A DEEPER DIVE INTO THE MEANING AND IMPLICATIONS OF INTEROPERABILITY FOR LUNANET COMMUNICATIONS AND NAVIGATION SERVICES

James Schier
National Aeronautics and Space Administration, 300 # St., SW, Washington, D.C. 20546 USA, +1 202-538-4514, james.schier-1@nasa.gov
Coralí Roura
National Aeronautics and Space Administration, 300 # St., SW, Washington, D.C. 20546 USA, +1 202 222 5889, corali.roura-1@nasa.gov
Phillip Paulsen
National Aeronautics and Space Administration, 21000 Brookpark Road, Cleveland, OH 44135, +1 216 433 6507, phillip.e.paulsen@nasa.gov
Karl Vaden
National Aeronautics and Space Administration, 21000 Brookpark Road, Cleveland, OH 44135, +1 216 433 8131, karl.r.vaden@nasa.gov
Jennifer Rock
National Aeronautics and Space Administration, 21000 Brookpark Road, Cleveland, OH 44135, +1 216 433 3354, jennifer.l.rock@nasa.gov
Angela Peura
Agile Decision Sciences, LLC, 350 Voyager Way, Suite 100B, Huntsville, AL 35806, +1 571 643 5037, APeura@asrcfederal.com
Marc Seibert
Agile Decision Sciences, LLC, 350 Voyager Way, Suite 100B, Huntsville, AL 35806, +1 240 405 6844, MSeibert@asrcfederal.com
Erica Weir
Teltrium Solutions, LLC, 600 Jefferson Drive Rockville, MD 20852, +1 720 318 7296, eweir@teltrium.com

Abstract

As part of planning efforts for cislunar exploration and science missions, space agencies have been collaborating with each other to enable communications, networking, Position, Navigation, and Timing (PNT) systems to exchange information and provide services to spacecraft and space systems in transit, in orbit, and on the surface, thus helping each other to achieve their common goals. To achieve commonality and lower cost for mutual benefit, the strategy of interoperability is being adopted to help all the pieces fit together and function smoothly. Interoperability gives cislunar users the ability to operate in a collaborative environment similar to the terrestrial Internet, allowing them to share information, navigate safely despite increasing radio frequency congestion, and follow common processes and procedures for effective joint operations. Unlike prior government-dominated efforts, this ecosystem is expected to include commercial for-profit businesses, non-profit organizations, and academic institutions. Ultimately, the goal is to enable a cislunar ecosystem of service providers and users to contribute and/or utilize infrastructure and capabilities to accomplish mission objectives spanning the full range of human endeavours while supporting a variety of business models. This paper reports on the results of an effort to assist in efforts to frame the development of the international LunaNet architecture by providing a canonical definition of interoperability broad enough to meet these needs, examine architectural and operational implications of the definition, and explore interoperability strategies and tactics for deploying and evolving these services. It describes key systems-of-systems (SoS) (Network-of-Networks) interoperability concepts in the context of sustainment of the ecosystem over time as systems evolve in technologies, standards and Standards Development Organizations, component and subsystem upgrades, and user applications.

1. Interoperability Definition

Interoperability is defined here as the ability to exchange and understand information between entities that agree on: a) the syntax used to format (encode) the information; b) the semantic meaning (content) that represents the information or references the contextual meaning; c) the means of exchanging the information; d) the system context in which the information exchange occurs; and e) the means of initiating and sustaining interoperability across asynchronous changes in entities and their operation.

Entities can include humans, human organizations, communication systems, networks, and other systems that exchange information. Due to communications latency, information exchange also includes information storage and retrieval which imposes its own solutions to information representation required to preserve semantic content. The context in which the information exchange occurs represents broad social, societal and organizational information such as reference systems, regulations, policies, and legislation. Interoperability in this definition represents ideal or perfect exchange and understanding. Imperfections in data transmission, information representation, and conversion at interfaces introduce errors and noise that contribute to reduction or loss of intended interoperation. Thus, interoperability is not a simple binary yes/no state; it allows a range of degrees of interoperability that can be designed, assessed, and evolved

1.1 Architectural Implications of the Definition

This expanded definition enables use of formal systems engineering methods. We offer one example of Model-Based Systems Engineering as evidence that interoperability can be modeled using tools like Object Management Group’s Systems Modeling Language™ (OMG SysML®). [1]

To apply structure and rigor to the methodology described, two defining elements of interoperability, language and architecture, are modeled. To develop the language of interoperability, lexicon and taxonomy are modeled, in pursuit of a full ontology. The architecture of interoperability is modeled as a Capability, which evolves through the addition of sub-capabilities. These models can be developed using multiple commercial tools:

Figure 1 shows a SysML logical architecture (yellow) for Interoperability and an example showing how it could be instantiated (orange). This logical architecture can be instantiated to build current and proposed models of interoperability. The abstract model extends the *syntax* to define terms for hierarchical Capabilities and Assessments. The instantiated model defines specific Capabilities decomposed into multiple sub-capabilities can trace to requirements. Capabilities may need to be refined into both functions and time-phased increments. The blue boxes denote how the five elements of the interoperability definition can be modeled. Subsequent sections show how this enables measurement of interoperable properties, verification of requirements that specify the extent of interoperability at distinct evolutionary points during system operation, and the ability to sustain a specified degree of interoperability over the lifetime of the systems(s) involved.

