Consultative Committee for Space Data Systems

DRAFT RECOMMENDATION FOR SPACE DATA SYSTEM STANDARDS

REFERENCE ARCHITECTURE FOR SPACE DATA SYSTEMS

CCSDS 800.0-R-1

RED BOOK

August 2004

AUTHORITY

Issue:	Red Book, Issue 1
Date:	August 2004
Location:	Pasadena, CA

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

This Recommendation is published and maintained by:

CCSDS Secretariat
Office of Space Communication (Code M-3)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

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

This **Recommendation** is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this **Recommendation** is voluntary. Endorsement, however, indicates the following understandings:

– Whenever an Agency establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommendation**. Establishing such a **standard** does not preclude other provisions, which an Agency may develop.

– Whenever an Agency establishes a CCSDS-related standard, the Agency will provide other CCSDS member Agencies with the following information:

  - The **standard** itself.
  - The anticipated date of initial operational capability.
  - The anticipated duration of operational service.

– Specific service arrangements are made via memoranda of agreement. Neither this Recommendation nor any ensuing standard is a substitute for a memorandum of agreement.

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

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

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommendation is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. 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 addressed to the CCSDS Secretariat at the 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.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency(JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- Russian Federal Space Agency (FSA)/Russian Federation.

Observer Agencies

- Australian Space Office (ASO)/Australia.
- Austrian Space Agency (ASA)/Austria.
- Belgian Science Policy Office (SPO)/Belgium.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Industry Canada/Communications Research Centre (CRC)/Canada.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status/Remarks
CCSDS 800.0-R-1	Reference Architecture for Space Data Systems	August 2004	Red Book, Issue 1

CONTENTS

CONTENTS

Section Page

CONTENTS (continued)

1 INTRODUCTION

1.1 SCOPE

This document presents a Reference Architecture for Space Data Systems (RASDS). The RASDS is primarily intended to provide a normative approach for description of data system architectures and high level designs within CCSDS. However, it is also suitable for use by mission and project design teams, to describe system architectures and designs within the space domain.

In reading this document please keep in mind that this is a Reference Architecture, not a set of blueprints for any given system architecture or even for developing specific standards architectures. It provides a required methodology for describing a variety of different space data system architectures within CCSDS. It includes some examples for illustrative purposes, but none of these examples or subsets of elements is complete.

NOTE – RASDS is based on the Reference Model for Open Distributed Processing (RM-ODP) that is defined in the international standard ISO/IEC 10746. However, since RM-ODP was mainly developed for terrestrial homogeneous distributed systems that consist of computers connected via the Internet and LANs, and the RASDS must deal with systems flying in space, it was necessary to make some modifications to it. Further, RASDS uses slightly different terminology from RM-ODP in order to be compatible with existing CCSDS recommendations. Specific differences between RASDS and RM-ODP are described in NOTES in the following sections.

NOTE – RASDS is also fully compliant with IEEE-P1471-2000, the Recommended Practice for Architectural Description of Software Intensive Systems.

1.2 PURPOSE

RASDS will be used for the following purposes:

a) to establish an overall CCSDS approach for defining and developing domain specific architectures;

b) to define a common language, taxonomy and representation so that challenges, requirements, and solutions in the area of space data systems can be readily communicated;

c) to provide a kit of architect’s tools so that domain experts will use it to construct different specific complex space system architectures;

d) to facilitate development of CCSDS recommendations in a consistent way so that any standard can be used with other appropriate standards in a space data system;

e) to provide a framework for presenting the recommended standards developed by CCSDS in a systematic way so that their functionality, applicability, interrelationships and interoperability may be clearly understood.

1.3 APPLICABILITY

The methodology described in this document shall be used for the description of all data system architectures in any relevant CCSDS documents. The Viewpoints and representational methodologies shall be used where applicable, but only those Viewpoints that are needed for the specific purpose are required. The methodology described in this document may be used in the description of other space data system architectures, but its use is not required.

1.4 DOCUMENT STRUCTURE

Section 2 provides an overview of RASDS and introduces Viewpoints and Views.

Section 3 introduces basic concepts and terms that are used throughout this document.

From section 4 through 8, RASDS is presented in detail for each of the Viewpoints introduced in section 2.

1.5 REFERENCES

[1] Reference Method for Open Distributed Processing (RM-ODP), ISO/IEC 10746.

[2] Recommended Practice for Architectural Description of Software Intensive Systems, IEEE P1471-2000.

[3] Standard for Application and Management of the System Engineering Process, IEEE 1220-1998.

[4] CCSDS Security Architecture White Paper, reference TBD.

[5] CCSDS Information Architecture White Paper, reference TBD.

[6] Unified Modeling Language (UML) v1.5, OMG formal/03-03-01, http://www.omg.org.

[7] Unified Modeling Language (UML) v2.0 Infrastructure, Final Adopted Specification, OMG ptc/03-09-15, http://www.omg.org

[8] Meta Object Facility (MOF) v 2.0 Core, Final Adopted Specification, OMG ptc/03-10-04, http://www.omg.org

[9] Systems Modeling Language (SysML) Specification (draft) v 0.7, SysML Partners SysML-v0.7-PDF-040705.pdf, http://www.omg.org.

[10] Unified Modeling Language (UML) for Systems Engineering (SE) RFP (requirements), OMG ad-03-03-41.doc, http://www.omg.org.

[11] DoD Architecture Framework (DoDAF), Version 1.0, Deskbook, Vol I, Vol II, US Department of Defense, http://www.aitcnet.org/dodfw/.

[12] The Open Group Architecture Framework (TOGAF), Version 8.1, www.opengroup.org/architecture/togaf/.

[13] XML Architecture Description Language (xADL) v 2.0, http://www.isr.uci.edu/projects/xarchuc

2 OVERVIEW

2.1 VIEWPOINTS

Since space data systems have many different aspects and it is not easy to depict these aspects in a single framework, the architecture of a space data system is described with multiple Viewpoints, each focusing on different concerns associated with the system. The RASDS reference architecture defines multiple Viewpoints to present architectures of space data systems.

A Viewpoint is 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 establishes the purpose and audience for a view and the techniques or methods employed in constructing a view.

Each of these Viewpoints is intended to be orthogonal, but there are some specific areas of overlap that allow the viewpoints to be related. Each exposes a different set of design concerns and issues, and each provides the means for reasoning about that aspect of the system.

Each of the Viewpoints describes the space data system in question as a set of Objects and interactions among them. An Object is an abstract model of an entity in the system.

RASDS defines five Viewpoints to describe the architecture of space data systems, which are explained in the following subsections. The user may decide not to use all of these five Viewpoints to describe a particular space system if the system can be characterized with less than five Viewpoints.

A View is 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 usually intended to correspond to exactly one Viewpoint. In RASDS some new combined Views will be defined that correspond to more than one Viewpoint. The user may also define a new View based on the basic concepts defined in Section 3 of this document if it is impossible to capture all the important aspects of the system with the five Viewpoints defined here. Some aspects of a system design may benefit from being examined from two or more views simultaneously and some examples of this are provided herein.

NOTE – The following Viewpoints are derived from RM-ODP Viewpoints (with some modifications), but the Physical Viewpoint and the Connectivity View were newly created in order to present issues and constraints derived from the physical environment of space data systems, which are distinct from those encountered in typical terrestrial distributed systems. Challenges from the physical environment in which these systems operate, particularly the motion, discontinuous connectivity, and extremely distant and broad distribution of physical devices, require specialized protocols and systems design. The RASDS approach also has its focus on architecture rather than implementation. Many aspects of the Engineering and Technology Views in RM-ODP are treated by RASDS in the Physical and Communications Viewpoints. For those aspects of these views that are not treated in RASDS, the RM-ODP approaches may be directly used.

NOTE – Security issues relevant for each viewpoint will be addressed in passing, but detailed explanations of security issues and approaches will be treated separately in the Security Architecture and related documents.

2.2 ENTERPRISE VIEWPOINT

The motivation for the Enterprise Viewpoint is that we have complex organizational relationships involving spacecraft, instruments, ground systems, scientists, staff, and contractors that are distributed among multiple organizations (space agencies, science institutes, companies, etc.). The Enterprise Viewpoint is used to address these aspects of space data systems.

The Enterprise Viewpoint describes the organizations involved in a space data system and the relationships and interactions among them. The relationships among the organizations are described in terms of the roles; responsibilities and policies of the organizations, and the interactions among the organizations are described in terms of agreements and contracts.

The Enterprise Viewpoint also includes other organizational concerns, such as Requirements, Use Cases, and other aspects of the stakeholder views of the system. Stakeholders may include: developers, designers, operators, maintainers, sub-contractors, or users, as well as the architects and system engineers themselves.

The Enterprise Viewpoint is depicted as a set of Enterprise Objects and interactions among them. An Enterprise Object is an abstract model of an organization (or facility) involved in a space data system. An Enterprise Object represents an independent Enterprise (such as a space agency, a government institute, a university, or a private company) or a department or a center of an Enterprise (such as a tracking network, a control center, a science center, or a research group). An Enterprise Object may be composed of other Enterprise Objects. A group of Enterprise Objects that plays some role in a space data system (such as a community, a committee, or a joint project) can also be an Enterprise Object.

Each viewpoint has a specific set of concerns that are addressed. For the Enterprise Viewpoint these include things like: Objectives, Roles, Policies, Activities, Lifecycles/Phases, Configuration, Contracts, Requirements, and Use Cases.

A simple example of the Enterprise Viewpoint is shown in figure 2-1, in which two Enterprise Objects (Enterprises P and Q) are shown as dotted boxes.

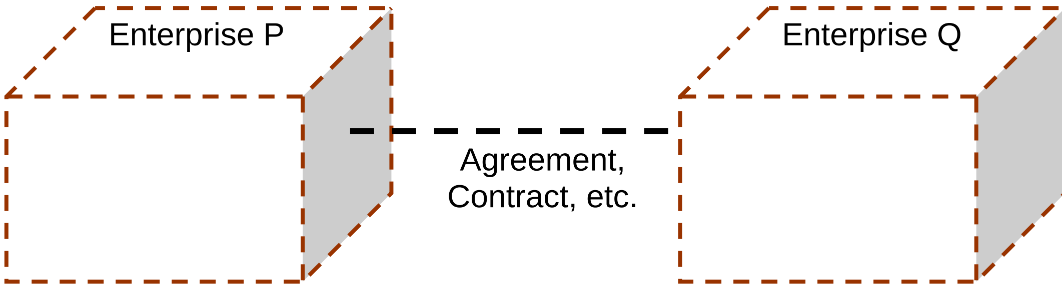

Figure 2-1: Simple Example of Enterprise Viewpoint

2.3 PHYSICAL VIEWPOINT / CONNECTIVITY VIEW

The motivation for the Physical Viewpoint is that space data systems are made up of elements that must operate in physical space, and where the physics of motion, structure, power and connectivity must be considered. For analysis of Space Systems all of these aspects, and the power, thermal, structural views associated with them, must be considered. For the description of Space ***Data*** Systems, we focus just on the Connectivity View of the Physical Viewpoint.

The motivation for the Connectivity View is that we have system elements that are in motion through space and consequently connectivity issues associated with pointing, scheduling, long round-trip light times, and low signal-to-noise ratios, all of which require special protocols and functionality to deal with. The Connectivity View is used to address these aspects of space data systems.

The Connectivity View describes the physical structure and physical environments of a space data system.

The Connectivity View is depicted as a set of Nodes and Links. A Node is an abstract model of a physical entity used in a space data system, which is operating in a physical environment. A Node represents a system (such as a spacecraft, a tracking system or a control system) or an individual physical element of a system (such as an instrument, a computer, or a piece of 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).

For the Connectivity View the Concerns include: Distribution, Communication, Physical Environment, Behavior, Performance, Constraints, and Configuration.

A simple example of the Connectivity View is shown in figure 2-2, in which three Nodes are shown as boxes and two Links are shown as solid lines.

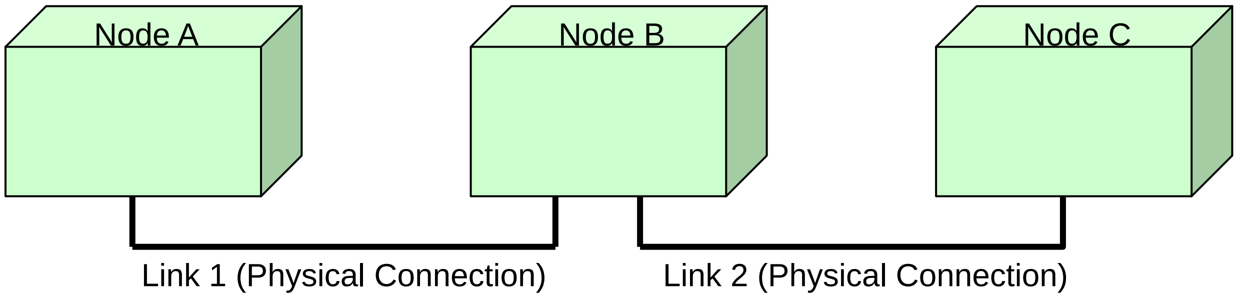

Figure 2-2: Simple Example of Connectivity View

2.4 FUNCTIONAL VIEWPOINT

The motivation for the Functional Viewpoint is that we have functional elements and their logical interactions that may be considered separately from the engineering concerns of where functions are housed, how they are connected, which protocols are used, or what language is used to implement them. The Functional Viewpoint is used to address these functional aspects of space data systems.

The Functional Viewpoint describes the functional structure of a space data system and how functions interact with each other.

The Functional Viewpoint is depicted as a set of Functional Objects and interactions among them. A Functional Object is an abstract model of a functional entity that performs actions and generates or processes data in a space data system. An Object that only moves data is called a Communications Object and these are treated in the Communications Viewpoint (see 2.6). A Functional Object may be realized as software and/or hardware. A Functional Object may also be composed of other Functional Objects.

A Functional Object may use a service provided by another Functional Object, provide a service to other Functional Objects, or perform actions jointly with other Functional Objects. These interactions are described in the Functional Viewpoint.

For the Functional Viewpoint the Concerns include: Behavior, Interactions, Interfaces, and Constraints.

A simple example of the Functional Viewpoint is shown in figure 2-3, in which three Functional Objects (Applications A, B and C) are represented as ovals and the functional interactions are represented by dotted lines.

