Reference Architecture for Space Data Systems

Authors:

Peter Shames
Mgr, JPL Information Systems Standards Program
Jet Propulsion Laboratory (JPL)
4800 Oak Grove Drive
Pasadena, CA 91109
818 354-5740

Dr. Takahiro Yamada
Institute of Space and Astronautical Science (ISAS)
3-1-1 Yoshinodai
Sagamihara 229
Japan

Focus Issues:

- Architectural representation and analysis
- Standards and interoperability
- Space and ground communication architectures

ABSTRACT

Architectures for terrestrial data systems that are built and managed by a single organization are inherently complex. In order to understand any large-scale system architecture, and to judge its applicability for its nominal task, a description of the system must be produced that exposes a number of distinct viewpoints. At a minimum such descriptions will typically cover the uses that are to be made of the system, the functions that the system performs, the elements that compose the system, the information that flows among these elements, and the specific technologies that are integrated into the system.

There are a variety of approaches that can be used to describe such system architecture and to capture these various viewpoints and their relationships. UML is a powerful and currently popular tool for describing software systems, but it does not include all of the constructs for readily describing distributed system architectures and hardware. A standard called Reference Model for Open Distributed Processing (RM-ODP) has been developed within ISO and ITU to provide a common way to describe large, multi-organization systems. This modeling approach provides views on a system that go from the organizational (Enterprise) to the abstract (informational, computational), to the more concrete (Engineering, Technology).

Within the CCSDS Architecture Working Group we have adapted RM-ODP to describe large, multi-national, space data systems. These systems exhibit all of the complexities of typical terrestrial systems, but are frequently compounded by involvement of several space agencies, some unusual organizational cross-support arrangements, and use of contractors in a number of roles. We also must deal with the complexities of operating systems in space, including all of the physical constraints and challenges that that environment brings. The most fundamental challenge is the physical space environment (motion, obscuration, long round trip light times, episodic connectivity, low signal strength, asymmetric data paths) which constrains how these systems are engineered and operated, and often requires different protocols for communications than those that can be used terrestrially.

We have produced a methodology, based upon RM-ODP, which provides the necessary concepts and notation for describing these complex space data systems. The reference architecture is intended for use by two different, but related, user communities: the system users and the system and standards developers. The system users are typically concerned with what is "outside" the box that is the system. They want to know what it does for them, what the interfaces look like, and how they can use its services, but are not particularly interested in how it provides these services. On the other hand, the developers of these systems, and the developers of the standards that ensure interoperability and cross-support, are vitally concerned with how the system provides these services for users and with how elements made by one organization can interoperate with, and provide cross support, to elements developed by another

organization. Our approach clearly identifies these two user viewpoints and describes their relationship as well.

The approach is intended to be general enough to permit description of civilian, military, and commercial space data systems, the spacecraft, ground systems, processing and communications resources, and organizational arrangements. We will describe the methodology and the set of viewpoints that we have derived, and describe their relationship to RM-ODP.

There is related work to identify means to capture these architectures and the behavior of the described elements in a machinable way, such that we can reason about the completeness and accuracy of the system as described. As a way of assessing performance and exploring design trades we hope to eventually be able to simulate at least the coarse grained overall behavior of such systems based upon their descriptions. The granularity of such models is intended to be scalable to permit finer grained detail where it is required.