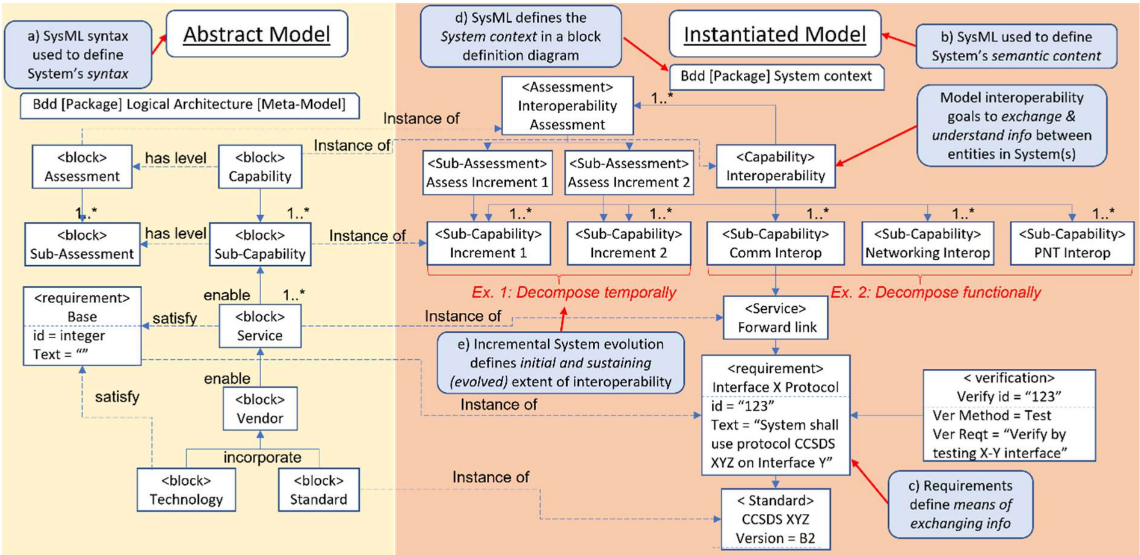

a) SysML syntax used to define System’s *syntax*
Abstract Model
d) SysML defines the *System context* in a block definition diagram
Instantiated Model
b) SysML used to define System’s *semantic content*
Bdd [Package] Logical Architecture [Meta-Model]
Bdd [Package] System context
<Assessment> Interoperability Assessment
1..*
Model interoperability goals to *exchange & understand info* between entities in System(s)
Instance of
Instance of
<block> Assessment
has level
<block> Capability
<Sub-Assessment> Assess Increment 1
<Sub-Assessment> Assess Increment 2
<Capability> Interoperability
1..*
1..*
1..*
1..*
1..*
1..*
1..*
<block> Sub-Assessment
has level
<block> Sub-Capability
Instance of
<Sub-Capability> Increment 1
<Sub-Capability> Increment 2
<Sub-Capability> Comm Interop
<Sub-Capability> Networking Interop
<Sub-Capability> PNT Interop
<requirement> Base
id = integer
Text = “”
enable
1..*
Ex. 1: Decompose temporally
Ex. 2: Decompose functionally
satisfy
<block> Service
Instance of
<Service> Forward link
e) Incremental System evolution defines *initial and sustaining (evolved)* extent of interoperability
enable
satisfy
<block> Vendor
Instance of
<requirement> Interface X Protocol
id = “123”
Text = “System shall use protocol CCSDS XYZ on Interface Y”
< verification> Verify id = “123”
Ver Method = Test
Ver Reqt = “Verify by testing X-Y interface”
incorporate
<block> Technology
<block> Standard
Instance of
< Standard> CCSDS XYZ Version = B2
c) Requirements define *means of exchanging info*

Figure 1. Example SysML Implementation of Interoperability

1.2 Operational Implications of the Definition

This section explains in more detail the essential characteristics that must occur for interoperability to become an operational capability.

Exchange and *understand information*: The Institute of Electrical and Electronics Engineers (IEEE) defines interoperability as "the ability of two or more systems or components to *exchange* information and to *use* the information that has been exchanged" [2]. This has two parts. First, information must be exchanged – transmitted by one entity and received by another entity. The exchange may be human to human, human to machine (system), or machine to machine. Second, the information must be understood by the receiver as the same information that was transmitted.

The complexity of the overarching concept can be seen in an Air Traffic Control (ATC) example (Figure 2). Three dissimilar systems constitute part of two interoperating networks. Systems 1 is an air traffic control system and System 2 is an aircraft (but could be a space or ground system). System 3 acts as the interface between networks and could be a ground, maritime, airborne, or space-based platform. System 1 and System 3 contain the communication subsystems that enable them to be members of Network 1 while System 2 and System 3 contain the communication subsystems that enable them to be members of Network 2. System 3 acts as a gateway exchanging data between Network 1 and Network 2. Operator 1 resides in System 1; Operator 2 resides in System 2; and System 3 is automated.

