Australian Government

Defence

Digital Engineering Modelling Guidebook

Enabling Digital Engineering with Models

Capability Acquisition and Sustainment Group

Document Number:	BM98999753
Issue Date:	04 March 2026
Version:	1.0
Classification:	Official

Endorsements

Endorsed by:

Appointment	Digital Signature
Ms. Asha Mathew Director, Digital Engineering Engineering Technology & Material Logistics Division CASG, Department of Defence	Asha MATHEW Digitally signed by Asha MATHEW Date: 2026.03.04 11:42:56 +10'30'

Document Ownership & History

Document Owner	Directorate of Digital Engineering Engineering Technology & Material Logistics Division CASG, Department of Defence
Subject Matter Expertise	Dr. Ayad Al-Mahturi Digital Engineering Modelling & Systems Lead Digital Engineering Engineering Technology & Material Logistics Division CASG, Department of Defence

Version	Date	Changes
1.0	03 Mar 2026	Original Document

Cancellation Date:

Valid for five years from date of approval. Document to be reviewed by cancellation date.

Major Reviewers

Name	Representing
Digital Engineering Working Group	• Senior Leaders in Engineering, Data, Digital and Cyber representing: Air Force - Dir Cyber Warfare & Networks • Army – Headquarters • Army - Land Network Integration Centre • Australian Submarine Agency - Digital Strategy • Associate Secretary Group - DD Data Mgt & Capability Dir • CASG - Land Engineering Agency • CASG (ETML) – Material Logistics • CASG (ETML) – Systems Engineering • CASG (Joint Systems Directorate) • Defence Digital Group - Eng Branch • Defence Intelligence Group • Defence Science and Technology Group - Research Engineering • Guided Weapons and Explosive Ordnance Group - DD Headquarters • Joint Capabilities Group - C4 (Policy and Strategy) • NAVY - Directorate Navy Engineering • Navy Intelligence & Information Warfare - Program Delivery • Naval Shipbuilding and Sustainment Group - Engineering and Technical • Security and Estate Group - Estate Engineering Policy • Vice Chief of the Defence Force - Force Integration - C4SIR Design Authority
Mr Rajat Sharma	• Digital Engineering, ETML, CASG

Table of Contents

1 Digital Engineering Modelling Guidebook 1
1.1 Purpose 1
1.2 Key Terms 2
2 Use Cases of Models in Defence 5
2.1 Strategy, Concepts, and Planning Cycle 6
2.2 Force Design and Assurance Cycle 7
2.3 Investment and Delivery Cycle 8
2.4 Sustainment and Operations 9
2.5 JFM Model Federation in Defence 9
2.6 Other Uses of Models in Defence 10
3 Models for Acquisition 11
3.1 Models Supporting Capability Definition 11
3.2 Model-Based Acquisition 13
3.3 Model-Based Proposal Assessment 14
3.4 Model-Based and Model-Aided Test and Evaluation 15
3.5 Iterative and Agile Procurement 16
3.6 CDD and ASDEFCON Support for Models in Acquisition 16
4 Models and Their Purpose 18
4.1 What Is a Model? 18
4.2 Types of Models 18
4.2.1 Formal vs Informal Models 18
4.2.2 Physical vs Abstract Models 19
4.2.3 Descriptive vs Analytical Models 19
4.3 Model-Based Design 20
4.4 Simulation vs Modelling 21
4.5 Models, Simulation, and Artificial Intelligence 23
4.6 Models and the Digital Engineering Functional Areas 23
4.6.1 Requirements Management 24
4.6.2 System Architecture and Design 24
4.6.3 Operational Analysis 25
4.6.4 Physics-Based and Mathematical Analysis 25
4.6.5 Physical Design and Specialty Engineering 25
4.6.6 Life Cycle Management 25
4.6.7 Engineering Management 26
4.6.8 Visualisation 26
4.7 What Are the Benefits of Using Models? 26

5 Getting Started Modelling 28
5.1 Focus on the Problem 28
5.2 Methodology, Data, and Tools 28
5.3 Crawl, Walk, Run 29
6 Modelling Standards 31
6.1 Modelling Languages 31
6.1.1 Unified Modeling Language 31
6.1.2 Systems Modeling Language 31
6.1.3 Behavior Modelling Language 32
6.2 Methodologies 32
6.2.1 Object-Oriented Systems Engineering Method 32
6.2.2 Object-Process Methodology 32
6.2.3 ARCADIA Methodology 32
6.3 Architecture Frameworks 33
6.3.1 Department of Defense Architecture Framework 33
6.3.2 NATO Architecture Framework 33
6.3.3 Unified Architecture Framework 33
6.3.4 ArchiMate 33
6.3.5 The Open Group Architecture Framework 34
6.4 Physics-Based and Mathematical Analysis 34
6.5 Simulation Interoperability Standards 34
6.5.1 Distributed Interactive Simulation 34
6.5.2 High Level Architecture 35
6.5.3 US Government Reference Architectures 35
6.6 Other Relevant Modelling Standards 35
6.6.1 Industry Foundation Classes 35
6.6.2 Business Process Model and Notation 35
6.6.3 Standard for the Exchange of Product Model Data 36
6.6.4 Open Services for Lifecycle Collaboration 36
7 Model-Based Systems Engineering 37
7.1 System Architecture and the Digital Thread 37
7.2 Advantages and Challenges of MBSE 37
7.3 Aspects of MBSE Models 38
7.3.1 System Structure 38
7.3.2 System Behaviour 39
7.3.3 Requirements 39
7.3.4 Analytics 40

8 Collaborative Modelling 41
8.1 Model-Based vs Model-Added 41
8.2 Modelling Environments 42
8.2.1 Server/Client Environments 42
8.2.1.1 Collaborative Real-Time Editing 42
8.2.1.2 Model Lock/Unlocking 42
8.2.1.3 Highly Modular Models 43
8.2.2 Browser-Based Environments 43
8.2.3 Distributed Environments 44
8.2.4 Alternate Visualisations 44
9 Model Federation 46
9.1 Differences in Modelling Structures 46
9.2 Inheritance and Reuse 47
9.3 Methods of Model Federation 48
9.4 Use Cases of Model Federation 49
9.4.1 Shared Modelling Framework for Mission Engineering 49
9.4.2 Using Industry Models for Proposal Assessment 50
9.4.3 Industry Traceability to Defence Specification 51
10 Model Exchange 52
10.1 Remember the Purpose 52
10.2 Define a Schema 52
10.3 Determine Transformations 53
10.4 Reference and Master Data 53
11 Model Verification and Validation 54
11.1 Verification of Models 54
11.2 Validation of Models 54
12 Model Curation and Reuse 56
12.1 Model Purpose: Should It Be Reused? 56
12.2 Understanding Model Maturity 56
12.2.1 Approach to Measuring Maturity 57
12.2.2 Application in Practice 57
12.3 Model Curation and Storage 58
13 Model Configuration Management and Governance 59
13.1 Scaled Configuration Management of Models 59
13.2 Configuration Management of Model Changes 59
13.3 Agile Model Management 60
13.4 Configuration Management Features of Tools 60

14 Modelling Best-Practice Tips 62
15 Conclusion 63
Point of Contact 64
Annex A Acronyms, Abbreviations 65
Annex B References 67

Figures

Figure 1: Examples of Models Connected via Shared Authoritative Data (from Digital Engineering Strategy 2024) 1
Figure 2: Foundational Guidance for Digital Engineering 2
Figure 3: How it fits together SE, ME, DE, MBSE, and Models and Simulations 3
Figure 4: Digital Engineering Supporting the ODCS 5
Figure 5: Joint Force Model 6
Figure 6: ODCS – Strategy, Concepts, and Planning Cycle 6
Figure 7: ODCS – Force Design and Assurance Cycle 7
Figure 8: ODCS – Investment and Delivery Cycle 8
Figure 9: JMD Joint Force Model (adapted from VCDFG) 10
Figure 10: Model Supporting CDD 12
Figure 11: Model-Generated CDD 13
Figure 12: Model-Based Specification and Proposals 14
Figure 13: Model-Based Proposal Assessment 15
Figure 14: Analog to Digital to Analytic Models 20
Figure 15: Models across Defence 21
Figure 16: Digital Engineering Functional Areas 24
Figure 17: Key aspects of Digital Modelling 28
Figure 18: Crawl (Documents to Data) 30
Figure 19: Walk (Digital by Default) 30
Figure 20: Run (Integrated Environment) 30
Figure 21: Model-Added vs Model-Based 41
Figure 22: Collaborative Real-time Editing 42
Figure 23: Model Lock/Unlock 43
Figure 24: Highly Modular Models 43
Figure 25: Browser-based Environments 44
Figure 26: Distributed Environments 44
Figure 27: Alternate Visualisations 45
Figure 28: Integrated Model Example 47
Figure 29: Bottom-up Federation 48

Figure 30: Top-down Federation 49
Figure 31: Seed, Base, and Mission Model Representations (From Ref B8) 50
Figure 32: Model-Based Proposal Assessment 51
Figure 33: Traceability Between Response and Design 51
Figure 34: Model Schema 52
Figure 35: Model Transformation 53
Figure 36: Model Maturity Metrics 57
Figure 37: Model Maturity Assessment Method for Use on Projects (From Ref B83) 58
Figure 38: Configuration Management of Models 60
Figure 39: Agile Model Management 60

How to Read This Document

Each section (top-level heading) in this document is intended to be a separate web page and has been written with this in mind.

As such, the sections in this document are written so that they can be read as standalone. Some content may feel repetitive where it is re-emphasised in different sections, acronyms being expanded in each section, and using hyperlinks for references rather than cross-references to reference lists.

Some group related sections have been arranged sequentially within this document. However, it is not intended to be read in any particular order.

1 Digital Engineering Modelling Guidebook

The Defence Digital Engineering Strategy 2024 (DDES) described Defence's vision for how digital engineering will be adopted to support data-driven decision making and increase the speed to delivery of Minimum Viable Capability. This focuses on the use of models that access authoritative data as shown in Figure 1.

This modelling guidebook supports Goal 2 of the DDES by providing guidance and best practices for models and modelling efforts in Defence. The second goal of the strategy emphasises the development, integration, and application of digital models to inform decision-making and design. The third goal highlights that trusted, enduring, and authoritative data is a fundamental input for accelerating capability delivery.

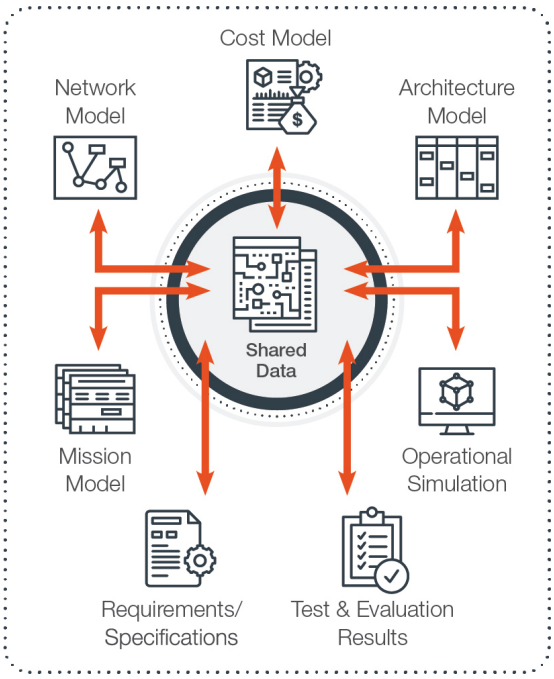

Figure 1: Examples of Models Connected via Shared Authoritative Data (from Digital Engineering Strategy 2024)

1.1 Purpose

The purpose of this guidebook is to assist newcomers to digital modelling and Digital Engineering more broadly, and in understanding how models are applied across the One Defence Capability System (ODCS).

As this guidebook informs the reader on models, it will also explain the 'so what' for use and application in the Defence context. This will assist the reader to better understand where and how models can be used to improve design for the realisation of the joint force systems and sub-systems.

This guidebook is part of Defence's foundational guidance provided to support Digital Engineering and development of models. This foundational guidance documents, as shown in Figure 2, supports the overall Defence Digital Engineering Strategy 2024 and Roadmap. This guidebook will provide tailoring guidance for the Groups and Services when applying Digital Engineering.

This guidebook should be read in conjunction with the other foundational guidance documents, particularly the Digital Engineering Framework, which provides initial steps and shared understanding to accelerate adoption of Digital Engineering practices across Defence. It is expected that this foundational material will be further developed and expanded through specific guidance tailored to Defence use cases. These use cases will address key Defence analysis and

decisions across the One Defence Capability System (ODCS) to include Joint Mission Design (JMD), Acquisitions, and Sustainment.

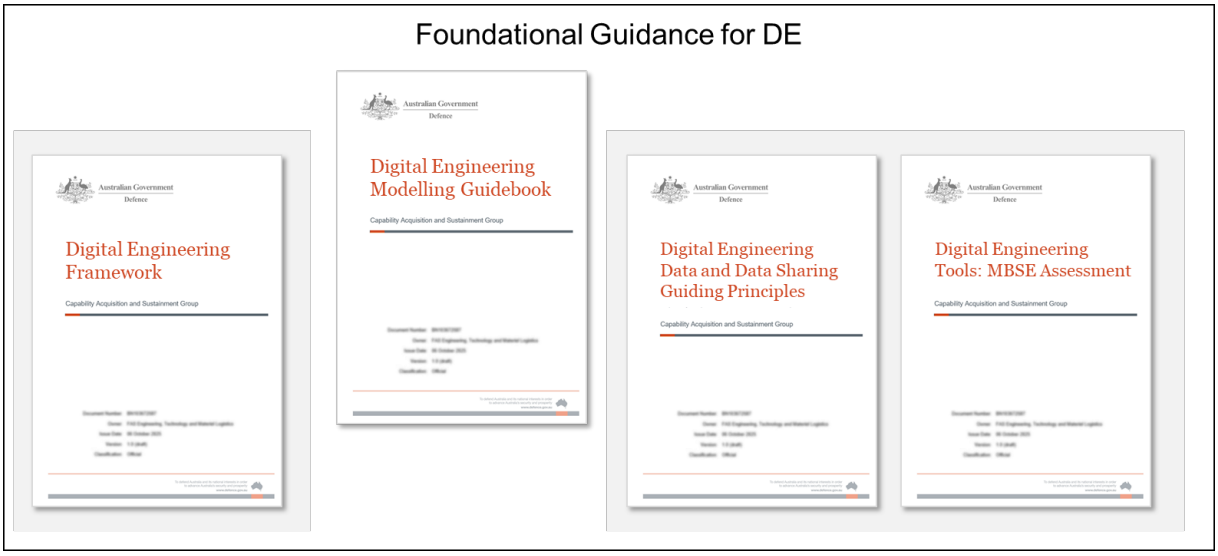

Figure 2: Foundational Guidance for Digital Engineering

The initial release of the guidebook does not address all potential use cases, but instead provides general modelling guidance and uses nominated use cases centred on the ODCS and the evolving models supporting JMD. The guidebook is organised into two major topic areas: Defence specific guidance aligned to the ODCS, and Model-based acquisitions approaches. This should be augmented with other modelling approaches currently used across Defence. This guidebook will be maintained via the Digital Engineering Knowledge Hub (DE KHub). The knowledge hub is available to all Defence personnel, via the Defence Adele (O:S) on the Defence Protected Network (DPN) for on-boarding and self-enrolment.

1.2 Key Terms

Modelling is useful to support many activities across the ODCS, such as Systems Engineering, Mission Engineering, detailed design and test & evaluation. This guidebook aims to align Defence basic terminology regarding models and modelling while providing guidance for developing models that are useful and effective. An initial focus of modelling will be on the Joint Force model and capability development activities within Defence. Models supporting Systems Engineering and Mission Engineering are most closely relevant to these activities. It is essential to consider the distinctions between Systems Engineering and Mission Engineering and how Digital Engineering can support these applications.

Systems Engineering (SE) is a transdisciplinary[1] and integrative approach that enables the successful realisation (ie design, integrate and manage), use, and retirement of engineered systems. It applies systems principles and concepts, as well as scientific, technological, and management methods. This approach encompasses individual systems, sub-systems, and systems of systems (SoS).

Mission Engineering (ME), as described in the DDES, is the deliberate planning, analysing and integrating of current and emerging operational capabilities to achieve desired warfighting mission

[1] Transdisciplinary and interdisciplinary refer to the both System and Mission Engineering encompassing many distinct engineering and disciplinary boundaries. The major distinction is for Mission Engineering the mission is the system, and therefore includes mission related disciplines not considered in traditional systems engineering.

effects. Mission Engineering (ME) is the application of Systems Engineering where the mission is the system. ME is used to design missions from the Joint Concept Framework.

While these two disciplines may appear similar and both utilise Digital Engineering and models, they are distinct in their focus. In Mission Engineering, the mission itself becomes the system of interest. Systems Engineering focuses on designing systems (including SoS) to meet specified technical performance metrics. Mission Engineering goes a step further by evaluating the performance of the SoS in achieving mission or capability objectives when implemented in a simulated, realistic scenario. Mission Engineering assesses whether the SoS achieves the expected or desired effects and determines whether those effects contribute to mission success. In the Defence context, the analysis and results of Mission Engineering serve as inputs to system design (such as requirements definition) as well as Force Design decisions.

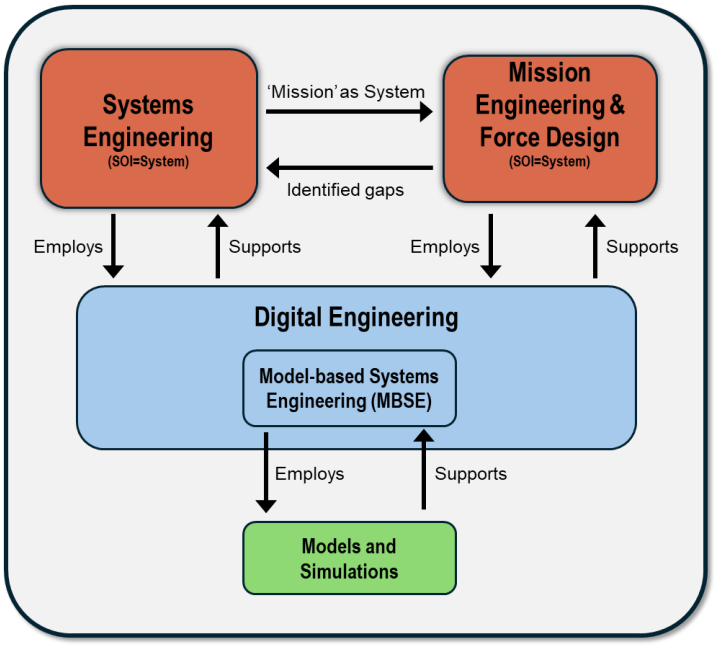

Figure 3: How it fits together SE, ME, DE, MBSE, and Models and Simulations

While Systems Engineering and Mission Engineering are methods which Defence *employs*, Digital Engineering is a supporting means for *how* they are achieved. As described in the DDES, **Digital Engineering** is an integrated digital approach that uses authoritative sources of systems data and models as a continuum across disciplines to support lifecycle activities from concept through to disposal. The DDES describes how Digital Engineering supports more than purely Systems Engineering or Mission Engineering.

Models are part of Digital Engineering, and **Model-Based Systems Engineering** (MBSE) is an example (but not the only one) of how models can support Defence business processes. MBSE is a component of Digital Engineering which can support both Systems Engineering and Mission Engineering. This guidebook considers modelling more broadly than MBSE, and also the wider applications of Digital Engineering. More information on MBSE is available as a separate topic.

Systems Modelling Language (SysML) is a widely accepted international standard language used by MBSE tools. SysML is a general-purpose graphical modelling language for specifying, analysing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. As a modelling language, SysML is only one aspect of a MBSE modelling effort. The other aspects, namely methodology and tools are also important considerations.

The relationships between Systems Engineering, Mission Engineering and Force Design, Digital Engineering, MBSE, and SysML are illustrated in Figure 3. Specific guidance for MBSE and other types of modelling efforts applied to Defence business processes is expected to be developed over time. This guidance will either supplement or expand upon the content of this guidebook. It is important to recognise that both Systems Engineering and Mission Engineering, and Force Design,

have traditionally been conducted without Digital Engineering or MBSE. These digital approaches enhance and accelerate decision-making in these areas but do require investment in workforce (people), processes, and environment (tools/technology).

This guidebook is not a static resource; it is intended to evolve and mature in collaboration with Defence Groups and Services.

2 Use Cases of Models in Defence

Models of various types play a critical role in supporting the development of engineering artefacts, improving decision making processes, and accelerating project success across Defence. The Digital Engineering Strategy 2024 and Digital Engineering Framework included conceptual digital engineering capability life cycle model (shown in Figure 4) based on the One Defence Capability System (ODCS Manual versions 1 and 2, respectively). It describes how digital engineering, using models, might support the life cycle stages and cycles of the ODCS.

