MARCH 2025

PRECISE SYSTEMS

SCALE YOUR **FUTURE**

A CLOSER LOOK AT **AWESUM®**

INCLUDING MODULES, PRODUCTIVITY AND PROGRAM BENEFITS, KEY TECHNICAL ENABLERS, AND MORE

EXECUTIVE SUMMARY

AWESUM® is an end-to-end digital engineering tool suite that will optimize your Program's systems development and integration process to reduce effort, schedule and cost while increasing quality and supporting high integrity safety and information assurance systems, large and geographically distributed teams, and emerging technologies and processes including the DOD Modular Open System Approach (MOSA). This means that the Program decision makers can benefit from an enhanced Integrated Development Environment (IDE) that gives them the data upon which to make decisions and enhances the complete systems engineering process so their Government team and industry support can be focused on delivering mission capability.

OVERVIEW

Precise Systems' AWESUM® is a comprehensive digital engineering tool suite that has been designed to optimize the systems development and integration process, greatly reducing effort, cost, schedule, and risk while enhancing quality across an engineering program.

At its core, AWESUM® utilizes System Unified Model (SUM), a system and software framework that employs formal methods to represent and store all aspects of the system lifecycle. This enables seamless end-to-end lifecycle management, covering requirements capture, system architecture, component and data representation, design, code generation, hardware and software validation, system integration and test, enabling rapid modernization and integration. The tool suite ensures that key documentation artifacts can be auto generated by maintaining structured data sets and managing the entire engineering process. AWESUM® benefits from the judicious application of Artificial Intelligence (AI) and Machine Learning (ML) technologies to support requirements model definition, code generation, test generation, and integration, while utilizing the SUM to ensure high integrity deterministic products.

AWESUM® is designed to support high-integrity engineering and defense-specific standards, including DO-331, DO-178C, DO-254, Future Airborne Capability Environment (FACE™), Hardware Open Standards Technologies (HOST), Sensor Open Systems Approach (SOSA), Weapon Open Systems Approach (WOSA), and MOSA practices. It streamlines information management and automates document generation ensuring program managers and decision makers maintain accurate status of the program and allows engineering teams to allocate more time to technical design and innovation.

To support the government's stated objective of model-based system engineering and open systems architecture, the AWESUM® suite of tools deliver significant savings through a Model-based MOSA (MMOSA™) approach. Our tools and processes significantly reduce the early-stage impact of open system approaches on the development timeline, easing product development, testing compliance and providing the basis for future products and system improvements.

The AWESUM® product line is composed of a modular suite of integrated model-based tools that align with the lifecycle Agile and V models, and components described by RTCA's DO 331 Model-Based Development and Verification supplement to DO-178C and D-278A. AWESUM® has been designed to provide a robust environment where a program can manage the development and delivery of high-integrity, high-capability, flexible, interoperable, and scalable systems, enabling both new and legacy programs to benefit from MOSA practices, and greatly enhanced integration activities through data driven integration methods.

The architecture of AWESUM® enables your Program to manage the system lifecycle data across all related Government, OEM and supplier entities. It will provide compatibility with the various industry IDEs from which it can import model data and documentation, providing a standardized single source of truth for the management and delivery of a program.

AWESUM® MODULES

Unlike other model-based market toolsets, AWESUM® unifies program lifecycle activities. Specifically, the Systems and Software Design phase, the leading edge of the V Model, is handled by the Develop™ module. The FAME™ module manages the modeling and verification of system data. The Verify™ module handles the Hardware and Software Test and Verification phase, the bottom side, and Verification phase, the following edge of the V Model. The AWESUM® model-based tool suite is especially designed to support the lifecycle activities required for high-integrity qualification efforts with our Simulate™ and Qualify™ modules. Finally, the Harmony™ module enables verification of hardware standard implementations, currently targeted to HOST, SOSA, and C5ISR/EW Modular Open Suite of Standards (CMOSS) systems.