Figure 2-3: Simple Example of Functional Viewpoint

Functional Objects actually reside in physical entities (i.e., Nodes) of the system. The distribution of the Functional Objects among the Nodes can be shown by overlaying the Functional Viewpoint on the Connectivity View of the same system. Such an example is shown in Figure 2-4, in which Figure 2-3 is overlaid with Figure 2-2.

The Functional Viewpoint also provides descriptions of a class of Functional Objects called Information Management Functional Objects that support the location, access, delivery, and management of Information Objects. These also include descriptions of the Information Management Functional Objects that support the operations of a Distributed Information Infrastructure (DII).

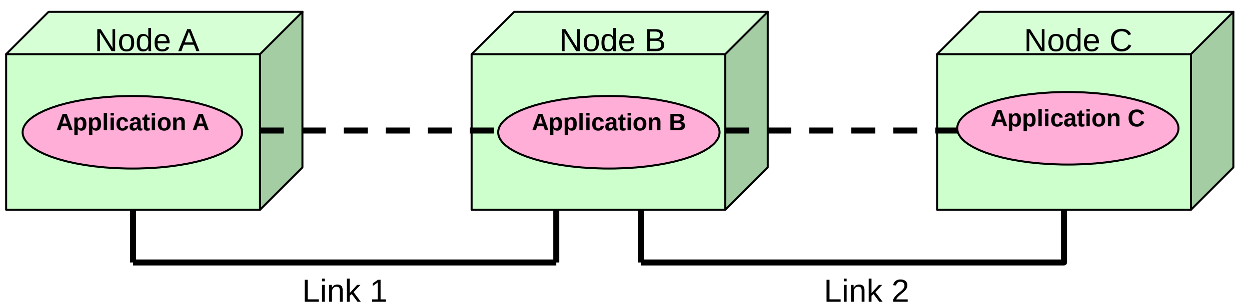

Figure 2-4: Connectivity View and Functional Viewpoint

2.5 INFORMATION VIEWPOINT

The motivation for the Information Viewpoint is that we have data objects with different structures, relationships, and policies that are passed among and operated upon by the functional and protocol elements, and managed (that is, stored, located, accessed, and distributed) by information infrastructure elements. The Information Viewpoint is used to address these aspects of space data systems.

The Information Viewpoint looks at the space data systems from the perspective of the Information Objects that are exchanged among the Functional Objects. 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 Objects, rules for their use and transformation, and policies on access. The basic relationships among Information Objects are shown in figure 2-5. Finally, this Viewpoint shows the relationship between the Information Objects and the Functional Objects that manipulate and exchange them.

For the Functional Viewpoint the Concerns include: Structure, Semantics, Relationships, Permanence, and Rules.

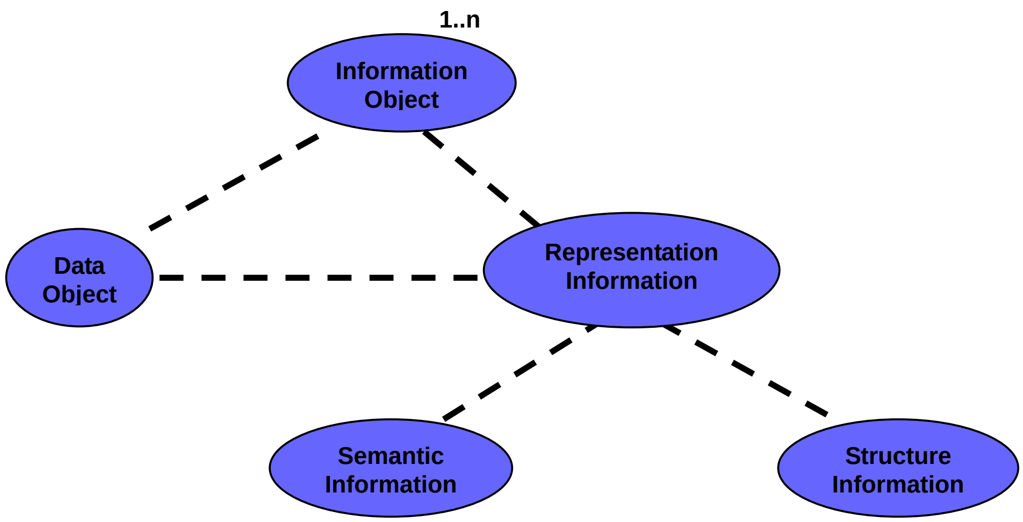

Figure 2-5: Information Viewpoint

2.6 COMMUNICATIONS VIEWPOINT

The motivation for the Communications Viewpoint is to describe the layered sets of communications protocols that support communications among the functional elements in order to meet the requirements imposed by the physical connectivity and operational challenges of communicating in space. The Communications Viewpoint is used to address these aspects of space data systems.

The Communications Viewpoint describes the mechanisms to support information transfer among Functional entities (i.e., Functional Objects) in a space data system. These protocol 'stacks' are most often associated with the links that exist between the physical nodes of a system, but they are functional elements in their own right, with responsibility for transporting data rather than transforming it.

The Communications Viewpoint is depicted as a set of Communications Objects and interactions among them. A Communications Object is an abstract entity that implements a communications protocol. A Communications Object may be realized as software and/or hardware.

Communications Objects support information transfer between or among Functional Objects over Links (i.e., physical connections between or among Nodes). A stack of Communications Objects is usually used to support information transfer from one Functional Object to another Functional Object for performing a sequence of functional interactions. In the stack, the topmost Communications Object directly supports the Functional Object, and the lowest Communications Object handles the Physical Link. Communications Objects implement Protocols, which are most often defined by the protocol data units (PDU) that are exchanged between peer entities and the actions performed by those Communications Objects when they receive any of several defined PDUs. Communications Objects may also be triggered by reception of service data units from users and by internal events such as time-outs.

For the Communications Viewpoint the Concerns include: Standards, Technology, Functionality, and Suitability.

A simple example of the Communications Viewpoint is shown in Figure 2-6, in which two stacks of Communications Objects (implementing Protocols 1, 2, and 3) are represented as two groups of ovals.

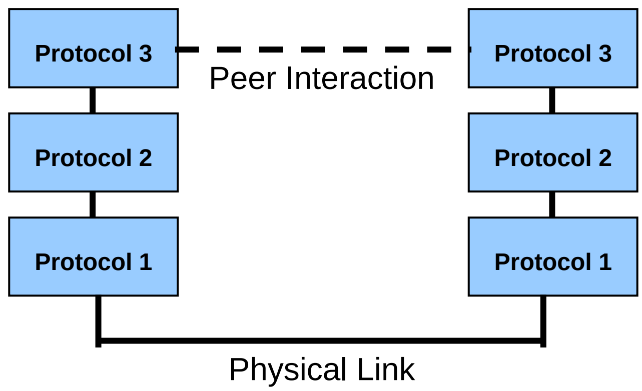

Figure 2-6: Simple Example of Communications Viewpoint

The selection of Communications Objects to support information transfer between Functional Objects over a Link between Nodes depends heavily on the characteristics of the Functional Objects, the Nodes, the Link, and the operating environment. Therefore, it is typical to show the Functional Objects, the Nodes, and the Link together with the Communications Objects in the Connectivity View.

Such an example is shown in Figure 2-7, in which the Connectivity View (Nodes and Links) is overlaid with the Functional Viewpoint (with Functional Objects) and the Communications Viewpoint (Protocols).

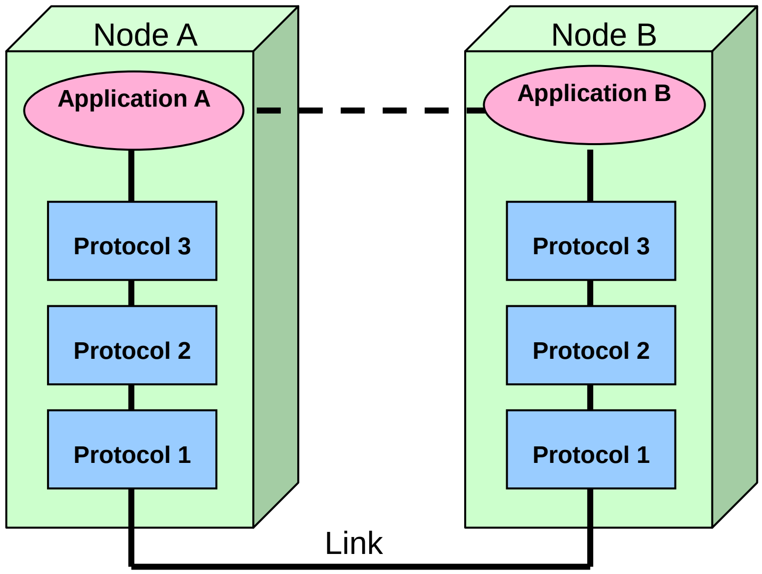

Figure 2-7: Connectivity View with Functional and Communications Viewpoints

2.7 OTHER VIEWS DERIVED FROM THE BASIC VIEWPOINTS

2.7.1 GENERAL

Depending on the particular concerns of a space data system, other Views can be constructed from the five basic Viewpoints explained above, either by combining two or three basic Viewpoints or by overlaying basic Viewpoints with other special entities.

2.7.2 CROSS-SUPPORT SERVICE VIEW

Many of the data system standards defined by CCSDS are intended to provide cross support between agencies, which typically are first identified as Enterprise level agreements. Where detailed relationships between Enterprise Objects are of particular concern in a space data system, a Cross-Support Service View can be constructed by overlaying the Functional and Communications Viewpoint with the Enterprise Viewpoint.

An example is shown in Figure 2-8, which has a combination of Enterprise Objects, Functional Objects, and Communications Objects. In this diagram, a pipe is drawn between a pair of Functional or Communications Objects. In each of the pairs, one Object provides a service to the other Object. In other words, one Object is a server (or service provider) and the other a client (service user). A pipe represents a server-client relationship in the same layer. In this way, the set of services provided by an Enterprise Object to another Enterprise Object can be described.

The reason this View is not a basic View is that no new Object is introduced in this View. These cross support service interfaces are just the exposed set of interfaces of Functional or

Communications Objects. All Functional Objects have interfaces; the ones that are exposed between Enterprise Objects are called Cross Support Services.

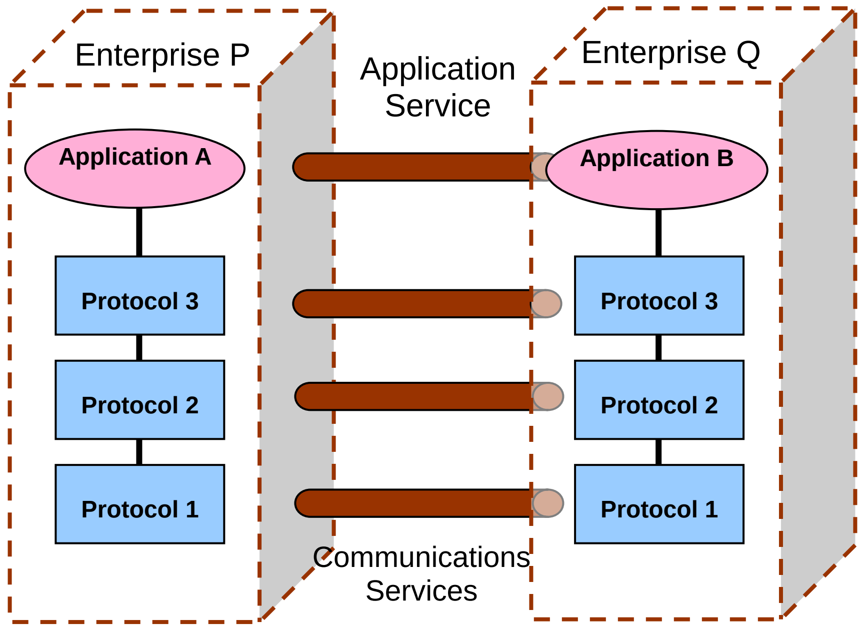

Figure 2-8: Cross-Support Service View (Enterprise, Functional, and Communications Viewpoints)

2.7.3 LAYERED VIEW

Another example of a constructed View is a Layered View, in which layers of Functional and Communications Objects are shown. If a group of Objects provides services of a certain level of functionality to another group of Objects with higher functionality, that group of Objects may be considered a layer. This particular View is often used to show the configuration of layers of Objects in a space data system.

A simple example is shown in Figure 2-9, which shows three layers that include Functional and Communications Objects. The Communications Layer can be further decomposed into sub-layers.

In this diagram, a pipe represents a service interface that is exposed to a higher layer or an Application Program Interface (API). APIs are often used to define the interfaces exposed from one layer to another.

The reason this View is not a basic View is that no new Object is introduced in this View.

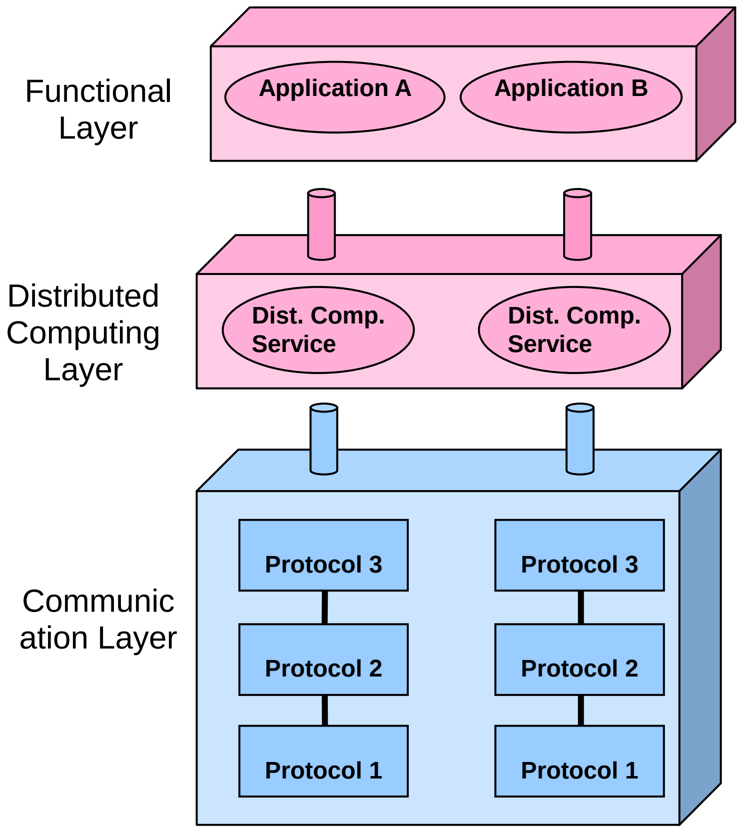

Figure 2-9: Simple Example of Layered View

3 BASIC CONCEPTS

3.1 GENERAL

This section introduces basic concepts and terms that are used throughout this document.

