AEROSPACE
& DEFENSE

# HOLISTIC MOSA CONSIDERATIONS FOR IMPROVING ACQUISITION SUCCESS

GARRETT THURSTON

DASSAULT SYSTEMES

# 1. INTRODUCTION

Modular Open Systems Approach (MOSA) has been part of the US Defense Authorization since 2017, and has become part of the milestone decision authority (MDA) statutory and regulatory obligations of acquisition category (ACAT) programs as a part of the adaptive acquisition system.[1]

Figure 1 is a structured goal model for Acquisition Success. In each of the outlined areas: Affordability, Acquisition Tradecraft, Capability Evolution, and Overmatch, there are associated sub-goals. One of the things that we highlighted in the paper "Transforming the Acquisition Value Network"[2] is that it is also important to have explicit goals to transform, evolve, and change. It is beyond the scope of this paper to dive into the totality of transformation, but this diagram establishes the context for MOSA, which we organized under the Affordability goal structure, as seen in Figure 2.

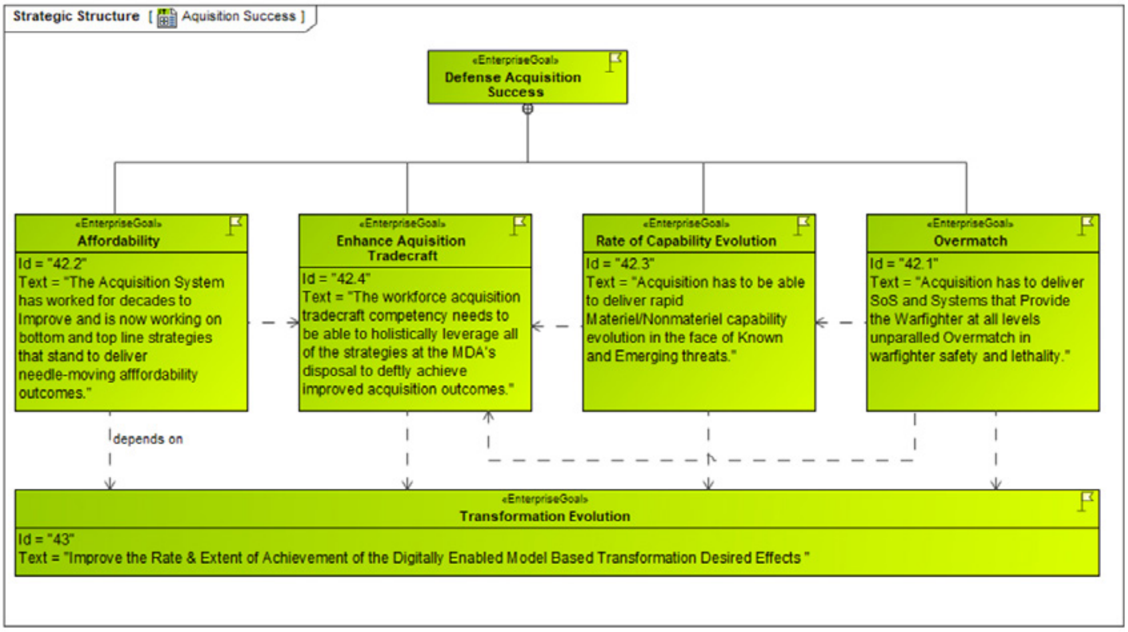



Figure 1: High-level MOSA Acquisition Success Context Figure 2 Dives Into the Affordability Elements Including MOSA

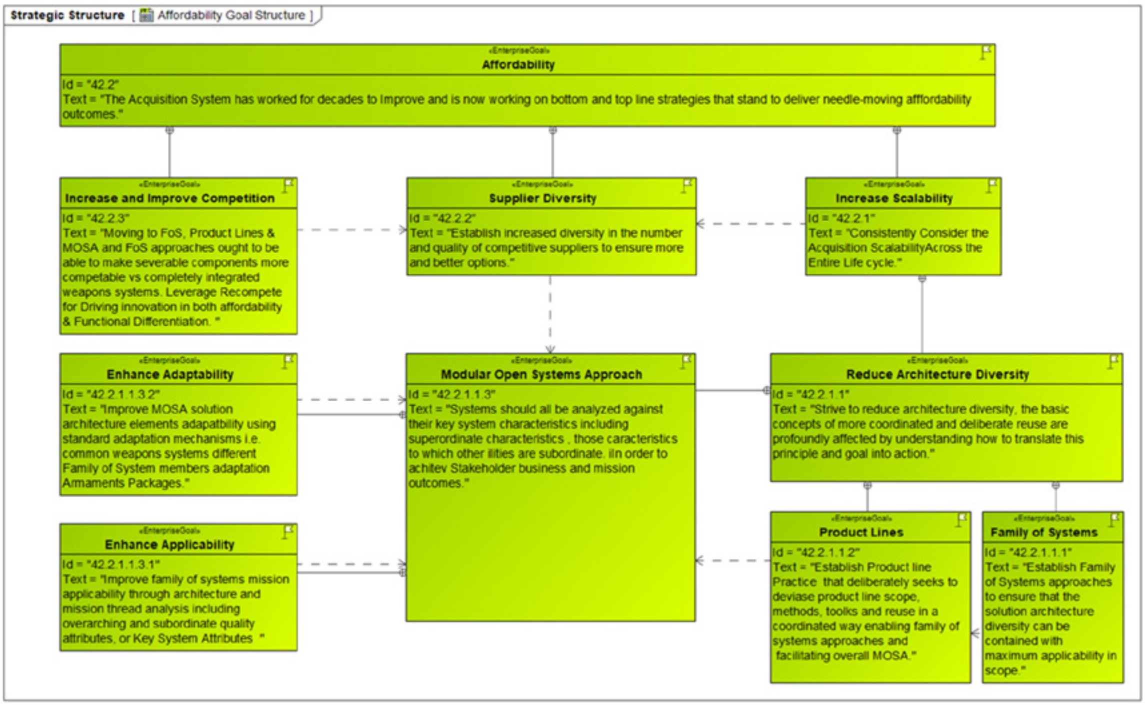



Figure 2: Affordability Element of Acquisition Success that Includes MOSA Structured Goals Contexts

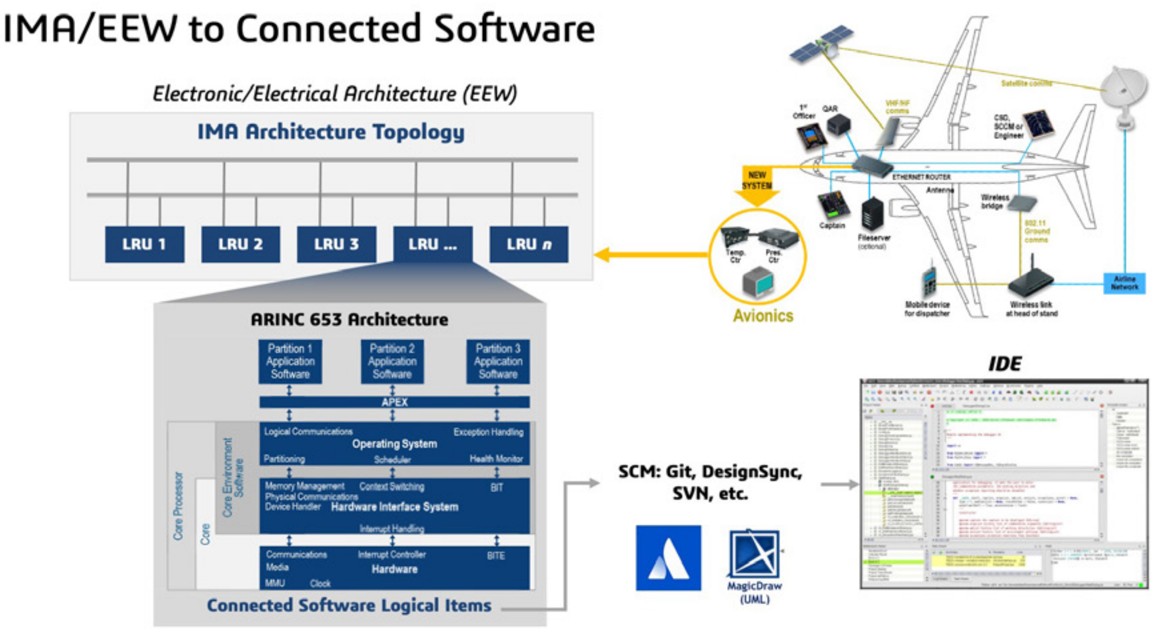




Figure 3: Illustration of the System of System from the System to the IMA Super Architecture, to the Hardware and Software Item Levels

System characteristics typically apply to the whole system. However, the MOSA quality characteristic is to create modules that are integrated into a whole. By its very nature, the architect formulates into severable modules.

A common and generally accepted architecture and design understanding of an Open System is that the system creators build the system from components they deliberately architect, design, and implement to be constructive and interchangeable. In addition, open systems clearly define and openly publish the interfaces to these interchangeable components. This architecture approach enables anyone to produce a new component that enhances the capabilities of the system into which it is incorporated.

The business model that supports this enterprise is one of mutual benefit, where the supplier of the Open System and the supplier of the open system compliant component enhance each other's market position.[3]

---

[1] DODI 5000.88 Engineering of Defense Systems 2020. In accordance with the authority in DoD Directive (DoDD) 5137.02 and the guidance in Section 133a of Title 10, United States Code (U.S.C.), this issuance establishes policy, assigns responsibilities, and provides procedures to implement engineering of defense systems.

[2] Thurston, Garrett, Transforming the Acquisition Value Network, Dassault Systèmes Whitepaper, 2022.

[3] Jean Tirole: Market Power and Regulation, Alfred Nobel Prize, 2014, Two-Sided Economy.

## 2. OPEN SYSTEM MODULARITY

An Open System is modular in characteristic when the system creator deliberately architects, designs, and implements the system and its constitutive modular components in such a way that its modules have precisely defined and publicly owned interfaces, where these interfaces facilitate independent suppliers to provide improved capability by providing innovative, compatible modules. We refer to such modular architecture structure with open interfaces as an Open Architecture.

It is crucial that one understand not only the concepts of modularity and openness and the benefits thereof but also the means by which architects synthesize such open systems. Modular Open System Synthesis is an integral part of System Engineering, Architecture, Detailed Design, and Implementation. The MOSA establishes a plausible argument that the acquirer stands to realize MOSA benefits by following the open system's principles, in particular, that the system architect clearly defines an Open System Architecture.

It is not essential that all the components of an Open System are open; indeed, it would be impossible to realize this ideal in most commercial scenarios. The openness refers to the component interfaces. These must be sufficiently generalized and well-defined, whereby producing compatible modular components is both practical and economically sensible.