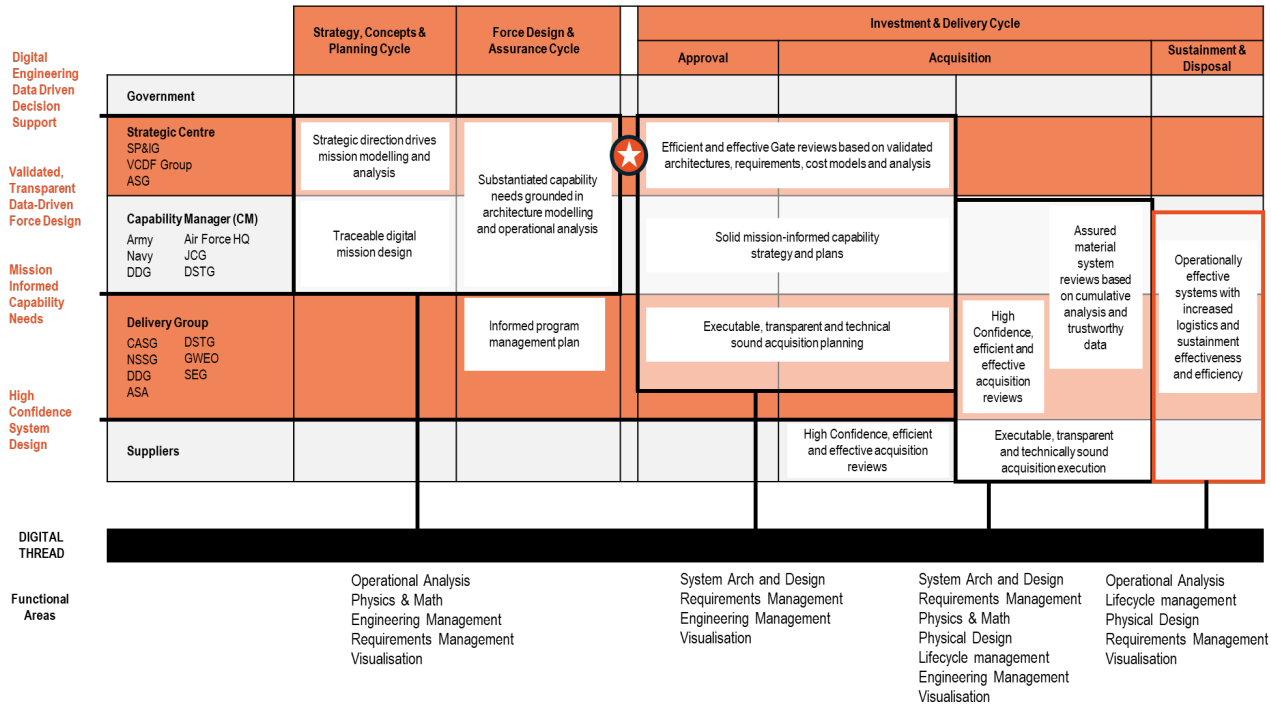

Figure 4: Digital Engineering Supporting the ODCS

This topic focusses specifically on how **models** can support digital engineering across the ODCS. This topic is important because, as part of the ODCS life cycle, it is possible to identify certain models that should be shared across Defence. Specific topics discuss each cycle in more detail. Detailed technical guidance for how models are implemented (adoption of specific methodologies, data, and tools) will be managed and owned by business process owner and may be prescribed for adoption to support that business process.

For example, Vice Chief of the Defence Force Group (VCDFG) is using these concepts to develop the Joint Force Model (JFM) which is based on the suite of Joint Mission Designs (JMDs) as derived from strategic guidance. The JFM will support Joint Force decision making including investment prioritisation and provide the basis for defining the scope and capability needs of those investments. The JFM can also be used to assess the potential contribution of emerging technologies through the lens of joint missions. This will become the centre piece to align Joint Force needs directly with Projects and delivered capabilities as shown in Figure 5.

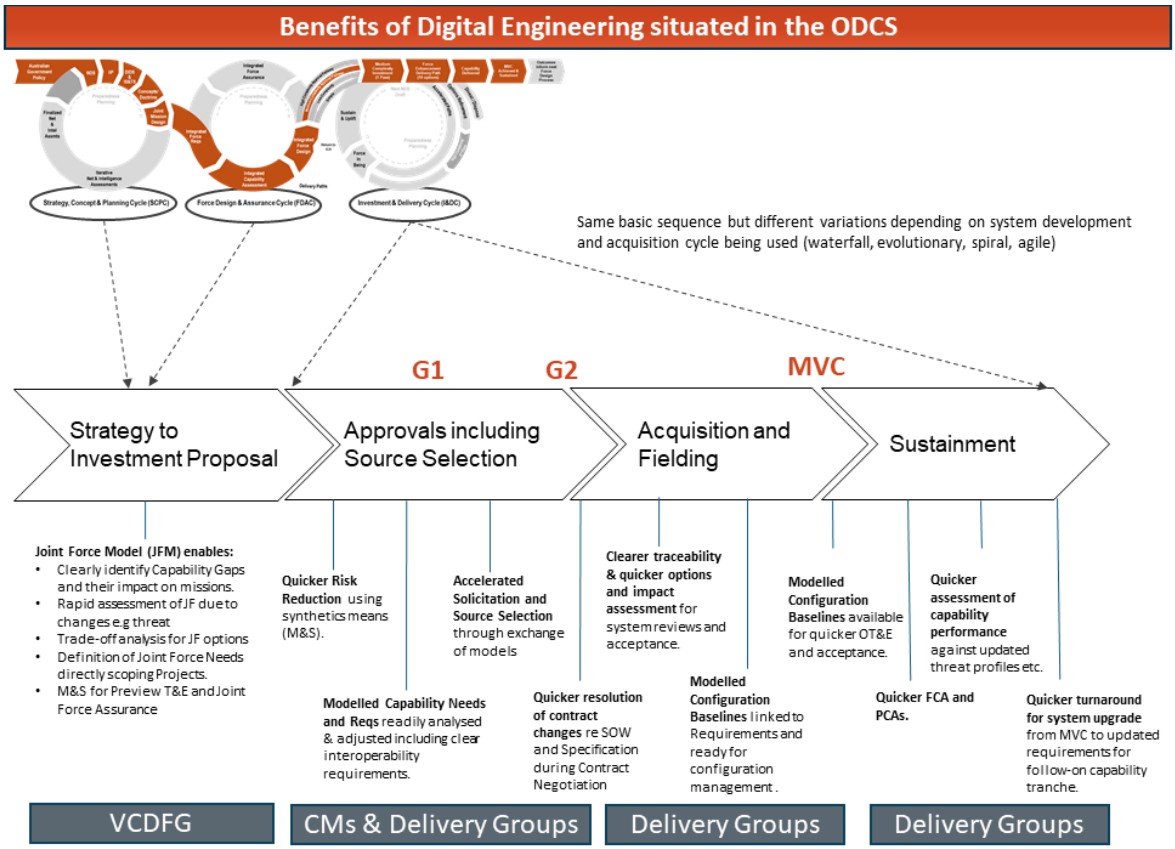

Figure 5: Joint Force Model

A key advantage of wider model adoption, and digital engineering more broadly, is the ability to connect models and data across the different cycles of the ODCS, building a “digital thread” of traceability from strategy through to detailed design and sustainment. The development of models to support Defence’s work helps develop and mature the datasets supporting the ODCS, and when connected, can provide more meaningful insights and the Joint Force Model is the initial step in establishing this thread.

2.1 Strategy, Concepts, and Planning Cycle

The Strategy, Concepts, and Planning Cycle (SCPC), shown in Figure 6, takes the Government’s priorities, develops joint concepts and mission designs, and defines Integrated Force Requirements to inform the Force Design and Assurance Cycle (FDAC).

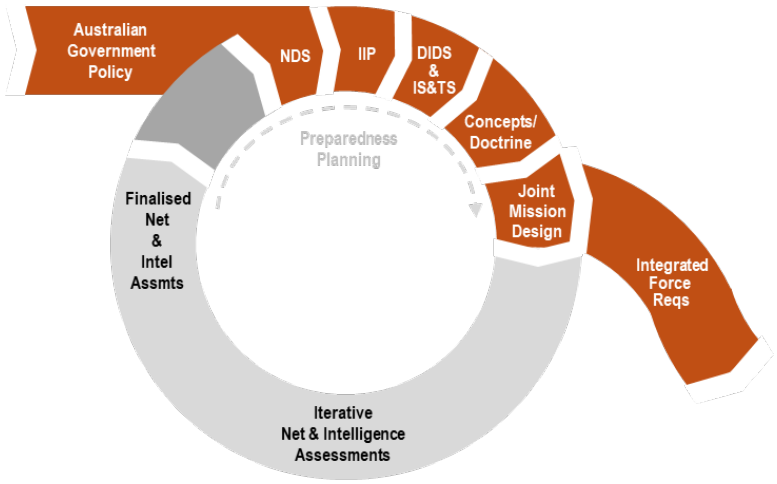

Figure 6: ODCS – Strategy, Concepts, and Planning Cycle

Mission engineering using operational analysis and system architecture models supports JMD by identifying gaps in the current Force-in-being and evaluating the Objective and Future Force to determine their capacity and capabilities to meet strategic objectives. The resultant Integrated Force Requirements are derived and substantiated from this model-based analysis.

JMD aligns with the strategic direction outlined in the National Defence Strategy, concepts, and doctrine, leveraging digitised documents within these models. These model-based approaches can extend beyond textual links, transforming concepts and doctrine into structured digital models for greater clarity and utility. This enables the synchronisation of documents and models.

Data generated across these activities and the other cycles will be able to be integrated through a shared data exchange environment, connecting authoritative sources of truth shared across the ODCS.

2.2 Force Design and Assurance Cycle

The FDAC, shown in Figure 7, reviews Defence's capabilities against the strategic direction, Joint Mission Designs, and Integrated Force Requirements via two primary activities:

1. Integrated Capability Assessment (ICA) identifies gaps and opportunities and conducts trade-offs between capability and affordability of current and future projects in the Integrated Investment Program, and
2. Integrated Force Assurance (IFA) ensures existing projects and programs are meeting the needs of government.

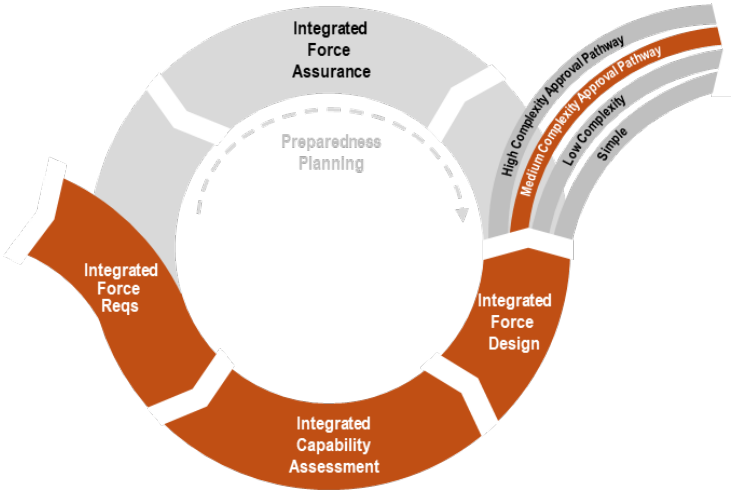

Figure 7: ODCS – Force Design and Assurance Cycle

Authoritative project and program data are captured and standardised in the FDAC through CapabilityOne. Physical asset data is captured in life cycle management tools supporting sustainment (such as Enterprise Resource Planning (ERP)) in the Investment and Delivery Cycle (I&DC). Joint Force architecture models are being developed through Joint Mission Design models in the SCPC. The effects (military outcomes) are derived from the National Strategy to inform integrated Force Requirements. Bringing these together will drive data-driven decision making in the FDAC.

Mission engineering using system architecture, operational analysis models, and Modelling and Simulation (M&S) can be used to support both ICA and IFA. For ICA, combining the Joint Mission Design architecture models, financial analysis models, and physical models using authoritative sources (such as CapabilityOne and ERP) can support understanding of gaps and opportunities and inform the trade-off decisions for investments. The impact on mission outcomes of prioritising or deprioritising projects can be digitally assessed using mission engineering techniques to support informed risk-based decision making.

For IFA, rather than the larger mission models used for ICA, individual mission analyses on identified problems of concern could provide a data-driven approach to the analysis.

Models can help establish and justify the traceability from Integrated Force Requirements through ICA to the Integrated Capability Directives (ICDs). This can provide Capability Managers with valuable context on the content in the ICDs by exposing the analysis through the previous cycles that led to their definition.

Depending on the analysis and the available model data, physics-based analyses and digital twins (if available) could enable operational analysis for ICA and IFA to a greater level of fidelity.

2.3 Investment and Delivery Cycle

The I&DC delivers and sustains products using projects and programs led by Capability Managers. There are various approval, delivery, and sustainment paths, as shown in Figure 8, depending on the product and the underlying maturity of Fundamental Input to Capability (FIC) elements to be delivered.

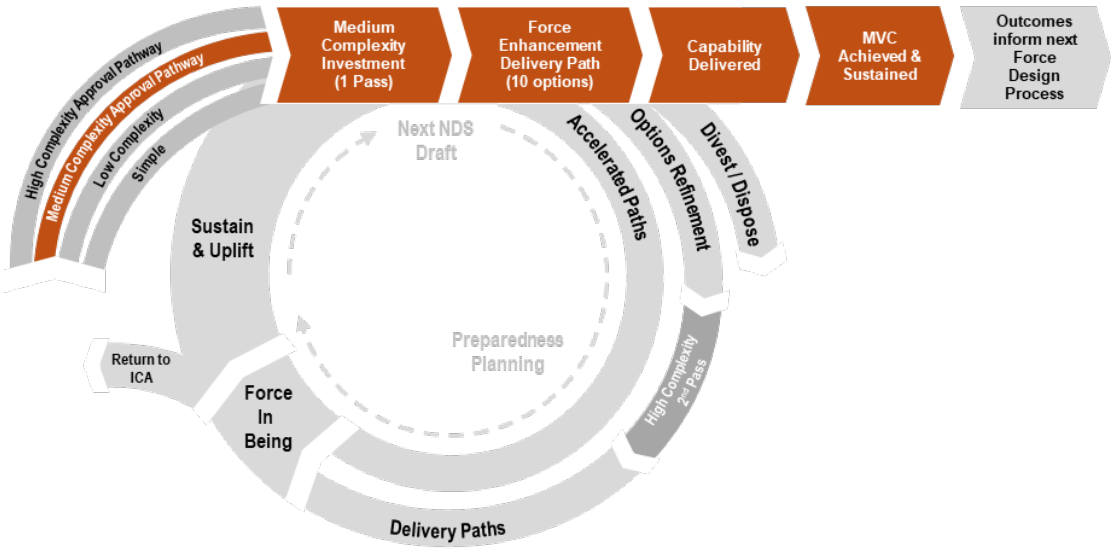

Figure 8: ODCS – Investment and Delivery Cycle

The use of models is critical across the I&DC to accelerate capability delivery and prioritise Minimum Viable Capability, required under the 2024 National Defence Strategy. The types of models applied will vary depending on the capability and the phase of the life cycle.

When obtaining government approval, models can be useful to give data-driven insights when assessing capability options, lowering risk, and informing assessments of value for money. Models can do this, as they reduce uncertainty in the transition to force by addressing this uncertainty in the digital world first. Models can demonstrate the technical viability of a proposed solution much earlier in the system lifecycle and therefore identify technical performance (or integration) concerns earlier to reduce both technical and non-technical (cost, schedule, etc.) risk.

During acquisition, system architecture and design models supporting Model-Based Systems Engineering (MBSE) and, connected to requirement models, are particularly effective for defining capability and facilitating collaboration between Defence and industry. Refer to the topic on models for acquisition for more information.

Once a capability is introduced into service, life cycle management models, alongside physical design and specialty engineering models, capture and develop essential data about the procured materiel and its operational characteristics. Critically, models should be updated and shared to ensure across ODCS that the “as conceived” by SCPC, “as employed” by FDAC, and “as delivered” by ID&C are updated, so each part of the cycle understands the force capability and intent.

Throughout sustainment and disposal, digital models provide traceability to manage change, address obsolescence, and enable capability upgrades. For iterative or agile acquisitions, models

ensure requirements traceability, design updates, and operational analysis, allowing validated requirements to drive improvements and maintain effectiveness against evolving threats.

Operational analysis models also support sustainment by assessing capability suitability against updated threat environments, informing future capability development and integration within the SCPC and FDAC.

2.4 Sustainment and Operations

The breadth of uses of models (in many forms) to support operations is beyond what can be described in this modelling guidebook, and their implementation details are sensitive. However, there are ways models can support operations that can be described generally.

Operational analysis techniques that are used in mission engineering to aid system-of-systems design can similarly be applied to mission planning and to analyse actual operations. Modelling and simulating real operations can provide more in-depth analyses of alternatives. This may include Live, Virtual, and Constructive in a single integrated environment. “Live” is real people on real systems, “Virtual” is live people on virtual digital systems, and “Constructive” is virtual people on virtual systems. For example, Modelling and Simulation can lead to better contingency planning before operations or more in-depth analysis of lessons learned after operations. Critically, lessons learned may identify new information that impact re-planning efforts and may impact beyond a single system (e.g., new capability deployed by adversarial forces or new employment/capability to integrate).

Digital twins with other physics-based and mathematical analysis models, can take live data from operations, feed them into detailed mission models, and simulate outcomes for timely informed decision making. Artificial intelligence (AI)–enabled systems can use the output of models to assist in their training.

Life cycle management models, like those implemented in ERP, are useful to support logistics. Physical design models can assist maintenance in deployed environments. There are also emerging potential integrations of models with operations (e.g., In-theatre additive manufacturing [3D printing)] which rely on physical design models [CAD]). These may be provided in earlier stages and can be updated and 3D printed (manufactured) in extremely short timeframes to forward deployed locations.

2.5 JFM Model Federation in Defence

Defence has targeted the development of a JFM using Federated Modelling based on the Shared Modelling Framework. JFM is currently under development but will assist to standardise models across Defence to support joint mission design. This is not the exclusive use case for modelling, but is a critical one aligned with the Defence strategic missions under Concept Aspire available on Defence Protected Network. This federated modelling environment comprises of multiple layers of contributions across Defence to a Joint Force Model supported by and supporting domain / capability models, individual system models, and solution (or proposed solution) models as shown in Figure 9.

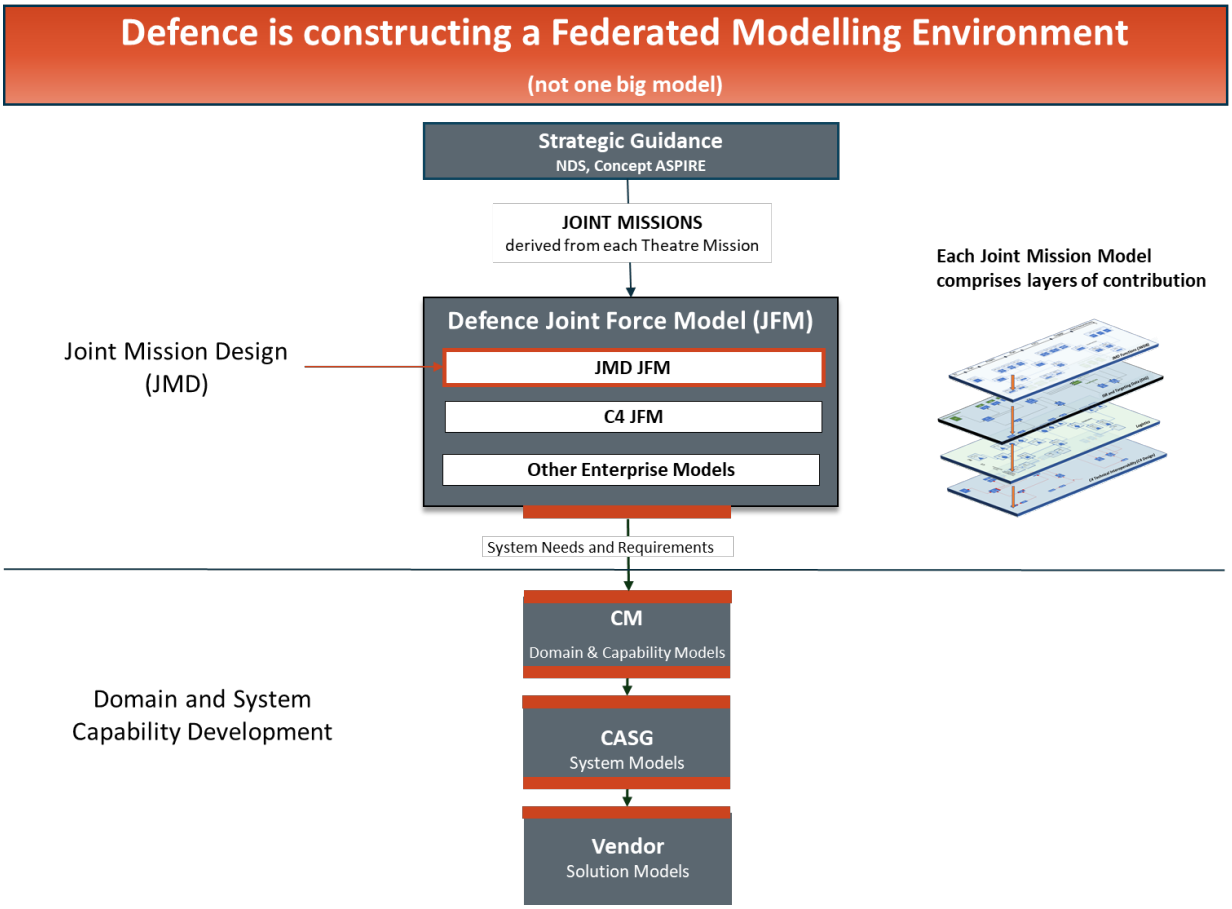

Figure 9: JMD Joint Force Model (adapted from VCDFG)

This creates a digital thread from the joint missions to the delivered and sustained systems. It is anticipated that as Digital Engineering is advanced across Defence that similar digital threads and enterprise models will be federated and linked. VCDFG as the owner of the JFM will develop specific policies and standards associated with this process, supporting the Joint Mission Design Guide. As this leverages the SysML-based Shared Modelling Framework, SysML based tools will be used to support JFM across Defence.

