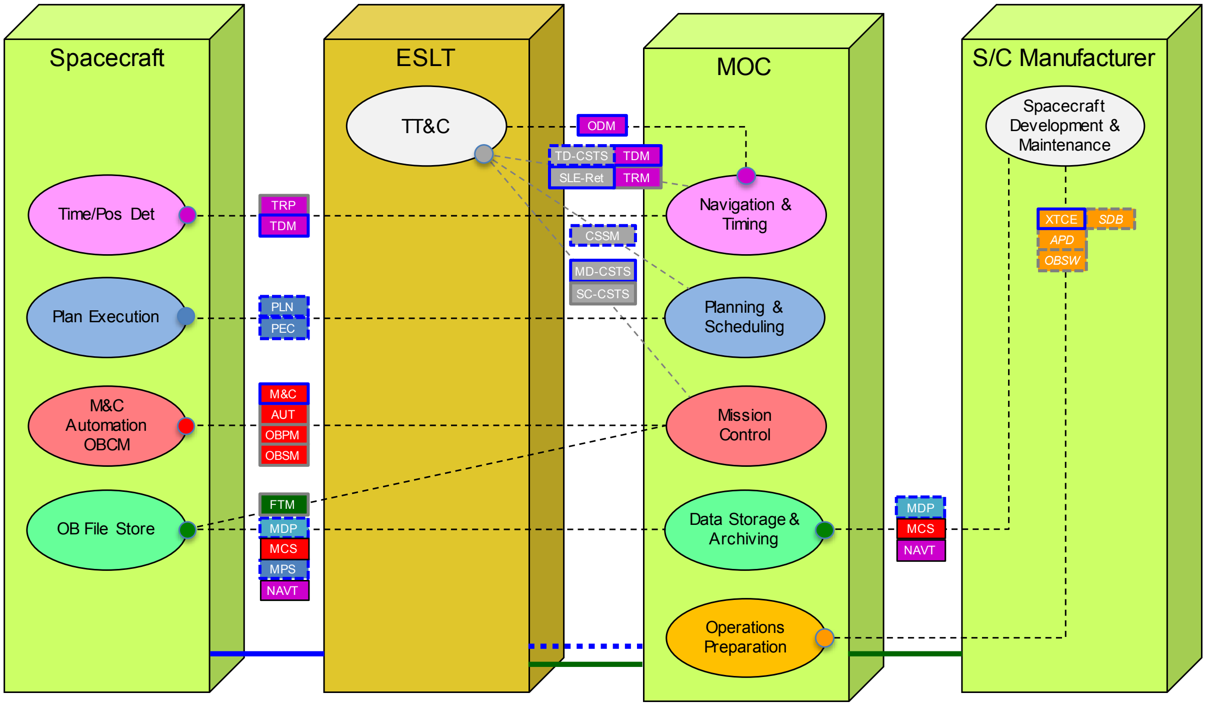


Figure 9-2: Example Functional Deployment (ABA): Communications Satellite

The example functional deployment shown in figure 9-2 represents a relatively simple traditional mission comprising a communications satellite, ground station (ESLT), and MOC. This deployment architecture could also support cases in which there are multiple satellites and ground stations, providing there is no interaction between the satellites. The only additional ground user node that is shown is the spacecraft manufacturer.

The communications architecture follows the ABA pattern identified in the CCSDS SCCS-ADD.

Onboard functions:

- Mission Control: basic spacecraft Monitoring & Control; onboard procedures (Automation); Onboard Configuration Management (procedures and software).
- Mission Planning: onboard scheduler (Plan Execution);
- Navigation and Timing: onboard GNSS receiver supporting Time and Position determination;
- Onboard File Store:
  - used to store M&C history,
  - OBCP definitions,
  - OBSW images, plans (schedules),
  - position and timing data,
  - mission data products.

MOC functions:

- Mission Control (all functions);
- Mission Planning;
- Navigation and Timing:
  - Orbit Determination and Propagation,
  - Attitude Determination,
  - Maneuver Planning,
  - Time Correlation;
- Data Storage and Archiving: Operations Archive;
- Operations Preparation.

These diagrams do not draw attention to the distinction, but many Mission Operations application interactions are exposed across the Spacecraft-to-MOC interface, and in most existing missions, non-CCSDS, proprietary, or bespoke protocols are currently used at the Application Layer. With the deployment of multiple spacecraft from different agencies that are intended to collaborate, and with the migration of increasingly complex functions onboard the spacecraft, there should be increasing benefit to the use of standardized MO services, or at least standardized, interoperable information exchanges, across these interfaces.

The ground stations (ESLT) support TT&C functions, which include spacecraft TT&C, tracking, and ranging, and these may be dedicated to the mission or, more often, are multi-mission capabilities providing cross-support services and interfaces. There is widespread use in existing ESLT service providers of the cross support services and Navigation message formats at the interfaces to the user MOC.

The spacecraft manufacturer typically provides initial telemetry and command definitions (spacecraft DB), OBSW, and OBCP definitions. CCSDS currently only provides limited coverage of these interfaces with the XTCE Recommended Standard for the exchange of telemetry and command definitions. The spacecraft manufacturer may also support performance monitoring and anomaly investigation throughout the mission lifetime, requiring access to historical Mission Control and Navigation data.

This example only addresses Mission Operations functions for the communications satellite, and the associated ground communications infrastructure (Network Operations Center [NOC], ESLT ground stations, and user terminals) is outside the scope that is considered.

9.2.4 EARTH OBSERVATION SATELLITE (ABA) EXAMPLE

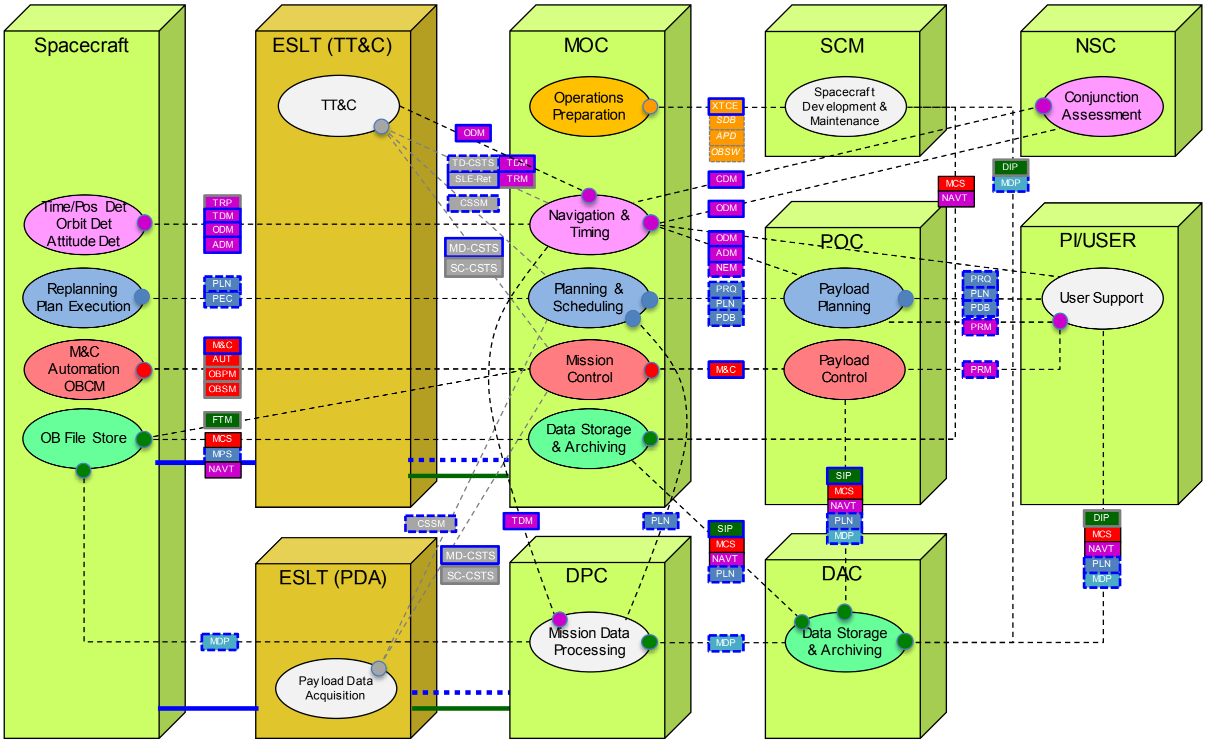


Figure 9-3: Example Functional Deployment (ABA): Earth Observation Satellite

This example builds upon the previous case of an ABA space link communications architecture but illustrates the kind of more complex ground segment architecture that is typical of Earth Observation missions.

Onboard functions are extended to include:

- Navigation and Timing: onboard Orbit and Attitude Determination;
- Mission Planning: Replanning in response to events detected onboard.

The migration of more complex functionality onboard the spacecraft exposes additional, and more complex, Mission Operations interactions across the Spacecraft-to-Ground Segment interface: Orbit and Attitude Data; higher-level Mission Planning interactions.

Instead of there being a single MOC, the mission ground segment in this example also includes separate POC, DPC, and DACs:

- MOC functions are as before.
- POC functions include:
  - Payload Planning [Mission Planning];
  - Payload Control [Mission Control].
- DPC hosts Mission Data Processing.
- DAC hosts a long-term mission Data Archive of both Mission Data Products and Operations History. This archive may be shared by multiple missions.

A community of PIs or users is responsible for tasking the mission (Payload Planning) and receives the acquired mission data. These primarily interact with the POC.

An external NSC supports Conjunction Assessment and provides collision warnings.

It is common in Earth Observation Missions to use separate communications frequencies and space data links for TT&C and the higher data rate mission data downlink. This often results in an architecture with separated ground segments for Mission Operations and Payload Data processing, possibly utilizing separate ground stations (ESLT) for TT&C and Payload Data Acquisition (PDA), connected to MOC and DPC, respectively. Alternatively, a single Ground Station can route data to separate MOC and DPC systems.

This distributed ground segment architecture results in many more MO application-level interactions being exposed to the interfaces between ground segment deployment nodes:

- Navigation messages are exchanged between the MOC and POC, PIs/Users, and NSC. In some cases, Mission Data Processing can also be the source of spacecraft position and attitude data derived as a by-product of image processing.
- Mission Control interactions between MOC and POC.
- Mission Planning interactions between MOC and POC, POC, and PIs/users and MOC and DPC.
- Given the presence of a separate downlink data path, Mission Data is virtually distributed at application level directly by the spacecraft to the DPC. The DPC forwards raw and derived Mission Data Products to the DAC, where they can be accessed by PIs/users.

9.2.5 DEEP SPACE MISSION (SSI) EXAMPLE

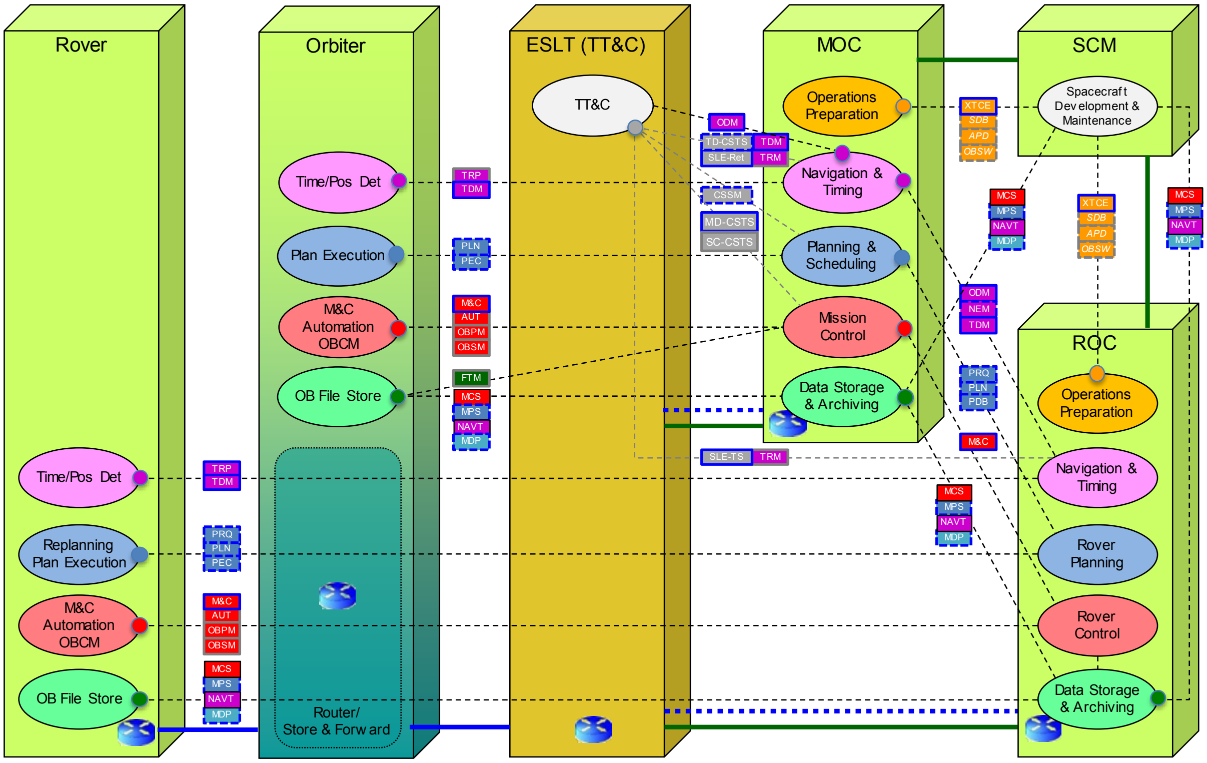


Figure 9-4: Example Functional Deployment (SSI): Deep Space Mission

The Deep Space Mission example shown in figure 9-4 has been chosen to illustrate a more complex space segment comprising a planetary Orbiter that also provides relay/router services for a surface-deployed Rover. It uses an SSI style of communications architecture based on DTN.

A simplified ground segment (compared with the Earth Observation example) comprises a MOC that is responsible for the Relay Orbiter and a separate Rover Operations Center (ROC). Such a mission may also include DPC, DAC, POCs, and PIs/users in a more distributed ground segment, but as this has already been covered in the previous example, it has been omitted here.

There is significant interaction between MOC and ROC, including Mission Control and Mission Planning coordination, and Navigation & Timing information exchange. They will also need to exchange communications link planning and geometry information to coordinate relay links during orbiter over-flights.

The Orbiter acts as a communications relay for the Rover. Orbiter, Rover, ESLT, MOC, and ROC all support the SSI routing functions that enable end-to-end communication using DTN store and forward routers onboard the Orbiter and at each of the nodes. This enables the ROC to have direct end-to-end interaction with the Rover using networking protocols that handle, in an automated fashion, the reliable delivery of commands and data over what may be frequently disconnected links. In this configuration, the store and forward DTN protocol deployments operate autonomously to send data whenever the links are available. Alternatively, the MOC could act as a ground-based Router for all data to/from the Rover.

The Rover is assumed to have a high degree of autonomy, both in terms of automation and onboard planning. GNSS is not available to support position determination, but other systems, such as landmark recognition and inertial navigation, may be used to provide position data. Autonomous path planning and hazard avoidance may also be deployed on the Rover to reduce the need to do 'joy-sticking' when there are long round trip light time delays. This may expose more complex, or at least different, Mission Control and Mission Planning interactions across the space-ground interface.

While only a single Spacecraft Manufacturer (SCM) node is shown, it is highly likely that there would be separate spacecraft manufacturers for the orbiter and rover, and different providers of the instrument suites hosted on these spacecraft is not uncommon.

