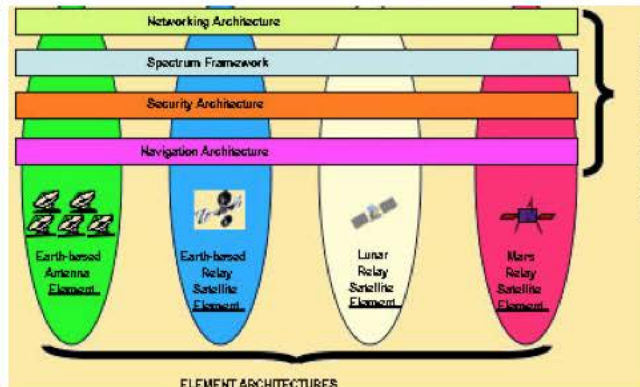
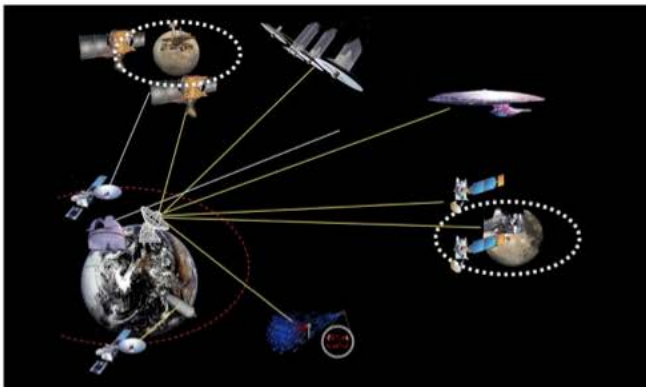
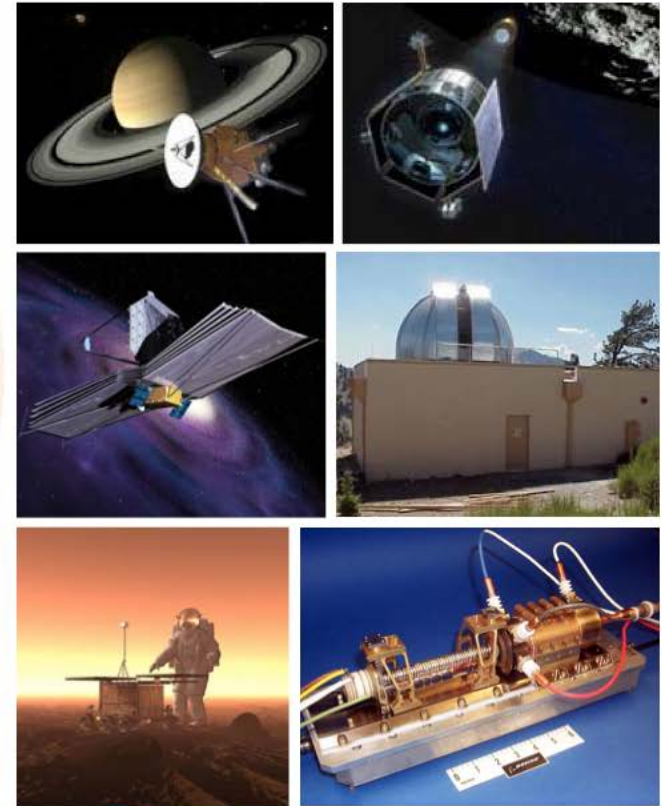


Modeling Systems-of-Systems Interfaces with SysML

Peter Shames, Marc Sarrel,
Jet Propulsion Laboratory, California
Institute of Technology
Sandy Friedenthal, SAF Consulting
SpaceOps 2016, 16-20 May 2016
Daejeon, Korea





Introduction – End-To-End Space Data Systems

- **End-to-End space data systems are inherently systems of systems**
- **Typically composed of spacecraft and mission operations systems (MOS) that belong to one (or more) organizations, and multi-mission communication assets that belong to another organization.**

**“The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces.”
Maier, 1998**

- **So ... how to accurately describe, model, and characterize these systems and their interfaces?**

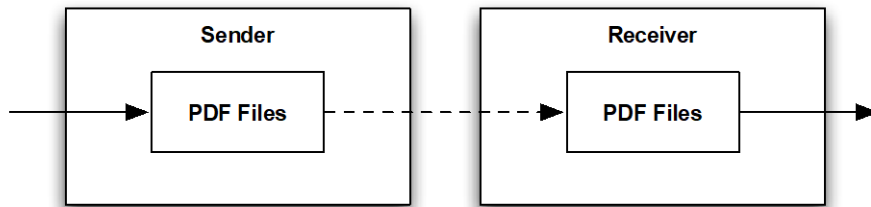
System-of-Systems Interfaces Considerations

- **Each major system has interfaces**
- **The elements within each system have interfaces**
- **Interfaces include the connection points on the interacting elements, the items that are exchanged, the constraints and/or rules that govern the exchange, and the medium for the exchange (i.e. link)**
- **Multi-layered communications interfaces include application layer to physical layer**
 - **For some interfaces the link aspects can drive performance (e.g. atmospheric affects of various signals)**
 - **For other interfaces protocol behavior and data transformations can drive performance (e.g. end-to-end throughput)**