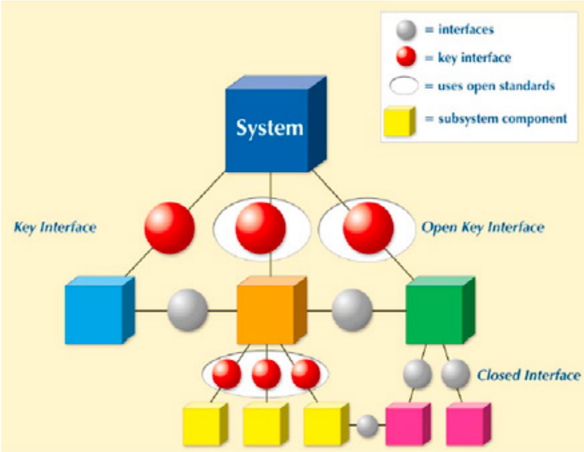



Figure 4: MOSA Conceptual Elements with Emphasis on Defined and Open Interfaces[4]

## 3. MODULARITY CHARACTERISTICS

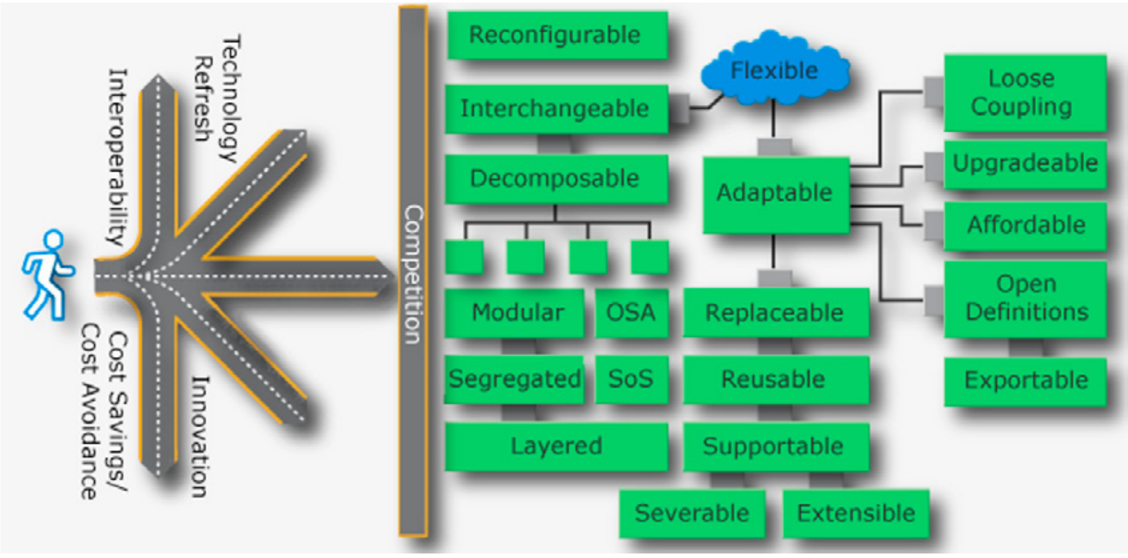



Figure 5: Generally Accepted Modular Open Systems Approach Motivations and Characteristics

The business motivations for moving towards MOSA include the rate of innovation and evolution, improved interoperability, and affordability. The approach intent promotes competition. MOSA affords such business outcomes through the ascribed architecture characteristics.

Modularity is a superordinate system characteristic, meaning that the architecture process drives modularity by subordinate system characteristics. This is important to understand since such characteristics are the basis for architecture evaluation methods that architects and stakeholders use to validate architectures well in advance of their design and implementation as prescribed for assurance planning.[5]

To illustrate this point, we can see that dependability is a superordinate system characteristic, while in the case below, confidentiality is a subordinate characteristic.

The modularity of the system has to satisfy multiple competing attributes, for example, survivability relates to the architecture principle of physical separation and has tradeoffs with schedulability and busload.

It is illustrative to look at some of the threads that are active during the system engineering process to factor in all of the competing concerns that shape MOSA.[6]

[4] Program Manager's Guide to Open Systems Version 2.0 2004, & Open Systems Architecture Contract Guidebook for Program Managers, Version 1.1, May 2013.

[5] SAE AIR6218 Constructing Development Assurance Plan for Integrated Systems, 2018.

[6] [Thurston, Garrett, Use of ACVIP Containment of the Accumulation of Program Technical Debt using AADL Implemented]() on Dassault Systèmes' **3D**EXPERIENCE Platform Software Engineering Institute, 2021.

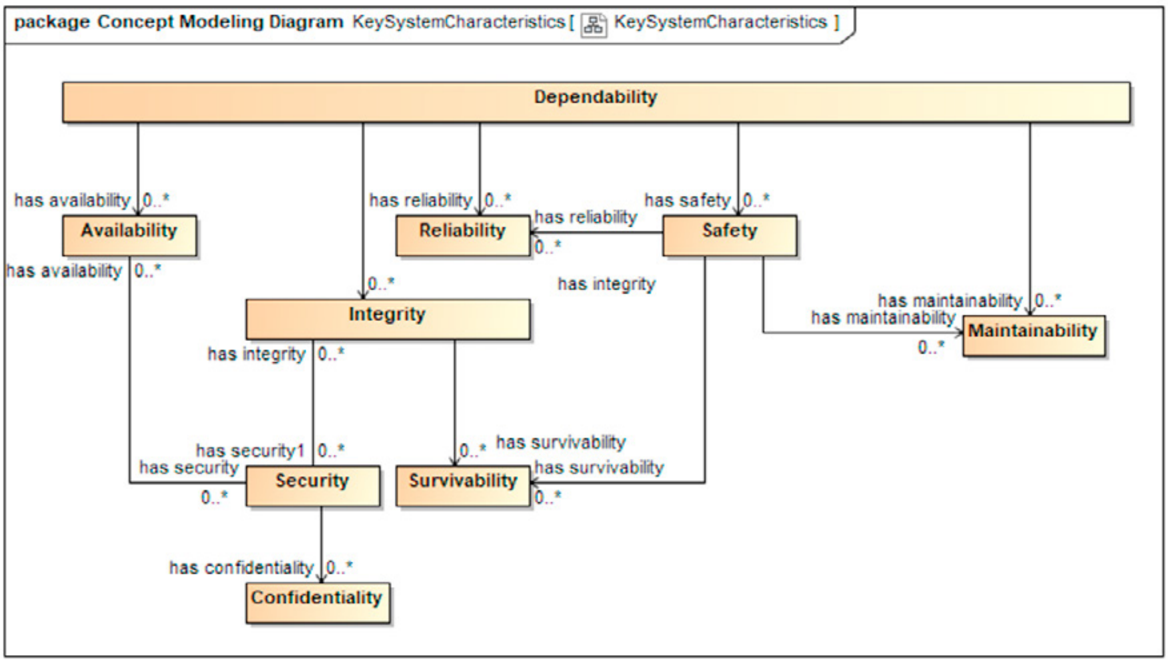



Figure 6: Illustration of Superordinate and Subordinate Quality Attributes/System Characteristics

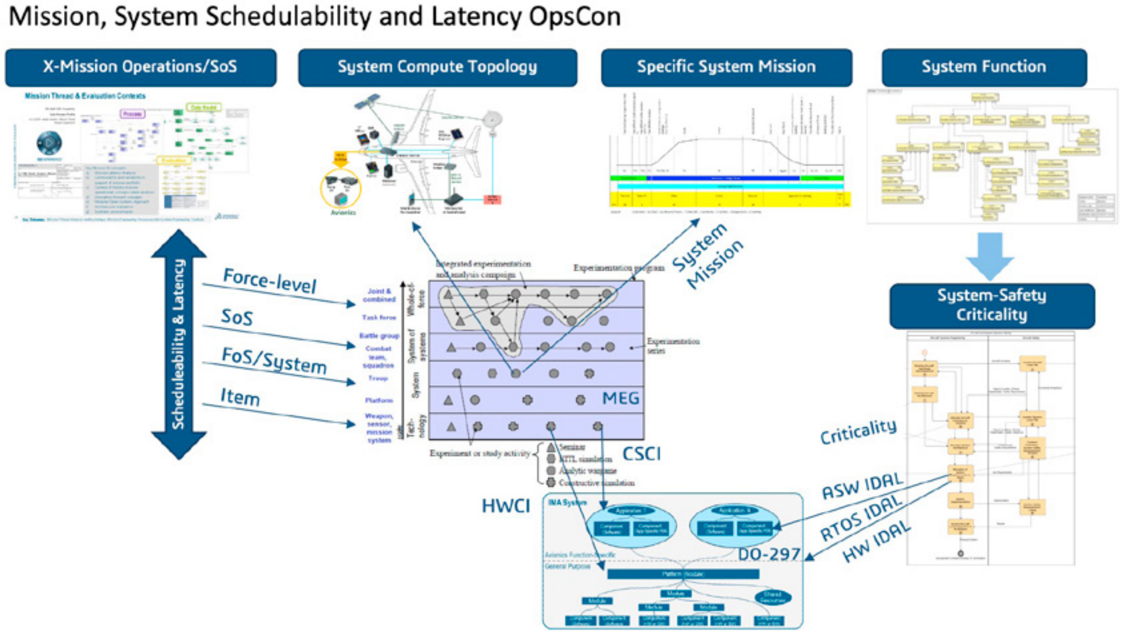



Figure 7: System Engineering Thread from Aggregated SoS Mission through System-Specific Considerations, Specific System Missions, System Function, Criticality and FACE/DO-297 Common Compute Topology and Partition Allocation.[7]

Some example considerations when conducting ARINC 653 partitioning to solve the resource allocation problem in satisfaction of various whole system characteristics criticality comes into play, in the first allocation on the left of Figure 8 there are no constraints on the MOSA CSCI (Computer Software Configuration Item) is a level A criticality item development assurance level (IDAL). In the second case, when we seek to allocate the CSCI, we see it is criticality B. In this case, we have to constrain the allocation options because there is a criticality clash in the first partition from the left; it can't be allocated to a partition with a different criticality. In the Functional and Logical Application, we achieve this by checking for the module criticality at the time of allocation and preventing this conflict and notifying the architect of the conflict.

In practice, this is important because we would want to use the **3D**EXPERIENCE® platform and trade automation applications like Process Composer to explore the system characteristic "Ilities" trade space. It may be interesting to note that an ARINC 653 partition is objective to criticality. Both the hardware and RTOS software that is intended to support criticality A application software have to be Criticality A themselves; then, once allocated to a partition, the partition takes on the criticality of the first allocated CSCI.[10] This allocation constraint goes both ways; if the partition contains A, the architect can't subsequently assign CSCIs of <A (i.e., B-E), and if B, the architect can't assign A or C-E, and so on.

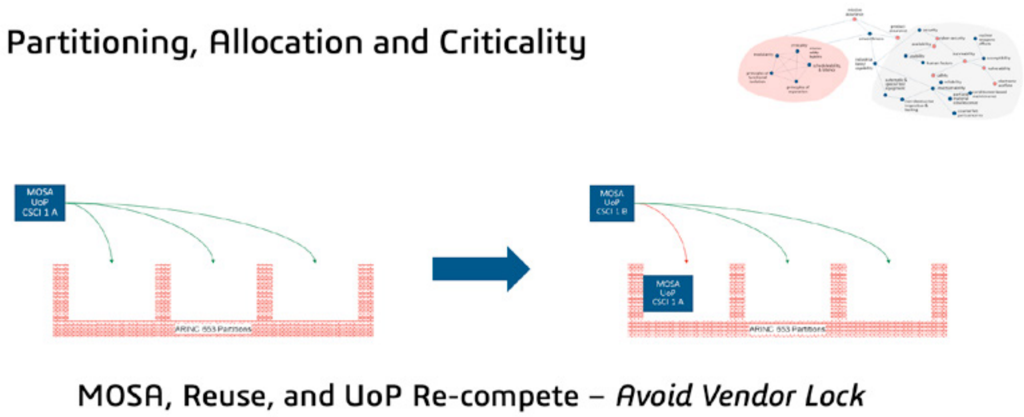



Figure 8: Computer Software Configuration Item (CSCI) Partition Allocation

[7] Office of the Deputy Director for Engineering, OUSDR&E Mission Engineering Guide, 2020.