2.6 Other Uses of Models in Defence

Business Process Modelling is used in the Enterprise Business Management System to capture how Defence operates as an organisation using structured data. These models are presented as intranet pages and help communicate policy and processes. Connecting these models to analytic models to simulate business processes may assist to identify ways of working that are more efficient.

3 Models for Acquisition

The One Defence Capability System (ODCS) Manual splits the Investment and Delivery Cycle (I&DC) into three parts: acquisition, introduction into service, and sustainment and disposal. This topic focusses on acquisition—models to support Defence engaging with industry to procure materiel.

The Defence Digital Engineering Strategy 2024 describes Defence's objective to use models to support acquisition decisions. The functional areas most relevant are system architecture and design, requirements management, operational analysis, physics-based and mathematical analysis, and physical design and specialty engineering. The most relevant functional areas depend on the ODCS Delivery Path and other project-specific considerations.

Within Defence, models to support acquisition typically focus on the following:

- Capability definition, including:
    - Describing the operational concept,
    - Refining the operational needs, as initially described in the Integrated Capability Directive (ICD),
    - Conducting analytics of a system and its behaviour, and
    - Integrating the product into the Joint Force;
- Analysing and comparing different industry proposals;
- Model-based and model-aided test and evaluation (T&E); and
- Supporting agile or iterative procurement.

For each of these, a central system architecture model supporting Model-Based Systems Engineering (MBSE) is often core to the engineering effort. This model, perhaps used in conjunction with others, provides a powerful tool to aid decision making.

Model use in industry varies significantly based on the focus of the organisation. For example, prime contractors of complex materiel will use life cycle management models to connect system architecture and design models with models for physical design and specialty engineering. Smaller contractors might work with only specialty engineering tools. Civil and structural engineering firms may focus more on physical design models using Building Information Modelling (BIM) and Computer-Aided Design (CAD).

3.1 Models Supporting Capability Definition

The primary digital engineering functional areas supporting capability definition in Defence are system architecture and design and operational analysis, supporting MBSE and mission engineering.

Beyond the general advantages of using models to support engineering, using models in acquisition helps to left-shift risk by identifying potential issues earlier in the project and helps identify key areas of complexity or risk in the system. For example, identifying where artificial intelligence (AI)–enabled systems might be used may lead to a greater emphasis placed by the project on T&E and the evidence of performance of that component of the system.

Much of the content described in the CASG-2-MANUAL (E&T) 12-3-003 Capability Definition Documentation (CDD) Guide is readily able to be modelled using architectural modelling. Capturing and describing this information in a modelling language and tool assists teams in developing a document suite that is consistent and coherent.

In cases where the maturity level in digital engineering and MBSE is low, traditional document-centric contracting mechanisms between Defence and industry can continue to be used as they have previously. Diagrams in the documentation are taken from the modelling tool (Figure 10), and while not forming part of the contractual documents, the model should be provided to industry to provide additional context.

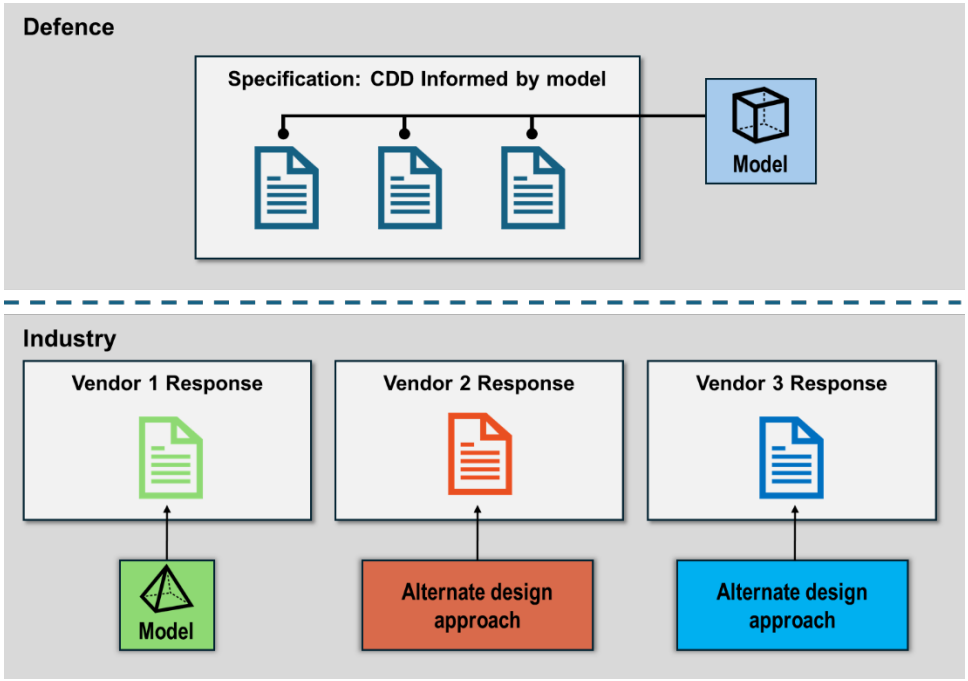

Figure 10: Model Supporting CDD

Many MBSE tools have features that allow CDD to be developed out of the tools, either partially or in full (Figure 11). Teams that have adopted a model-based approach to their work benefit from using these features to support document generation, rather than manually determining which diagrams or text needs to be updated to align with the model.

These features typically take some effort to configure so that the correct content is populated in the correct document template and to the correct section of the document. This overhead is typically worthwhile for delivery paths where Defence's need is not well understood, regular changes to the design are expected, or the documents go through multiple revisions over the course of the project. Additionally, this investment allows dependency analysis to allow key features prioritisation for a capability. There is potential for the tool to be used to verify high-risk features earlier in the design and facilitate earlier system delivery.

When documentation is so closely linked to the model content itself, industry benefits strongly from receiving the model alongside the published documents, which should be provided in an agreed-upon format. This permits industry to be able to directly trace their proposed solutions to the Defence CDD and model, as well as potentially create a response model. This acquisition does not explicitly require the industry partner to leverage digital engineering – partners which do not employ DE are able to use documents as they would have previously.

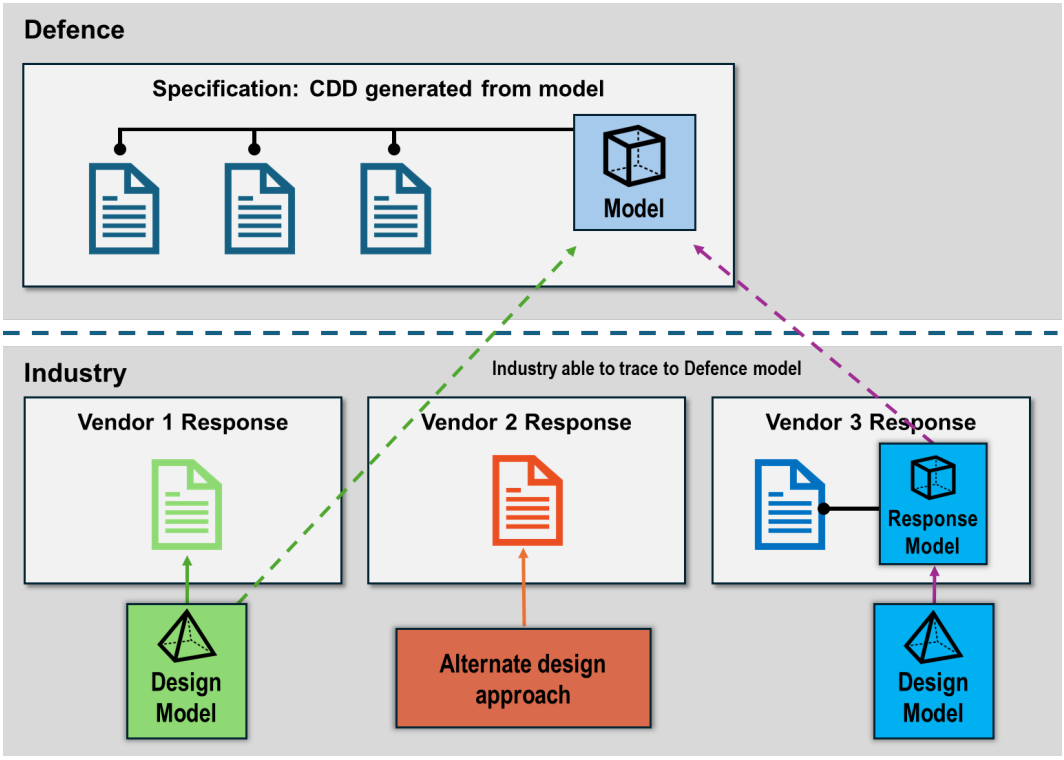

Figure 11: Model-Generated CDD

With a description of the system architecture in a model, the model can be used to inform further analysis, such as parametric analysis; safety analysis; cyberworthiness assessments; and Reliability, Availability, and Maintainability (RAM) analysis. The Reliability, Availability and Maintainability Manual (RAMMAN) (Ref 25) supports this by guiding specialist personnel in identifying user RAM requirements, planning and executing RAM activities, and aligning with industry practices to support Defence materiel. For particular problems of interest, physics-based and mathematical analysis using more advanced tools can readily build on the data in the architecture model.

Maintaining a model through the acquisition process helps Defence manage change. The impact of proposed changes on achieving Minimum Viable Capability is more readily shown—either by executing updated models to understand impact across multiple dimensions (e.g., costs, performance, schedule, etc.), or through traceability in the model to the ICD.

3.2 Model-Based Acquisition

Model-based acquisition refers to having a model to describe the boundary between Defence and industry. The model, rather than output documentation, is authoritative in describing Defence's needs and requirements. The operational concept and requirements exist in the model and are primarily communicated to industry in the model.

Delivery paths using the traditional Australian Standard for Defence Contracting (ASDEFCON) materiel acquisition approach (involving a Defence specification and industry response) provide a model as the specification, and require tenderers to respond, as depicted in Figure 12. Tender responses include models or data structures in a format specified in the Tender.

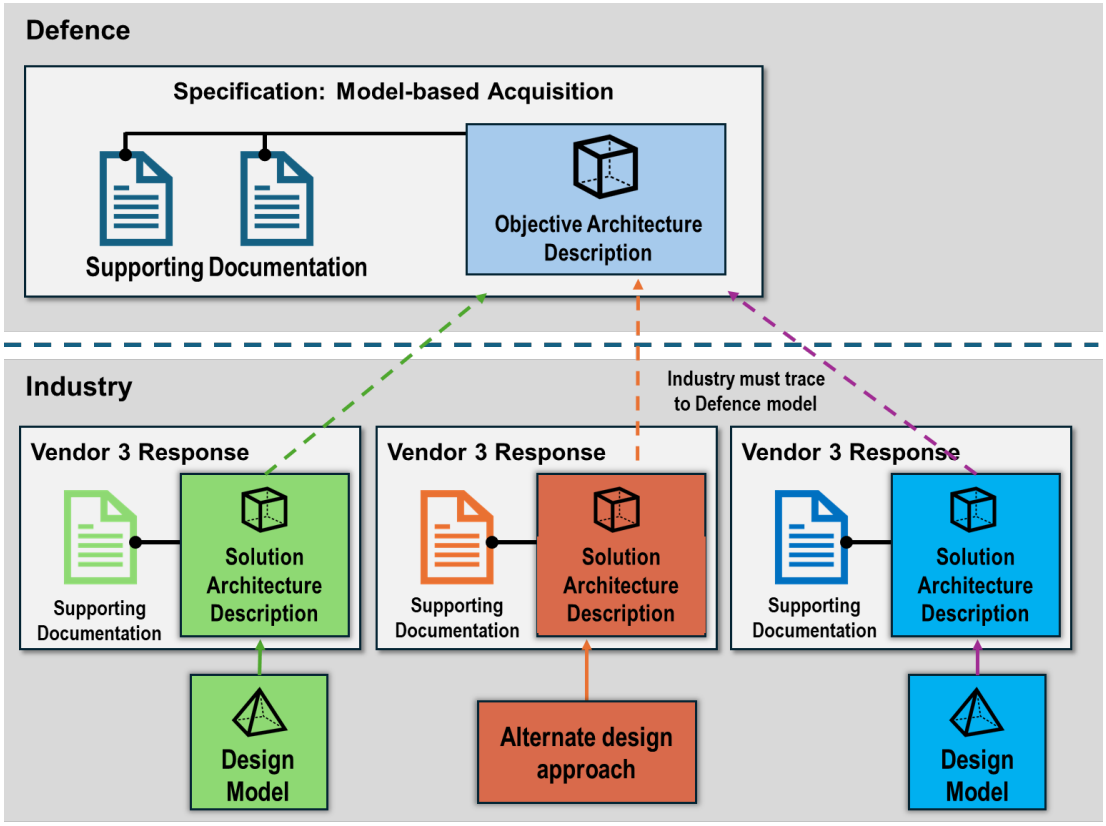

Figure 12: Model-Based Specification and Proposals

This approach has complicating factors that need to be considered, such as:

- Determining and clearly defining the specification and required response formats;
- Tenderers having competency to respond in formats specified in the tender (particularly if that constrains the specific tools the tenderer must use);
- Associated financial burden on tenderers, particularly small and medium enterprises; and
- Whether tenderers' Intellectual Property (IP) will be exposed through response models and how it will be protected. Some modelling practices are able to shield most of this information from the response models.
- Specific restrictions related to ITAR licensed / FMS restricted technical data and designs.

The Object Management Group has a community dedicated to resolving these challenges and are designing a template model structure and associated guide to facilitate Model-Based Acquisition using Unified Architecture Framework and Systems Modeling Language models. It brings together the lessons learned from the United States' Department of Defense/industry and other customer/supplier acquisitions to help the broader community and standardise approaches.

3.3 Model-Based Proposal Assessment

Where the system architecture models are either executable or linked with other analytic models (e.g., physics-based and mathematical analysis models) and tenderers are asked to respond using models, there is an opportunity to conduct model-based proposal assessments.

As depicted in Figure 13, this approach takes tenderer response models, simulates them within Defence's executable model(s), and compares outputs to inform the selection process. This requires Defence to define assessment metrics in the model and requires tenderer responses to include the necessary inputs to calculate those metrics. This may also include specification of a specific language e.g. SysML or interface formats in the tender.

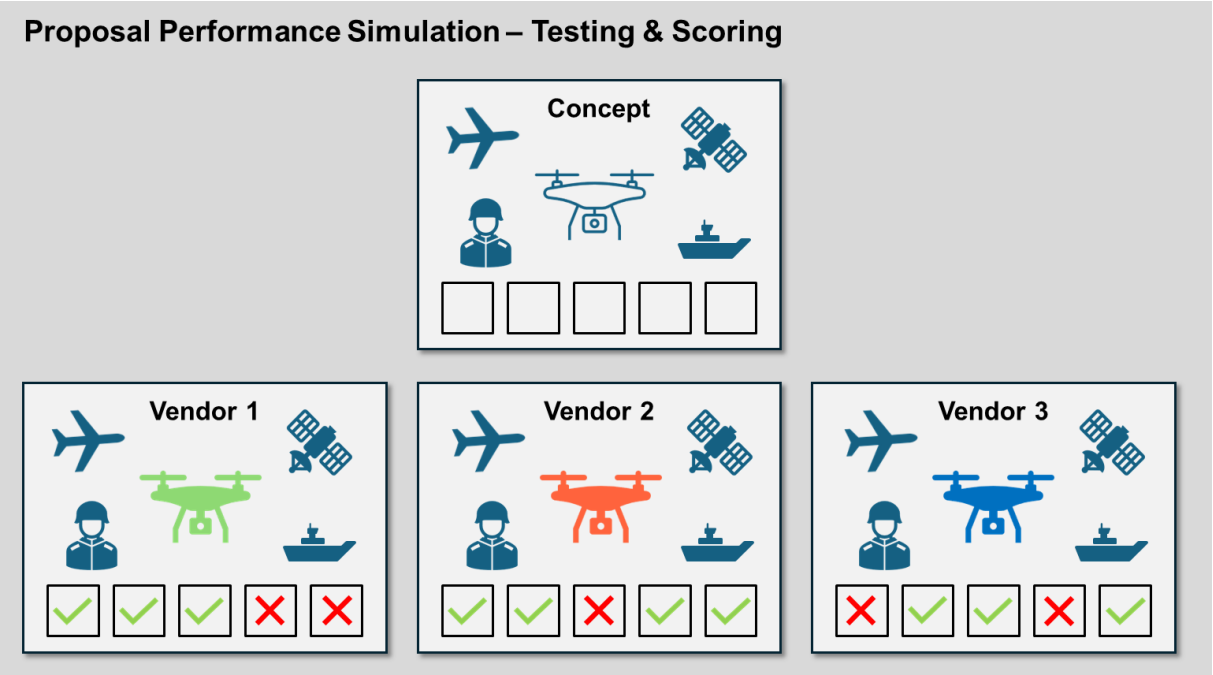

Figure 13: Model-Based Proposal Assessment

The types of metrics that might be used include:

- Key Measures of Effectiveness (MOE) and/or Measures of Performance (MOP);
- Space, weight, and power (SWaP) optimisation;
- Mission success metrics: calculated based on operational analysis models and M&S with parameters of the system of interest provided by tenderers;
- Requirement satisfaction metrics; and
- Reliability, availability, and maintainability (RAM) metrics.

Other metrics can be found in Section 12.2.1. Using models for proposal assessments allows for rapid re-assessments based on:

- Emerging priorities
- Updated tenderer responses, or
- Assessing the expected feasibility of newly defined use cases that were not in the original plan, but have been determined required due to changing operating conditions and/or validated mission need
- Assessment against higher classification threats.

Use of models to directly assess proposed models permits the vendors solutions to be validated against approved and well-known standards. As Defence pivots to a Minimally Viable Product (MVP)-based and a more agile capability development, the Model-based Proposal Assessment should align well and provide additional value.

3.4 Model-Based and Model-Aided Test and Evaluation

System architecture models are useful to support T&E by connecting requirements, architecture, and test data together to show how requirements are being met by design and verified in test. This model-based approach helps manage the complexity of modern test programs. Models can also help identify parts of the system with greater complexity or risk, to plan for evaluations that are most effective and account for system components that introduce variable or hazardous performance.

The Defence Test and Evaluation Manual defines the main types of T&E activities conducted in Defence. Advanced, high-fidelity models, particularly physics-based and mathematical analysis models, can support each of these types of T&E by conducting tests in a virtual environment—

within models instead of the real world. This model-aided T&E approach effectively allows more tests to be verified using analysis, rather than by demonstration or test.

An advantage of model-aided T&E is the ability to verify earlier. The performance of the system can be assessed based on design in the virtual environment prior to manufacture. As the design progresses, virtual verification can be re-run quickly and relatively cost effective to assess the impact of design changes.

Critical to building confidence with the accuracy of the simulated outcome is having high-fidelity models built for the purpose of T&E. These models that accurately represent the system's structure, behaviour, and performance characteristics at a level of detail sufficient to support decision-making and risk assessment, and that have undergone model verification and validation.

Digital twins are a particular type of high-fidelity model that can assist with model-aided T&E. They take real-world data about a system and update a digital model of the system to predict system behaviour. This can help with verification of similar test cases or verification of variants of the system—one real-world test result can be extrapolated with greater confidence than other test results using digital twins.

Understanding the uncertainty of the results from models used for this purpose is important. This uncertainty originates from the fact that a model is a representation of a system and may have known levels of accuracy (fidelity), sensitivity to certain test cases, or limits of the model itself. Test cases that have greater uncertainty in the result might be a priority for assessing that test case with the live system.

Model-aided T&E comes with a potentially substantial cost: models suitable for T&E are necessarily relatively high-fidelity models that must be built and have an associated sustainment tail. These models can be bespoke, or modular to form part of a wider T&E program, and a return on investment consideration must be put into whether conducting model-aided T&E is best for the acquisition program.

Developing such models involves a high level of technical expertise. Where industry supports Defence in building such models, probity and conflicts of interest must be considered. Industry developers of T&E models should not be tendering the product to be tested.

A final challenge of model-aided T&E is for simulations to keep pace with technology, especially when aiming to predict future performance. For example, simulation software can predict performance of older communication waveforms very well, but more modern waveforms may not be supported, or data may be proprietary and therefore not available in the simulation tool.