System-of-Systems Interfaces Considerations, contd

- **Different views may be used to address different concerns**
 - Different interface abstractions from logical flows to complete protocol stacks
 - End-to-end data flows, connectivity, and data transformations
 - Physical connections
 - Protocol specifications
 - Message definitions
 - Complete interface specifications that span discipline concerns
- **The type and detail of interface information for each view must be consistent**

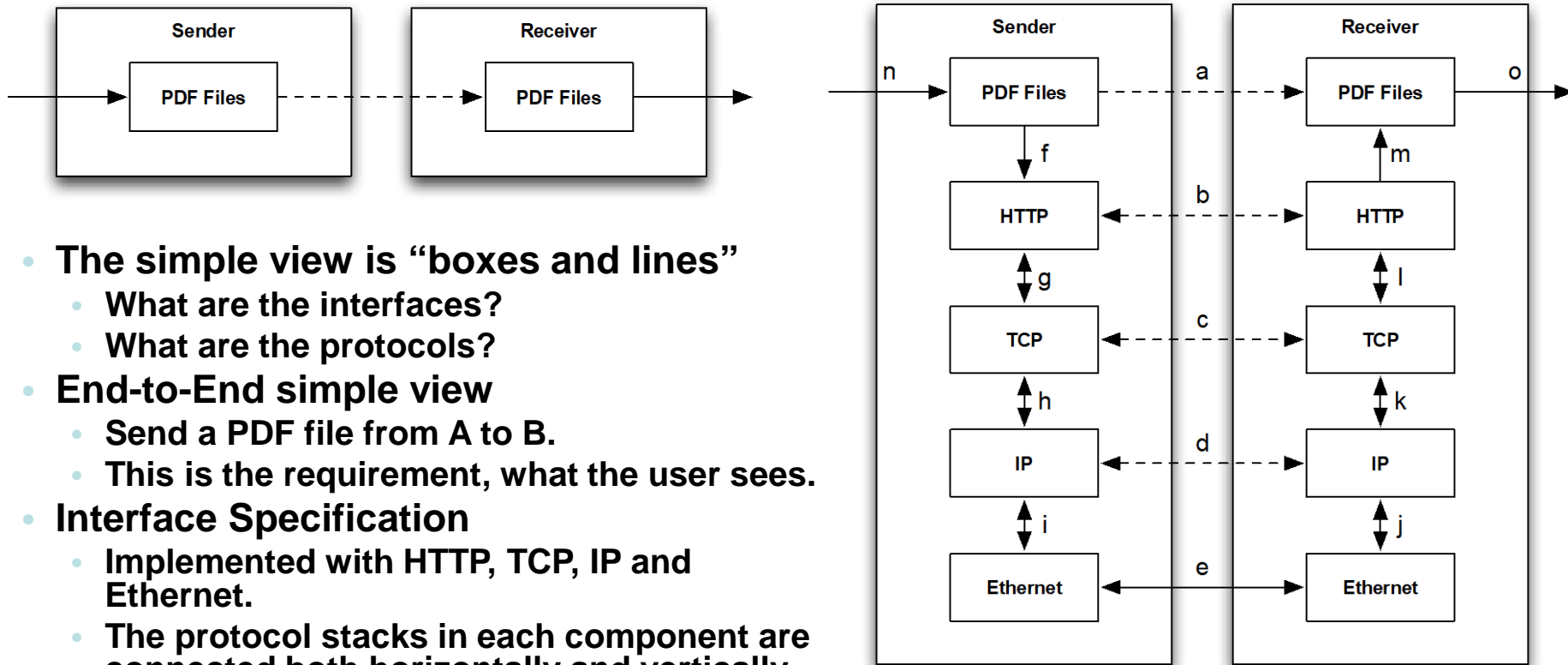
A Simple Example of Interface Stack Specification



- **The simple view is “boxes and lines”**
 - What are the interfaces?
 - What are the protocols?
- **End-to-End simple view**
 - Send a PDF file from A to B.
 - This is the requirement, what the user sees.
- **Interface Specification**



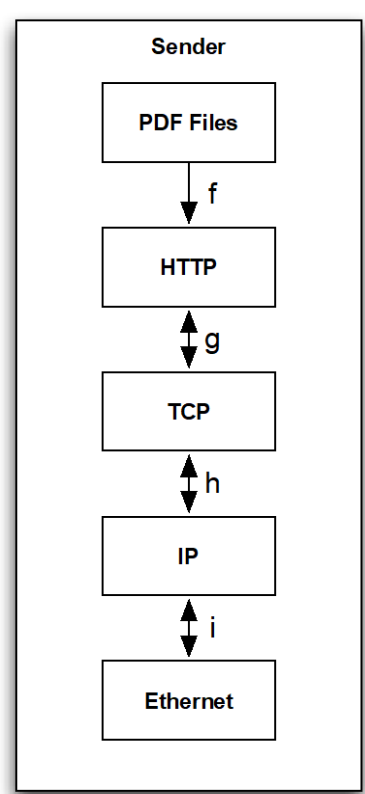
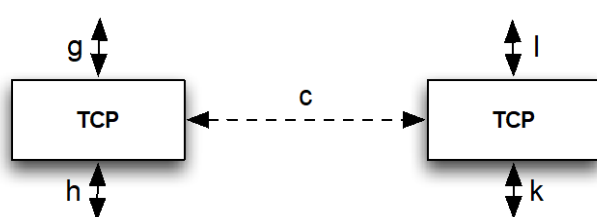
A Simple Example of Interface Stack Specification - Details



