Combining Federated UAF and SysML Models for a Bomb Live Unit, Optimizing Objectives with Compartmentalization

Logan Fegenbush
Colorado State University

Dr. Daniel Herber
Colorado State University, Fort Collins, CO

Special thanks to Zakary Wilde, Lani Seaman, and Kaelin Glover
Project Abstract

Model-Based Systems Engineering (MBSE) represents a new paradigm in executing large programs and projects for designing complex defense systems. Many defense systems are developed across multiple organizations, through multiple facilities, with stakeholders both within the federal government, private sector industry, University Affiliated Research Centers (UARCs), and Federally Funded Research and Development Centers (FFRDCs). The Needs, Goals, and Objectives (NGOs) of these stakeholders are often compartmentalized, and not commonly understood across these defense enterprises. While needs will serve as drivers to the system for what problem it is supposed to solve, goals will elaborate on the needs and establish the expectations of the system and allow a way to assess if the system has achieved them. Objectives in this context serve as the specific, measurable results of the system and how it meets the goals and consequently the needs of the stakeholders. Objectives will focus on the desired outcomes in a quantifiable and verifiable manner, directly informing, or even behaving as source requirements for the system.

Developing a federated MBSE framework that will enable design implementation and iteration, while addressing Measures of Effectiveness (MOEs) derived from the NGOs introduced by the variety of stakeholders, will help unify the defense sector in quickly adopting new solutions to emerging problems. Relying on the Modular Open Systems Approach (MOSA) terminology, a federated collection of stakeholders supplied operational and program models, existing design and manufacturing capabilities, models of the corresponding organizations, and multi-solution design models can be outlined to show complete traceability from the program objectives down to the critical components. Such a collection of federated models may enable defense programs to optimize the system design through iteration of the programmatic needs, flowing those NGOs down to the operational models, parsing the MOEs into a system design model, and fully exploring the trade space informed by technology readiness, manufacturing capabilities, budget, and schedule.
In this paper, the author explores the concept of using Unified Architecture Framework (UAF) models as programmatic, operational, organizational, and manufacturing capability descriptions in a federated context to inform major system platform System Modeling Language (SysML) design models, and major system component SysML models of the potential design space. The expectation is that these UAF models will be provided to the designers to begin concept exploration of the design space. UAF models will capture programmatic objectives of major system platforms, obfuscating the operational capabilities of these platforms as required by security policy and deriving the operational envelope for the specific capabilities needed for the mission. In this manner, the complete capabilities and intent offered by these platforms can be appropriately compartmentalized, while fully describing the objectives that need to be met by the major system components. In this example, military aircraft present the major system platforms, while conventional gravity bombs provide the major system components.

This study will extend the SysML attributes to the UAF metrics captured in platform and component models, explore the MOSA considerations of digital transformation for defense programs, and validate stakeholder needs beyond the context typically presented through exacting source and system requirements traceability. Additionally, modular interfaces will be demonstrated to capture how existing designs can be modified to meet new or changing objectives presented in such operational envelopes. These examples will provide a foundational approach for defense system acquisition and development, as well as demonstrating the art of the possible for applying MBSE to these acquisition and development cycles, without driving the need for complicated, monolithic, and hard to digest enterprise models.
Contents

Introduction ........................................................................................................................................... 7

Federated Modeling Methodology ........................................................................................................ 12

SysML Frameworks and Architectures ................................................................................................. 15

Guided Bomb Unit ............................................................................................................................... 33

GBU Operational Architecture ............................................................................................................. 37

UAF Models of Major System and Mission ............................................................................................ 38

Concluding Remarks ............................................................................................................................. 45

Appendix 1: Enabling System Verification ........................................................................................... 47

Appendix 2: Modeling Schema ............................................................................................................. 50

Appendix 3: Systems Engineering Review Dashboards .......................................................................... 51

Glossary .................................................................................................................................................. 52

References .............................................................................................................................................. 53

Public Web References on Munitions .................................................................................................... 54

List of Figures

Figure 1 Mission Metrics Overlap with MOSA Entity Metrics ............................................................... 8

Figure 2 Federated UAF Models Define Mission Space ......................................................................... 9

Figure 3 Three Pillars of MBSE (Delligatti 2014) .............................................................................. 10

Figure 4 UAF 1.2 Viewpoints (Columns) and Domains (Rows) (Object Modeling Group, 2022) .......................................................... 11

Figure 5 UAF and SysML Model Structures and Interfaces ................................................................. 13

Figure 6 Requirement Attributes Describe System Expectations ......................................................... 14

Figure 7 Notional Source Requirements Derived Structure .............................................................. 16

Figure 8 Derive System Requirements from Source Requirements .................................................... 17

Figure 9 Notional Functional and Logical Architectures .................................................................... 18
Figure 38 SysML Modeling Schema Derived from NASA Systems Modeling Handbook ..... 50
Figure 39 Dashboard Tracing Functional Requirements, Performance Requirements, Logical
Structure, and Physical Architecture............................................................................................ 51
Introduction

Large defense programs are often developed and executed as cooperations between many public and private organizations with different expertise and data constraints. Organizations must share specific domains of information to fully integrate these large defense program efforts. Much of this information sharing is constrained by classification, proprietary information, subject matter expertise, and other need-to-know cultures or concerns. This compartmentalization between these organizations for defense programs may result in suboptimal solutions and higher program execution costs. Such shortcomings are observed in the recent F-35 Joint Strike Fighter and Sentinel land-based missile programs (Government Accountability Office, 2023).

To address these programmatic shortcomings, the Department of Defense (DoD) leadership have begun embracing model-based methodologies, including Model-Based Systems Engineering (MBSE) and modular approaches to encourage sharing and optimization. In 2019, the DoD introduced the Modular Open Systems Approach (MOSA) as a business and technical strategy to overcome challenges in information constraints and costs (Department of Defense, 2019). In conjunction with MBSE, MOSA may be used as a conveyance of defining common major interfaces between systems and components to increase reusability, broaden supplier options, and allow for rapid development of new technologies.