3.2 DEFINITIONS

3.2.1 OVERVIEW

The following concepts and terms are used commonly in the Viewpoints presented in sections 4 through 8.

NOTE

1 In each of the sections that present Viewpoints, the definitions that are used in that Viewpoint will be repeated, and definitions that are used only in that Viewpoint will be given.

2 The following definitions were derived from RM-ODP, but were made somewhat simpler than those of RM-ODP so that they can be understood more easily and intuitively.

3.2.2 GENERAL

Architecture is the concepts and rules that define the structure, semantic behavior, and relationships among the parts of a system, a plan of something to be constructed. It includes the elements (entities) that compose the system, the relationships among the elements, the constraints that affect those relationships, a focus on the parts of the system, and a focus on the system as a whole.

Architecting is the process of creating an architecture specification. It is both a science and an art.

An **Entity** is any concrete or abstract thing of interest. For example, an entity may be a physical instrument, a computer, a piece of software, or a set of functions performed by a system.

A **model** is a formal specification of the structure and/or function of a system. All models are abstractions; abstraction is the suppression of irrelevant detail.

3.2.3 ELEMENTS

An **Object** is an abstract model of an entity.

An Object may be **composed of** two or more Objects. The behaviors of the composite Object are determined by those of its constituent Objects. Component Objects may sometimes be called Components.

A logical Object interacts with other Objects logically. A physical Object interacts with other Objects physically. The interactions of logical Objects are physically supported by physical Objects upon which they are instantiated.

A **Node** is a physical Object that is located at a single place, which may be moving at a high speed, and that is capable of providing a set of resources. A Node may be composed of two or more Nodes.

A **Link** is the locus of relations among Nodes. It may be implemented by a wired connection or with some RF or optical communications media. It may periodically become inactive because of the motion of a Node, or lack of availability of communications resources, for example.

A Viewpoint is 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 model based on specific abstraction criteria is a viewpoint. A Viewpoint defines a pattern or template from which to construct individual views and it establishes the techniques and methods employed in constructing a view.

A View is a representation of a system from the perspective of a set of concerns. Viewpoints define the rules for constructing views. A View may combine elements from one or more Viewpoints.

3.2.4 PROPERTIES OF ELEMENTS

A **policy** is a set of guidelines and constraints on the behaviors exhibited by the objects in the system.

An **action** is something that happens within an object, either with or without participation of another object. An **interaction** is an action performed by an object with participation of another object.

A **behavior** is a set of actions performed by an object for some purpose.

An **interface** is a set of interactions performed by an object for participation with another object for some purpose. An interface is therefore a kind of behavior of an object.

A **service** is a provision of an interface of an object to support actions of another object.

A **role** is a set of behavior and actions of an object that is associated with the relationship of that object with other objects.

3.3 GRAPHICAL REPRESENTATIONS

The icons shown in figure 3-1 are used throughout this document. Some minor variants of

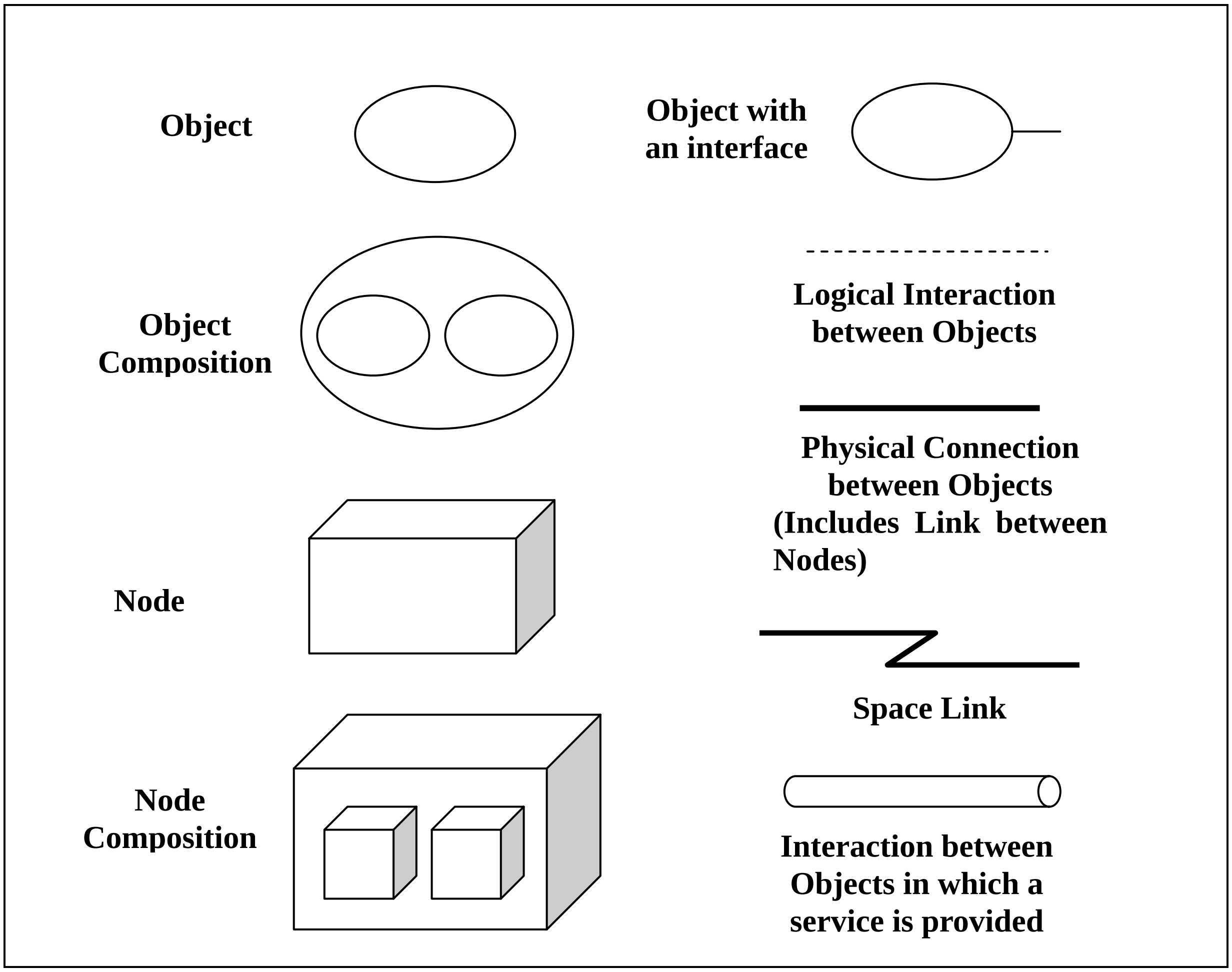

Figure 3-1: Icons Used in This Document

These icons are introduced as needed, in the body of the text.

3.4 ATTRIBUTES OF OBJECTS

As explained in section 2, the Viewpoints of RASDS are described with Objects and their interactions. The icons used to represent attributes of Objects and their interfaces are shown in figure 3-2, these representations are used throughout this document.

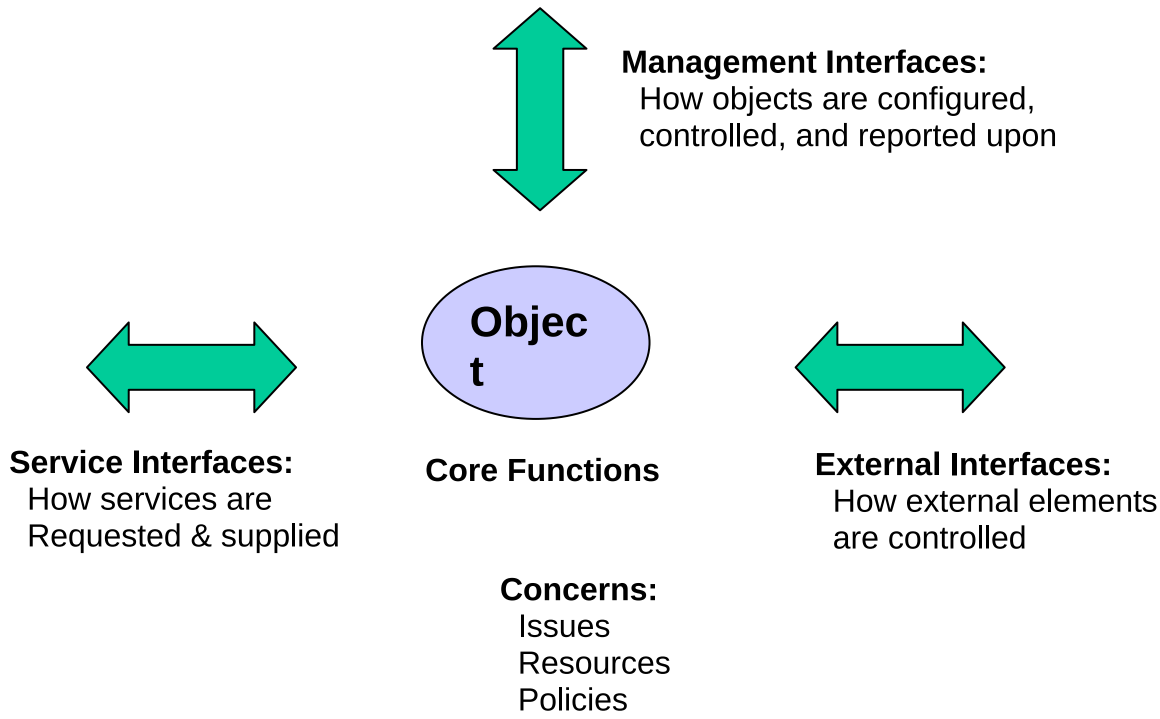

Figure 3-2: Representation of Objects

Any given Object may expose one or more Service Interfaces and provide one or more Core Functions. Through its External Interface, it may call upon other Objects to provide services. The Management Interfaces may be explicit (for instance, a Service Management call to a Protocol Object.) or they may be implicit and represented in internal tables or configuration items.

4 ENTERPRISE VIEWPOINT

4.1 OVERVIEW

The motivation for the Enterprise Viewpoint is that we have complex organizational relationships involving spacecraft, instruments, ground systems, scientists, staff, and contractors that are distributed among multiple organizations (space agencies, science institutes, companies, etc.). The Enterprise Viewpoint is used to address these aspects of space data systems.

NOTE – The enterprise viewpoint is based on the enterprise viewpoint of RM-ODP, but some modifications have been made to better describe the space data systems. In particular, special enterprise objects called Space Enterprises are introduced.

4.2 CONCEPTS

The **Enterprise Viewpoint** of a space data system focuses aspects related to the community, purpose, scope, and policies for that system. This viewpoint includes organizations as well as the Enterprise Objects that have assigned roles, responsibilities, and interactions.

In the Enterprise Viewpoint, a space data system is depicted as a set of Enterprise Objects, their relationships, and the Roles that they perform.

An **Enterprise Object** is governed by a single authority that has its own objectives and policies for operating the object.

An Enterprise Object may be a component of another larger Enterprise Object, which may in turn be a component of a third, even larger, Enterprise Object. Enterprise Objects may participate wholly or in part in other Enterprise Objects.

A **Resource** is an entity that has some role and performs some action within a system.

NOTE – System management, lifecycle views on systems and other views that are relevant to the Enterprise Viewpoint will be addressed in a later issue of this document.

4.3 ENTERPRISE OBJECTS

The following are special classes of Enterprise Objects, which represent various classes of organizations.

A **Space Enterprise** (e.g., NASA) is a top level, autonomous entity that is dedicated to the exploration and/or exploitation of space. It has its own objectives, resources, and policies and it is not a component of any other Space Enterprise.

A **Community** (e.g., Earth Science) can exist within one Space Enterprise or across multiple Space Enterprises. It is distinguished by being bound by common objectives and relationships and offers a set of resources that can be shared within the Community and with other Communities.

A **Domain** (e.g., NASA Code Y) is a type of Community that is under single organizational, administrative, or technical control. A domain may have resources, policies, access control, and possibly constraints on quality of service. A Domain may be subdivided into Sub-Domains. Multiple Domains may be collected into a Federation.

A **Federation** (e.g., CEOS or CCSDS) is a Community consisting of multiple Domains that come together to share resources while each domain retains its autonomy over those resources. Federations are bound by negotiated agreements. A Federation may only include some members of a Domain or Sub-Domain (e.g., a particular Earth Observing project). Members of a Federation agree on rules for sharing resources and for joining and/or leaving the federation.

The following table shows other typical Enterprise Objects. The first three rows represent organizations; the rest are resources or facilities.

How each Space Enterprise is decomposed into component Enterprise Objects highly depends on each space data system, and table 4-1 shows typical enterprise objects used in many space data systems and their containment relationships.

Table 4-1: Enterprise Objects

Enterprise Objects	Description
Mission	An Enterprise Object that is responsible for designing, building, and / or operating one or more spacecraft
Project	An Enterprise Object that is responsible for designing, building, and / or operating one or more space system components
Program	An Enterprise Object that is responsible for one or more Missions or Projects
Ground Tracking Network	This is a multi-mission facility that may be comprised of one or more Nodes and one or more tracking stations, used for communicating with spacecraft and performing radiometric measurements against spacecraft.
Space Tracking Network	This is a multi-mission facility that may be comprised of one or more Nodes (relay satellites) used for communicating with spacecraft and performing radiometric measurements against spacecraft.
Spacecraft Control Center	A center used for controlling a spacecraft.
Instrument Control Center	A center used for controlling (a) instrument(s). It may not exist as an enterprise object in all space data systems.

Enterprise Objects	Description
Science Facility	A Facility that requests activities of a spacecraft and analyzes data obtained from that spacecraft. It may not exist as an enterprise object in all space data systems.
Data Archive Center	A center that archives data obtained from spacecraft and delivers requested data to a science institute. It may not exist as an enterprise object in all space data systems.

