

Toward a Framework for Modeling Space Systems Architectures

Peter Shames, Joseph Skipper

NASA, JPL

ABSTRACT

There recently have been a number significant effort to develop powerful and extensible approaches for describing a general class of software intensive system architectures. These approaches are typically focused upon architectures of terrestrial systems and they range from generally applicable, but formalized, approaches like the System Engineering Modeling Language (SysML), to more focused approaches such as the Unified Modeling Language (UML) for RM-ODP (Reference Model for Open Distributed Processing). In alignment with the recommendations made in the IEEE 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems, both of these approaches intend to support an appropriate set of viewpoints for developing system architectural descriptions.

All of these standard architectural approaches are intended to describe large-scale terrestrial data systems that are inherently complex, but are typically fixed in one place and often designed and built by a single organization. In the world of space systems there is an even higher level of complexity in that these are most often multi-organizational creations that are best characterized as systems of systems. In further contrast to terrestrial systems the most challenging elements of these systems, the spacecraft, are not fixed in place, but are flying through space at high velocity, must use specialized ground and space communications assets and protocols, are often at great distance from the Earth and are frequently out of contact with their control centers.

These attributes of space communications systems drive architectural complexity and require consideration of issues that are not typical in terrestrial systems. The set of viewpoints that the existing standard approaches define are not completely adequate to the task of describing space systems. Work has been done during the last couple of years to model space data systems using a methodology called the Reference Architecture for Space Data Systems (RASDS) that is derived from RM-ODP. This modeling approach augments the RM-ODP viewpoints on a system, that include the organizational (Enterprise), the abstract (informational, computational), and the more concrete (Engineering, Technology), by adding ones that deal directly with space communications and protocols, physical element connectivity, and their interactions with the environment. The efficacy of RASDS has been demonstrated by its use in several NASA projects to describe the end to end architectures of their spacecraft data systems.

Recent work has been done at JPL in an internal research and development project called Model Based Engineering and Design (RAMSS) to extend this RASDS approach to capture all of the other physical aspects of space systems in an extended model. Modeling the design of space systems necessitates inclusion of one or more viewpoints that deal with the rest of the physical attributes of these systems and their interaction with the environment. These other attributes include mass, power, propulsion, thermal, structure and dynamic control, in addition to the gravitational and environmental aspects already in the RASDS connectivity viewpoint.

This conceptual approach is intended to be general enough to permit description of civilian, military, and commercial space systems, the spacecraft physical and logical design, ground

systems, processing and communications resources, and organizational arrangements. It does not, at this point, model the operational interactions captured in the Rational Unified Process (RUP-SE) nor the related Operational Viewpoint modeled in the DOD Architecture Framework (DoDAF). It also does not provide much guidance on how to describe software architectures in detail. Instead it focuses upon how to represent the end to end system architectural elements and their logical and physical interfaces and interactions.

In this paper we will describe this extended RASDS / RAMSS methodology, the set of viewpoints that we have derived, and describe their relationship to RM-ODP. While this methodology may be directly used in a variety of document driven ways to describe space system architecture, the real power of it will come when there are tools available that will support full description of system architectures that can be captured electronically in a way that permits their analysis, verification, and transformation.

This conceptual approach has been evaluated using an existing system engineering modeling tool that provided partial support for the full set of required viewpoints. This tool allowed us to capture an example spacecraft architecture and the behavior of the described elements in a machinable way. This has allowed us to simulate at least the coarse grained overall behavior of this system based upon its description, to provide means to assess some elements of end to end performance and explore design trades. This also exposed some of the limitations in existing tools and their ability to support extension to include the required conceptual viewpoints. Future work will explore how far we can get with capture and analysis of these formalized design descriptions and evaluation of completeness and correctness based upon these system models.

1. Introduction

Since the early 1990's there have been a number of efforts to define standard approaches for describing and analyzing system architectures. Many of these have been funded by the DoD, but some have sprung from international or national efforts in ISO or the IEEE. The most relevant of these, the IEEE Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE-Std-1471-2000) provides some very useful definitions and guidelines for what a system architecture is and for the use of viewpoint specifications to address the identified set of stakeholder concerns.