[8 ARINC 653 (Avionics Application Software Standard Interface) is a software specification for space and time partitioning in safety-critical avionics real-time operating systems (RTOS). It allows the hosting of multiple applications of different software levels on the same hardware in the context of an Integrated Modular Avionics architecture.

[9] In DO-297 this is an application software module, in FACE this is a Unity of Portability (UoP), and the Criticality varies depending on the organization or agency the FAA in ARP-4754A there are 5 Levels of Criticality that are according to the Top-level Failure Condition Severity (Catastrophic-A, Hazardous/Severe Major-B, Major-C, Minor-D, and No Safety Effect – E. These Functional Development Assurance Levels (FDAL) eventually influence the rigor of the Item Development Assurance Level (IDAL) of the Software and Hardware Items. This aligns with FAA AC 23-1309-1E SYSTEM SAFETY ANALYSIS AND ASSESSMENT FOR PART 23 AIRPLANES where these approaches are safety focused, other criticality systems look at mission, or logistic critical as well as others.

When considering schedulability, the system architect has to contend with three schedulability challenges:

- Intra-partition communication between CSCIs (same Criticality Level) lowest latency other than being in the same CSCI
- Inter-partition communications between CSCIs of the same or different criticality
- Inter-LRU protocol-dependent bus communications schedulability and Bus traffic

For example, as reflected in Figure 9, the Architect severs these modules of the same criticality, same rate, for some other quality attribute than schedulability; what are the implications, and does it make sense?

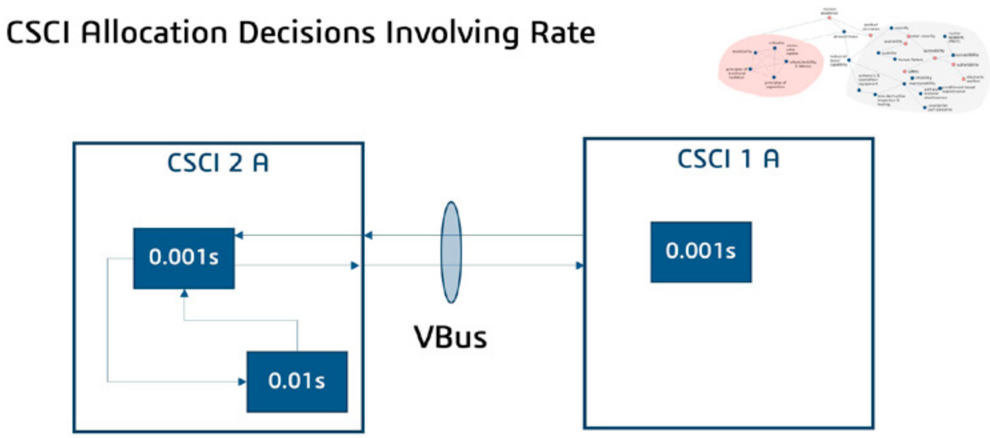



Figure 9: Severed Modules of the Same Criticality for the Purposes of Satisfying some Quality Attribute with Communication via some means.

This is an easily understood use case, but it illustrates the architecture evaluation considerations involving timing and also helps relate superordinate (modularity) and subordinate (schedulability/latency) quality attributes.

Figure 10 reflects a specific architecture alternative where the CSCIs of the same criticality are allocated to different partitions in different Common Compute Resources within the system and are connected through the protocol-dependent signal bus eliminating the intra- and inter-partition communications options. This architecture alternative leaves us with schedulability and busload analysis and exploration of the bandwidth trade-off analysis of the different deterministic[11] communications protocols available.

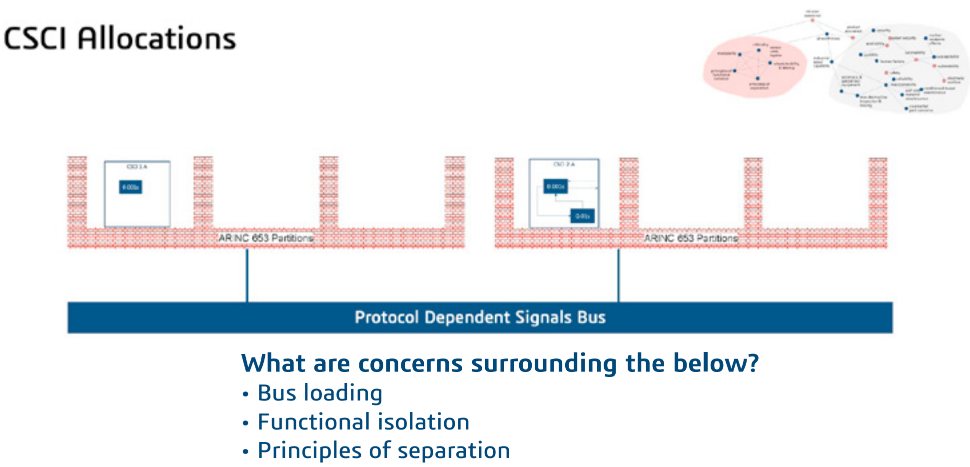



Figure 10: Architecture Alternative for the Severed MOSA CSCIs of the Same Criticality Allocated to Different Partitions on Different Common Compute Resource LRUs Protocol Dependent Signal Communication Means

[10] This is the source of the trick question I sometimes ask: "What is the criticality of the hardware and RTOS?".
[11] For example Avionics Full-Duplex Switched Ethernet (AFDX), also ARINC 664.

Figure 11 is only different from Figure 10 in that the CSCI on the left is criticality B, so it is different from A. In this case, our architecture alternatives include protocol-dependent signal bus communications and inter-partition communications since, because of the difference in their criticalities, they cannot be located in the same partition. This trade is not why the architect laid out this alternative; here, we want to explore the option of severing these modules differently. In this case, the system safety Functional Hazard Assessment (FHA) process has determined that the criticality is of the CSCI 2 as B. This is a technical assessment of the preliminary architecture. Might it be reasonable to evaluate the alternative of moving the high-rate thread of CSCI 2 into CSCI 1 and apply the additional development assurance level rigor in the interest of saving on that high-rate inter-partition or protocol-dependent signal bus traffic?

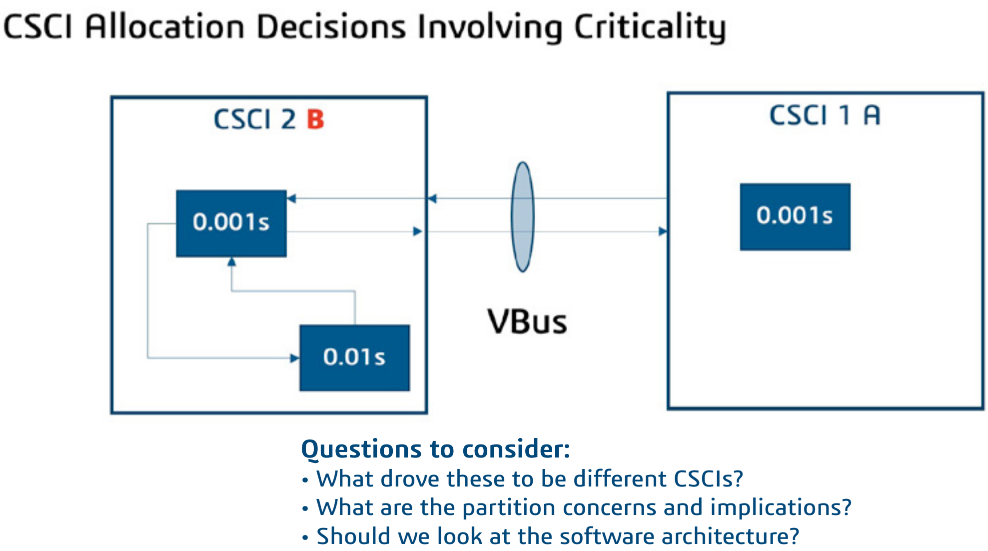



Figure 11: Severed Modules of Different Criticality for the Purposes of Satisfying Quality Attributes With Communication via Some Means.

The **3D**EXPERIENCE platform establishes shared data and marshals common objects and applications services into digitally connected technical applications and software infrastructure. In order to gain insight into schedulability during the early lifecycle stages, the architect leverages Cameo Enterprise Architect and a Future Airborne Capability Environment (FACE[12]) plugin from MITRE that integrates with OSATE and generates AADL[13] that is subsequently analyzed using OSATE2[14] or AADL Inspector.[15] OSATE also supports Ocarina code generation for several target environments.

Later in the lifecycle, as the system definition matures from system architecture into detailed design, the **3D**EXPERIENCE platform supports the design specificity tying together the functional and logical definition and the System Safety hazard assessments that include the protocol-dependent signals analysis where the architect allocates the software functionality or modular units of portability (UoP) across the IMA architecture topology.

In the case of the survivability-schedulability tradeoff identified earlier, the architect manages the physical separation of the Functional & Logical Application decomposition of the system through the spatial allocation reservation application and establishes the relationship to the physical model system elements. The reliability and safety architect uses the System-Safety application to conduct FHA, D/P-FMEA[16], and the IMA architect employs the Electrical and Electronic Architecture Application to facilitate protocol-dependent signal analysis across the IMA, or common compute topology. As desired, the architect can automatically generate AADL code during this lifecycle phase to ensure that the design schedulability is within the timing budgets established by the architecture earlier in the lifecycle.

Resource Allocation Problem (RAP) busload analysis tradeoffs are supported by the Process Composer Application, which also provides the authoritative source of truth (ASoT)[17] for configuration managing these trade studies. Larger architecture option Analysis of Alternatives (AoA) branched variant studies are also managed using the Variant Configuration Management application. The System Engineering community considers both these types of trade studies validation activities as the product specification and rationale move down the left-hand side of the system engineering V, as described in Section 10 MOSA Value Network ASoT Governance and Management. Product development programs manage both Validation and Verification objective evidence in the functional, logical, and physical contexts using the Test Strategy and Test Management Applications.

[12] FACE™ (Future Airborne Capability Environment) approach integrates technical and business practices that establish a standard common operating environment to support portable capabilities across avionics systems. The FACE Technical Standard defines the requirements for architectural segments and key interfaces that link the segments together. This enables the reuse of capability-based software components across different hardware computing environments. The idea is to avoid "reinventing the wheel" for every new platform system. It also enables rapid replacement of older software and insertion with new and improved capabilities throughout the system lifecycle. The FACE approach is relevant to both legacy systems and future systems, including new system designs, system-level upgrades, and component upgrades.

[13] SAE AS 5506 Architecture Analysis and Design Language 2011.

[14] Open Source AADL Tool Environment.

[15] Ellidiss Technologies.

[16] FHA and FMEA conducted in accordance with SAE ARP4754A or J1739. FHA determines module criticality.

[17] ASoT concept model indicates that it is legitimate, has an authority, is current, is trustworthy. An authoritative source of truth is an entity such as a person, governing body, or system that applies expert judgment and rules to proclaim a digital artifact is valid and originates from a legitimate source.