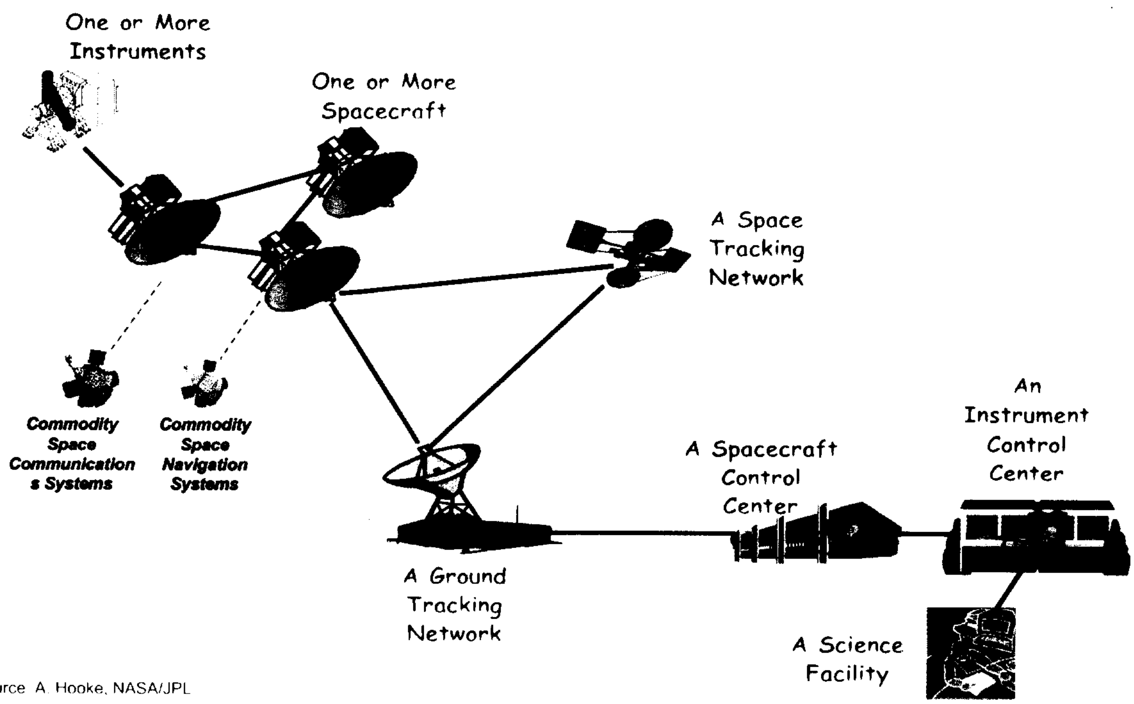
CCSDS
Architecture Working Group

Space Data Systems Reference Architecture

2 May 2003

Peter Shames, NASA/JPL, Takahiro Yamada, ISAS

A Physical View of a Space Data System

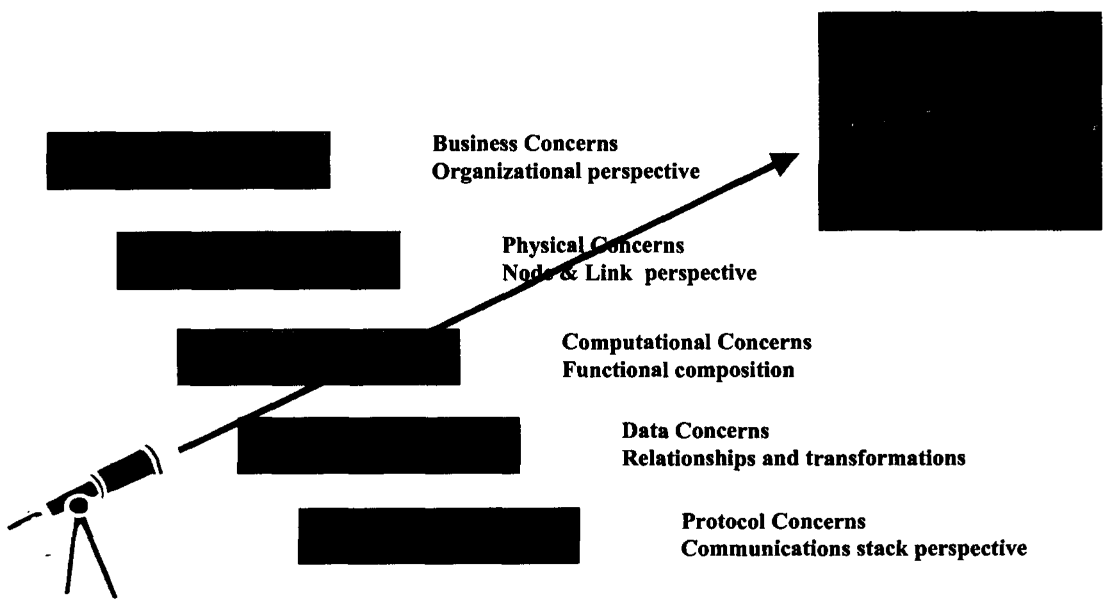


Source A. Hooke, NASA/JPL

Reference Architecture Purpose

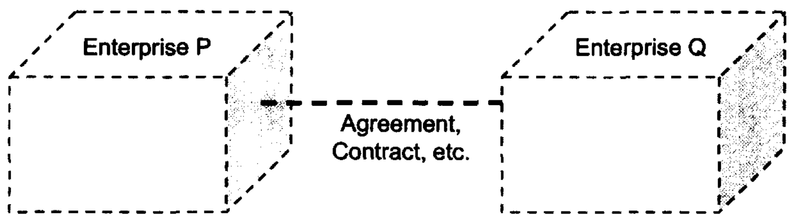
- Establish an overall CCSDS approach to architecting and to developing domain specific architectures
- Define common language and representation so that challenges, requirements, and solutions in the area of space data systems can be readily communicated
- Provide a kit of architect's tools that domain experts will use to construct many different complex space system architectures
- Facilitate development of standards in a consistent way so that any standard can be used with other appropriate standards in a system
- Present the standards developed by CCSDS in a systematic way so that their functionality, applicability, and interoperability may be clearly understood

Space Data System Several Architectural Viewpoints

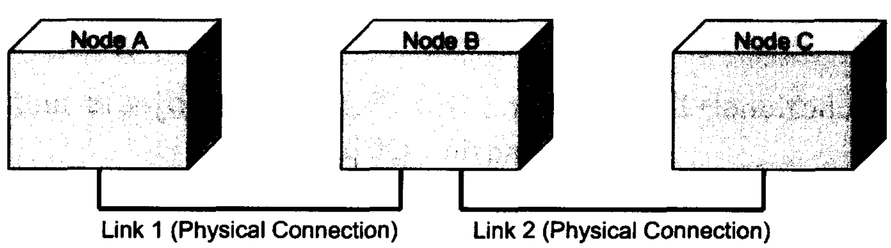


Derived from: RM-ODP

Enterprise View (Enterprise Objects)