3.5 Iterative and Agile Procurement

Model-based approaches support iterative and agile procurement. Models support a team to add use cases and requirements, update the architecture and design in the model, and digitally verify that the requirements are met by the updated design, all within short sprint cycles. For example, for highly complex products with high manufacturing costs, this approach in the design phase can lead to a design better aligned to user needs before manufacturing commences. For less complex products, each sprint might provide an updated design for the next prototype.

Delivery paths using more iterative or agile partnerships greatly benefit from having a model spanning the boundary between Defence and industry. Defence can update its concept of operations, requirements, and architecture in the model using short sprint cycles and have the industry partner update its design in accordance with the updated model using these agile systems engineering approaches.

3.6 CDD and ASDEFCON Support for Models in Acquisition

Models to support acquisition are not new to Defence. There have been many attempts to increase the use of models over the years, such as using enterprise architecture views to the definition of Operational Concept Documents (OCDs), and using Defence enterprise level tools. There has

been mixed success, ranging from model-added approaches that have become cumbersome to dynamic models helping to facilitate the boundary between Defence and prime contractors.

The guiding manuals, procedures, and templates used in Defence acquisition support the use of models to a limited extent:

- CASG-2-MANUAL (E&T) 12-3-003 Capability Definition Documentation (CDD) Guide (v 3.1) identifies MBSE as a useful tool to capture relevant information in lieu of documentation.
- Operational Concept Document (OCD) Procedure (v 3.0) mentions that models may replace the OCD in the future as the source of information, with the OCD becoming a point-in-time export.
- ASDEFCON templates have provisions for Defence to receive models (and other data) from contractors. For example, in the draft statement of work template for strategic materiel, the optional clause 4.5.5 describes that Mission System Architecture models and models forming part of formal verification and validation activities will be provided by the contractor.

It is expected that future updates to both CDD guidance and ASDEFCON templates will expand the use of models to support these activities.

4 Models and Their Purpose

4.1 What Is a Model?

Defence's adoption of digital engineering under the Digital Engineering Strategy 2024 is underpinned by the use of models: Goal 2 of the strategy encourages the use of digital models to inform decision making in Defence. Models play a critical role in enabling "systems thinking" and decision making across the life cycle.

A model is defined in the Australian Defence Glossary as "a physical, mathematical or logical representation of a system, entity, phenomenon or a process;" put more simply, a model is an approximation of the real world to help us understand it.

We use models of different forms in everyday life: whether it is a map of a city, a toy plane, the floorplan of our house, or weather prediction models. Defence models represent systems, platforms, and systems of systems such as orbital paths of satellites, 3-D geometric models of aircraft, or models that represent other aspects of real world systems. Using model based technology, the types of models we can build and use to assist in answering questions have become more complex and insightful. Defence uses, and increasingly will use, a variety of these advanced models across the One Defence Capability System (ODCS). These models will therefore represent Defence capability at different levels – from Defence as a Joint Force based on joint missions and groups of collaborating systems (system of systems), through to individual capability systems and their subsystems. Throughout this document the utility of models will be framed against the ODCS.

4.2 Types of Models

In digital engineering, models serve as the foundational artefacts for understanding, designing, analysing, and verifying complex systems. Models, rather than documents, provide these functions to support analyses that are more complete (and often conducted more quickly) than traditional approaches. These systems can range from individual units of military equipment (capability systems) to the systems of systems (SoS) in which this equipment is used. Models also may focus on a subsystem or even a small part of the behaviour/parameters of the physical system. For complex systems, a wide variety of models can be used, each tailored to represent specific aspects of the system. These models can differ in their levels of abstraction, formality, purpose, and domain specificity. A classification of models can be useful in selecting the most appropriate modelling approach for a given task.

4.2.1 Formal vs Informal Models

Formal models are built using well-defined modelling languages that have strict syntax and semantics. These models are precise, less ambiguous, and often machine readable, enabling automated analysis, validation, and integration. Examples of formal models include Systems Modeling Language (SysML) models, Computer-Aided Design (CAD) models, and mathematical specifications.

In contrast, informal models are less structured and may include sketches, textual descriptions, or ad-hoc diagrams. While useful for ideation or stakeholder communication, informal models lack the rigour needed for detailed engineering analysis and are not typically used as authoritative sources in digital engineering. Valuable information exists in informal models and can be very useful if it can be extracted and converted into a formal format.

This categorisation is not necessarily a binary choice but a continuum, where some models fall in-between the two. Furthermore, formal models may contain informal models within them in a form of textual descriptions or mock-up sketches.

Defence is aiming to increase the use of formal models to improve the design rigour of the integrated Joint Force, the definition of constituent capability systems, and the ability to share

precise engineering information between Defence stakeholders and traceability to the delivered material system.

4.2.2 Physical vs Abstract Models

Physical models are tangible or concrete representations of a system, such as prototypes, mock-ups, 3D component models, or physical scale models. These are often used in design validation, manufacturing, or testing.

Abstract models, on the other hand, represent systems conceptually or mathematically. They are not physical artefacts, but logical or mathematical constructs typically captured in software-based tools that describe system behaviour, structure, or performance. Abstract models are essential in early design phases and for simulating system behaviour before physical implementation. In the Defence context, abstract models are vital for designing and validating Joint Force through the representing missions and corresponding groups of collaborating SoS.

As Defence DE capabilities mature, there is increasing potential to combine physical and abstract models – sometimes referred to as synthetic modelling – to enable simulation, training, and analysis of SoS and capability systems in a representative (virtual) environment. DE permits the ability to perform this analysis on “to be” and “proposed” systems in a mission context.

4.2.3 Descriptive vs Analytical Models

Descriptive models focus on capturing the logical structure and relationships within a system. They illustrate how components are organised, how they interact, and what functions they perform. These models are essential for defining system architecture and behaviour in a way that is understandable to both engineers and stakeholders. This is true of systems representing a product, or of higher-order systems or system-of-systems such as the Joint Force or a coalition of forces. Common elements of descriptive models include system hierarchies, functional flows, interface definitions, and architectural diagrams. They are limited by what is currently known about the real-world system and should only capture what is relevant for their intended purpose.

Analytical models are used to perform quantitative analysis of system parameters. These models are typically mathematical in nature and support computations related to mission success, system performance, reliability, cost, and other measurable attributes.

Analytical models can be further divided into:

- Static models: which represent fixed computations like mass or power budgets, and
- Dynamic models: which simulate time-varying behaviours such as likelihood of detection, quality of connection, energy consumption, motion, or thermal dynamics. “Parametric” models are forms of analytical models.

These models are crucial for validating that a system meets its performance requirements and for supporting trade studies and optimisation.

Model-Based Systems Engineering (MBSE) languages such as SysML are frequently used to create both descriptive and analytic models, providing a standardised way to represent system structure and behaviour, and provide foundational analytic features for model execution e.g. parametric, state machines, and can provide linkages to more specialised tools.

In practice, many models are hybrid, combining both descriptive and analytical elements. For instance, a system architecture model might include structural diagrams (descriptive) alongside embedded equations for performance analysis (analytical). These hybrid models offer a more comprehensive view of the system, enabling both qualitative reasoning and quantitative evaluation. They are particularly valuable in integrated environments where multiple aspects of the system must be analysed and validated concurrently.

The use of descriptive and analytical models by Defence is driven by the level of system under consideration (Joint Force SoS, domain SoS, or individual capability system) and the engineering efforts undertaken.

For example, VCDF Group's role to design the Joint Force will increasingly rely on descriptive and analytical models. These models will capture the operational structure and behaviour of the Joint Force and permit analyses. These will serve as the basis for Joint Force design decisions and capability definition. While portions of analytical effort can be supported by industry, the models of the Joint Force will be held by Defence.

For individual capability systems being procured by Defence, the focus of Defence effort will often articulate (and in the future models) the system context and requirements. These must be sufficiently refined and elaborated for industry to design and/or deliver a solution which meets Defence needs. In many cases, industry will internally develop and use models independently to design, deliver and sustain the capability over its lifecycle. Defence envisages a future in which Defence can share Acquisition models and industry can respond with system models appropriate to the nature of the procurement.

4.3 Model-Based Design

Models, in some form, have always supported engineering and design, whether purely models of a design, hand-drawn schematics, or advanced digital twins. Modern engineering is almost always based on digital models of the artefact or system being designed.

Many different engineering disciplines have adopted digital models to support their work in different ways and at different scales. As they have transitioned over time, the ability to use the structured data captured in digital models for novel analyses and computationally advanced analyses has been transformative to those disciplines. This evolution of models from paper (analog) to digital to analytical is shown in Figure 14.

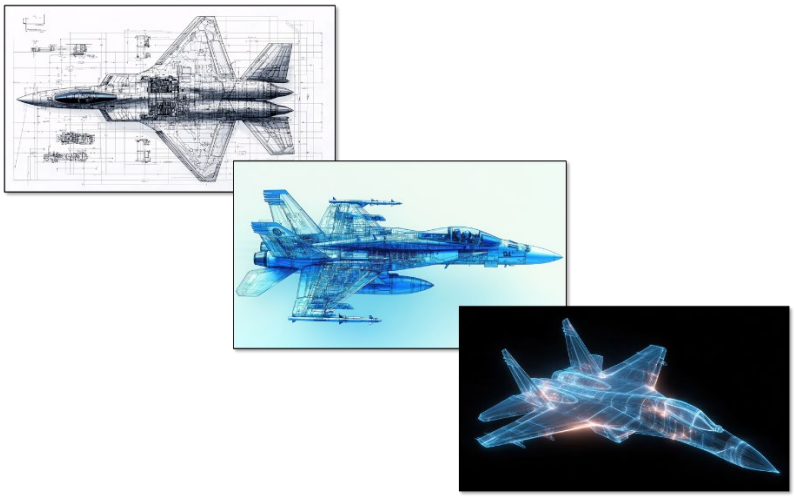

Figure 14: Analog to Digital to Analytic Models

Different engineering disciplines or application areas may implement their own formal models tailored to their discipline. These include models for electrical, mechanical, software, thermal, structural, and power systems, among others. These models can be further categorised by application domain, such as aerospace, information systems, or medical devices.

In Defence, multiple models are often linked to the same system or system of interest, with their content varying based on the model's purpose. Digital Engineering Strategy 2024 emphasises connecting or federating these models with authoritative data sources to support life cycle activities. Federated modelling addresses the reality that a system model may draw on contributions from several related and specialist models. It also reflects the fact that it is often impractical to build a single comprehensive model which spans operational through to detailed physics-oriented model content.

The benefits of a federated approach in Defence can be viewed at different levels with respect to the ODCS. At the Joint Force level, federated modelling mirrors the distributed nature of engineering effort in Defence in which the Joint Force design is realised through subordinate

domain and system-level engineering activities by Groups and Services. To support this top-down approach, VCDF Group is establishing a Joint Force Model (JFM) as the authoritative reference for force design and capability development. It will be approved by VCDF as the Joint Force Authority (JFA). The federated approach to Defence models will enable engineering content to be more readily shared, re-used and managed across the Strategic Centre and Domain Leads, Capability Managers, Delivery Groups and DSTG as shown in Figure 15. Industry may also provide updates to portions that reflect delivered systems. Industry would normally own the internal design models used to develop their systems and share portions or abstractions with Defence to support defence modelling of that platform for decision making, Verification and Validation (V&V), and sustainment activities.

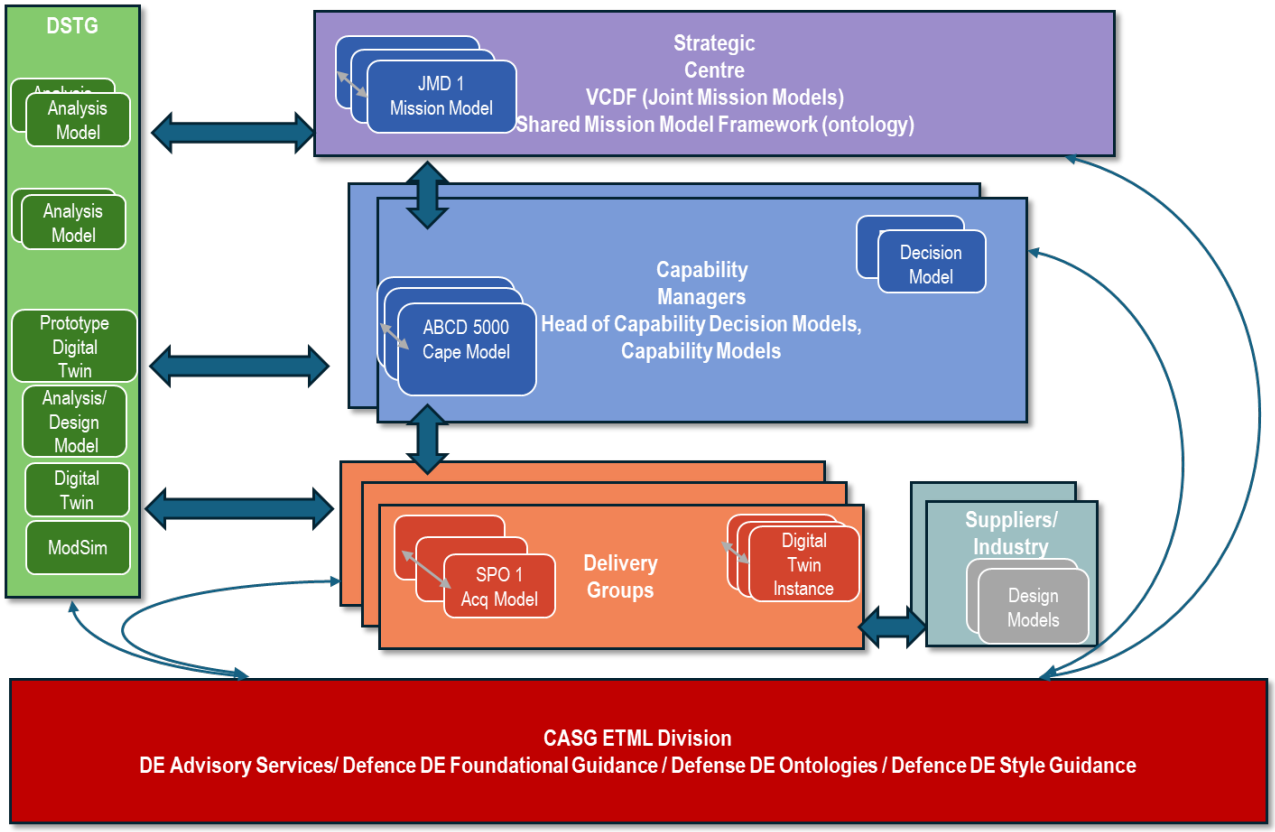

Figure 15: Models across Defence

The community of interest for a given digital engineering model will determine the specific types of tools, data, and methodologies it needs to create, offer, request, and exchange models, data, and analytic results for its domain. This community of interest may be aligned to a data domain, application domain, specific program(s), or community that requires models to be sharable and consistent. This federated approach to combining models requires careful consideration, and recognition that not all models are appropriate for sharing.

4.4 Simulation vs Modelling

The concepts of models and simulation are closely linked. While not all kinds of models are able to be simulated, having structured engineering data about a system captured digitally in a formal model has enabled engineering disciplines to conduct deeper analysis using simulation. While models serve as structured representations of the real world, simulations are how these models provide insights into how a system operates under specific conditions over time, without subjecting the real-world system to those conditions/time.

A simulation, as defined in the Australian Defence Glossary (ADG) and Modelling and Simulation (M&S) Policy, is a method for implementing a model over time. It is essentially the execution of a model, typically an analytical one, within a computational environment. It allows engineers to observe how a system behaves under various conditions, using defined inputs and contextual

scenarios. Simulations can be configured in many ways, such as deterministic or stochastic, continuous or discrete,[2] and are often used to analyse dynamic behaviour, test system responses, or support decision making. In essence, while models define the system, simulations enable us to explore its behaviour in action.

Defence modelling in support of the ODCS is driven by the level of 'system' being considered - from Joint Force down to individual capability systems - and the nature of engineering work and analysis to be undertaken. Similarly, simulation will be applied appropriate to the level of 'system', the domain under consideration (e.g. air-based refuelling capability) and the questions to be answered or the insights being sought (refuelling capacity required in a mission).

Modelling and simulation activities are both vital to Joint Force design and capability development in support of the ODCS. Simulation is essential for making Joint Force design decisions. These decisions are framed against mission performance and enterprise impacts. This includes applying trade-off analyses to define capability gaps and selecting optimal solutions given a set of capabilities. Modelling and simulation also support conducting 'what ifs' and impact assessments when, for example, the environment or threat changes (reactive) or Defence plans to withdraw systems from service (proactive).

VCDF Group's Joint Force Model (JFM) will be constructed so that aspects can be exported to simulation tools to aid analysis and design decisions which are subsequently reflected in the JFM. This approach allows for simulation capability to be drawn from a range of specialist areas within Defence, for example Defence Science and Technology Group (DSTG), and outside Defence across industry or academia noting the importance of security considerations.

Defence application of simulation tools for defining capability systems will be determined on a case-by-case basis and must reflect the nature of engineering work performed by Defence at this level. This will differ to simulation performed by vendors who will conduct more detailed design and development as they develop or modify the system in accordance with Defence needs.

The Modelling and Simulation (M&S) Policy focusses on the subset of models that are used to support simulation. Its four principles stated below, however, can apply equally to any model supporting digital engineering:

1. **Interoperability:** New or upgraded M&S systems, including those that are required to support the integrated force, should consider interoperability with the M&S systems of the broader Defence community, Commonwealth, state departments, and agencies; allies and partners; industry; and academia.
2. **Reuse:** Existing M&S assets should be considered for reuse; however, this should not drive disproportionate redevelopment of existing assets where they meet the Minimum Viable Capability requirements. M&S assets may include, but not be limited to, data, models, simulations, simulation systems, and licences. Acquisition or development of new M&S assets should consider future reusability of those assets by Defence.
3. **Asset Management:** M&S assets should be actively managed to support the principles of interoperability and reuse.
4. **Standards Convergence:** Adoption of preferred Defence M&S standards should be considered during integrated M&S planning. Beyond standards, convergence of approaches to modelling and of abstract model structure improves interoperability and reuse across Defence.

[2] Some of these terms are present in the Australian Defence Glossary (ADG) (e.g., deterministic simulation and discrete event [or time] simulation), but this refers to a more general academic definition. Stochastic in a simulation leverages probability or other non-deterministic traits, such that outputs are not always the same given a particular input. Discrete can also imply steps in discrete time (turns) or events (actions) or simply quantisation. Discrete can also refer to inputs or outputs (e.g., on/off states). The system of interest and its context is important to consider determining the specific discrete/continuous or deterministic/stochastic simulation. For example, it is more realistic to model physics/physical-based simulations using continuous inputs/outputs (e.g., velocity/acceleration/strain), but computer-based systems are always discrete and deterministic by nature – they always approximate continuous values (e.g., Pi or stochastic systems). This approximation can be extremely important in understanding the fidelity of simulation possible and its impact and limits on precision and representation.

4.5 Models, Simulation, and Artificial Intelligence

Artificial Intelligence (AI) is transforming digital engineering by enhancing modelling, simulation (M&S), and decision-making processes. Artificial intelligence (AI) tools are increasingly prevalent and useful, and the structured data produced by model-based design lends itself well to AI applications. Model-based design tools increasingly offer AI features to help accelerate design effort and to analyse models to provide insights. AI-enabled systems can utilise advanced M&S outputs as input data for training, while generative AI can inject simulated inputs or adversarial responses into digital models. AI can also perform M&S directly, using real-world data to predict outcomes in hypothetical scenarios. However, these applications come with limitations, uncertainties, and vulnerabilities that must be carefully managed, particularly in Defence and for critical systems used in decision making.

Advanced M&S and generative AI systems face challenges in generating previously unknown/unseen data as well as tendencies to “hallucinate” or invent data that can / does not exist. However, new vulnerabilities exploited by adversaries may be less known. For example, MITRE ATLAS™ serves as a publicly available resource, documenting known adversarial tactics and techniques targeting AI and AI-enabled systems, including real-world exploitations and some mitigations. Defence must ensure robust testing, validation, and monitoring to ensure the reliability and security of AI systems, including those used in digital engineering. In the short term, this is normally done with expert human oversight.

The adoption of AI in Defence and critical systems has also raised significant concerns among Australian parliamentary and Defence stakeholders. While AI offers substantial benefits—such as enhanced operational efficiency, predictive analytics, and decision-support tools—it also introduces risks, including adversarial exploitation, ethical challenges, and over-reliance on autonomous systems. Parliamentary discussions have emphasised the need for governance frameworks that prioritise security, accountability, and alignment with national interests. Defence stakeholders are particularly concerned about the unpredictability of AI behaviour in dynamic environments, the ethical implications of autonomous decision making, and the potential for bias in AI algorithms. These issues underscore the importance of maintaining human oversight and ensuring AI systems operate transparently and responsibly.

Looking ahead, advancements in Explainable AI (XAI), AI-driven optimisation of digital twins, and the integration of AI with emerging technologies such as quantum computing and the Internet of Things (IoT) are expected to further revolutionise digital engineering. However, these developments must be accompanied by robust ethical frameworks and human oversight to ensure their responsible and sustainable use.

