MTR210193
MITRE TECHNICAL REPORT

Digital Engineering Governance

A Perspective on Governance for the Era of Digital Engineering

Sponsor: NGA/CIO
Dept. No.: N443
Contract No.: W56KGU190004
Project No.: 0721P04Z-8S

McLean, VA

Author(s):
Kenneth J. Laskey
Martha L. Farinacci
Omar C. Diaz

Reviewer(s):
Melvin H. Dedicatoria
Michael S. Russell
Robert M. Kobee

May 2021

Abstract

This paper provides the reader with a perspective on the concepts surrounding governance for digital engineering. The Digital Engineering Strategy released by the Department of Defense in June 2018, noted that governance was about ensuring models and data are formally managed and trusted throughout the lifecycle. For this, a governance process should help stakeholders to resolve issues, ensure consistency and accuracy of the authoritative source of truth, and enable stakeholders to make data-driven decisions.

This page intentionally left blank.

Acknowledgement

The authors of this technical paper recognize Ms. Rosie P. Scott of the National Geospatial-Intelligence Agency for her leadership role to adopt a Digital Engineering approach and develop this material for her agency. Without Ms. Scott's leadership, it would have been challenging to define an integration path comprising of people, process, and technology, appropriate to realize an effective digital engineering ecosystem capable of supporting the Agency's needs.

Table of Contents

List of Figures

This page intentionally left blank.

1 Introduction

The release of the Department of Defense (DoD) Digital Engineering (DE) Strategy in 2018 set the vision to transition from document-based to model artifacts to optimize performance and affordability in the development of capabilities required to meet current and future challenges. Advances in modeling technologies are enabling the collaboration and integration of the multi-discipline engineering activities performed across the different stages of a systems development lifecycle.

The DoD DE Strategy further notes that,

> *Traditionally, the Department has relied on a linear process to develop complex systems that serve a range of missions and users. Often the acquisition engineering processes are document-intensive and stove-piped, leading to extended cycle times with systems that are cumbersome to change and sustain. Current acquisition processes and engineering methods hinder meeting the demands of exponential technology growth, complexity, and access to information.*[1]

This strategic endeavor of using advanced modeling technologies extends beyond the Department of Defense[2]. Other government and many commercial organizations are transforming their traditional engineering practices by incorporating technological innovations into an integrated, digital, model-based environment to comprise the sole source of truth to support their decision-making activities. However, most adoption efforts focus on the implementation of technology as the initial steps, neglecting to develop an appropriate plan that includes a governance framework.

Noteworthy, publications about digital engineering are slowly emerging in the literature. The majority of publications focus on model-based systems engineering (MBSE), a subset of the broader topic of digital engineering. Those publications predominately emphasize the utility, benefits, methods, use cases, and tools for applying MBSE. However, insufficient attention is given to the various factors to consider when implementing and tailoring digital engineering to a particular organization. One of these factors is governance. This paper will focus on factors to consider when developing a governing framework for an era of digital engineering.

2 A Perspective on the Purpose of Governance

Governance is about making decisions that are aligned with the overall organizational strategy and culture of the enterprise as envisioned by the executive leadership establishing these. It specifies the decision rights and accountability framework to encourage desirable behaviors towards realizing the strategy and defines incentives (positive or negative) towards that end. It is

[1] US DoD. (2018, June). Digital Engineering Strategy, Office of the Deputy Assistant Secretary of Dense for Systems Engineering.

[2] Huang, J., Gheorghe, A., Handley, H., Pazos, P., Pinto, A., Kovacic, S., ... & Daniels, C. (2020). Towards digital engineering: the advent of digital systems engineering. International Journal of System of Systems Engineering, 10(3), 234-261.

less about overt control and strict adherence to rules, and more about guidance and effective and equitable usage of resources to ensure the attainment of an organization's strategic objectives[3].

Effective governance communicates Leadership expectations, stating the objectives and outcomes Leadership expects to be realized through activities defined and executed by Management[4]. Leadership provides oversight of progress, where progress is evaluated based on metrics agreed to by Leadership and Management. Describing how Leadership specifies desired objectives and outcomes through governance consists of assigning responsibilities for setting objectives, outcomes, and measures. It is based on a philosophy of delegating decision authorities and accountability down to those with subject matter expertise.

An example of the desired outcome would be to drive decisions about Systems Vulnerabilities to Unacceptable Losses based on real-time access to dashboards linked to the digital engineering environment as shown in **Figure 1**. In this example, leadership in the cyber-security domain could make informed decisions to address and prioritize high risk vulnerabilities to a system that would cause unacceptable losses to the organization.