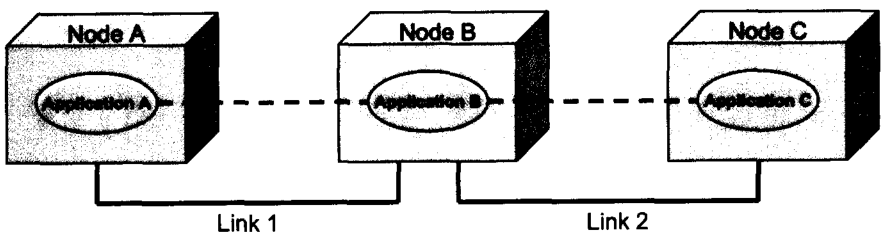


Connectivity View (Nodes and Links)

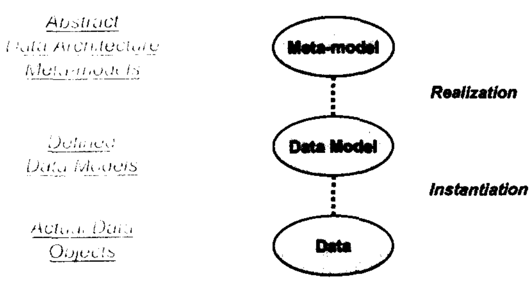


Functional View (Functional Objects)

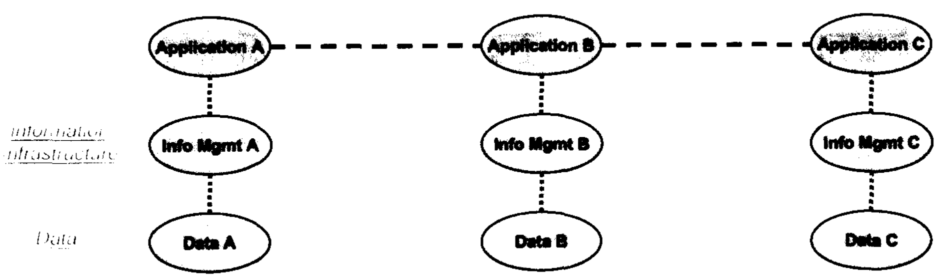
Connectivity+Functional View (Nodes, Links and Functional Objects)



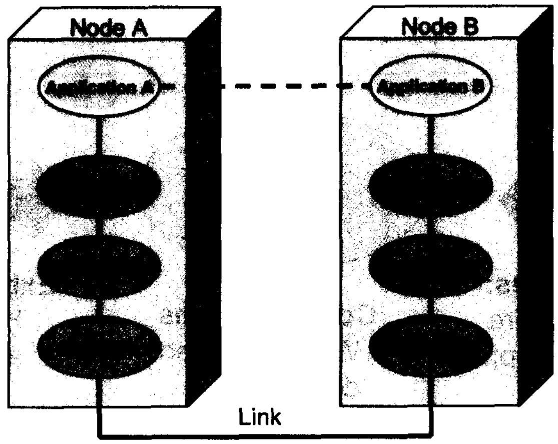
Information View (Information Objects)



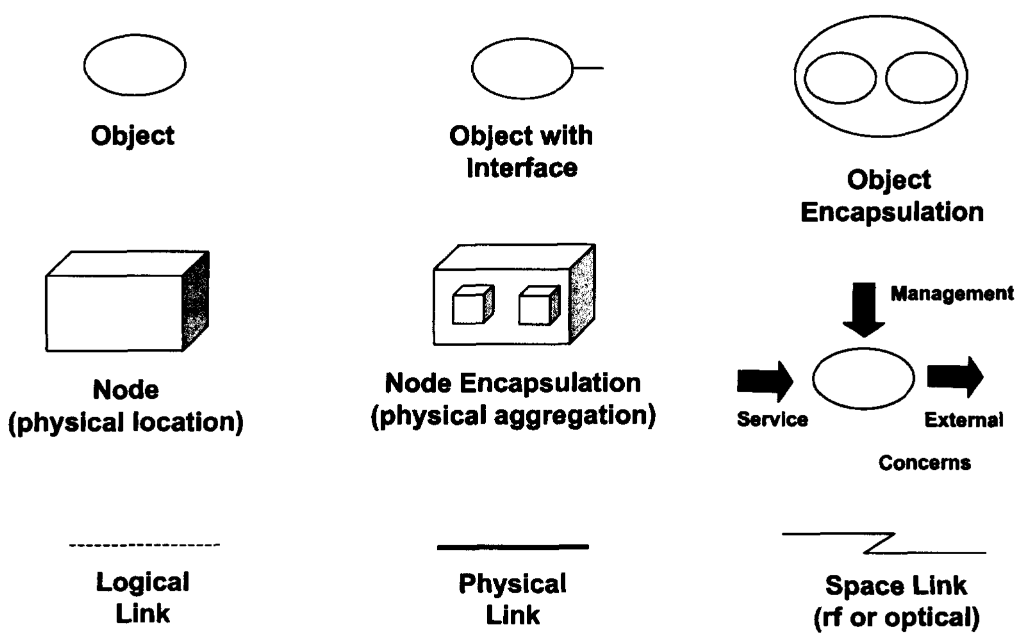
Functional+Information View (Functional Objects and Information Objects)



