AN ARCHITECTURAL DESCRIPTION FOR THE APPLICATION OF MBSE IN COMPLEX SYSTEMS

Adebowale Odukoya,
Department of Design, Manufacturing & Engineering Management,
University of Strathclyde.
Glasgow, United Kingdom.
ade.odukoya@strath.ac.uk

Dr. Ian Whitfield,
Department of Design, Manufacturing & Engineering Management,
University of Strathclyde,
Glasgow, United Kingdom.
ian.whitfield@strath.ac.uk

Dr. Laura Hay,
Department of Design, Manufacturing & Engineering Management,
University of Strathclyde,
Glasgow, United Kingdom.
laura.hay@strath.ac.uk

Neil Harrison
Department of Research and Technology
BAE Systems Naval Ships
Glasgow, United Kingdom
neil.a.harrison@baesystems.com

Dr. Malcolm Robb
Department of Research and Technology,
BAE Systems Naval Ships,
Glasgow, United Kingdom.
malcolm.robb@baesystems.com

Abstract— The design of a complex warship is a multi-disciplinary effort which often encounters major challenges, particularly with respect to integration across interfaces in the System of Systems (SoS). In principle, the goal of Model Based Systems Engineering (MBSE) with respect to system design is to provide a means of capturing and communicating the system design in a structured, consistent, and coherent fashion; that can be easily assessed by engineering teams and quickly analysed using queries and toolsets. The focus of this paper is to investigate the potential to achieve a consistent description, identify a viable methodology that minimises mismatch in requirements and to avoid an extended design lifecycle. This study highlights the need to develop a generic Architectural Description (AD) that is based on a common ontology which would clearly define the fundamental tenets of applying state-of-the-art Architectural Frameworks (AFs) in naval ship design. An investigation on the effectiveness and accuracy of a graph-based approach is needed to assess whether it is possible to create a 'Rosetta stone' for AFs, which links any two or more different model viewpoints in different AF's using the approach.

Keywords— Systems architecture, System of systems, Complex systems

I. INTRODUCTION

The design of a complex warship is a multi-disciplinary effort which often encounters major challenges, particularly with respect to integration across interfaces in the System of Systems (SoS). In practice, Model Based Systems Engineering (MBSE) is executed with respect to the needs of immediate tasks and constraints. In principle, the goal of MBSE with respect to system design is to provide a means of capturing and communicating the system design in a structured, consistent, and coherent fashion; that can be easily assessed by engineering teams and quickly analysed using data queries and software tools. To realise the main benefit of MBSE within the system; ease of access to information that supports the identification of structural and behavioural issues within the design: the product lifecycle must be planned effectively. MBSE will help to tackle complexities that arise because of integrating system interfaces across the SoS. Through effective planning, there should be a flow of information that is: simple to follow; rigorous in its complexity management; rapidly identifies any enterprise and product system interface problems that arise; and enables easy interchange of information across the SoS lifecycle irrespective of toolsets. Information relating to the enterprise system and product system are interconnected, this is captured within the SoS 'model' which reflects the state space and maturity levels of the system development. Definition rigor is manifested through the Architectural Framework (AF), language, abstraction, and automation toolset in a SoS. People also play a critical role in the engineering of systems and integration of interfaces in the SoS, interacting with other elements (descriptions, relationships and behavioural attributes) in the system through the language that is used. Ultimately, value is added on the path to developing a mature model as the system analysis will be more accurate.

Undesirable outcomes often stem from ontological mismatches in the language-based description that is used. One source of such mismatch is the implementation and emergence of different Architectural Frameworks (AFs) with inconsistent viewpoints and data structures, which make it difficult to compare or integrate the constituent information in a meaningful and coherent way. The adoption of different AFs in the same domain and across different domains within industrial entities, results in significant challenges during collaboration efforts as data is not easily transferable across different AFs. Progress has been made to facilitate compatibility and aid the transfer of 'meaningful' data through The Unified Profile for DoDAF and MODAF (UPDM). Furthermore, the Object Management Group's Unified Architecture Framework (UAF) is a superset MetaModel which defines over 70 viewpoints; evolved from the UPDM v2.1 and is compatible with the NATO Architecture Framework (NAF) which is a subset of the UAF Domain MetaModel (DMM v1.0).

The need to maintain coherence is key: the use of different language must be compatible across interacting elements within the system, to maximise consistency, remove laborious and error-prone translations and minimise opportunities for rework to be designed in. A common ontology for the entire

SE domain would accelerate end-to-end operations, resulting in delivery of capabilities at an accelerated pace. Current state-of-the-art AFs such as the UAF and NAF only partially address this. Therefore, exploring a pragmatic alternative such as a formal language to achieve linguistic rigor and clarity when describing the state spaces and processes would be a valuable contribution.