Even given these new tools and methodologies, problems remain between organizations where NGOs are analyzed and parsed down to a single set of requirements that are then passed off with minimal explanation or insight. These requirements, through security constraints related to classification or perceived need-to-know (NTK), form a choke point on information sharing where true system validation becomes challenging. With increased uncertainty and confirmation from stakeholders that the correct system is being pursued, technical shortcomings as well as cost and schedule overruns should be expected.

Extending the philosophies presented from model-based approaches and MOSA, a mapping of key attributes and metrics could be realized by model interfaces between the MOSA-defined Major System Platform and Major System Component entities. MOSA
defines the major system platform as the highest-level structure of a major weapon system that is not physically mounted or installed onto a higher-level structure and on which a major system component can be physically mounted or installed (10 USC 4401(b)(2)). MOSA defines the major system component as a high-level subsystem or assembly, including hardware, software, or an integrated assembly of both, that can be mounted or installed on a major system platform through modular system interfaces (10 USC 4401(b)(3)(A)). These together compose the major weapon system specified and designed for specific capability or mission execution.

![Diagram](image)

**Figure 1 Mission Metrics Overlap with MOSA Entity Metrics**

A proposed method of creating a more modular approach to requirements, while preserving as much classification as possible, involves attempting to map the key attributes from the major system component to the major system platform, as they apply to the stated mission space. While the capabilities of the major system platform, and the objectives of the mission can remain tightly held, the mission metrics as they apply to the platform and component can be derived and traced (Figure 1). In this manner, the organization responsible for the major system component can be explicit in exactly what information is needed to fully validate the solution.
Federally Funded Research and Development Centers (FFRDC), as well as University Affiliated Research Centers (UARC), are often the organizations that take on the tasks of validating mission metrics, as well as act as the design agencies of the key functional components. The unique capabilities offered by these organizations introduce a layer of abstraction between the Department of Defense and the vendors and contractors delivering the physical assemblies. Figure 2 introduces a proposed federated framework of UAF models and SysML models that represent different aspects required to realize a new capability. SysML, in its current incarnation, represents the best practical language to define a method for conducting system design trades as a system architecture is
Specific tools, when combined with a method and language, provide a litany of simulation and parametric analysis capabilities defining a core tenant of MBSE. These relationships are visualized in Figure 3.

Figure 3 Three Pillars of MBSE (Delligatti 2014)

UAF is a growing desire for DoD programs looking for a model-based language and method as they undergo a digital transformation to a models-based paradigm. UAF includes the framework of views and perspectives that have evolved out of such frameworks as the DoD Architecture Framework (DoDAF) and other similar efforts. UAF also includes a modeling language extended from SysML, referred to as UAFML. The addition of included stereotypes and artifacts in UAF provide an all-in-one open standard for these DoD programs to adopt with lower overhead. The UAF standard, however, does not include a native simulation capability and is not well suited to tailored solutions for very specific design efforts. UAF appears best suited for describing capabilities and resources from an enterprise perspective.

As UAF is not ideal for design work, SysML remains the choice for architecting new system solutions to meet the NGOs provided to the design agencies. Where UAF is best at
capturing capabilities and subsequent constraints, the flexibility of SysML, with the right tool and a defined method, can capture architecture definitions and execute trade studies. A federated structure of UAF and SysML domain models as shown in Figure 2 defines the mission space needed to map mission platform metrics and mission system metrics as MOEs as they apply to the solution context. Additionally, the federated UAF models can feed program success factors (Shao et al., 2012) into the architecture definitions to better evaluate trade space.

<table>
<thead>
<tr>
<th>Architecture Management</th>
<th>Motivation Mv</th>
<th>Taxonomy Tx</th>
<th>Structure Sr</th>
<th>Connectivity Cn</th>
<th>Processes Pr</th>
<th>States St</th>
<th>Sequences Sq</th>
<th>Information If</th>
<th>Parameters Pm</th>
<th>Constraints Ct</th>
<th>Roadmap Rm</th>
<th>Traceability Tr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategic St</td>
<td>Strategic Motivation St-Mv</td>
<td>Strategic Taxonomy St-Tx</td>
<td>Strategic Structure St-Sr</td>
<td>Strategic Connectivity St-Cn</td>
<td>Strategic Processes St-Pr</td>
<td>Strategic States St-St</td>
<td>Strategic Sequences St-Sq</td>
<td>Strategic Information St-If</td>
<td>Strategic Parameters St-Pm</td>
<td>Strategic Constraints St-Ct</td>
<td>Strategic Roadmap St-Rm</td>
<td>Strategic Traceability St-Tr</td>
</tr>
<tr>
<td>Operational Op</td>
<td>-</td>
<td>Operational Taxonomy Op-Tx</td>
<td>Operational Structure Op-Sr</td>
<td>Operational Connectivity Op-Cn</td>
<td>Operational Processes Op-Pr</td>
<td>Operational States Op-St</td>
<td>Operational Sequences Op-Sq</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services Sv</td>
<td>-</td>
<td>Services Taxonomy Sv-Tx</td>
<td>Services Structure Sv-Sr</td>
<td>Services Connectivity Sv-Cn</td>
<td>Services Processes Sv-Pr</td>
<td>Services States Sv-St</td>
<td>Services Sequences Sv-Sq</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personnel Ps</td>
<td>-</td>
<td>Personnel Taxonomy Ps-Tx</td>
<td>Personnel Structure Ps-Sr</td>
<td>Personnel Connectivity Ps-Cn</td>
<td>Personnel Processes Ps-Pr</td>
<td>Personnel States Ps-St</td>
<td>Personnel Sequences Ps-Sq</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resources Rs</td>
<td>-</td>
<td>Resources Taxonomy Rs-Tx</td>
<td>Resources Structure Rs-Sr</td>
<td>Resources Connectivity Rs-Cn</td>
<td>Resources Processes Rs-Pr</td>
<td>Resources States Rs-St</td>
<td>Resources Sequences Rs-Sq</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Sc</td>
<td>-</td>
<td>Security Taxonomy Sc-Tx</td>
<td>Security Structure Sc-Sr</td>
<td>Security Connectivity Sc-Cn</td>
<td>Security Processes Sc-Pr</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projects Pj</td>
<td>-</td>
<td>Projects Taxonomy Pj-Tx</td>
<td>Projects Structure Pj-Sr</td>
<td>Projects Connectivity Pj-Cn</td>
<td>Projects Processes Pj-Pr</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standards Sd</td>
<td>-</td>
<td>Standards Taxonomy Sd-Tx</td>
<td>Standards Structure Sd-Sr</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual Resources Ar</td>
<td>-</td>
<td>Actual Resources Structure Ar-Sr</td>
<td>Actual Resources Connectivity Ar-Cn</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Simulation</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4 UAF 1.2 Viewpoints (Columns) and Domains (Rows)**  
(Object Modeling Group, 2022)