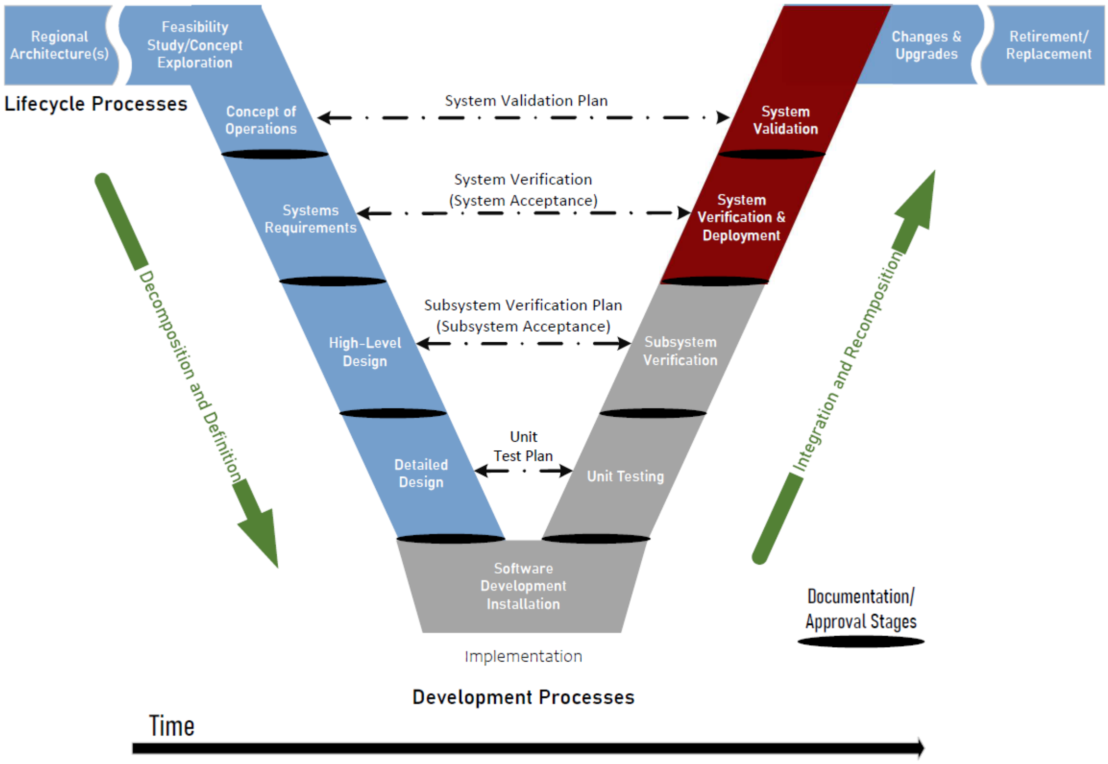

Figure 1: Systems Engineering V-Diagram

DEVELOP™

Develop™ provides a high-integrity software development environment that accelerates development and ensures traceability of high-level systems requirements through the lower-level requirements of software development phases. Develop™ imports, models, and automatically structures development projects from target device-level Interface Control Documents (ICDs). It has been specifically designed to import models from various modeling tools thus making it modeling tool agnostic, and more easily integrated into a program's IDE. System Engineers focus on core development tasks while Develop™ automatically generates device control code for common and modeled functions. All code in Develop™ directly reflects design requirements, establishing complete documentation traceability. Develop™ is used to create and manage the specification and design models (c.f., DO-331) throughout these important requirements and development lifecycle phases, i.e., the Product Definition or leading-edge side of the V-model.

FAME™

FAME™ provides an environment for developing Universal Domain Description Language (UDDL) data models and provides support for development to FACE™ Technical Standard Editions 2.1, 2.1.1, 3.0, and 3.1. FAME™ is a complete end-to-end development environment for creating UDDL data models and verification of system data through UDDL conformance testing. FAME™ reduces the complexities of managing large collections of model elements and streamlines the development and integration of software, interoperable system components, and conformance verification of 3rd party supplied system components. By simplifying the FACE™ development, you can reliably develop and deliver compliant systems at significantly reduced cost and schedule while assuring long term reuse of the modules.

VERIFY™

Verify™ provides a software testing and verification environment that accelerates and intensifies the robustness of Cyber-Physical Systems (CPS). Verify™ manages the verification lifecycle activities and assists in determining documentation and verification process completion, generates and executes test for device design and system functionality, and manages test results. Verify™ significantly deepens testing when compared to other industry tools, with robustness testing, and dramatically reducing testing time by automating verification activities. By maintaining bi-directional tracing links between requirements, modeled capabilities, and software code, Verify™ enables full bi-directional traceability and enables changes to the requirements documentation to generate code changes on the fly. As such, models or code artifacts are no longer “throw-away”, but represent the current state of your product line.