The creation of an extensible Architectural Description (AD) to interoperate AFs will be beneficial for the implementation and integration of large MBSE efforts. An AD would provide a basis to unite multiple AFs. This means the ontology would be compliant and traceable with ISO/IEC/ IEEE 42010:2011 (Systems and engineering and software engineering – Architecture description) ontology, and hence offer a shared foundation and means to translate data-sets across metamodels and the respective ontologies. A standalone generic ontology that is compatible with all AFs and presented in an easily realisable way alongside those descriptions. One way that could work would be that the AD highlights areas of potential ontological mismatch between types of description being used in the system development. It would also facilitate coherence in the ontology across AFs.

Through the consideration of state-of-the-art AFs such as The Open Group Architecture Framework (TOGAF), NATO Architectural Framework (NAF), Unified Architectural Framework (UAF), the AD will be based on a generic ontology for MBSE, ultimately aiming to be adaptable to defence AFs, and coherent with MBSE standards and best practices. An AD requires the key tenets of MBSE to be identified to ensure that system development is coherent with the standard for life cycle processes ISO/IEC/IEEE 15288:2015 (technical processes) and pre-existing frameworks.

This study clarifies the call to develop a generic AD and clearly define the fundamental tenets of applying state-of-the-art AFs in naval ship design The focus is to investigate the potential to achieve a consistent description. It is essential to minimise the mismatch in requirements early and avoid an extended design lifecycle.

In the following sections the purpose and value added to doing MBSE by way of an ontology-based approach for an AD will be discussed with respect to literature and current practices in the defence industry. Section 2 discusses the need for MBSE in complex systems. In section 3 emergent properties within complex systems and the importance of an agile methodology that aids preparedness to respond to them is discussed. Section 4 presents the value an ontology-based approach can add to the SE domain. Section 5 offers a brief overview of the historical background of AFs and considers AFs used within the defence industry. Section 6 clarifies the call for an ontology-based methodology for MBSE that can be used alongside AFs of complex SoS. Section 7 summarises the purpose and future direction of this research.

II.   THE NEED FOR MBSE IN COMPLEX SYSTEMS

The need for a proper systems engineering approach stems from the habit of systems engineers for "stating the problem in terms of the solution" [1], resulting in a frenzy of solution-based problems because of technology being "sought to solve the problem before it is stated"; rather than problem-based solutions. The myriad of interrelationships and interconnections among components within a constituent system and SoS makes MBSE complex. "Systems

Engineering is a transdisciplinary and integrative approach to enable the successful realisation, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods." [2].

The three evils of systems engineering are considered to be: communication problems; lack of understanding; and complexity. Literature indicates the key factors of a system are: politics, people, finances, structures, environment and things [3]. The term 'Systems Engineering' was coined by Bell Telephone laboratories in the 1940s. Primarily; the term has been applied to complex systems with elements possessing different behavioural characteristic to that of the holistic system. Contributions to the development of technological models of systems began in 1896 with Karol Adamieki, through the creation of Gantt charts; and later formalised by Gantt in 1910. Thompson Ramo Wooldridge (TRW) incorporated then developed system models through functional flow block diagrams (FFBD) in the 1950s. Further variations to the FFBD were developed and used by NASA in the 1960s. Yourdon then contributed to the development of models by popularising data flow diagrams in the 1970s. The development of models in the software and systems engineering domain resulted in hundreds of modelling notations, variations of data flow diagrams such as Harel's state charts and the use of object-oriented techniques. This led to the formation of the OMG group in late 1990s in an effort to create a unified software notation known as the Unified Modelling Language (UML); later becoming the industry standard, ISO 19501, in 2005, [4]. Subsequently, the Systems Modelling Language (SysML) was developed and released in 2007 as the first modelling notation tailored to systems engineering.
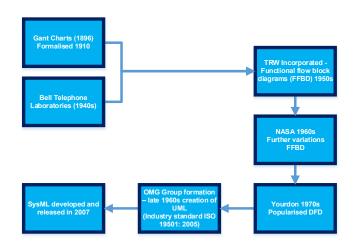

Fig. 1.   History of Technological Models

The expansion of SE to incorporate a robust understanding of the SoS lifecycle and process, for an architecture that adequately captures the breakdown structure regarding functional, physical and operational characteristics of a warship is long overdue [5], [6]. Vaneman and Carlson, [7] stated that value can be added through properly defined models that deliver an accurate virtual representation of enterprise, functional, physical, and parametric system entities. Similarly, Cherfa *et al.* [8] posited that strong links between the architecting and SoS analysis stage are required. The SE community is eager for a development framework that

is both data and model centric, which benefits from defined functional, operational, physical and logical architectures.
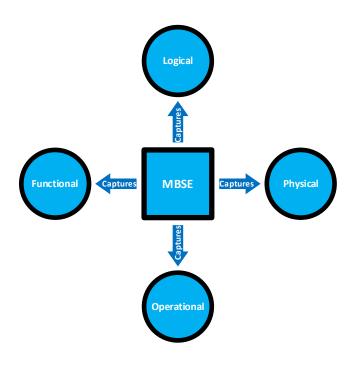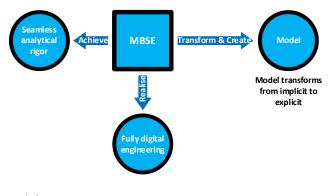