While a comprehensive discussion of digital engineering AI use and misuse cases is beyond the scope of this document, the role of AI in digital engineering—spanning model development, interpretation, updates, and M&S—is rapidly expanding. The impact of AI cannot be underestimated, and its evolution in commercial and potential application to Defence digital engineering efforts continues to accelerate almost daily.

4.6 Models and the Digital Engineering Functional Areas

The Digital Engineering Framework describes eight “Digital Engineering Functional Areas” (Figure 16) i.e., the groupings of functions that are typically supported by digital engineering. Many of these functions are achieved by models.

Figure 16: Digital Engineering Functional Areas

4.6.1 Requirements Management

Needs and requirements are often captured and managed as textual statements, and not in formal models. Models become very useful for requirements management when requirements are traced into models, typically system architecture models. In requirements management, needs are the high-level problems or goals that a user or stakeholder wants to achieve, while requirements are the specific, detailed conditions or capabilities that a product or system must meet to satisfy those needs. Needs define why something is needed or what problem is being addressed, focusing on the desired end result, whereas requirements define what the product must do to deliver that end result and are often written from a technical or solution perspective

Each need or requirement statement can be linked to system elements, behaviours, and verification methods. For design, this linkage ensures traceability from stakeholder needs through to design and test. Where models have been used to support the elicitation of requirements by Defence (in developing the Function and Performance Specification (FPS)), they can point to the model-based analysis, which justifies the requirement as written. Models help clarify intent, reduce ambiguity, and support impact analysis when changes occur. They also enable automated validation and verification processes. By embedding requirements within models, teams can monitor and maintain the completeness, consistency, and compliance throughout development.

Most system architecture and design tools support representing requirement statements as model elements. In supporting quality engineering and design of systems, they should be included and traced in these models as standard practice, even if formally managed in other tools.

4.6.2 System Architecture and Design

Models in this area provide structured representations of system structure, behaviour, performance, and interfaces. Descriptive and abstract models, such as system context diagrams, functional block diagrams, and architecture views, are used to define how components interact and how tasks, data flows, or control mechanisms are assigned across the system. These models support early design decisions, enable traceability to requirements, and ensure consistency across

engineering teams. They also help identify integration points, manage complexity, and maintain architectural integrity throughout development.

Examples of system architecture models generally include MBSE models and enterprise architecture models.

4.6.3 Operational Analysis

In operational analysis, abstract analytical models are used to understand how a system will operate within its intended environment. These models typically take the more descriptive models from system architecture models and apply them to operational scenarios, to assess performance, user interactions, and mission-level requirements. AI is used for Operational Analysis to model an entities behaviour. By simulating workflows and analysing use cases, engineers can assess system performance within operational contexts, identify constraints and inefficiencies, and validate that the design supports mission-critical functions and real-world operational demands.

4.6.4 Physics-Based and Mathematical Analysis

Analytical models are used to simulate and assess system performance, reliability, and physical behaviour across a range of conditions. These are often more physical than abstract models, but advanced physics and mathematics can be applied to abstract models as well. These include mathematical models, control system simulations, and finite element analysis, which help engineers understand how systems respond to real-world forces and constraints. Many Modelling and Simulation (M&S) assets fall into this functional area.

Digital twins extend this capability by integrating live operational data with physics-based models, enabling predictive analysis, fault detection, and performance optimisation. Together, these models reduce reliance on physical prototypes, support early identification of failure modes, and enable continuous refinement throughout the system life cycle. They are essential for ensuring that systems meet performance expectations and operate safely and efficiently.

4.6.5 Physical Design and Specialty Engineering

Physical models represent the tangible aspects of the system, including geometry, materials, and assembly. These include CAD models, 3D prototypes, and detailed engineering drawings that support the transition from conceptual design to manufacturing. They are also used to integrate specialty engineering disciplines, such as structural, thermal, and electrical analysis. Physical models ensure that systems are buildable, testable, and maintainable, and they support digital manufacturing, verification, and sustainment planning.

4.6.6 Life Cycle Management

Models play a key role in managing systems across their entire life cycles—from concept through to disposal. Descriptive, physical, and abstract models are used to track configuration, assess change impacts, and support decisions related to upgrades, maintenance, and obsolescence. These models help maintain alignment with stakeholder needs and regulatory requirements over time. They also support sustainment planning, cost estimation, and risk management, ensuring that systems remain viable and supportable throughout their operational lifetimes.

Life cycle management models include Asset/Application Lifecycle Management (ALM) and Product Lifecycle Management (PLM). These models focus on managing software and hardware products from design through to production, with an emphasis on overseeing the suite of products a vendor offers throughout their life cycles, including disposal. In practice, there is little distinction between the two, although historically the difference lay in the phases of the lifecycle or the focus on physical versus software systems. Defence's Enterprise Resource Planning (ERP) solution spans life cycle management, engineering management, and project management functions.

4.6.7 Engineering Management

Engineering management relies on models to plan, monitor, and coordinate technical activities. These include cost models, schedule models, risk assessments, and business process diagrams. Such models provide visibility into resource allocation, progress tracking, and delivery risks. They support informed decision making and help align engineering efforts with broader project and organisational goals. By using data-driven models, managers can optimise workflows and improve predictability across the engineering life cycle, while operating within their defined scope and the organisational constraints.

4.6.8 Visualisation

Visualisation tools transform model data into accessible, interactive formats that support understanding and collaboration. These include dashboards, 3D renderings, system animations, and interactive diagrams, each enabling users to explore complex systems dynamically. For example, interactive diagrams allow users to zoom, filter, and manipulate components to examine relationships and dependencies in real time. Visualisation helps communicate complex information to diverse stakeholders, including those without technical backgrounds. It supports design reviews, decision making, and stakeholder engagement. By presenting model data in a more intuitive manner, visualisation enhances transparency and accelerates problem solving.

Most modelling tools offer built-in visualisation capabilities, often tailored to the needs of various stakeholders. Visualisations for engineers can be very technical and not appropriate for all audiences. Specific data visualisation tools, such as PowerBI and Tableau, can take structured data from other tools and visualise flexibly for different stakeholders.

4.7 What Are the Benefits of Using Models?

Models integrated within a digital framework not only enhance transparency, traceability, and technical consistency but also act as powerful enablers of collaboration, informed decision making, and life cycle integration. The adoption of model-based approaches in digital engineering has revolutionised how complex systems are conceived, developed, and managed, driving greater efficiency and innovation.

Some of the most critical aspects to consider when creating models are their purpose and scope. Models must be “fit for purpose,” meaning they should address specific objectives and deliver meaningful value to the organisation that created them. Additionally, careful consideration should be given to the problem the model is solving, as well as how it will be used, maintained, and shared. These decisions are essential for scoping models to the appropriate level of content and fidelity, ensuring they remain relevant and effective throughout their life cycle.

The following sections outline some of the key benefits that models bring to digital engineering environments. By thoughtfully addressing these considerations, organisations can maximise the utility of their models while fostering collaboration, integration, and informed decision making across the engineering life cycle.

1. **Structured Data to Support Decision Making**: Models provide a structured, machine-readable representation of systems, which can support informed and timely decision making. They assist engineers and stakeholders capture system information in a logical, relational, and often mathematical format—forming a basis to perform impact assessments, conduct trade-off analyses, perform system-of-systems analysis, and automate verification and validation processes. By offering a consistent and traceable foundation for analysis, models reduce uncertainty and enhance the quality of decisions across the system life cycle. The advanced complexity of modern systems requires stronger objective evidence to inform decision making. Structured data supporting advanced analysis is required to provide this kind of information to decision makers in a timely fashion.
2. **Improved Communication and Reduced Ambiguity**: By replacing ambiguous textual descriptions with precise visual and formal representations, models significantly improve communication among multidisciplinary teams. They help ensure that all stakeholders,

including engineers, managers, and user representatives, share a common understanding of system requirements, architecture, and behaviour. Beyond these core elements, models such as physical, analytical, and mathematical models provide precise, domain-specific views that enhance clarity and support effective communication across disciplines. This clarity reduces the risk of misinterpretation and fosters more effective collaboration throughout the development process.

3. **Common Data Environment**: Well-formed, verified and validated models can serve as authoritative sources of truth by providing a robust, traceable, and stakeholder-aligned representation of system data across the broader Defence enterprise. Integrating information from various disciplines and tools establishes a common data environment that enhances collaboration, reduces duplication of effort, and ensures consistency/coherence across all system representations/artefacts. It also supports interoperability, enabling seamless data exchange and coordination among diverse engineering tools and teams.
4. **Advanced Analytics and Visualisation**: The integration of models with advanced analytics and visualisation tools enables timely insights into system and project performance and behaviour. Models can be connected to dashboards, simulations, and AI-driven analytics, allowing teams to explore scenarios, predict outcomes, and make data-driven decisions. These capabilities enhance situational awareness and support proactive system optimisation. These kinds of analytics may include using engineering design data across a digital thread to assess the impact of proposed design changes on the impact to achieving missions or using operational data with a digital twin to predict and assess different likely outcomes to inform command decisions. The sensitive nature of Defence operational data can make the connection of operational data to design data difficult, but there are opportunities for real-world operational data to inform Joint Mission Design and lead to better identification of gaps and opportunities in the Joint Force.
5. **Life Cycle Integration**: Models are foundational to establishing a digital thread that connects data and artefacts across the entire system life cycle, from concept and design to testing, deployment, and maintenance. This digital continuity supports agile and iterative development, facilitates compliance and certification, and ensures that systems can evolve efficiently in response to changing requirements or operational conditions. In particular, within Defence, this digital thread will support activities across the ODCS.

5 Getting Started Modelling

Starting a new modelling effort can be difficult, particularly for tasks that have not historically used models within Defence. This topic describes some key considerations that should be taken into account when getting started. Specifics for particular initiatives will heavily depend on the nature of the problem, and any approach will need to be tailored. This topic does not attempt to cover detailed advice for specific kinds of models, but does provide advice that is universally applicable.

5.1 Focus on the Problem

All models are wrong, but some are useful – George Box, statistician, 1987

When starting a new modelling effort, it is important to keep the problem you are trying to solve at the front of your mind. Any model approximates the real world in some way; for it to be useful, it needs to help solve a problem. Often the solution or objective to the problem is to inform a decision. Focussing on the problem helps in several ways.

First, it helps decide what kind of models would be most useful to assist in solving the problem. While a functional, abstract Model-Based Systems Engineering (MBSE) model of a ship might help develop a Function and Performance Specification (FPS), it will not help you determine the routing of cabling of a communications system to be installed across the vessel. A Computer-Aided Design (CAD) model would be a better choice for this problem.

Focussing on the problem also helps to define the scope the modelling effort. A common trap for modellers is to include unnecessary detail that does not contribute to the problem needing solved. Unnecessary detail adds to the maintenance load of keeping models up to date (if they are to be sustained long term) and can impact the performance of modelling and simulation tools.

5.2 Methodology, Data, and Tools

There is more to a digital modelling effort than simply selecting a modelling tool and entering data into it. Once the problem (and objective) is understood, methodology, data, and tools need to be considered in concert with one another, as shown in Figure 17. These are the three critical aspects that should be considered and agreed upon by the team working with the model.

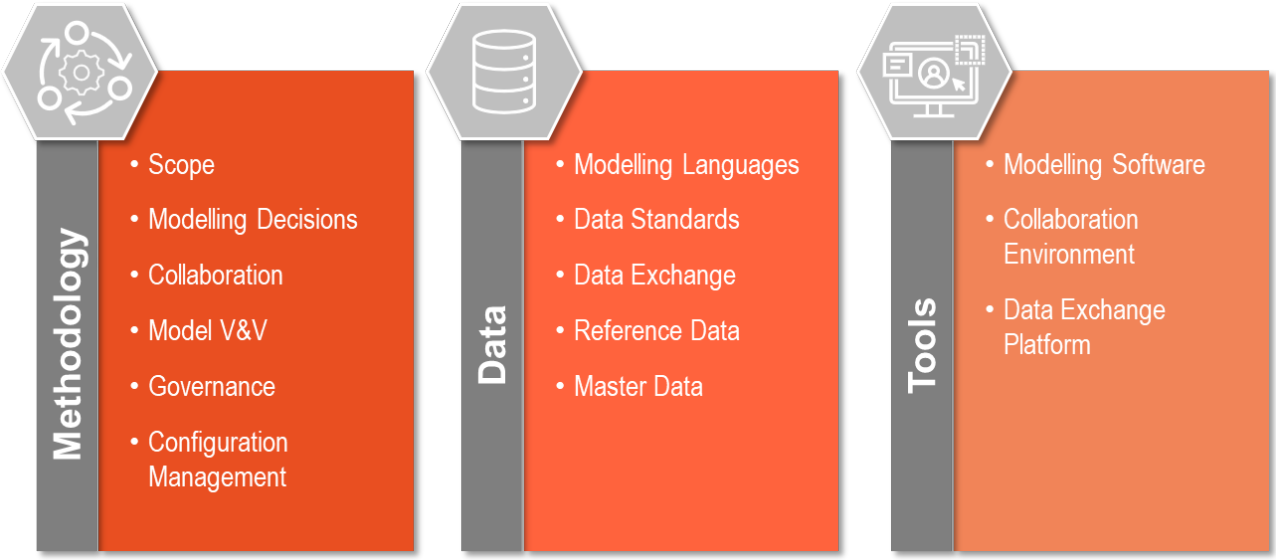

Figure 17: Key aspects of Digital Modelling

Tools are often the first focus when trying to get started, which is often the wrong approach. Using the wrong tool for the job can lead to wasted effort and unnecessary tool customisation. Focussing on the problem to solve should be the focus of selecting an appropriate tool. Consideration of the other two aspects can influence tool choice, so tools should not be considered in isolation.

Additionally, often a single tool may not be adequate to solve the problem, but an environment of tools and users may be required.

Some vendors have separate tools/ecosystems to facilitate collaborative modelling, and if a range of tools and models are required, separate data exchanges between tools may be needed. Interoperability considerations can drive the decision to choose one tool over another. Additional guidance on selecting appropriate tools is available in the Tools Assessment of the DE foundational guidance, with the first emphasis being on MBSE tool selection. In this specific use case, it still assesses at the overall ecosystem in which the tool resides.

Related to tools are the **data**, data standards, and modelling languages that will be used. The choice will similarly be driven by the problem to be solved, but considerations of interoperability will also inform this decision. It is often difficult for a single system owner to understand the value of data standards; in some cases, this has a perception of adding burden. If the model is developed only for single use and does not require data from other sources or sharing data outside itself, then the internals probably do not matter. When working in a larger eco-system the interchange of data with other models and tools—within the project, across Defence, with industry, and with international partners—should be considered. This relies on using industry wide, commercial standards for data interchanges.

Models leverage data, both in their creation and in their use. Existing sources of data within Defence, including Reference and Master data, should be identified for reuse, to minimise duplication. Additionally, often models transform or provide new data that can be shared across Defence. Some of these topics will be addressed in a separate Digital Engineering Data Guide. Choices made regarding data may inform the choice of appropriate tools.

The third aspect to consider is the **methodology** to be applied. Tools and data define what you will be working with; the methodology is how you are going to use it. This will include decisions related to the model, like how you will work within your team, what level of detail (scope) is considered sufficient for the model, how the model will be verified and validated, and any governance or configuration management that will be applied.

Some modelling languages and tools are highly flexible, and part of the methodology will include decisions the team has made on how a language or tool will be used in the context of the problem. This is particularly true of abstract models. There are usually many different valid ways of abstracting the real world and representing it in the model. Decisions made need to be clearly communicated and consistently applied across the team.

The modelling guidance in this guidebook largely describes options and considerations for developing the methodology. The methodology will likely be refined and evolve over time, and it should be documented. The documentation should be able to be readily updated. Where possible, document the methodology within the model itself.

5.3 Crawl, Walk, Run

Starting small and building upon success is the right approach to modelling. Often, lower fidelity models help to understand how to build and the considerations for higher fidelity or more complex models. This is more than taking large tasks and breaking them into smaller ones. It is about incremental process that delivers value which can be validated at each increment. It is better to commence with part of a solution and risk some rework to show progress and build skills within the team. Use the Crawl-Walk-Run approach:

Crawl: Establish the basic capabilities and data collection associated with the problem being solved or objective. As shown in Figure 18, this is typically about transitioning to data-driven approaches, establishing digital methodologies, and determining the appropriate models, data structures, and tools to support the effort. Define the problem and identify key metrics. This phase is as much about familiarising the workforce with models as it is about outcomes.

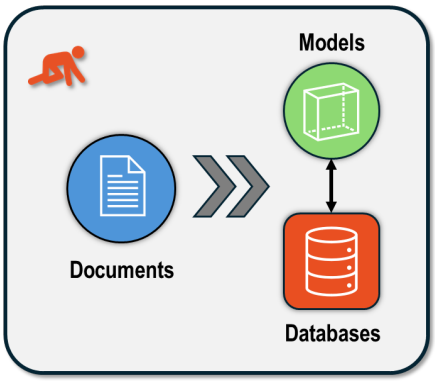

Figure 18: Crawl (Documents to Data)

Walk: The use of models becomes routine. Appropriate digital engineering tools are successfully used to solve problems and share information. At this stage (Figure 19), models and digital approaches are familiar and commonly used. Data across different tools starts to be integrated, and more advanced analytics become possible.

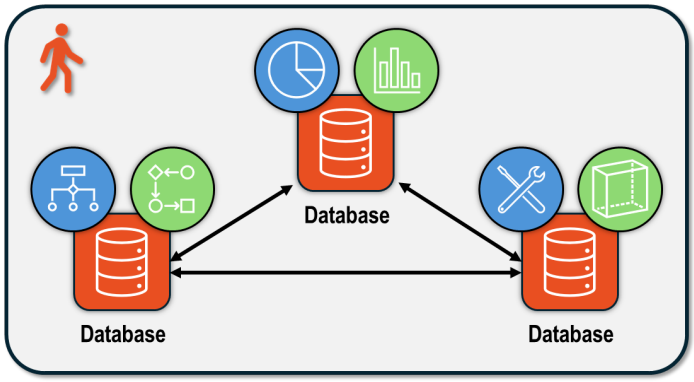

Figure 19: Walk (Digital by Default)

Run: This involves the integration of different models across the digital life cycle, as shown in Figure 20. Data can be interrogated and shared between models in different Groups and Services and between authoritative sources. This for Defence will leverage specific models to support the One Defence Capability System (ODCS) life cycle and using common data environments such as One Defence Data (1DD) for data sharing. As this progresses, it may include the ability to refine and develop capabilities collaboratively with industry in a digital environment/exchange.

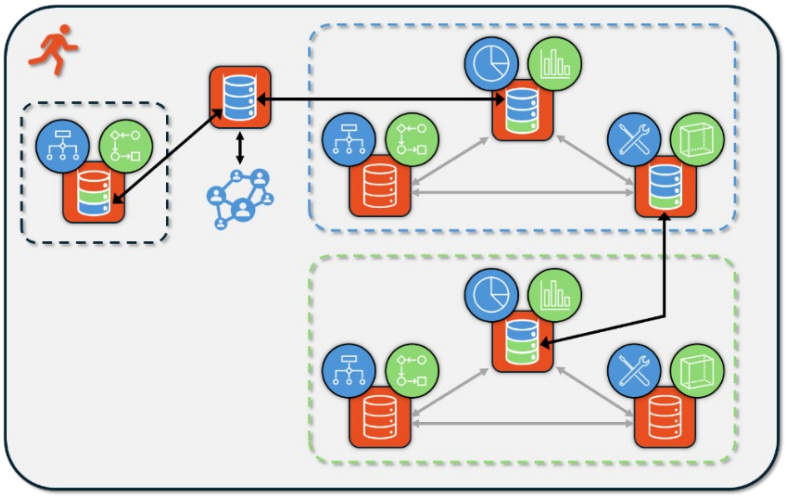

Figure 20: Run (Integrated Environment)

6 Modelling Standards

Across the breadth of models that can be employed to support digital engineering in Defence, there are a wide range of standards that might be applied such as modelling languages, architecture frameworks, and data exchange standards.

The Defence Digital Strategy 2024 outlines Defence's guiding principles for digital technology choices. These include using open architecture and open standards and enabling interoperability with Defence's military partners for a data-driven approach. Commonly used international open standards should be used for modelling wherever possible.

The fourth principle of the Modelling and Simulation (M&S) Policy concerns M&S standards convergence. It states that preferred Defence standards should be considered during the planning of modelling efforts.

6.1 Modelling Languages

Modelling languages have been a key development in interoperability and reuse of models across commercial industry. These standards allow multiple tools to operate and share information from a model. It allows the ability to communicate across tools, vendors, users, producers, and consumers.

6.1.1 Unified Modeling Language

The Unified Modeling Language (UML) is a standardised modelling language developed by the Object Management Group (OMG). It is primarily designed for software system specification and design. UML supports 14 different diagram types categorised into structural and behavioural views, including Class Diagrams, Use-Case Diagrams and Sequence Diagrams. UML is extensively used in software engineering but also provides a foundational base for system-level modelling. It is object-oriented and promotes model reuse, modularity, and abstraction principles that align well with modern engineering needs (Ref B43).