UAF parameters include Environments, Measurements, and Risks that crosscut the listed domains in Figure 4. These parameters can be included and applied to any domain. Specifically, measurements may be mapped that capture the previously mentioned mission platform metrics and traced down to mission system metrics, which may be delivered to the program and disseminated to the design as parameterized MOE spaces. From there, Measures of Performance (MOPs) from the major system component
architecture could be derived as system and subsystem parameters that fulfill the MOE spaces.

A strong motivation exists to describe a proposal to map UAF to SysML to maintain flexibility in design space with respect to the DoD program objectives. As the DoD and other federal agencies look to UAF as a fully realized solution, tailored SysML methods best describe the unique capabilities offered by the FFRDCs and UARCs performing significant and key aspects of the design. The current generation of SysML tools offer broad capabilities and relatively open access to plugins and extensions that increase the ability to tailor the system architecture analysis as needed.

**Federated Modeling Methodology**

For defining a system architecture in MBSE, as previously discussed, a tool, a modeling language, and a modeling methodology must be determined. In this case, the tools exist from multiple vendors, and the MBSE language of choice for this work is SysML. To fully understand how to map UAF attributes to SysML attributes in a meaningful way to the new system architecture, a SysML modeling methodology must be established. As UAF includes a modeling methodology in the current 1.2 standard, an understanding of the practices as described in the OMG literature (Object Modeling Group, 2021) is necessary.

Federated modeling incorporating the UAF description models of the capabilities, mission, and NGOs, as well as the SysML models of the design space, provides a complete description of the potential solutions to meet the needs. In addition to flowing the objectives refined into source requirements, the parameters for the mission, as they apply to the platform and the components per MOSA, can flow down to the major system component design space. This enables the design authority to not only meet the source requirements and objectives, but also provide margins and MOEs against the mission scope. Given the encapsulated black box parameters provided in the major system platform model logical grouping shown in Figure 2, parameters and instances may be refined from corner cases of the complete mission objectives, flowed through the major system platform model, down to the major system component model, allowing more beneficial sharing as they apply to the mission space.
Figure 5 UAF and SysML Model Structures and Interfaces

Figure 5 expands on the notional image in Figure 2 showing what specific interfaces may exist between the groupings of models. Figure 5 also shows the common modeling structure of the SysML model groupings, delving into defining operational, functional, and logical architectures that represent an overall framework, while physical architectures and physical reference architectures describe the realization of designs and existing art. These model structures are not absolutes, though they do represent the most convenient model
and package structures necessary for reuse and modularity of system and subsystem architectures.

Figure 6 Requirement Attributes Describe System Expectations

The proposed SysML modeling architecture is defined to provide context to how UAF elements may map to development and design elements in the architecture. As shown in Figure 6, MOEs directly correspond to source requirements as they are reiterated or derived from the NGOs of the program. In some cases, NGOs and source requirements may be redundant where source requirements only state a more refined description of the desired objectives. MOEs will describe the qualitative and quantitative attributes of the original objective. Similar to what is shown in Figure 1, the goal of this architecture is to map the entire parameter space from UAF to SysML.
Two or more MOPs will address the quantitative attributes of the MOEs per the capabilities and known art of the system design agency. Technical Performance Metrics (TPMs) further decompose the MOPs to measurable parameters which may include such examples as Geometric Dimensioning and Tolerancing (GD&T), stability, material properties, or uncertainty in performance. The referenced MOE, MOP, and TPM descriptions are further described in INCOSE and NASA literature (National Aeronautics and Space Administration, 2017; National Aeronautics and Space Administration, 2016). Further discussion of the verification methods as shown in Figure 6 will be included in Appendix 1 below.

**SysML Frameworks and Architectures**

The SysML detailed modeling will be broken up into four specific architectures: Operational Architecture, Functional Architecture, Logical Architecture, and Physical Architecture. These architectures will be used for expressing different domains of the complete system architecture, which include the system integration, subsystems, and components for design agnostic content, physical reference architectures and assemblies for design-specific content, all completed architecture decisions, metamodels, and style guide schemas for modeling definition. The SysML standard includes some definitions for the metamodel, and the modeling schema will provide additional required definitions. Some basic ontological definitions may be required depending on the domains being modeled and existing understanding of those problem statements and system context.

While any modeling domain may address any specific architecture, generally the overall framework models will consist of the operational, functional, and logical architecture, while the physical architecture will rely on those frameworks for behaviors, logical groupings, and parametric relationships. Figure 5 shows several details of this with the system integration modeling primarily consisting of an operational architecture and physical architecture. It is expected that different types of derived requirements will be distributed throughout the models based on the modeling context.

To begin, the operational architecture will focus on analyzing needs, goals, and objectives or other source requirements as a canvas to understand the problem space, including
parsing parameters from any source text to define measures of effectives for the proposed system context. The described architecture provides the canvas for understanding stakeholder objectives, validating provided requirements, and exploring how the functional and logical architectures may be created to meet those objectives. This step may apply to systems, subsystems, or components, allowing trade space exploration at each level of integration. Ideally a deeper understanding of the parameters and attribute space will provide traceability to source requirements, and traceability to UAF metrics to meet the customer objectives and fully validate the operational needs of the stakeholders.