- **The simple view is “boxes and lines”**
 - What are the interfaces?
 - What are the protocols?
- **End-to-End simple view**
 - Send a PDF file from A to B.
 - This is the requirement, what the user sees.
- **Interface Specification**
 - Implemented with HTTP, TCP, IP and Ethernet.
 - The protocol stacks in each component are connected both horizontally and vertically..
- **Simple lists of interface protocols are just not sufficient to understand the architecture**



Focus May be on the Protocol at a Layer or on the Interface Stack



- Different views enforce separation of concerns.
- We can focus on just the TCP layer.
 - How it is connected (horizontally).
 - How it behaves (horizontally).
- We can focus on just the stack.
 - How it is connected (vertically).
 - How it behaves (vertically).
- Data logically flows across the horizontal layers, the TCP spec describes the behavior of the peer protocol entities.
- Data actually flows “down the stack” through each successive layer until it gets to the physical layer where the “real” connections occurs.
- “On the wire” the whole stack is visible.

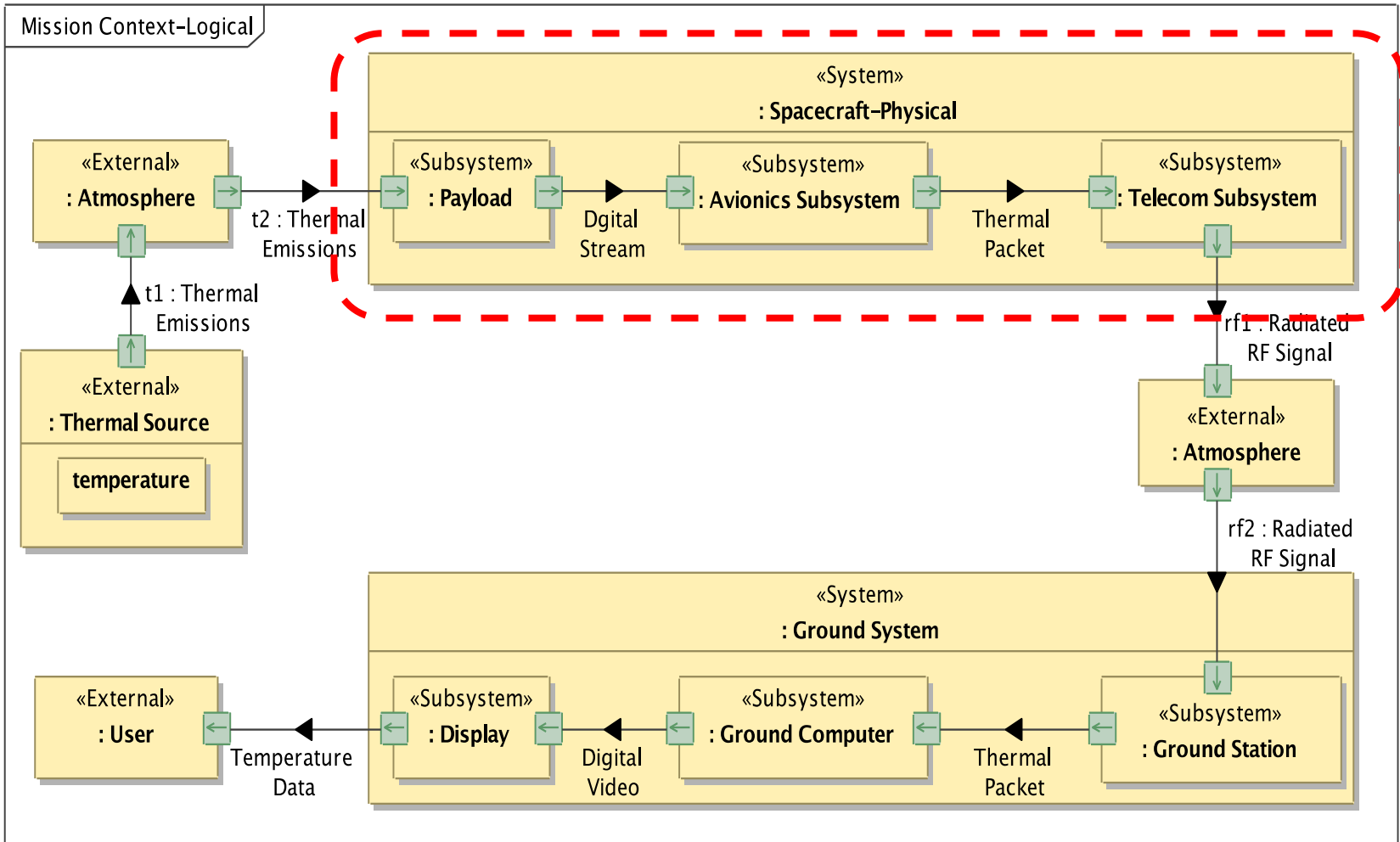


Notes on the System Interface Modeling Method

- **The method is intended to allow complete modeling of systems components, their interfaces, protocols, and protocol behaviors.**
 - **Supports evolution of detail as design progresses**
- **Several different views of these interfaces are presented.**
- ***Only selected views need to be used to address specific concerns (e.g., logical data flow, protocol stacks and behavior, message definition, physical interconnection).***
- **Appropriate use of the method, starting with the basic views, allows other views and details to be added when and where they are required.**