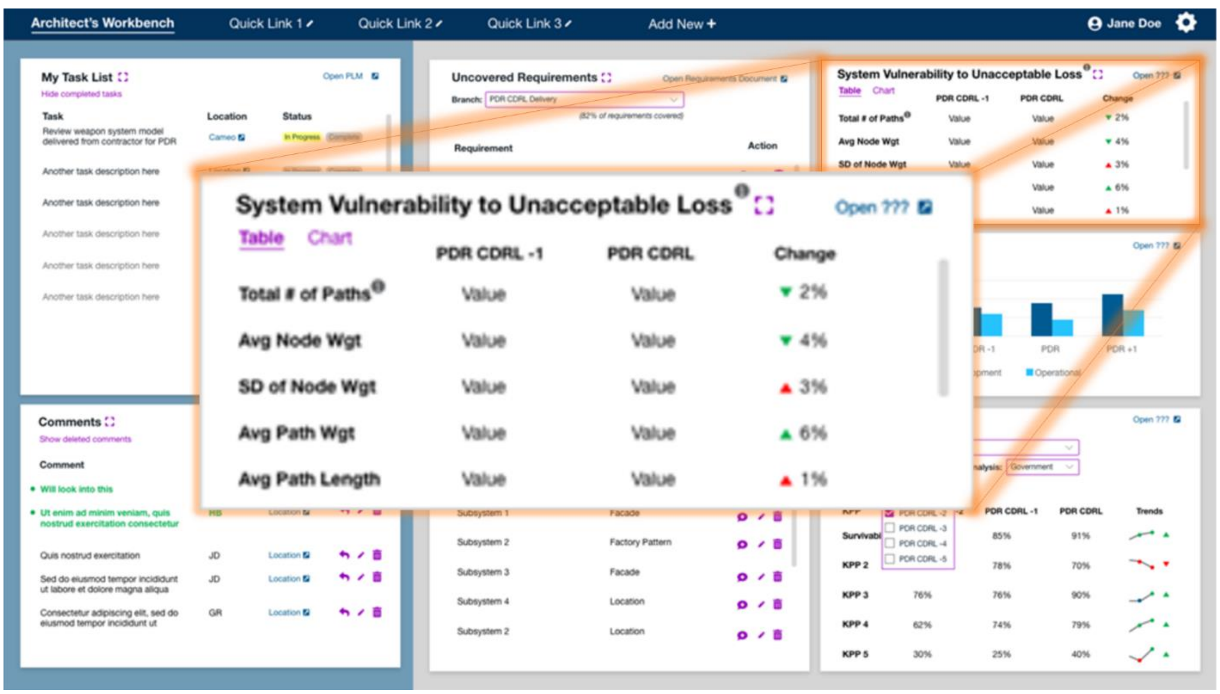

Figure 1 MITRE's DE Dashboard Mockup (source: The MITRE Corporation)

Ideally, the metrics are captured through automated processes that ensure consistency and repeatability. Automation also enables progress updates to be generated on-demand, avoiding

---

[3] Reference Architecture Foundation for Service Oriented Architecture Version 1.0. 04 December 2012. OASIS Committee Specification 01. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html.

[4] Sirisomboonsuk, P., Gu, V. C., Cao, R. Q., & Burns, J. R. (2018). Relationships between project governance and information technology governance and their impact on project performance. International journal of project management, 36(2), 287-300.

major deficiencies surfacing late in the project or during milestone reviews[5,6]. The overall intent is to create a well-defined and efficient decision-making process that will increase confidence in outcomes.

A growing trend in the information technology field is the adoption of agile development and project management methods. A fundamental of agile governance is to minimize reviews and decision gates that introduce avoidable delays[7]. The agile approach delegates decision authority to those with the subject matter expertise to make the necessary design decisions, without neglecting enterprise needs in a balance approach. For example, decisions on a data schema can be made by those who will need to use the schema and not by leadership who is focused on project outcomes and not on the technical details. Agreement is generated among those to whom decision authority has been delegated and matters where agreement is not reached are escalated to leadership.

The approach taken in this governance framework is to appreciate that different groups have and will come up with variations of solutions and it is more important to identify the variation used rather than necessarily mandating a single solution. The most successful specifications have a principled means to adequately define and unambiguously identify each variation. As an example, the Uniform Resource Identifier (URI) specification, defines the following syntax:

URI = scheme:[//authority]path[?query][#fragment].

Users are most familiar with the HTTP and HTTPS schemes; however, the specification further defines the syntax of the scheme portion and the mechanism for registering both additional permanent and provisional schemes without needing to change the widely used main specification. This provides robustness and stability while supporting past and future needs.

2.1 A Shift from Rigid to Flexible Governance

For organizations to realize the benefits of digital engineering transformation, a shift in governance is required. A governance approach must accommodate changes emanating from the mission, technology, the understanding of desired outcomes, and impediments to realizing those outcomes. Thus, governance must be responsive on the time scale of the changes. This indicates the need for governance that is less static and prescriptive about the details than the current governance structures that are commonly used across many government organizations[8].

A governance framework for digital engineering must be configurable within overall processes whose execution can be automated and whose results can be assessed for compliance in real-time as part of the process execution. The details can be specified through updated policies or the

[5] Arachchi, S. A. I. B. S., & Perera, I. (2018, May). Continuous integration and continuous delivery pipeline automation for agile software project management. In 2018 Moratuwa Engineering Research Conference (MERCon) (pp. 156-161). IEEE.

[6] Mohammad, S. M. (2018). Improve Software Quality through practicing DevOps Automation. Sikender Mohsienuddin Mohammad," IMPROVE SOFTWARE QUALITY THROUGH PRACTICING DEVOPS AUTOMATION", International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.

[7] Luna, A. J. D. O., Kruchten, P., & de Moura, H. P. (2015). Agile governance theory: conceptual development. arXiv preprint arXiv:1505.06701.

[8] Linkov, I., Trump, B. D., Poinsatte-Jones, K., & Florin, M. V. (2018). Governance strategies for a sustainable digital world. Sustainability, 10(2), 440.

results of focused technical activities by subject experts who can quickly respond to technical needs. Despite digital engineering not requiring the adoption and use of agile methodologies, both are mutually reinforcing. A governance framework that is adaptable to any organization's development lifecycle and has incorporated model-based systems engineering into its methodology can realize the benefits of consistency and expedited evolution.

The digital engineering approach introduces the need to identify and manage the quality, configuration, and change history of the models and data that make up an "authoritative source of truth"[9]. To accomplish this, governance requires processes for:

- Identifying and managing configurations of digital engineering policies, procedures, and standards.
- Identifying data and models that constitute authoritative sources.
- Specifying how to identify semantics being used and what constitutes adequate documentation of the chosen semantics.
- Specifying required provenance information, (e.g., change history, current state), so a user can decide what information is authoritative for a current purpose, who is authorized to access data and models, and how access is accomplished.

Leadership should identify those with the technical and process expertise to define and execute the details of a governance framework for their digital engineering adoptions. This does not always require standing committees and boards but can rely upon constituting bodies as needed with the expertise required at the time or included as agenda items on existing ones. Agile principles talk to self-organizing groups of resources available and willing, as needed to solve problems.