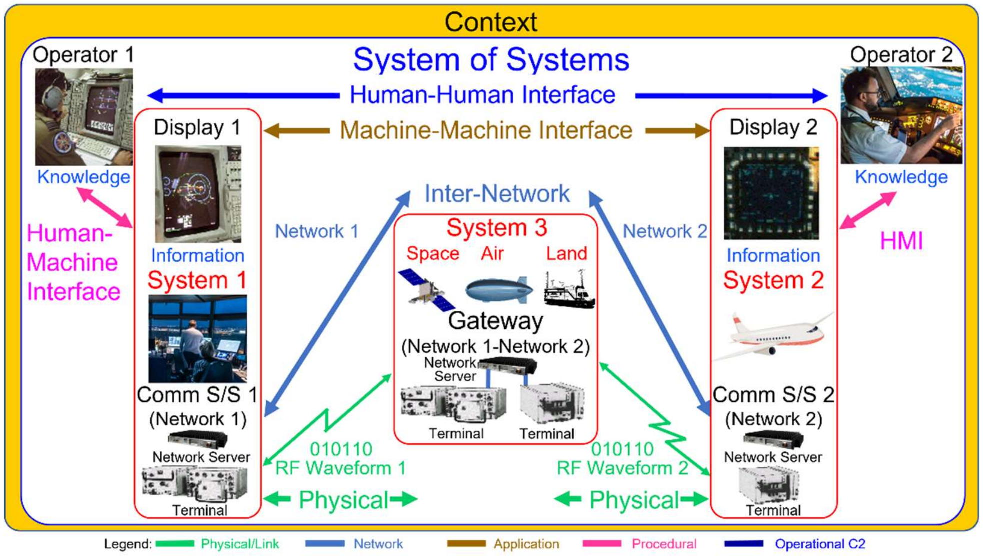


Figure 2. System-of-Systems (SoS) with Interoperability in an ATC Operational Context

- *Human-Human interoperation* occurs (end-to-end) when information sent by one human is received and understood by another human. This occurs if they: a) speak the same language, e.g., English, with the language-specific *syntax* that defines sentence structure; b) employ words to describe commands using defined *semantics*, i.e., the definitions and concepts of the words; and c) use a *method of information exchange*, e.g., transmitted by mouth and received by ear. Understanding is successful if the information received is equal to the information sent and both share the same understanding of that information – the *context*. The air traffic controller, Operator 1, gives the direction "Turn to heading 310°". The pilot, Operator 2, receives the information formatted as a command in English with semantics directing Operator 2 to change the direction of flight of System 2. This presumes that the operators share the *context* of a previously defined coordinate system that defines an absolute reference direction (north) and a relative offset from that measured in units of degrees wherein a circle is divided into 360°.
- *Human-Machine interoperation* occurs when information sent by the human is received and "understood" by the machine, i.e., the information is converted from a human format to a machine format. If Operator 1 speaks a command, the machine must have an audio receiver, e.g., a microphone, that converts the operator's modulated voice into a signal that encodes the same information as the spoken command. System 1 transmits the signal via System 3 to System 2 which translates the command into an audio signal to a speaker that reconstructs Operator 1's words. Understanding at the HMI level occurs if Operator 2 hears and understands

the command to “Turn to heading 310°”. A second HMI interface occurs in the exchange of visual information. Operator 1 uses eyes to observe System 1’s monitor showing tracking information which is transmitted to System 2’s display enabling Operator 2’s eyes to see the same tracked objects and achieve the same situational awareness as Operator 1.

- *System-System interoperation* occurs if the systems: a) employ the same information *syntax*, e.g., a message data format; b) understand the *message contents*, e.g., encode and decode the same information in the message; and c) combine that with *means of transmitting and receiving* the message, e.g., via communication subsystems. System 1 may interface directly with System 2 via a Machine-Machine Interface (MMI) if the systems use a common radio waveform.
- *Network-Network interoperation* occurs when information sent by one network can be received and understood by another network. In Figure 2, two networks are represented that do not share a common set of networking protocols. System 1 belongs to Network 1 while System 2 belongs to Network 2. Between them is System 3, which is a gateway – member of both Networks 1 and 2 – capable of translating from the Network 1 protocol stack to that of Network 2.
- *Physical-Physical interoperation* occurs when information sent by one terminal can be received and understood by another terminal. In the example, terminals 1 and 2 use different waveforms for communicating and are, therefore, incapable of direct interoperation requiring translation by System 3’s gateway. Waveforms are characterized by frequencies, modulation, coding, error correction, and other parameters as well as by the protocol used to control the energy radiated at the physical level to ensure correct information transmission.

Without the means of sustaining interoperability, interoperability may occur at an instant of time, but it decays due to the asynchronous changes made by Standards Development Organizations (SDO), equipment vendors, service providers (e.g., airports), and users (e.g., airlines) over the asynchronous evolution of systems from initial to final configurations. Sustainable interoperability across modifications to hardware and software requires two criteria:

- *Backwards compatibility* exists if information created in an entity with a newer version (improved *syntax*, *semantics* or *exchange* capability) can be sent to and understood by an entity operating on an older version. The older entity can extract the proper subset of information described by its older equipment version and accurately act on the subset of information that it understands.
- *Version transparency* exists if the receiving system, built using an older version, can then retransmit the information it received to other systems without loss of information present only in the newer version. The older entity must be capable of preserving and transmitting the information that it received rather than the information that it is capable of understanding.