End-to-End Space Data System Example

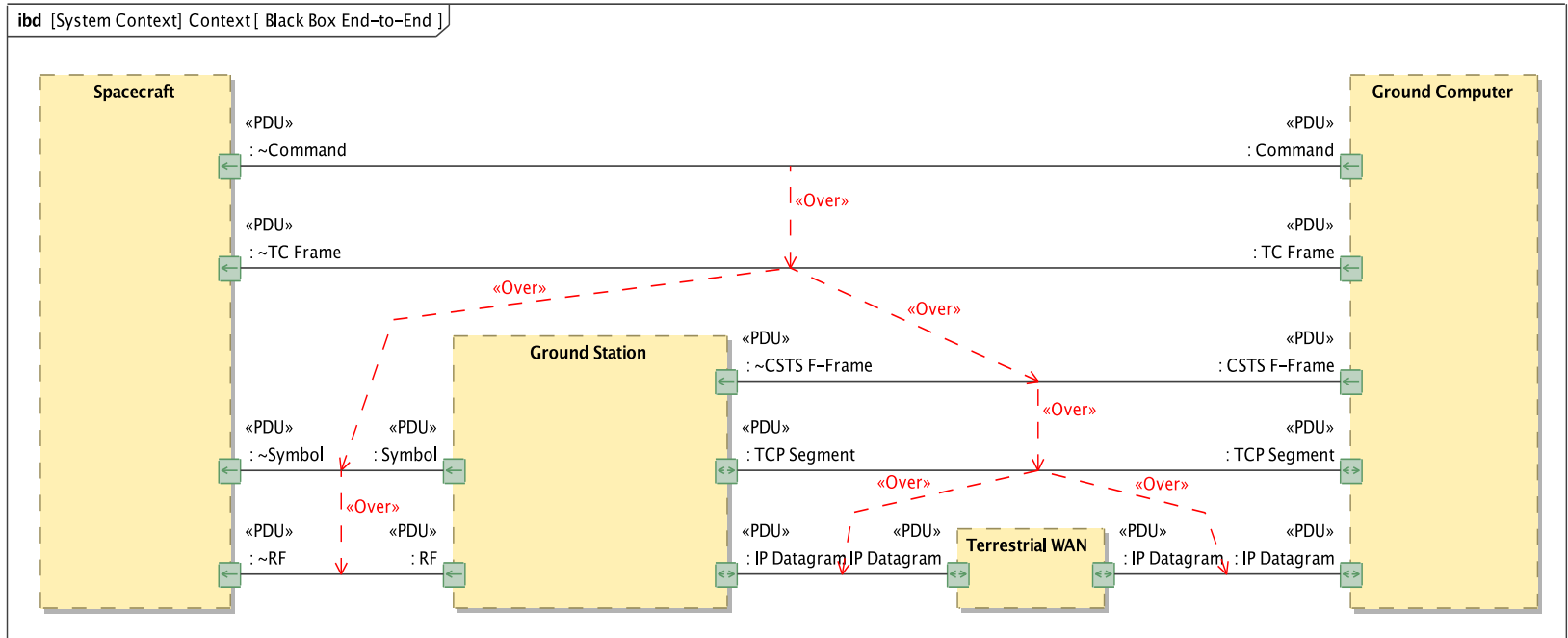




Overall Mission Model Context

- **Provides an End-to-End view of the whole context**
- **Shows the major physical elements that are being modeled**
- **Shows a coarse grained decomposition of the major system elements**
- **Shows the nature of the connections and the high level data flows**
- **Does not show organizations, ownership, or operational details, but views based on this could be developed to address these concerns**
- **Does not show interface or protocol details (yet...)**

End-to-End Protocol View



- **Major communications systems from the End-to-End view**
 - Shows End-to-End high level protocol flows
 - Shows typical CCSDS space communications standards in relationship to major system elements
- **Interface Specification**
 - Shows only the top level ports, protocol types, and data flows
 - Does not show interface nor protocol stack details