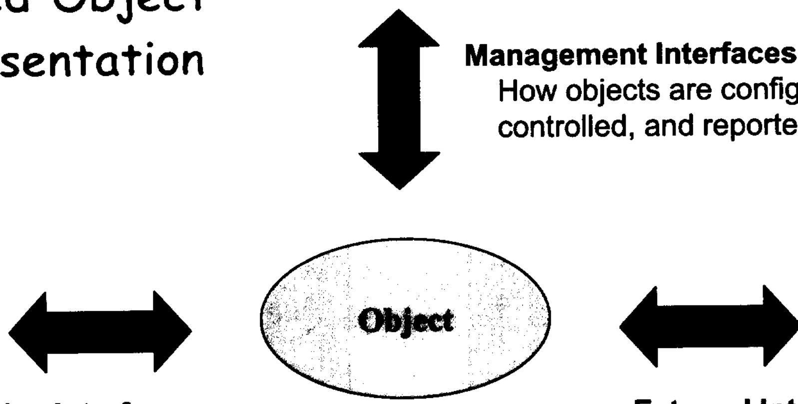
Connectivity+Functional+Communication View (Nodes, Links, Functional Objects and Communications Objects)



Space Data System Architectural Notation



Unified Object Representation



Management Interfaces:
How objects are configured controlled, and reported upon

Service Interfaces:
How services are requested & supplied

Core Functions
What the object does

External Interfaces:
How external elements are controlled