Fig. 2.   Attributes of an Architecture

The modern warship is one of the most complex, integrated human-made systems [9]-[11], which is increasing in complexity due to the vast number of capabilities required and the speed that software-based technology, artificial intelligence, electrification and military capability needs are evolving. As complexity increases, so does unplanned emergent behaviour within the programme, and risk of failure in a SoS. The capabilities required from a warship range from operating as an autonomous system, as part of a task force, endurance of various climatic environments, performing humanitarian missions, self-sufficiency, providing sustenance and a home for its sailors [6], [9]. Wymore is cited as one of the first to coin the term Model Based Systems Engineering [1]. Yourdon and Wymore, presented the coherent concept of models and their applications in the software and systems engineering domain. The International Council of Systems Engineering (INCOSE) followed suit with major contributions through the launch of the MBSE initiative in 1996, and emergence of the following key methodologies: Rational Unified Process (RUP) in 2004; the Object-Oriented Systems Engineering methodology (OOSEM) in 2014 by the Chesapeake chapter; and, the Vitech methodology 2007.

As evidenced in the preceding paragraph, MBSE has been practiced for decades. In 2007, INCOSE launched an MBSE initiative to 'promote, advance and institutionalise the practice of MBSE' [12]. Model-based processes are 'one of the most widely-discussed issues within the Department of Defence (DoD) today' [7]. MBSE is evolving and is considered in the INCOSE 2025 Vision to be a key component to add value to complex systems, by maintaining design traceability across the SoS; as well as 'manage design complexity including architecture, requirements, interfaces and test vectors'[13].

MBSE is defined as "the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [12]. MBSE can be described as the formalized application of modelling principles, methods, languages, and tools to the entire lifecycle of large, complex, interdisciplinary, sociotechnical systems [6], [14]. As a fundamental principle of good system design; the essence of MBSE is the application of appropriate formal models to a domain. MBSE is therefore indispensable to fully realise digital engineering, transform the engineering model from implicit to explicit, and achieve seamless analytical rigor in system models as demonstrated in Figure 3 below. The vision is to shift the design of complex systems from implicit models currently used by systems engineers in cognitive and natural language realms, to explicit formalised models with ontological tractable bases. Specifically, obtain a model-based digital representation that enables digital analysis and reasoning of systems independent of physical

prototypes.

Fig. 3.   Purpose of MBSE

MBSE involves building models and supports communication, visualisation, validation and verification during the lifecycle of a project [15]. When considering systems engineering and the engineering point of view, models will describe the: behaviour, functionality, requirements, activities, mission capability, structure, and interconnectivity between elements in the SoS as illustrated in Figure 4. MBSE enables multi-disciplinary system integration, continuous monitoring of changes and their effect with respect to the development of diverse interdependent systems that provide complex functions, and the assurance of confidential exchange of information. MBSE provides a structured framework and mitigates the risks posed by traditional design lifecycle, giving systems engineers the means to manage the functional, physical and operational requirements in a cohesive way.
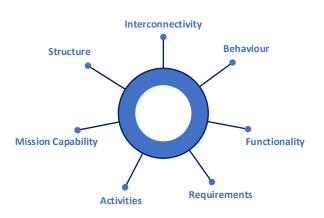
Fig. 4.   Properties described in a model

III.   EMERGENT BEHAVIOUR

Barat et al. stated: "The inherent characteristics of modern organisations that include the socio-technical characteristics, complex and dynamic organisational structure, significant uncertainty, and emergent behaviour make the decision-making process a complex endeavour" [16].

Ship 'design' is a noteworthy example of a wicked problem [17]-[20]. Wicked problems in the design process exhibit: complexity, uncertainty, unstable requirements and environments, value conflicts, dependence on social and cognitive abilities of humans [21], [22]. These are problems that are non-deterministic prior to designing or realising the solution. Consequently, the design process is unavoidably iterative as the solution development cannot be separated from the system requirement elicitation and refinement, [17], [19]. The only properties that are realised in a system are desirable and undesirable behaviour. Emergent properties of a system refer to the idea that there are multiple nodes in a system working in concert to create either intended or unintended results. In order to give a system desired characteristic, it is important to design for and mitigate pre-existing undesirable emergent behaviours for both the developed system, and the system being developed. While being prepared to respond as new emergence occurs.