SIMULATE™

Simulate™ brings systems level inputs into the AWESUM® tool chain, by adding a high-fidelity virtual environment with outside stimulation from physical or simulated hardware. Simulate™ enables virtual operation of system devices and platforms within their concept of operations, with real-time feedback through line replaceable units (LRU) equipment e.g., communications. By basing all modeled factors on the target interfaces (i.e., ICD) and a platform concept of operations, Simulate™ validates and verifies intended operations of cyber-physical systems before expensive, real-world operational testing occurs. Simulate™ brings all system components together to represent an as-built system and model, and simulates operational characteristics of platforms, devices and their interactions within a representative environment. reducing schedule and cost while assuring traceability to requirements and needs statements.

QUALIFY™

Qualify™ provides an integrated capability for producing and managing high-integrity lifecycle documentation (CDRLs and DIDs) required to satisfy all high-integrity design, implementation, test, and verification evidence objectives. Qualify™ automatically generates and allows for artifact engineering of the software aspects of airborne system documentation required for qualifications efforts applicable to airworthiness regulations, AR 70-62, AC 20-115C, DO-178C/DO-331, DO-278A, and the reuse guidelines of AC 20-148. In addition, Qualify™ is designed to produce the lifecycle documentation required for FACE™ conformance and verification efforts ensuring alignment in support of open systems the life cycle benefits they deliver.

HARMONY™

Harmony™ provides a conformance test environment for standardized hardware and consists of three main components: Test Application – the application software that manages conformance verification of an Article Under Test (AUT); Test Station – the Module Test Interface (MTI); TeamShare Server – the enterprise server application supporting the sharing of all standard conformance data digital artifacts. Harmony™ provides an integrated method to ensure the conformance of hardware system elements and maintain the hardware-related data within the SUM. Use of Harmony™ in the early stages of product development means the end item will be aligned with the open standards for hardware. The link of Harmony™ and FACE™ along with the components of AWESUM® significantly reduces the investment in the next generation of systems along with legacy components and platforms.

PRODUCTIVITY AND PROGRAM BENEFITS

Programs are challenged with the use of their limited resources to deliver ever increasing capability, within ever more stringent time and financial constraints. As explained throughout this paper, AWESUM® has been designed to deliver end-to-end management of all aspects of the engineering process to reduce unnecessary overhead through methodical process and integration of program engineering and design data. This section provides more detail on some of the key aspects of AWESUM® that will deliver great efficiency to large and geographically distributed teams. AWESUM® automates and speeds the flow through the system engineering V while creating reuse of components and systems, allowing a more complex industrial base to perform work on these systems by broadening the supply chain including Joint, Interagency, Intergovernmental, and Multinational partners and allies.