![Diagram](image)

*Figure 7 Notional Source Requirements Derived Structure*

When deriving source requirements from the consumer point of view, some functional requirements may be described, and the corresponding expected performance metrics parsed to form a structured decomposition of the source requirements that can address
Key Performance Parameters (KPPs) (International Council on Systems Engineering, 2023, p. 97). Figure 7 shows notionally some structure that may be constructed based on the source requirement or objective containing a KPP, with a base functional requirement, or several functional requirements, and the corresponding performance, constraint, and non-functional requirements (NFRs) traced directly to that derived function. The analysis of the requirements in this manner provides a level of requirements validation to the stakeholders or other providers of the NGOs to the system, demonstrating that the objectives of the system are understood.

![Diagram](image)

**Figure 8 Derive System Requirements from Source Requirements**

As shown in Figure 8, a top-level system functional requirement may be derived from that source requirement or objective structure that is based on the specific capabilities and
technologies of the system designers. This first step helps to define the type of systems that may serve the NGOs at differing levels of effectiveness.

The functional architecture structures the functional requirements to form a functional breakdown structure (International Council on Systems Engineering, 2023, p. 174). This functional breakdown structure (FBS) can support risk-based analysis methodologies, like failure modes and effects analysis (FMEA), and fault-tree analysis (FTA), or other hierarchical probabilistic risk assessment methodologies. Understanding the risk in a quantitative manner may help drive feasibility and decisions of different system types.

![Diagram showing functional and logical architectures](image)

*Figure 9 Notional Functional and Logical Architectures*

With a general functional breakdown defined, the modeler may generate activities that satisfy the functional requirements, and structure activities with functional flow block diagrams (FFBD) to define functional interfaces and sequential behaviors, forming the other half of the functional architecture, describing which functions have dependencies.
that crosscut the functional breakdown structure. Examples of this are shown in the next section.

The logical architecture notionally shown in Figure 9 through the system and subsystem block classifiers should inherit or otherwise import the behaviors defined through the functional architecture. These blocks, through one or more methodologies, incorporate functional dependencies to define systems and subsystems as solution-agnostic modules. The blocks of this logical architecture will contain attribute definitions, behavioral allocations, dominant functional interfaces, parametric relationships, and traceability back to the functional architecture. Once completed, the operational, functional, and logical elements will create a completed framework that can be used as design patterns for various types of systems.

Figure 10 Notional Tracing of Performance through a Framework
The physical architecture defines product definition and assemblies, which inherit the attributes, behaviors, parametric relationships, portions of the structure, and requirements from the logical architecture. At this level, attributes will have values defined based on the physical implementation. The physical architecture may include a physical definition, a product breakdown structure (PBS), including part number information for high fidelity models, and serial number information for lifecycle maintenance models.

Figure 11 Notional Tracing of Performance and Constraints through a Framework

Figure 10 and Figure 11 demonstrate physical definition trade options as they trace through the frameworks, and the corresponding performance requirements and constraints that may be relevant to that logical architecture. Specific design constraints may trace directly to a physical definition if such options include additional concerns like environmental conditions a particular selection must adhere to. Using this logical architecture structure to trace performance requirements allows the physical architecture options to be switched back and forth, creating a repeatable method of conducting a trade study analysis even as the objectives and source requirements change.
Figure 12 Completed Notional Structure Builds System Architecture for Conducting Trades
Figure 12 conveys the overall framework and physical architecture approach to show the ability to apply consistent recursive modeling to the SysML system architecture. The type of system, and the form that system takes, are potential outputs generated from this methodology. Specific outputs from the completed architecture can be documentation related to requirements, requirement verification and validation planning, trade spaces, and even parametric analysis of the defined attributes. As shown, the two physical architecture options are parametrically evaluated with the results captured as instances with passing criteria highlighted in green, and failing criteria highlighted in red.

This proposed SysML modeling methodology establishes an easy to maintain and easy to create architecture that does not require deep existing knowledge of how MBSE works, when used with the described modeling schema and overall federated modeling structure. The examples below should clarify the practical application of these frameworks and modeling methodologies to meet the objectives of multiorganization federated modeling, establishing a baseline approach to evaluating the NGOs and attributes provided by the customer.

**Bomb Live Unit**

As a working example, a federated collection of SysML models will describe a Guided Bomb Unit (GBU), composed of a Bomb Live Unit (BLU), Flight Kit, and corresponding performance metrics that will later be mapped to mission metrics of the given major system platform, in this case an attack aircraft and the corresponding mission space. To start, the BLU functional, logical, and physical architecture will be described leading to the completed GBU architecture.
Defining the functional architecture of the BLU includes creating the FBS and defining the functional behaviors to later generate the different behavior sequences that may occur as shown in Figure 13 (International Council on Systems Engineering, 2023, p. 74). Some functional requirements are both needed to satisfy the parent functional requirement, e.g., “Survive hard-target penetration” and “Fuze warhead,” others may be optionally excluded, e.g., “Contact fuze” or “Delay fuze.” Later it will be shown that functional requirements will naturally drop out of the system definition if no corresponding performance metrics are traced to them for a given system architecture. The described traceability creates a cut set of needed functionality from the defined structure. A cut set in this context describes a minimum set of subfunctions that accomplishes the system objectives. In graph theory, a cut is the act of breaking an edge such that several broken edges may result in a connected subset of vertices or nodes. This usage is analogous to cut sets in FTA, where the minimal cut set determines all failure paths through the events that results in system failure (Ostrom, Wilhelmsen, 2019, p. 192). This cut set of functions
will generalize the idea behind the “type” of system being considered. Types of systems may be thought of as trains, cars, or aircraft as means of performing transportation. In this context, the system type may be a hard-target penetrator or general-purpose bomb with different functional intent. The logical gate icons shown express the combinatorial options that help define the potential logical architecture definitions.

While the functional breakdown structure illustrates the “type” of system that is being defined, or option space of systems, the logical sequence of the corresponding behaviors, through a FFBD, or activity diagram in SysML, will illustrate what “form” the system may take, which naturally leads into a formative discussion of the logical system definition. In this context, the form may refer to key features of the system, like how many wings it may have, or the means of powering the locomotion, depending on the form and sequence of the needed functions. In addition to the sequence, the FFBD captures the inputs, the outputs, and the interfaces between those functions that compose the logical architecture artifact.