Backwards compatibility and version transparency can be achieved if the information being transmitted across a set of interacting entities of heterogeneous versions can be represented as *self-describing*. Prior technologies designed hardware and software that either had specific capabilities built in or allowed reconfiguration via reprogramming. Both approaches have limited ability to support backwards compatibility or version transparency.

Self-describing information can be implemented in three ways: a) By exchange of information that supports a *syntax* capable of describing the capabilities embodied in its version as well as the *semantic content* to define the specific behavior and performance provided by those capabilities; b) By exchanging the *semantic content* using a shared *syntax* and schema prior to its use in operations enabling the entities to rely on stored information, or c) By using a hybrid combination of those two methods. eXtensible Markup Language (XML) uses a combination of referencing a specific version and defining an internal extension that allows users to create customized extensions to base capabilities. Software-defined radios (SDR) now use similar abilities so that a component of any version can interpret the schema for that version *or earlier versions* while passing on the complete information containing definition of capabilities of later versions.

Independent of backward compatibility and version transparency, procedures can be used by cooperating organizations and systems to preserve interoperability while evolving. An example of this is considered in the section on Sustaining Interoperability.

2. Interoperability Strategies and Tactics

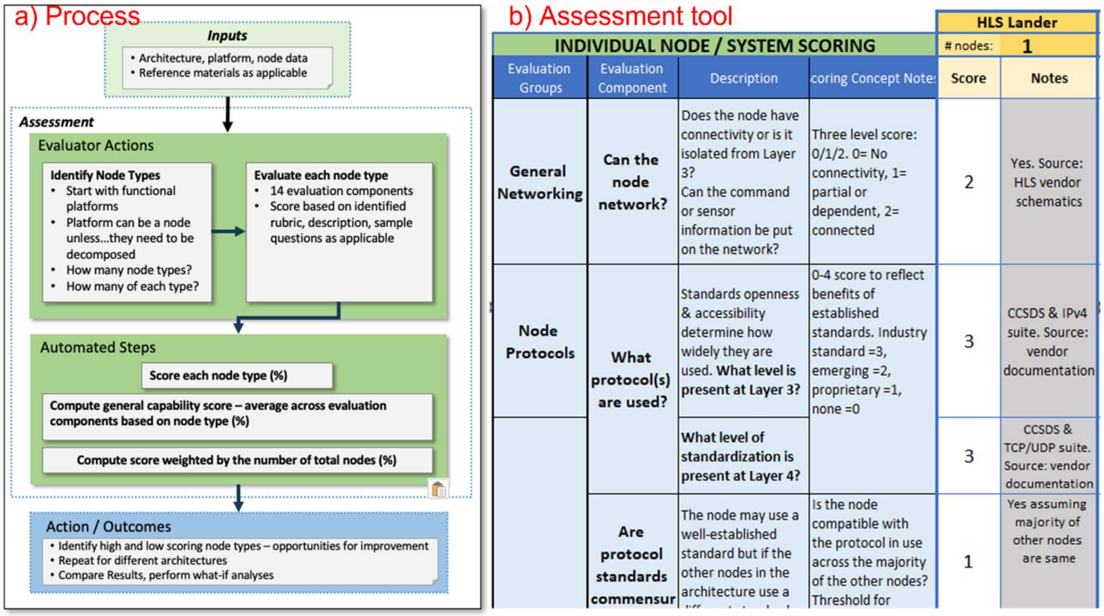
Interoperability viewed strategically focuses on sustaining operations between entities over the long-term as the SoS configuration changes dynamically, frequently to introduce new or improved capabilities. Tactically, interoperability focuses on short-term operations between entities (e.g., day-to-day steps and actions) where the SoS configuration is relatively static but flexibility for making operational changes must be supported. Strategies and tactics for measuring and assessing, verifying and validating, and sustaining interoperability were explored to assist in the deployment and evolution of LunaNet the Lunar Internet being developed to support NASA's Artemis Program to return humans to the Moon. These strategies and tactics aim at supporting full automation even during modifications to maintain 24/7 operations.

2.1 Measuring and Assessing Interoperability

We have developed a scoring rubric and tool for analyzing, measuring and assessing the interoperability of almost any architecure and individual nodes within it. The tool provides objective metrics (0-100% per normalized metric) and the results are useful for many purposes such as identifying optimum upgrades to directly improving the interoperability of a system or SoS.

The goals of the interoperability assessment are to provide a "baseline" score for the architecture and individual node interoperability, expose problem areas by identifying low-scoring nodes (interoperability bottlenecks), and to enable decision makers to evaluate alternatives for improving problem areas. In addition, the results allow systems engineers to reassess the architecture (and the nodes within it) periodically (e.g., before and after improvements) using quantitative metrics.