4.4 ATTRIBUTES OF ENTERPRISE OBJECTS

The attributes of Enterprise Objects are shown in figure 4-1. The interfaces of Enterprise Objects are characterized by such things as Requirements, Memorandum of Understanding, Service/Support Agreements, Interface Control Documents, and so on. Interfaces among Enterprise Objects are often created because of shared science or exploration goals and may involve cross support agreements, interoperability requirements, and agreements on data sharing and access. Standards are often employed as the means for enabling these interfaces to work.[1]

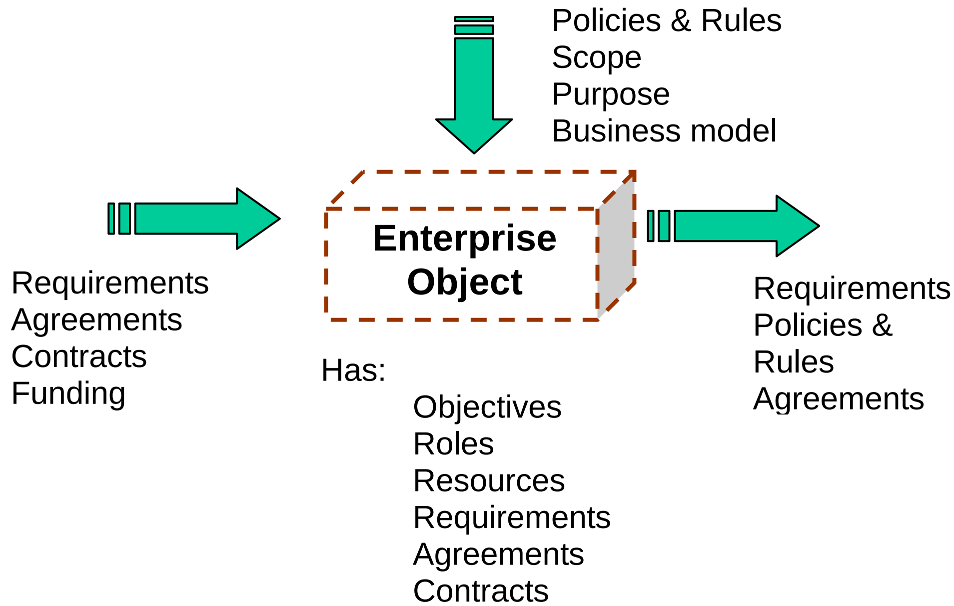

Figure 4-1: Attributes of Enterprise Objects

The Enterprise Viewpoint is also associated with Use Cases and Operations Concepts. This is the primary viewpoint in the system where personnel, operations, and other organizational concerns are expressed.

[1] Detailed descriptions of the attributes of Enterprise Objects will be provided in a later issue of this document.

4.5 EXAMPLES OF SPACE DATA SYSTEMS DESCRIBED WITH ENTERPRISE VIEWPOINT

Some examples of the Enterprise Viewpoint are shown in the following examples.

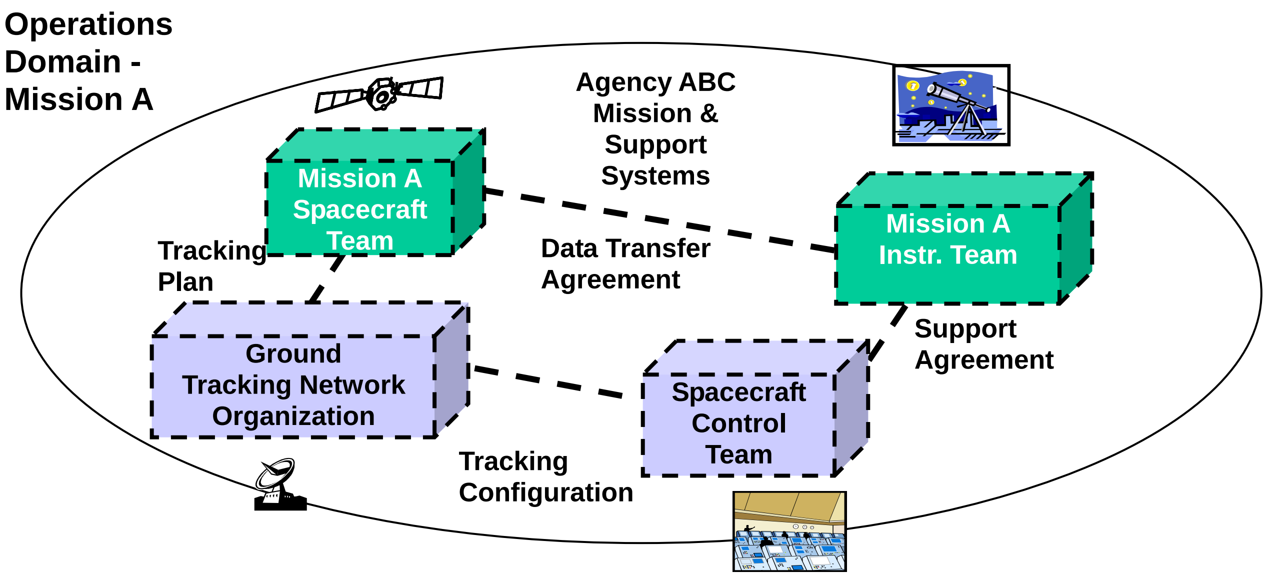

Figure 4-2: Example of Enterprise Viewpoint (Mission A)

Enterprise Objects involved in the operations of Mission A are shown in Figure 4-2 together with the interfaces between them. Mission A is a mission of Agency ABC, not involving cooperation with other Agencies, and therefore all the Enterprise Objects belong to Agency ABC.

Enterprise Objects involved in the operations of Mission Q are shown in Figure 4-3 together with some of the relevant interfaces between them. Mission Q is a joint mission between Agencies ABC and QRS, and therefore some Enterprise Objects belong to Agency ABC and some to Agency QRS. Shared missions often are based upon some quid-pro-quo arrangements, involve some sort of agreement or contract and require a relationship of trust and interdependence between organizations. There are two primary kinds of elements in the Enterprise Viewpoint, organizations and facilities.

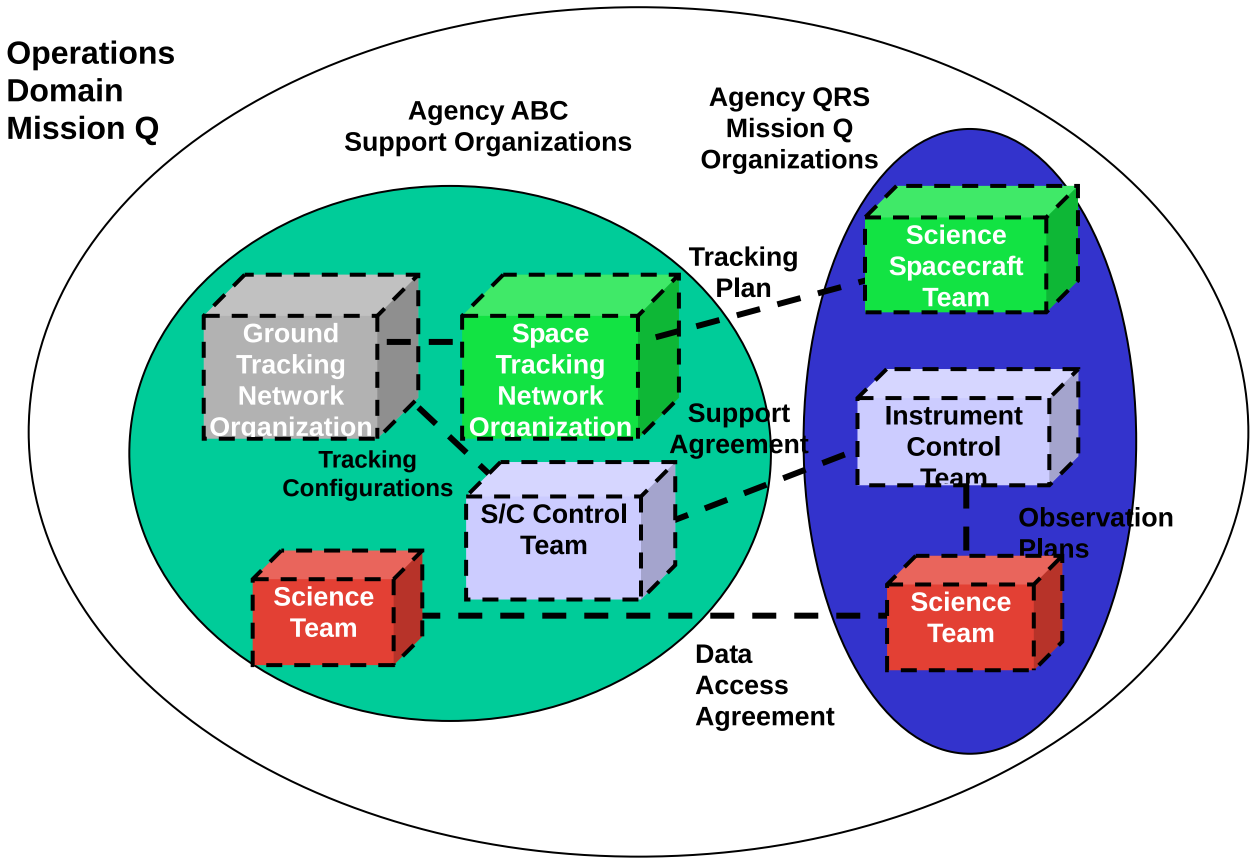

Figure 4-3: Example of Enterprise Viewpoint (Mission Q)

4.6 SECURITY ISSUES IN THE ENTERPRISE VIEWPOINT

At the Enterprise Viewpoint the security issues that will be addressed include organizational policies, rules, trust relationships, domain boundaries (e.g., operational vs. science) and cross support security agreements. The implementation of the mechanisms to enforce these rules and agreements are detailed in other views.

5 PHYSICAL VIEWPOINT/CONNECTIVITY VIEW

5.1 OVERVIEW

The motivation for the Physical Viewpoint is that space data systems are made up of elements that must operate in physical space, where the physics of motion, structure, power and connectivity must be considered. For analysis of Space Systems all of these aspects, and the power, thermal, structural views associated with them, must be considered. For the description of Space ***Data*** Systems, we focus just on the Connectivity View of the Physical Viewpoint.

The motivation for the Connectivity View is that we have system elements that are in motion through space and consequently connectivity issues associated with pointing, scheduling, long round-trip light times, intermittent connectivity, and low signal-to-noise ratios, all of which require special protocols and functionality to deal with. The Connectivity View is used to address these aspects of space data systems.

NOTE – The Connectivity View is one of the aspects of the Engineering Viewpoint of RM-ODP. It is called out separately in RASDS because it exposes physical issues and constraints in the design of space data systems, which are distinct from those encountered in typical terrestrial distributed systems.

5.2 CONCEPTS

The **Connectivity View** is an engineering view on a space data system, which is focused on the Node and Link view of a system, the physical connection among Nodes, their physical and environmental constraints, and their physical dynamics.

In the Connectivity View, a space data system is depicted with Nodes and the physical connections among them (Links). This view also describes how they move and the effects that the environment has upon their behaviors. This is important in order to understand the challenges from the physical environment in which the systems operate (particularly the motion, discontinuous connectivity, and extremely distant and broad distribution of physical devices), which require specialized protocols and systems design.

The Connectivity View may often be used in conjunction with the Functional Viewpoint to show how Functional Objects (which will be described in detail in section 6) are distributed among the Nodes of a space data system, and where instances of the same (or similar) Functional Objects may exist in one or more Nodes in a system.

5.3 DEFINITIONS

A **Node** is a physical Object that has some well-understood, possibly rapidly moving, location and has a set of resources to support the activities of the Functional, Information, and Communications Objects contained in that Node. A Node may be composed of two or more (sub-)Nodes.

A **Link** is the locus of relations among Nodes.

A physical **Port** is the physical element of a Node where a Link is connected. The physical Port may connect to one or more physical Ports on Nodes that are contained within the Node.

One or more logical connections between Functional Objects contained within two Nodes may be mapped onto a single physical Link between the Nodes.

5.4 NODES

Nodes are the physical locations where instances of Functional, Information, and Communications Objects are located. Depending upon the design of the specific system, Nodes may contain other Nodes. Thus, a ground Tracking Site (a Node) may consist of one or more Tracking Stations, which are Nodes. Each Tracking Station contains a common set of Functional Objects for Tracking, Data Handling, Radiometric Data Collection, and Directive Execution (slews, pointing). Similarly, a single spacecraft is a Node that is composed of Nodes such as instruments, Command and Data Handling (C&DH) computer, and RF system. Each of these Nodes implements one or more other Functional Object. Each Node is owned by an Enterprise (see section 4), but it may contain Nodes owned by other Enterprises (e.g., spacecraft owned by Agency A flying instruments owned by Agency B).

Table 5-1 shows typical Nodes that are used in space data systems. Which Nodes are used in a space data system may differ from system to system, and the following lists show only typical Nodes used in many space data systems.

Table 5-1: Typical Nodes

Nodes	Description
Spacecraft	A spacecraft (or a lander, rover, balloon, etc) used to achieve mission goals (e.g., observations or experiments).
Relay satellite	A spacecraft (or a lander) that relays data between spacecraft and a tracking network or between different sets of spacecraft. It may not exist as a physical object in all space data systems.
Instrument	A subsystem of a spacecraft used to achieve mission goals (e.g., observations or experiments). It may not exist as a physical object in all space data systems. If it exists, it is a component of a spacecraft.
Onboard data system	A subsystem of a spacecraft used to store and manage data. It may not exist as a separate physical object in all spacecraft.

Tracking station	A subsystem of a ground communications facility that is used to track spacecraft, transmit commands, receive telemetry, and optionally to produce radiometric and tracking data.
Spacecraft control facility	A facility that is part of a mission operations system that is used to plan, control, and monitor spacecraft operations.
Instrument control facility	A facility that is part of a mission operations system that is used to plan, control, and monitor instrument operations. It may not exist as a separate facility in all enterprises.
Orbit determination facility	A facility that is part of a mission operations system that is used to analyze radiometric tracking data and to determine the orbital and attitude of a spacecraft. It may not exist as a separate facility in all enterprises.
Trajectory design facility	A facility that is part of a mission operations system that is used to design a spacecraft trajectory and plan maneuvers. It may not exist as a separate facility in all enterprises.
Mission planning facility	A facility that is part of a mission operations system that is used to create, control, and monitor mission operational plans. This may include overall observation and mission scenario planning. It may not exist as a separate facility in all enterprises.
Data analysis facility	A facility that is part of a mission operations system that is used to process instrument data and to perform a variety of additional data analyses. It may not exist as a separate facility in all enterprises.