The scope of this recommended practice encompasses those products of system development that capture architectural information. This includes architectural descriptions that are used for the following:

- a) Expression of the system and its evolution
- b) Communication among the system stakeholders
- c) Evaluation and comparison of architectures in a consistent manner
- d) Planning, managing, and executing the activities of system development
- e) Expression of the persistent characteristics and supporting principles of a system to guide acceptable change
- f) Verification of a system implementation's compliance with an architectural description
- g) Recording contributions to the body of knowledge of software-intensive systems architecture

The IEEE 1471 specification describes the process for developing architecture descriptions under a number of scenarios, including preceded and unpreceded design, evolutionary design, and capture of design of existing systems. In all of these scenarios the overall process is the same: identify stakeholders, elicit concerns, identify a set of viewpoints to be used, and then apply these viewpoint specifications to develop a set of relevant views of the system.

Although IEEE-1471 does mention the possibility of defining system, functional, and technical views (among others), it does not go so far as to define any specific set of views nor what these might be. While all of these definitions provide useful guidance as to process and terminology, they provide little in the way of practical direction for actually defining an architecture methodology for space systems, nor do they offer pragmatic guidelines for describing system architectures, particularly space system architectures. For this we need to look to other, more domain specific, approaches that adhere to these general principles.

The ISO Reference Model for Open Distributed Processing (RM-ODP) was published in 1996 to provide a useful framework for describing the architecture and design of large scale distributed systems. The RM-ODP, also known as ISO/IEC, provides domain specific guidance that aligns with the principles defined in IEEE 1471. The RM-ODP was developed to provide a useful framework for describing the architecture and design of large scale distributed systems. Among the contributions that RM-ODP provides are the following:

- RM-ODP offers a conceptual framework and an architecture that integrates aspects related to the distribution, interoperation and portability of software systems, in such way that hardware heterogeneity, operating systems, networks, programming languages, databases and management systems are transparent to the user. In this sense, RM-ODP manages complexity through a “separation of concerns”, addressing specific problems from different points of view.

- RM-ODP offers a coordinating framework for the standardization of ODP, able to integrate current and future standards, and maintain consistency among them.

- RM-ODP provides a short, clear and explicit specification of concepts and constructs that define semantics, independent of the representation, methodologies, tools and processes used for the development of open distributed applications. RM-ODP offers a vocabulary and a common semantic framework to all the applications’ participants (from managers to users, from designers to developers), and encourages the use of formal notations for the definition of those concepts and the specification of the architecture.

A good architectural framework should allow different parts of the design to be worked upon separately if they are independent, but should clearly identify those places where different aspects of the design constrain one another. RM-ODP has seen use in the design of major terrestrial telecom systems (TINA) and other large, multi-user, distributed systems. In the telecommunications industry, TINA (Telecommunications Information Networking Architecture), defined by TINA-C (TINA Consortium), describes an architecture for the development of telecommunication applications based on the concepts defined by RM-ODP. Currently TINA provides the most widespread and accepted architecture in this field. With the

support of those technologies, building systems using the RM-ODP concepts is no longer a visionary and risky business.

While the terrestrial distributed systems that RM-ODP was designed to describe may be very large and complex space systems are even more so and they don't stand still. RM-ODP provides an excellent framework with which to tackle terrestrial systems, but additional views and viewpoints are essential for describing space systems, largely because there is an entirely new set of concerns. The Consultative Committee on Space Data Systems (CCSDS) has been working on a space domain adaptation of RM-ODP for the last few years. This is called the Reference Architecture for Space Data Systems (RASDS). The RASDS, CCSDS 311x0-R-1, is designed to address these added complexities of space data systems.

Flight elements in space systems are always in motion, whether they are in transit to their destination, in orbit around it, landed upon some remote part of the Solar System, roving some distant terrain, or on their way out of it to even more distant domains. The physical environment plays a large role because physics acts upon these systems in a way that must be modeled in our design processes and during control system design, planning, commanding and operations. Even such subtle energies as solar pressure, outgassing, and gravity must be analyzed during design and countered during operations.