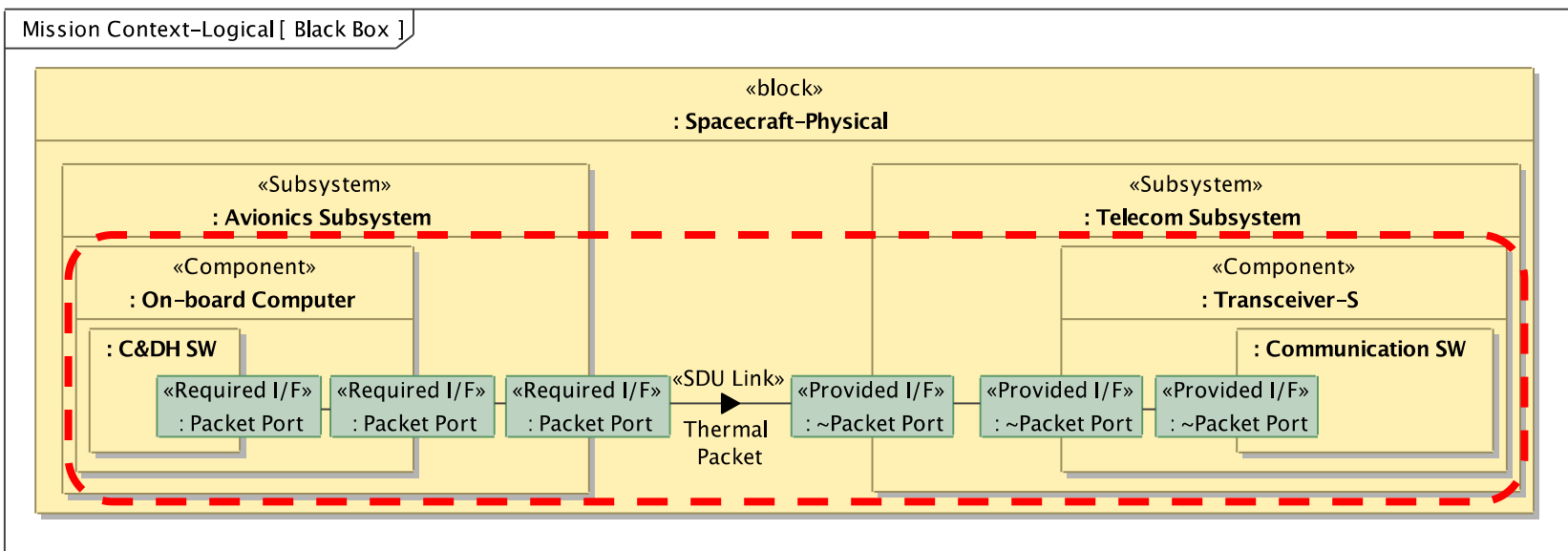


End-to-end Space Data System Flows and Connectivity

- **Unlike terrestrial systems, space data systems use a variety of different protocols in order to communicate end-to-end.**
- **Often these are asymmetric, with different protocols used for forward and return data paths.**
- **These protocols may be locally developed and used within a single organization, or standard, interoperable, ones like CCSDS.**
- **Terrestrial space communication assets are usually multi-mission and typically offer standard cross support interfaces like SLE or CSTS.**



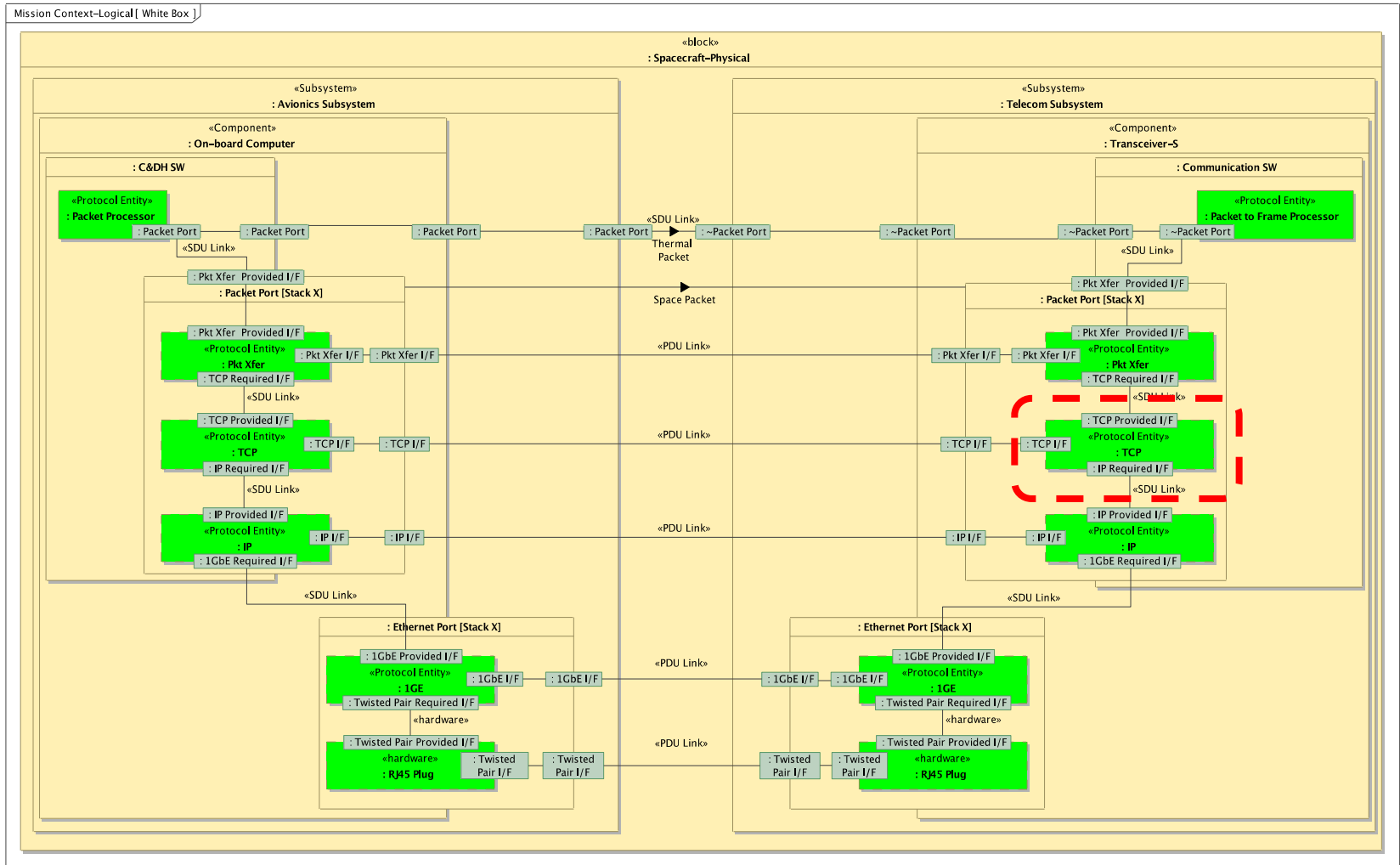
Modeling the Spacecraft System Decomposition – “Black Box” View