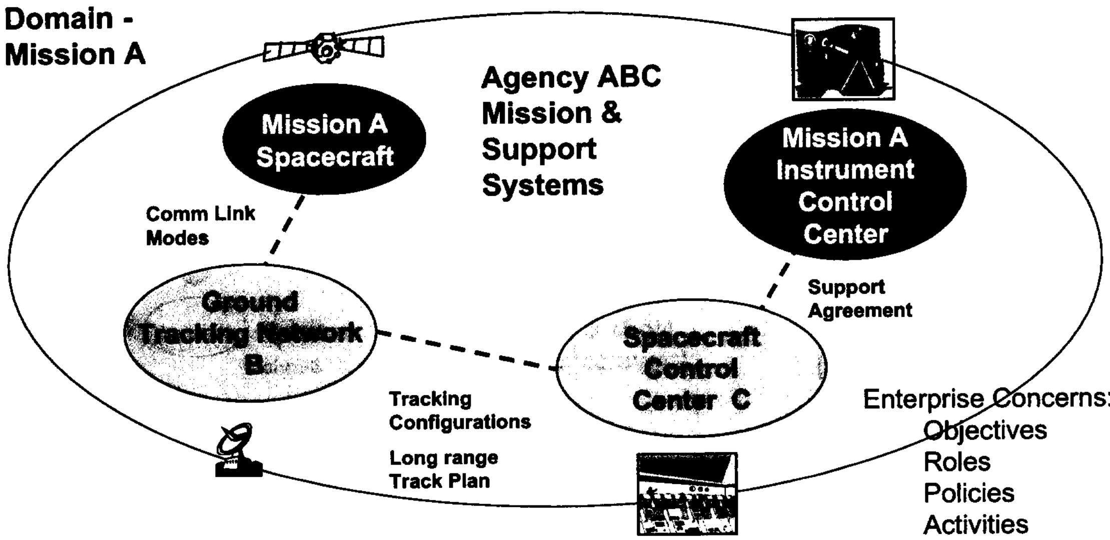
Concerns:
Issues
Resources
Policies

Enterprise View

Single Agency Mission Domain & Enterprise Objects

Operations Planning Phase

Operations Domain - Mission A

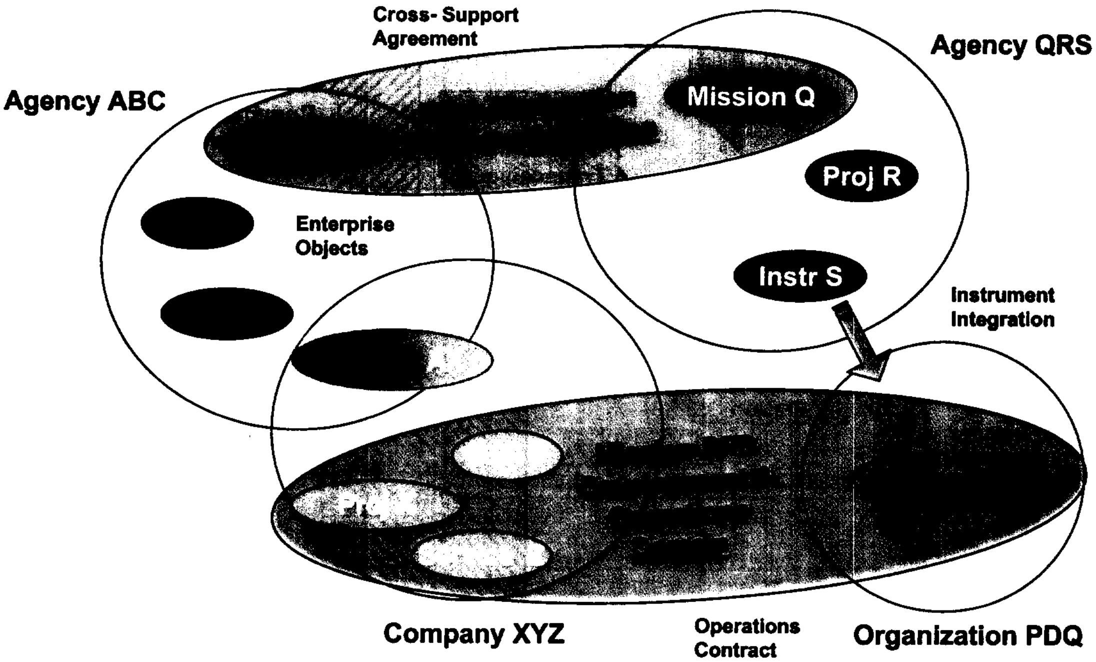


Enterprise Concerns:
- Objectives
- Roles
- Policies
- Activities
- Lifecycle / Phases
- Configuration
- Contracts

Enterprise View

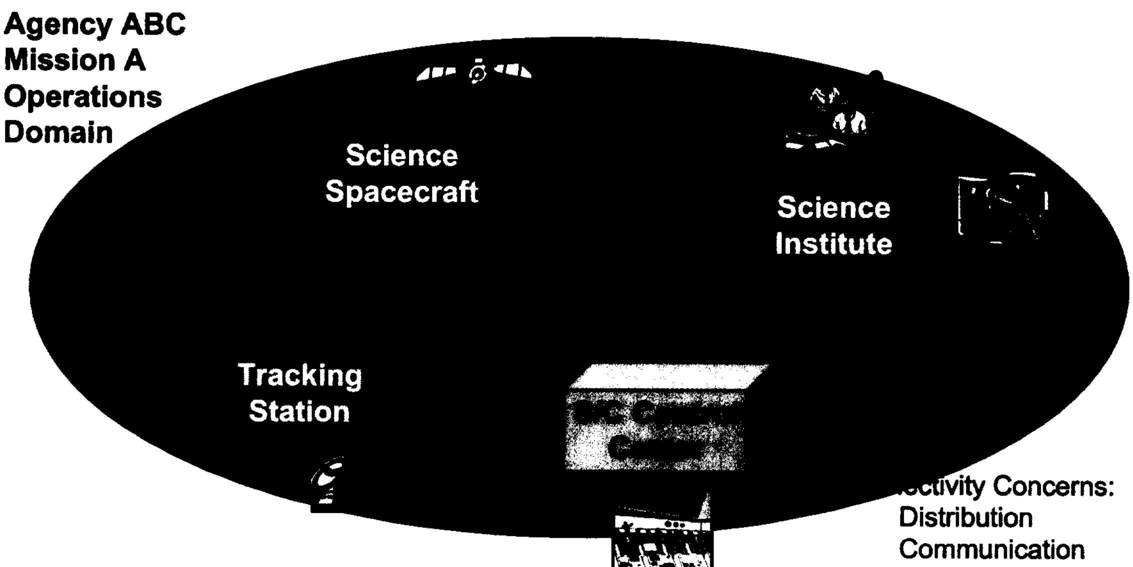
Federated Enterprises with Enterprise Objects