2.2 Integration of Models in the Authoritative Source of Truth

An organization's digital engineering environment will manage and maintain a diverse set of models for a variety of purposes. A commonly agreed definition of a model is:

> *A selective representation of some system, at a point in time or space, whose form and content are chosen based on a specific set of concerns. The model is related to the system by an explicit or implicit mapping to promote understanding of the real system*[10,11].

The Authoritative Source of Truth (ASoT) serves as the central reference point for models and data across the lifecycle, capturing the current state and the history of changes to systems[12]. This provides traceability as a system of interest evolves, capturing historical knowledge, and

---

[9] Huang, J., Gheorghe, A., Handley, H., Pazos, P., Pinto, A., Kovacic, S., ... & Daniels, C. (2020). Towards digital engineering: the advent of digital systems engineering. International Journal of System of Systems Engineering, 10(3), 234-261.

[10] Object Management Group. 2010. MDA Foundation Model. OMG document number ORMSC/2010-09-06.

[11] Bellinger, G. 2004. Modeling & Simulation: An Introduction. Accessed on 15 April 2021. Available at http://www.systems-thinking.org/modsim/modsim.htm

[12] Huang, J., Handley, H., Pazos-Lago, P., Pinto, A., Kovacic, S., Collins, A., & Daniels, C. (2019). Digital Systems Engineering: Concepts, Challenges and Enabling Technologies.

connecting authoritative versions of the models and data[13,14]. Changes to models and/or data must propagate to the respective authoritative sources throughout the lifecycle of the affected systems and/or functions[15].

Systems Engineering leadership, as described by the International Council on Systems Engineering (INCOSE[16]), should assign the roles, authorities, and responsibilities to perform the model-related activities in a digital engineering cycle as described below:

- Identify Verification, Validation, and Accreditation (VV&A) or other quality review activities and assign accountability for such activities. These could be different roles for different types of models. For example, the systems engineer lead within a program office (PO) may be assigned the authority to approve or not approve of system design models. On the other hand, the Independent Government Cost Estimate (IGCE) lead may be assigned the authority to approve or not cost models.
- Develop and apply measures of model "truth", which may include completeness, consistency, coherence, and accuracy of the models.
- Develop a mechanism for publishing the quality measures and for linking these to the models for stakeholders who contribute or consume from the DE environment to validate the accuracy and utility of the models supporting their decisions.

2.3 The Purpose for Managing Models

The purpose of establishing a process for model management is to unambiguously define the models and the relationships among them, to identify and, as needed, remediate inconsistencies between models, and to manage and control configurations. Within an organization, many types of models will be developed, maintained, and used. There will be different categories of models, such as requirements, architecture, design, interface, cost, and schedule. Within each category, there will be different levels of models, ranging from high-to-low level, abstract models used for executive decision-making down to detailed design models that drive development, acceptance testing, and deployment among others. Within each category and level, models could be decomposed to support parallel development by different teams, resulting in the evolution of many model variations, configurations, and baselines.

A process for model management should include managing Configuration Items (CIs), including the following attributes:

- *Variation* – a change of a model CI for a given context.
- *Version* – an identifier of a variation.
- *Configuration* – an identified set of variations.

[13] Bone, M., Blackburn, M., Kruse, B., Dzielski, J., Hagedorn, T., & Grosse, I. (2018). Toward an interoperability and integration framework to enable digital thread. Systems, 6(4), 46.

[14] Kruse, B., & Blackburn, M. (2019). Collaborating with OpenMBEE as an Authoritative Source of Truth Environment. Procedia Computer Science, 153, 277-284.

[15] McDermott, T., Collopy, P., Nadolski, M., & Paredis, C. (2019). The Future Exchange of Digital Engineering Data and Models: an Enterprise Systems Analysis. Procedia Computer Science, 153, 260-267.

[16] INCOSE. 2015. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0

- *Baseline* – an immutable configuration, often used to freeze model CIs at critical points in the model development life cycle, such as milestone reviews, approval to deploy, and authorization to operate.

An example use of model management is to track variations of models as part of a trade study, as illustrated in Figure 2. In this use case, a new capability may be represented within the digital engineering environment as a collection of model variations. An analysis may discover problems with the operational performance, and the engineer may then create model variations by altering some aspects of the design. These variations would be represented as branches off the original design model in the digital engineering environment. For each design variation, other related models would be updated. Several evaluation factors would also be identified, such as design feasibility, operational performance, cost, and schedule.

Analyses of each variation would be conducted against each of the evaluation factors and constraints. Upon completion of the analysis of alternatives, the engineer would work with the mission owner to select the best variation. An important function of model management is to ensure that all other models (e.g., cost, schedule, risk, technical and operational performance) are updated and linked to the new design. The updated collection of related models would be tagged with new configuration metadata and retained in the digital engineering environment.