- Shows one of the systems from the End-to-End view
 - Shows a partial decomposition and component relationships
 - Shows hardware and software relationships
- Interface Specification
 - Shows only the top level ports and data flows and type of data
 - Does not show interface details nor protocol stacks



Modeling the Interface – Protocol Stack Details "White Box" View



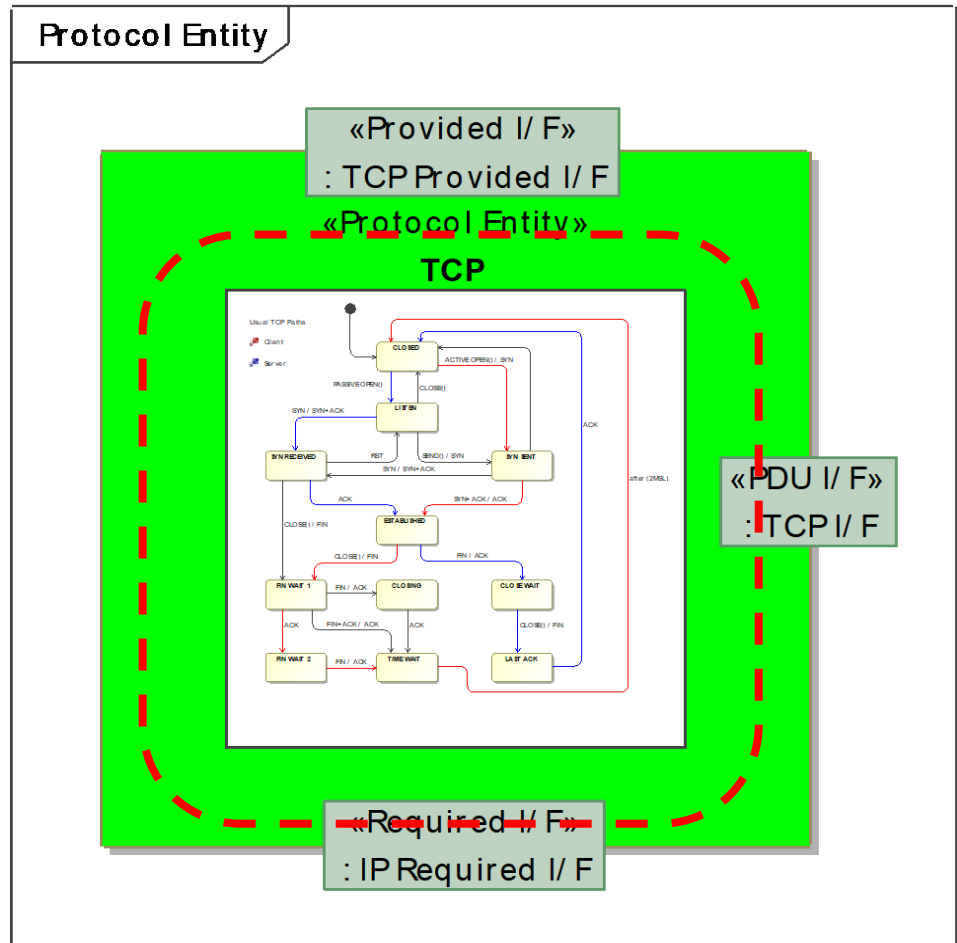


Interface Model – Protocol Stack Details

- Shows the details of the interface protocol stack
- Defines the Interface Binding Signature
 - Application layer: packet transfer protocol, manages exchange of packet data between applications.
 - Transport layer: Transmission Control Protocol (TCP), provides end-to-end, once only, in order, complete delivery of data.
 - Network layer: Internet Protocol (IP), provides network layer routing over any number of intermediate network nodes.
 - Data link layer: 1 Gb Ethernet, provides data link layer services that may involve a fabric of switches and hubs.
 - Physical layer: twisted pair cable (Cat-5) and RJ-45 plug terminations.
- Distinguishes which components own the physical layers and which the data layers

Protocol Entity Interfaces

- Every protocol entity has three ports: the interface that provides services to the upper (N+1) layer; the interface that requires services of the lower (N-1) layer; and the interface with the peer protocol entity at the same layer
- There may also be a management interface, which can be in-line or separate