Figure 14 Formative Functional Flow of a Penetrator Bomb

Figure 14 visualizes the first potential cut set of behaviors captured in the FBS in Figure 13 through the FFBD shown. This set of behaviors begins to describe the logical architecture definition of hard-target penetrator. This first logical system definition allows these behaviors to be binned to a common classifier for the stated system type. As previously discussed, the inclusion of the functional architecture shown in the FBS and FFBDs, coupled with the logical architecture shown in SysML block definition diagrams (BDDs) and internal block diagrams (IBDs) complete the framework of a proposed architecture.
As a contrast to Figure 14, Figure 15 visualizes a cut set of behaviors for the general-purpose bomb logical system. In this case, the outputs remain unchanged as weapon effects, however, the fuzing method of the warhead changes significantly. Additional functional architectures described through FBSs and FFBDs may introduce inertial fuzing options that detonate the warhead based on additional environmental factors beyond contact fuzing.
While not all behaviors of the functional architecture are described, the logical architecture can begin to inform the formative considerations of the system, including system types alluded to by the FBS. In this case, Figure 16 details specific features and composition of the BLU and include placeholders for the key attributes of these features. Additional types of general-purpose or hard-target penetrators are captured. Iteration between the FBS, the FFBDs, and the logical architecture may increase the potential trade space of solutions that exist within the overall framework.
The physical architecture of the BLU describes the different types and forms of BLUs, with specific physical references to existing or proposed physical definitions. In this case, the BLU-109 represents a penetrator, and the BLU-117 represents a general-purpose solution. The penetrator option has the inherited attributes and parameters from the framework defined. Additionally, material properties may be inherited. For instance, the bomb casing definition uses gun-barrel hardened steel. This material may be captured in a physical reference architecture that is used across many models.
Performance metrics can be traced to the functional breakdown structure. Using the logical elements described in the frameworks maximizes the potential of testing multiple physical solutions without significant modeling rework. In Figure 18, as the framework attribute is used for the satisfy relationship, tracing multiple physical definitions to the same requirement isn’t necessary. Additional specific design constraints identified by the physical definition may be traced to the functional requirement, for example a requirement constraint on impact angle specific to the BLU-109 could be used.
Flight Kit

As shown in Figure 19, the flight kit FBS, with corresponding activities as satisfying conditions, describes the main function the flight kit design space needs to fulfill. As before with the BLU FBS, some requirements will be determined by the type of flight kit used on the completed guided or unguided bomb unit. A cut set of one or both functional sub requirements may be included, and the corresponding activities may be sequenced into a formative functional architecture of the flight kit. As an argument of where the FBS could be considered complete, any functional requirement that has more than one activity satisfying it suggests it is not fully decomposed within the FBS. The FBS may need to continue until such functions are met by commercial off the shelf (COTS) components, well understood components for existing art within the organization, or until a first principle parametric relationship can be identified. Additional decomposition and definition may better define what functional interfaces may exist between the various cut sets of FBS options.
Logical models of the flight kit describe several different subsystems that can change the flight dynamics, and ultimately, the operational envelopes of the weapons. Many attributes will exist for consideration however, none are shown in Figure 20 for this example. This logical architecture does convey a sense of how many things can change between one completed bomb unit or a different option using the same basic BLU. This logical architecture also alludes to what the higher-level completed system may look like. Significant changes in performance attributes could depend on what type of fin assembly is used in the physical definition. Additionally, the accuracy of the weapon system may change significantly depending on flight kit choices. A change to accuracy may map back to what the operational envelope of the major system platform needs to accomplish for the specific solution.
With a logical architecture for the flight kit defined, additional physical architecture definition can be modeled to describe specific capabilities. Figure 21 proposes a Joint Direct Attack Munition (JDAM) fin assembly (US Air Force, 2011) as a solution to a conical fin assembly. This tail kit is compatible with BLU ring groove solutions for attaching different tail kits. In this case, the JDAM is compatible with a Mk84 Ring Adapter groove, a shared ring adapter groove design on the BLU-109/B, the previously modeled warhead, as well as the BLU-117 general-purpose bomb shown in Figure 17. This modularity emphasizes a key aspect of MOSA for mechanical interfaces, allowing a BLU to be upgraded with a guided tail kit, or more accurate tail kit, as technology matures, without resorting to a complete ordnance redesign.
Figure 22 Flight Kit Performance Requirements

As shown in Figure 22, the accuracy of the selected JDAM tail kit is very good, exceeding the performance requirement that is described for hit probability (purely notional values). The totality of performance between all the system components will describe the final effectiveness of the point solution or describe the option space for additional solutions.

As shown in the next section, the FBS can be combined into a single functional architecture with additional cut sets for solution space. These FBS are combined by the top-level functional intent of the major system component. In the next section, Guided Bomb Unit, the system will be integrated into a completed solution designed to meet the stated system intent.
Guided Bomb Unit

Using the BLU Physical Reference Architecture and the Flight Kit Physical Reference Architecture in a combined Guided Bomb Unit integrated model will allow the functional breakdown structures of both models to be combined into a single functional architecture. Physical reference architectures may describe pre-existing technology and subsystems with well-defined operational and performance capabilities. In Figure 23, the top-level function of “Destroy intended target” may have a performance criterion traced to it that describes the probability required for achieving the function of a given system. Additional analysis may reveal functional cut sets or physical options that increase the capability or probability of achieving that function. This forms a trade space against the objectives described by the customer through a common modeling structure.