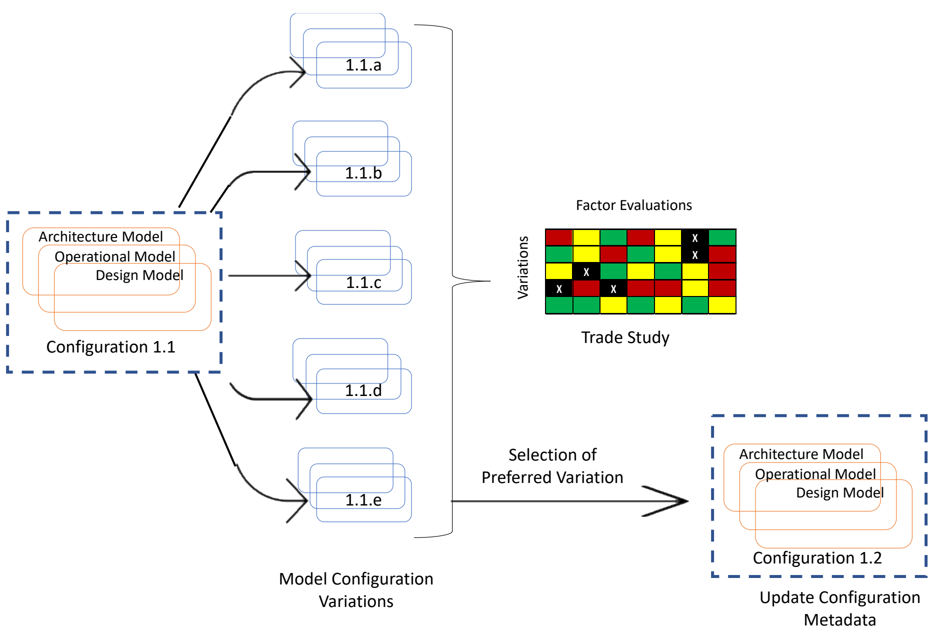

Figure 2 Variations of Models in Trade Studies (source: The MITRE Corporation)

The synchronization of models enables stakeholders to access up-to-date views of the system models and to continuously evaluate those models. Model Management enables stakeholders to assess capabilities and interactions throughout the system lifecycle. This will result in the early detection and correction of problems. Model Management introduces automated workflows, including totally electronic and asynchronous reviews, supporting digital approvals that enable audits to provide process compliance. This enforces consistency in the workflow process and results.

Naming conventions (e.g., model numbers) and state identities (baselines, configurations, variations) provide the basis for Model Management governance to track, control, and manage models effectively. Models that are identified per an organization's approved naming conventions and marked for Model Management control at the appropriate lifecycle states can be managed like any other product format.

Systems Engineering leadership should assign roles, authorities, and responsibilities to establish the processes for synchronizing the create, read, update, and delete (CRUD) operations on heterogeneous models throughout the system development lifecycle. This includes:

- Identifying the categories and levels of models that will be subject to DE Model Management.
- Defining the metadata specifications to unambiguously track and control model configurations and variations, as well as their relationships.
- Identifying the lifecycle stages when baselines need to be delivered to and retained by the government organization, and the retention timeframes. Baselines may be required for generating contractually required artifacts, such as those specified in the Contract Data Requirements List (CDRL).
- Defining automated workflows, automated digital approvals, and retention policies.
- Developing processes and acquiring the tools to automate DE Model Management.
- Ensuring compliance with DE Model Management processes and specifications.

2.4 Adapting a Model Lifecycle Management

Model Lifecycle Management (MLM) is the curation, maintenance, and synchronization of the modeling information over time to ensure a consistent representation of the system being modeled[17]. System models will typically consist of a collection of integrated, multi-domain models. Furthermore, those models will evolve continuously during the lifecycle. For example, during the early phases of system development, system engineers will develop requirements, concept sketches, block diagrams, flow charts, back-of-the-envelope calculations (such as cost roll-ups), test plans, and architectural trade-off models.

As the system definition matures, the focus will shift to developing high-fidelity designs (structural and functional), analyses, and optimization models of various subsystems and components. Further along in the lifecycle, the systems models and requirements will be linked to technical, operational and test results as well as virtual Authorization to Operate (ATO). Although these models will be developed by different teams, using different software tools, methodologies/workflows, and at different stages in the system lifecycle, they all represent different aspects of the same system.

[17] Fisher et al, (2014) Model Lifecycle Management for MBSE, Object Management Group, http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:model_lifecycle_management_for_mbse_v4.pdf

Models are the central object of digital engineering. In digital engineering, governance focuses on managing the models and associated data, regardless of the lifecycle methodology used to develop those models. The governance needs to ensure that the system is represented using a modeling language that is well understood by the model creator and the model consumer. This will facilitate the transformation of architectures, enable reusable cross-platform component technologies, and facilitate faster response to new system needs[18]. At most organizations, the modeling activities of the engineering lifecycle are comprised of defining, designing, analyzing, and documenting the system under development.

2.5 Managing Hierarchical Federation of Models

In addition to synchronizing models across the development lifecycle, models need hierarchical alignment. Within an organization digital engineering models could be created at the enterprise level, the solution level, the program level, and the implementation level (see figure 3). Typically, the higher-level models will include organization-wide capability requirements, the enterprise architecture, master schedules, and standards. Those models will inform capital planning and investment decisions and be assigned to a specific department or division within the organization for refinement and development.

At the solution level, a program office maintains a solution backlog, defines capabilities, and maps capabilities to the enterprise level. A program office could also assign capabilities to other programs and model any interoperability requirements or interface controls that are needed between the programs. At the program level, product line teams will decompose solution-level capabilities into minimum-viable product (MVP) functions, maintain the program-level backlog, and define the continuous delivery pipeline.

At the implementation level, development teams maintain and prioritize the collection of user stories with links to the product line feature sets, and incrementally develop, test, and deploy working software to satisfy the user stories. Maintaining unambiguous hierarchical alignment of models and relationships between the levels will enable the organization to assess and manage the impacts of changes.

For example, an unexpected schedule slip at the program level could be quickly analyzed for impact at the solution or enterprise levels, triggering a need to automatically update the master schedule. Likewise, an unexpected budget cut at the enterprise level could be traced to the solution and program levels, triggering a need to review and update program and solution roadmaps and milestones. As software is completed and deployed, backlogs would be updated at the implementation, program, and solution levels, and enterprise capability gaps and shortfalls would be updated.

[18] Dove, R., Schindel, W., & Garlington, K. (2018, July). Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group. In INCOSE International Symposium (Vol. 28, No. 1, pp. 303-320).