9.2.6 CONSTELLATION MISSION (SSI) EXAMPLE

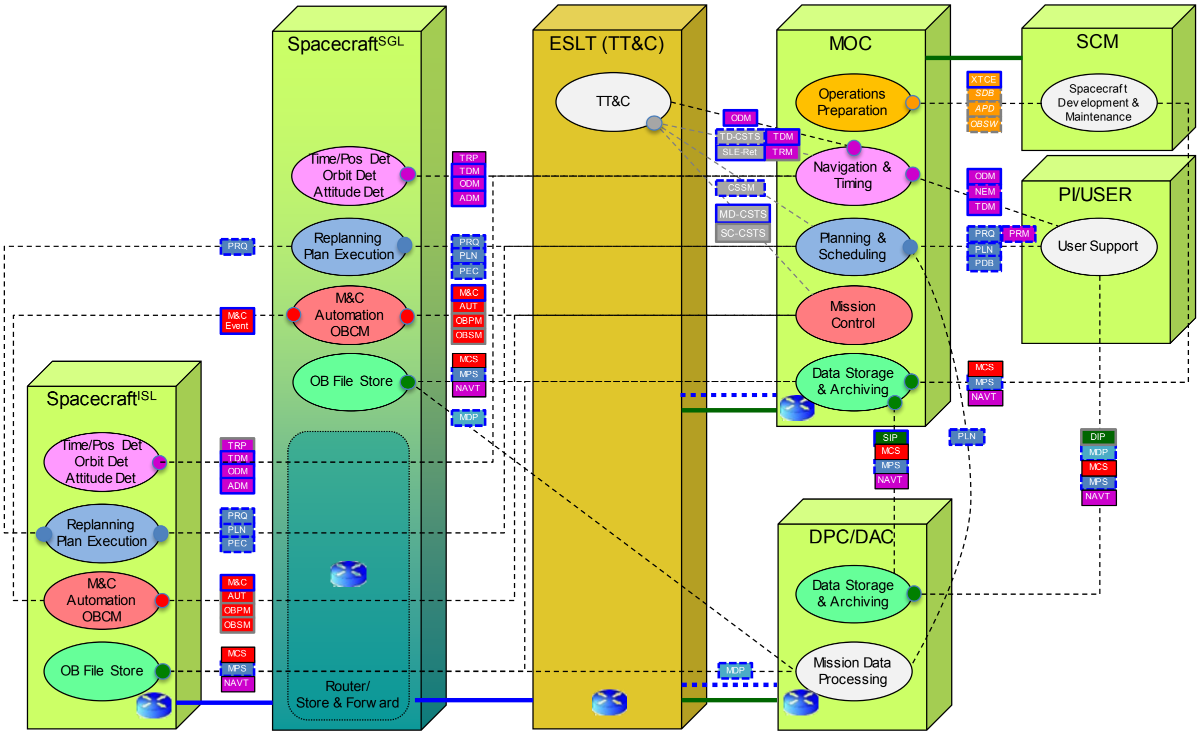


Figure 9-5: Example Functional Deployment (SSI): Constellation Mission

The example shown is of an Earth-orbiting Constellation of Earth Observation satellites. Like the Deep Space example, it is based on the SSI communications architecture.

There may be many different Spacecraft nodes in any given constellation. An underlying assumption is that there may be multiple satellites in each of potentially multiple constellation ‘planes’. Satellites within the same ‘plane’ may be connected via Inter-Satellite Links (ISLs). At least one satellite per ‘plane’ will be in direct contact with the ground via a Space-Ground Link (SGL) to an ESLT.

Only two Spacecraft instances are shown in the diagram: a Spacecraft that is currently using an SGL, and another Spacecraft that is connected to the ground via an ISL to the first. These are roles that the spacecraft are fulfilling at a particular point in time; the satellites themselves may be interchangeable.

In this example, onboard functionality includes:

– onboard GNSS receivers and orbit determination;

– onboard event detection and collaborative replanning between Spacecraft.

The existence of ISLs allows direct communication between satellites. This is assumed, in this example, to be aimed at enabling autonomous operation of the satellite constellation. Events may be detected onboard the satellite (e.g., cloud cover, forest fires or volcanic eruptions, oil slicks). These can be used by the onboard re-planning function to schedule observation of the detected event, to defer a planned observation, or to signal another satellite to take follow-up observations. Given a short period of target visibility, it may not be possible to respond to the detected event locally on the detecting satellite. Instead the event can be communicated to the following satellites, so that they can re-plan to perform their own observations of the event. Alternatively, a PRQ could be forwarded to another satellite.

The fact that the mission plan can be updated locally on the satellites as a result of onboard re-planning requires more complex interaction among the onboard planners, and between the onboard planner and ground-based mission planning to ensure visibility of the current mission plan and its execution status.

Compared with the Earth Observation example given in 9.2.4, a simplified ground architecture is shown here. This was done to focus on the constellation aspects of the mission. A single payload processing center is shown, hosting DPC and DAC functions and supporting direct interaction with external users. This could be omitted altogether and integrated with the MOC, or separate DPC, POC, and DAC nodes could be identified as for the previous Earth Observation example.

9.2.7 HOSTED PAYLOAD MISSION (SSI) EXAMPLE

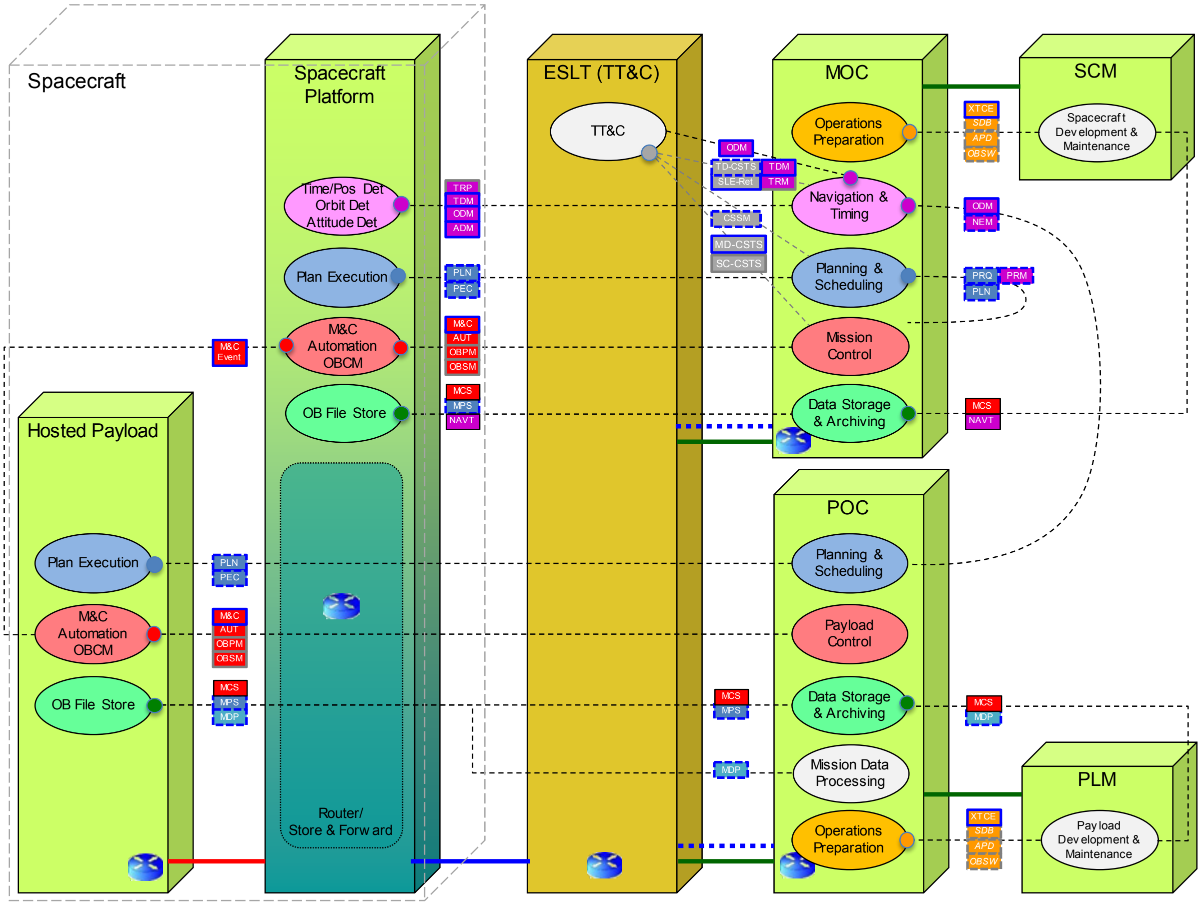


Figure 9-6: Example Functional Deployment (SSI): Hosted Payload

The Hosted Payload example shown above is topologically similar to the Deep Space Mission example given previously, and is also based on the SSI communications architecture (although an ABA equivalent would also be possible).

The key difference is that the Hosted Payload is physically located onboard the Spacecraft (even though it is shown separately in this diagram) and interfaces with it via a SO compatible onboard subnet link.

In this example, the Spacecraft is assumed to be in Low Earth Orbit, so it has onboard GNSS providing Time and Position Determination, together with onboard Orbit Determination.

Both the Spacecraft and the Hosted Payload independently have onboard Automation (procedures) and onboard schedulers (Plan Execution). The Hosted Payload performs Image Acquisition, storing acquired mission data in files in an onboard file store. Events detected by instruments or processes onboard the host Spacecraft are passed to the Hosted Payload to enable synchronization of payload operations.

The simplified ground segment architecture that is shown in this example has an independent MOC (for Spacecraft) and a POC (for Payload), the latter responsible for operation of the hosted payload and associated mission data processing.

Given the SSI communications architecture, the Hosted Payload may have end-to-end communication with its POC, supporting internetworked interaction for Mission Planning, Payload Control, and direct transfer of acquired image data to the Mission Data Processing function.

In this example, interaction between MOC and POC is restricted to transfer of Navigation Data from MOC to POC and Mission Planning interactions. This could be extended to support Mission Control interaction between Mission Control and Payload Control functions.

Separate manufacturers are assumed for Spacecraft and Payload.

9.2.8 CREWED MISSION EXAMPLE (SSI)

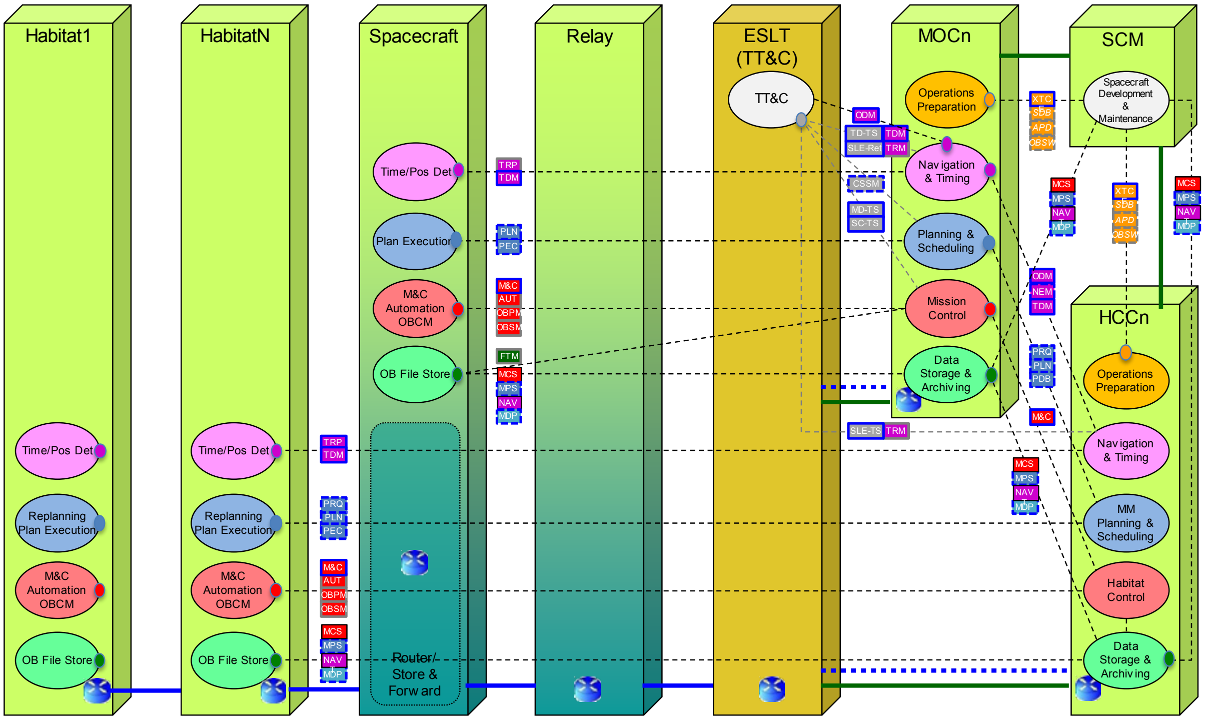


Figure 9-7: Example Functional Deployment (SSI): Crewed Mission

The Crewed Mission example shown above is essentially similar to the Deep Space Mission case, but with multiple Habitats (space stations, surface bases, or space suits) instead of a single Rover. There may be one or multiple Habitat Control Centers (HCC) associated with the Habitats, as well as potentially multiple MOCs. SSI protocols are used throughout to provide a communications fabric that enables many different kinds of interactions among the elements. This example also assumes that while the mission as a whole is a collaboration, each of the elements is developed and managed by a different organization. Adoption of interoperable and cross-supportable communications protocols is essential as chaos is to be avoided.

For the case of a surface operations, a surface relay station may be an essential part of deployment, but this has been omitted for clarity as it would provide much of the same functionality and services as a space-based relay. That said, surface relay stations may have to provide a range of 'terrestrial' as well as space communications protocols in order to support all the different kinds of 'Habitats' that may be in operation at any one time.

There may also be multiple kinds of relay spacecraft serving as communications proxies for Habitats and supporting the communication links between the habitat and its HCC. Habitats may also be dependent on a single spacecraft, rather like a hosted payload, with a dedicated 'local' communications link to that spacecraft and internetworked relay services to other elements.

There may also be tiered habitats with differing purposes and characteristics, such as an orbiting space station or a surface base and its associated suits or vehicles.

This highlights that there is a wide range of potential deployment architectures, but that the set of interactions between Habitats and their HCCs are essentially the same and can be supported by the same set of end-to-end information exchanges or services all operating over SSI communications protocols and local or 'long haul' space data links. Human-rated missions will have requirements for Class A implementation approaches and added redundancy, but the fundamentals of the communications and navigation architecture remain much the same.

Human-rated missions will also have higher performance data rate requirements to support audio and video communications, and these will typically require implementations that reduce timing delays and jitter that can disrupt these streaming services. Of course, these considerations really only apply out to Lunar distances where the Round Trip Light Times (RTLTs) are of order 2-3 seconds. At greater distances than that, these communications will take on the characteristics of store and forward delivery, even if the data are transmitted immediately.

The elements deployed on the Earth may well be as complex, or even more complex, than those shown in Fig 9-3 through 9-6. Each of the Habitats, Spacecraft, or Relays may have their own MOC and/or their own DPC. The Relay spacecraft may be an Orbiter servicing multiple downlinks from a stationary orbit, or it may have different orbital characteristics and only be visible to landed elements for portions of its orbit. The Habitats and the Orbiting Spacecraft may offer communications and MOSes to other elements, or they may be dependent upon others for certain services.

Since all of the elements of such a Crewed Mission 'constellation' are unlikely to arrive all at once, the communications architectures that are chosen must be sufficiently powerful and flexible as to support an evolving set of elements. The same is true of the MOSes that are provided, although, as software elements, these are rather more easily adapted than the more hardware-focused communications services.

9.3 SO END-TO-END DEPLOYMENT VIEWS

9.3.1 GENERAL

SO service, function, and protocol deployments onboard, in relationship to ABA and SSI space to ground link configurations, appear in 7.4. In any of these deployment diagrams, it is possible that onboard Applications might be implemented using the MAL, or other software framework. (See 9.5 for more details on this.)

The SO functions, protocols, and services may be deployed on any of the space elements that are shown in 9.2. All of the space elements typically make use of onboard subnets, real-time operating systems, flight software, and suitably robust physical architectures. All of them will require onboard use of subnets, packet services, software, or other message busses, and services to interface both low-level and capable devices. As discussed in sections 4 and 5, SO services, functions, and protocols provide capabilities intended to make the development and deployment of devices, both sensors and effectors, easier to design and manage.

This section describes deployment scenarios in which SO capabilities can provide communications for organizational units near a spacecraft. Similar examples within a single spacecraft appear in 7.4.

- extravehicular activity (9.3.2);
- multiple subnetwork technologies in one spacecraft (7.4.4);
- multiple processors in a vehicle (7.4.5);
- docked vehicles (9.3.3).

The EDS is designed to be used to describe interfaces of different kinds of devices and their behavior. An area for future development is to extend the EDS to also enable description of multiple devices in various deployments [Future]. Such deployments occur in network concentrators, such as remote interface units and wireless networks (references [12], [13], and [14]). So, EDSes at one level of specificity describe each device and at the next level of specificity describe assemblies constructed using multiple devices and subnet technologies.

9.3.2 EXTRAVEHICULAR ACTIVITY

SO protocol deployments for extravehicular activity [Future] appears in figure 9-8. The tree-like command and data handling pattern of communication serves in this context. A short-range radio connects to the spacecraft subnetwork and acts as a concentrator for one or more endpoints at each extravehicular participant. The EDS-specified device service presents the interface for the short-range radio.