Figure 23 Combined Functional Breakdown Structure of the Integrated Guided Bomb Unit
The completed GBU physical architecture of the system shown in Figure 24 shows additional required physical definition as the flight kit and BLU reference architectures are combined. In particular, the fuze assembly for a delay type is installed between the BLU and the flight kit tail section. This holds true if the fuze assembly should be installed between a guided nose assembly and the BLU. A further need for a fuze solution provides the delay fuze attribute shown in Figure 23 through FMU-143 device. Additional physical reference architectures and framework models may define other fuze types and performance attributes, including new functionality, like an inertia fuze. Next, the performance of the delay timer of the fuze currently defined will be captured as a performance requirement for the integrated system.
As before shown in the BLU models, performance criteria can be applied to the functional breakdown structure of the GBU physical solution space. The specific combination leads to a worst-case published result of 30 meters Circular Error Probability (CEP) as shown in Figure 25 (US Air Force, 2011). Furthermore, the minimum time of the delay fuze is described with a Fuze Timing performance requirement of greater than or equal to 4 seconds. This integrated block diagram again demonstrates how the framework models can be used to trace many physical options to the same attribute. If optional fuze solutions are defined, they will have specific delay timer performance attributes that can be evaluated against the required time.
Figure 26 demonstrates the synthesizable model to explore the trades. A well-formed model that simulates once, and is validated for corner cases, can be configured for statistical studies referenced against the UAF metrics of the operational envelope. Within verbose tool environments, simulating a model assures continued validation that the model is well formed with respect to the modeling schema and SysML validation rules. Validating the model does not verify or validate the model outputs with respect to the design, and additional steps must be taken to create a predictive model for verification and validation of the system once realized. As shown in this image, the selected tool provides a stop light chart of go/no-go evaluation of the solution, where red rows display failed criteria. From the major system component perspective, this simulation can continually show when the model meets the customer objectives, even when those objectives change. This capability shows one side of validating the mission space.
The operational architecture of the integrated system model will serve as a canvas for modeling how the proposed or existing solution will map to and meet the corresponding MOEs. The MOEs are defined here from the needs, goals, and objectives that are pulled in from the UAF model or parsed out of the stakeholder requirements. Figure 27 demonstrates the relationships between source requirements or objectives, as shown on the far left, and the physical architecture through the warhead effectiveness constraints shown in the center. This diagram is notional and doesn’t expand upon the needed physics and engineering parametric relationships to validate the solution.

In this example, the UAF content may present the attributes from the major system platform, like flight trajectory, speed, and drop height, while the SysML content will address the attributes of the major system component, including drop trajectory, final impact angle, and penetration effectiveness. The attributes of the major system platform...
may change as the operational context of that platform changes, as such, the intermediate UAF model will provide the expected envelope of values (Figure 1) but may not provide all the details of the major system platform normally captured in the SysML major system platform design modeling content.

This federated approach will allow a level of compartmentalization to occur between the major system platform, how that platform is to be used for the objectives of the stated program, and the major system component models consuming that content as delivered by the UAF modeling. The next few modeling figures demonstrate a notional UAF model of a source capability that can be used to generate the black box subset of parameter boundaries to be fed to the described system architecture captured in SysML.

**UAF Models of Major System and Mission**

![Diagram of UAF models]

*Figure 28 Strategic Notional Flow Informs Needs, Goals, and Objectives*

UAF provides several different domains to provide context for different aspects of the information being modeled. In this paper, the three key domains used are Strategic,
Operational, and Resources domains which are shown on Figure 4. The first domain shown, the strategic domain, shows three types of artifacts, driver, challenge, and opportunity. Per the UAF metamodel, these artifacts can be related through relationships shown in Figure 28. Notionally, the opportunity is motivated by the challenge, which is presented by the driver. These artifacts may be analogous to the NGOs described by the stakeholders where a driver defines the needs, and an opportunity may allude to the objectives. Additionally, UAF provides viewpoints of different aspects of each of these domains. In particular, the parameters viewpoint can provide the required attributes for the needed context. As previously stated, parameters can be applied to any domain within a UAF model.

Figure 29 Parameterizing Capabilities to Define Mission Envelope

From a given opportunity, the capability, either existing or proposed, is decomposed into the key capabilities that are required to meet the parent capability. The parent capability shown in Figure 29 is impacted by the new technology opportunity, potentially expanding the parameter space for deploying and delivering a weapon. As each capability is defined, an included measurement set is created that captures the key attributes needed for that capability. Generalizing a mission envelope measurement set from the defined capability measurements captures in one place the needed attributes to achieve the parent capability.
Further UAF modeling incorporates the operational domain of the model context. These operational performers provide the capabilities to perform operational activities, which in turn perform the capabilities described in the strategic domain. The major system, which is composed of the major system platform, and major system components, is defined. As such, the measurement sets applied to the capabilities are also relevant to the operational performers. At this point, the major system component attributes can be defined as a set of the mission envelope attributes. The major system platform measures may change depending on the selected solution, often pre-existing, that brings the payload to the target area. This selected measurement set will also refine the delivered black box of parameters needed for the design space.

With the operational performers identified, and the complete set of relevant attributes captured, the boundaries on the attributes can be established. This can include identifying information constraints for security concerns between organizations, necessitating information security requirements as applied to existing capabilities. UAF supports these types of structural definitions and relationships through the security domain.
A resource artifact in UAF represents a type of man-made object that contains no human beings. This can be viewed as a ship or an aircraft, or any other technology or software. The attack aircraft resource artifact shown in Figure 31 is assigned a constraint by the security control classifier that constrains the potentially published speed of the aircraft. This UAF classifier behaves very much like a requirements classifier from SysML. This security domain perspective can begin to compartmentalize the information on what is shareable and what may be critical to the major system component design space. Additional considerations or analysis may be necessary to determine if the information security is over constrained. With the information constraints identified, and the complete measurement sets captured, the parameterized requirements for the black box can be determined.
The model structure shown in Figure 32 demonstrates how parameterized requirement classifiers can be structured from the measurement sets assigned to the resource artifacts. Each measure has an interval with a minimum and maximum value assigned. The resource artifact of the attack aircraft defines the measurement set for the major system platform, and the previously defined major system component measures are assigned to the resource artifact of ordnance. These intervals can be mixed and matched with analytical techniques and administrative choices to define which parameters and attributes to assign to the black box.

On the UAF side, the satisfy relationship is also used between the resource artifacts and in this case, requirement classifiers from SysML. These requirements are parameterized and essentially interchangeable with SysML constraint block classifiers. Additionally, the constraints defined must satisfy the security control requirements, while also eliminating overlaps or bounding conditions present. Any changes imposed on the UAF model will reflect in the requirements space forcing updates to the black box space or inheriting new boundary conditions from the black box space.