Typical terrestrial assumptions about immediate and interactive communications break down when command and response round trip times rise from tens of milliseconds to tens of minutes or even hours. Communication protocols designed to work with normal communication delays break down as interaction times exceed a few seconds. Control paradigms must be reconsidered, systems autonomy rises in importance, and any notions of distribution transparency must be completely rethought.

The one new viewpoint which RASDS introduces for space systems is the Connectivity one that deals with system components (Nodes), connectors (Links), the environment within which these systems operate, and the physical interactions of the system elements with the environment. This is largely a sub-set of the RM-ODP Engineering Viewpoint, but it adds in physical aspects of space data system architectures. RASDS has also distinguished a Communications Viewpoint that is used to address the complexities of communications protocols and end to end information system (EEIS) design in space data systems. Other viewpoints in RM-ODP only require minor changes in order to be used for the purpose of describing space systems architectures.

2. The RM-ODP Reference Model

Distributed systems can be very large and complex, and the many different considerations which influence their design can result in a substantial body of specification, which needs a structuring framework if it is to be managed successfully. The purpose of the RM-ODP is to define such a framework, consisting of viewpoints, object specifications and various assertions on properties of the entities modeled in each viewpoint.

2.1 Viewpoints

Most complex system architecture specifications are so extensive that no single individual can fully comprehend all aspects of the specifications. Furthermore, we all have different interests in a given system and different reasons for examining the system's specifications. A business executive will ask different questions of a system make-up than would a system implementor. The concept of RM-ODP viewpoints framework, therefore, is to provide separate viewpoints into

the specification of a given complex system. These viewpoints each satisfy an audience with interest in a particular set of aspects of the system. Associated with each viewpoint is a *viewpoint language* that optimizes the vocabulary and presentation for the audience of that viewpoint.

The RM-ODP framework provides five generic and complementary viewpoints on the system and its environment:

- The **enterprise** viewpoint, which focuses on the purpose, scope and policies for the system. It describes the business requirements and how to meet them.
- The **information** viewpoint, which focuses on the semantics of the information and the information processing performed. It describes the information managed by the system and the structure and content type of the supporting data.
- The **computational** viewpoint, which enables distribution through functional decomposition on the system into objects which interact at interfaces. It describes the functionality provided by the system and its functional decomposition.
- The **engineering** viewpoint, which focuses on the mechanisms and functions required to support distributed interactions between objects in the system. It describes the distribution of processing performed by the system to manage the information and provide the functionality.
- The **technology** viewpoint, which focuses on the choice of technology of the system. It describes the technologies chosen to provide the processing, functionality and presentation of information.

A viewpoint is a subdivision of the specification of a complete system, established to bring together those particular pieces of information relevant to some particular area of concern during the design of the system. Although separately specified, the viewpoints are not completely independent; key items in each are identified as related to items in the other viewpoints. However, the viewpoints are sufficiently independent to simplify reasoning about the complete specification. The mutual consistency among the viewpoints is ensured by the architecture defined by RM-ODP, and the use of a common object model provides the glue that binds them all together.

2.2 Objects

RM-ODP system specifications are expressed in terms of objects. An object is a representation of an entity in the real world. It contains *information* and offers *services*. A system is composed of interacting objects.

The use of the object paradigm provides abstraction and encapsulation, two important properties for the specification and design of complex systems. Abstraction allows highlighting those aspects of the system relevant from a given perspective, while hiding those of no relevance. Encapsulation is the property by which the information contained in an object is accessible only through interactions at the interfaces supported by the object. Because objects are encapsulated, there are no hidden side effects of interactions. It also implies that the internal details of an object are hidden from other objects, which is crucial for dealing with heterogeneity, multiple implementations, interoperability and portability.

2.3 Distribution Transparencies

Transparencies arise from the fact that, when contemplating a distributed system, a number of

concerns become apparent which are a direct result of the distribution: the system components are heterogeneous, they can fail independently, they are at different and, possibly, varying locations, and so on. These concerns can either be solved directly as part of the application design, or standard solutions can be selected, based on best practice. If standard mechanisms are chosen, the application designer works in a world which is transparent to that particular concern; the standard mechanism is said to provide a *distribution transparency*. RM-ODP defines several distribution transparencies: access, failure, location, migration, relocation, replication, and transaction.