In the context of digital engineering, UML is valuable when systems have significant software components. It helps model software architecture, behaviours, interactions, and system-level scenarios. Although UML lacks native support for physical systems, requirements modelling, or performance analysis (which SysML addresses), it remains an essential skill for engineers involved in digital representation of software-intensive systems. Its influence is evident in how languages such as the Systems Modeling Language (SysML) and the Unified Architecture Framework (UAF) derive from UML's core architecture.

6.1.2 Systems Modeling Language

SysML is a widely adopted general-purpose modelling language tailored for systems engineering. Developed by the OMG, SysML extends a subset of the UML to support specification, analysis, design, verification, and validation of complex systems (Ref B44). It includes specialised diagrams such as Requirement Diagrams, Parametric Diagrams, and Block Definition Diagrams that are essential for modelling system-level requirements, physical structures, and engineering constraints. SysML supports multidisciplinary collaboration by integrating different aspects of system design and mechanical, electrical, software, and human systems, within a single modelling framework.

SysML is a key enabler of Digital Engineering and MBSE. It promotes traceability from stakeholder needs to system implementation and verification, helping teams manage complexity, reduce design errors, and accelerate development cycles. SysML also supports integration with simulation and analysis tools, making it ideal for digital twin and verification workflows. The language is currently evolving to SysML v2, which aims to improve usability, precision, and semantic rigour, particularly for digital engineering automation and tool interoperability. In the Defence context, SysML is the language that will be used in the Joint Force Model and its federated modelling framework.

6.1.3 Behavior Modelling Language

There are other types of modelling languages such as the Behaviour Modelling Language (BML) which is used to inform (or assess extant) requirement definition to determine fitness-for-purpose of the Capability Definition Documentation (e.g. FPS) and the subsequent system. These are currently in use across multiple Defence projects. BML bridges the gap between operational intent and the development of requirements and architectural views that underpin OCDs, FPSs, and other ASDEFCON artefacts.

6.2 Methodologies

6.2.1 Object-Oriented Systems Engineering Method

The Object-Oriented Systems Engineering Method (OOSEM) is a model-based systems engineering (MBSE) methodology developed by INCOSE that utilises SysML to integrate top-down systems engineering practices with object-oriented design principles. It defines a comprehensive process for the specification, analysis, design, and verification of systems, emphasising the use of models as managed artefacts throughout the system lifecycle. OOSEM provides the "how" for implementing systems engineering processes, complementing standards such as ISO/IEC/IEEE 15288:2023, and is characterised by activities such as requirements analysis, logical and architectural design, and system testing. Many MBSE tools leverage OOSEM to support their workflows.

6.2.2 Object-Process Methodology

Object-Process Methodology (OPM) is a holistic modelling approach that integrates structural and behavioural system aspects into a single coherent model. Developed by Prof. Dov Dori (Ref B46), OPM treats objects (things that exist) and processes (things that happen to or with objects) as primary modelling elements. Unlike UML or SysML, which separate models into multiple diagram types, OPM uses a single diagrammatic format known as Object-Process Diagrams, combined with a natural-language representation called Object-Process Language. This duality enhances accessibility for both technical and non-technical stakeholders.

OPM is recognised as an international standard (ISO 19450:2024) and is gaining traction in conceptual modelling and early system design phases. It excels at capturing the essence of complex socio-technical systems with minimal overhead, making it particularly suitable for high-level conceptual design, digital transformation planning, and academic research. Although OPM is not as widely adopted in industry as SysML, its simplicity, integrated semantics, and unified modelling paradigm are compelling advantages for systems requiring tight coherence between function and structure.

6.2.3 ARCADIA Methodology

The ARChitecture Analysis and Design Integrated Approach (ARCADIA) method is a model-based, tool-supported approach to systems and architecture engineering that explains how to understand real operational needs, define and justify architectures, and prepare Integration, Validation, Verification, Qualification (IVVQ) activities.

It organises work into five core activities which include Operational Analysis, System Need Analysis, Logical Architecture, Physical Architecture, and Building Strategy. These core activities build a coherent part of the engineering model used for analysis and decision-making.

Arcadia promotes a viewpoint-driven approach to reconcile functional and non-functional constraints, enabling early validation and architecture trade-offs that complement life-cycle standards such as ISO/IEC/IEEE 15288:2023.

6.3 Architecture Frameworks

Architecture frameworks have provided the basis of how models can represent or be used in context. They have been important in ensuring common understanding of “Views” into a model or how a model can be used. These standards allow humans and machines to use information from a model and agree on its meaning or even what information should be present. Many of these frameworks can be expressed in the modelling languages previously discussed and, as such, often use the same tools.

6.3.1 Department of Defense Architecture Framework

The Department of Defense Architecture Framework (DoDAF), developed by the U.S. Department of Defense, provides a comprehensive framework for describing and analysing the architecture of complex systems. It includes a suite of viewpoints, such as operational, capability, system, and project views that support decision making across the acquisition life cycle (Ref B47). DoDAF 2.0 emphasises the creation of “fit-for-purpose” views tailored to specific stakeholder needs, which is particularly valuable in digital engineering environments (Ref B48).

The framework is supported by the DoDAF Meta-Model, which enables integration with MBSE tools and supports data consistency across views. DoDAF has been widely adopted in Defence programs for its ability to capture complex interdependencies and support capability-based planning. However, its effectiveness depends on the quality of input data and the ability to tailor views to decision-making contexts.

6.3.2 NATO Architecture Framework

The NATO Architecture Framework (NAF) is designed to support multinational defence planning and interoperability among NATO member states. It provides a standardised approach to describing architecture that supports coalition operations, capability development, and mission planning. NAF aligns with other frameworks such as DoDAF and can be integrated with Unified Architecture Framework (UAF) for broader applicability (Ref B49).

NAF emphasises operational coherence and shared understanding across national boundaries. It supports the modelling of operational scenarios, capability requirements, and system interactions in a way that facilitates joint decision making.

6.3.3 Unified Architecture Framework

UAF, standardised by the OMG, was developed to unify and extend earlier frameworks such as the U.K. Ministry of Defence's Architecture Framework (MoDAF), DoDAF, and NAF. UAF provides a structured approach to enterprise and systems architecture using a metamodel that supports multiple domains, including operational, strategic, services, personnel, and resources. It incorporates SysML and UML profiles to enable rigorous modelling of system behaviour, structure, and performance (Ref B50).

UAF supports the development of integrated views that span capability, operational, and resource perspectives, enabling traceability across life cycle stages. It facilitates the modelling of a Concept of Operations (CONOPS), capability realisation, and performance standards in a way that aligns with digital engineering principles (Ref B51). UAF’s structure allows practitioners to represent operational needs independently of technical solutions, supporting early-phase analysis and stakeholder engagement.

6.3.4 ArchiMate

ArchiMate is an open and independent modelling language for Enterprise Architecture developed by The Open Group. It is designed to provide a clear and structured way to visualise the complex relationships between business processes, systems, and technologies. ArchiMate also offers a unified framework to map digital capabilities, model system-of-systems architectures, and align engineering workflows with enterprise strategy. It supports the integration of MBSE practices,

enabling traceability from high-level business goals down to technical implementations. With its layered and service-oriented approach, ArchiMate helps engineers and system architects capture the full life cycle of systems, from conceptual design to operational deployment, while maintaining traceability and coherence across domains.

6.3.5 The Open Group Architecture Framework

The Open Group Architecture Framework (TOGAF) is a widely adopted enterprise architecture framework developed by The Open Group. It provides a structured approach for aligning business goals with IT strategies, ensuring that all parts of an organisation work together effectively. This Architecture Development Method is a step-by-step process that guides architects through the design, implementation, and maintenance of enterprise architectures. TOGAF offers a disciplined way to manage architectural change, integrate systems, and ensure traceability from strategic intent to operational execution. Its modular structure supports both traditional and agile practices, making it adaptable to a wide range of organisational needs. TOGAF also complements modelling languages such as ArchiMate, enabling clear communication and consistency across architecture artefacts.

6.4 Physics-Based and Mathematical Analysis

A wide range of physics-based and mathematical analysis tools are used across Defence, industry, and broader digital engineering ecosystems to support modelling, simulation, and analytical workflows.

Commonly used environments include commercial numerical-analysis and simulation platforms such as MATLAB/Simulink, mission and engagement level simulation tools like AFSIM, SAFESIM, STK, and EADSIM, as well as engineering analysis suites such as ANSYS, Siemens Simcenter, Mathematica, and Maple. Standards-based modelling approaches such as MARTE (for real-time and embedded systems) and languages like Modelica (for multi-domain physical modelling) also contribute to this landscape.

Together, these tools enable tasks ranging from control design, multi-domain physical simulation, and finite element analysis to computational fluid dynamics, symbolic mathematics, operational research, and real-time system analysis. Even spreadsheet-based models may be used for lighter-weight analytical tasks. All of these constitute models in their own right, though they typically remain standalone unless explicitly parameterised, linked, or integrated into broader system-level modelling frameworks.

6.5 Simulation Interoperability Standards

The rapid advancement of computer-based technologies has facilitated the integration of previously isolated systems into large, interconnected, and complex networks of systems. As noted in the Defence Simulation Manual, these large-scale, interconnected systems can be utilised in computer-based distributed simulations that enable the integration of independently constructed simulations to create larger, more complex, and challenging simulation environments to support understanding, training, and analysis of modern warfare and military capabilities in Defence. The following sections present the two key interoperability standards, outlined in the Defence Simulation Manual, that support distributed simulation.

6.5.1 Distributed Interactive Simulation

Distributed Interactive Simulation (DIS) is an Institute of Electrical and Electronics Engineers (IEEE) standard designed to enable real-time, platform-level simulation across multiple systems operating in a shared synthetic environment (Ref B56). It allows geographically dispersed simulation entities, such as vehicles, sensors, and personnel, to interact and exchange state information over a communication network. DIS is particularly suited for defence, training, and mission rehearsal applications, where synchronised behaviour and interoperability between

systems are essential. The protocol uses standardised messages called Protocol Data Units (PDUs) to represent actions, movements, and interactions, ensuring consistent simulation behaviour across all participating nodes.

DIS is defined by IEEE Std 1278.1, which specifies application protocols and PDUs for simulation interoperability, while IEEE Std 1278.2 outlines the communication services and profiles for reliable data transmission. Together, these standards form the foundation for building scalable and interoperable simulation environments that can support complex, distributed operations across multiple domains.

6.5.2 High Level Architecture

High Level Architecture (HLA) is an IEEE standard, formally defined by the IEEE 1516 standard series, which provides a framework in which multiple simulation applications—known as **federates**—can interact within a shared environment called a **federation** (Ref B57). Each federate can represent a simulation, a data logger, a visualisation tool, or any other component, and communicates through a common interface known as the Run-Time Infrastructure. HLA supports time management, data distribution, and synchronisation, making it suitable for defence applications where modularity, scalability, and long-term system integration are critical. Its flexibility allows it to support both real-time and non-real-time simulations, making it a robust solution for distributed simulation challenges.

6.5.3 US Government Reference Architectures

Although not recognised as an international standard, the US Government has developed several reference architectures for key systems. These at a high level describe how systems interact and the standards that they use. This has included some reference architectures for Modelling and Simulation to include the overnment Reference Architecture (GRA) for Synthetic Training. Whilst these are not widely adopted standards, they may provide future value if we leverage similar system from the US as a partner and facilitate sharing models.

6.6 Other Relevant Modelling Standards

6.6.1 Industry Foundation Classes

Industry Foundation Classes (IFC) is an open standard developed by buildingSMART to enable interoperability in Building Information Modelling (BIM). It provides a standardised way to represent building and infrastructure data, including geometry, spatial relationships, materials, systems, and life cycle attributes, across different software platforms (Ref B60). IFC is a critical enabler of model-based collaboration across disciplines and project phases. It supports the integration of independently developed models into a unified digital environment, facilitating coordination, clash detection, and data-driven decision making. As the core data model underpinning the OpenBIM approach, IFC ensures that information remains accessible and usable throughout the asset life cycle, regardless of the tools or vendors involved.

6.6.2 Business Process Model and Notation

Business Process Model and Notation (BPMN), standardised by the OMG, is a graphical notation for modelling business processes. BPMN 2.0 introduces enhanced capabilities for modelling interactions, collaborations, and choreographies, making it suitable for representing workflows in both business and technical domains (Ref B59). It supports execution semantics, enabling integration with business rule engines and process automation platforms.

BPMN is widely used for aligning operational workflows with system behaviour, especially in enterprise and service-oriented architectures. Its visual clarity makes it accessible to both technical and non-technical stakeholders. However, BPMN is primarily focused on process logic and lacks the depth needed for detailed systems modelling, which limits its use in engineering-intensive domains.

6.6.3 Standard for the Exchange of Product Model Data

The Standard for the Exchange of Product Model Data (STEP) is an ISO standard (ISO 10303) developed to support the exchange of product model data across different Computer-Aided Design (CAD), manufacturing (CAM), and engineering (CAE) systems. It enables the representation of 3D geometry, assemblies, and Product and Manufacturing Information (PMI), including geometric dimensions and tolerances. STEP Application Protocols (APs) such as AP203, AP214, and AP242 are widely used in aerospace, automotive, and manufacturing sectors for life cycle data interoperability (Ref B62).

STEP supports both semantic and graphical PMI, allowing for machine-readable and human-readable annotations. This dual capability enhances interoperability across the digital thread, from design to inspection. However, the complexity of the standard and the need for rigorous conformance testing can pose challenges for implementation. Tools like the STEP File Analyzer and Viewer developed by the National Institute of Standards and Technology help validate conformance and improve data quality (Ref B62).

6.6.4 Open Services for Lifecycle Collaboration

Open Services for Lifecycle Collaboration (OSLC) is an open standard that facilitates the integration of software development and systems engineering tools (Ref B63). OSLC is a framework for integrating tools and data across different stages of the product or system lifecycle. It provides a standardised way for tools to communicate with each other, enabling the sharing of data and linking of artefacts (e.g., requirements, test cases, designs, and code) across diverse tools and platforms for both Systems and Software.

7 Model-Based Systems Engineering

The use of digital models has changed the way various engineering disciplines undertake their work over the past 50 or more years. Systems engineering is no different, but the transition to model-based design has taken longer than in other disciplines.

The CASG-2-MANUAL (E&T) 12-0-001 Materiel Engineering and Maintenance introduces systems engineering as the technical management framework for Defence acquisition. It describes systems engineering as:

> *A comprehensive, iterative and recursive problem-solving process, applied sequentially top-down by integrated interdisciplinary teams.*

At its core, systems engineering applies "systems thinking"— understanding a problem by defining a collection of components interacting with each other to achieve an outcome—to the design of a system. Model-Based Systems Engineering (MBSE) does not foundationally change this concept. It simply uses digital tools to clearly define the system, its boundary, components, functions and interfaces using structured data.

MBSE does not require a complete overhaul of systems engineering processes in an organisation. The same processes, gate reviews, and output documents can be applied with a model-based approach, the key difference being that the digital model is the authoritative description of the system and its behaviour.

The use of models also opens avenues for processes to change. Systems engineering is able to be less sequential, more iterative (even agile) in its implementation when it is model-based. The use of a centralised model facilitates a faster pace of change. It is necessary for Defence to adapt its ways of working to achieve the increased speed to capability required under the 2024 National Defence Strategy.

7.1 System Architecture and the Digital Thread

The MBSE approach is foundationally enabled by system architecture, and architecture models. Similarly to systems engineering bringing together various specialty engineering disciplines in interdisciplinary teams, the value of MBSE is when the system architecture model is able to bring together the physical design and specialty engineering models, often via a life cycle management model, which connects them back to the original system concept. This is referred to as the "digital thread."

For Defence projects, much of the physical detailed design work is conducted by industry. Some of this detailed design work is contractor IP that might not be shared with Defence. However, the concept of a digital thread still applies. Internal to Defence, MBSE and the system architecture model primarily support acquisition processes such as understanding the operational concept, developing capability definition documentation (CDD), evaluating and comparing proposals, assisting the interaction between Defence and suppliers, and supporting test and evaluation.

The "digital thread" is fundamental for Defence where the system architecture is able to interact:

- across the One Defence Capability System (ODCS), and bridge the Joint Mission Designs of the Strategy, Concepts, and Planning Cycle;
- with the Integrated Capability Directives of the Force Design and Assurance Cycle; and
- with the logistics, sustainment, and operations data of procured products when introduced into service.

7.2 Advantages and Challenges of MBSE

Many of the advantages of MBSE are similar to the advantages of model-based design. For example, using structured data for decision making, improved communication and advanced analytics.

As described in the CASG-2-MANUAL (E&T) 12-0-001 Materiel Engineering and Maintenance, systems engineering helps to identify and address defects earlier in the design, when they are cost-effective to address. With a model, this advantage is strengthened – such as considering the system model in multiple iterations and in different representations (ie in different documents), improves the coherence of the requirements and design – and therefore identifies design discrepancies earlier in the design. Another advantage is the ability to apply analytics to the structured data in a model allows more in-depth analysis earlier in the designs

There are challenges in implementing MBSE, and the realisation of some of these challenges has limited adoption in Defence to date. The most common challenge is a failure to keep focussed on the problem to be solved. Well-meaning teams have developed models capturing a wealth of information in models, but the efforts have not been targeted well, leading to exaggerated models with higher maintenance costs and without equivalent benefit.

As articulated in the first goal of the Digital Engineering Strategy 2024, there are workforce challenges in Defence that act as a barrier to digital engineering adoption. The skills across Defence in MBSE are not yet well established, so there is a training, experience and mentoring cost when implementing new projects. The fifth goal of the strategy identifies another challenge: access to tools in Defence's digital environment can be difficult and expensive to source.

Another challenge is managing the transition to MBSE while still delivering at pace. This, along with the workforce and tool challenges, can be what leads some teams to implement a model-added strategy, rather than model-based, which can ultimately be ineffective. However, the utility of the approach increases significantly as more areas contribute to the overall modelling environment.

These challenges can typically be mitigated with good management, a focus on the problem and the resources being developed by Defence as part of the Digital Engineering Roadmap.

7.3 Aspects of MBSE Models

As for any modelling application, an MBSE effort requires consideration of methodology, data and tools. For MBSE, 'modelling languages' provide a standardised means of representing systems using structured data. MBSE tools are system architecture modelling tools which adhere to a particular modelling language.

Common to all MBSE languages are means to represent the key aspects of a system and systems engineering approach:

- The structure of the system, including its boundary, components, properties and interfaces;
- The behaviour of the system, including the functions it performs; and
- The requirements of the system and how they are traced to the system structure and behaviour.

Some languages, notably including the Systems Modeling Language (SysML), add a fourth aspect for analytics of the system model. In SysML, this is the "Parametric" pillar, which constrains the property values of a system using mathematical relationships.

Different languages and tools support differing levels of flexibility in the methodology applied. Some have an assumed engineering process built into the structure of the language and tool, while others allow complete flexibility. Regardless, some decisions will always need to be made by the team regarding how the language and tool will be used on a particular project.

A means for capturing some of these decisions is by using MBSE Style Guides to capture modelling "rules" that the team will follow to develop coherent, consistent models. Some modelling tools can check compliance with these rules, depending on the rules and the tools used.

7.3.1 System Structure

Capturing the structure of a system helps the team align to their agreed definition of the system, its boundary, the elements which are considered external to the system, the structure of the system's

internal components, and both the internal and external interfaces. The schema may include physical, functional, conceptual, or informational aspects, as well as additional properties of the system and its components, such as dimensions or mass.

Much of the “art” of systems engineering is in the definition of the system structure: the placement of the boundary and the decisions made in decomposing the system into constituent elements. The typical systems engineering approach is to decompose according to function, which drives groupings of components based on the purpose that they serve in the wider system.

By capturing system structure in the architecture model, decisions can be clearly articulated, captured and unambiguously shared, supported by a common viewpoint that enhances information exchange and shared understanding across the team.

7.3.2 System Behaviour

As articulated in the CASG-2-MANUAL (E&T) 12-3-003 Capability Definition Documentation (CDD) Guide, the level of design by Defence is typically identified as the solution-class, and not a detailed system design and decomposition. The system design is generally available for industry to innovate and propose solutions. The focus, particularly in the early stages of a project, should instead be on *function*, which is articulated in architecture models by the system behaviour.

There are a range of ways that behaviour can be captured in different modelling languages, and many support more than one representation. Options include:

- **A focus on process flows:** sequences of actions performed by the system or by external entities,
- **A focus on information exchanges:** the sequence and timing of messages exchanged between the system and external entities, and
- **A focus on state-based behaviour:** what the system will do and how it will react to inputs when it is in different states or modes of operation.

Behaviour and structure are very strongly linked, and by associating behaviour with the elements of the system structure that perform different behaviours, the required functions of the system, and assumptions about the expected actions of external entities, can be clearly articulated. The same concepts can be equally applied to lower-level components of the system as design progresses.

7.3.3 Requirements

Requirements are usually textual statements that together form a specification of the system of interest. The CASG-2-MANUAL (E&T) 12-3-005 Function and Performance Specification Development Guide is made up of these requirements for “function” or what the system must do and performance “how well” it performs those functions. These agreed-upon requirement sets typically are important aspects of the contractual arrangements between Defence and industry.

System architecture models represent requirement statements as elements in the model and can be linked with both system structure and system behaviour to enable rich traceability between the model and requirement text.