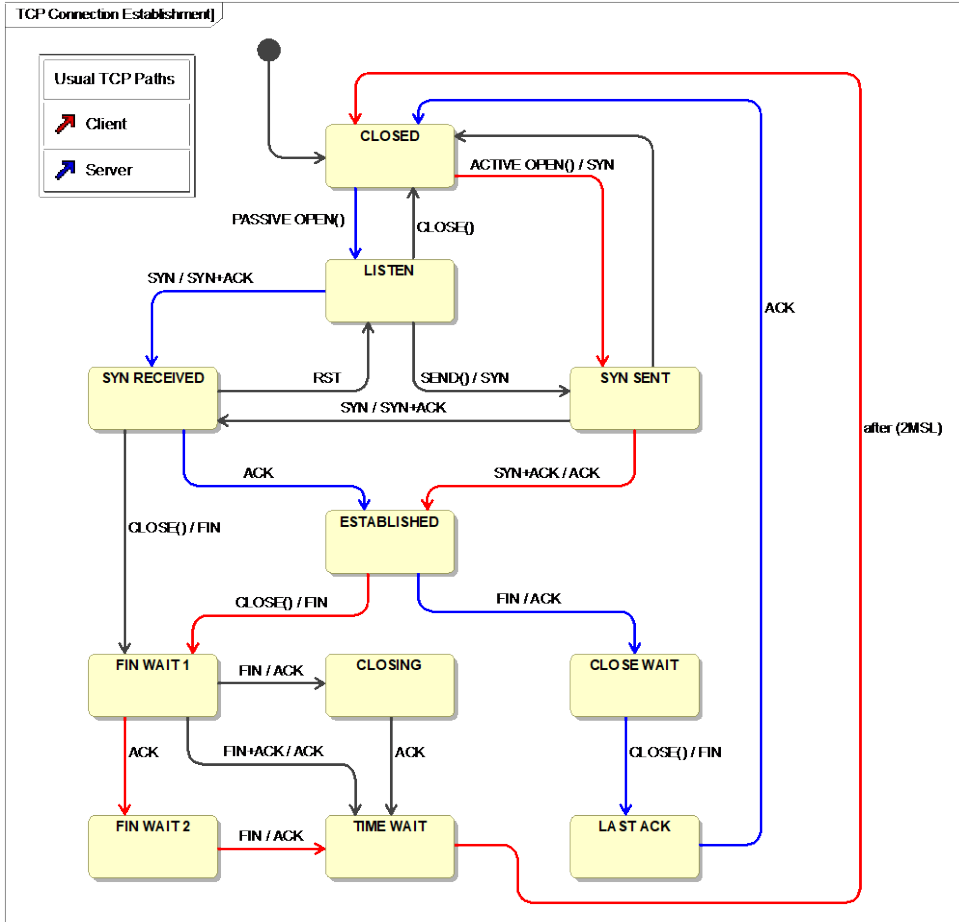


Protocol Entity Port Details

- **Description of protocol entity ports:**
 - **Provided service port:** the services offered to any upper layer (N+1) protocol, defined as an abstract service and using a layer N Service Data Unit (passes SDUs)
 - **Required service port:** the services required from any lower layer (N-1) protocol, defined as an abstract service and using a layer N-1 Service Data Unit (passes SDUs)
 - **Peer protocol port:** the port that enables the protocol entity to interact with its peer entity at the same layer, defined by the protocol specification and using the layer N Protocol Data Units (exchanges PDUs)
 - **Protocol management may be in-line with the protocol, via a MIB, or a separate interface**



Protocol Behavior Specification - TCP Protocol State Machine Example



- Protocol behavior specification
- Might be described as:
 - State machine
 - or ...
 - State tables
 - Sequence diagrams
 - Simple English text
- State machines are clear, formalized, and integrated with the rest of the model