In efforts to improve the development of a complicated system such as a warship, with planned and unplanned emergent behaviour; it is paramount that the design problems encountered in the systems development are understood, as well as all the significant effects due to iterative changes. As ship design is a wicked problem, it is not possible to understand potential issues that may emerge until they are realised. Therefore, an agile approach which deals with problems as they arise should be taken in order to minimise negative impact to the progress of system maturity. Desirable and undesirable behaviour emerge in the design process and the physical product. The occurrence of unplanned emergent behaviour is dependent on the degree of consistency between the planned and actual behaviour of the system with respect to system requirements. The aim is to design out undesirable emergent behaviour in the product, through identifying inaccurate and unattainable requirements early.

The capacity to view the system from multiple viewpoints builds a holistic view of the SoS. Measuring parameters which indicate convergence or divergence with respect to system requirements for Engineered to Order (ETO) systems poses major challenges within the SE domain. Ideally, the majority of desirable and undesirable characteristics should be realised through performance modelling and simulation. However, some characteristics do not materialise until the system realisation phase of the lifecycle, which makes a complex system such as the modern warship vulnerable to cascading failure [23]. The advancement of digitalisation presents significant opportunities to more comprehensively verify that predicted system behaviour falls inside the parameters of the requirements. The journey to optimal convergence of a system model with the requirements requires a degree of divergence from the current processes of 'doing' engineering.

IV.   ONTOLOGY BASED APPROACH

An ontology is an artefact (meta-model) and a technique (tool) in the software/ systems life cycle. The structure of the modelling language is defined by the meta-model while the tool captures the semantics. Yang, Cormican and Yu, [24] defined ontology as "a formal, explicit specification of a shared conceptualization". The key function of ontology is to support the representation and management of knowledge. Efforts in developing a computable knowledge base responds to the need "to guide transdisciplinary collaboration across several organizational entities working on the same SoS" [25] and model a domain of knowledge. Ontology Based Systems Engineering (OBSE) focuses on using ontologies to enable the betterment of the SE body of knowledge; a clear roadmap of the usefulness of ontologies and their application in SE has been presented by Yang, Cormican and Yu, [24].

The emergence of undesirable behaviour such as ambiguity – lack of precision and co-operation during the product development process – occurs because of the absence of a generic and universal ontology. An ontology that would underpin both concepts and terminology [26], [27] while serving as a domain knowledge representation tool [24] would mitigate ambiguities. Vaneman and Carlson, [7] concurred that the system design should be expressed in a manner that is succinct, clear, and semantically consistent. This is further reiterated by Yang, Cormican and Yu, [24] whose study findings indicated that a consistent ontology will eliminate miscommunication amongst stakeholders; while providing traceable requirements. This would subsequently facilitate effective and efficient communication amongst stakeholders with varied skills and competency levels through reusable, sharable and explicit knowledge.

Literature confirms that an ontology-based approach could address inconsistencies in and between various pockets of system description, which often are the root cause of undesirable consequences such as extended project delivery timelines, increased cost and rework. The development of a coherent knowledge space aims to tackle ontological inconsistencies . The SE community has made significant progress to facilitate compatibility and aid the transfer of 'meaningful' data evidenced by the UPDM. OMGs UAF is a superset MetaModel which evolved from the UPDM v2.1 and is compatible with the NATO Architecture Framework (NAF) which is a subset of the UAF Domain MetaModel (DMM v1.0).

Current state-of-the-art AFs are being gradually superseded, with some being made obsolete, as evidenced by the UK Ministry of Defence Architecture Framework (MODAF). Although some AFs (such as MODAF) are

becoming obsolete, systems description within their respective conventions should be leveraged and interoperable with other AFs as the similarities between current state-of the art AFs are more obvious. To a certain extent these similarities have been combined in the UAF, with TOGAF as a notable exception. Although the fundamental principles of SoS (INCOSE) and AF standards, [28] are accessible; to date there is very little guidance on the application of ontologies in the SE domain and their respective concepts [27]. Therefore, a ontological interoperability across multiple AFs is a must in the SE domain.

However, gaps remain with respect to how information between two AFs can be merged into a single knowledge base. All AFs are developed to serve different purposes and disciplines with different taxonomies for defining their system as shown in the Survey of Architecture Frameworks [29]. Therefore, there would be benefit in the application of an ontology-centric approach to identify common elements between the architectures, towards merging information in different AF conventions into a single knowledge base. A potentially promising evolution is toward ontological interoperability across multiple AFs.