RM-ODP defines a set of functions and structures to achieve those transparencies. The system designer can choose which ones to use, since each one may or may not be relevant to a particular application, and each one conveys a cost (in time and resources). RM-ODP does not force the designer to select them all, but in case one is incorporated, it should conform to the model.

3. Adapting RM-ODP and RASDS to Describe Space Systems

While terrestrial distributed systems may be very large and complex, space systems are even more so. Flight elements in space systems are always in motion, whether they are in transit to their destination, in orbit around it, landed upon some remote part of the Solar System, or on their way out of it to even more distant domains. Physics acts upon these systems in a way that must be modeled in our design processes and during control system design, planning, commanding and operations. Thermal and structural factors in the vacuum of space must be modeled and analyzed and even such subtle energies as solar pressure, outgassing, and gravity must be analyzed during design and countered during operations.

Because of the long round trip light times (RTLTL) command, control and monitoring paradigms must be reconsidered, autonomous systems rise in importance, and any notions of distribution transparency must be completely rethought. Assumptions about immediate, continuous, and interactive communications break down when ground and space communications assets must be scheduled months in advance and command and response round trip times rise from tens of milliseconds to tens of minutes or even hours. Communication protocols designed to work with normal terrestrial communication delays break down as interaction times exceed a few seconds.

While RM-ODP provides an excellent framework with which to tackle terrestrial systems, in space additional views and viewpoints are essential, largely because there is an entirely new set of concerns. In our effort to extend the RASDS space data system model to encompass other attributes of space systems the one new viewpoint which we introduced is the Physical one. Other viewpoints in RASDS only require minor changes in order to be used for our purposes. In the rest of this paper we will refer to this extended model as the Reference Architecture for Modeling Space Systems (RAMSS).

RAMSS is intended to produce a model driven design and engineering process for space systems. Central to this concept is the development of an information model which is rich enough to capture all of the critical elements of space mission design, including requirements, mission goals, observational objectives, spacecraft design, development, and operations, space to space and space to ground interactions and communications, and science planning, operations, and processing. RAMSS is initially focused on the early design phases, but is intended to be of use throughout the mission lifecycle.

3.1 Fundamental Concepts - RAMSS

As with RM-ODP, a good framework for space system design should allow different parts of the design to be worked on separately if they are independent, but should clearly identify those places where different aspects of the design constrain one another. In order to achieve this, RAMSS uses

several structuring approaches:

- The specification of a complete system in terms of *viewpoints*.
- The use of a *common object model* for the specification of the system from every viewpoint.
- The use of *views* to tailor user or domain specific analyses of the system.
- The definition of a *modeling infrastructure* that provides support services for system applications, hiding the complexity and problems of defining mission specific models.
- The definition of a set of *common transformation functions* that provide general services needed during the design and development of space systems.
- A *framework* for the evaluation of conformance of models and designs based on conformance points.

3.1.1 Viewpoints - RAMSS

Most space system specifications are so complex and extensive that no single individual can fully comprehend all aspects of the specifications. Furthermore, different stakeholders in the system design have different needs for a given system and different reasons for examining the system's specifications. A mission planner will ask different questions of a system make-up than would a system implementor. The concept of the RAMSS viewpoints framework is to provide separate viewpoints into the specification of a given space system. These viewpoints each satisfy an audience with interest in a particular set of aspects of the system. Associated with each viewpoint is a viewpoint language that optimizes the vocabulary and presentation for the audience of that viewpoint.

“A viewpoint establishes the conventions by which a view is created, depicted and analyzed. In this way, a view conforms to a viewpoint. The viewpoint determines the languages (including notations, model, or product types) to be used to describe the view, and any associated modeling methods or analysis techniques to be applied to these representations of the view. These languages and techniques are used to yield results relevant to the concerns addressed by the viewpoint. An architectural description (AD) selects one or more viewpoints for use. The selection of viewpoints typically will be based on consideration of the stakeholders to whom the AD is addressed and their concerns.” – IEEE 1471-2000

A viewpoint defines 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.