# 4. MOSA FRAMEWORK

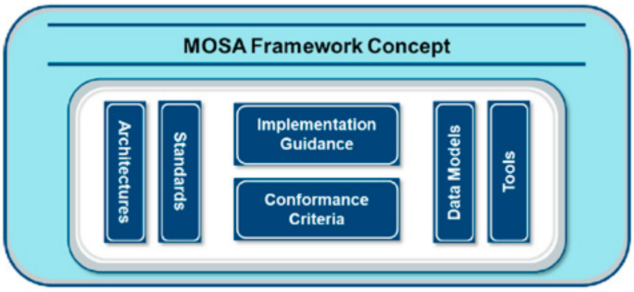



Figure 12: Illustration of MOSA Framework Elements

The MOSA architect must create a "super architecture," the architecture of the system as a whole that models the potential for evolution and innovation that would make producing new system components attractive to an independent organization. To help build MOSA consistency, it is crucial for the NATO ecosystem to establish suitable MOSA standards, reference architectures, conformance criteria, implementation guidance, schema, and tools. To this end, the US DoD has used the term "framework" to identify proposed MOSA solutions that satisfy similar technical requirements and common elements across related applications within a domain.[18]

The MOSA framework itself needs to be a value network governed and managed set of assets very much following the same pattern as outlined in Section 10 MOSA Value Network ASoT Governance and Management.

It is important to note that Open Systems and Open Source Software are not the same things, although the ideas and the principles are clearly related. In addition, though many standards and guidance structures can guide the MOSA software approaches, it is essential that the innovating organization appreciate that the application of Open Systems Principles and their architecture influence should not be limited to software. In fact, the aforementioned super architecture and the associated standards explicitly extend into hardware, and even then, the principles should be extended to the total of the system of systems, not just complex hardware and software.

In addition to the technical characteristics and principles of the modular open systems approach, it is crucial that the innovating organizations appreciate that the governance and management of the standards and system synthesis execution and conformance be an integral part of the integral processes.[19]

Examples of such approaches include DO-255[20], DO-297[21], ARINC-653[22], ARINC 661[23], FACE[24], & HOST[25].

[18] Modular Open Systems Approach (MOSA) Reference Frameworks in Defense Acquisition Programs, 2020.

[19] Program Manager's Guide to Open Systems Version 2.0 2004, & Open Systems Architecture Contract Guidebook for Program Managers, Version 1.1, May 2013.

[20] **DO-255** provides the Requirements Specifications for the Avionics Computer Resource (ACR) intended to facilitate certification efficiency and economy of scale for the computer platform. These Requirements Specifications define a computer platform suitable for hosting multiple, independent software applications and serve as an enabling step towards standardized, re-usable avionics software applications.

[21] **DO-297** contains guidance for Integrated Modular Avionics (IMA) developers, application developers, integrators, certification applicants, and those involved in the approval and continued airworthiness of IMA systems in civil certification projects. This guidance focuses on IMA-specific aspects of design assurance. IMA is described as a shared set of flexible, reusable, and interoperable hardware and software resources that, when integrated, form a platform that provides services, designed and verified to a defined set of requirements, to host applications performing aircraft functions. The primary industry-accepted guidance for satisfying airworthiness requirements for IMA components is included and it describes application properties as they relate to their integration with a platform.

[22] Ibidem **ARINC 653** reference earlier.

[23] **ARINC 661** standard normalizes the definition of a Cockpit Display System (CDS), and the communication between the CDS and User Applications (UA) that manage aircraft avionics functions. The binary Definition Files (DF) completely define the Graphical User Interface (GUI).

[24] Ibidem **FACE™** referenced earlier.

[25] **HOST** is an open technical standard that enables an objective way to realize the Modular Open System Approach initiative goals. HOST lays out requirements that a program manager or integrator can leverage to create a verifiably open system. By following the HOST's open system requirements, the tenets of modularity, interoperability, and upgradeability are capable of a high degree of confidence. As an open standard, all parties utilizing HOST will also have free, unlimited access to the information needed to create HOST conformant components.

# 5. MOSA BUSINESS DRIVERS FROM NATO STANAG 4626[24]

The three principal drivers for the architecture are:

1. Reduced Life Cycle Cost: A major objective is to reduce the accumulated costs over the life cycle of a system, i.e., the development, acquisition and support costs.
2. Improved Mission Performance: The system must be capable of fulfilling the missions and satisfy all possible airborne platforms in terms of functionality, capability, reliability, accuracy, configurability and interoperability under the full scope of operating conditions.
3. Improved operational performance:

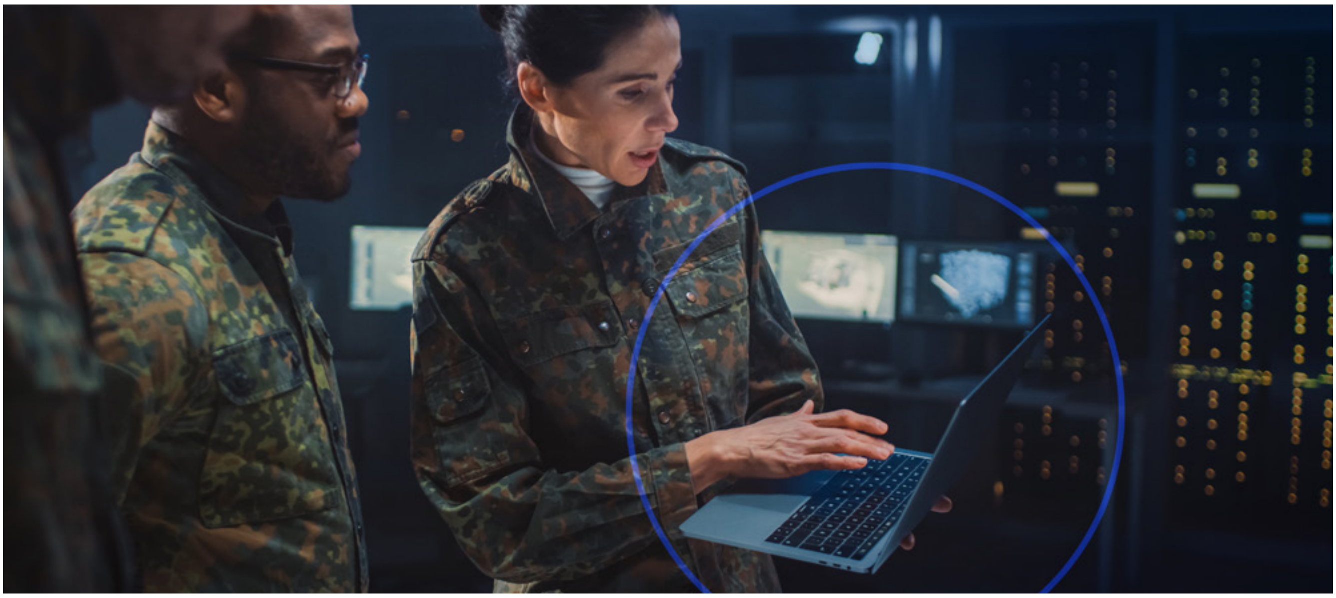
- The goal adopted is that the system (aircraft) should achieve a combat capability of 150 flying hours or 30 days without maintenance, with an availability of at least 95%.
- This goal far exceeds that achievable today, and an IMA System will be required to exhibit fault tolerance so that it can survive the occurrence of faults with the required level of functionality.

| Architectural Principles | Mission Performance | Operational performance | Life Cycle Costs |
|---|---|---|---|
| Define a small module set with wide applicability | - | ✓ | ✓ |
| Design modules to be replaceable at the 1st line | - | ✓ | ✓ |
| Maximize interoperability and interchangeability of modules | - | ✓ | ✓ |
| Adopt the use of an open-system architecture | - | - | ✓ |
| Maximize the use of commercial off-the-shelf technology | - | ✓ | ✓ |
| Maximize technology transparency for both hardware and software components | - | - | ✓ |
| Minimize the impact of hardware and OS upgrades | - | - | ✓ |
| Maximize software reuse and portability | - | ✓ | ✓ |
| Define comprehensive BIT and fault tolerance techniques to allow deferred maintenance | ✓ | ✓ | ✓ |
| Provide support for a high degree of both functional and physical integration | ✓ | - | ✓ |
| Ensure growth capability with reduced re-certification | ✓ | - | ✓ |

Table 1: Architecture Principles Mapping to the MOSA Business Drivers

[26] STANAG 4626 Modular and Open Avionics Architectures Part 1, 2004 Edition 2 is CLASSIFIED and was updated in 2022.

| Item No | Doc. Type | Short Title | Lang | Long Title / Link To Document (if available) | Promulgation Date |
|---|---|---|---|---|---|
| 1 | Cov + Std | STANAG 1059 Ed: 8 | EN FR | LETTER CODES FOR GEOGRAPHICAL ENTITIES CODES LETTRES DES ENTITES GÉOGRAPHIQUES | 19-02-2004 |



# 6. 1ST PARADOX OF MOSA | REALIZED BY PRUDENT SYSTEM DECOMPOSITION

MOSA is the ironic system characteristic; typically, system characteristics are applicable to the whole system; however, the MOSA quality characteristic is to create severable-integrative modules that compose the whole. Intended and anticipated deployment contexts shape Families of Systems (FoS), Product Lines (PLe), and MOSA through their subordinate and superordinate key quality attributes.

The architect views these deployment contexts as market and or mission segments. The subordinate quality attributes (characteristics) gleaned from this segmentation drive the superordinate modularity characteristic.

Both DoD and NATO intimately relate acquisition success to the capability to create, govern, and manage increasingly modular systems. Such modular systems contribute to the capacity to innovate and the scalability of acquisition in terms of both containing architectural diversity and affecting affordability at scale.

## Introduction of the FoS Evolution Concept

### Relationship of FoS Evolution and Product Line Evolution

- **Initiate FoS:**
  Provides Foundational Information to Initiate the FoS
- **Conduct FoS Analysis:**
  Provides Analysis of the "as is" FoS and Basis for its Evolution
- **Develop FoS Architecture:**
  Develops/Evolves the Persistent Technical Framework for FoS evolution and a Migration Plan Identifying Risks and Mitigations
- **Plan FoS Update:**
  Evaluate FoS Priorities, backlog of FoS Change, and Options to Define Plans for the Next FoS Upgrade Cycle
- **Implement FoS Update:**
  Oversees System Implementations and Plans/Conducts FoS-Level Testing, Resulting in a New FoS Baseline
- **Continue FoS Analysis:**
  Ongoing FoS Analysis Revisits the State of and Plans for the FoS as the Basis for FoS Evolution

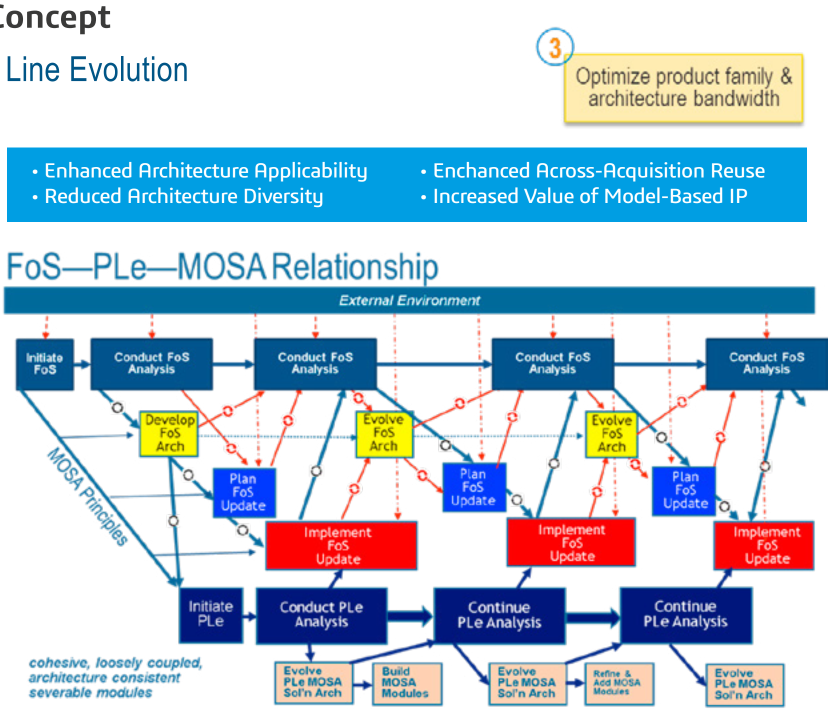



Figure 13: Illustration of Key Family of System (FoS), Product Line Engineering (PLe), and MOSA Processes and Relationships in the Context of Product Family Architecture