Methods for handling the interpretation of taxonomies across different AFs could be realised through the world of graph theory and semantic data. For example, the Computer Aided Engineering for Systems Architecture (CAESAR) initiative-: OpenCAESAR approach is a process framework and suggested software toolchain to enable an integrated model-centric approach to SE. Based on semantic web vocabularies and expresses information in the Ontology Modelling Language (OML) which is tool-neutral. OML is based on the Web Ontology Language 2 (OWL2). This approach could serve as an experimental environment to explore and study the real effect of translating multiple architecting frameworks to a single ontologically consistent system expression in the Ontological Modelling Language (OML).

An AD based on a common ontology will ultimately be beneficial for the implementation and integration of MBSE. Through the extension of an AD which initiates convergence across state-of the art AF's, information from AF's could be extracted allowing for a natural and continuous evolution of the AF's. Ultimately creating a generic AD that is not specific to any one AF (TOGAF, NAF, UAF, etc.) is proposed as the most viable way forward as current state-of the art AFs will be continuously succeeded and become obsolete, shifting the focus from identifying differences in ontology/metamodel to a graph-based ontological approach. The aim would be to have a digital twin of the system, from which information sets can be easily injected into the AF that any system stakeholder uses.

V. ARCHITECTURAL FRAMEWORKS

Historically, SE has broadly been considered and applied throughout the design of a system, while embracing both subjective and objective methods. An architecture is defined as the "fundamental concepts of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" [28]. The architecture includes principles associated with realising system capabilities, provides guidance for information authoring and governance, establishes context, captures key business concepts, and is a mode of communication between stakeholders. The

functional, physical, operational and logical characteristics of a system are represented through the system architecture viewpoints. Characteristics typically represented in an AF are structure (what it is), interfaces (what it interacts with), and behaviour (what it does) at different levels of abstraction (logical and physical) [23]. These viewpoints are supported by an underlying metamodel. The guiding principles for architecting are modularity "functional, technical and physical views of product development" [30], high cohesion, and loose coupling, [31].

ADs and AFs are assets with the purpose of codifying "conventions, common practices of architecting and the description of architectures within different communities and domains of application" [28].

The purpose of the architecture is to "assist the understanding of the system's essence and key properties pertaining to its behaviour, composition and evolution" [28]. An AF is defined as: "conventions, principles and practices for the description of architectures established within a specific domain of application and/ or community of stakeholders" [28]. AFs allow for a consistent and coherent way to communicate the architecture of a system to systems engineering architects, while aiding integration across system interfaces, controlling the design, and defining the problem domain. An AD is defined as a "work product used to express an architecture" [28] and it provides a conceptualisation of the system architecture.

An AF ensures the AD is useful and acts as a tool for managing abstractions in the system, while promoting the 6C quality goals in the SoS detailed below:

1. Correctness: semantic and syntactic correctness (correctly represents architecture)
2. Completeness: define all viewpoints
3. Consistency: agreement between statements
4. Comprehensibility: single well-defined ontology and alignment of viewpoints
5. Confinement: information hiding and context specific
6. Changeability: low coupling and high cohesion – dividing architecture into views makes it modular

ISO/IEC/IEEE 42010, [28] addresses ADs and AFs; by facilitating analysis and evaluation of communication and cooperation of other architectures, used in managing modern systems [32].

The nature of AFs and the initial divergence (emergence of multiple AFs with different viewpoints and data structures) across different industries within the SE community made it difficult for system engineers to compare and integrate architectures in an efficient way. AFs used in the defence industry are: TOGAF, DoDAF, NAF and UAF. AFs specifically tailored to serve Enterprises are: Zachman and TOGAF. As each AF is developed to serve a different purpose within different disciplines; they are defined using different ontologies/ metamodels. The differing viewpoints and ontologies make collaboration efforts more difficult owing to the lack of methods and tools to amalgamate and compare information sets. Furthermore, although TOGAF and NAF include an ADM which provides guidance on how to build the system architecture, other AFs listed in the survey of Architecture Frameworks [28] do not. The variable amount of guidance for the application of AFs means engineers

improvise some unique aspects with respect to applying the AFs to their systems. Naturally this causes increased challenges when attempting to collaborate with other disciplines as domain specific taxonomies arise in the architecture.

The respective dates, types, viewpoints, and descriptions of commonly used within AFs are presented in Table 1. As noted previously, progress has been made to facilitate compatibility and aid the transfer of 'meaningful' data through the progressive convergence of AFs evidenced by the UPDM (from DoDAF and MODAF), UAF (from DoDAF and UPDM) and NAFv4 (from UAF and MODAF). This is further evidenced as NAFv4 is a DMM of the UAF v1.0 .