Some requirements impose a constraint on the design, and some languages and tools support the extraction of constraints from requirements and impose them on the structure or behaviour in the model. Where this has occurred, tools can report on the satisfaction of requirements by the system or different system variants.

Architecture modelling tools can represent requirements, and in some cases are able to be used to manage requirements over the system life cycle. These tools are typically less well-featured as specific requirements management tools. Particularly for the configuration management of specifications. On larger projects, the management of requirements in an architecture tool can become unwieldy, and those projects can benefit from integrating a separate requirements management tool with the system architecture tool.

7.3.4 Analytics

As with other engineering disciplines, having engineering information captured as structured data in models has unlocked new means for analysing systems. Different tools and languages have different features to support this.

The Parametric “pillar” of SysML refers to relating property values to each other using constraints. For example, the fuel burn rate of an aircraft might be constrained based on a range of factors: the attributes of the engine, mass of the aircraft and load, airspeed and associated drag, altitude, etc. Mathematical relationships can be established to calculate the value.

MBSE tools can include features that make models executable. This is a level of simulation of the model and might include: calculating values based on constraints; stepping through process flows to assess performance; and inspecting how values change over time. This helps in several ways, including a level of verification of the model. Executable models can be made more powerful if used in conjunction with physics-based and mathematical analysis, sometimes provided by integration with third-party tools.

Some tools include checks of the model and its quality, based on pre-defined or user-configurable modelling rules. Some examples are:

- language compliance,
- coverage of requirement traceability, and
- documentation completeness.

Tool vendors are constantly adapting their software to provide new features. Recent developments include AI features to help build and interrogate models.

8 Collaborative Modelling

An advantage of modelling is improved communication across a team, as models can establish an authoritative source of truth for engineering information, and modelling tools and standards help enforce ways of representing information that are less ambiguous than the written word.

Having the whole engineering team collaboratively contribute to model development is important to realise this benefit. The whole team becomes more familiar with the data and its structure, individuals are empowered to extract and visualise the data in ways that are more useful for their needs, and changes to the model can be agreed upon and implemented more readily.

This is particularly true for abstract models, like mission engineering and Model-Based Systems Engineering (MBSE) models, but is also true for physical models, particularly when different specialty engineering disciplines are sharing information in common digital engineering environments.

Building models collaboratively requires tools configured appropriately and an agreed-upon methodology applied across the team so that outputs are coherent and consistent.

8.1 Model-Based vs Model-Added

A common trap for projects intending to introduce models to their work is establishing a small "modelling team" to work alongside engineers who are otherwise focussed on extant non-model artefacts. Rather than being "model-based," this approach is more accurately described as "model-added" and is illustrated in Figure 21.

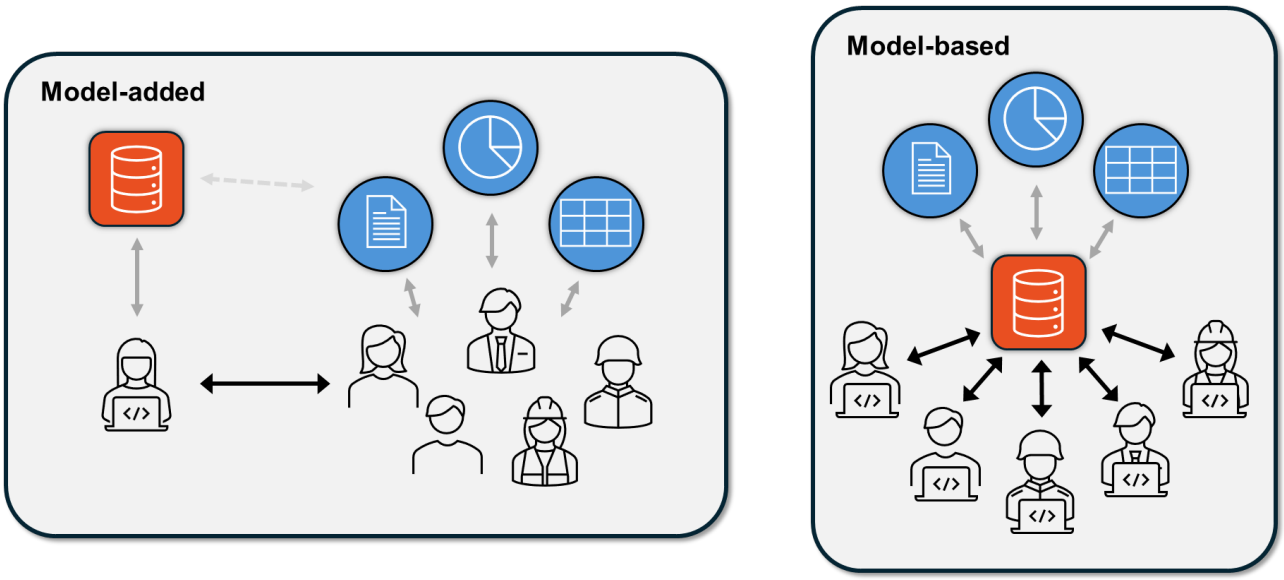

Figure 21: Model-Added vs Model-Based

In many cases, this "model-added" way of working becomes inefficient and ineffective. The engineering work is effectively double-handled—once by the off-tool engineers, and again when the modelling team must convert it into the agreed-upon data standards and tool sets. These types of approaches often lead to a perception within the team of models adding overhead for little benefit.

For it to be "model-based," the wider team needs to understand and contribute to the model's development directly. This does require a wider understanding of the modelling data and tool across the team, and an associated learning curve, but with proper model configuration management and governance, it eliminates the double-handling of the effort and leads to models far more beneficial to the project overall.

8.2 Modelling Environments

Collaborative modelling requires software environments that are configured for team members to read, access, and modify the model content and data as appropriate for their role in the team. Different tool vendors apply different approaches in supporting collaborative modelling, and many have established tool ecosystems that support simplified exchange of data between tools in their environment.

8.2.1 Server/Client Environments

Many modelling tools are highly specialised pieces of software with notable computing requirements. A common tooling arrangement for collaborative modelling is specialised modelling tools installed on an individual's machine (or virtual desktop), connected to a central server or database. The method of collaboratively editing the server content varies.

8.2.1.1 Collaborative Real-Time Editing

Some tools support multiple users freely modifying model content at the same time, such as the approaches used for document editing in Google Docs or SharePoint. As long as two users are not changing the same element at the same time, users are able to freely modify what they need as shown in Figure 22. This shows a blue user editing the blue element and green the green element. Updates occur real time, and do not require check-in. This requires a persistent connection between the client and server to keep the model up to date and minimise conflicts.

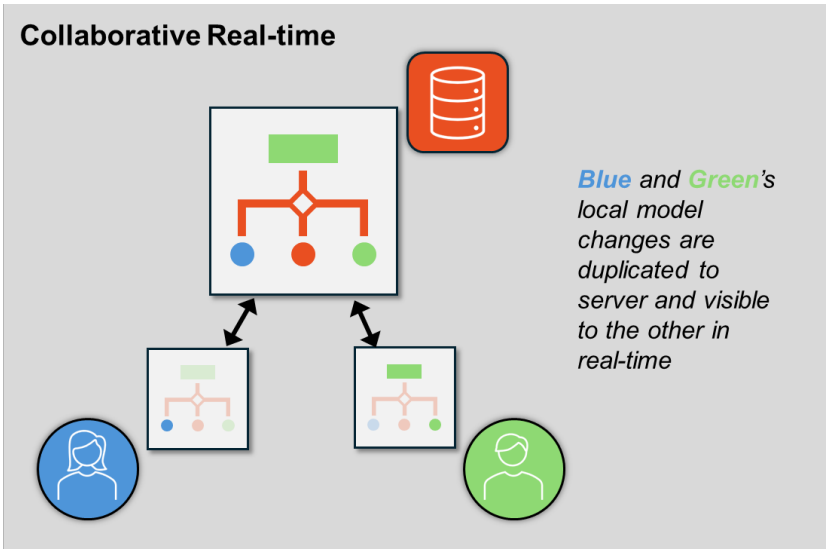

Figure 22: Collaborative Real-time Editing

8.2.1.2 Model Lock/Unlocking

An alternate approach is taking a copy of the model from the server to the client, and having users lock out parts of the model for them to edit as shown in Figure 23. Other users can see what is locked, but not the changes being made, and they cannot edit that content until it is released. This approach requires check-in / unlock of the model element and can enable offline edits. This approach is used by the CATIA Magic suite of architecture modelling tools.

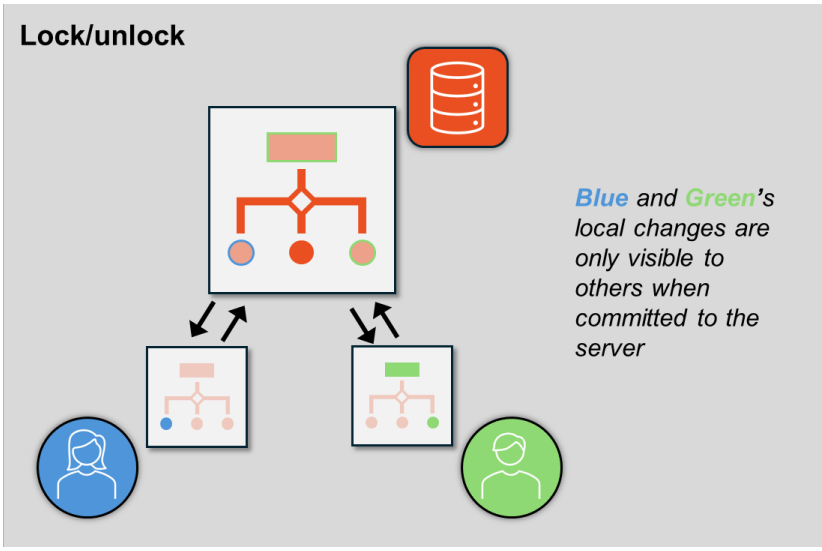

Figure 23: Model Lock/Unlock

8.2.1.3 Highly Modular Models

Instead of a large model that multiple users need to access at the same time, some tools instead favour many smaller models that reference and load each other as required. They are of the scale that only one user should need to access any one model at a time, and these can be locked to a user in their entirety as shown in Figure 24. Here the blue and green components are considered sub-systems/components and are represented in their own models.

This approach has been used for CAD models, where individual components— down to standard nuts and bolts—might each have their own models stored in extensive libraries for reuse. A higher-order model collects these components together to build more complex systems, with read-only access to each component model used. If a component model is modified, higher-order models can detect the change and notify users of potential conflicts.

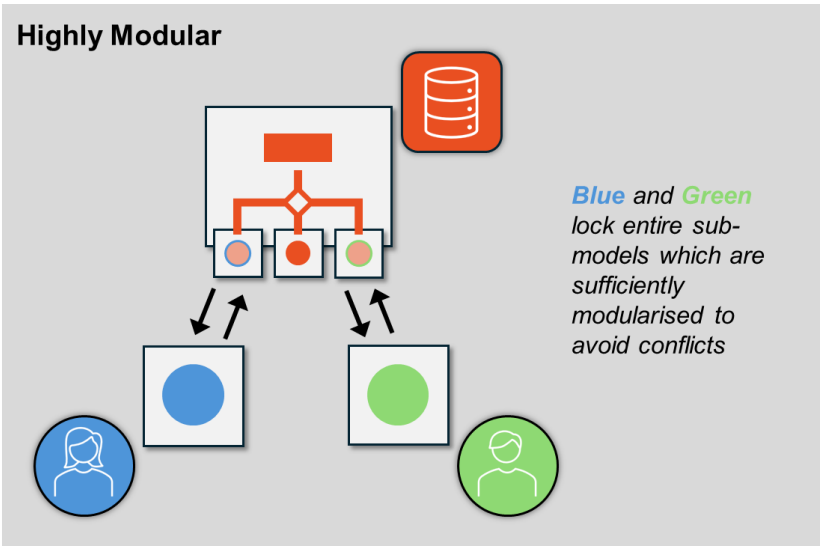

Figure 24: Highly Modular Models

8.2.2 Browser-Based Environments

Some tool vendors either include, or have fully transitioned their software suites to, browser-based tools that do not require specific client software to be installed on a user's machine. An example of this is IBM's Engineering Lifecycle Management (ELM) suite, which includes tools for requirements management, system architecture, test management, workflow management, and configuration

control, all in browser-based apps as shown in Figure 25. These use typical web browsers for access.

These tools have minimal data being replicated to a user's machine, with changes being applied to the server directly.

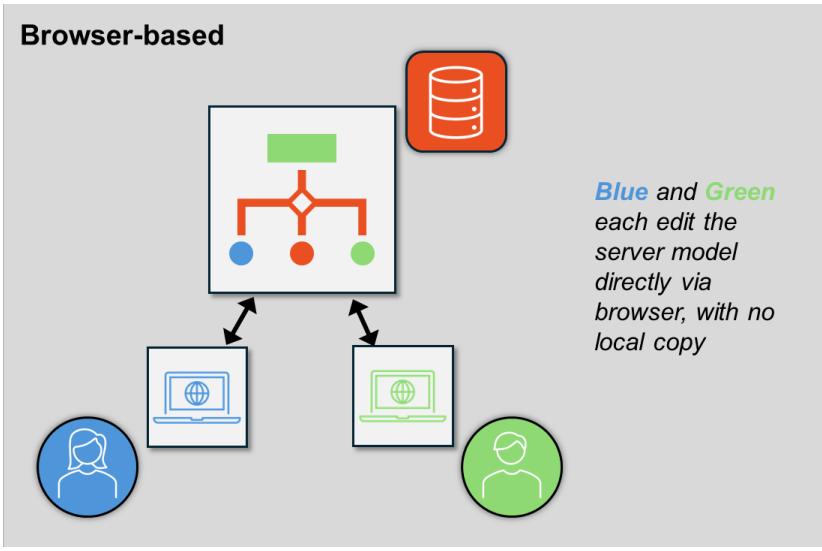

Figure 25: Browser-based Environments

8.2.3 Distributed Environments

Another approach, particularly useful for analytic and mathematical models or other code-centric models, is to use distributed source code management (SCM) techniques, like Git as shown in Figure 26. This approach takes a clone of the entire repository to each user's machine, and the modelling software works locally with that clone. Collaborative modelling is reliant on branching and merging of the repository and its content, tracked and managed by the SCM using workflows agreed upon by the team. This type of modelling does rely on agreed upon standards to minimise issues with merge and is more akin to software development.

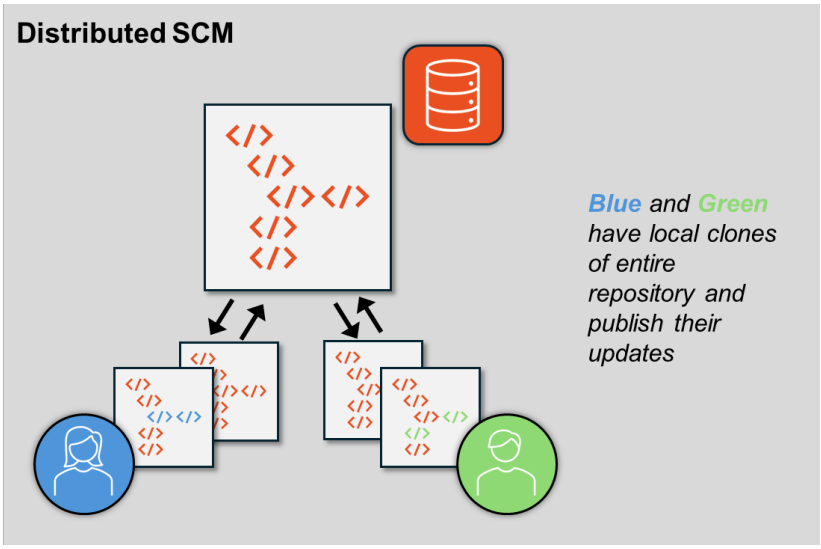

Figure 26: Distributed Environments

8.2.4 Alternate Visualisations

Not all users need the same level of read-write access to model content for their role. Senior leadership, for example, may not require the technical content of detailed engineering models, and simplified views or summary metrics would be more useful.

Some tools support simplified, customisable views of models for this purpose. Others, like Power BI, are specialised visualisation tools, able to connect to sources of data and present it in different formats as shown in Figure 27. The model presents to the blue and green engineers as the model / data entry, but to orange manager as a bar chart report.

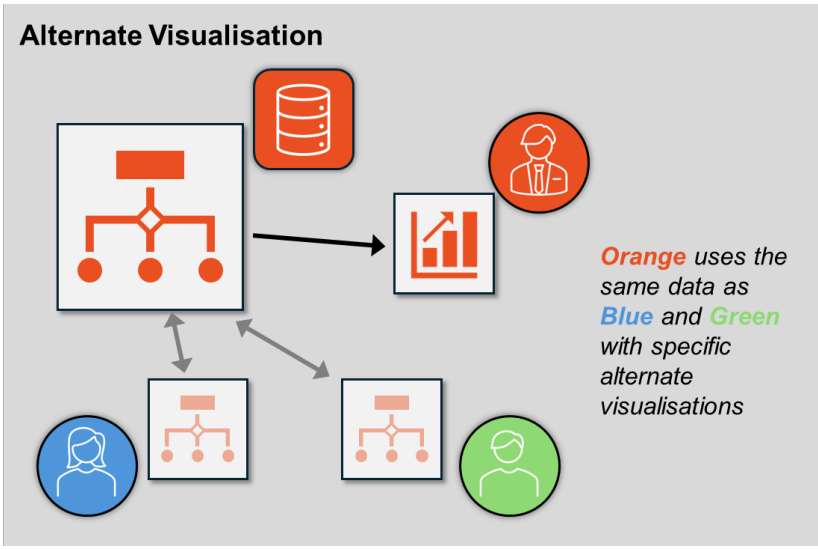

Figure 27: Alternate Visualisations

9 Model Federation

A federated model is a collaborative, decentralised approach where multiple independent entities work together under a shared framework, each maintaining its autonomy while contributing to a collective goal.

Mature digital engineering (when in the "Run" phase) usually involves different models intended to solve different problems interacting with each other to share data in a federated environment. This is true of:

- Different models within the same project for different stages of the life cycle of the system, or different aspects of the design;
- Different models across Defence for different stages of the One Defence Capability System (ODCS); and
- Different models for a system owned and managed by different organisations (e.g., Defence vs industry models, Defence vs Defence partner models).

The second principle of the Modelling and Simulation (M&S) Policy concerns M&S reuse. It states that developers of new models should consider future reusability of those assets by Defence in a federated environment. This assessment should include whether the investment in reusability is valued.

This topic focusses on principles that govern how models can be structured to support federation. Further information is available on the exchange of models and model data.

9.1 Differences in Modelling Structures

Building useful models requires models to be built to solve specific problems. Models across Defence will be structured in support of solving those problems. This will lead to different models describing the same thing that look very different, even if they are modelled in the same tool. This is particularly true of system architecture and design models, which are typically more abstract and so have more flexibility in their representation.

Models will likely vary along these dimensions:

- **Hierarchical decomposition:** A model may be decomposed consistent with different hierarchies. For example, systems models may be decomposed into constituent component models, forming a multi-layered parts-whole hierarchy. In a control hierarchy, the relationships typically represent organisational reporting structures or command structures. Other hierarchies might indicate categorisation, where child elements in the hierarchy inherit the characteristics and behaviours of their parents, underscoring how problem framing through hierarchy shapes our approach to solutions. Appropriate hierarchies are essential.
- **System and model boundary:** When modelling a system, the modeller will need to decide what is internal to the system of interest and what is external. Internal components are generally modelled, and external elements typically provide boundary conditions to the model. The location of the boundary is subjective and depends on the purpose of the model. For example, an engineer who is responsible for the detailed design of a component system will need a different view than a systems engineer who is responsible for managing the interfaces between systems in a complex system-of-systems environment, or an architect who needs to demonstrate the linkage of the architecture to enterprise strategy.
- **Subsystem structure:** Similar to drawing the system boundary, different models will group components within a system into different subsystems to view the system's internal structure in different ways for different purposes. For example, when designing a ship, a logical representation might focus on functions: propulsion system; communications system; Heating, Ventilation, and Air Conditioning system; electrical system; etc. These systems might span the vessel. To build and maintain the ship, a perspective that focusses on physical layout, categorised by decks and compartments, might be more useful instead of functional systems. This also raises the question of model depth, whether it is sufficient

to say the ships 'detects', or if we must detail the specifics of the radar (as subsystem) used, and its behaviour, to explain when and how detection occurs.

- **Descriptive versus prescriptive:** Some models are prescriptive (e.g., requirements models, “to-be” architecture models, performance specifications). Other models are descriptive (e.g., as-built specifications, life cycle management models, operational test results).

It is possible for a project to connect different models about a particular system of interest that represent the system in different ways.

For example, a Concept of Operations (CONOPS) model may describe the system’s behaviour from the point of view of the user, whether that be an operator, commander, or the system/platform, each offering distinct perspectives and interpretations of actions and reactions. Several variants of the CONOPS could demonstrate behaviours under different external conditions. System models could define the structure and performance of software and hardware elements. Those models could also be linked to physics-based and mathematical models that estimate the performance of the overall system or its subsystems under various loading conditions. Cost models could provide estimates of the life cycle cost of developing and managing the system.