## MOSA Reduced Architecture Diversity

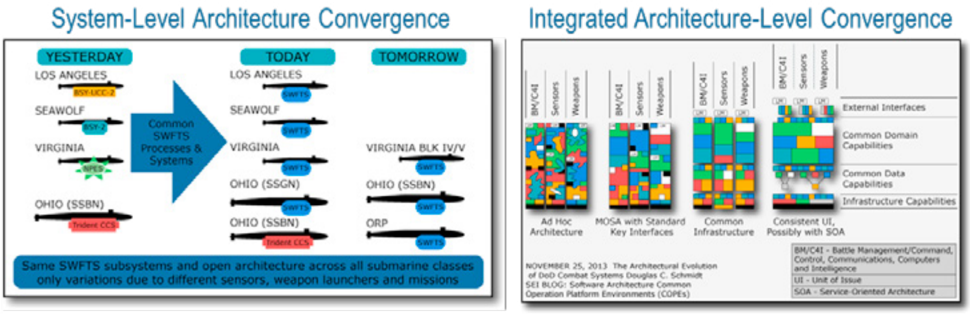



Figure 14: Illustration of MOSA effect on Architecture Diversity and Modularity

Family of Systems, Product Lines, MOSA: MOSA is a Platform Family enabler. There is a direct tie between the concepts applied based on Market Segmentation and those applied based on Mission Segmentation. The Platform-Level Architecture intent is enabled by the Integrated Architecture intent as reflected above. The "How" of the right side, can not trade off the "What" of the left side."

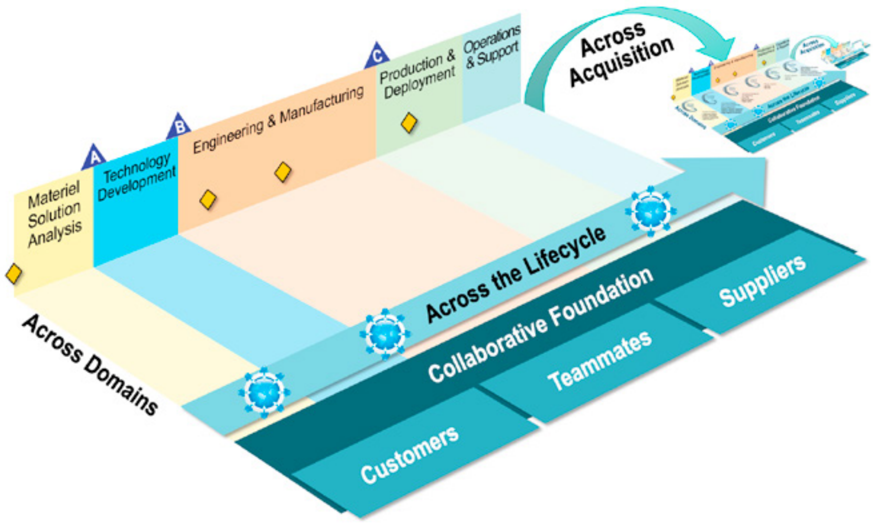
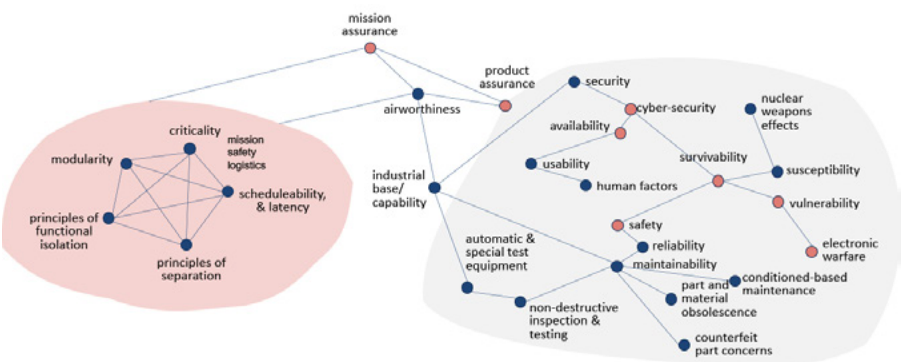
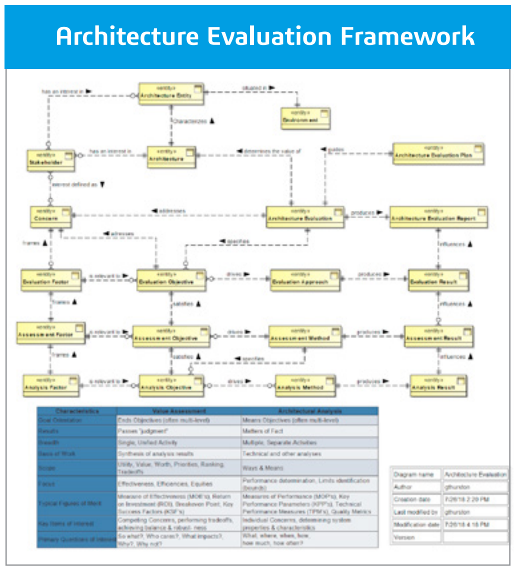



Figure 15: Representing the MOSA Value Network Implications Cross-Lifecycle and Cross-Acquisition.

## Mission and System Architecture Evaluation Characteristics




**Superordinate and Subordinate System Characteristics Or Quality Attributes**




**ISO-42020/42030**

Figure 16: Specialty Engineering Validation and Shaping of MOSA and its Integral Part in Coordinated Reuse

# 7. 2ND PARADOX OF MOSA | THE SYSTEM HAS TO BE MORE OPEN AND SECURE

MOSA is also ironic in that security is a big part of the bounded deterministic aspects of any MOSA architecture; that is to say, yes, open but secure; hence the paradox. As mentioned above, other characteristics, such as affordability, scalability, and configurability, are also desirable. Both the SoS and Systems of Interest have these concerns, as well as the systems used to engineer this system. The OMG Unified Architecture Framework (UAF) explicitly considers the Security Domain and hence affects the architecture, detailed design, and implementation.

Model-Based Assurance methods, as reflected in the various[27,28,29,30] initiatives at NASA[31,32], FAA[33,34,35], and DoD[36], are being targeted to improve the rigor and trust of complex engineered systems that can overwhelm conventional methods. It is essential here that we tie in the important concept of latency. When we go back to the factors in acquisition success and consider the rate of capability evolution, affordability, and overmatch, it cannot be underestimated how the latency of traditional manual assurance processes could deleteriously affect all these dimensions. It is not just a technical imperative; it is a moral imperative to develop such assurance methods.

It is impossible to cover the breadth of Cybersecurity in this paper. We added this section to elaborate on the MOSA-Security ironic nature; Figure 20 enumerates the key cybersecurity architecture principles that frame the architecture concerns in Figure 19.

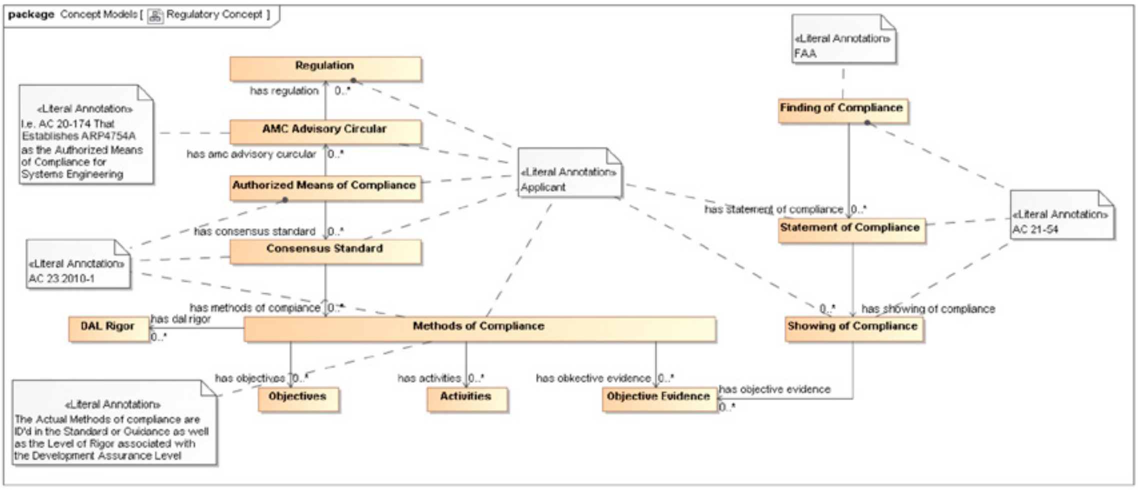



Figure 17: Regulatory Contexts Concepts for Model-Based Assurance; AMC - Authorized Means of Compliance; DAL - Development Assurance Level

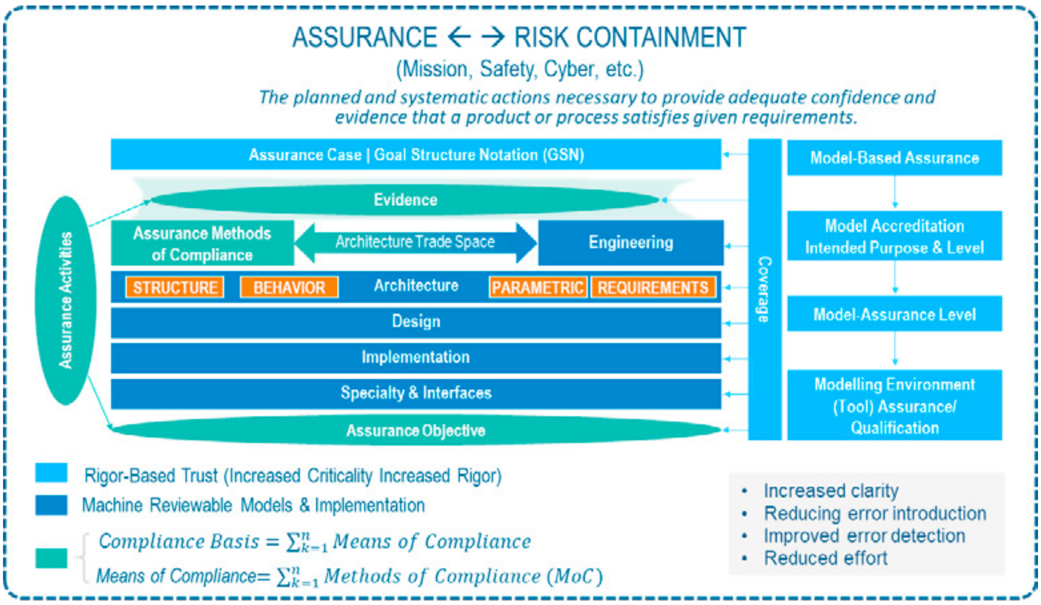



Figure 18: Model-Based Assurance Reflecting Concepts Fused from FAA, NASA, and DoD.