Assessing node-level interoperability: Analysts can define an architecture and it's nodes at any system level. Node-level elements evaluated by the tool include: *General communication or networking capabilities, Node network/communication protocol evaluation,* and *Node resilience*. Figure 3 shows the assessment flow and example scoring results for an Artemis-3 element .


a) Process
Inputs
• Architecture, platform, node data
• Reference materials as applicable
Assessment
Evaluator Actions
Identify Node Types
• Start with functional platforms
• Platform can be a node unless…they need to be decomposed
• How many node types?
• How many of each type?
Evaluate each node type
• 14 evaluation components
• Score based on identified rubric, description, sample questions as applicable
Automated Steps
Score each node type (%)
Compute general capability score – average across evaluation components based on node type (%)
Compute score weighted by the number of total nodes (%)
Action / Outcomes
• Identify high and low scoring node types – opportunities for improvement
• Repeat for different architectures
• Compare Results, perform what-if analyses
b) Assessment tool

INDIVIDUAL NODE / SYSTEM SCORING				HLS Lander	
				# nodes:	1
Evaluation Groups	Evaluation Component	Description	coring Concept Note:	Score	Notes
General Networking	Can the node network?	Does the node have connectivity or is it isolated from Layer 3? Can the command or sensor information be put on the network?	Three level score: 0/1/2. 0= No connectivity, 1= partial or dependent, 2= connected	2	Yes. Source: HLS vendor schematics
Node Protocols	What protocol(s) are used?	Standards openness & accessibility determine how widely they are used. **What level is present at Layer 3?**	0-4 score to reflect benefits of established standards. Industry standard =3, emerging =2, proprietary =1, none =0	3	CCSDS & IPv4 suite. Source: vendor documentation
		What level of standardization is present at Layer 4?		3	CCSDS & TCP/UDP suite. Source: vendor documentation
	Are protocol standards commensur	The node may use a well-established standard but if the other nodes in the architecture use a	Is the node compatible with the protocol in use across the majority of the other nodes? Threshold for	1	Yes assuming majority of other nodes are same

Figure 3 – Interoperability assessment: a) process and b) sample of scoring results

Assessing architecture-level interoperability: Once all nodes in an architecture are assessed and scored, groups of the same node type are combined and scored with other node groups using a mathematical algorithm, to produce a normalized composite Interoperability score (0-100). It is possible to modify the assessment tool to apply "organizationally" or "situationally" unique weighting factors to node and architecture scoring rubrics.

2.2 Verifying and Validating Interoperability

For complex systems, a key part of the engineering process is to develop, maintain and document system level verification and validation (V&V) spanning the entire project (concept to flight). For stand-alone flight projects, the V&V process is exercised in an iterative fashion for every aspect of the system design process.

The V&V processes are similar in nature, but have fundamentally different objectives:

- Verification shows compliance with requirements—that the system can meet each "shall" statement using various methods, e.g., test, analysis, inspection, and/or demonstration.

- Validation shows that the system accomplishes the intended purpose in the intended environment—that it meets the expectations of the customer and other stakeholders. Validation uses different procedures than verification but may use the same methods.

V&V methods may differ between incremental phases as capabilities and designs advance.

For SoS flight projects, V&V becomes more complex, particularly when mission-unique, non-interoperable services are included. These capabilities allow commercial vendors to offer unique, “niche” services that will require exclusive interfaces, putting the government at risk of becoming dependent on single vendors and incurring higher costs. From an interoperability standpoint, this may be acceptable, providing the unique services do not introduce issues with the interoperable core services. Whether operating in a single system or SoS environment, organizations can develop their V&V plans more productively from set of requirements shared by stakeholders and formed using good system engineering practices. Figure 1 is intended to give a glimpse into how interoperability can be formally verified and validated in this manner.

Unfortunately, this philosophy may be at odds with current initiatives to utilize “Public Private Partnerships” (PPPs) for acquisition. Commercial providers routinely develop and fly new systems that do not rely on extensive government-drafted requirements or test procedures. If the government does not include specific interoperability requirements in the procurements for applicable systems, upfront costs may appear to be lower, but can have unanticipated cost and schedule consequences during system and SoS testing as well as in operations.

Changes in Service Commercialization/Comingled Services: To reduce costs while improving overall end-to-end system responsiveness and latency, there has been a recent push to comingle space system services, operations, and data among service and data providers, (e.g., NASA, commercial, and international space agency partners). As a key element of this effort, there has been a particular emphasis on obtaining commercially provided SATCOM services wherever possible. Each commercial SATCOM service provider may offer a set of services ranging in interoperability compliance from full to none, driving the need for interoperability among the providers and users enabling user systems to autonomously “roam” or transition between providers. This approach requires the use of open, commercial standards and protocols in all systems expected to interoperate whether through contractual means or voluntary adoption.

Based on discussions with service providers, the high cost of space systems coupled with the business environment creates challenges in convincing SATCOM service providers to adopt common standards. Absent financial incentives, a compliance-driven acquisition strategy coupled with use of some form of industry-led consortium to voluntarily cooperate on common use cases, requirements, V&V approach, and coordinated testing would be favorable.

The government can provide value-added services for such an arrangement by establishing and moderating a “level playing field” that spans all providers without perceived bias, particularly in the area of providing a common test capabilities and metrics. Alternatively, a purely commercial solution can be considered for V&V testing and certification.