TABLE I. ENTERPRISE AND DEFENCE ARCHITECTURAL FRAMEWORK

Architecture Framework	When	Last Updated	Type	Viewpoints	Description	Evolved From
Zachman	1987	1987	Enterprise Architecture	• 36 viewpoints • 6 by 6 matrix (Perspectives and Aspects) • Perspectives: Executive, Business management, Architect, Engineer, Technician, Enterprise • Aspects: What, how, where, who, when, why	• Most mature, oldest, and widely used AF • Complete and not customisable	-
TOGAF	1995	2018 (TOGAF v9.2)	Enterprise – IT methodology	• 9 phases; multiple steps within each phase • Layers: Business, Data, Application, and Technology • Continuous requirement management activity	• First AF with an architecture development method (ADM).	Zachman (1987) TAFIM (1996)
DoDAF	2003	2010 (DoDAF v2)	US Department of Defense AF	• 8 viewpoints • Viewpoints: All, Data information, Standards, Capability, Operational, Services, Systems, Project	• Concepts do not align with ISO 420100	C4ISR AF (see US DOD, 2020)
MODAF	2005	2010 (MODAF v1.2.004)	UK Ministry of Defence AF	• 7 viewpoints • Viewpoints: All, technical, strategic, operational, service, system, acquisition	• Obsolete – Superseded by NAF V4 • Most divergent from ISO/IEC/ IEEE 42010:2011	DoDAF V1.0
UAF	2017	2020 (UAF v1.1)	Enterprise, Defence, General Purpose AF	• OVER 70 Viewpoints • 13 Domains and 11 model kinds • Domains: Metadata, strategic, operational, services, personnel, resources, security, projects, standards • Model kinds: Taxonomy, structure, connectivity, processes, states, interaction scenarios, information, parameters, constraints, roadmap, traceability	• Superset of potential viewpoints • User can select an appropriate subset. • DoDAF v2.02, MODAF v.13, and NAF v3.1 & v4 compliant ADs can be constructed • Applicable to all SoS	UPDM v2.1 for DoDAF and MODAF
NAF	2004	2018 (NAF v4)	Enterprise level, Business and Defence	• 46 Viewpoints • 5 subjects and 9 Aspects • Subjects: Concepts, Service Specifications, Logical Specifications, Physical Resource Specifications, Architecture Meta-Data • Aspects: Taxonomy, Structure, Connectivity, Processes, States, Sequences, Information, Constraints, Roadmap	• Subset of UAF • NAFv4 compliant with *ISO/IEC/IEEE 42010*, 2011 • NAFv4: 2018 has an ADM (based on TOGAF ADM • Clear methodology guidance in line with TOGAF ADM. . • Latest version uses third party metamodels: UAF Domain metamodel (DMM) and ArchiMate metamodel.	DoDAF MODAF v1.2.004

VI. DISCUSSION

Desirable and undesirable behaviour emerge in the design process and the physical product.

An ontology-based approach may address the cause of undesirable emergent behaviour. The causes are discussed throughout this paper, one of them being: data is not easily transferable across AFs due to inconsistent viewpoints and data structures both throughout multidisciplinary teams within and across different organisations using different AFs. It is acknowledged that the SE community has made significant progress to facilitate consistency, compatibility and aid the transfer of 'meaningful' data across AFs; however, a coherent knowledge space that provides ontological consistency could be a powerful tool to address mismatches in agreements in the SoS.

When developing an architecture, due diligence must be given to ensure that it fits its intended purpose and gives guidance to stakeholders with respect to proper application. Hence, the architecture must be tailored to capture information that is pertinent to the needs of the stakeholders. Exploring the effectiveness of an AD based on a common ontology will ultimately be beneficial for the implementation and integration of MBSE. Through the extension of an AD which initiates convergence across state-of the art AF's, information from AF's can be extracted and allows for a natural and continuous evolution of the AFs. Ultimately creating a generic AD that is not specific to any one AF (TOGAF, NAF, UAF, etc.) is the most viable way forward as current state-of the art AFs will be continuously succeeded and become obsolete.

An investigation on the effectiveness and accuracy of a semantic web ontology approach is needed to assess whether

it is possible to create a 'Rosetta stone' for AFs, which links any two or more different model viewpoints in different AF's through translation. The AD is intended to be a tool to manage and achieve real-world outcomes. Joining up the collective power of the systems engineering discipline by equipping ourselves to overcome description discontinuities that have been introduced by multiple architecting frameworks, and modelling conventions.

The starting point in efforts to create a generic AD is to identify the stack of semantic ontologies that underpins state-of-the-art AEs and specialise them for every discipline. This requires tracing back to the root of the vocabulary stack and establishing the foundational design pattern (semantic ontology which defines classes and associations) of the AF. The foundational design pattern defines the architectural structure, semantics and syntax. In order to be able to specialise the AF for every discipline through extensions to the ontological stack, a formalised methodology, ontology and architectural development process must be defined. The business activities and the formalised process must be agreed upon by stakeholders to ensure the formalised process is achievable on a day-to-day basis. Understanding the extent that people adhere to formalisations and the impact it has on consistency across AFs would empower an organisation, as measures could be put in place to reinforce adherence to formalisations as well as identifying the root cause of undesirable emergent behaviour. This knowledge will also be reflected in the model.