Planning Phase



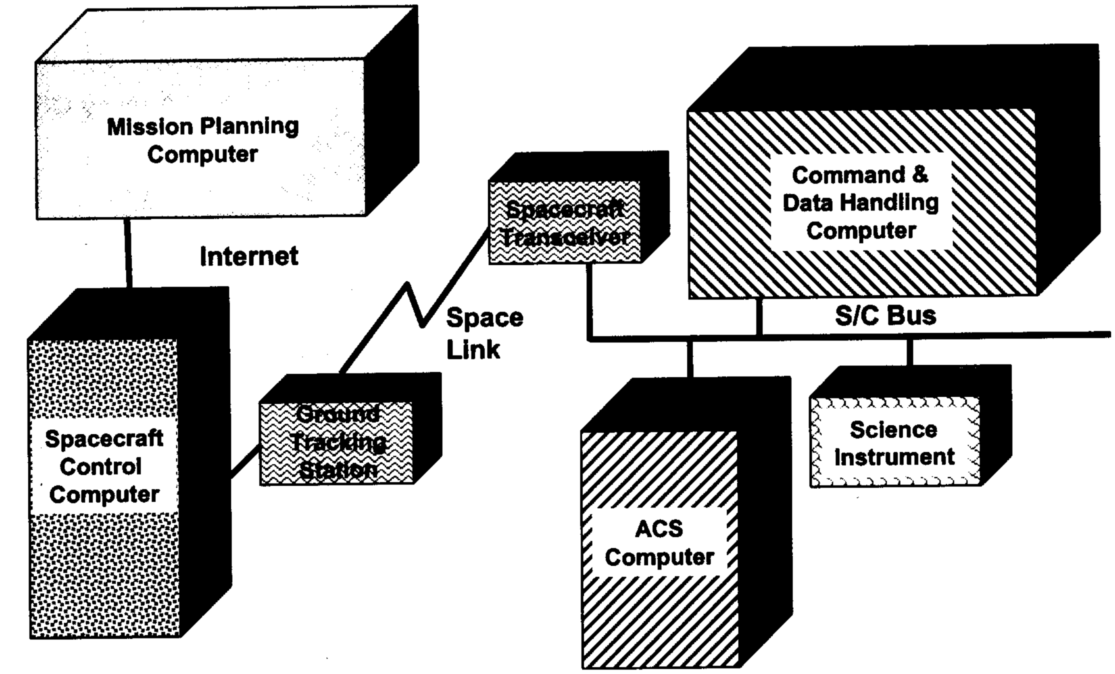
Connectivity View

Single Agency Mission Domain & Nodes

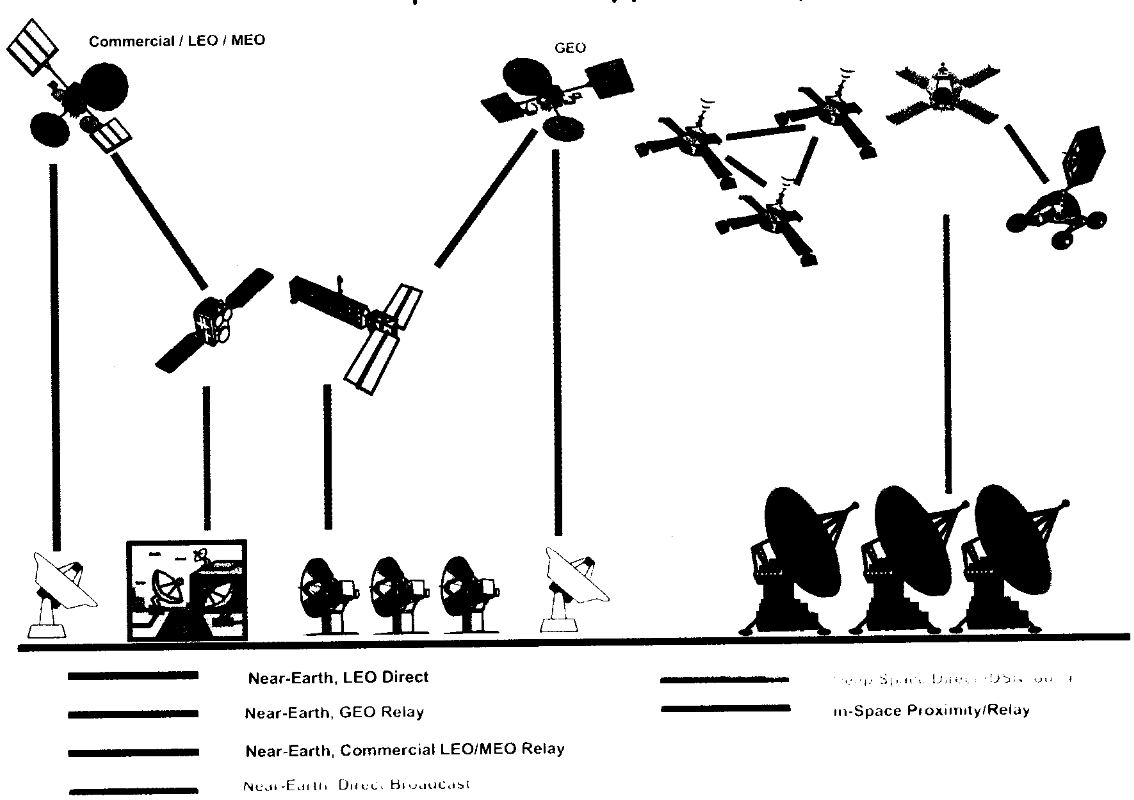


...ctivity Concerns:
Distribution
Communication
Physical Environment
Behaviors
Constraints
Configuration

Connectivity View
Nodes & Links



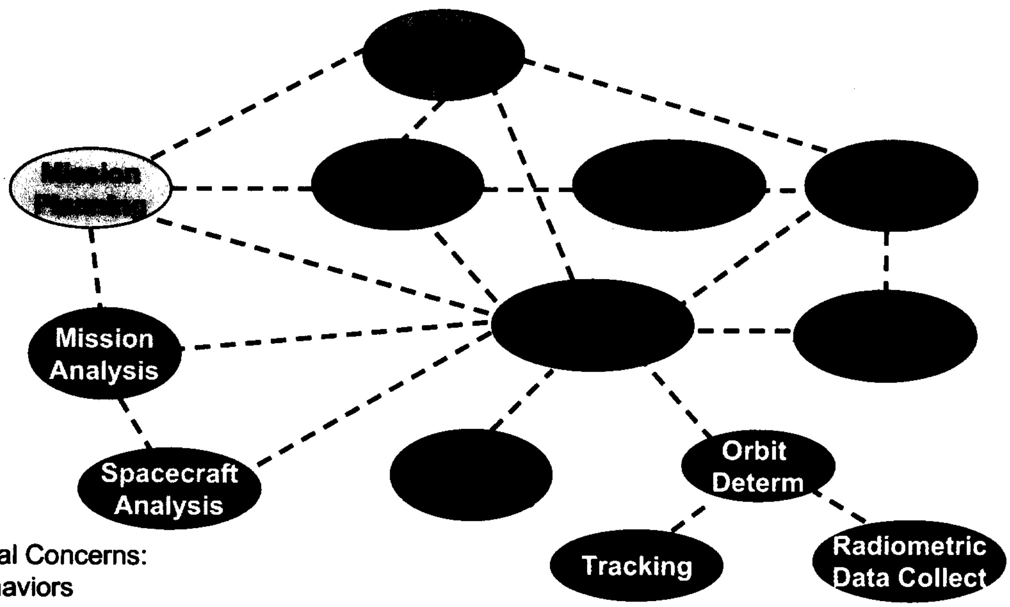
Connector Properties: Types of Space Links