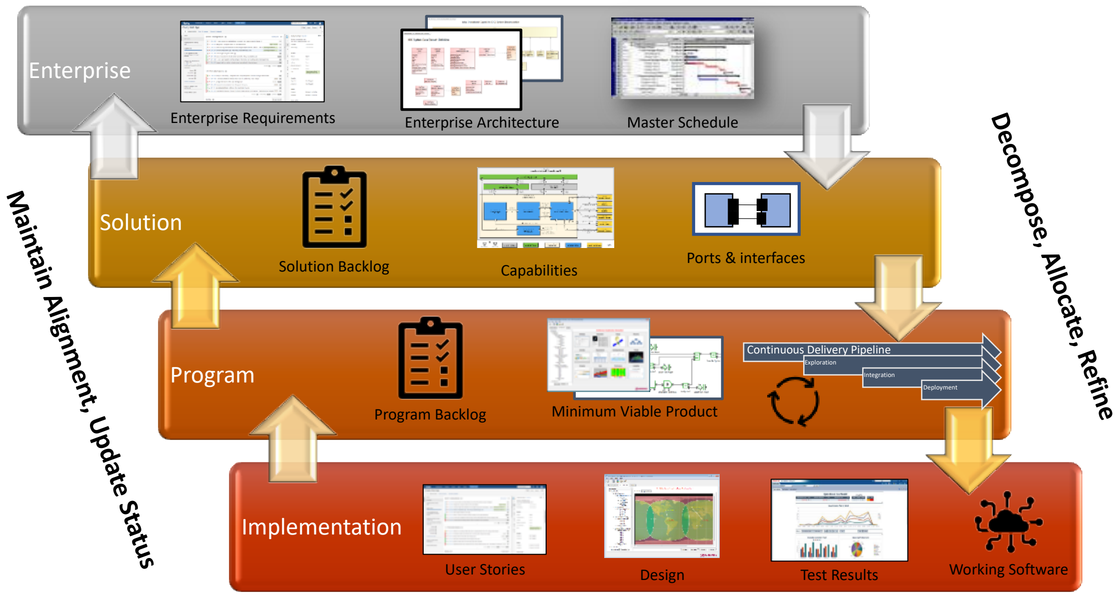

Figure 3 A Hierarchy Federation of Models (source: The MITRE Corporation)

3 Data Management Considerations for Digital Engineering

Since introduced by Capital One in 2003, the role of Chief Data Officer (CDO) is becoming a more common across organizations to include the federal government[19,20,21,22,23]. In this role, the CDO is the responsible for implementing the enterprise data management strategy[24]. Considering the likelihood that authoritative data sources will be distributed throughout the organization, the data management strategy should include mechanisms to deliver trustworthy data and to ensure coherence of models and analyses needed for mission engineering[25]: Data management activities and products include:

- Definition and documentation of vocabulary that is machine-processable or computable.
- Documentation including data types, formats, and dimensions, as applicable.
- Vocabulary, including different versions, should have an identifier by which vocabulary documentation can be retrieved.

[19] Lee, Y., Madnick, S. E., Wang, R. Y., Wang, F., & Zhang, H. (2014). A cubic framework for the chief data officer: Succeeding in a world of big data.

[20] Griffin, J. (2008). The role of the chief data officer. Information Management, 18(2), 28.

[21] Nie, Y., Talburt, J., Dagtas, S., & Feng, T. (2019). The influence of chief data officer presence on firm performance: does firm size matter?. Industrial Management & Data Systems.

[22] Zhang, H., Lee, Y., Wang, R., & Huang, W. (2017). Chief data officer appointment and origin: A theoretical perspective.

[23] Nie, Y., Talburt, J., Li, X., & Xiao, Z. (2018). Chief data officer (CDO) role and responsibility analysis. Journal of Computing Sciences in Colleges, 33(5), 4-12.

[24] Lee, Y., Madnick, S. E., Wang, R. Y., Wang, F., & Zhang, H. (2014). A cubic framework for the chief data officer: Succeeding in a world of big data.

[25] Zimmerman, P., & Dahmann, J. (2018, October). Digital Engineering to Support Mission Engineering. In Proceedings of the 21st Annual NDIA Systems and Mission Engineering Conference, Tampa, FL, USA (pp. 22-25).

- Instances of vocabulary use should identify which vocabularies, including versions, have been used in creating the instance.
- Translation between vocabularies, including different versions, should use vocabulary identifiers as parameters for the translation services.
- Associated metadata should have both vocabularies defining metadata properties and vocabulary defining valid values that may be assigned to those metadata properties.
- Provenance (see Section 4.2 for definition) is an essential component of the metadata for data, models, and other entities to establish trust and evaluate if data, models, or other entities can be considered authoritative and trusted for current use.
- Provenance includes but not limited to the identification of approved versions, configurations, baselines, and the pedigree of these items.

4 The Use of Semantics and Pedigree Concepts in Digital Engineering

Systems Engineering leadership should assign the roles, authorities, and responsibilities to perform the activities in underpinning the definition and use of semantics and provenance. This includes identifying semantics in use, specifying what information provides the basis for assessing provenance and collecting corresponding values to assemble and maintain a body of evidence to support determining provenance. Also, the Systems Engineer leadership is responsible for the development and enforcement of the style guide.

4.1 Disambiguating Interpretation and Understanding

Semantics formalizes the meaning of symbolic models to enable the unambiguous representation and interpretation of a system in question[26,27]. The organization's Enterprise Architecture metamodel could define the semantics for each of the modeling elements as the building blocks relevant to the agency's mission to enable effective and efficient model-based systems engineering outcomes. A style guide could define the set of rules and principles for consistent modeling. Adherence to the style guide reduces the number of diagram types and stylistic choices used in the construction of organization conformant architectures, with the following expected results[28]:

- Fewer modeling symbols to learn as there is a set of approved notations and notational elements to use.
- Limited number of modeling "dialects" to reduce the cognitive overload for both the designers and users. The resulting architecture views will become easier to understand by all stakeholders.

[26] Lankhorst, M. & et al. 2013. Enterprise Architecture at Work: Modelling, Communication and Analysis (2th. ed.). Springer Publishing Company, Incorporated.