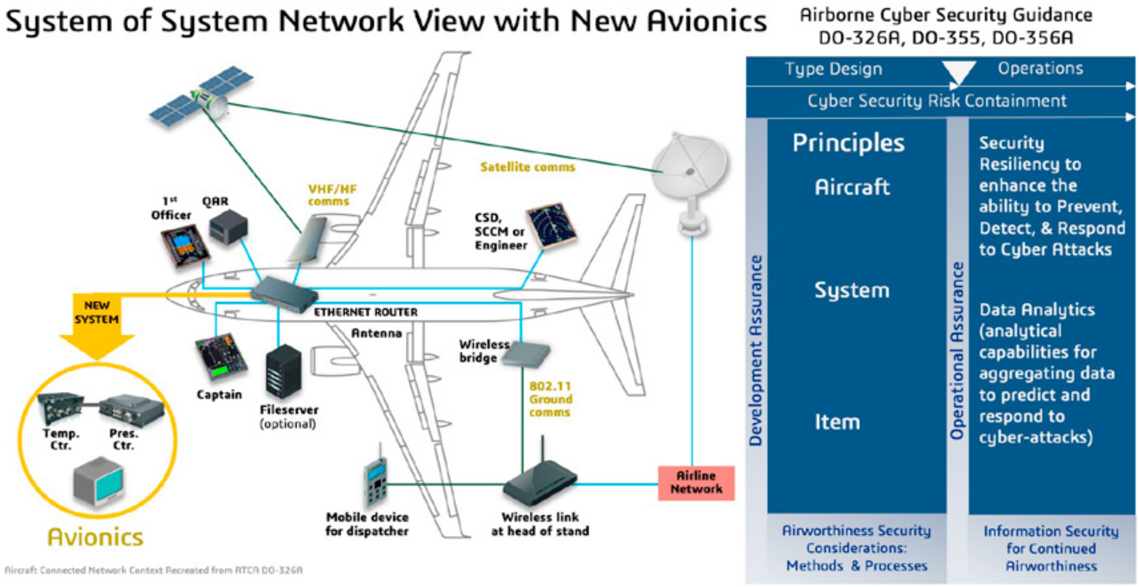



Figure 19: System of Systems Aircraft Cyber Security Context and System Engineering and Operational Cybersecurity Guidance.

## Cyber Principles

**Security Architecture Principles at Aircraft Level**

Principle 1 - Defense-in-Depth
Principle 2 - Integrity of Connected Equipment
Principle 3 - Continued Airworthiness
Principle 4 - Praevent Bypass of Security
Principle 5 - Keep Security Architectures as Simple as Possible
Principle 6 - Detection and Restoration

**Security Architecture Principles at (Multi-) System Level**

Principle 7 - Attack Path Refinement at System Level
Principle 8 - Consider Security Process Specifics
Principle 9 - Minimize External Interfaces
Principle 10 - Disable All Unused Interfaces
Principle 11 - Independence and isolation

**Security Architecture at Item Level**

Principle 12 - Ensure Proper Error Handling
Principle 13 - Least Privilege
Principle 14 - Control Access to Connections

Figure 20: Cybersecurity Principles from DO-326A, and the Compliance Assurance enablement of the **3D**EXPERIENCE platform in accordance with the elements

[27] Julie S. Fant and Robert G. Pettit, Model Assurance Levels (MALs) for Managing Model-based Engineering (MBE) Development Efforts, The Aerospace Corporation, Chantilly, Virginia, U.S.A., 2019.
[28] Goal Structured Notation (GSN) Standard, Version 3, 2021.
[29] OMG Risk Analysis and Assessment Modeling Language (RAAML) Libraries and Profiles, Version 1-Beta, 2021.
[30] MITRE Architecture Quality Assessment Version 2.0 R. Hilliard M. Kurland S. Litvintchouk T. Rice S. Schwarm August 7, 1996.
[31] NASA/CR-20210017388, Architectural Modeling and Analysis for Safety Engineering Danielle Stewart, et. al., 2021.
[32] John Evans, Steven Cornford, Martin S. Feather, Model Based Mission Assurance: NASA's Assurance Future, ~2016.
[33] RTCA DO-330 Tool Qualification; DO-333 Formal Methods; DO-326A Cyber Security.
[34] FAA AC 21-51 Applicant's Showing of Compliance and Certifying Statement of Compliance, 2011.
[35] IFAA AC 23.2010-1 : FAA Accepted Means of Compliance Process for 14 CFR Part 23, 2017.
[36] DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A), 1996.

# 8. MISSION LIFECYCLE CONCEPTS SHAPE MODULARITY TRADES

The applications used to gain insight into the Modular Open Systems provide a variety of means for the architects and stakeholders to evaluate the organizational, operational, and specialty engineering characteristics associated with the known and anticipated deployment contexts. It is crucial for the stakeholders to appreciate that these deployment contexts include all operational lifecycle concepts for explicit consideration in the mission thread analyses.[37]

Figure 21 illustrates several publically available Mission Contexts for Future Long Range Assault Aircraft. As we start to consider the problem space of these various missions, we can look at the mission segmentation as we typically do when we are looking at Market Segmentation and start to leverage these concepts to look at what is common across these mission aspects and what is variable. As we start to look at the solution space, we may look at the Family of Systems, the Product Family, and product line aspects in order to gain some understanding of the right variability. Incorporating variability into the architecture that uniquely contributes to the overall value of the architecture applicability is essential to meet the broadest possible scope in the interest of capitalizing on lifecycle investments in a deliberate and coordinated way.

[37] Dassault Systèmes' Industry Solution Experience Framework (ISEF), "Needs Analysis Supplement | Mission Thread Analysis", 2020.

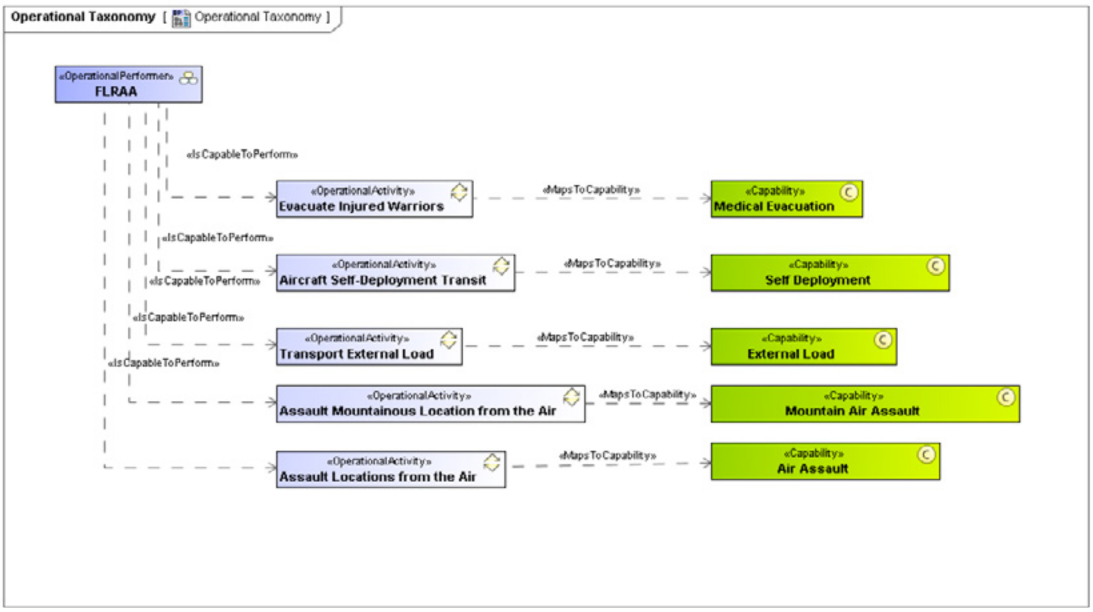



Figure 21: Example Mission Portfolio Drawn From US Future Vertical Lift Program

# 9. MOSA ARCHITECTURE SYNTHESIS AND EVALUATION

One means by which architects can envision system modularity is using the design structure matrix (DSM).[38] Using model descriptions of the systems, it is also possible to apply advanced mathematical methods such as clustering, mutating genetic algorithms, and simulated annealing to help to evolve the system's modularity.[39,40,41]

## MOSA Evaluation is Afforded at Each Architecture Level

Design Structure Matrix: DSM provides a means by which to visualize, evaluate, evolve each layer in the architecture: Mission Architecture, System Architecture, Component Architecture & Design, and Implementation that provides key enablement. Enduring fleet architecture constraints need to be accommodated by MOSA module architecture.

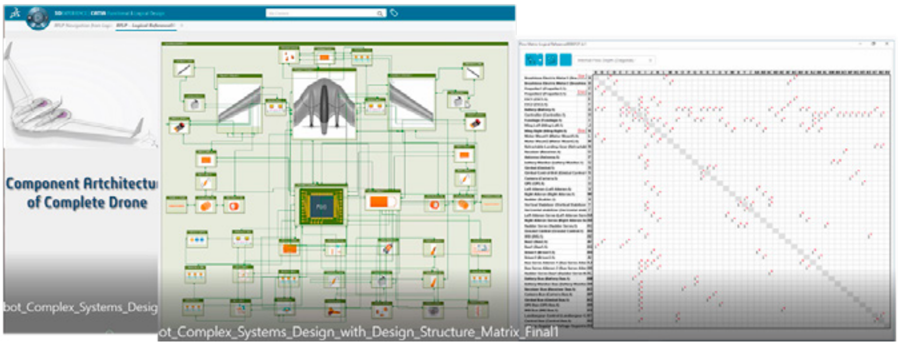




Figure 22: Illustration of **3D**EXPERIENCE platform Modularity Analysis and Evolution

For example, the MOSA key quality attributes such as security, safety, severability, configurability, schedulability, commonality, variability, reusability, survivability[42], etc., the architect organizes into a standard Architecture Tradeoff Analysis Method (ATAM) Matrix.[43]

The architect applies these methods against the synthesized architecture alternatives can be evaluated in accordance with ISO-42020/42030 against these quality attributes in accordance with the Mission Thread analysis to evaluate the identified mission deployment contexts and their quality attributes (key system characteristics).

| # | Name | totalDX : Real | totalV280 : Real |
|---|---|---|---|
| 1 | flraa | 315 | 285 |
| 2 | flraa.aircraft Self-Deployment Transit | 60 | 50 |
| 3 | flraa.aircraft Self-Deployment Transit.availability | 5 | 10 |
| 4 | flraa.aircraft Self-Deployment Transit.configurability | 0 | 5 |
| 5 | flraa.aircraft Self-Deployment Transit.dependability | 10 | 5 |
| 6 | flraa.aircraft Self-Deployment Transit.maintainability | 5 | 5 |
| 7 | flraa.aircraft Self-Deployment Transit.modularity-severability | 5 | 10 |
| 8 | flraa.aircraft Self-Deployment Transit.reliability | 10 | 10 |
| 9 | flraa.aircraft Self-Deployment Transit.reusability | 5 | 5 |
| 10 | flraa.aircraft Self-Deployment Transit.safety | 5 | 0 |
| 11 | flraa.aircraft Self-Deployment Transit.security | 10 | 0 |
| 12 | flraa.aircraft Self-Deployment Transit.survivability | 5 | 0 |
| 13 | flraa.assault Locations from the Air | 65 | 70 |
| 24 | flraa.assault Mountainous Location from the Air | 70 | 65 |
| 35 | flraa.evaluate Injured Workers | 65 | 55 |
| 46 | flraa.transport External Load | 55 | 45 |

Figure 23: ATAM x-Mission Assessment Matrix Data, Completely Fictionalized Data, Just Illustrating the Method in the Cameo Application

[38] Browning, Tyson, "Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities" IEEE Transactions, 2015.