Source: A. Hooke, NASA/JPL

Functional View

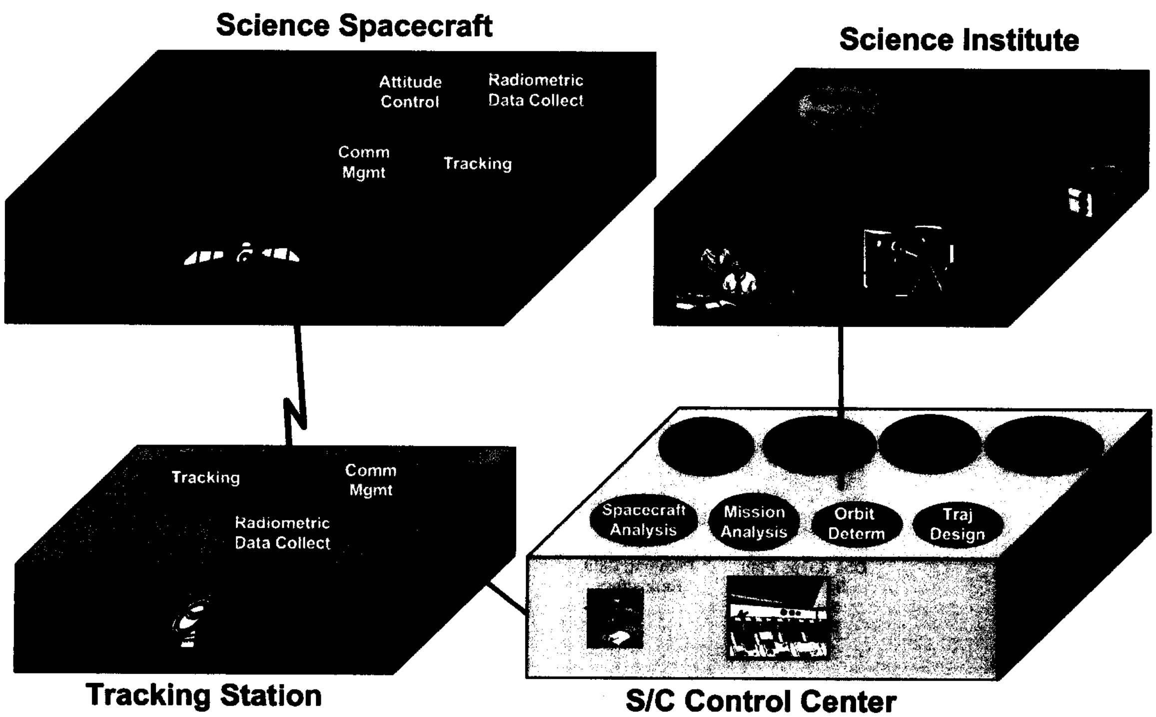
Example Functional Objects & Interactions



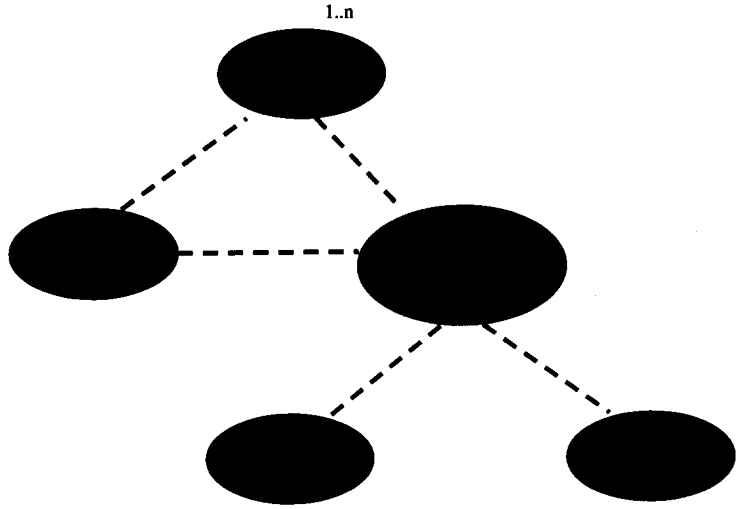
Functional Concerns:
- Behaviors
- Interactions
- Interfaces
- Constraints