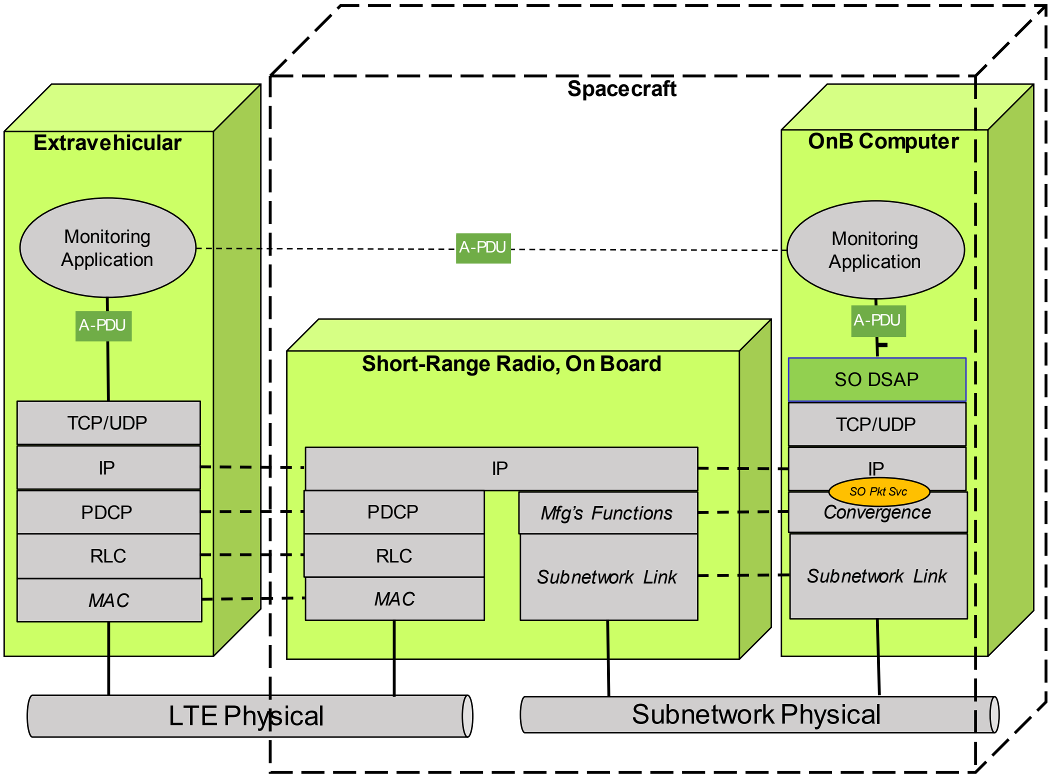


Figure 9-8: Extra-Vehicular Activity Link

- The monitoring application in the spacecraft in figure 9-8 uses a non-SO function to encode and decode its packets for use by the extravehicular component(s). In this case, the Internet Protocol (IP) is used for the packet format. The SO device service is shown here as DSAP, which provides an application programming interface to pass packets to/from the radio device but is not an end-to-end protocol layer. The onboard computer includes an IP layer to match IP routing in the network for the extravehicular communications.

- The SO Packet Service routes packets to/from the spacecraft subnet, such as SpaceWire, but is not a protocol layer. Convergence functions match device manufacturer's functions, if any, for the spacecraft subnetwork or are 1-sided, or are absent.

- A short-range radio acts as an IP router. The Packet Data Convergence Protocol (PDCP) layer compresses the IP headers. The Radio Link Control (RLC) assembles packets into segments. The MAC puts the segments onto the local area radio network for the extravehicular communications.

9.3.3 DOCKED VEHICLES

When vehicles dock, they can have a requirement to share instruments. For example, if the vehicles are of comparable size, they may need to share attitude control sensors and actuators so that one of the two can maintain attitude. If one vehicle is servicing the other, it can use sensors on the client vehicle to determine the effects of the service actions.

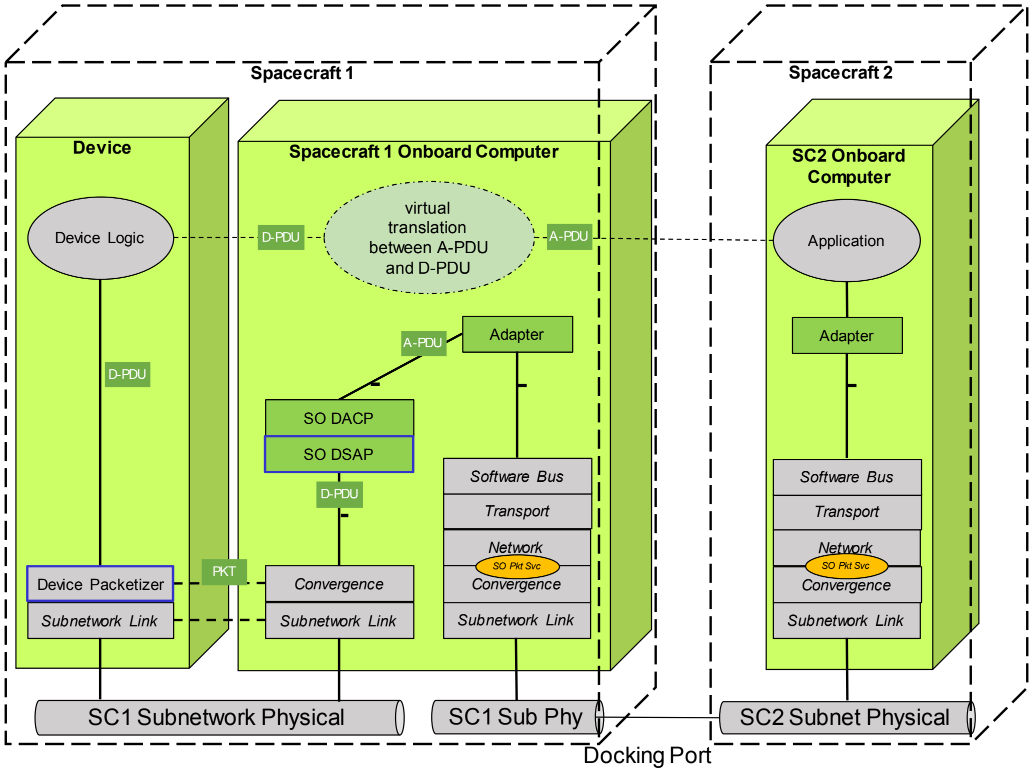


Figure 9-9: Docked Vehicles

When docked, a direct connection is possible between the subnetwork of one vehicle and the processor of the other vehicle, resulting in the configuration shown in figure 9-9. This configuration is efficient when there is no concern over unauthorized access to the data and devices on either vehicle. Alternatively, access control and security zone separation methods may be employed, but that is not shown in this diagram.

When the owners of the two vehicles are different parties, there may be a need to implement access control on both vehicles to filter the data and devices that each can access on the other. Docking occurs contingent upon positive identification of vehicles. Device Enumeration Service on spacecraft 1 provides a list of accessible devices, their network addresses, and metadata, filtered by access control, to spacecraft 2. These steps protect the instrumentation in Spacecraft 1; a similar arrangement can protect the instrumentation in Spacecraft 2. Protection of application endpoints and the data that they serve can be accomplished by means of Session Layer access control or by employing a Network Layer routing firewall at the point of connection.

9.4 SERVICE AGREEMENTS AND ACCESS ARRANGEMENTS

At this point, there are not any MO or SO cross-support or interoperability standards for service agreements or service access arrangements. The most typical uses involve local agreements between organizations that may be captured in some form of partnership document or operational support agreement, or, more formally, in an Interface Control Document (ICD). That said, there are some distinct differences between how services are contracted for, agreed upon, and provided in a typical MO deployment and how this might work in a typical SO deployment. The following lists identify some of the key distinctions between these two different operational domains:

a) In typical MO arrangements:

   – service agreements are expected to reference the standardized MO interfaces, but there is no standard form for such agreements;

   – service catalogs do not exist, per se, but services are described in standard MO books;

   – service provision requires selection of the sets of services to be used and agreement on the selection of the same transport mapping by all parties;

   – service interfaces are standardized by using agreed-upon data format exchanges or standard service interfaces, but additional bespoke services (domain-specific) can be defined for specific cases;

   – there is a Directory service that can provide a service interface on request, but there are no standard service management interfaces;

   – there are a variety of different services and data exchange formats, which are currently defined or under development as internationally agreed-upon standards;

   – addressing, interface binding, and deployment arrangements must be agreed upon multi-laterally, and this information must be captured in a mutually agreed-upon document; the multiplicity of possible bindings makes this essential.

b) In expected SO arrangements:

   – service agreements are not typical, but partnership agreements may be; where they exist, they are most often arbitrarily structured text documents and ‘in-kind’ arrangements;

   – service catalogs are not typical, but in the future, they may exist in the form of a directory of compliant systems and components offering EDS specifications;

   – Interface control documents may, in part, be specified using a set of EDS specifications within, and between, spacecraft;

   – component integration requires adoption of the same EDS representation and system structure mappings by all parties;

- service interfaces may be defined by using agreed-upon EDS exchanges;
- there are no standard service or device management interfaces provided; management of onboard devices is typically a local matter;
- there is a variety of different component types that may be documented by EDS, but libraries of components have yet to be defined using internationally agreed-upon standards;
- addressing, interface binding, and deployment arrangements may be described by EDS specifications, but there are many possible combinations, and onboard configurations are largely controlled by the spacecraft;
- Future iterations of the EDS specifications will address descriptions of spacecraft component deployments and also configurations of spacecraft.

9.5 TRANSITIONAL STRATEGIES

9.5.1 OVERVIEW

There are two different dimensions of transitional strategies that are relevant in the ASL context:

a) Transition from MO functions being deployed only on the ground to MO functions also being deployed in flight.

b) Transition from standard ABA, single spacecraft/single link deployments, to SSI deployments of multiple networked spacecraft.

9.5.2 INTEGRATION OPTIONS FOR THE MO/SO INTERFACE

9.5.2.1 General

The SO and MO services are two independent sets of technologies that can be integrated together. Three different (but not exhaustive) possibilities for their integration are captured in this section in order to provide a set of reference points along the spectrum of possibilities.

Neither SO nor MO services and functions were designed with any implementation inter-dependencies, so the coupling between them must be separately specified. This may occur on the ground or through a service access point on each onboard computer where an MO service or function is present onboard.

The software onboard differs from that typically found on the ground. Resources are most often highly constrained onboard, for a variety of reasons. Time is often constrained by scheduling processing onboard to occur within specific, periodic deadlines, so that real-time control loops can be serviced reliably. These control loops are essential for real-time homeostasis. High-priority activities, such as FDIR, must have resources commensurate with the importance of the fault to be serviced. Flight software must be qualified according to strict requirements for use onboard a vehicle. These considerations effectively block the simple migration of mission-operations software from the mission control center to the space vehicle processor and operating environment. Instead, mission-operations software is likely to either be limited to a proxy interface onboard, or it must be redesigned to execute within the constrained environment onboard, or it may be hosted in its own processor if such resources are available.

Three different cases of MO service integration with SO functions are presented in detail:

a) Case 1: SO device interfaces, subnets, and services are deployed onboard with the usual Real Time (RT) Flight SoftWare (FSW) for guidance, navigation and control, M&C, C&DH, FDIR, and power & thermal management, operating within the usual resource constraints. MO services are only on ground, with typical TT&C interfaces between flight and ground. This is a traditional case without any overlapping areas of interfaces between MO services and SO services and functions deployed onboard the spacecraft.

b) Case 2: The same SO services and RT FSW are deployed onboard, but MO Proxy interfaces are provided onboard, connecting to a subset of the usual RT FSW. This case uses MAL message exchanges over TT&C space link to the MAL Proxy onboard. This is a transitional deployment in which MO service interfaces are integrated with the onboard environment, but the real time onboard environment and services continue to operate in a normal fashion.

c) Case 3: The same underlying SO services and RT FSW are deployed onboard, but MO MAL-based services and frameworks have been adapted to the RT environment and migrated onboard as appropriate. MAL message exchanges are done over the TT&C space link. Some devices may also have ‘MAL native’ interfaces. This is an integrated situation in which MO functions and interfaces are adapted and operated in Real Time in the onboard environment.

9.5.2.2 Case 1: MO on Ground, SO in Spacecraft

Figure 9-10 shows the Case 1 configuration in which MO functions are deployed only on the ground, not onboard the spacecraft. SO functions are used for communications between an onboard device and a bespoke device handler. The device handler reads telemetry from the device and relays it through a TM/TC device onboard (nominally a radio), which further relays the telemetry across a space link to MO functions on the ground. An MO action service can send telecommands, for example, formulated as Space Packets, over the space link to the spacecraft, in which the TM/TC device relays the telecommands to an SO function, which sends the telecommands to the device handler. The device handler routes the telecommands through the SO service for the device, resulting in delivery of the telecommands to the device.

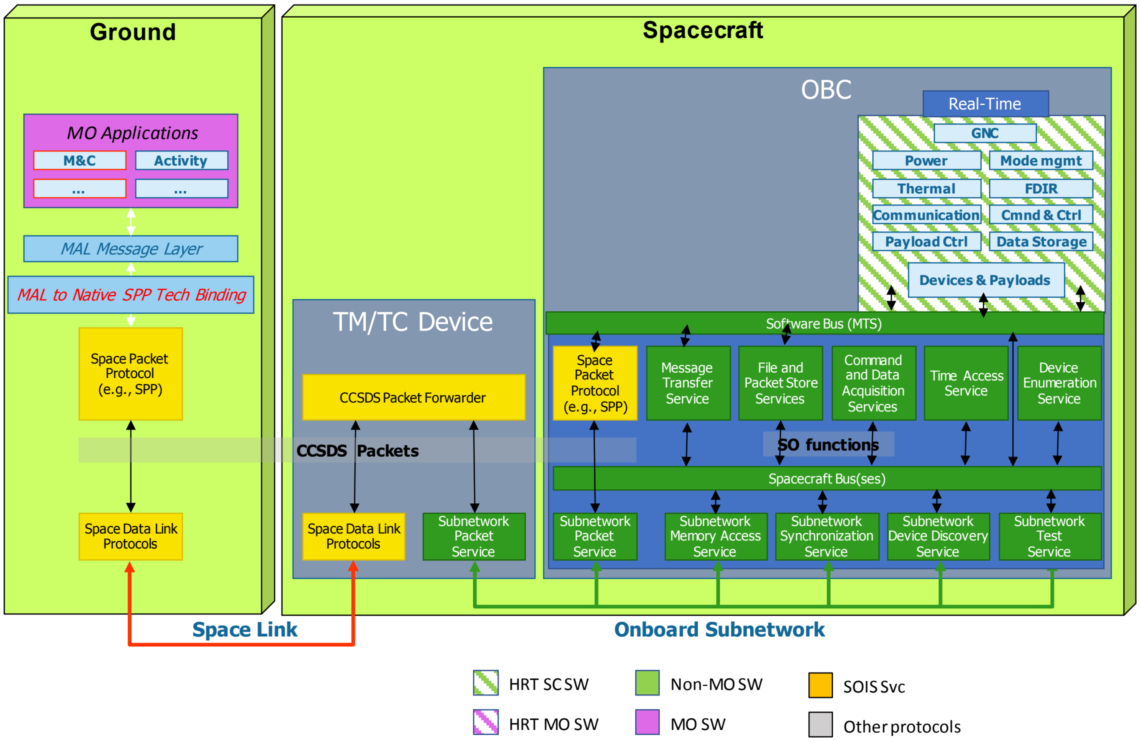


Figure 9-10: Case 1 Deployment, SO/MO Integration with MO on Ground

- Real-time applications are responsible for time critical S/C functions, execution control, health and safety, redundancy management, communications, failover, etc.
- All onboard apps are under control of the RTOS and R-T executive *in the onboard computer*.
- For Case 1 MO Application software and framework is only on the ground, not running onboard.
- MO Applications may be implemented to agency terrestrial software standards.

- All onboard devices and payloads, interfaces, and behaviors are described using SO EDS & DoT.
- *Onboard interfaces and devices will adopt the SO subnet and DACP/DSAP interface functions.*
- The SO transfer (Network and Transport) layer has been eliminated in this view; only packets and Software Bus (MTS) are shown.

9.5.2.3 Case 2: MO Proxy in Spacecraft with Space Link to MO on Ground

Figure 9-11 shows a Case 2 deployment with communication between an MO proxy function deployed onboard and MO services operating on the ground across a space link. SO functions deliver communications between onboard devices and an MO proxy element onboard. The MO proxy reads telemetry from the device using SO subnets and services, translates the onboard communications into an MO-compliant action service response, and relays the resulting MAL messages through a TM/TC device onboard, which further relays the telemetry over a space link to MO functions on the ground. An action service on the ground can send telecommands as MAL messages over the space link to the spacecraft. On the spacecraft the TM/TC device relays the telecommands containing MO action service requests to the MO proxy, which translates these MO commands into onboard communications compliant with SO services and protocols, and then the SO communications route the telecommands to the SO service for the device, resulting in delivery of the telecommands to the device. Given such an MO Proxy onboard, it would also be possible for an MO application onboard, such as a monitoring and control function, to communicate with the device through the SO proxy service interface.