Changes in Automation Accommodation: Automation holds the promise of reducing overall system latency and cost, making it possible to conduct rapid response science in ways that were previously impossible. Without interoperable systems, automation may be limited within each system and expensive to implement across systems, particularly with multiple providers. Unfortunately, “Although several technology demonstrators for highly automated systems already exist, there is a severe lack of cost-effective, commonly accepted verification & validation (V&V) methods and the need for tools supporting these methods”. [3]

Changes in Security: Emerging cyber threats to systems in space coupled with changes in technology and capabilities and new, co-mingled commercial, civil, and defense systems and services, are going to require dramatic changes in space security architectures and systems. The United States, long a dominant space player, is witnessing unprecedented change in this arena.

From a V&V standpoint, given the potential for bad actors to inject malicious code into space systems, seize control, and disrupt operations with impacts ranging from service interruptions up to complete loss of the system, it seems obvious that full scale, independent testing of the end-to-end, multi-system security scheme in relevant conditions will be required.

2.3. Sustaining Interoperability

We divide sustainment into governance and implementation. Governance is essential in developing and sustaining interoperability. It is defined here but will be treated in a future paper.

In 2022 the Interagency Operations Advisory Group (IOAG), which promotes interoperabilty between civil space agencies, established the Committee to Study LunaNet Governance which defines governance in its Terms of Reference [4]: “The initial working definition of LunaNet governance is *the development and application by governments, the private sector and civil society, in their respective roles, of shared principles, norms, rules, decision-making procedures, and organizations that shape the evolution and use of LunaNet in the context of the relevant international legal framework*.” This is based on the definition of Internet governance established by the UN-initiated *World Summit on the Information Society*. [5]

Implementation approaches are addressed in this section for sustaining interoperability at the international architecture level and coordinating evolution of interoperable services.

Sustaining Interoperable Architecture: The first aspect of a stable ecosystem that sustains interoperability over an extended period of time occurs within the community of stakeholders concerned with defining the interoperable architecture and maintaining that architecture as it evolves over time due to changes in User community needs, standards revisions , technology improvements, introduction of new capabilities, and obsolete capability retirement.

Figure 4 depicts the simplified concept for maintaining configuration control of the architecture via a notional Architecture Board (AB). Members of the Architecture Board would be subject matter experts in various aspects of the architecture including operations, security, global and local architecture, Provider/User interaction, etc. Participation in this Board must be broad and allow a wide assortment of stakeholders access to information that defines the architecture, the status and details of changes being considered, educational and training material, access to the accumulated body of information related to the architecture for historical and trend analysis, and other material.

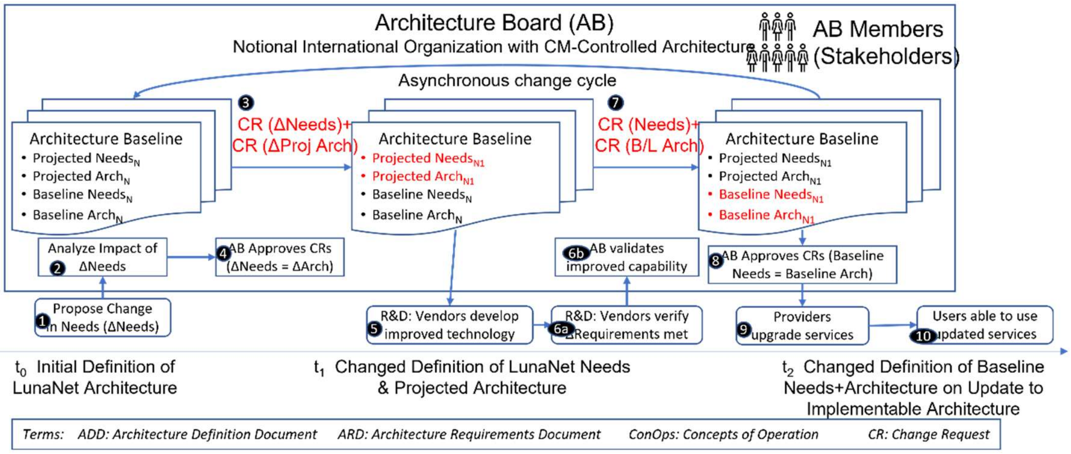

Architecture Board (AB)
Notional International Organization with CM-Controlled Architecture
AB Members (Stakeholders)
Asynchronous change cycle
Architecture Baseline
• Projected Needs$_N$
• Projected Arch$_N$
• Baseline Needs$_N$
• Baseline Arch$_N$
3
CR (ΔNeeds)+
CR (ΔProj Arch)
Architecture Baseline
• Projected Needs$_{N1}$
• Projected Arch$_{N1}$
• Baseline Needs$_N$
• Baseline Arch$_N$
7
CR (Needs)+
CR (B/L Arch)
Architecture Baseline
• Projected Needs$_{N1}$
• Projected Arch$_{N1}$
• Baseline Needs$_{N1}$
• Baseline Arch$_{N1}$
2 Analyze Impact of ΔNeeds
4 AB Approves CRs (ΔNeeds = ΔArch)
6b AB validates improved capability
8 AB Approves CRs (Baseline Needs = Baseline Arch)
1 Propose Change in Needs (ΔNeeds)
5 R&D: Vendors develop improved technology
6a R&D: Vendors verify Requirements met
9 Providers upgrade services
10 Users able to use updated services
t_0 Initial Definition of LunaNet Architecture
t_1 Changed Definition of LunaNet Needs & Projected Architecture
t_2 Changed Definition of Baseline Needs+Architecture on Update to Implementable Architecture
Terms: ADD: Architecture Definition Document ARD: Architecture Requirements Document ConOps: Concepts of Operation CR: Change Request