Connectivity View - Redux

Mapping Functions to Nodes

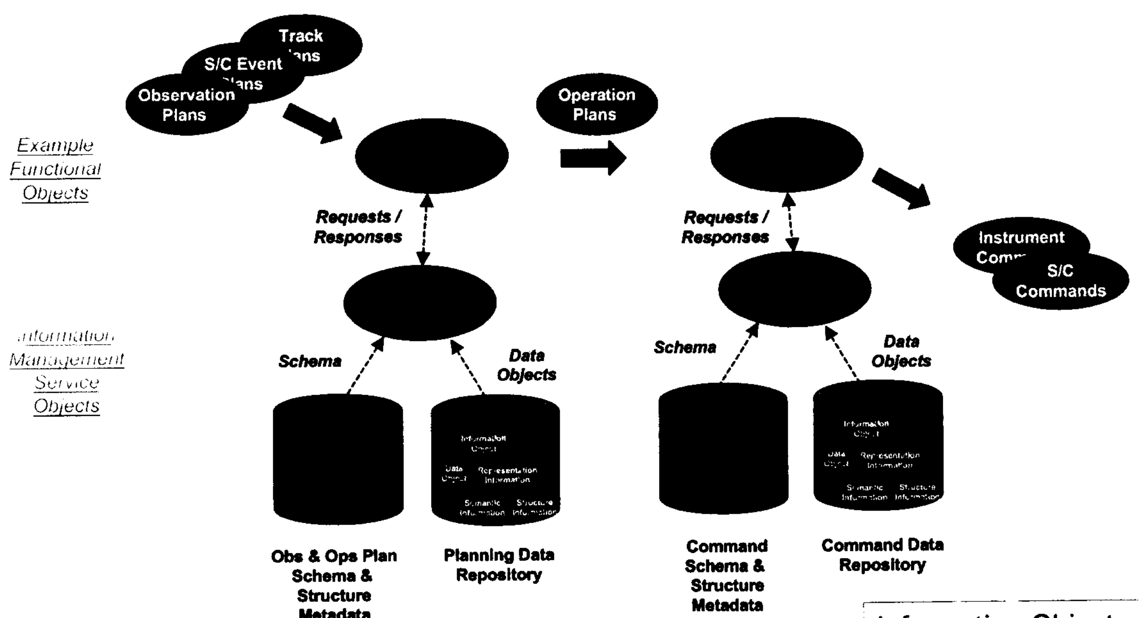


Information Object Basic Relationships



Information Concerns:
- Structure
- Semantics
- Relationships
- Permanence
- Rules

Information Objects Relationship to Functional View



Schema & structure may be embedded in the function itself

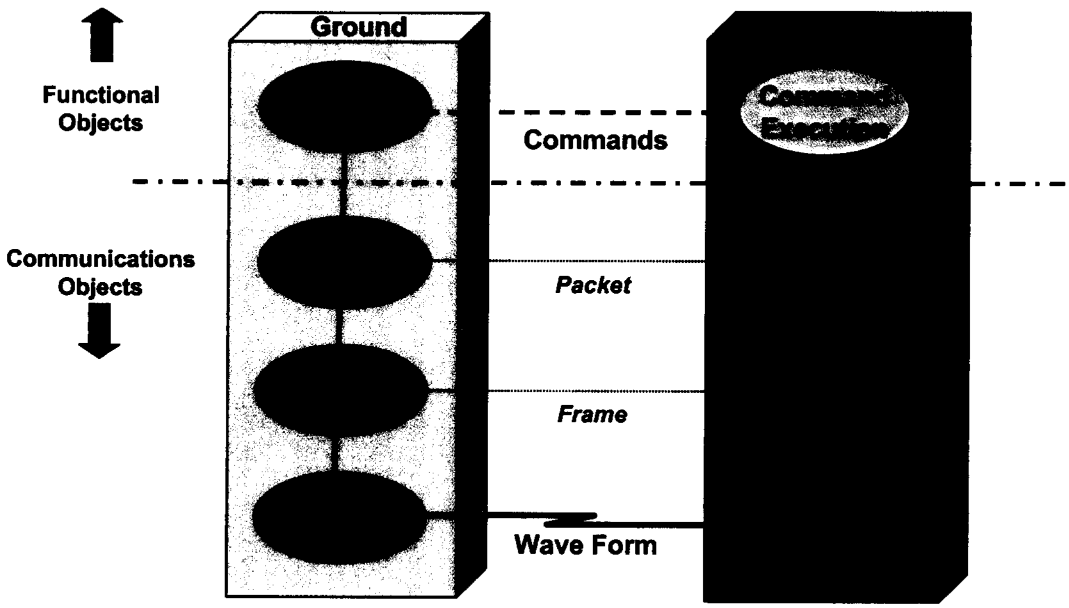
Actual data objects that are exchanged between Applications at run time

Core Schema
Common Schema
Extension Schema

Information Objects are exchanged by Functional Objects

Communications View

Simple Example



Communications View

Onboard Decomposition

(Real-time Commanding)