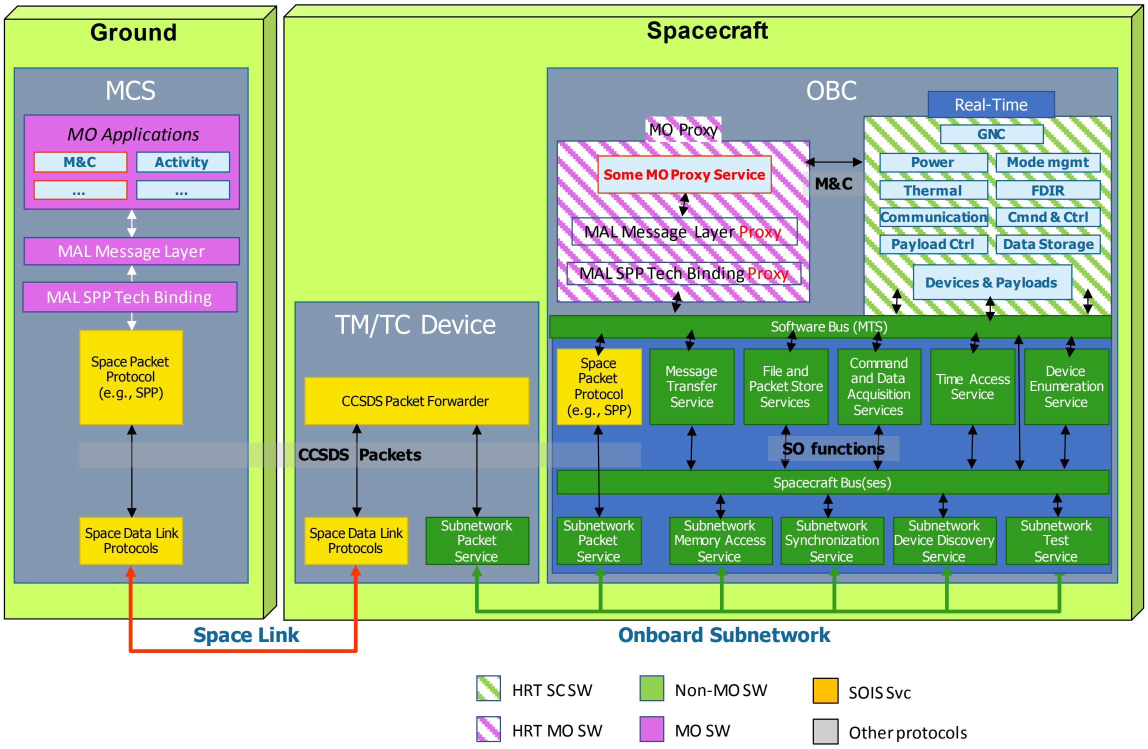


Figure 9-11: Case 2 SO/MO Integration for MO on Ground and in Spacecraft

- Real-time applications are responsible for time critical S/C functions, execution control, health and safety, redundancy management, communications, failover, etc.
- All onboard apps, including MO Proxy, are under control of the RTOS and R-T executive.

- For Case 2, an MO Proxy may run as onboard software *but* in a separate partition from Real-Time.
- The MO Proxy, and local MAL 'stack' onboard, must be implemented to Class A/B standards if they share a common processor w/ R-T software.
- All onboard devices and payloads, interfaces, and behaviors are described using SO EDS & DoT.
- *Onboard interfaces and devices will adopt the SO subnet and DACP/DSAP interface functions.*
- The SO transfer (Network and Transport) layer has been eliminated in this view; only packets and Software Bus (MTS) are shown.

9.5.2.4 Case 3: Communication between MO Services Onboard Spacecraft

Figure 9-12 shows a Case 3 communication between MO applications deployed onboard a spacecraft. An optional software bus, such as AMS, provides publish/subscribe capabilities. For applications on separate onboard computers, the communications would pass through the whole stack and use the physical subnetwork to communicate. For applications on the same computer, the communications may pass through the stack to a local 'loopback' function in the software bus.

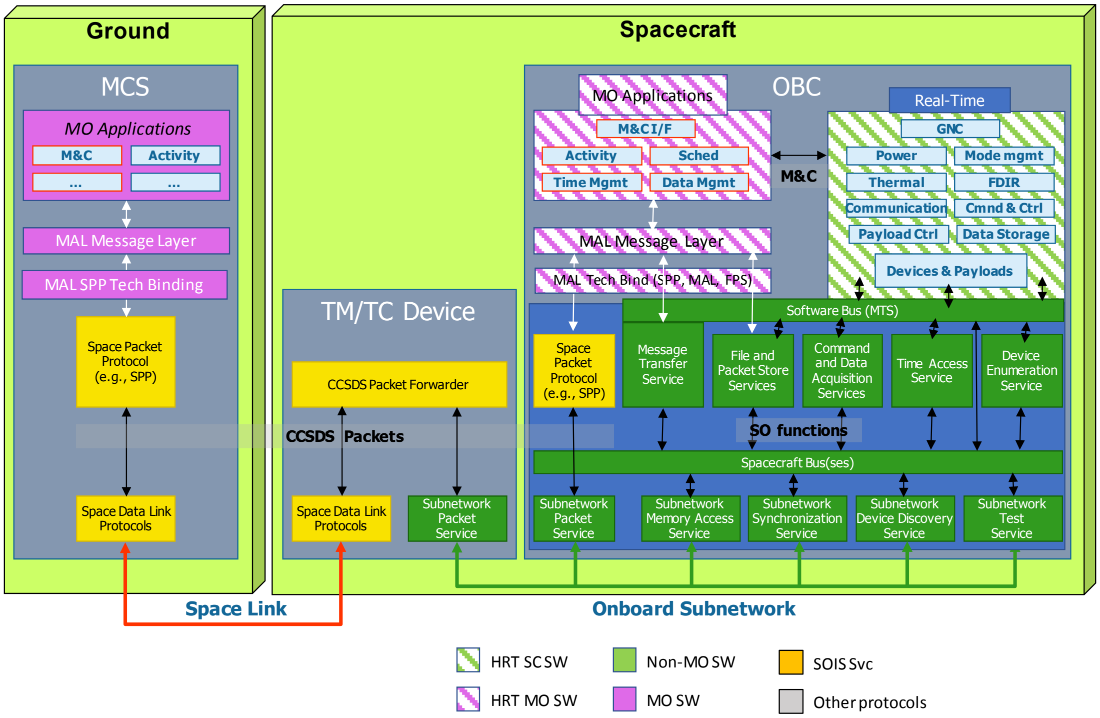


Figure 9-12: Case 3 SO/MO Integration for MO functions Deployed on Spacecraft

- Real-time applications are responsible for time critical S/C functions, execution control, health and safety, redundancy management, communications, failover, etc.
- All onboard apps are under control of the RTOS and R-T executive.
- For Case 3 MO Applications may be migrated onboard, running in a separate partition from Real-Time, or even in a separate processor.
- MO Applications onboard must be implemented to Class A/B standards if they share a common processor w/ R-T.
- The MAL will require technology bindings to SPP, the Software Bus (MTS), and the File and Packet Store.
- All onboard devices and payloads, interfaces, and behaviors are described using SO EDS & DoT.
- *Onboard interfaces and devices will adopt the SO subnet and DACP/DSAP interface functions.*
- The SO transfer (Network and Transport) layer has been eliminated in this view; only packets and Software Bus (MTS) are shown.

If a separate onboard processor is deployed for MO functions and services, then some of the Class A/B restrictions may be loosened at the risk of this part of the mission software not being as inherently robust. It is also possible to deploy MO functions and services into a payload device if it has an adequate processing environment to support the added resource impact.

9.5.3 OPTIONS FOR DEPLOYING MO AND SO IN ABA OR SSI ENVIRONMENTS

It is entirely possible to design the sort of ABA mission configurations and deployments shown in figure 9-13 to use the MO standard services and protocols only on the ground. Whether constrained to just the EUN or distributed among other Earth Nodes of different types (e.g., navigation, data processing, archival), deployment of MAL based services using a single technology binding can be used, along with a data representation and technology mapping suitable for use over the space link. The terrestrial deployment, if it does involve several nodes, would likely be built on familiar and widely supported TCP/IP networking suite. The space link in this instance, would likely use a mapping from MAL to some suitable local command and control protocol based on the CCSDS Space Packet, running directly over the usual CCSDS Space Data Link protocols.

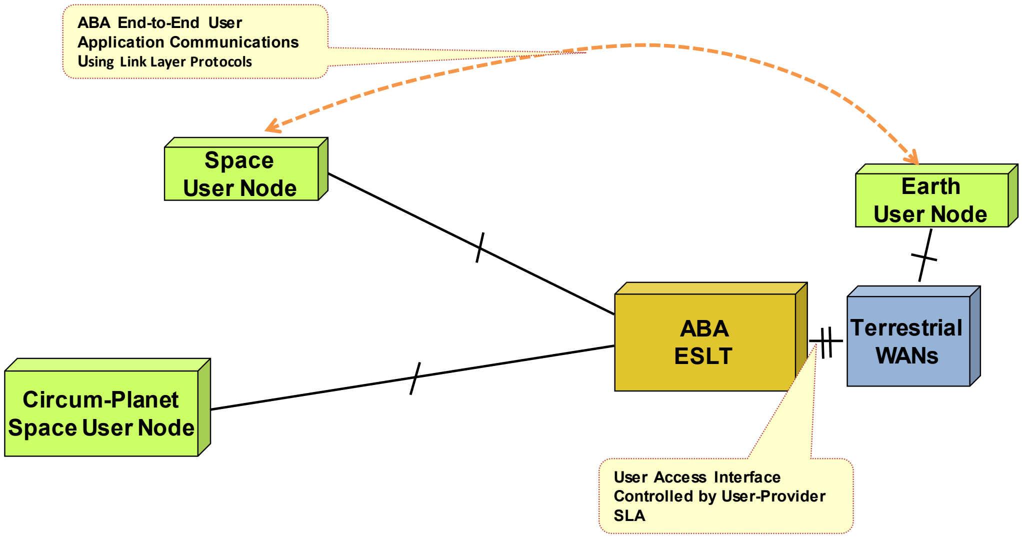

ABA End-to-End User Application Communications Using Link Layer Protocols
Space User Node
Earth User Node
ABA ESLT
Terrestrial WANs
Circum-Planet Space User Node
User Access Interface Controlled by User-Provider SLA

Figure 9-13: End-to-End View of ABA Nodes and Their Connectivity

In this kind of deployment, it would be entirely reasonable to have MAL-based services only in the EUN and to have more traditional, real-time, spacecraft architectures deployed in the SUN. SO services, of course, would only be deployed in the SUN. End-to-end services in this configuration would require two sorts of translations that have already been described, one from the SM&C MAL to Space Packets, and another from a terrestrial networked environment to transfers over a single space link.

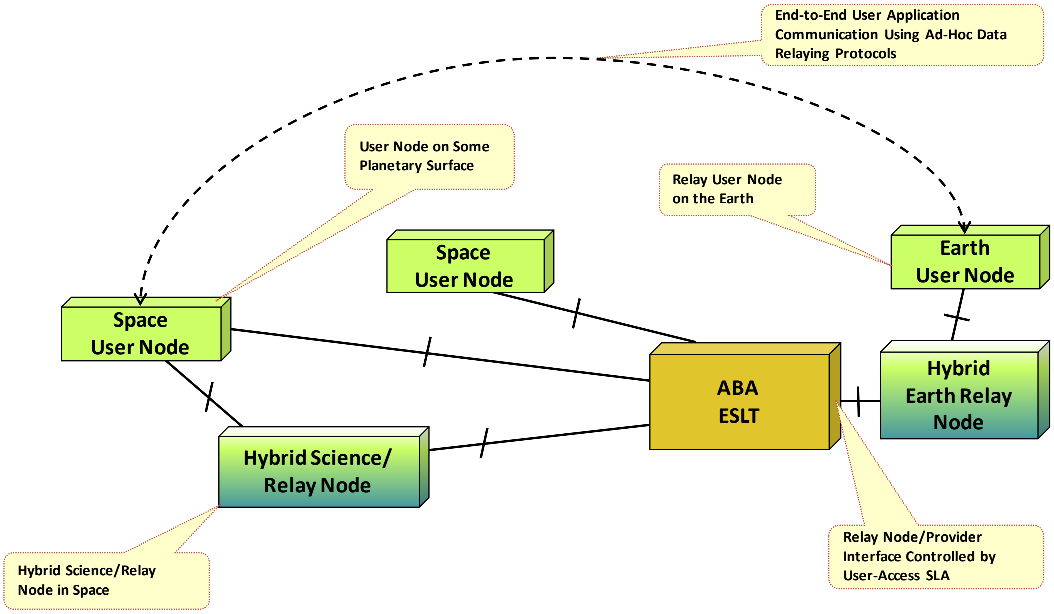

End-to-End User Application Communication Using Ad-Hoc Data Relaying Protocols
User Node on Some Planetary Surface
Relay User Node on the Earth
Space User Node
Earth User Node
Space User Node
ABA ESLT
Hybrid Earth Relay Node
Hybrid Science/ Relay Node
Hybrid Science/Relay Node in Space
Relay Node/Provider Interface Controlled by User-Access SLA

Figure 9-14: Interim ABA-Style Data Relaying Configuration

The SCCS-ADD (reference [50]) describes a transitional approach toward a fully networked environment that implements the SSI protocols in what are called the hybrid Earth Relay Node and Space Relay Node that continues to use an ABA ESLT configuration rather than deploying

SSI protocols and services in the ESLT. As shown in figure 9-14, this allows the EUNs and SUNs to adopt the basic SSI protocol set, which permits more automated delivery of data among multiple nodes and simpler operations without requiring ESLT upgrades. Once the SSI Network Layer is in place, the MO services can continue to be operated as in figure 9-13, or a transition of some MO services and framework can be implemented onboard.

However, the Interagency Operations Advisory Group (IOAG) Space Internetworking Strategy Group (SISG)-recommended approach is to upgrade the ESLT to include full SSI functionality and then adapt the Earth relay nodes and the space relay nodes to provide full SSI routing services. This can work with the sort of hybrid science/SSI mission that is shown here doing routing services. These elements then provide the simplest backbone configuration for SSI services. What is required next is to either develop the end-user nodes in space and on ground to also include basic SSI end-node capabilities, or to use the last-hop and first-hop services to communicate with them as non-SSI nodes. The former approach is preferable, but the latter will work and may be required as a transition path for many existing missions.

The strongest multi-agency motivation for taking the recommended transitional approach is that it is the best way to deploy an interoperable collection of elements that can evolve to support the full SSI services. In the absence of this approach, the alternative will be to continue to handcraft, mission by mission, the protocols, interfaces, and functions for doing simple data relaying. While this works, and it may be a lower-cost approach when viewed on a narrowly focused mission-by-mission basis, in the long run, it is more costly for each agency and for all of the agencies as a whole because these specialized, one-of-a-kind interfaces and protocols are continuously being invented, developed, and operated. Aside from being difficult to interoperate, they are also likely to be more idiosyncratic, convoluted, and fragile than well-designed and tested approaches using well-tested and vetted protocols.

9.6 SECURITY CONCEPTS FOR DEPLOYMENT VIEWPOINT

Security for the end-to-end Deployment Viewpoint utilizes all of the security elements described in earlier sections. In addition, this view is where use of end-to-end user encryption will be shown as a means to secure transfers from the ground to the final space routing nodes or SUNs. This was described in 7.6. For SSI missions, Network Layer security may also be applied, or users may choose to apply encryption end-to-end at the Application Layer. This was also described earlier.

Figure 9-15 shows two approaches for securing the data in an ABA end-to-end configuration. These optional stack elements are shown in green. SDLS is used to provide end-to-end link-layer security for a Space Packet Protocol (SPP) packet flow. The ABA user node may apply one (or both) of the encryption or authentication algorithms to packet data, load this into an AOS or TC frame inside the SDLS security block, and then use the normal service delivery mechanisms to transmit the data to the user space node, as reflected by the SDLS authentication/encryption layer. It should be noted that the ESLT service provider stack in figure 9-15 does not show the presence of the SDLS authentication/encryption layer since that security block is opaque to the CSTS F-Frame service.

Alternatively, the user MOC may apply encryption or authentication to a data file (File Secure), and that file may then be merged into the secure forward frame stream. There are no specific services defined in MO for doing such application data encryption of the frame data contents, so this would need to be a bespoke implementation. In both of these cases, the ABA SUN must be prepared to undo the security that has been applied, and some private means for establishing and managing keys must also be implemented. Both of these methods offer end-to-end link-layer security.