AWESUM® TEAMSHARE AND COLLABORATE

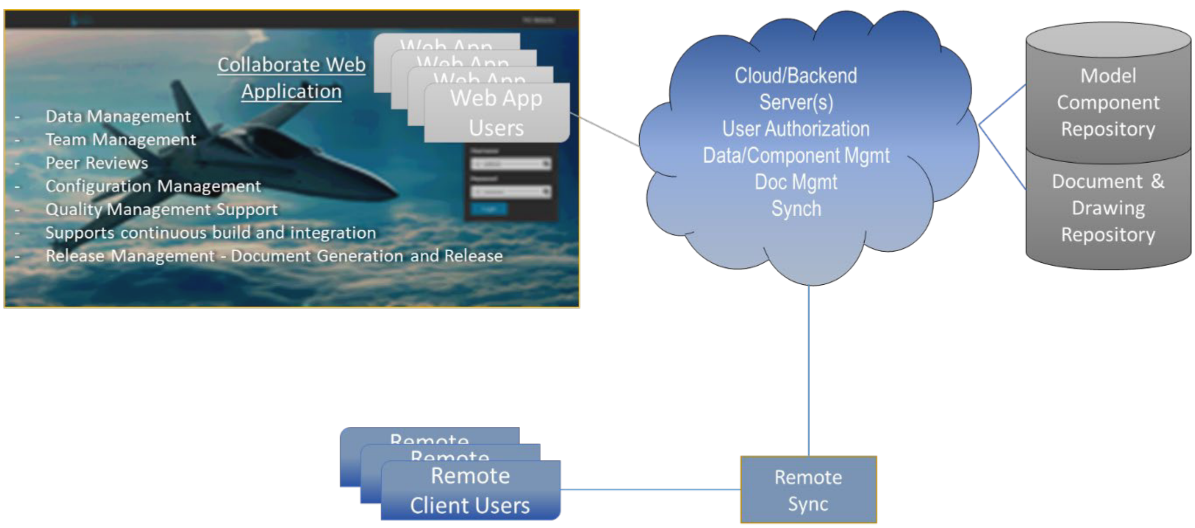

Figure 2: AWESUM® TeamShare & Collaborate Architecture

Our TeamShare and Collaborate architecture – illustrated in Figure 2 above – utilizes secure cloud and client technologies to enable large program teams to work together on a single source of truth. The ability to collaborate across geographically diverse locations, and across the multiple disciplines that are required to contribute to large programs, ensures that the system development, technical integrity, team productivity, management and engagement are maximized. The ability to manage a large and complex technical dataset, and ensure availability of dashboard information, deep technical information, and general documentation artifacts means AWESUM® is ideally placed to be the central control for all program stakeholder engagement.

DIGITAL ENVIRONMENT INTEGRATION

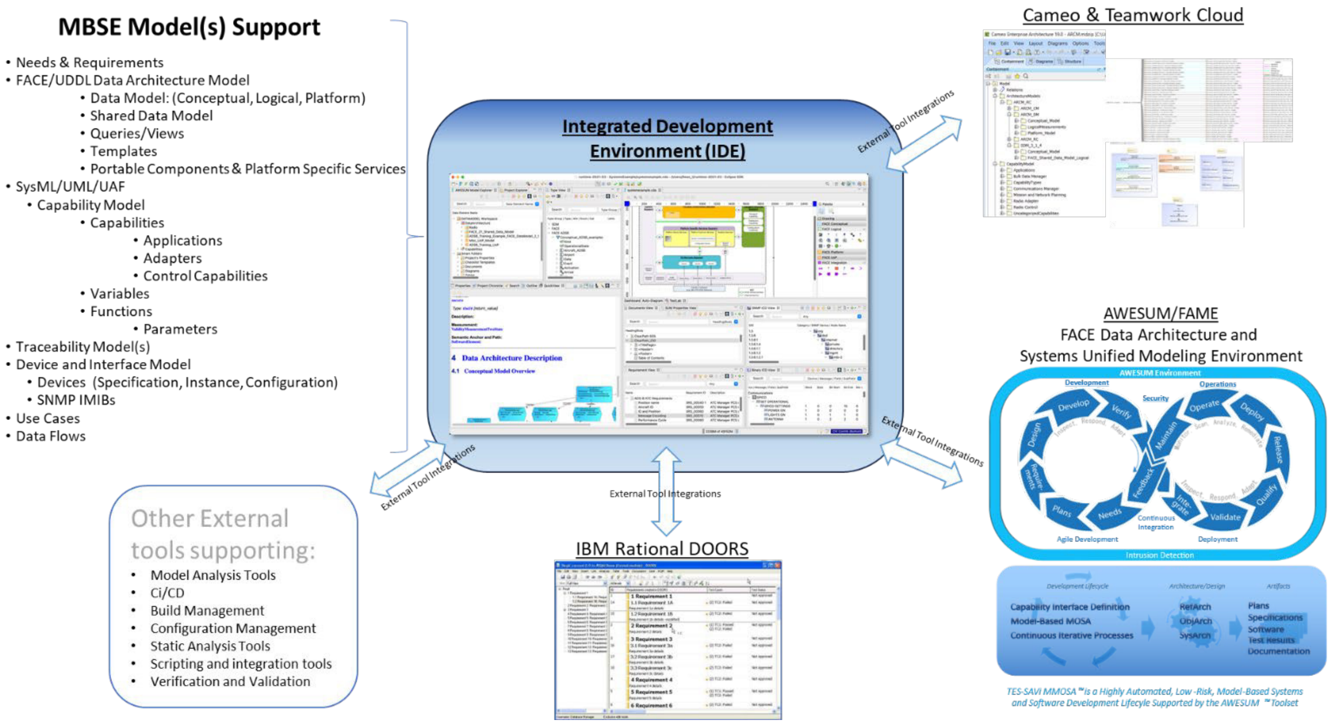

Figure 3: Integrated Development Environment

The IDE in Figure 3 connects different vendor tools to provide the program leadership with a unified and synchronized view for documentation and analysis. It supports the modeling of system needs and requirements to specify the desired objectives and constraints placed on the system. Due to its modular nature, the AWESUM® tool suite can be a keystone in a distribution of disparate IDEs as it can manage the end-to-end development lifecycle, while also seamlessly integrating other Model Based Systems Engineering (MBSE) tools, model formats, and coordinating the various contents and products.