To share specific packages from a source model like UAF, a copy relationship is used between the parameterized requirement from the measurement sets to a parameterized
requirement in the black box package. The direction of the relationship is chosen to minimize violation of the validation rules in the given tool.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Drop Speed</td>
<td>Id = &quot;8&quot;&lt;br&gt;Text = &quot;Ordinance must function between 0.3 and 1.6 Mach speed&quot;</td>
</tr>
<tr>
<td>System Ceiling</td>
<td>Id = &quot;9&quot;&lt;br&gt;Text = &quot;Ordinance must function at up to 40000 feet&quot;</td>
</tr>
<tr>
<td>System Drop Height</td>
<td>Id = &quot;13&quot;&lt;br&gt;Text = &quot;Ordinance must be dropped from a height between 25000 and 35000 feet&quot;</td>
</tr>
<tr>
<td>System Pitch at Drop</td>
<td>Id = &quot;15&quot;&lt;br&gt;Text = &quot;Ordinance must release from Attack Aircraft at an angle between -45 and 45 degrees&quot;</td>
</tr>
</tbody>
</table>

![Figure 33 Black Box Attributes from Mission Context](image)

The requirements shown in Figure 33 are fully parametrized in the SysML tool as used, enabling automated Boolean validation. This black box of parameterized requirements fulfills the mapping of attributes described in Figure 1, as well as enabling the complete federated modeling description captured in Figure 2. Thus, one of several possible methods is described on how mission envelope parameters can be shared across organizations and models while observing information security and NTK.
Figure 34 Model Relationships Through Federated Structure
The diagram shown in Figure 34 demonstrates the relationships between the federated models defined for the example project used in this discussion. This diagram can also be displayed as a more traditional graph showing the projects and project usages as nodes and edges as the definition fidelity is built up through other tools. Such visualizations of project structures can prove invaluable in navigating the content and understanding the value of each model with respect to the holistic system. In this context, the specific model can be inferred by the context of the package. This figure visualizes a portion of the proposed structure shown in Figure 2.

Concluding Remarks
The contributions of this paper include the necessary objectives to meet for enabling information exchange through model federation. The core contribution outlines how federated modeling enables stakeholders to preserve NTK while maximizing key parameters for enabling architecture trades. Given current interests and direction in MBSE and Digital Engineering at the US Federal level, it is expected that Enterprise models and other customer models will be captured in UAF or a similar DoDAF like framework for defense programs. Continued use of independent tools and SysML will require a method to integrate with these source models. The parameter-sharing method described provides a path to an enabling solution for such integration. Mapping the parameters through the combined major system platform and mission context through a black box exchange are the key contributing philosophies to tying back to the SysML system architecture models. The ability to share this information readily, as shown in Figure 1, will speed up and optimize design and procurement of major system components similar to the conventional ordnance as described, thus optimizing the major system of aircraft and ordnance for the stated mission.

Additional contributions from this work include a SysML modeling methodology, which through leveraging the pure SysML validation rules, the capabilities that may be available in the chosen tool, and the functional, logical, and physical architecture construction as described, the ability to create concise, consistent, correct, and coherent hooks to leverage the architecture against the customer’s NGOs. The proposed SysML modeling
methodology enables architecture trades, requirements verification and validation, and complete lifecycle modeling support as described in the included appendices. Additionally, this modeling methodology enables other contributing models, such as those that evaluate risk and provide failure analysis. This enables an overall hub and spoke modeling architecture resembling the visualization on page 199 of the INCOSE handbook (International Council on Systems Engineering, 2023, p. 199).

Finally, a basic approach to how to start a UAF model may be considered a supporting contribution. The core strategic, operation, and resource domains describe increasing fidelity for the NGOs of the Enterprise. Any structures or architecture captured may be further refined by the parameters of UAF, including programmatic drivers. Program or project success factors, manufacturing constraints, and security constraints will further inform the trades of the system architecture options. These potential considerations are notionally shown on Figure 2.

Future work on this topic includes defining additional possible methods for sharing the black box attributes beyond the stated SysML requirements classifiers, extending the example provided using the UAF metamodel, and demonstrating the complete analytical chain between all captured parameters to fully validate the model outputs. As new tools and capabilities are created, similar work could further extend the proposal to leverage those capabilities with respect to Figure 1 as well as better adoption of modular open systems approaches. Beyond this, additional modeling and discussion describing constraints and attributes related to program success factors can be included and expanded upon, as well as manufacturing and logistical constraints from organizations and facilities. Additional exploration of iterative modeling between program work breakdown structures and systems architecture context may help provide additional optimization to meeting the objectives of the system.
Appendix 1: Enabling System Verification

Figure 35 Proposed Federated Model for Verification Testing

Figure 35 as shown in this Appendix 1 demonstrates a supporting federated model framework for generating verification evidence against the requirements and final physical architecture. In this case, the verification requirement defines the verification method needed and the kind of verification evidence that would be satisfactory. A test activity may be used to outline all the possible test scenarios for capturing that data or related data, including existing or future diagnostic capabilities. These test activities could be used to generate NGOs that refine into source and system requirements for the systems engineering and architecting activities involved in engineering this verification evidence test.
Figure 36 Modeling How Verification Testing Traces to Design

Figure 36 begins to show some detail of how the verification evidence models will outline the experimental context including the test diagnostics, the derived system functional test requirements, and how the test report data traces back to the physical architecture trade option that is being evaluated. Using such methods may enable experienced test and diagnostics engineers and subject matter experts to iterate on what tests may provide what types of data. This better informs how to verify the listed system performance requirement with one or more verification requirements and one or more tests based on capabilities.
The final Figure 37 demonstrates the overall federated verification structure of models that support different tests, as well as models that support different lifecycles of the system. The methodology and modeling framework provides the foundation to providing verifying evidence of all the system MOPs that will eventually trace to the established system architecture that is mapped to the corresponding NGOs from the Enterprise UAF models.
A common modeling schema, applied to the outlined SysML federated modeling structure, will allow repeatable use of each framework and physical reference architecture model type to map the integrated system architecture solutions to UAF consistently. The modeling schema shown in Figure 38 is used to generate all SysML content for the modeling efforts contained in this paper and relies on NASA literature to inform this overall modeling schema (National Aeronautics and Space Administration, 2022). UAF content references the OMG UAF 1.2 literature (Object Modeling Group, 2021; Object Modeling Group, 2022) for the modeling methodology.