The RAMSS framework extends the RM-ODP framework and provides six generic and complementary viewpoints on the system and its environment:

- The **enterprise** viewpoint, which focuses on the purpose, scope and policies for the system. It describes the organizational entities, requirements, goals, objectives, scenarios, constraints, and how to meet them.
- The **information** viewpoint, which focuses on the semantics of the information and the information processing performed. It describes the information managed by the space system and the structure, content, semantics, type, and relationships among the data used within the system.
- The **functional (RM-ODP computational)** viewpoint, which defines the abstract functional decomposition of the space system into objects which interact at interfaces. It describes the functionality provided by the space system, the behavior of the functional elements and their functional decomposition.

- The **physical** viewpoint, which defines the physical decomposition of the space system into components which interact across connectors. It describes the physical aspects of the space system and the external environment within which it operates, the physical behavior (and motion) of the components and their physical decomposition. The connectors may be manifestly physical (nuts and bolts, struts, network or power cables), or they may be more ethereal (RF & optical signals, thermal radiation, gravitational).
- The **engineering** viewpoint, which focuses on the allocation of implemented functionality to engineered components of the system and on the mechanisms and functions required to engineer and implement the space system, including implementation choices. It describes the distribution of processing performed by the space system to manage the information and provide the functionality.
- The **technology** viewpoint, which focuses on the choice of technology and standards to develop the space system. It describes the standards and technologies chosen to provide the communications, processing, functionality and presentation of information in the space system. It also describes any technology risks that must be assessed during design and development.

A viewpoint is a subdivision of the specification of a complete system, established to bring together those particular pieces of information relevant to some particular area of concern during the design of the system. Although separately specified, the viewpoints are not completely independent; key items in each are identified as related to items in the other viewpoints. However, the viewpoints are sufficiently independent to simplify reasoning about the complete specification. The mutual consistency among the viewpoints is ensured by the architecture defined by RAMSS, and the use of a common object model provides the glue that binds them all together. Figure 1 shows the relationships among the top-level objects in the RAMSS model.

3.1.2 Objects - RAMSS

The RAMSS viewpoint specifications are expressed in terms of objects. An object is an abstract representation of an entity in the real world. It contains information and offers services. A system is composed of interacting objects. Each viewpoint defines its own objects and their relationships and interactions. In the enterprise viewpoint the objects are organizations and the interactions involve requirements, contracts, and policies. In the functional viewpoint the objects are abstract functions and the interfaces are via abstract interfaces. In the physical viewpoint the objects are components with mass and structural properties that are related to one another by some sort of physical connector.

The use of the object paradigm provides abstraction and encapsulation, two important properties for the specification and design of complex systems. Abstraction allows highlighting those aspects of the system relevant from a given perspective, while hiding those of no relevance at that moment. Encapsulation is the property by which the internal implementation details or information contained in an object is accessible only through interactions at the interfaces supported by the object. Because objects are encapsulated, there are no hidden side effects of interactions. It also implies that the internal details of an object are hidden from other objects, which is crucial for dealing with heterogeneity, multiple implementations, interoperability and portability.

3.1.3 Views - RAMSS

Viewpoints provide the conventions, rules, and languages for constructing views. A view is a representation of a whole system from the perspective of a related set of concerns. Views are themselves modular and well formed, and each view is usually intended to correspond to exactly one viewpoint. In RAMSS some new combined views have been defined that correspond to more than one viewpoint. The user may also define a new view based on the basic concepts defined by RAMSS if it is impossible to capture all the important aspects of the system with the six viewpoints defined here. Some aspects of a system design may benefit from being examined from two or more views simultaneously.