- UDDL is leveraged to document the semantic definition of a system's key data elements.
- SysML/UML/UAF are utilized for documenting the key elements of the system and the system architecture as a satisfaction of the system requirements.
- Use case models are used to identify the use threads and data flows tracking needs and requirements throughout the system design.
- Traceability Models map the requirements, design, tests, and results.
- Device and interface models represent the Interface Control Documents (ICDs).

DOCUMENT GENERATION

AWESUM® contains and manages all the engineering and design data. As a result, it has the capability to auto-generate all a program's engineering and design documentation. This drives massive efficiencies in support of development reviews, high-integrity evidence reviews, cross-discipline gateway reviews, and stakeholder briefings.

KEY TECHNICAL ENABLERS

This section describes some of the key technical enablers within AWESUM® that enable it to be the single source of truth for all of your Program's engineering and design data, and to cohesively integrate cross-discipline concerns into the program's systems engineering process.

SYSTEM UNIFIED MODEL

The AWESUM® System Unified Model (SUM), which is a unified system, software, and hardware model, is employed to capture and store all aspects of the system lifecycle with sufficient specificity for formal methods to be applied and is implemented in a hypergraph database.

The SUM manages all the data and relationships across the following comprehensive set of models.

- Needs & Requirements Model
- Traceability Model
  - Directional Traces with Bidirectional navigation
- Trace Matrices Data Architecture and Model
- Document Artifact Model with document templates
- Architecture Analysis Description Language (AADL) subset
- Capability-Functional Model
  - Capability Application
  - Capability Adapters (Device to Capability Interface)
  - Capability Interfaces (Abstract functional interfaces)
  - Capability Devices (Abstracted device interfaces)
  - Semantic Modeling for Capability Interfaces
  - Data Transformations
  - Device Component Interface Control Description (ICD)
  - Device Components and Interfaces
  - Binary ICDs
  - Tag Based Textual ICDs
- SNMP/MIB
- Test Model
  - Test Harness
  - Test Cases
  - Test Procedures
  - Modeled Tests
  - Test Results

MODEL-BASED MOSA (MMOSA™)

To support the government's stated objective of model-based system engineering and open systems architecture, the AWESUM® suite of tools deliver significant savings through a Model-based MOSA (MMOSA™) approach. Our tools and processes significantly reduce the early-stage impact of open system approaches on the development timeline, easing product development, testing compliance and providing the basis for future products and system improvements.

Precise Systems' MMOSA™ process is focused around formally capturing the architectural information required to deliver an open architecture system within a MOSA enterprise. Key Concepts to MMOSA™ are **Continuum** of full Systems Lifecycle; **Multi-Discipline** design and management for Authoritative Source of Truth (ASOT), including

Provenance maintained and supported in Models, Holistic Systems Development Process, Standards and Architectural levels used as foundation for requirements and design; **Validation** throughout Process; comprehensive **Certification /Qualification** support; **Unified** Project data. The MMOSA™ approach greatly increases the information and data management throughout the program lifecycle, enabling reuse of design artifacts, standardization of architectural concepts, and increased efficiency during system integration. The three architectural levels are.

Reference Architecture: A Reference Architecture defines the enterprise / community of interest. Within the enterprise scope, it then captures the business goals and key business drivers of the enterprise stakeholders. To manage the control of programs within the enterprise, the Reference Architecture details/provides all applicable patterns, processes, standards, tools, guidance, and doctrine requirements. It may also contain commodity technical products that are expected to be reused across the enterprise, for example a Domain Data Model. The Reference Architecture must be unambiguous and available to the whole enterprise as it will support and manage the development of multiple product lines, and will support traceability from products, back to the reference concepts.