[39] Thebeau, Ronnie, Knowledge Management of System Interfaces and Interactions for Product Development Processes, MIT MS Thesis, 2000.

[40] Dassault Systèmes' Collaborative work: Shatad Purohit, Berenger Winckler, Ganesh Chavan and Bhatu Patil Dependency Structure Matrix (interactions matrix "N2") with Genetic Algorithm (DSM with GA) project on AI initiative named Co-bot for 3DEXPERIENCE platform collaborated with a team working on 'Advanced Mathematics Library' to create APIs.

[41] Purohit, Shatad, Azad Madni, A Model-Based Systems Architecting and Integration Approach Using Interlevel, and Intralevel Dependency Matrix, IEEE Systems Journal, 2021. Want to consider as well SAE Aerospace Information Report AIR6988 Artificial Intelligence in Aeronautical Systems: Statement of Concerns, 2021.

[42] Carelson, Terance, Using MOSA to Address System Survivability in Army Weapons Systems, Defense Acquisition University, 2019.

[43] Dassault Systèmes' Industry Solution Experience Framework (ISEF), "Validation | Architecture Evaluation", 2022.

# 10. MOSA VALUE NETWORK ASOT GOVERNANCE AND MANAGEMENT

The Value Network and or Enterprise[44] use the **3D**EXPERIENCE platform to govern and manage the MOSA system throughout its lifecycle. In addition, there are several tools and applications that Dassault Systèmes specifically architected to be able to conduct key analyses that prospectively shape the super architecture and module architectures, design, and implementation. As with any system, we necessarily manage MOSA throughout the lifecycle.[45]

## Improving Acquisition Scale

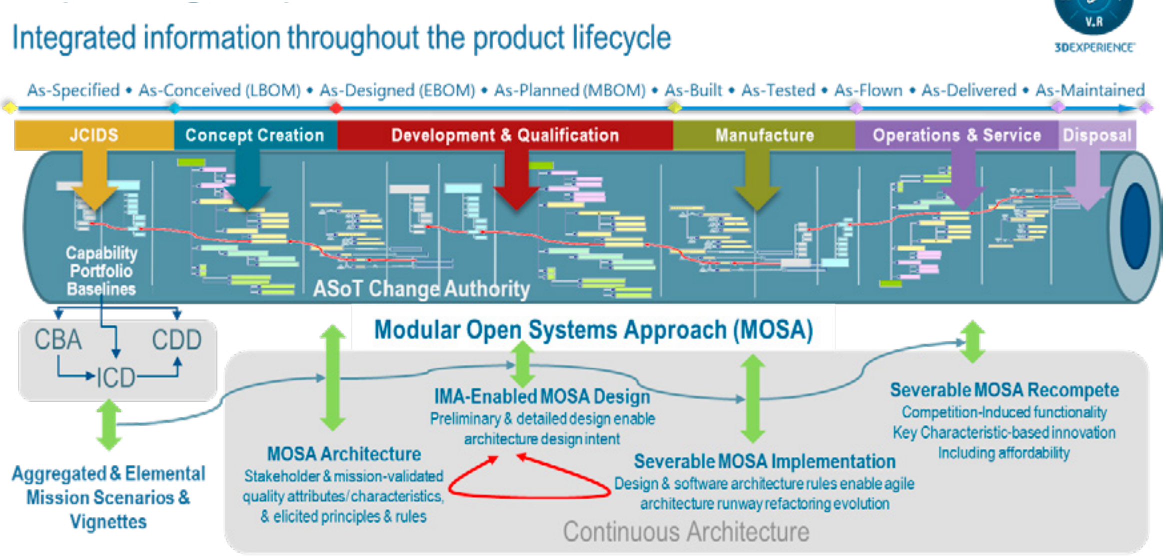



Figure 24: Acquisition Lifecycle Implications Establishing Cross-Acquisition MOSA Exploitation Framing

## Lifecycle Considerations: Specialty Engineering Objectives and Concerns

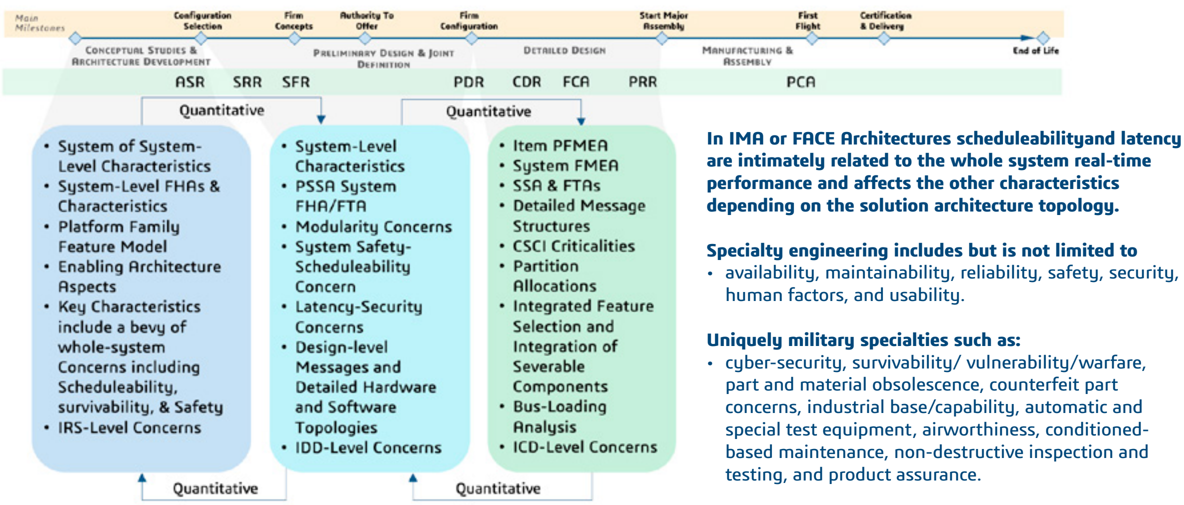



Figure 25: Lifecycle Specialty Engineering Incremental Architecture, Design, & Implementation Validation Implications

[44] Thurston, Garrett, Brian Christensen, Model Based Enterprises, Dassault Systèmes Whitepaper, 2015.
[45] Modular and Open Systems Approach (MOSA) Considerations Throughout the Systems Engineering Lifecycle, MITRE Technical Report Number 180267, 2018.

The MOSA Governance and modular component management are crucial in order that the reuse is deliberate. ARP4754 [5.3.1.5] for Systems and DO-297 for IMA not only prescribes the asset reuse deliberate process but includes tasks of incremental recertification/qualification in the integrated deployed context. SEBOK[46], ARP4754A, EIA-632, and other guidance materials and standards prescribe architecture, requirements, and assumptions validation and management thereof.[47] As such, the ASoT[48] needs to facilitate the integral management of such validation artifacts, including but not limited to AoA[49], Trade Studies (SoS, System Architecture, Item Design, Implementation), and Specialty Engineering artifacts.[50]

MOSA context makes the accessibility and disciplined management of validation evidence even more crucial to its success. Validation artifacts are integral to the finding of compliance, statement of compliance, showing of compliance, and means and methods of compliance. In addition, in the context of MOSA, the first step to reuse is to revalidate the MOSA component in its new intended deployment context. Failing to have an integrated ASoT for this purpose will produce business outcomes for MOSA that fall far short of the value network ecosystem aspirations.[51]

# Improved Operational Discipline

## ASoT Architecture Implementation including Change

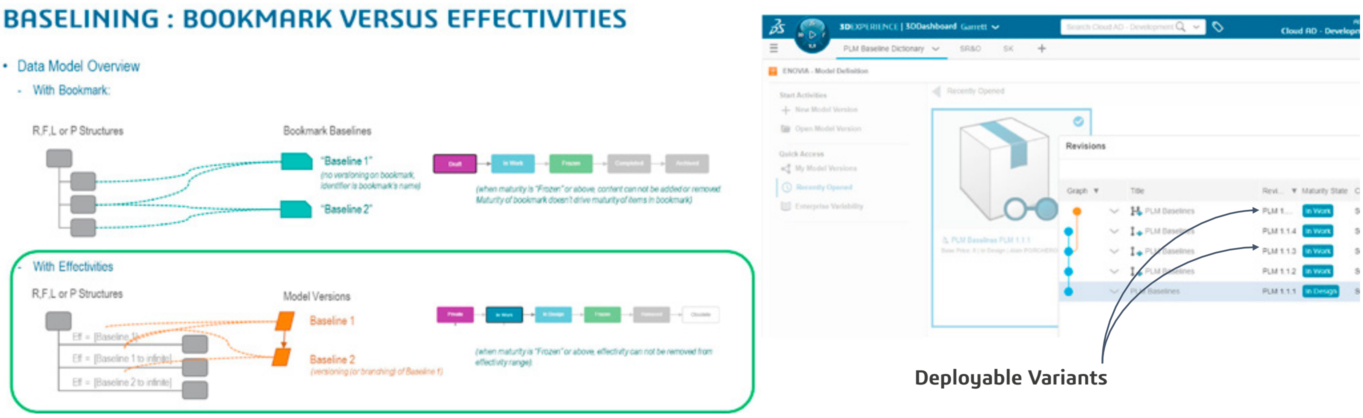



Figure 26: Illustration of Disciplined Configuration and Change Management **3D**EXPERIENCE platform Operation; Part 1 Are the Plans and Procedures.

In addition, to the management applications afforded by the platform, the STIMULUS application is crucial to establishing complete, correct, and consistent requirements.

The program/product team uses the same lifecycle management pattern for verification methods of compliance as for the validation, though as provided for in the aforementioned guidance, upon reuse, the verification methods can be different. For example, what may have been a verification test on initial use may be verification analysis in the reuse context. This is, in fact, why we specified the Classification Manager Application in Figure 26 since it affords to organize MOSA assets into Module Library classes to afford improved asset awareness and reuse exploitation.

[46] SEBOK.
[47] Chen, L., Muhamed Ali Barbar, Bashar Nusebeh, Characterizing Architecturally Significant Requirements, IEEE Software, 2013.
[48] An authoritative source of truth has an authority, is configuration managed against all the principles of EIA-649C and or IEEE-828 Requirements, is trusted, and is current.
[49] USAF Materiel Command (AFMC) Analysis of Alternatives (AoA) Handbook: A Practical Guide to AoA, 2008.
[50] The matrix should exist for each System Segment, System, and Configuration Item (CI). Since requirements start to be defined in the conceptual design phase, there has to be validation and verification statements/elements defined (MATTERS and AIDT) as a requirement is drafted. Therefore as the modeling and associated requirements are developed for an element of the Functional, Logical, and Physical PBS (Product Breakdown Structure) the validation matrix is initiated. Therefore the validation matrix is (1) for Systems and System Segments – Conceptual Design. (2) for CIs – Conceptual and Preliminary Design Phases (depending upon how low you are in the PBS). In short, once you have written your MOSA Validation Plan you should start creating the Validation Matrix and the disciplined management thereof that is implied.
[51] DODI 5000.88 explicitly 3.a.(1) "... The modular and open systems approach will be documented in the digital authoritative source of truth." Refined in MIL-HDBK-539 Digital Engineering and Modeling Practices, 2022.