[27] M. Schmit, S. Briceno, K. Collins, D. Mavris, K. Lynch and G. Ball, "Semantic design space refinement for model-based systems engineering," 2016 Annual IEEE Systems Conference (SysCon), Orlando, FL, 2016, pp. 1-8, doi: 10.1109/SYSCON.2016.7490579.

[28] Enterprise Architecture based on Design Primitives and Patterns Guidelines for the Design and Development of Event-Trace Descriptions (DoDAF OV-6c) using BPMN. Office of the Deputy Chief Management Office. December 17, 2009.

- Standardized representation of architecture modeling elements will enable the comparison of different architectures, enabling the reuse of common patterns. This also lowers the construction cost for enterprise architectures.
- Standardized modeling methods increase the potential of acquiring qualified professionals, define uniform training, and accelerate incorporating new team members into projects.

The style guide enables the assessment of a model's syntactic and semantic consistency against the following attributes:

- Syntactical Correctness – the model satisfies the rules of the modeling language: A model must satisfy the vocabulary and grammar restrictions of the modeling language chosen.
- Semantic Correctness – the model satisfies the semantic requirements of the problem domain depicted.
- Problem Relevance – an architecture model must contain content that is relevant to the problem domain surveyed.
- Maximum Coverage – an architecture model must capture all relevant aspects of the problem domain.
- Minimum Size – a model should not contain details that are irrelevant to the model users.
- Systematic Design – the results of the architecture design process should follow a systematic layout.
- Cost-effective Design – the creation of architecture models should not incur prohibitive construction costs.
- Clarity of Design – crossing lines and overlapping symbols should be avoided wherever possible.
- Comparable Design – related content should be arranged similarly to enable a cross-check to uncover structural analogies.

4.2 Establishing Trustworthiness of Sources to Support Decision-Making

Provenance is information about entities, activities, and people involved in producing a piece of data or thing, e.g., a model, which can be used to form assessments about its quality, reliability, or trustworthiness[29]. Data provenance is a type of metadata that is important to confirm the authenticity of data and to enable it to be reused[30]. It provides a critical foundation for assessing authenticity, enabling trust, and enabling reproducibility[31]. Put simply, provenance answers the questions of why and how the data was produced, where and when, and by whom[32].

[29] https://www.w3.org/TR/prov-overview/

[30] Governance will specify the intrinsic quantitative and qualitative characteristics of models and data and the processes through which those characteristics are assigned values and how those values are appropriately updated.

[31] Dai, C., Lin, D., Bertino, E., & Kantarcioglu, M. (2008, August). An approach to evaluate data trustworthiness based on data provenance. In Workshop on Secure Data Management (pp. 82-98). Springer, Berlin, Heidelberg.

[32] https://www.ands.org.au/working-with-data/publishing-and-reusing-data/data-provenance

Figure 4 is a model for provenance developed by the World Wide Web Consortium (W3C) and summarized in the PROV standard [33]. The core concepts defined by this model are Entity, Activity, and Agent.

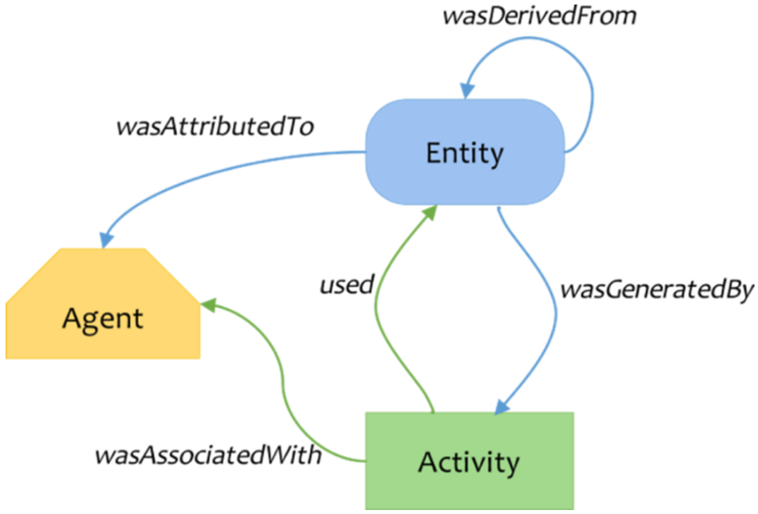

Figure 4 High-Level Overview of the Structure of PROV Records (source: W3C)

- An *Entity* captures a thing in the world (in a particular state). The entity was derived from some other entity and was generated by an *Activity* that used other entities.
- An *Agent* (e.g., a person or software execution) is associated with the activity, and the entity that was generated by the activity is attributed to that agent.

Information used for provenance must be provided and made available through links to other entities, such as links to configurations, versions, baselines, and analysis results. Each of these entities must be unambiguously identified and accessible through its corresponding identifier.

Provenance metadata with enterprise scope, e.g., used for access control or expressing intellectual property, must be consistent across that enterprise. Also, accommodations must be made to share provenance metadata across contracts or organizations that develop models and deliver these to the Operations & Maintenance (O&M).

Governance of provenance includes the capture of relevant assumptions. Data producers would establish which assumptions are important to adequately describe the models or data and then how to convey this to users. Here, provenance looks to how assumptions align with existing conditions and provenance is concerned with suitability. For users, it is necessary to understand context so they can determine if the data is "authoritative" for their purposes.

To maintain the trustworthiness, confidence and continued applicability of models and data, it is important to automate, to the extent possible, the activities (or workflows) establishing and assessing the provenance of the models and the data used in models[34]. With automation in place, all stakeholders can update provenance conclusions throughout the digital engineering

[33] https://en.wikipedia.org/wiki/PROV_(Provenance)

[34] Zafar, F., Khan, A., Suhail, S., Ahmed, I., Hameed, K., Khan, H. M., ... & Anjum, A. (2017). Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. Journal of Network and Computer Applications, 94, 50-68.

lifecycle. Some reviews could become electronic and asynchronous, thereby adding consistency, and avoiding surprises during milestone reviews. The electronic approvals become data recorded in the ASoT that is available to support audits.