There is also a need for a fully realised implementation method to investigate intra-consistency between AFs. This could be addressed by identifying the link between ontology stacks across AFs, and ontological satisfiability within each respective stack. The ontology stack should be assessed to check the following requirements are satisfied:

1. Formalised to an operational level

2. Statements consistent with vocabulary

3. Formalisation is representative of full methodology

4. Embodied system descriptions compliant with framework.

While inter-consistency checks the AF is consistent and satisfies description logic.

The ability to spot inconsistencies would reduce cost and time spent on the project as reworks would decrease as a result. Because of the subtle difference in coherence and consistency, it can be difficult to determine whether emergence of undesirable behaviour is due to ontological incoherence or inconsistency. It is easier to maintain coherence in a small team working side by side as it is easier to communicate and clarify information. However, this proves to be more difficult when working in larger teams that are geographically dislocated and working without a formalised process and pre-specified flow of information. Due to the increased complexity of the project it is increasingly difficult to determine the cause of mismatch in the data sets. Coherence and inconsistency cannot be easily measured because they lack form, uniformity and homogeneity. As a result, there is a reliance on the human knowledge space to find inconsistencies in AFs. Alternative methods that may be considered to address ontological inconsistencies are: machine learning, knowledge automation, pattern matching

and leveraging the power of neural network knowledge combination.

The desired high level capability of a complex system such as a warship is to 'get it right first time'. Efforts are continuously being made to deliver a product that is better (more complex), faster (more agile), cheaper. In effect knowing the risk level associated with ontological mismatch in a SoS and taking action to minimise the risk to the system is vital. The proposed method to mitigate the risk of emerging undesirable behaviour and attaining real –time knowledge of business practices with respect to architecting is to define a formalised methodology which can be reflected in the SoS model. The ability to nail down a formalised process means that the ontological stacks for AFs can then be defined and links can be drawn across the ontology stacks towards efforts to create a rosetta stone for AFs. Ultimately addressing ontological inconsistencies that arise during collaboration efforts across multi-disciplinary teams and organisations.

VII.   CONCLUSION

This is a position paper which describes the logic and need for a generic AD that is based on a common ontology for the entire SE domain to accelerate end-to-end operations. The authors believe inconsistency in the ontology that underpins AFs makes it difficult to integrate and compare constituent information sets. Hence, calling for the development of a generic AD, based on a common ontology to facilitate effective and efficient communication amongst stakeholders and exchange data within multi-disciplinary efforts.

VIII.   REFERENCES

[1] A. W. Wymore, Model-Based Systems Engineering. Boca Raton: CRC Press, 1993.

[2] INCOSE, "Systems Engineering." https://www.incose.org/systems-engineering (accessed Jun. 29, 2021).

[3] J. Holt and S. Perry, Don't Panic - The Absolute Beginners Guide to Model-Based Systems Engineering. INCOSE UK Ltd, 2019.

[4] ISO/ IEC 19501, "Unified modeling language specification version 1.4.2," vol. 4, no. 1, pp. 25–59, 2005, Accessed: Aug. 09, 2021. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Unified+Modeling+Language+Specification#2.

[5] B. Zeigler, S. Mittal, and M. Traore, "MBSE with/out Simulation: State of the Art and Way Forward," Systems, vol. 6, no. 4, p. 40, 2018, doi: 10.3390/systems6040040.

[6] W. J. Tudor and C. Bach, "Designing marine systems in complex warships : an exploration into the deployment of model based systems engineering," 2016.

[7] W. Vaneman and R. Carlson, "Evaluating Current Systems Engineering Models for Applicability to Model-Based Systems Engineering Technical Reviews," Nav. Eng. J., vol. 132, no. 2, pp. 51–58, 2020.

[8] I. Cherfa, N. Belloir, S. Sadou, R. Fleurquin, and D. Bennouar, "Systems of systems: From mission definition to architecture description," Syst. Eng., vol. 22, no. 6, pp. 437–454, 2019, doi: 10.1002/sys.21523.

[9] W. J. Tudor and N. Harrison, "Virtual integration: managing complex warship design through model based engineering," Proc. Engine As A Weapon Int. Symp., vol. 1, no. July, pp. 1–13, 2019, doi: 10.24868/issn.2515-8171.2019.009.