# 11. MOSA MATURITY ROADMAP

Finally, MOSA doesn't just happen because policy dictates it. Governance should provide for Ecosystem Value Network, Enterprise[52], Business Unit, Program, and Organizational MOSA Maturity Levels and Assessment thereof. DoD has a crude MOSA evaluation tool[53], and the defense industry has also provided MOSA adoption guidance.[54] By analyzing these artifacts and others, it is possible to program the MOSA goals and structured objectives that are incremental, time-bounded, business operating model targeted, quantitative, and interdependent aspects into an enterprise phase roadmap also using the tools and applications of the **3D**EXPERIENCE platform including roadmap monitoring and control dashboards.

[52] Thurston, Garrett, Transforming the Acquisition Value Network, Dassault Systèmes Whitepaper, 2022.
[53] MOSA Program Assessment Rating Tool (PART)
[54] NDIA Modular Open Systems Approach: Considerations Impacting Both Acquirer and Supplier Adoption, 2020.

# 12. DASSAULT SYSTÈMES MOSA ENABLING INFRASTRUCTURE, APPLICATIONS AND TOOLS

| Role | Operational Scenario | Brief Application Description | x-Reference |
|---|---|---|---|
| TUR-OC (Systems Solution Architect) | Define system architecture and federate detail design by discipline. | Allows modeling of requirements and system functional and logical characteristics. | Section 10 |
| SAX-OC (Logical Product Architect) | Establish a configured product architecture as a referential for product definition. | Enables the creation of a reference (logical) architecture and management of variants. | Section 3 |
| TRM-OC (Requirements Engineer) | Manage requirements (to include MOSA requirements) and allocate to system functions and logical components. | Provides for requirements management and traceability to the architecture and configurations that satisfy requirements. | Section 3 |
| TRY-OC (Systems Traceability Analyst) | Trace MOSA requirements to physical, logical, and functional architecture elements and V&V artifacts. | Allows the creation of traceability links between data objects (3D models, functional and logical architecture, requirements, et al.). | Section 10 |
| CHG-OC (Change Specialist) | Coordinate design changes and manage the impact on system architecture. | Offers a comprehensive, automated change process with traceability to affected items. | Section 10 |
| CFG-OC (Configuration Engineer) | Design for modularity with system variants, each developed to unique mission and technical requirements. | Enables definition of system variants in a single configured product structure. | Section 10 |
| PDM-OC (Product Manager) | Define MOSA strategy and plan system evolutions (model revisions) to meet mission and stakeholder needs. | Provides for model version lifecycle management from the portfolio perspective. | Section 10 |
| VRP-OC (Test Manager) (CA) | Define and oversee V&V strategy, and monitor test and simulation execution. | Delivers a dashboard to organize virtual simulations and physical tests and to plan and monitor their execution. | Section 3 |

| Role | Operational Scenario | Brief Application Description | x-Reference |
|---|---|---|---|
| EXH-OC (Exchange Manager) | Efficient collaboration sharing product engineering data across the value network. | Navigate the configured 3D product definition, including search and volumetric query, import, & export. | Section 10 |
| XEN-OC (Product Release Engineer) | Manage the product structure to include modular elements and their development maturity, revision, and release. | Permits management of the product definition and structure, simplifying product governance and ensuring adherence to standards through defined part numbers. | Section 10 |
| PAU-OC (3D Product Architect) | Develop 3D models and create multi-CAD versions for MOSA. | Provides the ability to explore and author multi-CAD product structures. | Section 10 |
| XSF-OC (Connected Software) | Integrate functional and logical architecture with source code. | Delivers the capability to manage software source code and build artifacts, and link to architecture. | Section 10 |
| DPM-OC (Project Manager) | Manage system and modular development by project and task. | Enables project management with system development data linked to a project for real-time status. | Section 3 |
| XPP-OC (Project Planner) | Simple and assisted, team-based iterative planning, execution and monitoring. | Define project scope, dependencies and milestones. An innovative engine automatically schedules and optimizes projects activities to meet key milestones | Section 10 |
| PPL-OC (Process Engineer) | Define manufacturing operations for the system and its modules. | Permits development of manufacturing process plans with a 3D graphical view of processes and sequences. | Section 10 |
| MFN-OC (Manufacturing Items Engineer ) | Create the Manufacturing Bill of Material. | Allows definition of the MBOM based on 3D design data. | Section 10 |
| MGA-OC (Manufacturing Process Engineer) | Assign parts and assemblies from the MBOM to specific operations. | Streamlines the work plan, allowing allocation of resources, tools, parts and assemblies to operations and updating with any changes from design data. | Section 10 |
| CCM-OC (Classification Manager) | Manage IP throughout system development. | Provides data classification schemes and libraries with classification attributes, allowing different functional groups to organize content simultaneously from different perspectives. | Section 10 |

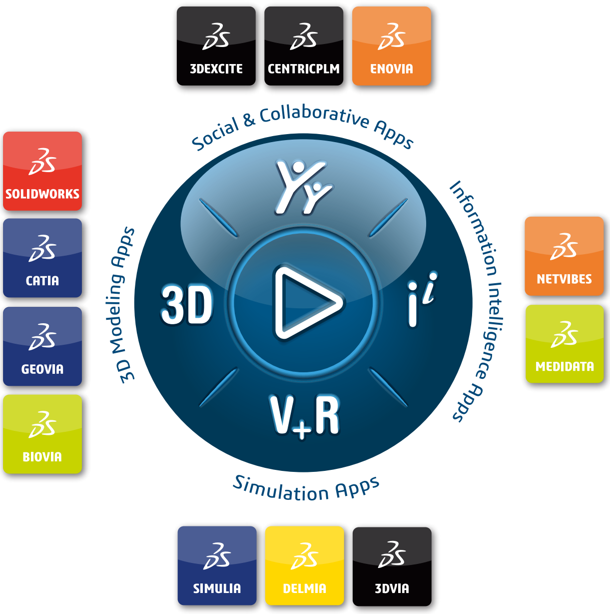
| Role | Operational Scenario | Brief Application Description | x-Reference |
|---|---|---|---|
| PUS-OC (Project Intelligence Consumer) | Analytics application for performing monitoring, diagnostic and trend analyses on past and ongoing projects in the interest of control. | Predictive analytics capabilities to all project stakeholders. | Section 10 |
| CUS-OC (Change Intelligence Consumer) | 360° view of changes for efficient program management by performing diagnostic, impact and trend analyses on past and ongoing changes. | Analytics application for performing diagnostic and trend analyses on past and ongoing changes and issues. | Section 10 |
| DAA-OC (Data Analyst) (CA) | Explore and analyze enterprise data to create Data Perspectives based on business needs. | Explore your enterprise data to reveal and discover meaningful information using a powerful query system. | Section 10 |
| DAE-OC (Data Engineer) (CA) | Collect and prepare data to provide ready-to-use datasets. | Collects raw data and configures data ingestion from multiple sources into the 3DEXPERIENCE Semantic Lake. Establish pipelines to automate the different stages of data acquisition, from extraction to storage. Orchestrates data processing, including data normalization and data cleansing to provide datasets to Data Analysts. | Section 10 |
| DAS-OC (Data Steward) (CA) | Organize knowledge and ensure governance of your data. | Define, develop and validate your ontologies through dedicated user interfaces for authoring classes and properties. | Section 10 |
| AEPA-OC (Data Viewer) (CA) | Reveal information intelligence from data to identify levers to enhance your business | Navigate into shared Data Perspectives to read KPIs and gain insights into your business. | Section 10 |
| BED-OC (Business Experience Designer) (CA) | | | Section 10 |
| BPW-OC (Business Process Analyst) (CA) | Analyze and monitor the execution of business processes to identify bottlenecks and take appropriate actions. | Audit, track and monitor end-to-end business process execution in real-time to identify process bottlenecks, optimization opportunities and potential compliance issues | Section 10 |

| Role | Operational Scenario | Brief Application Description | x-Reference |
|---|---|---|---|
| BPR-OC (Business Process Designer) (CA) | Design, configure, simulate and deploy end-to-end enterprise business processes. | Effectively modeling business processes using the BPMN 2.0 standard.  Define your business processes in the 3DEXPERIENCE Platform to increase organizational efficiency and standardization. You can configure forms, automation activities and KPI's to increase organizational efficiency, compliance and standardization. | Section 10 |
| BPO-OC (Business Process Player) (CA) | Instantiate new processes or execute tasks, which have been assigned to you through a business process | Enables assigned business process task execution. | Section 10 |
| Multidisciplinary Optimization Engineer | Explore variations of design parameters to identify the optimal design that satisfies requirements. | Provides industry-standard algorithms for the design of experiments, parametric optimization, surrogate model generations, and uncertainty quantification. | Section 3 |
| Spatial Allocation Reservation (CATSPL_AP) | Ensure the aircraft industrial spaces definition with the right dimensions to provide the right zones partitioning to the engineering and manufacturing activities | Large projects, such as aircrafts, it is often necessary to divide the working space into zones. | Section 3 |
| System Safety | Perform FMEA and FTA. | Enables failure mode definition and analysis of effects in the system architecture and association of failure modes to logical and functional references to perform fault tree analyses. | Section 3 |
| Electrical & Electronic Application (EEW) | Functional to Topological architecture for protocol-dependent integrated modular architectures (IMA) | Affords, analysis of such integrated architectures against the desired characteristics of the integrated system i.e. bus load | Section 3 |
| EEW - AADL Code Generator | Generates AADL code from the IMA architecture | Affords, schedulability analysis provided for dynamic priorities assuming constrained deadlines and based on the demand bound function | Section 3 |
| MSoSA - Magic Systems of Systems Architect (MYH) | SoS & System Architecture Commonly referred to as Cameo Enterprise Architect | Architecture Description, Rationale, evaluation, trades, and analysis. | Section 9 |

| Role | Operational Scenario | Brief Application Description | x-Reference |
|---|---|---|---|
| Introp | Workflow | Workflow specification and accountability. | Section 7 |
| STIMULUS | Simulate requirements for validation. | Allows simulation of requirements in early design to identify ambiguous or conflicting requirements, and in a V&V phase, enables test engineers to generate objectives automatically and verify embedded code compliance. | Section 10 |
| Magic Cyber Systems Engineer (MAH-N) | CATIA Magic Portfolio is a commercial portfolio that contains a small number of products packaged, named, and priced specifically for qualifying academic institutions. | Systems architecture | Section 3 |
| Magic Model Analyst (MSI-N) | Model Analyst | Model execution | Section 3 |

**Dr. Thurston** is a Sr. Director Enterprise Transformation with 37 years of Aerospace & Defense and Security experience. He is a futurist and has affected strategic direction including the initiation of what has culminated in the acquisition of No Magic.







DASSAULT SYSTEMES | The **3D**EXPERIENCE® Company

**Europe/Middle East/Africa**
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

**Asia-Pacific**
Dassault Systèmes K.K.
ThinkPark Tower
2-1-1 Osaki, Shinagawa-ku,
Tokyo 141-6020
Japan

**Americas**
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA