2024 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
MODULAR OPEN SYSTEMS APPROACH TECHNICAL SESSION
AUGUST 13-15, 2024 - NOVI, MICHIGAN

THE ART OF OPEN-STANDARD SYSTEM ARCHITECTURE FOR MILITARY GROUND VEHICLES

Macam Dattathreya, PhD[1]

[1]US Army, Warren, MI

ABSTRACT

The art of open-standard system architecture (OSSA) and its usage for military ground vehicles is often misconstrued while using the Modular Open Systems Approach (MOSA) and Model Based Systems Engineering (MBSE) principles. This miscomprehension leads to the production of highly expensive, unmaintainable, and unscalable systems. This paper discusses some of the major symptoms of a misconstrued OSSA and its impacts while using MOSA and MBSE, reasons for misinterpreting, and a corrective perspective of the art of OSSA for military ground vehicles.

Citation: Macam Dattathreya, "The art of open-standard system architecture for military ground vehicles," In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 13-15, 2024.

1. INTRODUCTION

The application of the 10 U.S.C. 2446a Modular Open Systems Approach (MOSA) [1], Model Based Systems Engineering (MBSE) [2], and the US Army's Unified Data Reference Architecture (UDRA) [3] are strategically changing how the military is going to build its next generation of military ground vehicles using an open-standard system architecture (OSSA) [4] as a risk reduction method before starting production.

OSSA is an architecture development approach that reduces total ownership cost for military and allows them to field its ground vehicles faster [4]. However, developing an OSSA using MOSA and MBSE can add complexity and challenges for the traditional defense industry and the military, who may not be accustomed to building vehicles using the OSSA approach.

The OSSA is an art before science. The development of an OSSA requires a holistic system perception. However, a lack of experience and understanding OSSA with MOSA and MBSE can lead to preemptively hiring incorrect skillsets, using immature products/tools, or embracing incorrect programs of record. Additionally, these early misinformed efforts could lead to the development of many ill-conceived MOSA enabling standards, OSSA, and products with no relevance to military ground vehicles, resulting in misconstrued OSSA development and usage by all stakeholders of

DISTRIBUTION A. Approved for public release, distribution unlimited. OPSEC #: 8189

active military ground vehicle programs. This in-turn leads to the production of highly expensive, unmaintainable, and unscalable systems.

1.1. What is an OSSA enabled by MOSA?

An OSSA enabled by MOSA is a recipe for building and maintaining a system using a set of non-proprietary technical specifications, processes, and data model using a domain specific (e.g., military ground vehicles) abstract building blocks of software and hardware components of the system.

Each abstract block will have a set of cohesive functions/behaviors with precise interface definitions based on the consensus-based standards (open standards) and constrained by a domain specific data model. An interface is an interaction portal between two interacting components. Each interface will have its detailed definitions as a contract between communicating components of a system [6].

Each block will also have its predefined modular boundary, processes, and data models built according to the principles of MOSA. A modular boundary exists at its interfaces [6]. An OSSA influences higher degree of modularity or modular boundaries for system components, where a lesser number of unique system components exist to build a system. As the number of modular components decrease, there is a chance that the resulting system will be simple and easy to maintain or scale, leading to reduced dependency at the system level. Modularity is one of the principles of MOSA and is used for managing the complexity of a system. Without modularity, the complex systematic interconnections cannot be eliminated [6].

An OSSA guides system developers to construct non-proprietary and modular software and hardware systems using widely implemented consensus-based standards (open standards) rather than proprietary ones.

The functions and specifications of a system being built, described in an OSSA, allow system developers to construct modular system components with higher degree of cohesion and reduced dependency with other components. The modular components with defined modular boundaries of functions with a set of interfaces allow other components to interact with it without knowing the implementation details of the interacting components.

An OSSA reduces the complexity in synchronizing and eliminating all possible technical debts across all problem spaces by clearly articulating the modular boundaries, interface definitions, data model, specifications, functions, and the open standards to adopt.

Section 2 of this paper discusses a correct approach to the art of OSSA for military ground vehicles, and Section 3 discusses some of the major symptoms of a misconstrued OSSA and its impacts when MOSA and MBSE are employed.

2. A PERSPECTIVE ON THE ART OF OSSA

An OSSA uses a technical architecture that adopts open standards supporting a modular, loosely coupled, and highly cohesive system structure that includes the publishing of key interfaces within the system and relevant design disclosure (DAU Glossary 2020). However, an OSSA is a piece of art rather than a science. The development of an OSSA requires a holistic system approach and creativity. If OSSA is successfully developed and executed, the military will be able to utilize the resulting technical architecture in support of other vehicles within its portfolio.

An OSSA for military ground vehicles must start with a data model, functional architecture, OSSA decisions, safety and cybersecurity allocations, and modularity/modular boundaries. Then, logical/physical architectures, open interfaces, open standards, the cohesion and coupling factors of system components must be established. Then, an iterative cycle must be followed to mature the OSSA. A complete OSSA must be used to guide the design and development of a system and should be used as a risk reduction method before production.

A clear OSSA leads to the production of economical, highly maintainable, and scalable systems. The subtopics in this section will cover some of the notable elements of a clear OSSA and its usage for military ground vehicles while using MBSE and MOSA principles.

2.1. The Team Structure, a Clean Beginning

A culture change among all stakeholders is required. All SMEs should be modelers or at least know the modeling syntaxes and semantics for review purposes. All team members should know the context of the full system engineering practices as well as the DoD acquisition processes. There should be a single team using modeling as a process to build the OSSA with all subsystem/technical SMEs as a cohort. However, this is not an ideal case for all projects. Achieving this goal requires a series of targeted on-the job training sessions rather than just the typical modeling courses [7]. It is easier to train SMEs to become modelers than it is to train modelers to become SMEs.

The team should architect together with a common vision by combining all MBSE practices and SMEs/technical concerns. There should be no ambiguity in the architecture, and it should have the whole context of a system and its problem spaces.

2.2. System Architecting Approach

There are no accepted or standard definitions for top-down or bottom-up architecture development approaches. However, top-down system architecting is preferred when building complex systems and several open or military standards, and constraints must be enforced. Top-down architecting enables reasoning a complex system-level problem space via smaller, comprehendible problem spaces called subsystems or subsystem components.

To facilitate reuse and reduce schedule constraints in a complex military ground vehicle development program, bottom-up architecture process can be followed. However, without a top-down flow of system concerns/requirements in a timely manner, there is potential for incorrectly choosing an existing subsystem to solve system level problems. If bottom-up architecting is used or a combination of bottom-up and top-down approaches are chosen, then architects should be careful in ensuring none of the subsystem problem spaces are driving the system level decisions/technical choices or conflicting

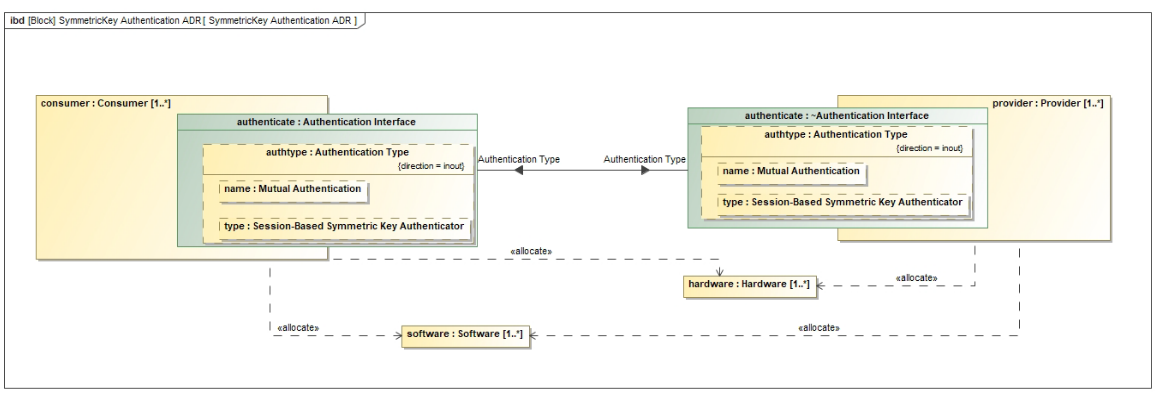

Figure 1: An example of a system architecture decision.

with the system level choices. In this situation, the development of OSSA decisions and using them to guide the union of top-down and bottom-up architecting is very important. Refer to Section 2.3 to understand more about OSSA decisions.

An architecting process must be chosen to reduce the complexity in synchronizing and eliminating all possible technical debts across the entire bottom-up problem spaces. However, in some cases, resolving issues is not easy, but using a system level vision and later adopting an existing subsystem only when necessary is prudent.

An OSSA should be developed as a digital model using best practices of MBSE. This will ensure a framework is adopted for exchanging various technical solutions, enabling the early validation and verification of models before production.

2.3. OSSA Decisions

The OSSA decisions are like the ornaments of a clear system architecture. They aid in efficiently guiding the team to mature the system architecture and its downstream implementation, especially when MBSE and MOSA principles are employed. Generally, architectural decisions influence the non-functional characteristics of a system, such as modularity, security, reliability, safety, or maintainability.

Each architectural decision describes an architecturally significant design problem that could have many potential technical solutions. The system level architecture decisions avoid each subsystem defining its own architecture choices that could amplify challenges during all subsystem integration at a system level.

Figure 1 represents an example architecture decision in an OSSA model that uses as a modeling language such as SysML, where the rationale for this example decision is to use a mutual authentication with a session-based symmetric key for all consumers and producers of a system to prevent spoofing attacks in software and hardware modules of the implemented system. This type of decision will inform all software and hardware designs to follow a common method. Without this decision in place, each software/hardware designer will likely develop independent solutions. The unique solutions will result in integration challenges and hindrances to achieve a true MOSA-enabled system.

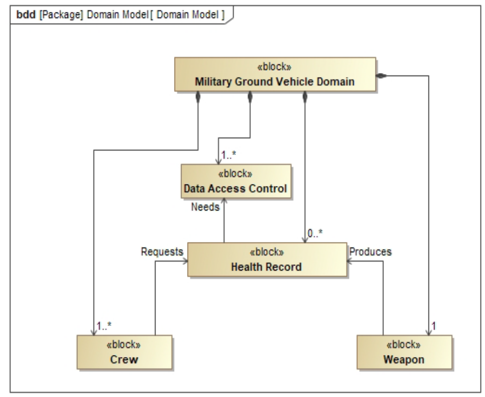

Figure 2: An example of a partial domain model of a military ground vehicle.

2.4. A Clear System Level Data Model

An OSSA should include a clear, precise, and complete domain-specific data model (DSDM) to address all data elements and their usages, while developing system component interfaces for military ground vehicles. A data model is a structure of data in various abstractions that describes the way data is organized, documented, and defined. For example, a data model description in Future Airborne Capability Environment (FACE) [7] uses three abstractions: conceptual (known as observables), logical (known as measurables), and platform as implementation specific.

A data model will establish a common vision for the data requirements across subsystems designs and the organizations participating in military ground vehicle acquisition programs. Moreover, the data model will provide a context for defining interoperable and non-proprietary interfaces between system components. The data being communicated via those interfaces will be constrained to the data model. In this way, there are no loose ends from the data perspective while building an OSSA.

A data model should be derived from the domain model of the system being architected such as a military ground vehicle. A domain model contains all the entities pertinent to the domain that must be addressed in a system architecture. Figure 2 shows an example of a partially completed domain model for a military ground vehicle.

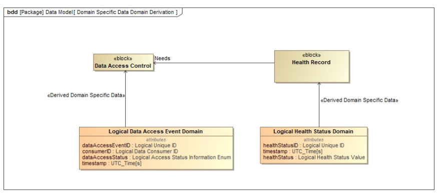

Figure 3: An example of a partial domain specific data derived from the domain model shown in Figure 2.

A DSDM provides the knowledge of the context of a system and the observable properties of the various entities in the domain model. The DSDM derived from the domain model of the system enables the system architects to define appropriate modularity contexts and modular boundaries, open standards-based interfaces, and efficient software and hardware configuration items. Moreover, they enable the successful development of logical and physical architecture elements. Figure 3 represents a partial data model derived from the domain model represented in Figure 2. It represents the '*Health Status*' and '*Data Access Event*' as domain specific data derived from the domain model at a logical level.

2.5. A Complete Functional Architecture and its Allocation to OSSA elements

An OSSA should define all the system functions in a cohesive functional architecture, consisting of subfunctions that are grouped for single, well-defined purposes.

A functional architecture should be derived from all aspects of a ground vehicle system development such as stakeholder/ performance requirements, MOSA, shared system level constraints such as vehicle infrastructure consisting of processing, networking, data sharing, safety, cybersecurity, etc., and functions derived from the applicable open standards/military standards chosen for the system development. Without these derivations, the functional architecture is incomplete. Before finalizing the OSSA, an iterative process must be followed to make sure the functional architecture is complete. Without the complete functional architecture, the system allocations will be incomplete and the whole system will be incomplete/inaccurate.

A complete functional architecture should be used to allocate system level functions to all components of a system and subsystems appropriately, resulting in cohesive components of subsystem/subsystem. This leads to correct system development and minimizing the number of issues found during testing. Completely allocated functions to system components enhances the architect's ability to derive appropriate modular boundaries or modularity, resulting in an optimal number of modular components. This reduces the maintenance footprint and increases the chances of developing scalable and economical system components.

Additionally, a complete functional architecture along with the complete data model enables the development of interoperable interfaces, providing a basis for performing critical function analysis and building a solid program protection plan for the vehicle.

2.6. Selecting Open Standards

An OSSA should be developed, first by understanding all possible system-level problem spaces, and then all applicable open standards must be selected as applicable to solve system level concerns such as interoperability, network, cybersecurity, safety, and modularity. This approach will help evolve the OSSA and standards separately by reducing the impacts of changing standards outside the context of OSSA. Preemptively selecting wrong standards adds unnecessary constraints to the design and that yields to the development of costly systems and asserting the wrong technical data.

2.7. Cybersecurity and Safety Allocations

A complete OSSA includes appropriate allocations and constraints for the system components from the perspective of cybersecurity and safety in military ground vehicles.

This allocation ensures there are no redesigns of the system components during integration. Additionally, it makes it easy to develop optimal safety and cybersecurity solutions utilizing the MOSA principles, and to perform hazard analysis effectively and determine all technical solutions to avoid hazardous behaviors of a military ground vehicle. These allocations result in an economical and reduced development cycle and reduced challenges during software safety certifications from the test commands.

2.8. Perspective on Logical and Physical system components

A complete and clear OSSA ensures all derived logical and physical components have an established traceability to the requirements, open-standards, DSDM, functional architecture.

Each logical and physical component in the OSSA should have associated behavior definition to include how a software function will be carried out.

2.9. Using OSSA in the System Development Cycle

The authors in [9-12] discuss some of the challenges and successes of OSSA in different phases of system development. However, an OSSA must be used as a guiding tool throughout the system development cycle. Having the OSSA model, facilitates the stakeholders to conduct simulations of the risk areas and uncover all fixable errors prior to production.

3. THE SYMPTOMS AND IMPACTS OF A MISCONSTRUED OSSA

Section 2 described a correct perspective of the art of OSSA and its correct usage for building military ground vehicles. This section highlights the major symptoms and impacts of a misconstrued OSSA.

3.1. Incorrect Use of MBSE

If MBSE is applied without understanding the full context, scope, and vision for the military ground vehicle system domain, the OSSA model will only contain information populated by the modelers, with no input from the subject matter expert (SME)s of the technology being used (e.g., software, safety, or cybersecurity), or the subsystem domains (e.g., mobility, vehicle electronics, or survivability). This lack of system level holistic vision and the model clarity influences subsystem designers to begin working without accounting for system level requirements or constraints. This will result in an inefficient integration of subsystems, development of non-interoperable interfaces, or the development of duplicate system level solutions.

3.2. OSSA is Used as Documentation Only

If OSSA is not used as a risk reduction method prior to production of military ground vehicles, it will only be used for documentation purposes. With a lack of clear use of OSSA from its inception, the system development programs will have large technical debts from the start i.e., increased / unknown future costs, increased likelihood of schedule slip to delivering a working vehicle, and increased burden in synchronizing efforts to eliminate technical debt across the entire subsystem space simultaneously.

If OSSA is used as documentation, the focus will be more on the modeling aspect and not the architecture. These models will not reliably reflect the concerns from subsystem domain experts such as survivability, mobility, fire control, etc.; nor will the models reliably convey the concerns from technical subject matter experts such as software, vehicle electronics, cyber, or safety engineering areas. This is an early indicator that the individual subsystems such as mobility, vehicle electronics, or survivability are designed in silos with no system context from the beginning. This deficiency leads to unexpected redesigns, misunderstood requirements, poorly defined modularity scope, and bad design choices at the subsystem or its component level.

3.3. Missing OSSA Decisions

If top-down holistic system level OSSA decisions are not used to guide the development of OSSA and its usage for building military ground vehicles, the whole system design will be developed without a shared vision, which leads to inefficient and unmaintainable system designs.

The lack of sufficient holistic OSSA decisions will hinder the employment of holistic system architecture for the system being built, especially when the best principles of MOSA, MBSE, and UDRA need to be applied from the inception of the development program. The scalability, maintainability, and cost of building an efficient system depends on well thought architecture decisions. Without those holistic system level OSSA decisions, it is difficult to guide the development of logical and physical architectural elements with the properly allocated constraints and system level functions. Without the system level architecture decisions, each subsystem design will drive a system architecture which leads to complex, costly, and pro-longed integration when all the subsystems must be synchronized at the system level. Therefore, it will be difficult to have a holistic convergence across many duplicate subsystem specific designs, incompatible designs and interfaces, integration challenges with no enforcement of MOSA and effective MBSE practices.

3.4. Lack of a Data Model

A data model normally defines all data used by the system using conceptual (observable), logical (measurable) and physical (implementation) abstractions. For example, a vehicle’s speed is defined as observable data, meters/second is defined as a measure of vehicle speed data, and ‘double (64-bit precision data)’ is used as an implementation (software) data type for indicating vehicle speed in meters/second. Refer to Section 2.4 for more details about the data model.

If a data model is not present in the OSSA, the system will not have clear definitions for the data being used by military ground vehicles. For example, if the data model does not exist, then the vehicle speed observable data may not be defined, its measurable information as meters/second may not be identified, and its physical abstraction, such as ‘double,’ may not be assigned. This will lead to the development of incompatible interfaces, resulting in unnecessary interfaces or adapter capability developments.

Lack of a clear data model upfront or employing a data model at a later stage of the system architecture development complicates the interface designs of the system components and the development of inefficient software architectures. A lack of upfront data model introduces interoperability issues between the system components. Moreover, it severely restricts the flexibility in building modular open systems with non-proprietary (open) and interoperable interfaces for the system and subsystem components.

3.5. Lack of Functional Architecture and its Allocations

A functional architecture defines how the system functions will operate together to perform the system mission(s) [7]. For example, '*Detect Target*' and '*Fire Weapon*' functions will work together to achieve a mission of '*Destroy Target*'. A cohesive function will consist of subfunctions which are grouped for a single, well-defined purpose. For example, the subfunctions of the '*Detect Target*' function consists of all functions related to detecting a target such as '*Sense Target*' and '*Identify Target.*"

If a functional architecture is not present in an OSSA or is not cohesive, then the required system level functions may not be allocated to all system components appropriately. This leads to incorrect system development and the issues will be found during testing, when it is too late to fix errors without prohibitively significant cost and program delays.

When an OSSA is misconstrued, a functional architecture may be derived purely from the stakeholder or performance requirements with no additional derived functions related to common constraints such as infrastructure, safety, cybersecurity, MOSA, and functions that could be derived from the chosen open standards. This lack of correct functional architecture influences the development of systems with reduced cohesion for system components, inaccurate derivation of modularity/modular boundaries for the components, development of unnecessary interfaces in the system components, and a lack of completeness for the system being built. Additionally, it will be challenging to perform critical function analysis and build a solid program protection plan for the vehicle.

3.6. Lower Degree of Modularity

A misconstrued OSSA will have a lower degree of modularity or incorrect modular boundaries for system components/ subsystems, where a greater number of unique system components/subsystems exist to build a system. As the number of modular components increase, there is a chance that the resulting system will be too complex and difficult to maintain or scale, leading to increased dependency at the system level.

3.7. Lack of System level Safety and Cybersecurity Allocations

A misconstrued OSSA for military ground vehicles will have no system level safety and cybersecurity functions allocated for its system components/subsystems. This lack of allocations influences the redesign of the system components during integration. Additionally, it makes it difficult to develop optimal safety and cybersecurity solutions utilizing MOSA principles. The lack of allocations also leads to increased cost and prolonged process for obtaining software safety certifications from the test commands.

3.8. Lack of Synchronization of Top-down and bottom-up architecting approaches

The presence of a poorly synchronized combination of top-down and bottom-up architecting approaches–where each subsystem/system component design determines the system level OSSA–is a symptom of misconstrued OSSA, which will tend to lead to the development of systems that are not maintainable or scalable. Moreover, it results in a lower degree of modularity for the system components.

3.9. Preemptive Open standards selection and flow down as requirements

An open standard is preemptively chosen for an OSSA development, and a full standard is used as a requirement for subsystems/system components design without first analyzing system level concerns and then allocating appropriate lower-level requirements. This method will increase the development of duplicate solutions. Moreover, the employment of unnecessary open standards creates unwanted constraints and complexity in the system design. In some cases, some open standards are used for no reasons, which adds unnecessary cost and schedule risks.

4. CONCLUSION

Clear OSSA with architecture decisions will enable the Army to build highly scalable and cost-effective military ground vehicles. A correct OSSA reduces the complexity in the system development lifecycle with the accurate employment of shared vision, data model, functional architecture, open-standard based interfaces, modularity/modular boundaries, and well-allocated safety and cyber strategies at the system level throughout the development process.

5. DISCLAIMER

Reference herein to any specific commercial company, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the Department of the Army (DoA). The opinions of the authors- expressed herein do not necessarily state or reflect those of the United States Government or the DoA and shall not be used for advertising or product endorsement purposes.

6. REFERENCES

[1] DSP MOSA, online: https://www.dsp.dla.mil/Programs/MOSA/

[2] "Model-based systems engineering," Wikipedia, May 23, 2022. https://en.wikipedia.org/wiki/Model-based_systems_engineering

[3] E. Christ, "PEO EIS collaborates with DASA(DES) on turning army enterprise data mesh into a reality," online: https://www.eis.army.mil/newsroom/news/peo-eis-wide-data/peo-eis-collaborates-dasades-turning-army-enterprise-data-mesh

[4] AcqNotes, online: https://acqnotes.com/acqnote/careerfields/open-standard-system-architecture

[5] AcqNotes, online: https://acqnotes.com/acqnote/careerfields/functional-architecture

[6] Macam Dattathreya, "Verification of an architecture in a system model using domain-specific operational scenarios and contexts," In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 15-17, 2023

[7] M. Dattathreya, COL Jurand, S. Dawson, "Digital Engineering Learning Curve," Army AL &T magazine, Fall 2023, pages 76-80.

[8] The Open Group, "FACE Technical Standard Edition 3.1," online: https://publications.opengroup.org/standards/face/c207

[9] E. Rechtin, M.W. Maier, "The art of systems architecting," CRC press, 2010

[10] E. Bennett, M. Machart, "The Unwritten Truths of Military Ground Vehicle Architecture," In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 15-17,2023.

[11] M.A. Kurfman, R.B. Stone, M.V. Wie, K.L. Wood, K. N. Otto, "Theoretical underpinning of functional modeling: Preliminary experimental studies," 2000 ASME Design Engineering Technical Conferences, Maryland, 2000.

[12] R. Peters, B. McDonald, R. Mikola, M. Dattathreya, "Model Based Approaches to Systems Implementing Modular Open System Approach (MOSA)," In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 16-18, 2022.