MBED Top Level Ontology

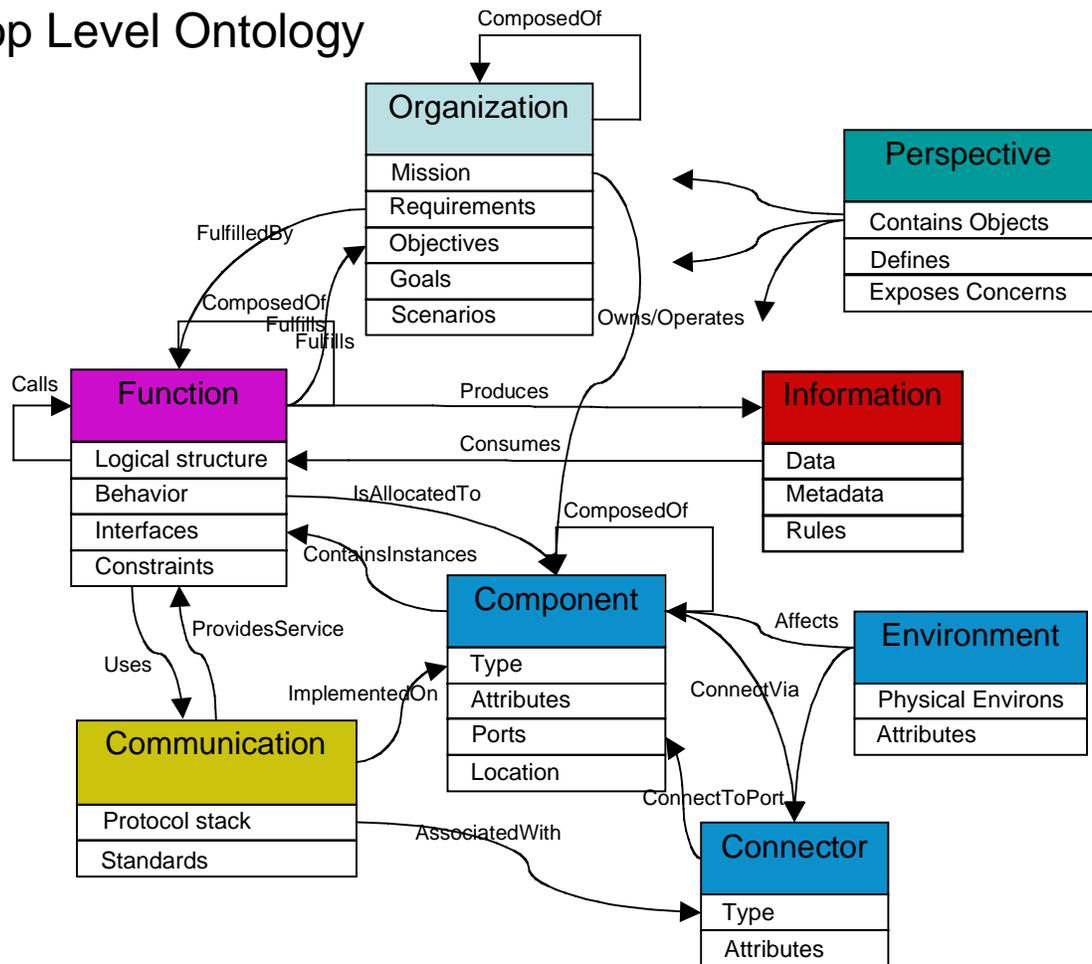


Figure 1 – MBED Top level Objects

“A view may consist of one or more architectural models. Each such architectural model is developed using the methods established by its associated architectural viewpoint. An architectural model may participate in more than one view. NOTE—In a complex system, ADs may be developed for components of the system, as well as for the system as a whole. In this case, it may be that one AD will have a view corresponding to a particular viewpoint and another AD will have a view corresponding to the same viewpoint. Although the system being described by these two

views has the whole-part relationship, this is not an instance of multiple views corresponding to one viewpoint. The ADs are considered separate even though they are related by the systems they describe.” IEEE-1471-2000.

Here is a nominal, and incomplete, set of views that may be defined, associated with one or more viewpoints. Not all of these views may be used for any one project and other views may be defined as necessary. Note that for some analyses elements from multiple viewpoints may be combined into a new view, possibly using a layered representation.

Enterprise Viewpoint

Organization view – Includes organizational elements and their structures and relationships. May include agreements, contracts, policies and organizational interactions.

Requirements view – Describes the requirements, goals, and objectives that drive the system. Says what the system must be able to do.

Scenario view – Describes the way that the system is intended to be used. Includes user views and descriptions of how the system is expected to behave.

Information viewpoint

Metamodel view – An abstract view that defines information elements and their structures and relationships. Defines the classes of data that are created and managed by the system and the data architecture.

Information view – Describes the actual data and information as it is realized and manipulated within the system. Data elements are defined by the metamodel view and they are referred to by functional objects in other views.

Functional viewpoint

Functional Dataflow view – An abstract view that describes the functional elements in the system, their interactions, behavior, provided services, constraints and data flows among them. Defines which functions the system is capable of performing, regardless of how these functions are actually implemented.

Functional Control view – Describes the control flows and interactions among functional elements within the system. Includes overall system control interactions, interactions between control elements and sensor / effector elements and management interactions.

Physical viewpoint

Data System view – Describes instruments, computers, and data storage components, their data system attributes and the communications connectors (busses, networks, point to point links) that are used in the system.

Telecomm view – Describes the telecomm components (antenna, transceiver), their attributes and their connectors (RF or optical links).

Navigation view – Describes the motion of the major elements within the system (trajectory, path, orbit), including their interaction with external elements and forces that are outside of the control of the system, but that must be modeled with it to understand system behavior (planets, asteroids, solar pressure, gravity)

Structural view – Describes the structural components in the system (s/c bus, struts, panels, articulation), their physical attributes and connectors, along with the relevant structural aspects of other components (mass, stiffness, attachment)

Thermal view – Describes the active and passive thermal components in the system (radiators, coolers, vents) and their connectors (physical and free space radiation) and attributes, along with the thermal properties of other components (i.e. antenna as sun shade)

Power view – Describes the active and passive power components in the system (solar panels, batteries, RTGs) within the system and their connectors, along with the power properties of other components (data system and propulsion elements as power sinks and structural panels as grounding

plane)

Propulsion view – Describes the active and passive propulsion components in the system (thrusters, gyros, motors, wheels) within the system and their connectors, along with the propulsive properties of other components

Engineering viewpoint

Allocation view – Describes the allocation of functional objects to engineered physical and computational components within the system, permits analysis of performance and used to verify satisfaction of requirements

Software view - Describes the software engineering aspects of the system, software design and implementation of functionality within software components, select languages and libraries to be used, define APIs, do the engineering of abstract functional objects into tangible software elements. Some functional elements, described using a software language, may actually be implemented as hardware (FPGA, ASIC)

Hardware views – Describes the hardware engineering aspects of the system, hardware design, selection and implementation of all of the physical components to be assembled into the system. There may be many of these views, each specific to a different engineering discipline.

Communications Protocol view – Describes the end to end design of the communications protocols and related data transport and data management services, shows the protocol stacks as they are implemented on each of the physical components of the system.

Risk view – Describes the risks associated with the system design, processes, and technologies, assigns additional risk assessment attributes to other elements described in the architecture

Control Engineering view - Analyzes system from the perspective of its controllability, allocation of elements into system under control and control system

Integration and Test view – Looks at the system from the perspective of what must be done to assemble, integrate and test system and sub-systems, and assemblies. Includes verification of proper functionality, driven by scenarios, in satisfaction of requirements.

IV&V view – independent validation and verification of functionality and proper operation of the system in satisfaction of requirements. Does system as designed and developed meet goals and objectives.

Technology viewpoint

Standards view – Defines the standards to be adopted during design of the system (e.g. communication protocols, radiation tolerance, soldering). These are essentially constraints on the design and implementation processes.

Infrastructure view – Defines the infrastructure elements that are to support the engineering, design, and fabrication process. May include data system elements (design repositories, frameworks, tools, networks) and hardware elements (chip fabrication, thermal vacuum facility, machine shop, RF testing lab)

Technology Development & Assessment view – Includes description of technology development programs designed to produce algorithms or components that may be included in a system development project. Includes evaluation of properties of selected hardware and software components to determine if they are at a sufficient state of maturity to be adopted for the mission being designed.