This figure also shows use of IPsec as an additional layer of encryption that optionally may be applied at the Network Layer to secure the IP traffic over the Internet. In most configurations, only one or another of these mechanisms would be employed, and it would be unusual to elect to use all of them, as shown in this figure.

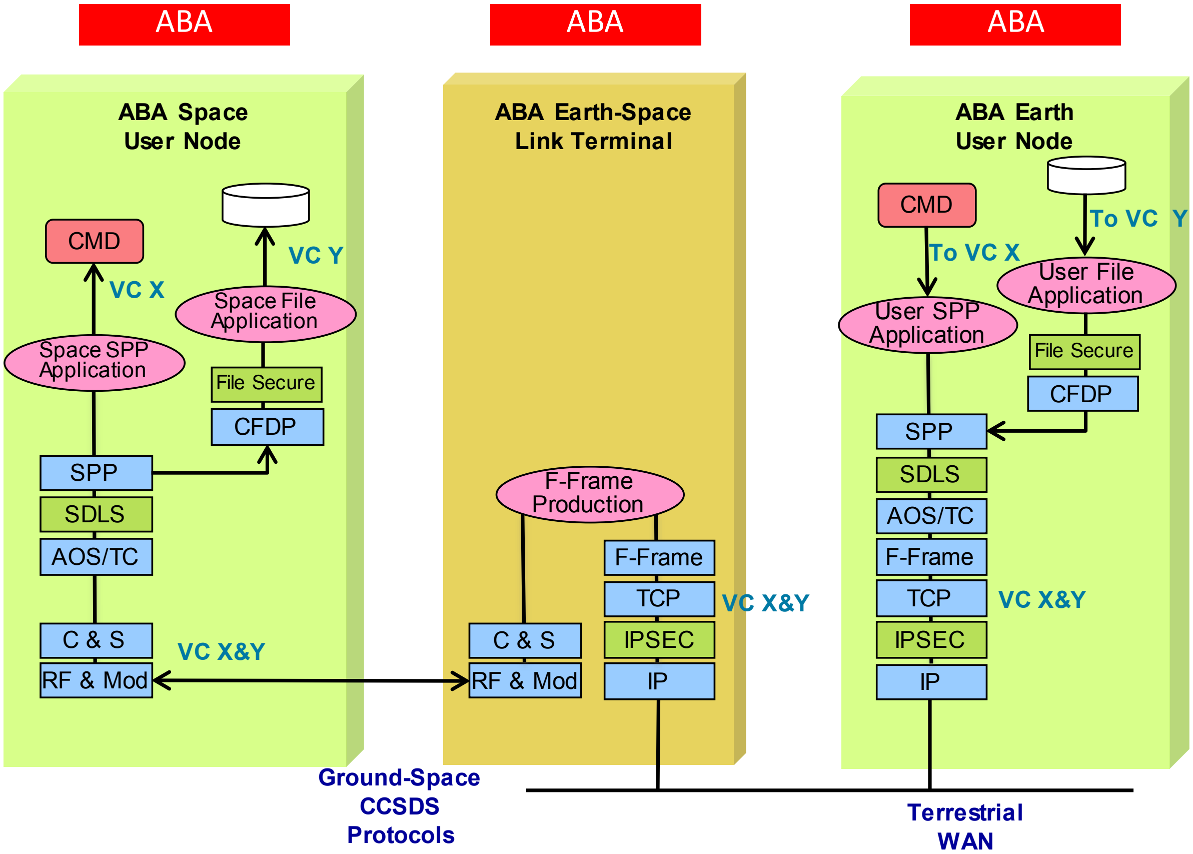


Figure 9-15: ABA Secure End-to-End Forward: ABA Agency Supporting ABA Agency

Figure 9-16 shows three separate, optional alternatives for securing an SSI end-to-end data transfer, using DTN and either AMS or CFDP in this example. The SSI EUN may apply one of the encryption or authentication algorithms to its command data and load that into a file. The secured file provides end-to-end data security. That secured file, or AMS messages as in this example, may be transferred in a BP bundle, which itself may be secured by the Bundle Security Protocol (BSP). On the terrestrial hop, BP/BSP may then use TCP/IP and IPsec to securely transmit the data to the ESLT. The ESLT, which merges data from multiple sources, will re-apply BSP security for the ESLT to SUN hop. Once the file is securely on the space node, the file security may be removed. Here too, as in the ABA example, there is no defined MO service for doing such application data encryption of the data contents at the Network Layer, so this would need to be a bespoke implementation.

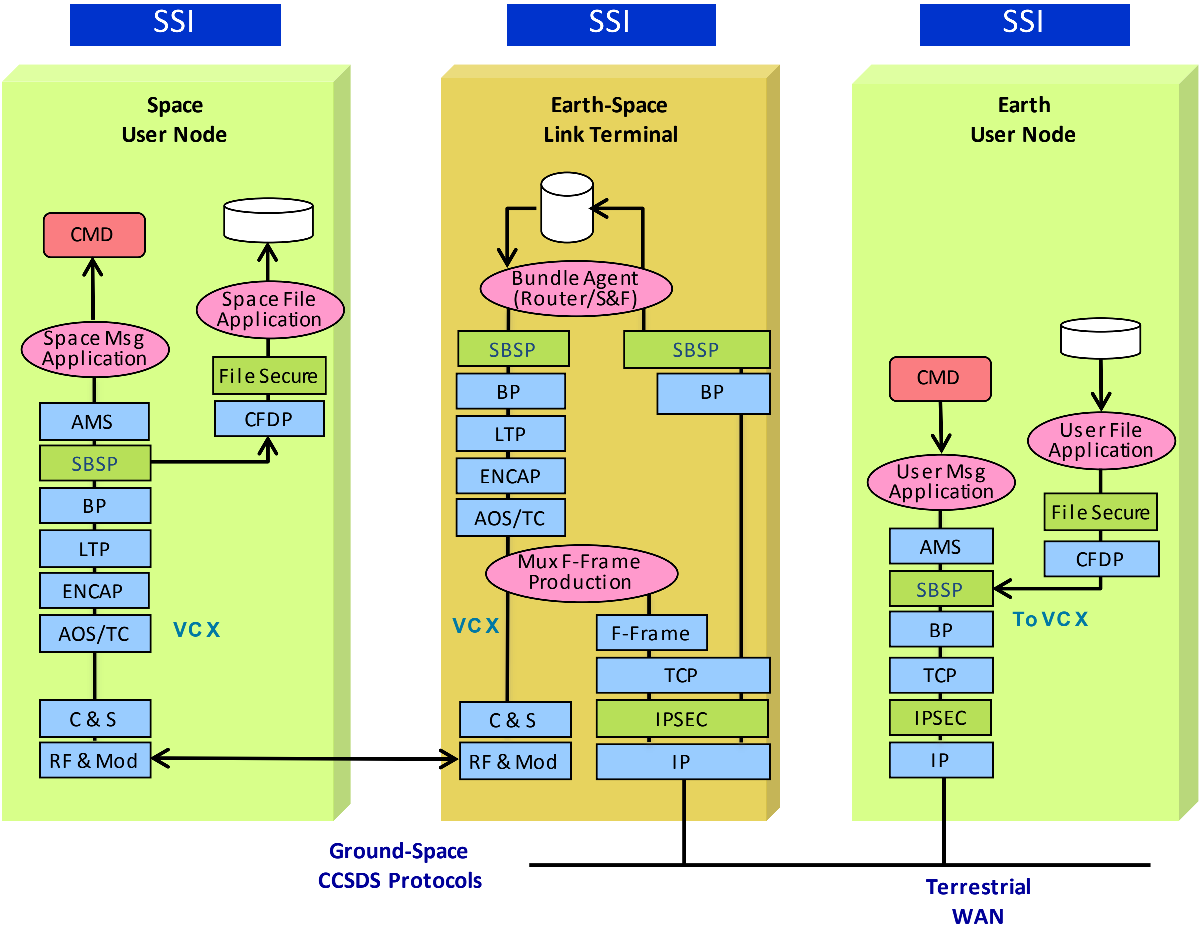


Figure 9-16: SSI Secure End-to-End Forward: SSI Agency Supporting SSI Agency

These same security mechanisms work across multiple SSI hops. The final layers of security are only applied and removed in the end nodes for any given type of deployment.

It will seldom be the case that all of these security measures will be applied in any one deployment, but it is possible. When terrestrial security is essential, IPsec may be applied, as this masks Internet traffic from any snooping. BSP may be employed end-to-end for command security or when service providers that do not belong to the same agency as the user are involved. If a given SUN hosts instruments belonging to different agencies, it might elect to separately secure their command uplinks, or data returns, by use of file-level security. In situations in which web interfaces are used in the ground system, such as in the EUN, it is possible to deploy those using HTTPS to provide security over the terrestrial links.

10 IMPLEMENTATION VIEWPOINT

10.1 INTRODUCTION

In contrast to many of the other CCSDS Recommended Standards, one of the distinguishing characteristics of both MO MAL framework and SO standards, services, and protocols is their support for a rather wide variety of implementation approaches and technology mappings. Most of the CCSDS Recommended Standards are directly prescriptive of the mapping from protocol or service or data abstractions to concrete realizations. These standards will not say 'this is the way to implement X', but they will say 'when you implement X, this is the way we expect the bits to appear on the wire and how the protocols must operate'.

Both the MO framework and SO standards and services tend to deal with an intermediate level of abstraction, and to intentionally use some process for mapping from these abstractions to a variety of different concrete realizations. This brings added complexity, and the possibility of developing multiple concrete realizations that cannot interoperate, but it also allows for a degree of flexibility and portability to other deployment types that have some benefits in terms of portability and 'future-proofing'.

In order to provide an understanding of how this works, in each of these contexts, this Implementation Viewpoint has been provided. This will describe the features, benefits, and potential issues of each of these different capabilities. This is mostly a process viewpoint, rather than any sort of technical viewpoint.

10.2 MO IMPLEMENTATION VIEWS

10.2.1 GENERAL

The MO Implementation Viewpoint is focused on the MO services framework, which provides for the binding of abstract MO service specifications, expressed in terms of the MAL, to underlying technologies for the encoding and transport of MO service messages. It also provides for the transformation of abstract MO service operations into an API for a specific programming language. The tool chains that perform this may be made available by different organizations, but they are not standardized.

The MO Implementation Viewpoint is represented from three related views:

- Information;
- Component;
- Process.

10.2.2 MO INFORMATION VIEW

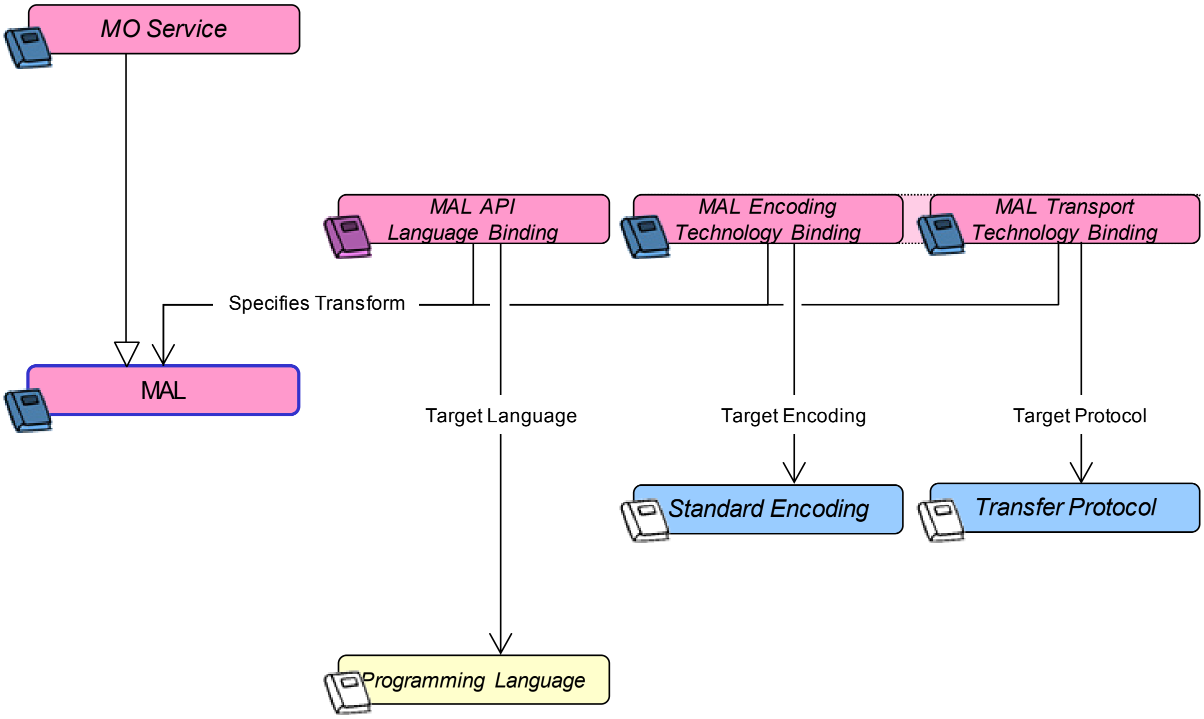


Figure 10-1: MO Implementation Viewpoint—Abstract Information Relationships

Abstract MO service specifications are derived from and expressed in terms of the MAL. (See figure 10-1.)

MAL API Language bindings specify how to implement the API for MO service operations in a specific target programming language. Different language bindings can be used on each side of the service interface between communicating functions as long as the underlying technology mappings are fully interoperable.

MAL Technology bindings specify how to map the MO service operations to specific underlying data representations and communications technologies. There are two parts of any binding that must be mapped: the encoding of the service messages into specific message formats, and the transfer of the service messages using a specific target transfer protocol. These two parts may be combined in a single MAL Technology binding standard, or an encoding and a transport from different binding documents may be selected separately. In either case, both are required to generate a concrete implementation of an MO service . For interoperability between communicating functions, the MAL Technology bindings must match on both sides of the service interface.

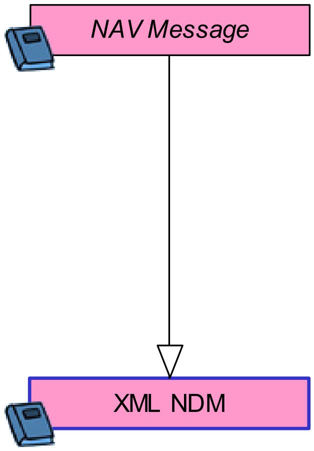


Figure 10-2: MO NAV Implementation Viewpoint—Information

The MO Navigation standards just define data formats, not services, as shown in figure 10-2. For Navigation, the messages are all defined in terms of a common XML-based format called Navigation Data Messages (NDM) (reference [24]). There is no specified transport binding, but message- or file-based transports may be used as desired.

10.2.3 MO COMPONENT VIEW

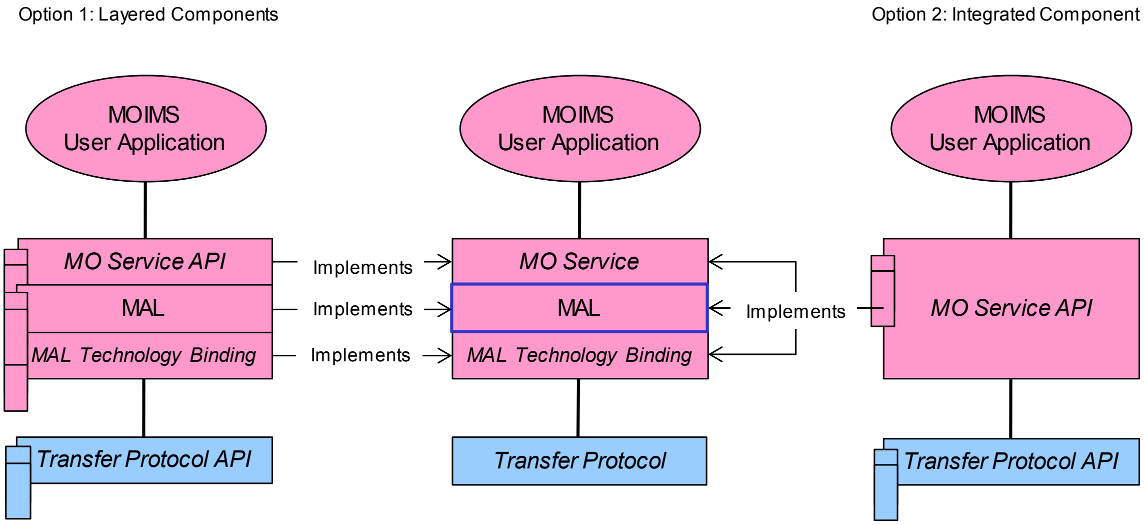


Figure 10-3: MO Implementation Viewpoint—Components

The MO Framework is based upon a layered model, as described in the Communications Viewpoint. The MO user application invokes the operations of a specific MO service . The MO service is defined in terms of the MAL. As shown in figure 10-3, the MAL is mapped to a specific underlying communications-layer transfer protocol through a MAL Technology binding (both for encoding and transport) to effect the transfer of the MO service messages to another MO user application.