5.5 LINKS

Nodes are connected together using Links that have specific behavioral, functional, and physical attributes. The attributes include performance and physical constraints upon the Links (e.g., spacecraft and planetary motion, physical distance, environmental noise, interference, occultation, pointing, etc.). Links are connected to Nodes at a Physical port.

Table 5-2 shows typical Links that are used in space data systems. Which Links are used in a space data system differs from system to system, and the following lists show only typical Links used in many space data systems.

Table 5-2: Typical Links

Links	Description	Attributes
Space Link	A Link between a Node in space (e.g., a spacecraft) and a Node on the ground (e.g., a ground station), or a Link between two Nodes in space (e.g., between two spacecraft).	• Continuous vs. episodic connectivity • Pointing and view periods • Frequency band • Delay and signal attenuation • Single vs. multiple access • Bit rate, possibly variable
Ground Link or Network	A Link between two Nodes or a network among multiples Nodes on the ground.	• Wide area or local area • Dedicated or public • Single vs. multiple access • Bit rate
Onboard Link or Bus	A Link between two Nodes or a bus among multiple Nodes on the same spacecraft.	• Single vs. multiple access • Redundancy • Bit rate

5.6 ATTRIBUTES OF NODES

The attributes of Nodes are shown in figure 5-1. The functional interfaces of a Node are determined by the Functional and Communications Objects contained in the Node, that need to communicate with Functional and Communications Objects contained in other Nodes, and each of the interfaces is associated with one or more Links attached to the Node. Links attach to physical ports on nodes, and Functional or Communications Objects typically have associated logical ports.

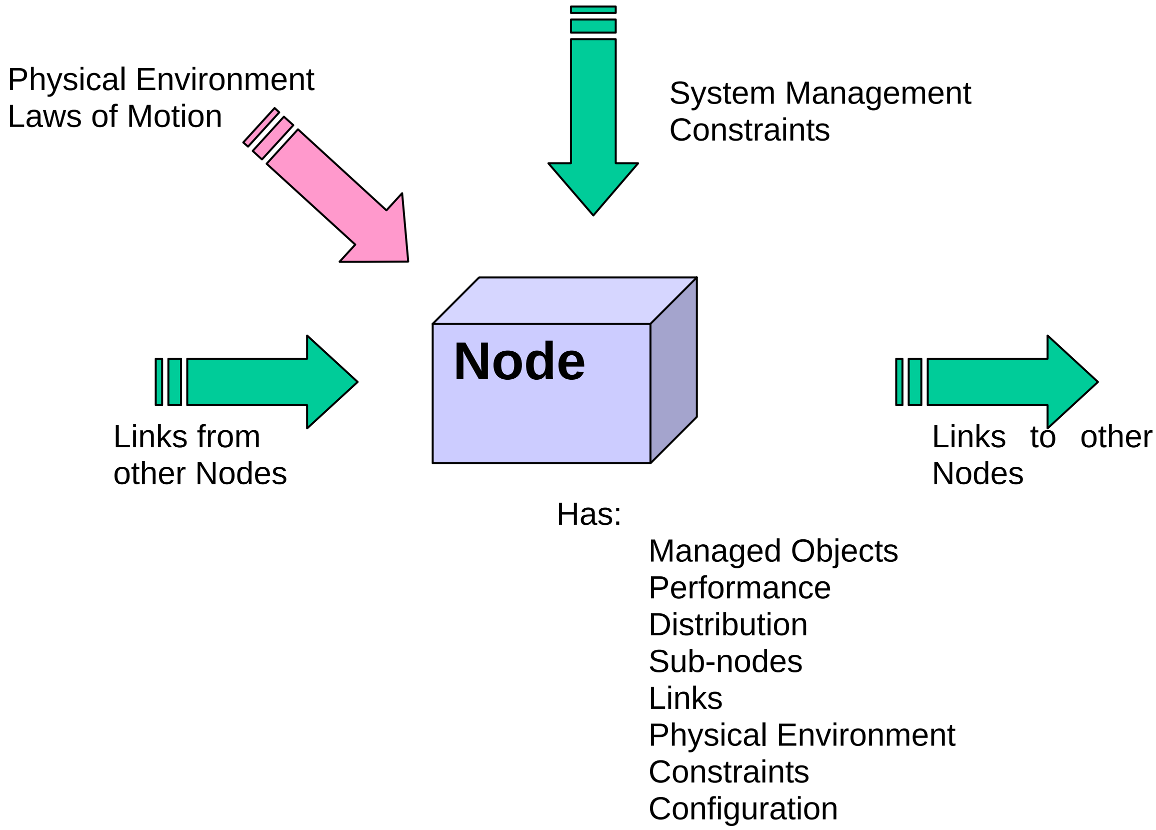

Figure 5-1: Attributes of Nodes

The functional behavior of a Node is determined by the Functional Objects contained in the Node, but the mechanisms of the interactions with the other Nodes through its interfaces are determined by the Communications Objects in the Node that are used to support the Functional Objects.[1]

Some Nodes, particularly spacecraft, also have physical behavior that is determined by the forces acting upon the Node. These forces may be propulsive, gravitational, or due to other elements in the environment that determine the velocity, direction of motion, and acceleration of the spacecraft. The physical behavior of the spacecraft, some of its Components (e.g. antenna aperture, transmitter power, receiver sensitivity) and the physical characteristics of the environment, all exert a strong influence on the performance of the communications systems and the behavior of the Links. The protocols that are described in the Communications Viewpoint are selected to deal with these behavioral and environmental challenges.

[1] Detailed descriptions on the attributes of Nodes will be provided in a later issue of this document.

5.7 EXAMPLES OF SPACE DATA SYSTEMS DESCRIBED WITH CONNECTIVITY VIEW

Figure 5-2 shows Nodes used supporting Mission A, as shown in figure 4-2.

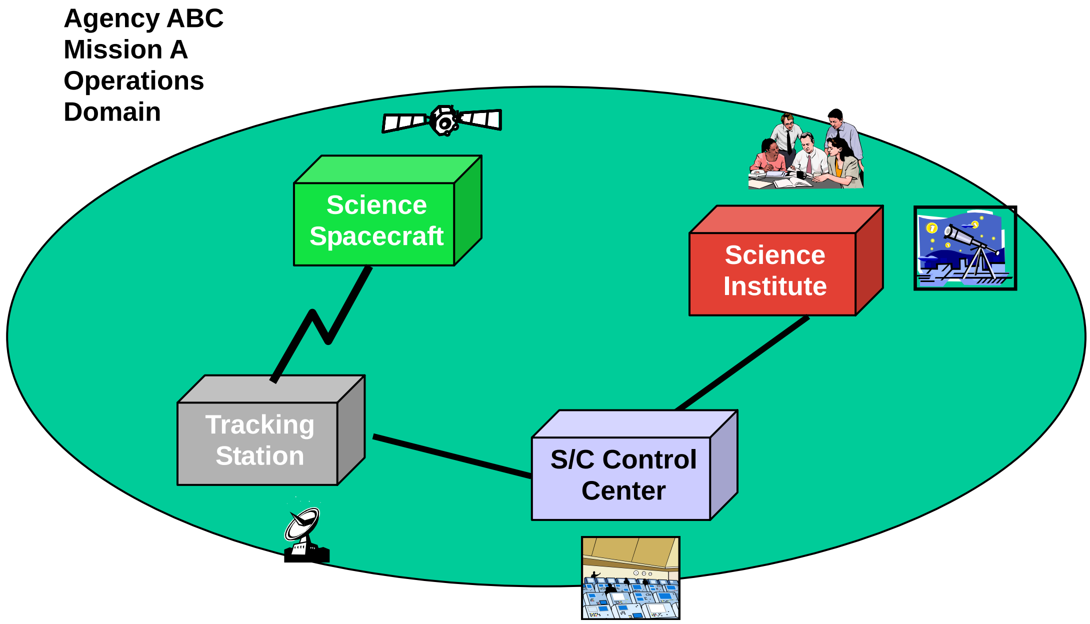

Figure 5-2: Example of Connectivity View (Nodes for Mission A)

Figure 5-3 shows Nodes supporting for Mission Q.

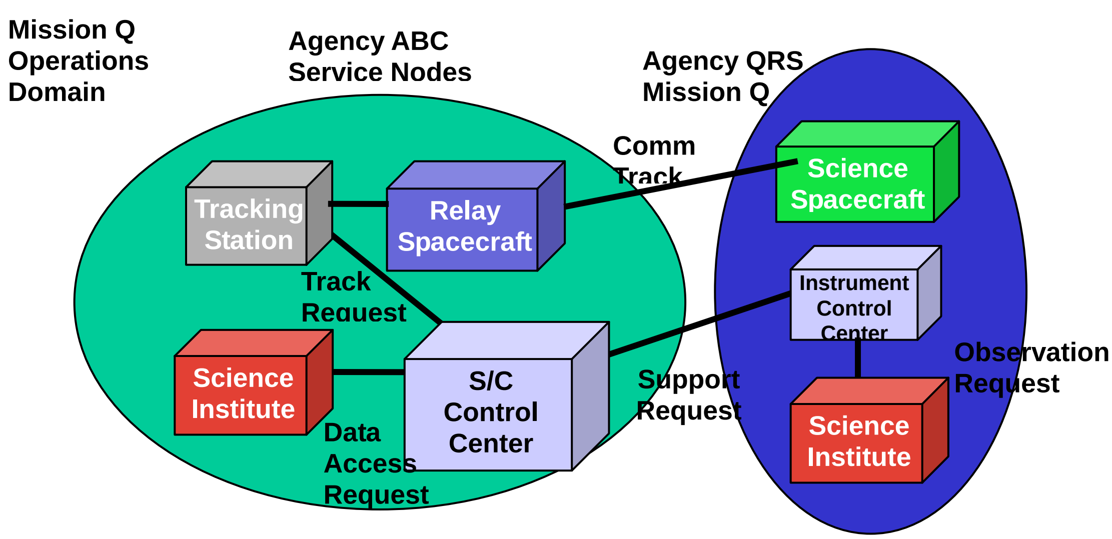

Figure 5-3: Example of Connectivity View (Nodes for Mission Q)

Figure 5-4 shows the decomposition of the Nodes used for Mission A (shown in figure 5-2) into component Nodes. In this case, the two Nodes shown in figure 5-2, Spacecraft Control

Center and Tracking Station, are merged into one Node (i.e., Spacecraft Control Center) in this figure.

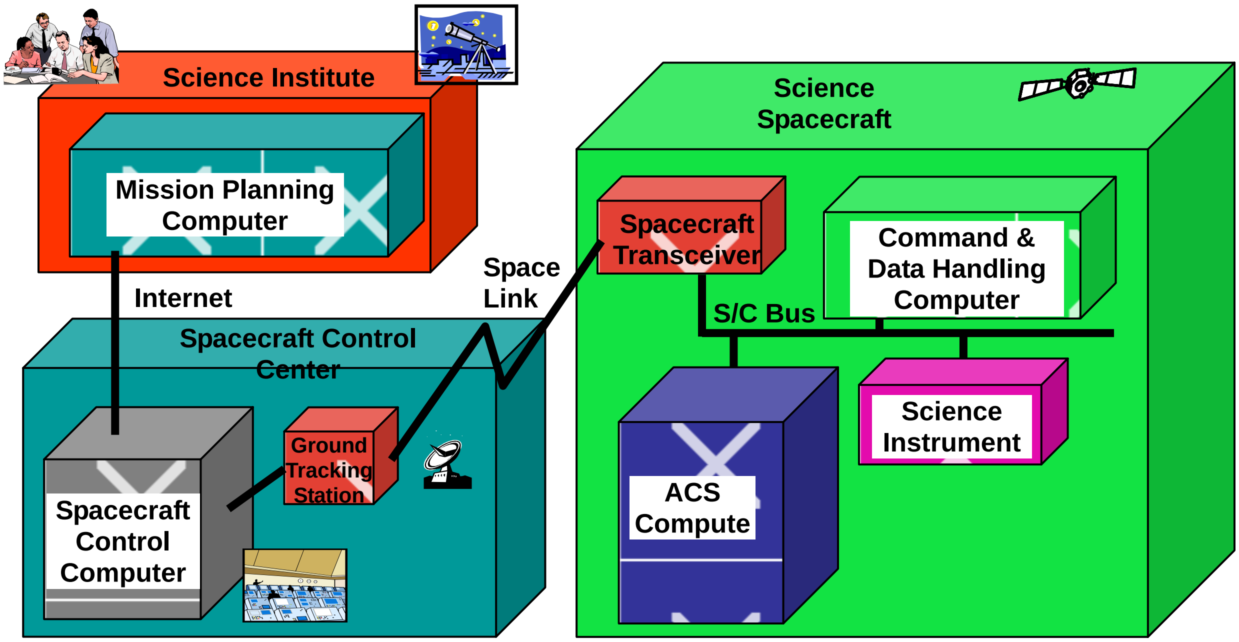

Figure 5-4: Example of Connectivity View (Node Decomposition)

Of course, the nodes shown in figure 5-4 may be further decomposed into lower-level nodes, with their own links.

5.8 SECURITY ISSUES IN THE CONNECTIVITY VIEW

In the Connectivity View the security issues that are dealt with include the physical entities that are used to implement security policies. These will include: routers and firewalls, hardware encryption devices, and possibly physical boundaries such as shielded rooms or air gaps. Protocol entities that may implement elements of security functionality, such as security protocols or routing filters will appear in the Communications Viewpoint.

6 FUNCTIONAL VIEWPOINT

6.1 OVERVIEW

The motivation for the Functional Viewpoint is to separate functional elements and their logical interactions from the engineering concerns of where functions are housed, how they are connected, which protocols are used, or what language is used to implement them. The Functional Viewpoint is used to address these functional aspects of space data systems.

NOTE – The Functional Viewpoint corresponds to the computational viewpoint of RM-ODP. The computational viewpoint of RM-ODP describes the structure of application processes in a distributed processing system. In RM-ODP, application processes are always implemented as pieces of software residing in computers, hence the computational viewpoint. In space data systems, however, application processes do not always reside in computers. They may reside in simple devices or be implemented in hardware for efficiency. For this reason the word 'functional' was chosen instead of 'computational'. The engineering of these application processes, whether in software (e.g. choice of languages) or hardware (e.g. selection of discrete logic or FGPA), is not directly treated in RASDS.

6.2 CONCEPTS

The **Functional Viewpoint** is a view on a space data system that focuses on the structure of the functions performed by that system and their behavior, and on the interactions among the functions. This includes functional objects, the logical connections between them, their interactions, and logical interfaces.