Figure 4. Sustaining Interoperable Architecture at the Architecture Control Level

The change control process outlined in Figure 4 begins (Step ❶) with stakeholder(s) introducing information describing changes in needs perceived to be emerging or in development such as increased demand for specific services or under different scenarios or environmental conditions. The AB must be able to draw on expertise capable of evaluating the impacts of these proposed changes in needs. The evaluation (Step ❷) must validate the proposed changes and ascertain whether one or more viable approaches exist capable of meeting those needs. This includes reviewing the Architecture Baseline at time N (Step ❸) and the current state of capacities and capabilities over the set of Providers to quantify differences. Assuming a favorable outcome in this analysis, the result yields proposed changes to the Architecture Baseline (Step ❹) to capture the change in Projected Needs. This would include documentation of analysis of alternatives and test results. The result is captured in a Change Request (CR) that proposes specific changes from the baselined Projected Needs$_N$ and Projected Architecture$_N$ – the accumulated result of prior approved CRs – that will produce an updated baseline including Projected Needs$_{N+1}$ and Projected Architecture$_{N+1}$. This CR advances to the AB to formally approve the CR, thus recognizing the revised User needs and architectural response. The Projected Architecture should be defined independently of the technologies that may be used.

Industry responds to the revised Projected Needs by investing in R&D which may generate competing means of meeting the needs (Step ❺). One or more feasible solutions emerges and is verified by industry (Step ❻a) as complying with the Projected Architecture, possibly with proposed modifications. Industry submits proof of their ability to the AB (Step ❻b) and the AB again relies on expertise to validate industry's claims. The AB generates a second set of proposed changes to the baseline via a CR that will revise the Baseline Needs$_N$ and Baseline Architecture$_N$ (Step ❼). The AB approves the CR (Step ❽) producing Baseline Needs$_{N+1}$ and Baseline Architecture$_{N+1}$. This enables the Provider community to upgrade their systems and begin providing the enhanced services (Step ❾) at the performance level required to meet the increased User needs. Users can request services from Providers offering enhanced services (Step ❿).

Sustaining Interoperable Operations: Interoperability is more than an abstract ideal. It requires effort to develop and sustain over the life of a SoS that evolves over time. Concepts associated with sustaining interoperability are integral to the definition. Aspects related to sustainment must be feasible, affordable, and incorporated from the outset of establishing the SoS. This section describes key concepts for sustainable interoperability and demonstrates their feasibility.

Figure 5 shows a notional process used to maintain interoperability with little or no service interruption while propagating a revision to one service. Interactions occur between a SDO that owns the standard(s) defining the interoperable services offered by a set of Service Providers to a set of Service Users. We treat these users as space systems. They are launched with a configuration of hardware (HW) and software (SW) with no assumed in-space servicing capability. Consequently, flight SW can be modified from the ground-based Mission Operations Center (MOC) while flight HW may not be modifiable or have limited modification ability, e.g., using Field Programmable Gate Arrays. The ability to modify the flight system's configuration in space constrains the ability to evolve the space system in operation, e.g., via a SDR. For this example, only the flight SW is assumed to be modifiable and service modifications are similarly constrained.