The figure 10-3 diagram shows two possible implementation options. The fundamental structures of the MO Framework as shown in the middle, with two options for implementing this shown on either side:

- In Option 1, on the left, each layer of the MO Framework is implemented as a separate software component: the MO service itself, the MAL, and the technology binding to the underlying Transfer Protocol.
- In Option 2, on the right, a single software component implements all three layers. The MO service operations are implemented directly in terms of the underlying Transfer Protocol, in accordance with the embedded technology bindings. While it remains compliant with the abstract MAL specification, there is no separate concrete layer corresponding to the MAL or to the technology binding.

10.2.4 MO PROCESS VIEW

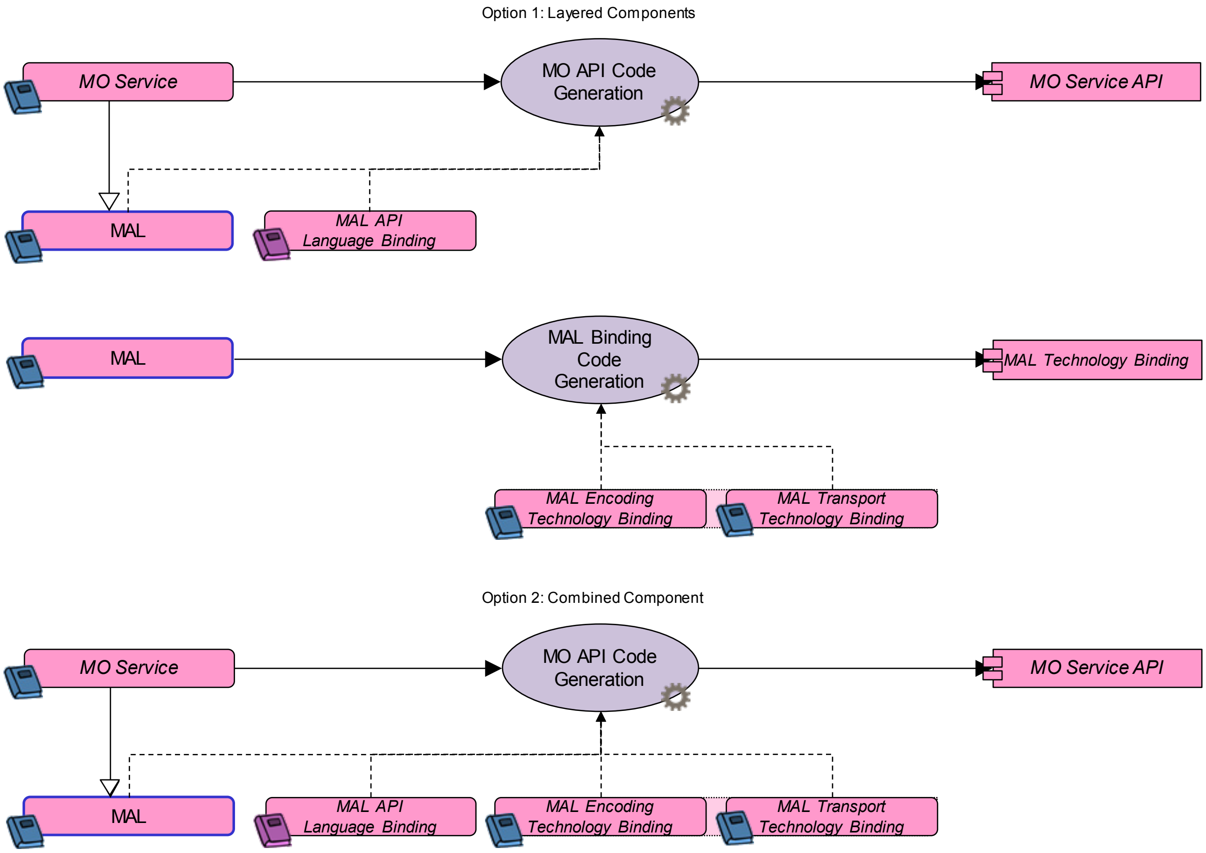

Option 1: Layered Components
MO Service
MO API Code Generation
MO Service API
MAL
MAL API Language Binding
MAL
MAL Binding Code Generation
MAL Technology Binding
MAL Encoding Technology Binding
MAL Transport Technology Binding
Option 2: Combined Component
MO Service
MO API Code Generation
MO Service API
MAL
MAL API Language Binding
MAL Encoding Technology Binding
MAL Transport Technology Binding

Figure 10-4: MO Implementation Viewpoint—Process

The Abstract MO service specifications themselves can be extracted and transformed into concrete MO-compliant software components for a given target language and set of technology bindings. This is done by a code-generation function that can either be a manual process or use autocoding techniques.

Depending on the level of sophistication, autocoding may only generate software stubs and skeletons that require the body of the component to be completed manually, or it can generate the complete software component, as shown in figure 10-4.

For Option 1 (separate software component for each layer):

– The MO service specification is an input to an (unspecified) MO API Code generation function, which also references (or implements) the language transformation contained in a MAL API Language Binding and the abstract MAL specification. The output is the MO service API layer component for the selected language.

– A generic MAL layer software component implements the MAL layer for any MO service .

– The MAL is an input to the MAL Binding Code Generation function, which also references MAL technology bindings for encoding and transport. The output is a MAL Technology binding component for the selected encoding and transport technologies.

For Option 2 (single software component):

– The MO service specification is an input to an (unspecified) MO API Code generation function, which also references the MAL, MAL API Language Binding, and MAL Technology bindings for encoding and transport. The output is the MO service API component for the selected language and target technologies.

10.3 SO IMPLEMENTATION VIEWPOINT

10.3.1 GENERAL

The SO Implementation Viewpoint is focused on the ability of an EDS to describe devices, interfaces, behaviors, and configurations [Future] in a way that can be transformed by suitable tool chains directly into executable code.

Figure 10-5 shows the relationship between the SO Information Model (described in 5.2.11.4) and the SO Functional Model (described in 4.3). The diagram consists of an information model on the left, and onboard system functions on the right. The important things to identify in figure 10-5 are the following:

– SO consists of application support services and subnetwork management services, which are specified by the SO Information Model.

– The SO Information Model includes capabilities to formally model an onboard system in a transformable way, which enables a design-time tool chain to adapt services to the platform architecture chosen for a project.

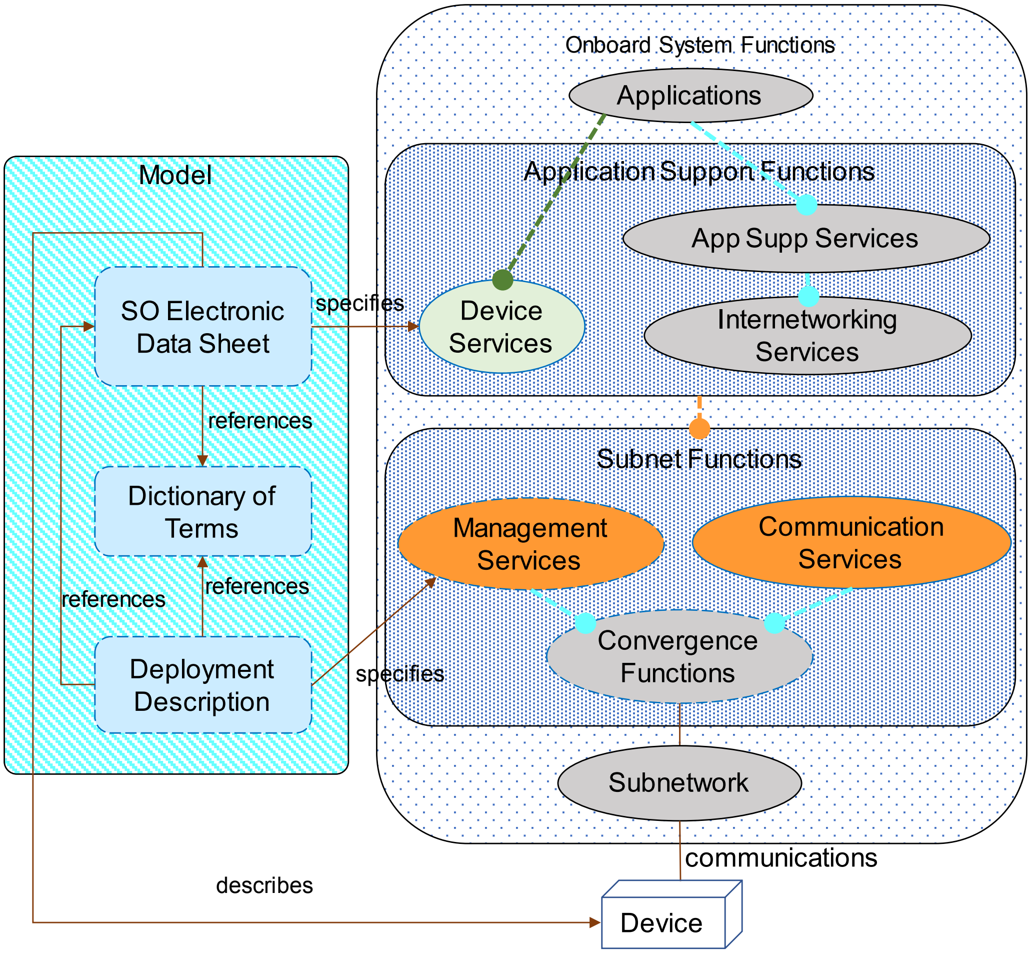


Figure 10-5: Relation between SO Model and SO Functions

This section is structured as follows:

a) **SO EDS-Derived Device Services**: tool chain functions using EDS to generate device services;

b) **Application Support Functions**: tool chain functions using EDS to integrate application support services;

c) **Tool Chain Concepts**: summary of the use of the SO information model in an agency's tool chain for composing a vehicle;

d) **Assembly of Convergence Functions**: description of selection of protocol convergence functions to normalize capabilities across subnetworks.

10.3.2 SO EDS-DERIVED DEVICE SERVICES

The SO abstract services are described in Magenta Books, one for each of the subnetwork functions shown in figure 4-10. There is also a SEDS instance in SANA (https://sanaregistry.org/r/sois) that describes the interfaces of each of these subnetwork functions. In an implementation of SO services that does not use SEDS, the subnetwork functions abstractly described by the Magenta Books may be directly implemented as concrete APIs. In an implementation of SO services that uses SEDS, the subnetwork functions are defined as abstract SAP by the SEDS subnetwork instance in SANA. The SEDS created for various devices use the abstract interface descriptions in the SEDS subnetwork instance to describe interactions with a device through the subnetwork, and a tool chain is used to build concrete implementations of those interactions that work with the concrete subnetwork API of the agency that owns the tool chain. Tool chains may be shared among agencies, but they are not standardized.

The preceding paragraph describes the subnet interfaces required by a SO device service. A device service provides an interface to applications in which the parameters and command arguments of a device are exposed as specific data types. There is a capability to substitute strings in SEDS before schema validation, which can adjust for specific features of missions and of platforms, such as sizes of arrays and word sizes. From this information in a SEDS instance for a device, a tool chain can generate a concrete interface in a suitable programming language for a device service.

How the generation of device services occurs is shown in figure 10-6. An EDS describes the data interfaces of a device. That description includes a specification of the device service using concepts taken from UML, including those listed here.

- Activities represent sequences of actions to be taken by the device service.
- State machines represent responses of the device service to events, such as arrival of packets.
- State machines may invoke activities in the context of states and transitions between states.
- State machines may specify conditions required for transition from one state to another state, such as passage of time since entering the first state.

Spacecraft Onboard Interface Services—XML Specification for Electronic Data Sheets (reference [10]) provides a full account of the descriptive capabilities of activities and state machines.

A tool chain that is specific to the agency responsible for a vehicle generates the software for the device services during development. During flight, the device services communicate with the device through an onboard subnetwork using these generated service interfaces. Onboard applications use the device services for command and data handling.

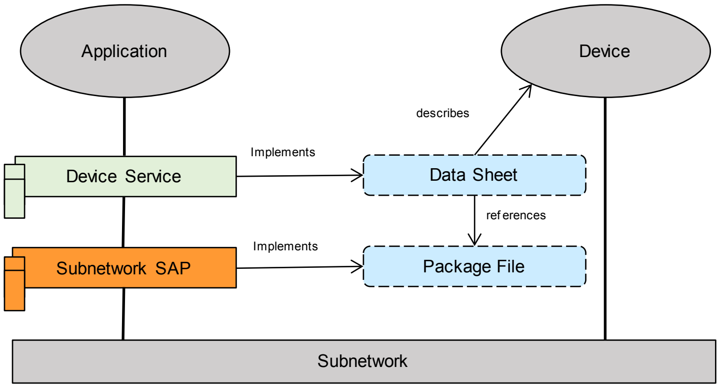


Figure 10-6: Derivation of Device Services from EDS

The Device Service specification in an EDS has two parts that correspond to the SO DSAP and DACP services. The DSAP presents the raw device data interface, in which the data elements are often counts from analog-to-digital converters. The DACP, at a minimum, restates the data elements in engineering units. The DACP is optional onboard because many designers of spacecraft prefer to work with the raw counts. The DACP information is needed for presentation of engineering units in a mission control center.

When mission control concepts appear onboard as a proxy that converts mission control commands into device commands, then the DACP is needed onboard. Not only does DACP convert from engineering units to units expected by a device, but the DACP also contains information for conversion from succinct mission control commands to more specific commands needed by a device. For example, it may be necessary to send one command to prepare a device for a subsequent command, with a constrained delay between those two commands, and the two commands together implement a single succinct command from the mission control center.

10.3.3 SO APPLICATION SUPPORT FUNCTIONS

Application support functions can be integrated into a vehicle by means of their descriptions in EDSs, as shown in figure 10-7.

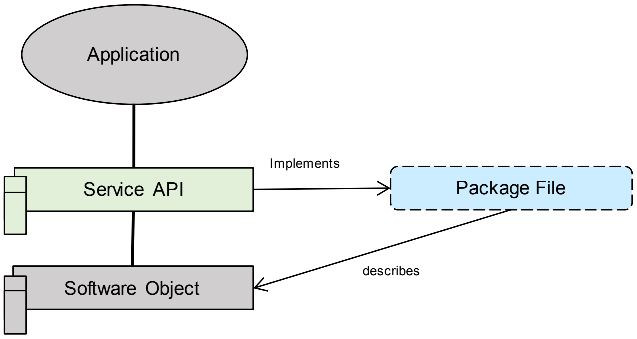


Figure 10-7: Integration of Application Support Functions

As in the case of a device, an EDS can be used to describe the data interface of an application support service. Mission- and platform-specific EDSs describe features and requirements that are relevant to a particular vehicle, such as word sizes, array sizes, and interface bindings. A tool chain uses that description to tailor the interface of the application support service. Examples of interface bindings might be an application programming interface or a message bus.

The tailoring may take the form of generating a wrapper for a concrete interface, or it may alter parameters in a C language header file (for example) before compilation. The former is appropriate for a software object distributed as a library or script; the latter is appropriate for a software object distributed as source code. In the former case, the tailoring fits the model of a language binding. In the latter case, the tailoring results in software appropriate to a specific platform used in a specific mission.

10.3.4 SO SYSTEM MODEL

This section describes how the SO System Model can work with a tool chain to configure the flight software for a vehicle.

The general process to integrate a device into a vehicle appears in figure 10-8. A SEDS describing the device is the primary input to the process; its base element is a Datasheet, which contains both device and Namespace sections. There may be other SEDS that define common data elements that have shared syntax, such as CCSDS packet headers. One or more SEDS shared across the mission provide metadata that represents design choices for platforms and for the mission as a whole. The SEDS for shared syntax and for shared mission metadata have a PackageFile for their base element. The output for the tool chain is executable software artifacts in a particular programming language for execution on a particular platform. Figure 10-8 shows this process.