In the Functional Viewpoint, a space data system is depicted as a set of Functional Objects, the logical connections among them, and the information that flows between them. The details of these information objects are covered in the Information Viewpoint, described in Section 7.

A Function is the set of actions or activities performed by some element to achieve a goal. A **Functional Object** performs Functions to achieve a goal of a space data system or to support actions of another Functional Object, and this may involve data transfer, generation, or processing in performing those actions.

This view of Functional Objects includes their behavior and other attributes and the logical flow of information among objects. In the design of any given system, instances of these Functional Objects may appear in one or more physical locations, associated with one or more Nodes. The issue of allocation of Functional Objects to Nodes is treated by a combined view formed of the Functional Viewpoint and Connectivity View. The physical means for providing connections among them are treated in the Connectivity View (Section 5). Physical attributes of the connections and their behavior are also described in the Connectivity View.

The Functional Viewpoint contains Functional Objects that are used for the control and management of system behavior, such as planning, scheduling, monitoring, and active control elements that are part of describing the functional behavior of the system. For describing the full behavior of a complex system, separate sets of data flows, control flows and behaviors may be provided for the same common set of Functional Objects.

6.3 FUNCTIONAL OBJECTS

Table 6-1 shows typical functional objects used in space data systems. These are provided as examples, and any given system may decompose these differently or use other names for the same functions. For instance, Orbit Determination and Trajectory Design are often called Flight Dynamics.

Depending on the system, each of these Functional Objects may be decomposed into sub-functions, each of which is performed by a component Functional Object of the parent Functional Object. How Functional Objects are decomposed into component Functional Objects depends heavily on the system, and it is beyond the scope of this architecture to define specific decompositions of these Functional Objects.[1]

Table 6-1: Typical Functional Objects

Functional Objects	Description
Experiment control	A function to control an experiment or observation (data acquisition, sample acquisition, etc.).
Data transport	A function to manage and control the execution of data transport functions supplied by Communications Objects.
Directive execution	A function to execute a set of directives (goals or a time-ordered set of directions).
Directive management	A function to remotely manage a set of directives (goals or a time-ordered set of directions).
Directive generation	A function to generate a set of directives (goals or a time-ordered set of directions) based on a mission plan.
Monitor & Control	A function to monitor the status of other functional objects and to execute necessary actions when a pre-defined anomaly or deviation occurs.

[1] Table 6-1 will be revised in a future issue of this document so that the Functional Objects shown therein will represent services provided by CCSDS and other space data systems standards.

Functional Objects	Description
Mission planning	A function to generate a mission plan (time-ordered set of goals or sequence of activities).
Spacecraft analysis	A function to analyze the status of a spacecraft using data from a data store.
Mission analysis	A function to analyze the status of instruments and to assess the level of achievement of mission goals, using data from a data store.
Tracking	A function to steer an antenna to maintain communications links with a spacecraft or a ground station.
Radiometric data collection	A function to collect radiometric data (e.g., range and Doppler).
Orbit determination	A function to estimate the state vector of a spacecraft using radiometric data and possibly image or other position sensitive data taken by the spacecraft.
Trajectory design	A function to design the trajectory of a spacecraft including plans for orbit change maneuvers.

In table 6-2 a number of typical infrastructure objects are shown. These are also Functional Objects, but they are distinguished because they are often considered as providing supporting services for the more application oriented Functional Objects shown in table 6-1. Sometimes these infrastructure objects are shown as a layer, as mentioned in Section 2.7.3.

Table 6-2: Typical Infrastructure Objects

Functional Objects	Description
Information Management	A set of functions to store, locate, access, and deliver data, see the Information Viewpoint for more details on these elements.
System Management	A set of functions to monitor, manage, configure, and control other functions in a system, usually via their Management interfaces.
Messaging Middleware	A set of functions to provide services for naming, locating, accessing, and interfacing with elements of a distributed system.

6.4 INTERFACES OF FUNCTIONAL OBJECTS

The attributes of Functional Objects are shown in figure 6-1.

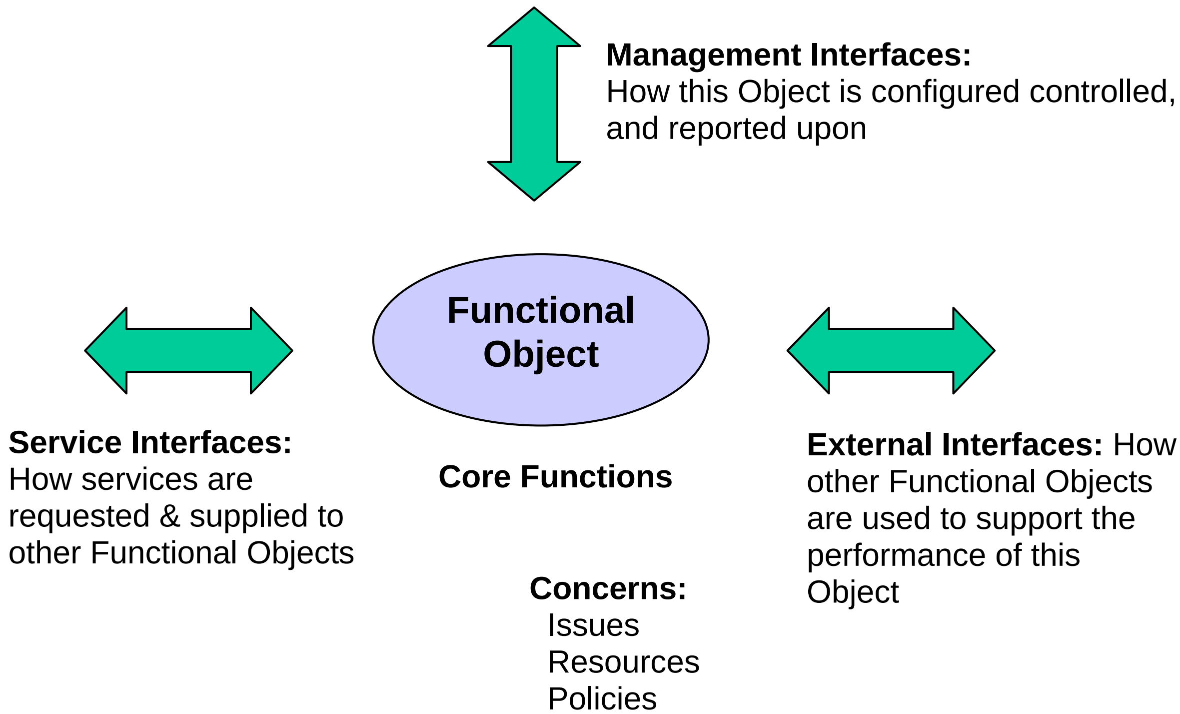

Figure 6-1: Typical Infrastructure Objects

The interfaces of Functional Objects are classified into three categories: Service Interfaces, External Interfaces, and Management Interfaces.[1] Every Functional Object has one or more interfaces through which the actions of the object are invoked. These interfaces may be shown explicitly or just implied as the locus of the connection between one Functional Object and another. One or more external Functional Objects may connect to an interface, but it may only connect to one internal Functional Object of which a high level object is composed.

6.5 EXAMPLES OF SPACE DATA SYSTEMS DESCRIBED WITH FUNCTIONAL VIEWPOINT

Figure 6-2 shows a representative set of Functional Objects used in typical space data systems together with the logical interactions that occur among them (shown with dotted lines).

[1] Detailed descriptions of the attributes of Functional Objects will be provided in a later issue of this document.

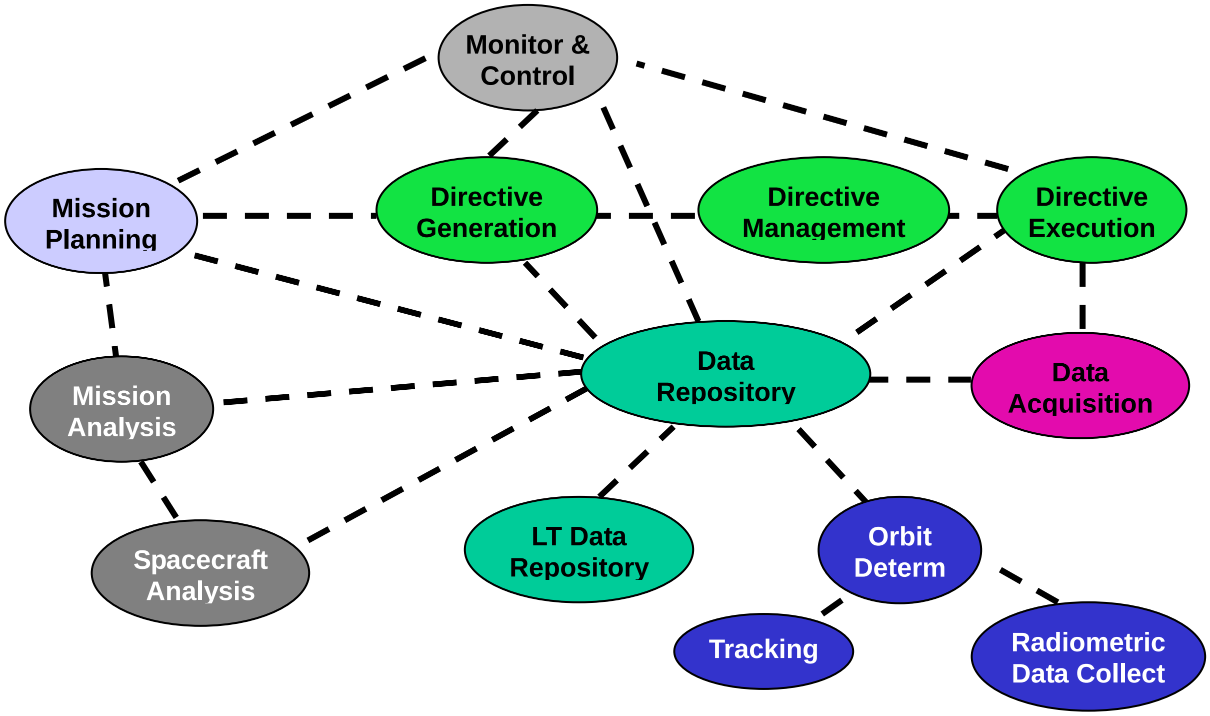

Figure 6-2: Example of Functional Viewpoint (Functional Objects and Interactions)

Figure 6-3 shows how the Functional Objects shown in figure 6-2 may be distributed among the Nodes shown in figure 5-2.

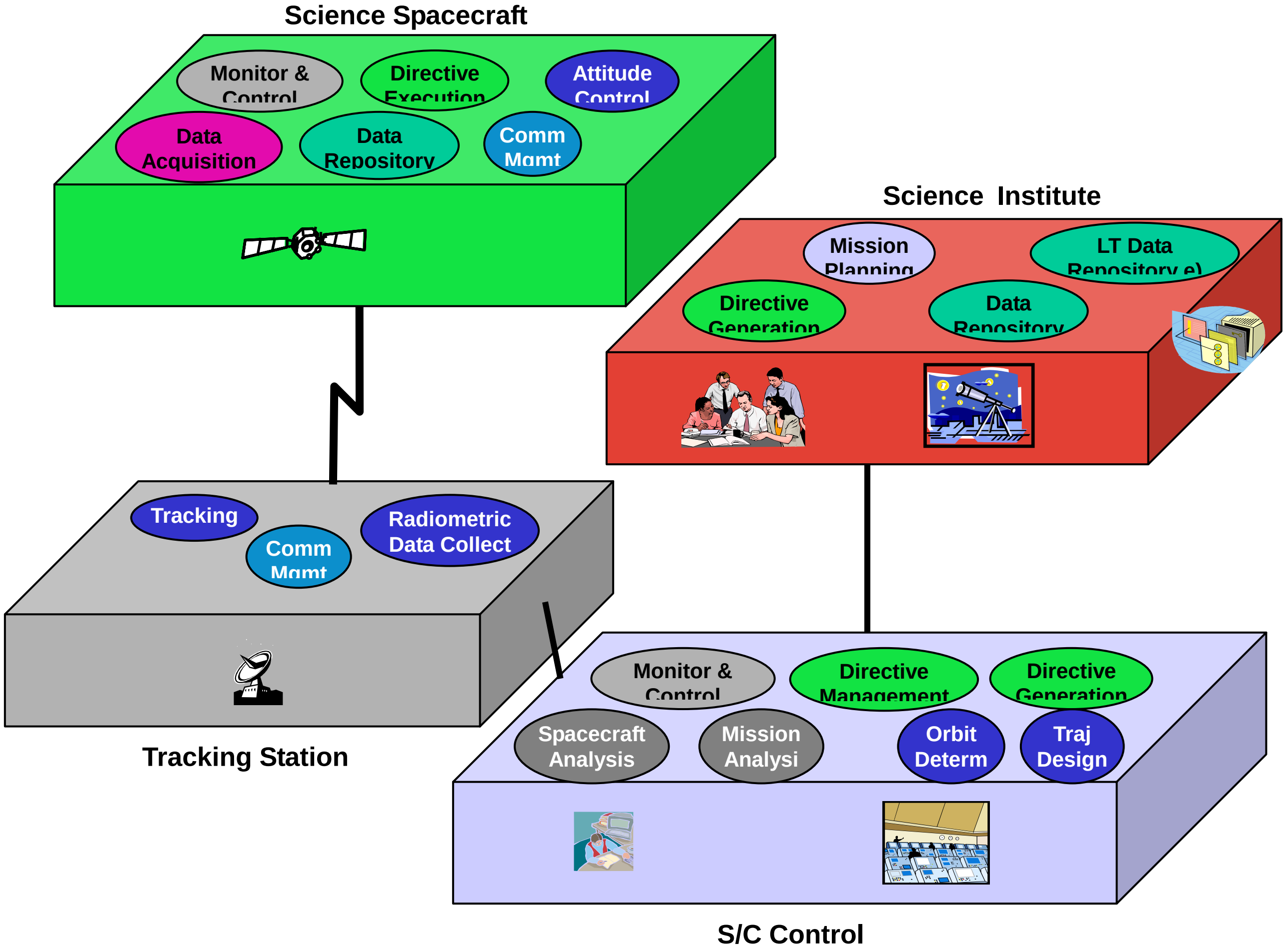

Figure 6-3: Example of Connectivity View with Functional Objects and Nodes

If figure 6-3 were drawn to show an autonomous spacecraft, some of the Planning, Directive Generation, and Monitor & Control functions might be moved on-board. Exploring the implications of these options, on system functionality, performance, and support requirements, is possible once all of the elements of the Functional and Connectivity Views have been specified.