Figure 38 SysML Modeling Schema Derived from NASA Systems Modeling Handbook
Figure 39 demonstrates a potential table query structure based on the SysML Modeling Methodology section as well as the modeling structures shown in the prior two Appendices. This table walks through the source requirements, the functional requirements as derived from the source requirements, and the corresponding performance requirements, constraints, or NFRs that may be related to the stated function, shown in the first three columns. The table then shows the verification requirements, which describe the data needs of the verification, the verification test, which describe the methods of collecting those data, the test requirements, which become the source and system requirements for designing and executing that test, the test report, which contains the test data collected, and the link to that test data. Finally, the table traces to the physical architecture and corresponding logical architecture components that the data applies to, and any related attributes.
# Glossary

<table>
<thead>
<tr>
<th>#</th>
<th>Term</th>
<th>Description</th>
<th>Synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ResourceArtifact</td>
<td>UAF artifact that represents a type of man-made object that contains no human beings. This may include something tangible like a ship or aircraft, software, or technology.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OperationalPerformer</td>
<td>UAF artifact that represents a logical entity that is capable of performing operational activities. This artifact can produce, consume, and process resources. In this context, a ResourceArtifact implements an OperationalPerformer.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MeasurementSet</td>
<td>UAF artifact that contains a set of measurement artifacts. This artifact can be applied to artifacts from other UAF domains.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Capability</td>
<td>UAF artifact that represents an enterprise's ability to achieve a desired outcome or effect through a combination of that enterprise's means and processes.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Driver</td>
<td>UAF artifact that represents a high-level consideration motivating architecture decisions. Drivers ensure the architecture aligns with the goals of the enterprise.</td>
<td>need</td>
</tr>
<tr>
<td>6</td>
<td>Challenge</td>
<td>UAF artifact that represents an existing or potential obstacle that must be overcome to meet the motivation of the Enterprise.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Opportunity</td>
<td>UAF artifact that represents a favorable circumstance to meeting an Enterprise Challenge such that Drivers and overall motivation can be met.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Major System Platform</td>
<td>The term “major system platform” means the highest level structure of a major weapon system that is not physically mounted or installed onto a higher level structure and on which a major system component can be physically mounted or installed.</td>
<td>weapon platform</td>
</tr>
<tr>
<td>9</td>
<td>Major System Component</td>
<td>The term “major system component”—(A) means a high level subsystem or assembly, including hardware, software, or an integrated assembly of both, that can be mounted or installed on a major system platform through modular system interfaces; and (B) includes a subsystem or assembly that is likely to have additional capability requirements, is likely to change because of evolving technology or threat, is needed for interoperability, facilitates incremental deployment of capabilities, or is expected to be replaced by another major system component.</td>
<td>aircraft</td>
</tr>
<tr>
<td>10</td>
<td>Joint Direct Attack Munition</td>
<td>The Joint Direct Attack Munition (JDAM) is a low-cost, autonomously controlled, adverse weather, accurate guidance kit tailored for Air Force/Navy general purpose bombs.</td>
<td>JDAM</td>
</tr>
<tr>
<td>11</td>
<td>Bomb Live Unit</td>
<td>The explosive warhead portion of a completed bomb which generally needs additional components to activate.</td>
<td>BLU</td>
</tr>
<tr>
<td>12</td>
<td>MOSA</td>
<td>Modular Open Systems Approach</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>JDAM</td>
<td>Joint Direct Attack Munition</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>GBU</td>
<td>Guided Bomb Unit</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>BLU</td>
<td>Bomb Live Unit</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Unguided Bomb</td>
<td>An aircraft-dropped bomb that does not contain a guidance system and hence follows a ballistic trajectory.</td>
<td>free-fall bomb, iron bomb, gravity bomb, dumb bomb</td>
</tr>
<tr>
<td>17</td>
<td>Modular Open Systems Approach</td>
<td>An acquisition and design strategy consisting of a technical architecture that uses system interfaces compliant with widely supported and consensus-based standards (if available and suitable). The strategy supports a modular, loosely coupled and highly cohesive system structure that allows reversible system components at the appropriate level to be incrementally added, removed, or replaced throughout the life cycle of a system platform to afford opportunities for enhanced competition and innovation.</td>
<td>MOSA</td>
</tr>
<tr>
<td>18</td>
<td>Guided Bomb Unit</td>
<td>A precision-guided munition designed to achieve a smaller circular error probability.</td>
<td>smart bomb, guided bomb, GBU</td>
</tr>
<tr>
<td>19</td>
<td>Major System</td>
<td>A combination of elements that will function together to produce the capabilities required to fulfill a mission need. The elements may include hardware, equipment, software, or any combination thereof, but excludes construction or other improvements to real property. A system shall be considered a major system if the dollar value is estimated by the DoD component head to require an eventual total expenditure for research, development, test and evaluation (RDT&amp;E) of more than $200 million in Fiscal Year (FY) 2030 constant dollars, or if the system is designated a &quot;major system&quot; by the head of the agency responsible for the system.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Measurement</td>
<td>UAF artifact representing a unit or scale which can be used to describe the value of a property in the physical world qualitatively or quantitatively.</td>
<td></td>
</tr>
</tbody>
</table>
References


Public Web References on Munitions

https://aeqdefense.com/product/aircraft-ordnance/

https://aeqdefense.com/products/

https://aeqdefense.com/product/bomb-live-unit-blu/

https://qph.cf2.quoracdn.net/main-qimg-45f9298b7c42d521158861e8b93263ad


http://characterisationexplosiveweapons.org/studies/annex-e-mk82-aircraft-bombs/

https://cds.library.brown.edu/projects/WWII_Women/Bombs.html

https://navyaviation.tpub.com/14023/