[10] Z. X. Liang, L. Yan, and J. Z. Shang, "Collaborative multidisciplinary decision making based on game theory in ship preliminary design," J. Mar. Sci. Technol., vol. 14, no. 3, pp. 334–344, Sep. 2009, doi: 10.1007/s00773-009-0048-3.

[11] D. . Andrews, "Simulation and the design building block approach in the design of ships and other complex systems," Proc. R. Soc. A Math. Phys. Eng. Sci., vol. 462, no. 2075, pp. 3407–3433, Nov. 2006, doi: 10.1098/rspa.2006.1728.

[12] INCOSE, "Systems Engineering Vision 2020," 2007.

[13] INCOSE, "A WORLD IN MOTION. Systems Engineering Vision - 2025," 2014.

[14] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 42, no. 1, pp. 101–111, 2012, doi: 10.1109/TSMCC.2011.2106495.

[15] VitechCorp, ""The Value of Model-Based Systems Engineering" with Mark Simons - YouTube," 2016. https://www.youtube.com/watch?v=aFNlDRA20YQ&t=28s (accessed Apr. 23, 2019).

[16] S. Barat, V. Kulkarni, T. Clark, and B. Barn, Model-Driven Engineering and Software Development, vol. 880, no. ii. Springer International Publishing, 2018.

[17] J. A. Bayliss, "FOR TOPSIDE DESIGN AND INTEGRATION IN PRELIMINARY Warship Design," p. 380, 2003.

[18] W. Wang, "The nature of evolutionary artefact and design process knowledge coupling," no. June, 2008.

[19] T. Imron, "Socio-technical architectural model of collaborative engineering design," 2017.

[20] É. Boisseau, J. F. Omhover, and C. Bouchard, "Open-design: A state of the art review," Des. Sci., vol. 4, pp. 1–44, 2018, doi: 10.1017/dsj.2017.25.

[21] T. Mettler, M. Eurich, and R. Winter, "Communications of the Association for Information Systems On the Use of Experiments in Design Science Research: A Proposition of an Evaluation Framework On the Use of Experiments in Design Science Research: A Proposition of an Evaluation Framework On the ," Commun. Assoc. Inf. Syst., vol. 34, no. 10, pp. 223–240, 2014, [Online]. Available: http://aisel.aisnet.org/cais%0Ahttp://aisel.aisnet.org/cais/vol34/iss1/10

[22] A. M. Madni, B. Boehm, R. G. Ghanem, D. Erwin, and M. J. Wheaton, Disciplinary Convergence in Systems Engineering Research. 2017.

[23] M. A. Parsons et al., "Application of a Distributed System Architectural Framework to Naval Ship Concept and Requirements Exploration (C&RE)," Nav. Eng. J., vol. 132, no. 4, pp. 105–124, 2020, [Online]. Available: https://www.ingentaconnect.com/contentone/asne/nej/2020/00000132/00000004/art00031%0A.

[24] L. Yang, K. Cormican, and M. Yu, "Ontology-based systems engineering: A state-of-the-art review," Comput. Ind., vol. 111, pp. 148–171, 2019, doi: 10.1016/j.compind.2019.05.003.

[25] R. Nilsson, D. Dori, Y. Jayawant, L. Petnga, H. Kohen, and M. Yokell, "Towards an Ontology for Collaboration in System of Systems Context," INCOSE Int. Symp., vol. 30, no. 1, pp. 666–679, 2020, doi: 10.1002/j.2334-5837.2020.00747.x.

[26] M. Acierno, S. Cursi, D. Simeone, and D. Fiorani, "Architectural heritage knowledge modelling: An ontology-based framework for conservation process," J. Cult. Herit., vol. 24, pp. 124–133, 2017, doi: 10.1016/j.culher.2016.09.010.

[27] D. Dori, Model-based systems engineering with OPM and SysML. 2016.

[28] ISO/IEC/IEEE, "INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — agile environment," ISO/IEC/IEEE 26515 First Ed. 2011-12-01; Corrected version 2012-03-15, vol. 2011, p. 94, 2011, doi: 10.1109/IEEESTD.2011.6146379.

[29] IEEE, "Survey of Architecture Frameworks," 2020. http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html (accessed Jun. 30, 2021).

[30] J. S. Meehan, A. H. B. Duffy, and R. I. Whitfield, "Supporting 'design for re-use' with modular design," Concurr. Eng. Res. Appl., vol. 15, no. 2, pp. 141–155, Jun. 2007, doi: 10.1177/1063293X07079319.

[31] M. Wilkinson, "Z8: System Architecture," INCOSE UK, 2020. www.opengroup.org/togaf (accessed Aug. 09, 2021).

[32] A. Morkevicius and J. Towers, Don't Panic! The Absolute Beginner's Guide to Architecture Frameworks. INCOSE UK Ltd, 2020.