6.6 EXAMPLE OF SPACE DATA SYSTEM WITH INFORMATION MANAGEMENT INFRASTRUCTURE

Included in the Functional Viewpoint are descriptions of active Information Management Functional Objects, elements of a Distributed Information Infrastructure (DII) that supports the location, access, delivery, and management of Information Objects. These Information Management Functional Objects are Functional Objects, but they are often considered together with Information Objects because of the close relationship between them.

Figure 6-4 shows a representative set of Functional Objects that might be used to carry out some activity. Supporting these is the set of Information Management Functional Objects that provide an infrastructure for managing, accessing, locating, and distributing the information exchanged by the Functional Objects.

In some systems, these infrastructure elements may be instantiated by simple files, tables, or even data stored in memory. In other systems these will be system functional elements in their own right, implemented as subsystems and using various commercial elements like DBMS and distributed system frameworks.

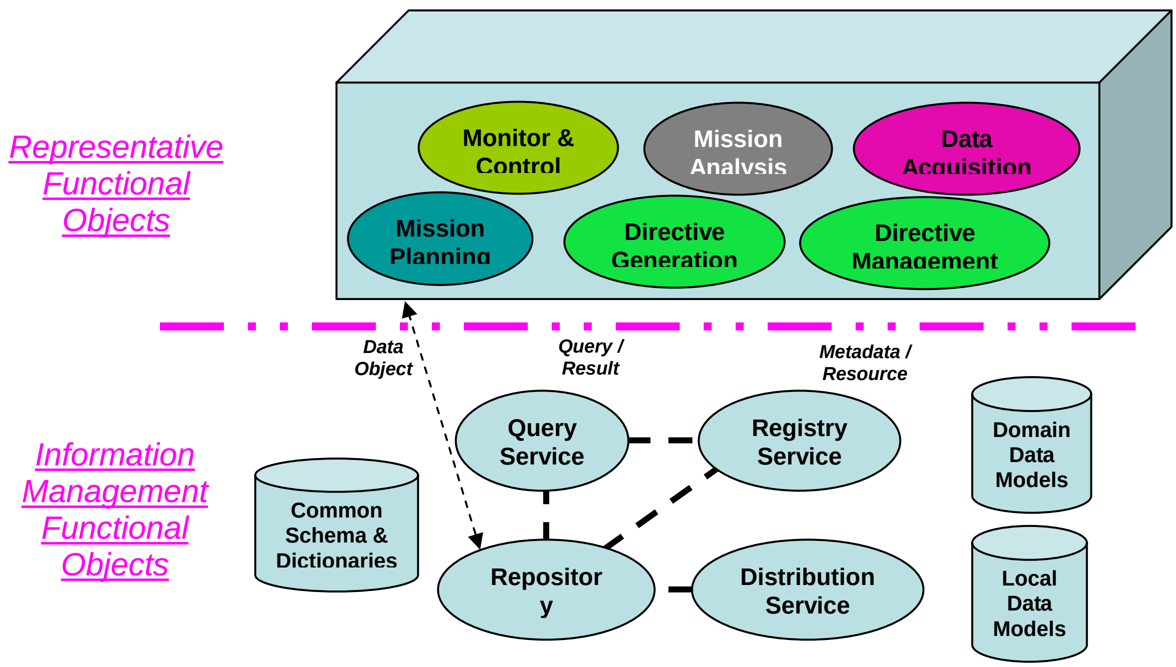

Figure 6-4: Representative Functional Objects and Information Management Infrastructure

These basic Information Management Functional Objects may be composed into a broad set of other information management services to support mission operations functions as well as on-board data management. They may also be combined with other functions that do transaction management or data ingest to produce federated data systems and back end archival systems. The description of these Information Management Functional Objects, their functions and interfaces is being separately addressed in the Information Architecture Reference Model.

6.7 SECURITY ISSUES IN THE FUNCTIONAL VIEWPOINT

In the Functional Viewpoint the Functional Objects and services that are used to implement security policies and approaches are defined. These will include: access control, authentication, source level encryption, and service elements such as key management systems. These may be shown as Functional Objects in their own right, or as attributes of other functional objects.

7 INFORMATION VIEWPOINT

7.1 OVERVIEW

The motivation for the Information Viewpoint is that data objects with different structures, relationships, and policies are passed among the functional elements, and managed (that is, stored, located, accessed, and distributed) by information infrastructure elements. The Information Viewpoint is used to address these aspects of space data systems.

NOTE – The Information Viewpoint corresponds directly to the information viewpoint of RM-ODP. This abstract view on the system will be expressed during implementation (RM-ODP Engineering and Technology Views) using concrete specifications. In the case of Information Objects, there will be a set of data specifications and data elements bound to some particular language or framework.

7.2 CONCEPTS

The **Information Viewpoint** is a view of a space data system that focuses on the information used by that system. This includes structural (syntactic) and semantic views of the information, the relationships among information elements and rules for their management and transformation.

The Information Viewpoint looks at space data systems from the perspective of the Information Objects that are exchanged among Functional Objects.

It includes descriptions of Information Objects (their structure and syntax) and information about the meaning and use of these Objects (contents and semantics), as well as the relationships among Objects, rules for their use and transformation, and policies on access.

This Viewpoint shows the relationship between the Information Objects and the Information Management and other Functional Objects that manipulate and exchange them.

7.3 INFORMATION OBJECT VIEWS

Information Objects are often represented from several different views, ranging from abstract to very concrete. These different views are shown in figure 7-1.

The most concrete representation of an Information Object is the set of bits or bytes of actual data that are used to store information in memory or to exchange it across a communications link. If the Information Object is ‘self descriptive’ it may contain both the semantic content and a description of the syntax.

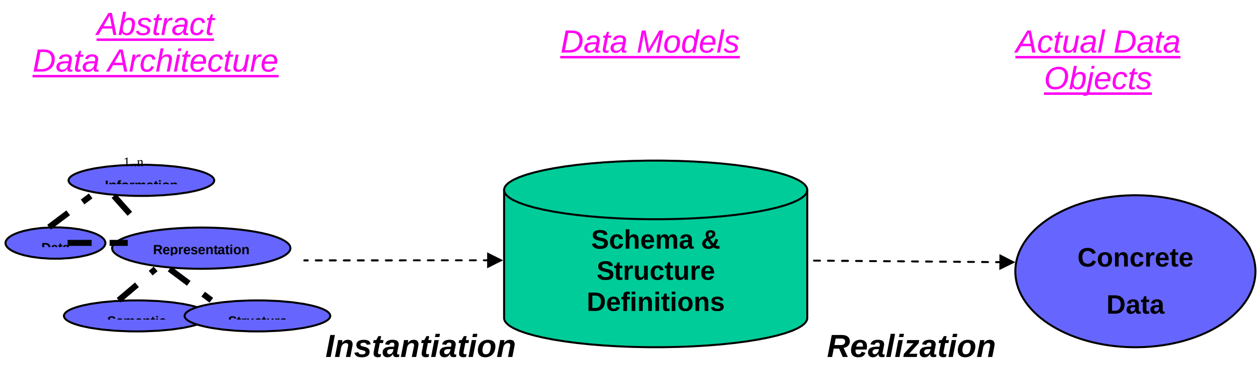

Figure 7-1: Information Object Views

Often a separate description of an Information Object is required to interpret it. This data model or metadata may be in the form of structure definitions within a program, schema definitions in a database, or metadata stored in some other form.

A further level of abstraction used with Information Objects is the data architecture, an artifact that describes the data elements and their relationships. This may be stored in a machine accessible format or it may just be defined in a document. Information Objects may also be associated with an Ontology, which describes in more detail the relationships among a broad set of Information Objects, i.e., is related to, is part of, or is used by. Increasingly, mechanisms are being used to permit machine access to all these levels of abstraction.

7.4 ATTRIBUTES OF INFORMATION OBJECTS

The attributes of Information Objects are shown in figure 7-2. Information Objects are exchanged among Functional and Communications Objects, where they may undergo various forms of transformations.

Information Objects have schema that describe their structure, rules for use and transformation, and policies on access and permanence. Unlike most other objects considered in this document, Information Objects do not have input or output interfaces.

NOTE – Detailed descriptions of these views and the means for expressing them will be supplied in a separate Information Architecture Reference Model document. This document also provides details on the Information Management Functional Objects.

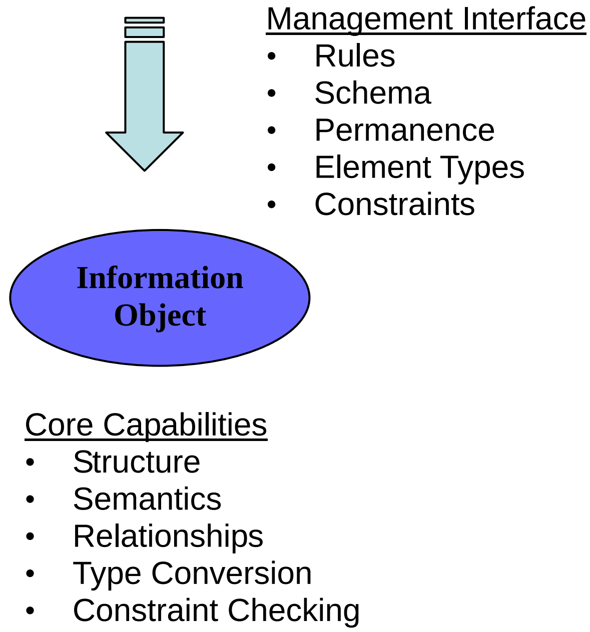

Figure 7-2: Attributes of Information Objects

7.5 EXAMPLE OF SPACE DATA SYSTEM FUNCTIONS WITH INFORMATION VIEWPOINT

Figure 7-3 shows the relationship between some typical space data system Functional Objects and the Information Objects that they exchange. This example shows a mission planning flow, where the green objects are the Functional Objects and the striped blue objects are the Information Objects.

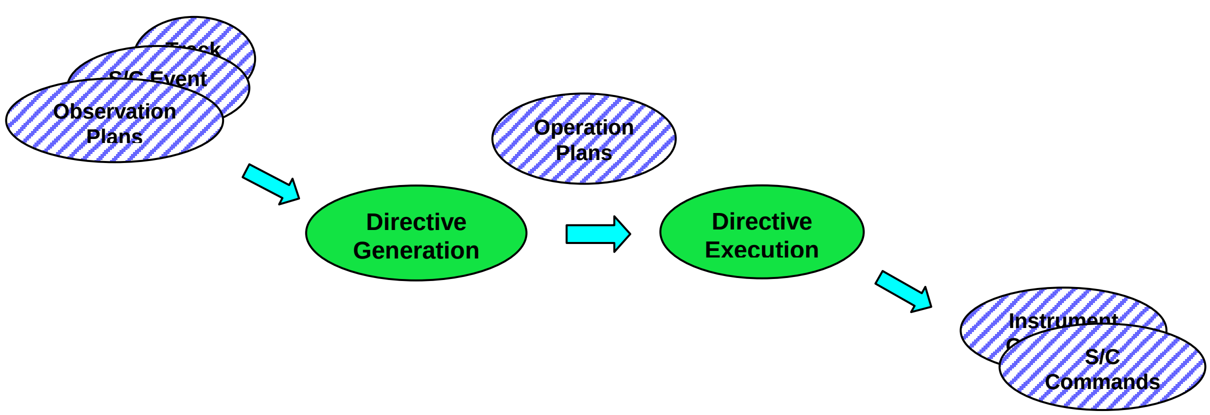

Figure 7-3: Example of Functional Objects with Information Objects

8 COMMUNICATIONS VIEWPOINT

8.1 OVERVIEW

The motivation for the Communications Viewpoint is that there are layered sets of communications protocols that support communications among the functional elements. These protocols need to meet the requirements imposed by the connectivity and operational challenges. The Communications Viewpoint is used to address these aspects of space data systems.

NOTE – This Viewpoint is both an Engineering and Technical Viewpoint of RM-ODP, but one which is important for Space Data Systems because of the need for specialized protocols to deal with the challenges we exposed in the Connectivity View. It is also the first of the Viewpoints where we have addressed the lower layers of the ISO seven-layer communications stack rather than the upper ones. In some sense, all of this is orthogonal to the upper layer viewpoints, where we have multiple perspectives of a distributed system. Here we are really talking about protocols and the layered approach used to develop them.

8.2 CONCEPTS

The **Communications Viewpoint** is an engineering and technology view of a space data system that focuses on the protocols and mechanisms of information transfer performed by that system.

This viewpoint, which is orthogonal to the first four viewpoints, provides details on layers one through five of the ISO seven-layer model. The first four viewpoints are directly related to the top, or application layer, of the model and to the Information Viewpoint or representational layer of the model.

In the Communications Viewpoint, a space data system is depicted with Communications Objects. These are often shown overlaid on the Connectivity View (Nodes and Links) and with Functional Objects, to show how these functions actually communicate with one another.

A **Communications Object** performs actions to exchange or transfer data in a space data system (as distinguished from a Functional Object that generates or processes data).

Communications Objects are used to support interactions between two Functional Objects or among a group of Functional Objects that are contained in separate Nodes.

There may be Nodes in space data systems that have only Communications Objects (without Functional Objects). A Router or Bridge is an example.

NOTE – Management approaches that are relevant to the Communications Viewpoint will be addressed in a later issue of this document. Security issues will be identified here, but dealt with in detail in the Security Architecture document.

8.3 COMMUNICATIONS OBJECTS

Table 8-1 shows typical Communications Objects used in space data systems.