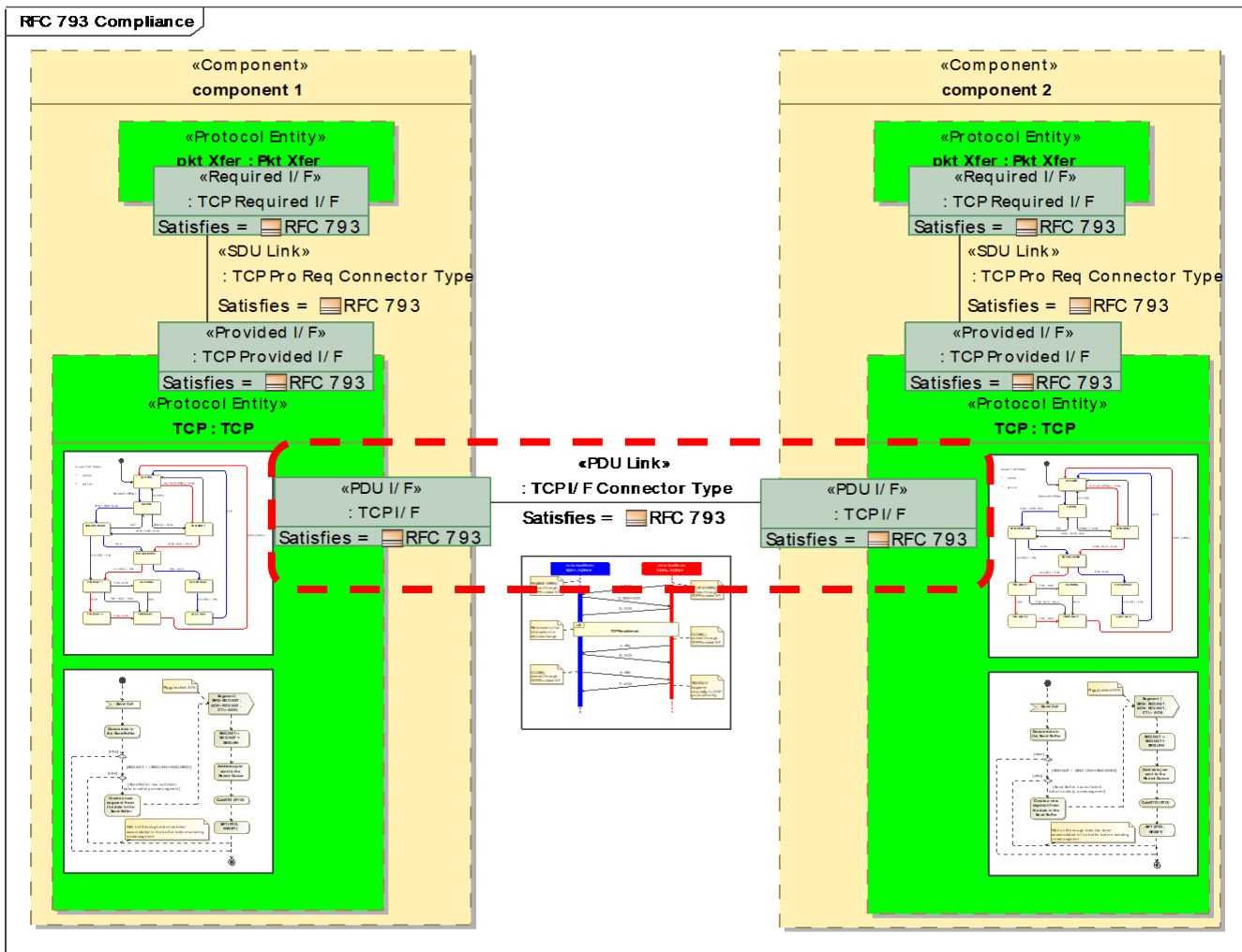
Protocol Behavior Description

- **Describes how a protocol entity behaves when receiving a Protocol Data Unit (PDU) from a peer entity**
- **Describes the exchange(s) of PDUs between peers**
- **May describe the behavior at the required and provided interfaces, such as start-up, connection establishment, and SDU transformation into PDUs**
- **Typically involves describing the dynamics of PDU exchanges, including nominal and error conditions**



Interface Compliance Specification Example

TCP, RFC 793





Interface Compliance Specification Details

- **Where necessary, interfaces may be defined by a full stack specification and by tying the details of each layer not just to a whole standard, but to specific subsections of a standard**
- **Compliance includes interface data, PDU structures, and behavior as illustrated by a state machine or by activity, and sequence diagrams.**
- **Compliance is captured explicitly in the model by the “Satisfies” notation shown on the diagram.**
- **If the protocol at any given layer is fully specified in a published standard, just using “Satisfies” for these layers may be adequate**



Modeling Flexibility – Re-use of Components and Views

- **Models may contain multiple views, from top level context, to composition and components, and down to low level interface and protocol details**
- **Appropriate use of carefully chosen stereotypes and views allows different levels of abstraction to be expressed clearly**
- **Development of libraries of components and of protocol entities allows them to be re-used and combined as needed in different views**
- **Different views allow the systems, and systems-of-systems, to be understood from different perspectives and at different levels of detail, as needed**



Systems-of-Systems Interface Modeling Benefits

- The method permits space data systems, and systems-of-systems, their structure and interfaces, to be modeled with a high degree of fidelity
- *Not all systems require all of these layers of details*
- Adopting a principled approach to component and interface modeling allows successive levels of detail to be added as and when needed
- A well constructed partial model can document systems, interfaces, and behavior at a much deeper level of specificity than is typical in an ICD “protocol list” or “boxes and lines” diagrams
- Inclusion of clear descriptions of connectivity and composition, and mapping to specifications and behaviors, supports analysis of flow and continuity, and design of interfaces
- Further parametrization of the interfaces can be used to support analysis of end-to-end behavior and performance



References

- [1] Object Management Group. **OMG Systems Modeling Language (OMG SysML™). V1.4.** Available at: <http://www.omg.org/spec/SysML/>.
- [2] **A Practical Guide to SysML, The Systems Modeling Language, Third Edition** by Sanford Friedenthal, Alan Moore, and Rick Steiner, Morgan Kaufmann, 2014
- [3] Shames, Peter M, Sarrel, Marc A, Freidenthal, Sanford A, **A Representative Application of a Layered Interface Modeling Pattern, to be published 26th Annual INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, July 18-21, 2016**
- [4] Shames, Peter M, Sarrel, Marc A, **A modeling pattern for layered system interfaces, 25th Annual INCOSE International Symposium (IS2015), Seattle, WA, July 13 – 16, 2015**
- [5] Postel J., **“Transmission Control Protocol,” RFC 793, September 1981.**^[1]_[SEP]
- [6] Zaghal, R, Khan, J, **EFSM/SDL modeling of the original TCP standard (RFC793) and the Congestion Control Mechanism of TCP Reno, Kent State University report, TR2005-07-22-tcp-EFSM.pdf, 2005**
- [7] **Information technology - Open Systems, Basic Reference Model, ISO/IEC 7498-1, revised June, 1996**
- [8] **Systems and software engineering — Recommended practice for architectural description of software-intensive systems, ISO/IEC 42010, July 2007, revised 2011**
- [9] **Reference Architecture for Space Data Systems (RASDS), CCSDS 311.0-M-1, Sept 2008**
- [10] **Consultative Committee for Space Data Systems, CCSDS Space Packet Protocol, CCSDS 133.0-B-1c2, Sept 2010**
- [11] Jackson, et al., **“Architecting the Human Space Flight Program with Systems Modeling Language (SysML)”**, Infotech 2012, AIAA 2012-2556.
- [12] Karban, et al., **“MBSE Initiative – SE2 Challenge Team, Cookbook for MBSE with SysML, Issue 1, INCOSE, 2011.**