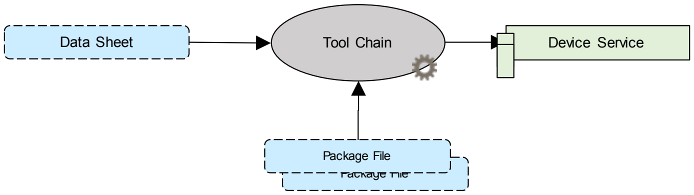


Figure 10-8: A Tool Chain Integrates a Device into Flight Software

The general process to use a tool chain to compose the description of an entire vehicle appears in figure 10-9 [Future]. The input to the process includes a collection of SEDS describing the devices that compose the vehicle. A deployment description describes topology and scheduling of the devices in onboard subnetworks. The union of the sets of shared SEDS described in figure 10-8 provide common syntax. The same SEDS shared across the mission in figure 10-8 provide metadata that represents design choices for platforms and for the mission as a whole. The output is a number of artifacts for the vehicle, including a mission operations data base, network utilization schedules, flight software files, and an integration and test schedule. Initial work has been done to demonstrate this capability, and a description will soon be available in an Orange Book.

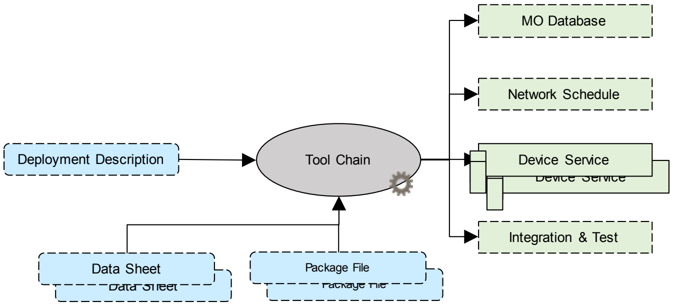


Figure 10-9: A Tool Chain Composes a Vehicle [Prospective]

A more concrete example of tool chain activities to compose a software bus appears in figure 10-10. In this example, the software bus is implemented in shared memory using queues provided by the operating system. The packets on the bus use the format specified by CCSDS Space Packet Protocol. The APID in each packet identifies its topic in a publish/subscribe design-pattern. In this example, the tool chain is generating software suitable for the NASA core Flight Software (cFS) bus, but the same collection of SEDS could be used by another agency's tool chain to generate software suitable for a CCSDS Asynchronous Messaging Service, with some differences in the platform SEDS. For cFS, the deployment description generates the routing table that maps APIDs to destination applications, along with queues and buffers. The device SEDS populate the device manager database. The deployment description also provides the schedule for the 1553 bus.

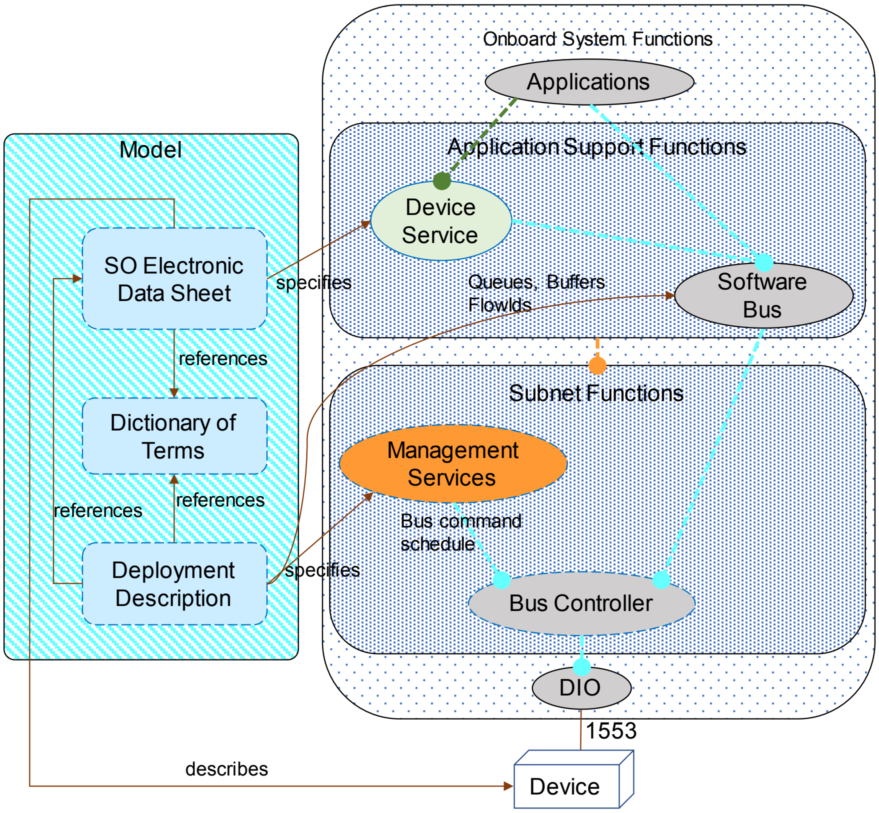


Figure 10-10: Composing a Software Bus

Figure 10-11 extends the example in figure 10-10 to describe the configuration of the management information for the SO Subnet Services collection.

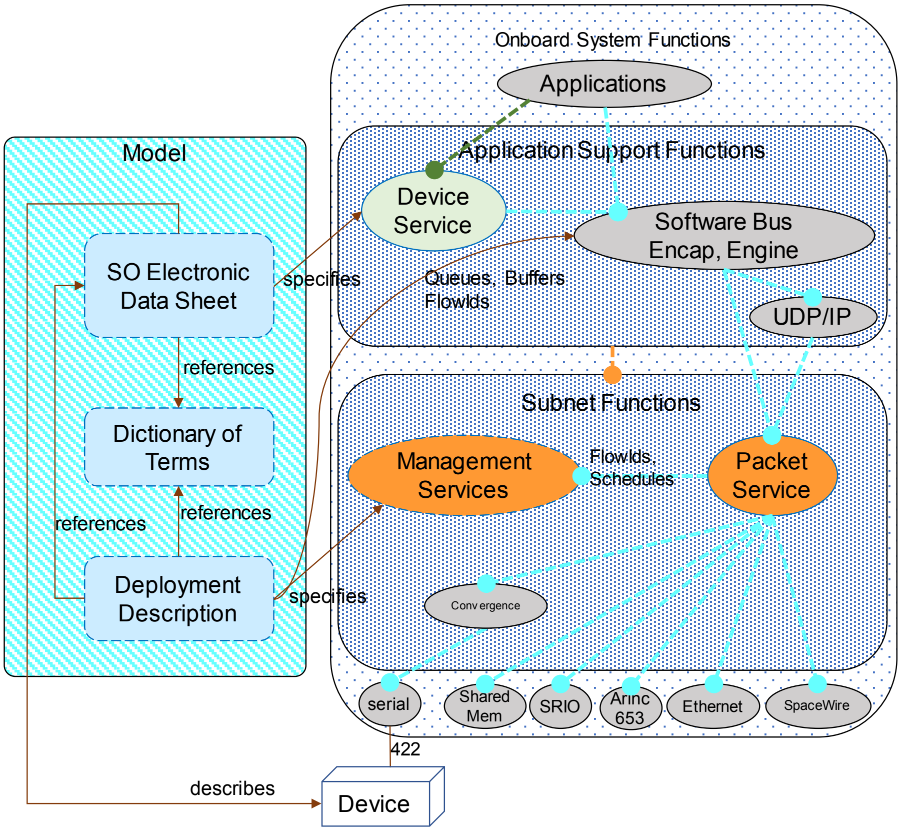


Figure 10-11: Configuring Management Data for SO Subnet Services

Figure 10-12 is a SO business process diagram, which summarizes the usage of SEDS and tool chains to compose flight software for a mission. It includes a feedback mechanism to maintain the content of the DoT. This diagram also identifies the roles of the human participants in the process:

- mission design team;
- mission control team;
- manufacturer;
- ontology team.

The processes appear as beige blocks; the data artifacts appear as green blocks.

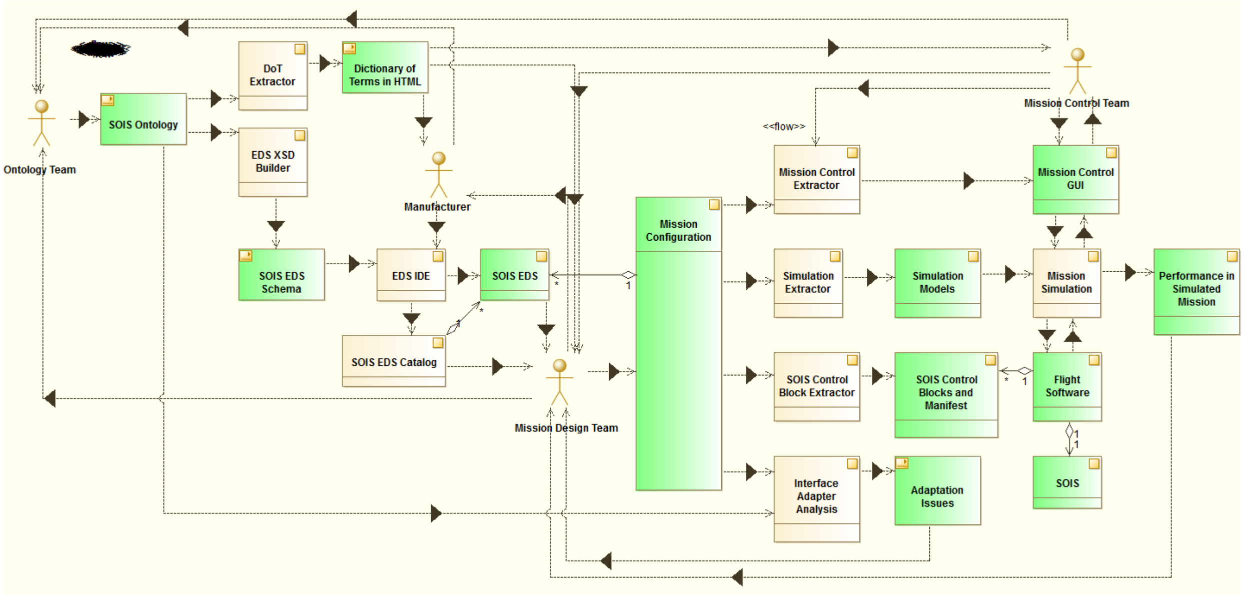


Figure 10-12: Flow of Information in SO Tool Chain

10.3.5 ASSEMBLY OF SO PROTOCOL CONVERGENCE FUNCTIONS

The SO protocol convergence functions are intended to provide uniformity of protocol services across a variety of subnetwork protocols, which might themselves each lack some of the required convergence functions.

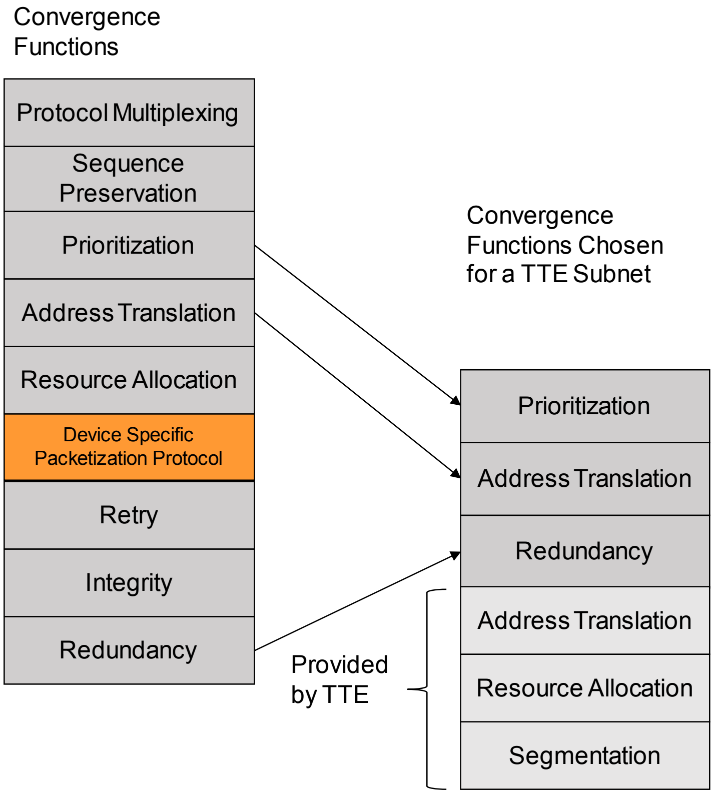


Figure 10-13: Flow of Information in SO Tool Chain

By describing convergence functions in SEDS, a tool chain can match interfaces and provide shims, as in figure 10-13. For example, Time-Triggered Ethernet (TTE) puts some functions of UDP/IP into the subnetwork chips in order to fragment and to reassemble packets in hardware. These networking functions would normally appear in a layer above the Data Link Layer, and not within a classic ‘subnet’, which is usually considered to implement only link-layer functions.

In figure 10-13, some additional SO convergence functions have been selected to complete the stack above what is offered by the TTE chip. Address translation in the SO protocol stack is between spacecraft network addresses and Internet Protocol (IP) addresses. Address translation by TTE is between IP and MAC address.

10.4 SECURITY CONCEPTS FOR IMPLEMENTATION VIEWPOINT

10.4.1 GENERAL

The Implementation Viewpoint has some different security concerns than those related to the other viewpoints. These have to do with the security of the systems and implementation processes, as well as the supply chains for software and parts. A discussion of all of these potential issues and vulnerabilities is outside the scope of this document, but some of these are discussed in *Security Threats Against Space Missions* (reference [71]). There are, however, three aspects of security relating to the implementation space data software. These are:

a) security of formalized spacecraft information models;

b) security of tool chains and related intellectual property;

c) security of the software development process itself.

10.4.2 SECURITY OF FORMALIZED SPACECRAFT INFORMATION MODELS

To the extent that a formalized model of spacecraft provides an avenue of attack, it should be treated as a vulnerability in the process of system development. As such it should be subject to at least the level of security as other spacecraft-command-related artefacts, using techniques described in *Security Architecture for Space Data Systems* (reference [38]) or *CCSDS Cryptographic Algorithms* (reference [39]), or *Security Threats Against Space Missions* (reference [71]). A discussion of all of these potential issues and vulnerabilities is outside the scope of this document.

10.4.3 SECURITY OF TOOL CHAINS AND RELATED INTELLECTUAL PROPERTY

Any tool chains that are developed to transform formal specifications, of any sort, into executable code represent a concrete investment on behalf of the organization that creates it. As such, it may be viewed as valuable intellectual property that should be protected. On the other hand, organizations may see the distribution of such tools as an important community service and one that should be widely supported. A discussion of the issues and vulnerabilities associated with tool chains and software development processes is outside the scope of this document.

10.4.4 SECURITY OF THE SOFTWARE DEVELOPMENT PROCESS ITSELF

The software development process for embedded real-time systems is itself a potential vulnerability. Malicious code can be inserted during the development process in order to provide a back door, or other vulnerability, in the control system. A discussion of the potential issues, risks, and vulnerabilities related to software development processes is outside the scope of this document but is briefly addressed in *Security Threats Against Space Missions* (CCSDS 350.1-G-2, reference [71]).