Table 8-1: Typical Communications Objects

Communications Objects	Type	Description
Space Message Transfer	Messaging	Provides message transfer services between functions
CCSDS File Delivery Protocol (CFDP)	File transfer protocol	Transfers files over one or multiple space links
SCPS File Protocol	File transfer protocol	Transfers files over space links
File Transfer Protocol (FTP)	File transfer protocol	Transfers files over Internet protocols
SCPS Transport Protocol	Transport protocol	Provides end-to-end communications over space links
Transmission Control Protocol (TCP)	Transport protocol	Provides end-to-end communications in Internet
Space Packet Protocol	Network protocol	Provides routing through a network involving space links
SCPS Network Protocol	Network protocol	Provides routing through a network involving space links
Internet Protocol	Network protocol	Provides routing through Internet
TM Space Data Link Protocol	Data link protocol	Provides communications over a point-to-point space link
TC Space Data Link Protocol	Data link protocol	Provides communications over a point-to-point space link
AOS Space Data Link Protocol	Data link protocol	Provides communications over a point-to-point space link

Communications Objects	Type	Description
TM Synchronization and Channel Coding	Channel coding	Provides mechanisms for data synchronization and error control
TC Synchronization and Channel Coding	Channel coding	Provides mechanisms for data synchronization and error control
Proximity-1 Space Link Protocol	Data link + physical protocol	Provides communications over a point-to-point space link
Point-to-Point Protocol (PPP)	Data link protocol	Provides communications over a point-to-point link for Internet protocols
CCSDS RF & Modulation	Physical protocol	Physically transmits and receives signals over a space link

8.4 ATTRIBUTES OF COMMUNICATIONS OBJECTS

The attributes of Communications Objects are shown in figure 8-1.[1]

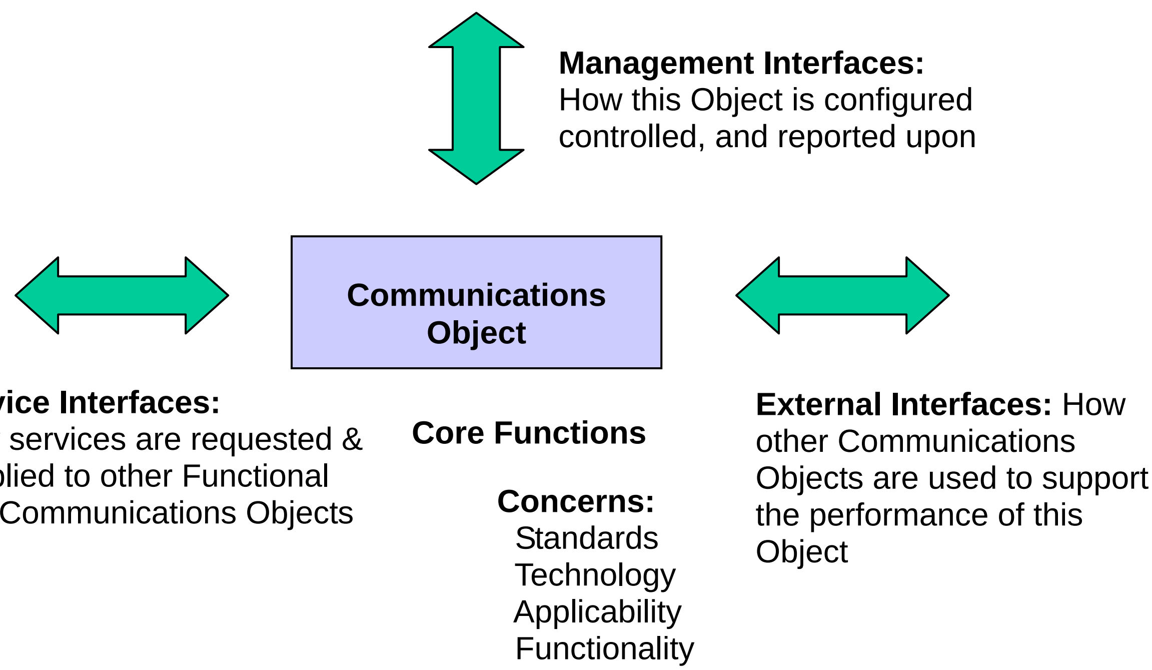

Figure 8-1: Attributes of Communications Objects

[1] Detailed descriptions on the interfaces of communications objects introduced above will be provided in a later issue of this document.

The interfaces of a Communications Object with Functional Objects or other Communications Objects are described by requests, indications, responses, and confirmations.

Communications Objects communicate with peer Communications Objects, either directly or indirectly through other lower-layer Communications Objects. The interactions between peer Communications Objects are described by exchanges of Protocol Data Units (PDUs), and the activities of the Communications Objects, in response to PDUs, are most often described as a state machine. Activities within a Communications Object may also be triggered by events such as timers.

8.5 EXAMPLES OF SPACE DATA SYSTEMS DESCRIBED WITH COMMUNICATIONS VIEWPOINT

Figure 8-2 shows Communications Objects, which support the communications between two Functional Objects (Data Acquisition and Data Monitor), which are contained in two Nodes (Payload and Ops Center) through two other typical Nodes (which are the on-board C&DH and Ground Station).

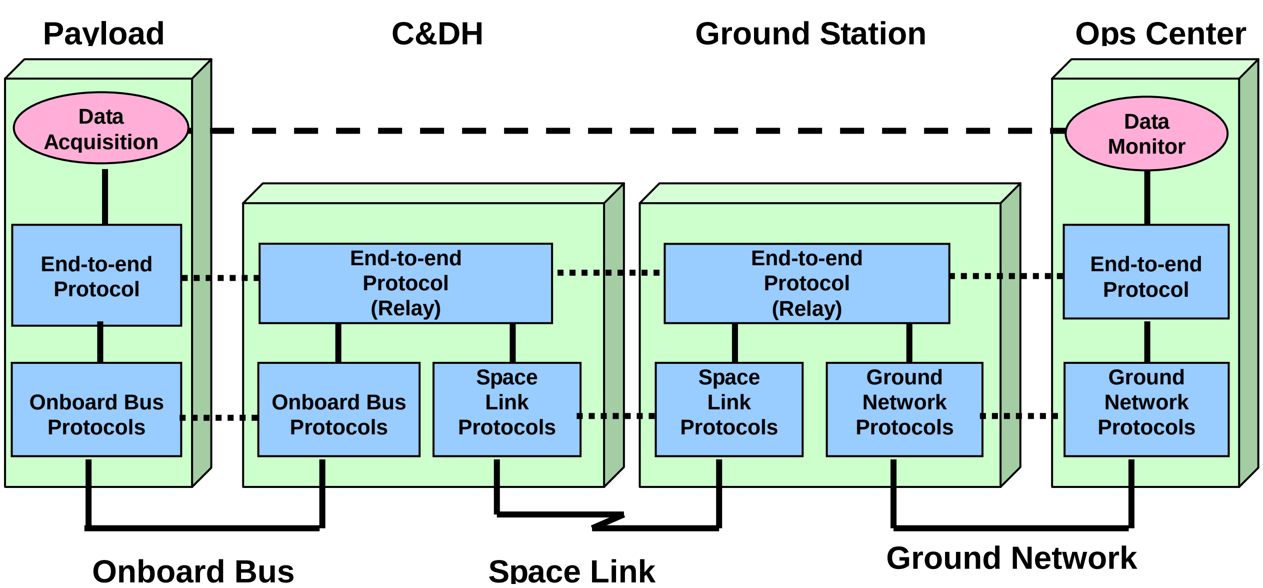

Figure 8-2: Example of Communications Viewpoint

8.6 SECURITY ISSUES IN THE COMMUNICATIONS VIEWPOINT

Certain elements of implementing data system security may be allocated to the Communications Viewpoint. These will typically include: network layer security, link layer encryption, spread spectrum and related jamming avoidance approaches.

ANNEX A

FORMAL METHODS AND TOOLS

Overview

As noted in the Introduction, one of the primary motivations for the RASDS is to provide a kit of architect's tools that domain experts can use to describe and construct many different specific complex space system architectures. The RASDS can be used very effectively in its current form to provide a vocabulary for describing systems architectures, viewpoints from which to examine them, and guidelines for concerns at each viewpoint and issues to consider.

The methods that we have identified are derived from the RM-ODP, and adapted as necessary to deal with the realities of flying systems in space. This methodology must be validated and this work has only just begun. The RM-ODP itself has developed some formal methods, but tools that implement the required functionality for this method are not yet mature.

RASDS can be used in its present form to describe space data systems. However, in order for it to be most useful for large scale systems design, tools are required that will permit the ready creation of these system descriptions and that will automatically maintain the complex relationships between objects as seen from different viewpoints.

Developing a comprehensive tool suite is a major undertaking in its own right. We wish to leverage broadly supported processes and tools wherever possible, in order to minimize development costs and to minimize the costs of adoption. At present, we have identified two possible approaches that will utilize current active developments that support software and system engineering: UML 2.0 and SysML. Requirements on these environments and tools, and specifications on the selected formal methods, have been described in separate documents.

ANNEX B

GLOSSARY AND ACRONYMS

Abstract Data Architecture Meta-Models - Models for Specification and Standardization of Data Elements (e.g. ISO/IEC 11179, DEDSL)

Action - Something that happens within an object, either with or without participation of another object. An interaction is an action performed with participation of another object.

Architecting - The process of creating an architecture specification. It is both a science and an art.

Architecture - The concepts and rules that define the structure, semantic behavior, and relationships among the parts of a system, a plan of something to be constructed. It includes the elements (entities) that comprise the thing, the relationships among the elements, the constraints that affect those relationships, a focus on the parts of the thing, and a focus on the thing as a whole.

Behavior - A set of actions performed by an object for some purpose.

Communications Object - Perform actions to exchange or transfer data in a space data system. Communications Objects are used to support interactions between two Functional Objects or among a group of Functional Objects that are contained in separate Nodes. Communications Objects implement protocols.

Communications Viewpoint - An engineering and technology view on a space data system that focuses on the protocols and mechanisms of information transfer performed by that system.

Community – An entity (e.g., Earth Science) that may exist within one Space Enterprise or across multiple Space Enterprises. It is distinguished by being bound by common objectives and relationships and offers a set of resources that are sharable within the Community and with other Communities.

Connectivity View - An engineering view on a space data system that focuses on the Node and Link view of a system, the physical connections among Nodes, their physical and environmental constraints, physical dynamics, and (optionally) the distribution of functions on Nodes.

Data Architecture – Models of the structure and relationships among the data elements used within a system

Data Models - Schema & Structure Definitions

***Data Objects* –** Passive Information objects, may be either a Physical Object or a Digital Object.

Domain - A Community that is under single organizational, administrative or technical control (e.g. NASA Code Y). A domain may have resources, policies, access control, and possibly quality of service constraints. A Domain may be subdivided into Sub-Domains. Multiple Domains may be collected into a Federation.

***Enterprise Object* –** An organizational entity that is governed by a single authority that has its own objectives and policies to operate the object. An Enterprise Object may be a component of another larger Enterprise Object. Enterprise Objects may participate wholly or in part in other Enterprise Objects.

***Enterprise Viewpoint* –** A view of a space data system focuses on the community, purpose, scope and policies for that system. This viewpoint includes organizations as well as the Enterprise Objects that have assigned roles, responsibilities, and interactions.

Entity - Any concrete or abstract thing of interest. For example, an entity may be a physical instrument, a computer, a piece of software, or a set of functions performed by a system.

Federation - A Community consisting of multiple Domains (e.g., CEOS or CCSDS) that come together to share resources while retaining their autonomy over those resources. Federations are bound by negotiated agreements. A Federation may only include some members of a Domain or Sub-Domain (e.g. a particular Earth Observing project). Members of a Federation agree to rules for sharing resources and for joining and/or leaving the federation.

***Function* –** The set of actions or activities performed by some element to achieve a goal.

***Functional Object* –** An object that performs Functions to achieve a goal of a space data system or to support actions of another Functional Object, and transfers, generates or processes data in performing those actions.

Functional Viewpoint - A view on a space data system that focuses on the structure of the functions performed by that system and their behavior, and on the interactions among the functions. This includes functional objects, the logical connections between them, their interactions and logical interfaces.

***Information Management Functional Objects* –** Active functional elements that support the location, access, delivery, and management of passive Information Objects. These Information Management Functional Objects are a class of Functional Objects, but they are considered along with Information Objects because of the close relationship between them.

Information Objects – Data, along with the necessary structure and syntax to allow use of these Objects. May also have associated ***meta-data***, i.e. information about the meaning of the data, as well as the relationships among Data Objects, rules for their use and transformation, and policies on access.

Information Package - A primary Information Object and associated supporting information that is needed to use the Information Object. The Information Package has associated Packaging Information used to delimit and identify the primary Information Object and Supporting Information.

Information Viewpoint **–** A view of a space data system that focuses on the information used by that system. This includes structural (syntactic) and semantic views of the information, the relationships among information elements and rules for their management and transformation.

Instantiation **–**Data Models must be instantiated as real information objects in order to participate in system activities

Interface - A set of interactions performed by an object for participation with another object for some purpose. An interface is therefore a kind of behavior of an object.

Link - The locus of relations among Nodes. It may be implemented by a wired connection or with some RF or optical communications media. It may periodically become inactive because of the motion of a Node, or lack of availability of communications resources, for example.

Model - A formal specification of the structure and/or function of a system. All models are abstractions; abstraction is the suppression of irrelevant detail.

Node - A physical Object that has some well understood, possibly rapidly moving, location and has a set of resources to support the activities of the Functional, Information and Communications Objects contained in that Node. A Node may be composed of two or more (sub-)Nodes.

Object - An abstract model of an entity. Objects may be physical or logical. An Object may be composed of two or more Objects. The behaviors of the ***Composite Object*** are determined by those of the component Objects. Component Objects may sometimes be called Components.

Policy - A set of guidelines and constraints on the behaviors exhibited by the objects in the system.

Protocol **–** The description of the state machines within a Communications Object and the PDUs that are exchanged between these entities.

Protocol Data Units (PDU**) –** The actual data objects that are exchanged between peer protocol entities. May contain data, metadata, and requests for action.

Realization - Abstract data architecture elements must be realized as Data Models and stored in some sort of repository

***Relationship* –** Describes the way that two or more entities can be associated with each other.

Resource - An entity that has some role and performs some action within a system. A resource may be shared by more than one activity.

***Role* –** A role is the way that an entity participates in a relationship. A set of behaviors and actions of an object that is associated with the relationship of an object with other objects.

Semantics - rules by which syntactic expressions are assigned meaning

Service - A provision of an interface of an object to support actions of another object.

Space Enterprise - A top-level autonomous entity (e.g. NASA) that is dedicated to the exploration and/or exploitation of space. It has its own objectives, resources and policies and it is not a component of any other Space Enterprise.

***Standard* –** A standard is a document containing a formal specification, which defines and governs functions and protocols at interfaces of a data system. It describes in detail the capabilities and establishes the requirements to be met by interfacing subsystems to achieve compatibility.

***View* –** 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. In RASDS some new combined Views will be defined that correspond to more than one Viewpoint.

***Viewpoint* -** 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 establishes the purpose and audience for a View and the techniques or methods employed in constructing a View.