Objective Architecture: An Objective Architecture is a controlling architecture, that is derived from the overarching requirements of the Reference Architecture and is applicable to a family of systems (FoS) branch within the enterprise – this branch could alternatively be called a product line. The Objective Architecture sets the requirements from which the various System Architectures in the product line will be developed and the architectural properties and requirements that will define a product line. It will also determine the Product Line Engineering (PLE) strategy and process which will control the necessary variance across the product range. The Objective Architecture must be unambiguous and available to the product line enterprise as it will support and manage the development of multiple systems within the product line, and will support traceability from products, back to the reference concepts.

System Architecture: A System Architecture is a design specification that is derived from the product line requirements of the Objective Architecture and is applicable to a product. The System Architecture sets the requirements from which a product will be designed. The System Architecture includes the functional decomposition of system components, data requirements for interfaces, system performance requirements, interoperability and communications requirements, hardware and software architecture requirements, and system-level behavior requirements. All relevant scope must be traced back to the requirements set by the Objective Architecture.

Through the systems engineering process, and in consideration of the layered architectural approach described above, the MMOSA™ process provides your Program with an intrinsic approach to include high-integrity aspects of the requirements set. The AWESUM® tool suite provides an environment to manage all the MBSE aspects of the enterprise, the incorporation of DevSecOps, Agile Methods, and Formal Qualification, and the optimization of integration and test activities. The scope of the AWESUM® tool suite is detailed in Table 1.

Lifecycle Effort	Needs Space			Solution Space								
	Needs	Plans	Requirements	Design		Develop	Verify	Integrate			Validate	Qualify
				Software Models	Hardware Models			SW/SW Integr.	HW/SW Integr.	System Integration		
SysArch (PSM) / ObjArch (PIM) / RefArch	Requirements Model Semantic Data Model Use Cases	Process Model (BPM)	Requirements Model Semantic Data Model Use Cases	AADL (SW Design, Platform Model) Capability Interface	AADL (HW Logical Design)							
SysArch (PSM) / ObjArch (PIM)		Document Model		Logical Data Model		Build Process Model Code & Test Generator(s)	*Test Harness*					
SysArch (PSM)		*Costing & Scheduling Model & Tool (COSYSMO)*		Capability Adapters/ Applications Model Platform Data Model, UoP Model Action Behavior Model Device Component ICD	*Mechanical Design* (SolidWorks, FreeCAD, ...) *PCB design* (Altium) Action Behavior Model Device Component ICD	*Compiler*	Executable Model Test Model *Source Code Analyzer*	DM Integration Model AADL	AADL Device Component ICD	AADL Device Component ICD	Needs Use Cases Scenarios	Requirements & Traceability Models *Structure Coverage* Reviews & Checklists Verification Plans, Results Document Final Artifact Model

Table 1: AWESUM® MMOSA™ Scope

CAPABILITY ENGINEERING

Modelers begin with the system needs and concept of operations which allows for early system specification and provides a means to virtually integrate and validate the system throughout the lifecycle. The system architecture, requirements and design are then modeled. The detailed external interfaces of the system are modeled to produce a foundation for document, code and test generation, and verification against high-level and low-level requirements. All levels of the model are traced to their predecessor elements to provide a complete navigable SUM.

The ease of use of the AWESUM® is enhanced through the provision of a simple drag-and-drop user interface. This facility is enabled for activities related to data and system modeling, code generation, and drag-and-drop verification.

DATA ARCHITECTURES

AWESUM® utilizes the Universal Domain Description Language (UDDL) for modeling system data. Data models capture all data entities, their characteristics, and relationships relevant to an identified system (or system-of-systems). Unlike other data model approaches, UDDL utilizes a formal modeling approach to capture the semantics of the system data. Using the model, a data semantic can be traced from a data instance back to a core concept. The data model approach enforces rigorous capture and documentation of the uniqueness of data instances, the specialization of data types, and data-sub-domain branching and representation of data. The result is a machine-readable construct where each data entity is defined uniquely ensuring that semantic ambiguity and polymorphic entities are eliminated when processing data.

The UDDL data architecture in AWESUM® can be utilized to develop, manage and deploy a data model that documents all data related to the following mission-relevant domains.

- Intra-system communications
- Inter-system communications
- Command and control
- Sensor data propagation
- Mission systems and weapon management

The data architecture not only captures the data aspects described above, but it also supports the projection of cohesive sets of data items, known as views, which can be used to represent interface data requirements.