ANNEX A

ABBREVIATIONS AND ACRONYMS

Term	Definition
ABA	Agency A (spacecraft) – Agency B (ESLT) – Agency A (MOC)
ADM	Attitude Data Message
AEM	Attitude Ephemeris Message
AIP	Archival Information Package
AIV	assembly, integration, and verification
AOS	Advanced Orbiting Systems [AOS Space Data Link Protocol]
APD	automated procedure definition
APID	application process identifier
APM	Attitude Parameter Message
ASL	application and support layer
AUT	MO Automation
BP	Bundle Protocol
BSP	Bundle Security Protocol
CAIS	Consumer Archive Interface Specification
CAR	Common Archive [MO COM Archive Service data]
C&DH	command and data handling
CDM	Conjunction Data Message
CFG	Configuration Service (MO Common Service)
cFS	core Flight Software
COM	Common Object Model
CS	cross support
CSD	common services data
CSS	Cross Support Service
CSTS	CSS Transfer Service
DAC	data archive center
DACP	Device Abstraction Control Procedure
DAI	Data Archive Interoperability
DSAP	Device-Specific Access Protocol
DEDSL	Data Entity Dictionary Specification Language
DIP	Dissemination Information Package
DoT	dictionary of terms

Term	Definition
DMZ	demilitarized zone
DPC	data processing center
DSA	data storage and archiving
DVSACP	Device Virtualization Service Abstraction Control Procedure
EAST	Enhanced ADA Subset (data definition language)
ESLT	Earth-space link terminal
EUN	space user node
FCLTU	Forward CLTU (SLE)
FDIR	fault detection, isolation, and recovery
FPSS	File and Packet Store Service
FSW	flight software
FTM	file transfer and management
GNSS	global navigation satellite system
HRT	Hard Real-Time
IOAG	Interagency Operations Advisory Group
ICD	interface control document
IP	1) Internet Protocol; 2) Information Package
ISL	Inter-Satellite Links
LAC	Login and Authentication Credentials
M&C	Monitor & Control
MAL	Message Abstraction Layer
MCS	Mission Control System
MD-CSTS	CSTS Monitored Data Service
MDP	Mission Data Product
MO	Mission Operations
MOC	mission operations center
MO	Mission Operations and Information Management Services
MOS	Mission Operations Service
MPS	Mission Planning and Scheduling
NAV	Navigation
NAVT	Navigation and Timing
NDM	Navigation Data Message
NEM	Navigation Events Message
NOC	network operations center
NSC	navigation services center
OAIS	Open Archival Information System

Term	Definition
OBCP	onboard control procedure
OBPM	Onboard Procedure Management
OBSM	Onboard Software Management
OBSW	onboard software
ODM	Orbit Data Message
OEM	Orbit Ephemeris Message
OMM	Orbit Mean-Elements Message
OPD	operations preparation data
OPM	Orbit Parameter Message
OSM	onboard software management
PACT	Planning Activity
PAIS	Producer Archive Interface Specification
PDA	payload data acquisition
PDB	planning database
PDCP	packet data convergence protocol
PDD	planning data definition
PDI	Preservation Description Information
PDS	Plan Distribution Service
PEC	Plan Execution Control Service
PED	Plan Edit Service
PEVT	Planning Event
PI	principal investigator
PIM	Plan Information Management Service
PLN	Plan
POC	payload operations center
PRES	Planning Resource
PRM	Pointing Request Message
PRQ	planning request
PRS	Planning Request Service
PVL	Parameter Value Language
RAF	Return All-Frames (SLE)
RCF	Return Channel Frames (SLE)
RASDS	Reference Architecture for Space Data Systems
RDM	Re-entry Data Message
RLC	radio link control
ROC	rover operations center

Term	Definition
RT	real time
RTLT	round trip light time
SA	situational awareness
SAP	service access point
SC-CSTS	CSTS Service Control
SCCS	Space Communications Cross Support
SCCS-ADD	Space Communications Cross Support Architecture Description Document
SCCS-ARD	Space Communications Cross Support Architecture Requirements Document
SCM	spacecraft manufacturer
SDB	spacecraft database
SDIR	service directory
SEA	Systems Engineering Area
SEDS	SOIS Electronic Data Sheet(s)
SGL	space-ground link
SI	Space Internetworking Service
SIP	Submission Information Package
SIS	Space Internetworking Service
SISG	Space Internetworking Strategy Group
SLE	Space Link Extension
SLS	Space Link Service
SM	Service Management
SM&C	CCSDS Spacecraft Monitoring & Control Working Group
SMES	Service Management Event Sequence
SOC	Science Operations Center
SOIS	Spacecraft Onboard Interface Services
SPP	Space Packet Protocol
SSA	Space Situational Awareness
SSI	Solar System Internet
SST	Space Surveillance and Tracking
SUN	space user node
TCM	Time Correlation Message
TCP	Transmission Control Protocol (RFC)
TDM	Tracking Data Message
TIM	Technical Interchange Meeting
TRM	Time Reception Message
TS	Transfer Service (SLE)

Term	Definition
TSP	time and space partitioned
TT&C	telemetry, tracking, and command
TTE	time-triggered Ethernet
UDP	User Datagram Protocol (RFC)
USLP	Unified Space Data Link Protocol
XTCE	XML Telemetric and Command Exchange

ANNEX B

FUNCTIONAL VIEWPOINT ALTERNATIVE STYLE DIAGRAMS

The following MO Functional Viewpoint diagrams are equivalent to those contained in section 4 for each of the MO functional groups:

– Mission Control;

– Navigation and Timing;

– Mission Planning and Scheduling;

– Operations Preparation;

– Data Storage and Archiving.

They show an alternative representation (see 3.3.2, figure 3-5), which puts all the connections that share a common interface type at the center of the picture as a set of horizontal color-coded ‘tramlines’. Functions are then attached to these with vertical lines, distinguishing between service provider and service consumer as before. This approach makes it easier to see which functions are the providers and consumers of each service.

Color coding is by MO functional group and is consistent with that introduced in figure 3-2. Acronyms used for information objects (data) and services are consistent with those used in 4.2 and can be found in annex A.

B1 MISSION CONTROL

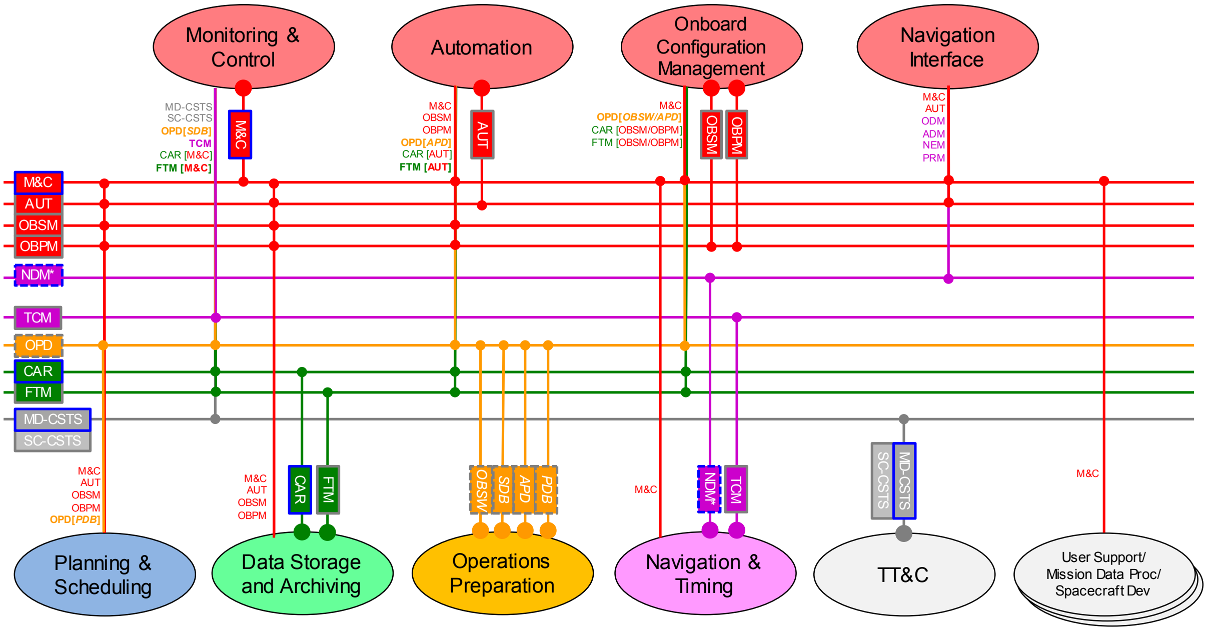


Figure B-1: MO Level 2: Mission Control Functions (Alternative View)

B2 NAVIGATION AND TIMING

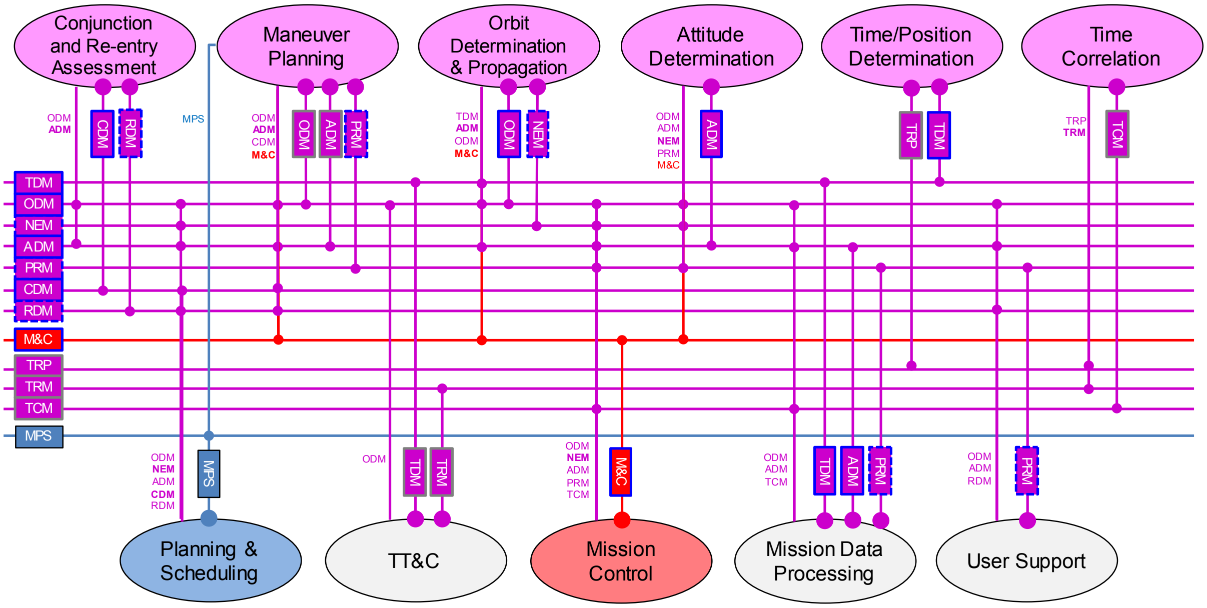


Figure B-2: MO Level 2: Navigation and Timing Functions (Alternative View)

B3 MISSION PLANNING AND SCHEDULING

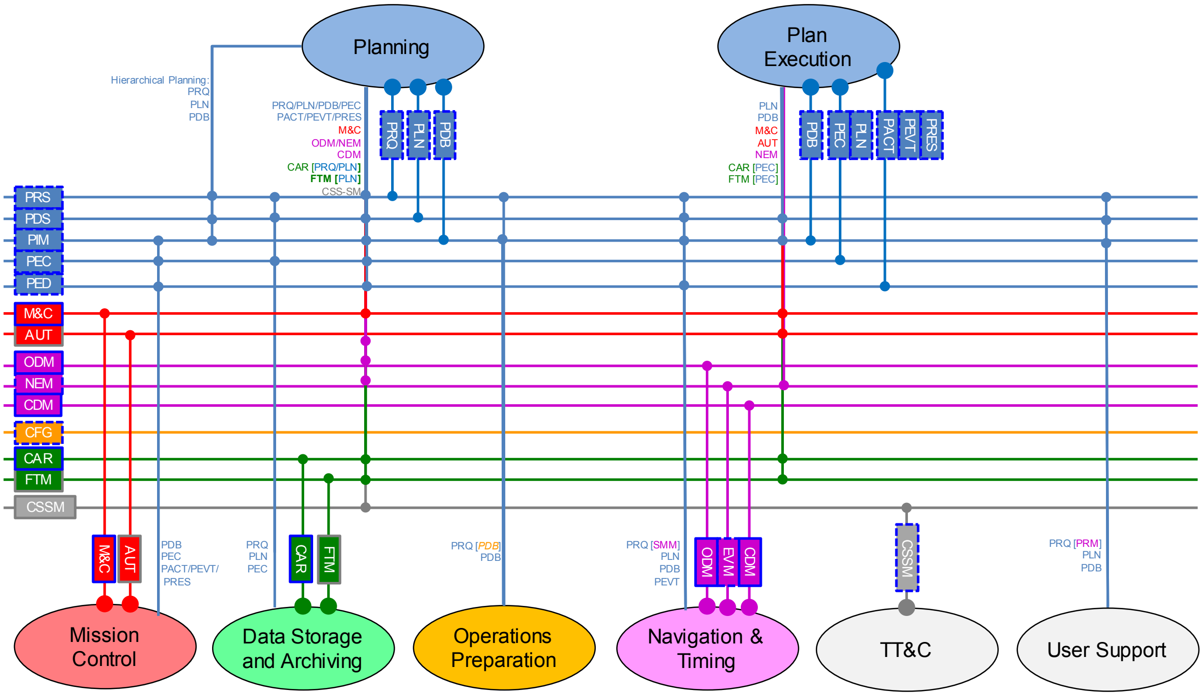


Figure B-3: MO Level 2: Mission Planning and Scheduling Functions (Alternative View)

B4 OPERATIONS PREPARATION

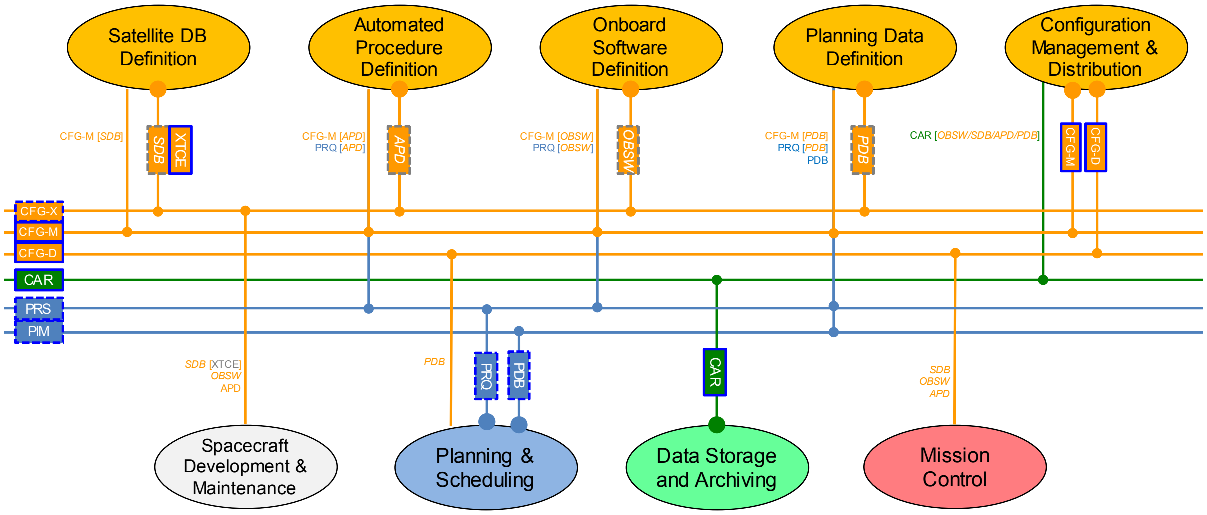


Figure B-4: MO Level 2: Operations Preparation Functions (Alternative View)

CFG-M corresponds to configuration management operations of the MO Common Configuration service (Add/Remove/Activate). CFG-D corresponds to configuration distribution operations of the MO Common Configuration service (List/Get). CFG-X corresponds to non-standardized configuration exchange.

B5 DATA STORAGE AND ARCHIVING

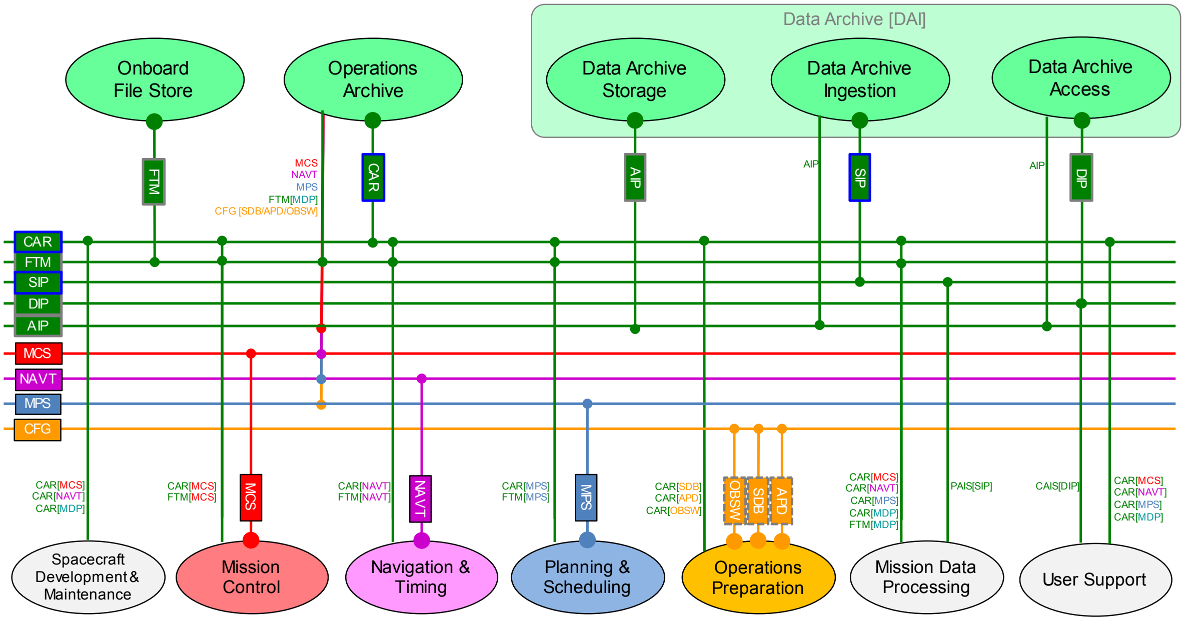


Figure B-5: MO Level 2: Data Storage and Archiving Functions (Alternative View)