5 An Exemplar Adoption

The National Geospatial-Intelligence Agency (NGA) is adopting digital engineering. The transition from disparate sets of documents (i.e., physical, or digital state) management process to the integration of critical data into digital models enables insight to address broader questions about technology investments. The implementation of an integrated environment fuses data from domain-specific authoritative sources of truth to allow multi-disciplinary stakeholders to collaborate through the common metamodel to support a broad range of activities.

This adoption requires an approach focused on governing the interaction between the diverse authoritative sources of truth and the digital engineering environment, integrating these into the agency's metamodel. Figure 5, although it does not represent the Agency's digital engineering environment, does illustrate the three components (inside the highlighted orange oval) governed in the interactions between authoritative sources and the Model Integration Environment; these are:

- The *file schema* defining the data elements from each of the authoritative sources.
- The *interface* implementing the connection between the Model Integration Environment and the authoritative source.
- The *Model Integration Environment* linking data and model elements from other authoritative sources.

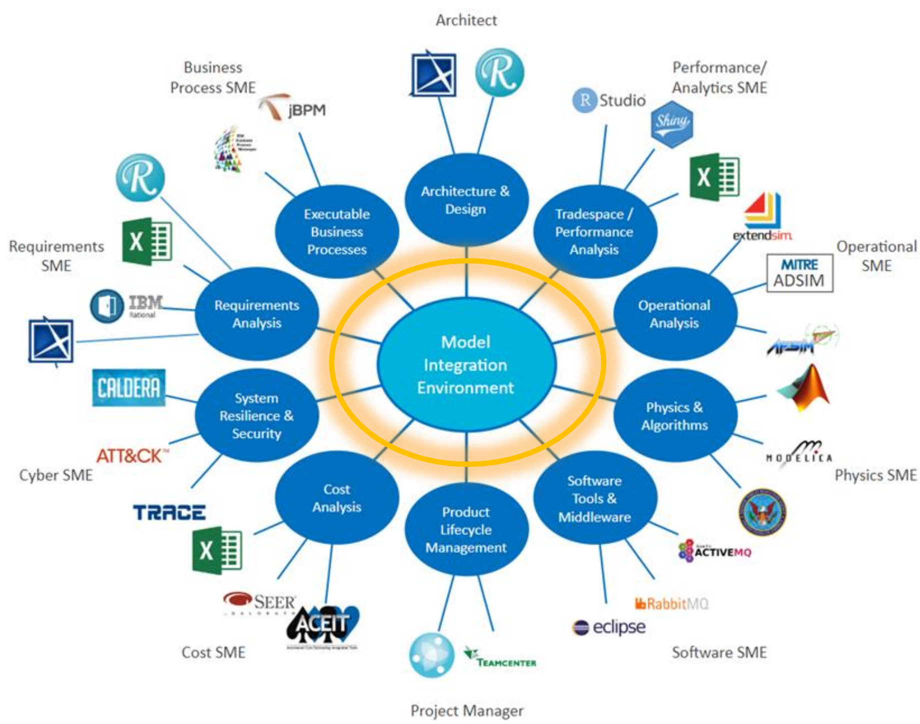

Figure 5 Governed ASoTs and DE Environment Interactions (source: The MITRE Corporation)

Like most federal government organizations, NGA has governing boards in place performing oversight of domain specific business processes. The approach to define the appropriate governance framework for digital engineering is cognizant of the number of governing boards. Thus, governance is defined to provide oversight of the interactions implemented in the interfaces, the files describing data elements from the authoritative sources, and the DE environment containing the integration of all the autorotative sources. This approach avoids adding to the preexisting complexities, generating additional structures that could have adverse effects such as stifling innovation or impeding the needed agility expected from the digital engineering ecosystem.

6 Summary

Much of the literature on MBSE and DE focuses on utility, benefits, methods, tools, and infrastructure. As important as these topics are, the full benefits of these technologies in complex government organizations require effective governance. Governance addresses oversight and escalation protocols in cases where agreement is needed and has not been achieved. For digital engineering, a governance framework should cover the topics of “authoritative source of truth” for integrating models and data, lifecycle management, data provenance, and semantics for establishing trustworthiness.

Noteworthy, those involved in implementing digital engineering must thoughtfully define the appropriate governance framework, as there is not a “one-size-fits-all” approach to follow. As organizations transition from primarily paper-based engineering to model-based engineering, governance processes will need to be modified to cover the topics of management and control of digital artifacts. This paper provides a perspective on governance in the era of digital engineering. It covers the topics of ASoT for models and data, lifecycle management, data provenance, and semantics.

7 References/Bibliography

Arachchi, S. A. I. B. S., & Perera, I. (2018, May). Continuous integration and continuous delivery pipeline automation for agile software project management. In 2018 Moratuwa Engineering Research Conference (MERCon) (pp. 156-161). IEEE.

Berners-Lee, T., Masinter, L., & McCahill, M. (1994). Uniform Resource Locators. Network Working Group. Internet Engineering Task Force. Retrieved from https://tools.ietf.org/html/rfc1738

Bone, M., Blackburn, M., Kruse, B., Dzielski, J., Hagedorn, T., & Grosse, I. (2018). Toward an interoperability and integration framework to enable digital thread. Systems, 6(4), 46.

Dai, C., Lin, D., Bertino, E., & Kantarcioglu, M. (2008, August). An approach to evaluate data trustworthiness based on data provenance. In Workshop on Secure Data Management (pp. 82-98). Springer, Berlin, Heidelberg.

Data Provenance. (n.d.). The Australian National Data Service. Retrieved from https://www.ands.org.au/working-with-data/publishing-and-reusing-data/data-provenance

Dove, R., Schindel, W., & Garlington, K. (2018, July). Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group. In INCOSE International Symposium (Vol. 28, No. 1, pp. 303-320).