It is clear that for some of these viewpoints, the physical ones in particular, it can easily be argued that there need to be separate viewpoints for each of structural, thermal, propulsion, power, telecomm because each of these items belong to different subsystems, have different properties, and are analyzed by different means. All of this is true, but it is also true that the underlying physical reality is that there is a set of physical components that are assembled using various kinds of connectors, into something we call a spacecraft. But the critical point is that many of these components have properties that appear in more than one view.

For example, a panel that is part of the spacecraft bus and appears in a structural view may also be a part of the ground plane from the power view and will also appear in the thermal view

because it connects heat producing components that are inside the spacecraft and shields them from solar radiation. The approach that we have taken in modeling is to

4. Summary

The adaptation of RM-ODP viewpoints and concepts to describe space systems appears to offer us some significant advantages:

-First, it may help us thinking from different perspectives (or viewpoints), greatly improving the requirement collection and analysis phases of the development of applications, and providing a set of well proven concepts for analyzing space systems design and behavior.

-Second, RM-ODP offers a conceptual infrastructure and a common reference model within which different views, expressed in separate languages (those from the viewpoints), can be consistently integrated.

-Third, RM-ODP provides a set of already established reasoning patterns to help us specify and design our system. Those patterns assist us to identify the fundamental entities of the system and the relationships among them. In this sense, RM-ODP encourages us to ask the right questions of the right people, and with the appropriate degrees of abstraction and precision for building useful system specifications.

-Finally, RM-ODP provides system designers and developers with a set of mechanisms and common services to facilitate their jobs, and permits models of these complex systems to be developed.

We are working to develop a technological infrastructure that can support these models. Once this infrastructure has been matured it will be able to be used for building robust, efficient and competitive applications, interoperable with other systems that also conform to the same standards, and backed by industrial products with proven capability.

5. References

1. K. Baclawski, H. Kilov, A.E. Thalassinidis, K. Tyson (Eds). *Proceedings of the 8th OOPSLA Workshop on Behavioral Semantics*. Denver (CO). November 1999.
 2. D. D'Souza. Enterprise integration. Enterprise components with Catalysis/UML. In [1] pp. 49-63.
 3. H. Kilov, W. Harvey (Eds.). *Object Oriented Behavioral Specifications*. Kluwer Academic Publishers, 1996.
 4. H. Kilov, B. Rumpe, I. Simmonds (Eds.). *Behavioral Specifications of Business and Systems*. Kluwer Academic Publishers, 1999.
 5. C. Szyperski. *Component Software. Beyond Object-Oriented Programming*. Addison-Wesley Longman, 1998.
- IEEE 1471
ISO 10746
RASDS
RUP-SE
MBSE

6. Acknowledgements

The RM-ODP introductory section of this paper is an adaptation (with permission) of an introductory paper from the ISO web site, written by Antonio Vallecillo of ETSI Informática. Universidad de Málaga av@lcc.uma.es. The terminology used in this paper is drawn from the ISO/IEC 10746 documents defining RM-ODP and also from IEEE-1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems. Most of the concepts for extending RM-ODP to describe space data systems were developed within a working group in CCSDS, the Consultative Committee on Space Data Systems, in the creation of the Reference Architecture for Space Data Systems (RASDS), CCSDS 800x0-R-1. The work to extend these concepts in order to describe the full architectural framework for space systems was developed in a JPL funded Research and Technology project called Model Based Engineering and Design, whose PI was Steven Wall.

Biography

Peter Shames has been engaged in the process of turning computers into useful tools for scientists for the bulk of his professional career. He manages JPL's Information Systems Standards Program in the Interplanetary Network Directorate (IND). He is working within the Consultative Committee for Space Data System (CCSDS) to define an end to end reference architecture and formal methodology for describing space data systems. He has developed architectures for a variety of NASA programs, including JPL's mission operations system, the Hubble Space Telescope science processing and archiving systems, and real time data acquisition. He has served on working groups in the National Academy of Sciences and the Internet Activities Board. He is the Director of the newly formed Consultative Committee for Space Data System's System Engineering Area. Once upon a time he used to know how to program.

Joseph Skipper has been engaged ...