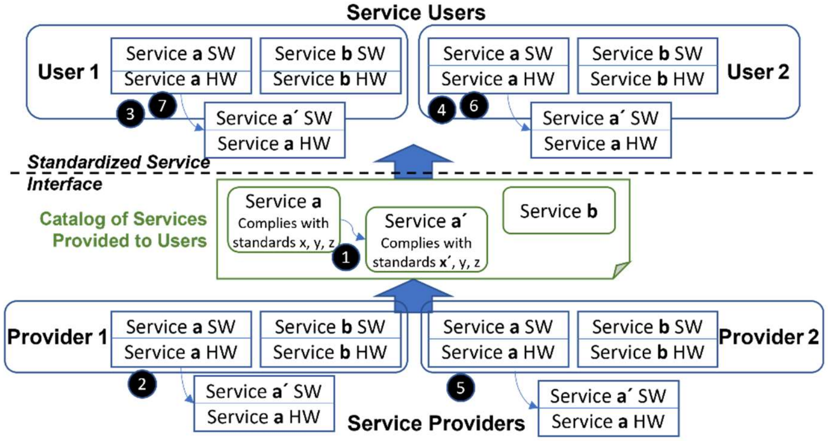


Figure 5. Process Preserving Interoperability Through Asynchronous Change

Assume Users 1 and 2 have SDRs and have Service Level Agreements (SLA) with both Providers 1 and 2 to use Services **a** and **b**. In Step ❶, the SDO revises its standard from Revision **x** to **x´** which impacts Service **a.** The revised standard is published and the user community is notified. There is no impact to current Service **a** between Providers 1 and 2 and Users 1 and 2. In Step ❷, Provider 1 revises its software to deliver Service **a´** per standard **x´** and announces availability of Service **a´** to users while continuing to provde Service **a**. In Step ❸, User 1 modifies its software from receiving Service **a** to Services **a** or **a´** and then changes its SLA with Provider 1 to receive Service **a´**. User 1 uses Service **a** from Provider 2 and Service **a´** from Provider 1. No impact occurs to current Service **a** between Providers 1 and 2 and User 2. In Step ❹, User 2 modifies its software from receiving Service **a** to receiving Services **a** or **a´** and then changes its SLA with Provider 1 to receive Service **a´**. Users 1 and 2 use Service **a** from Provider 2 and Service **a´** from Provider 1. Again, no impact occurs to current Service **a** between Provider 2 and User 2. In Step ❺, Provider 2 modifies its software from delivering Service **a** to **a´** while continuing to provide Service **a** to Users 1 and 2. In Step ❻, User 2 modifies its software from receiving Service **a** to Services **a** or **a´**, then changes its SLA with Provider 2 to receive Service **a´**. At this point, User 1 uses Service **a** from Provider 2 and Service **a´** from Provider 1. Finally, in Step ❼, User 1 already has software updated to handle Service **a´.** It updates its SLA with Provider 2 to change from Service **a** to Service **a´.** User 2 uses Service **a´** from Provider 2.

Key points in this process:

- The key requirement in this process is that Providers and Users must maintain two active versions of a service throughout the transition. If a problem occurs during the process, Providers and Users must be able to roll back to the prior stable version.
- A User could implement the transition in two sub-steps rather than one (Steps 2 and 6). The first sub-step would be to modify the flight SW to accept the format and characteristics of Service **a´** while continuing to behave as if it is still receiving Service **a**, i.e., by ignoring differences between Services **a** and **a´**. The second sub-step would complete the process of changing the flight SW to operate on Service **a´** including differences from Service **a**. This might offer tactical advantages to the user by delaying the cost and effects of Service **a´** while keeping up to date with respect to the Service Provider's interface. Several modern languages have built-in capability for version transparency. For example, a browser that handles HTML 4 can accept HTML 5-formatted input and will ignore unrecognized capabilities in HTML 5.
- On process completion, Users 1 and 2 and Providers 1 and 2 no longer need to retain the Service **a** SW**.** The SW can be discarded enabling the systems to reclaim storage space.

Further scenarios can be devised to accomplish a "make before break" approach to sustainment that preserves interoperability throughout the evolution process.

3. Conclusion

Interoperability has been converted here from a high-level concept into an engineering entity amenable to systems engineering methods. This enables systems to be modeled, developed, tested, operated and evolved with high assurance of achieving quantifiable and verifiable goals.

Is interoperability worthwhile? While commercial terrestrial SATCOM has virtually no interoperability among providers, NASA and its international partners envision the cislunar environment as being born with interoperable communication and PNT services offered by several providers to a rapidly increasing set of missions. Interoperability becomes a game-changer for the cislunar community if it can be sustained due to the many benefits it offers including:

- Commercial providers have the opportunity to pursue the entire lunar market as users come to depend on interoperable services
- Users see increased capacity, coverage, availability and reliability of communication services with reduced latency because of the flexibility to use any or all providers as well as enhanced ability to recover from off-nominal conditions
- Users see higher accuracy PNT service for position and orbit determination by combining broadcast signals from all providers
- A competitive cislunar market offers the benefits of access to capital for further investment, stimulus for R&D, and robust management of cost and scehdule risks

LunaNet will offer new services in space for space systems, rather than over space links for terrestrial systems. Interoperability in LunaNet will enable broad use of new capabilities including Delay/Disruption Tolerant Networking with in-orbit data storage, distributed processing and cloud functions, Internet of Things, new science services and supports autonomy in mission systems. The full version of the paper is available from the authors.

4. References (All references accessed 10 September 2023)

[1] OMG Systems Modeling Language, http://www.omgsysml.org/.
[2] Source: IEEE, "A Compilation of IEEE Standard Computer Glossaries," New York, Standard 1990.
[3] Enable-S3, Testing and Validation of Highly Automated Systems: Summary of Results, May, 2019, https://www.tugraz.at/fileadmin/user_upload/Institute/IHF/Projekte/ENABLE-S3_SummaryofResults_May2019.pdf.
[4] Committee to Study LunaNet Governance Terms of Reference (ToR), June 12, 2023, https://www.ioag.org/Documents/Committee%20to%20Study%20LunaNet%20Governance%20ToR_for%20signatures%20-%20signed.pdf.
[5] Tunis Agenda For The Information Society, WSIS-05/TUNIS/DOC/6(Rev.1)-E, November 18, 2005, https://www.itu.int/net/wsis/docs2/tunis/off/6rev1.pdf, clause 34.