Enterprise Architecture based on Design Primitives and Patterns Guidelines for the Design and Development of Event-Trace Descriptions (DoDAF OV-6c) using BPMN. Office of the Deputy Chief Management Office. December 17, 2009.

Estefan, J. A., Laskey, K., McCabe, F. G., & Thornton, D. (2009). Reference Architecture Foundation for Service Oriented Architecture. Version, 1, 14. Retrieved from http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf

Fisher, A., Nolan, M., Friedenthal, S., Loeffler, M., Sampson, M., Bajaj, M., ... & Hart, L. (2014, July). 3.1. 1 model lifecycle management for MBSE. In INCOSE International Symposium (Vol. 24, No. 1, pp. 207-229).

Fisher, A., Nolan, M., Friedenthal, S., Loeffler, M., Sampson, M., Bajaj, M., ... & Hart, L. (2014, July). 3.1. 1 model lifecycle management for MBSE. In INCOSE International Symposium (Vol. 24, No. 1, pp. 207-229). Retrieved from http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:model_lifecycle_management_for_mbse_v4.pdf

Griffin, J. (2008). The role of the chief data officer. Information Management, 18(2), 28.

Groth, P., & Moreau, L. (2013). An Overview of the PROV Family of Documents. W3C Working Group Note 30 April 2013. Retrieved from https://www.w3.org/TR/prov-overview/.

Hansen, T., Hardie, T., & Masinter, L. (2006). Guidelines and Registration Procedures for New URI Schemes. Network Working Group. Internet Engineering Task Force. Retrieved from https://tools.ietf.org/html/rfc4395

Huang, J., Gheorghe, A., Handley, H., Pazos, P., Pinto, A., Kovacic, S., ... & Daniels, C. (2020). Towards digital engineering: the advent of digital systems engineering. International Journal of System of Systems Engineering, 10(3), 234-261.

Huang, J., Handley, H., Pazos-Lago, P., Pinto, A., Kovacic, S., Collins, A., & Daniels, C. (2019). Digital Systems Engineering: Concepts, Challenges and Enabling Technologies.

Hybertson, D. W. (2016). Model-oriented systems engineering science: a unifying framework for traditional and complex systems. CRC Press.

INCOSE. 2015. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0

Kruse, B., & Blackburn, M. (2019). Collaborating with OpenMBEE as an Authoritative Source of Truth Environment. Procedia Computer Science, 153, 277-284.

Lankhorst, M. & et al. (2013). Enterprise Architecture at Work: Modelling, Communication and Analysis (2th. ed.). Springer Publishing Company, Incorporated.

Lee, Y., Madnick, S. E., Wang, R. Y., Wang, F., & Zhang, H. (2014). A cubic framework for the chief data officer: Succeeding in a world of big data.

Linkov, I., Trump, B. D., Poinsatte-Jones, K., & Florin, M. V. (2018). Governance strategies for a sustainable digital world. Sustainability, 10(2), 440.

McDermott, T., Collopy, P., Nadolski, M., & Paredis, C. (2019). The Future Exchange of Digital Engineering Data and Models: an Enterprise Systems Analysis. Procedia Computer Science, 153, 260-267.

Mohammad, S. M. (2018). Improve Software Quality through practicing DevOps Automation. Sikender Mohsienuddin Mohammad," IMPROVE SOFTWARE QUALITY THROUGH PRACTICING DEVOPS AUTOMATION", International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.

Nie, Y., Talburt, J., Dagtas, S., & Feng, T. (2019). The influence of chief data officer presence on firm performance: does firm size matter?. Industrial Management & Data Systems.

Nie, Y., Talburt, J., Li, X., & Xiao, Z. (2018). Chief data officer (CDO) role and responsibility analysis. Journal of Computing Sciences in Colleges, 33(5), 4-12.

Object Management Group. 2010. MDA Foundation Model. OMG document number ORMSC/2010-09-06.

PROV_(Provenance). (n.d.). Wikipedia. Retrieved from https://en.wikipedia.org/wiki/PROV_(Provenance)

Schmit, M., Briceno, S., Collins, K., Mavris, D., Lynch, K., & Ball, G. (2016, April). Semantic design space refinement for model-based systems engineering. In 2016 Annual IEEE Systems Conference (SysCon) (pp. 1-8). IEEE.

Sirisomboonsuk, P., Gu, V. C., Cao, R. Q., & Burns, J. R. (2018). Relationships between project governance and information technology governance and their impact on project performance. International journal of project management, 36(2), 287-300.

URL. (n.d). Wikipedia. Retrieved from https://en.wikipedia.org/wiki/URL#Syntax

US DoD. (2018, June). Digital Engineering Strategy, Office of the Deputy Assistant Secretary of Dense for Systems Engineering.

Zafar, F., Khan, A., Suhail, S., Ahmed, I., Hameed, K., Khan, H. M., ... & Anjum, A. (2017). Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. Journal of Network and Computer Applications, 94, 50-68.

Zhang, H., Lee, Y., Wang, R., & Huang, W. (2017). Chief data officer appointment and origin: A theoretical perspective.

Zimmerman, P., & Dahmann, J. (2018, October). Digital Engineering to Support Mission Engineering. In Proceedings of the 21st Annual NDIA Systems and Mission Engineering Conference, Tampa, FL, USA (pp. 22-25).

This page intentionally left blank.

Appendix A Abbreviations and Acronyms

Term	Definition
ASoT	Authoritative Sources of Truth
ATO	Authorization to Operate
CDRL	Contract Data Requirements List
CI	Configuration Items
CONOPS	Concept of Operations
CRUD	Create, Read, Update, and Delete
DE	Digital Engineering
DoD	Department of Defense
IGCE	Independent Government Cost Estimate
MBSE	Model-Based Systems Engineering
MLM	Model Lifecycle Management
MVP	Minimum-Viable Product
O&M	Operations and Maintenance
URI	Uniform Resource Identifier
VV&A	Verification, Validation and Accreditation
W3C	World Wide Web Consortium