Typically, each of these models would be linked using formal relationships, forming a comprehensive integrated model, as illustrated in Figure 28. The relationships support an important function in that they enable the analysis of secondary or tertiary effects of changes. For example, the impact of a system design change could be analysed in terms of impact on cost and performance.

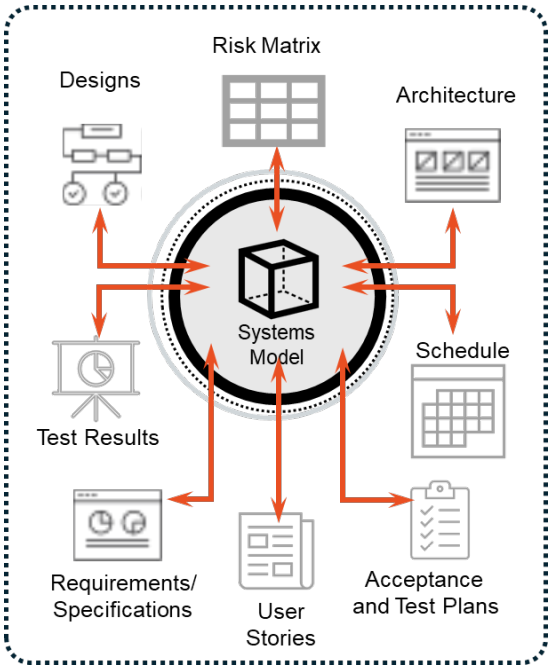

Figure 28: Integrated Model Example

9.2 Inheritance and Reuse

The use of models to represent systems creates opportunities to reduce rework using the concepts of inheritance and reuse. These concepts can apply within a model but are particularly useful to support methods of model federation.

Reuse is typically about identifying elements that are duplicated in multiple places. Instead of modelling each instance individually, these elements can be modelled once and reused in different contexts; for example, creating one model of an AS/NZS 3112 power socket and reusing it for every power point in every building design. Since the model is reused across multiple instances, any changes made in one place are automatically reflected everywhere it is used.

Inheritance is about identifying elements that are similar with minor variations. Instead of re-modelling the similar parts individually, a common parent element is created once, and each variant inherits from it, requiring only the unique differences to be modelled. For example, if considering different options for how a C-17 air frame is loaded for different missions, there could be a generic “unloaded” model that captures all the information about the plane itself. Child models could then be created that inherit all the information about the plane and describe the different options for loading. Since the common information about the plane is inherited everywhere, there is a single place it can be changed, and all the children will be updated automatically.

Not all models or modelling tools support the concepts of inheritance and reuse, but they are very powerful tools to support complex modelling efforts.

9.3 Methods of Model Federation

There are different ways models can be connected to each other in federated environments.

Bottom-up federation relies on model reuse of components in composite models as shown in Figure 29. It is about creating libraries of common models that are readily available to include within other models. CAD tools usually have libraries of standard fasteners or fittings that can be reused; mathematical analysis tools usually have libraries of standard equations.

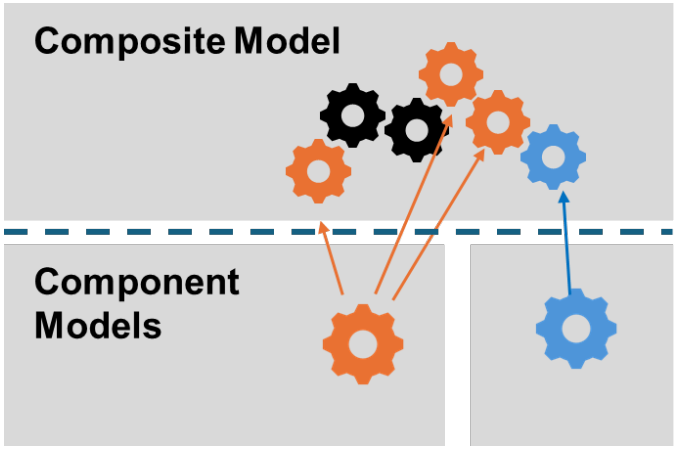

Figure 29: Bottom-up Federation

This method of federation relies heavily on standards. Models intended for reuse must be defined in clear, standardised model styles and formats to ensure they can be interpreted and applied consistently. This is simpler in models from the same tool created for similar purposes, like in the examples above.

Building Information Modelling (BIM) relies on standard exchange formats so models from different specialty engineering disciplines (electrical, structural, mechanical, etc.) be integrated and aggregated in the BIM model. This allows the complete building design to be viewed in a common view and enables features like clash-detection analysis.

Top-down federation relies on model inheritance as shown in Figure 30. It is about defining simple, generic descriptions of systems in higher-level models, which are inherited by more detailed representations in lower-level models. The lower-level models expand on what was inherited for a particular purpose.

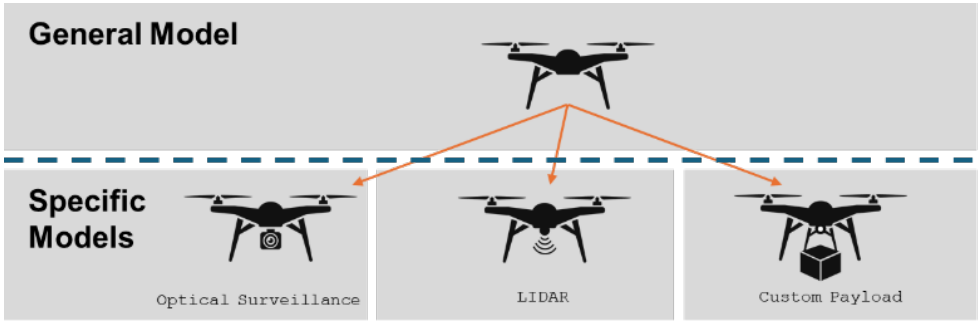

Figure 30: Top-down Federation

This method of federation is more flexible; if the higher-order models are sufficiently simple, the elements can be easily inherited by models created for very different purposes (potentially in different tools via import/export or some sort of transformation).

Traceable element federation is the simplest method of connecting models, but also the least efficient. This connects models by identifying elements in different models that have some form of relationship (or represent the same thing) and creating a link between them. This is particularly common between different disciplines, tools, or life cycle stages.

Powerful queries across models can be built using these relationships. Nothing is automatically aligned or updated as elements change, but flow-on impact of changes (for example, using "suspect link" tool features) can be interrogated across models to determine which might need to be manually updated.

9.4 Use Cases of Model Federation

9.4.1 Shared Modelling Framework for Mission Engineering

A Shared Modelling Framework can be used to reduce the need for rework in mission engineering. This approach relies on federated seed, base, and mission models, which use inheritance to define systems generically and then specialise their attributes in the context of a particular mission scenario. This approach has been applied for system architecture and design models using the Systems Modeling Language (SysML). These relies on the concepts of seed, base, and mission models as shown in Figure 31.

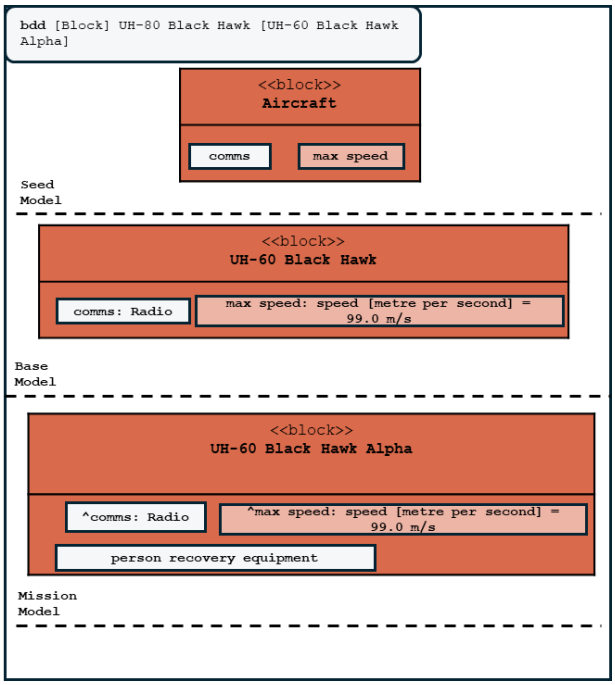

Figure 31: Seed, Base, and Mission Model Representations (From Ref B8)

- The **seed** model has generic properties for classes of systems that serve as “blueprints.”
- The **base** model inherits from the seed model but identifies specific systems and their standard attributes.
- The **mission** model adjusts values as required and adds additional details as necessary to represent the system in the context of a mission.

Defence has moved toward the adoption of these concepts of linked models through its Joint Mission Design which is detailed more in the Defence Federation section.

9.4.2 Using Industry Models for Proposal Assessment

When conducting model-based proposal assessment, vendor models, built in accordance with a standard defined by Defence, are reused in models of the operational context to assess the proposal’s performance with identifying pass and fails for the proposed solution against the concept as shown in Figure 32. The form of vendor models must be tightly controlled for this to work seamlessly.

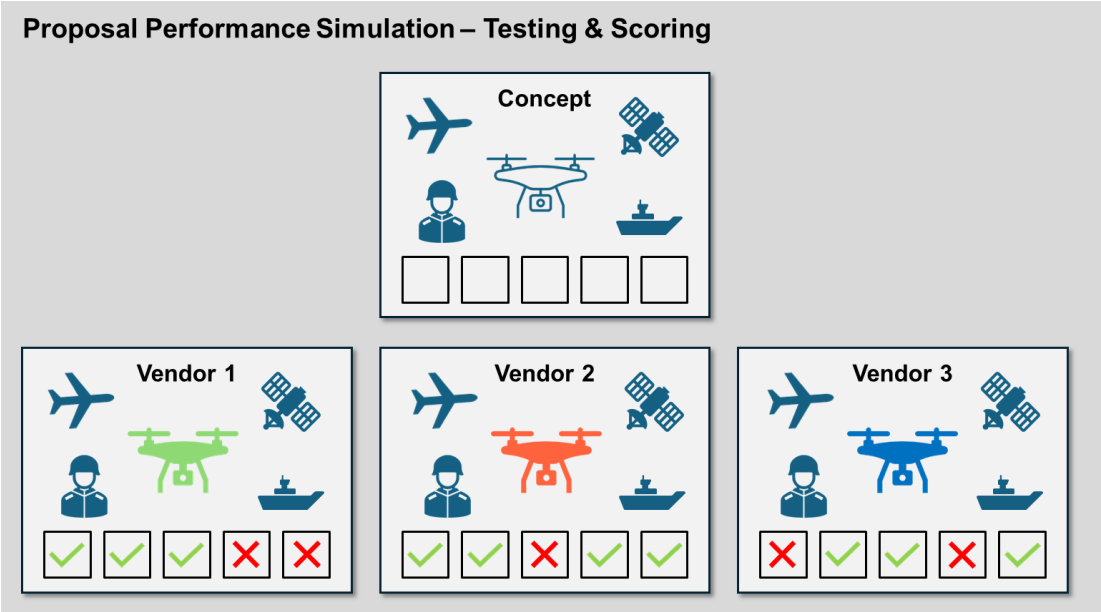

Figure 32: Model-Based Proposal Assessment

9.4.3 Industry Traceability to Defence Specification

Industry partners have developed their own modelling practices and methodologies, which may not align with the standards that might be mandated by Defence for proposal responses. This can be mitigated by industry maintaining a high-level model in accordance with Defence's requirements, and tracing relevant artefacts to an equivalent in the working design model, as illustrated in Figure 33.

This also assists partners to maintain control over their intellectual property. Design detail and sensitive information can remain in the design model with only the necessary information provided in the response model.

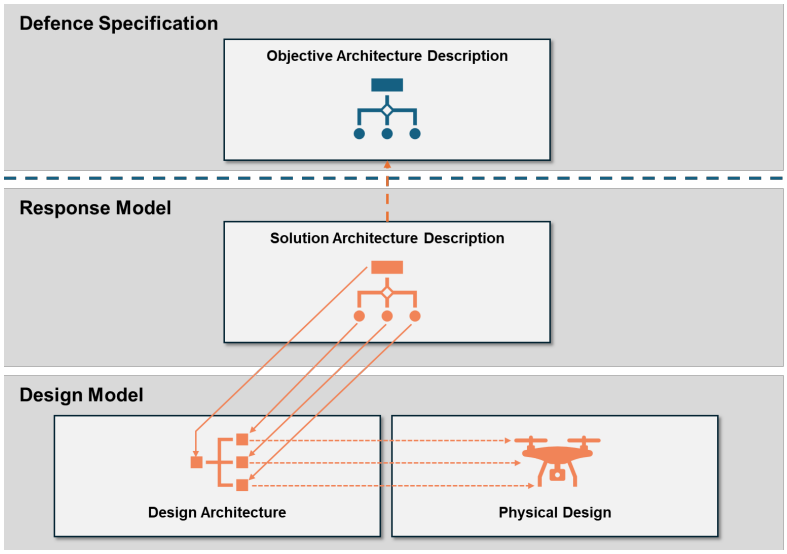

Figure 33: Traceability Between Response and Design

10 Model Exchange

The first principle of the Modelling and Simulation (M&S) Policy concerns model interoperability. It states that interoperability with the Defence community, Commonwealth agencies, allies, partners, and industry should be considered as part of the planning for any modelling effort. Defence's guiding principles for digital technology choices articulated in the Defence Digital Strategy 2024 include enabling interoperability with Defence's military partners for a data-driven approach. Whether, and how, models and model data is exchanged is an important consideration.

The topic on model federation describes some techniques that can be applied to federating models. This topic focusses on some key principles for establishing the data exchange between models. More information will be provided in the Digital Engineering Data and Data Sharing Guiding Principles.

10.1 Remember the Purpose

Useful models are built to solve a specific problem and/or to answer specific questions. The data in a model can still be largely geared toward solving that problem, and this needs to be considered when determining whether that model or its data can be used for your modelling effort.

All models are approximations of the real world, and the way in which a different model has approximated may not be appropriate for your model. It is essential to understand the model approximation before accepting data from another model or database.

10.2 Define a Schema

Data from a model should be produced according to a defined schema,[3] so that it is consistently produced and easily readable by other tools and users. Many modelling tools provide various output formats for their data, and some of these are customisable. A simple schema is shown in Figure 34 shows some of the relationships such as 1:many, parent:child, etc. that are possible.

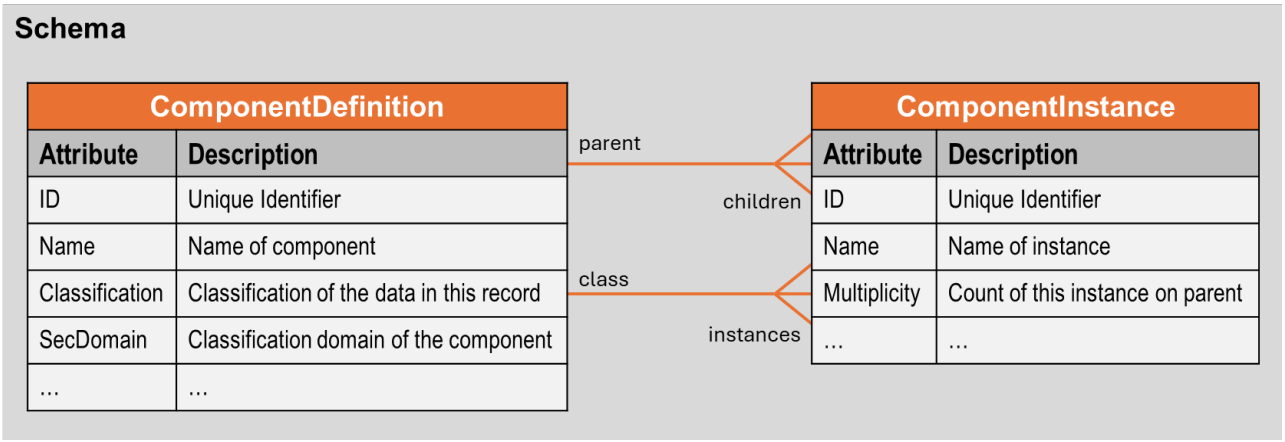

Figure 34: Model Schema

Beyond output formats, some tools support (and some standards mandate, such as the Systems Modeling Language, SysML v2.0) particular Application Programming Interfaces (APIs) for data exchange, which allow for more dynamic exchange of data between tools. These APIs have defined schemas for producing and receiving data.

However, the schema is more than the output format or API, similar to a modelling effort is more than just data and tools. The methodology used in building models influences the **meaning** of the

[3] Schema in the context of digital engineering refers to the formal structure or blueprint for organising and managing data and models within a digital ecosystem, enabling interoperability, traceability, and integration across tools and platforms.

data in the model, making it essential to understand not only the label or variable name, but its role and context within the model.

For example, a schema might have a “security classification” attribute. If describing the Defence Secret Environment (DSE), that attribute might be used to capture the classification that the DSE exists (OFFICIAL) or the classification that the DSE is accredited to store (SECRET).

It is important to describe the meaning of different attributes and values so that users of the data can fully understand the schema and use its data correctly.

Schemas can be defined using a form of modelling as well, called “data modelling.” Data division has resources available on data modelling, including the Defence Data Modelling Framework.

10.3 Determine Transformations

When different models using different data and tools are used for different designs, the schemas for each will be different. Moving data from one to the other, particularly in a repeatable or dynamic fashion, will require the data to be transformed from one schema to another as part of the exchange.

The transformation will be a mapping between schemas and a set of rules to migrate data from one schema to another as shown in Figure 35. Where the data structures are similar—for example, where the same modelling language, data and format are used in different modelling tools—this may be as simple as mapping a set of attributes with different names to each other. Where the underlying data structures are fundamentally different, the mapping rules will be harder to define.

It is important to remember that the more the schemas vary, the greater risk of data loss and ambiguity.

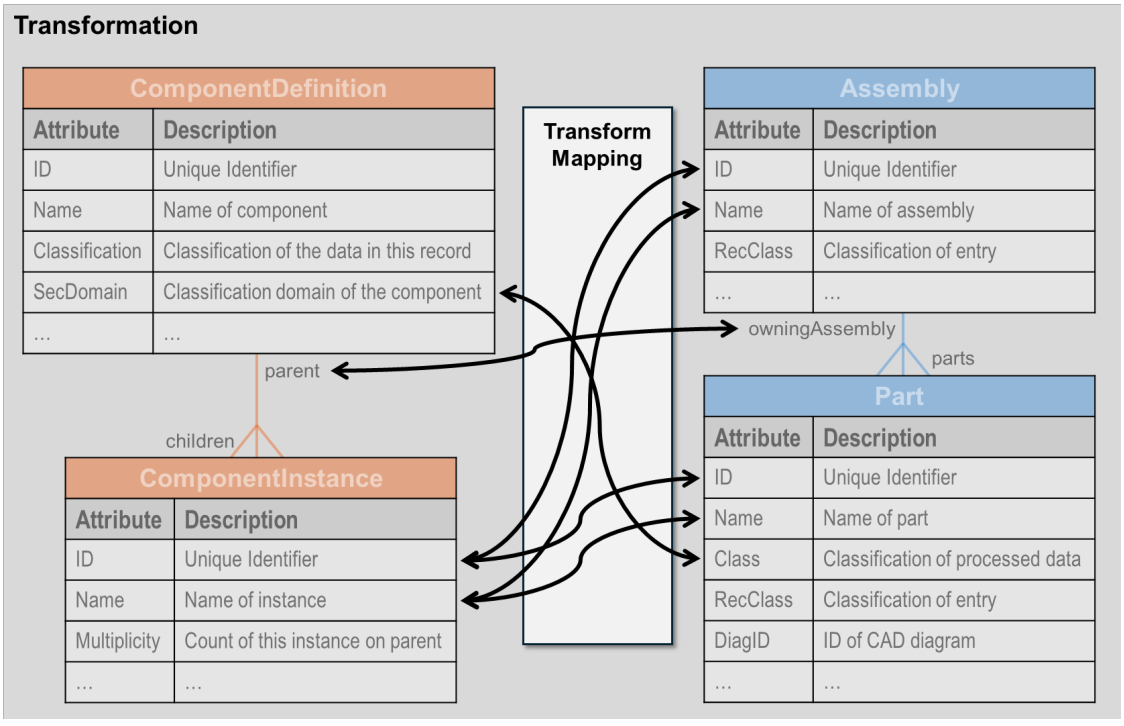

Figure 35: Model Transformation

10.4 Reference and Master Data

The Data Division provides guidance and services supporting the definition and maintenance of Defence Reference Data and Defence Master Data. As a general principle, existing Reference and Master data should be reused wherever possible, and where a model is producing data that is useful across Defence, its data should be captured as a Master Data source for others to reuse.

11 Model Verification and Validation

Any model is an approximation of the real world, and understanding that approximation and its limitations is important in determining whether the model's outputs are appropriate, reliable, and suitable for the intended use. As systems require to be verified and validated to confirm they are ready for use, models themselves should undergo verification and validation (V&V).

The Defence Test and Evaluation Manual describes verification as a process to determine whether a product meets specified requirements.

In the context of models:

- Verification focusses on whether the model is built correctly: is it in accordance with its methodology and data standards.
- Validation is a process to determine whether a product performs as specified or intended. Validation focusses on whether it meets its intended purpose or contributes to solving the problem it is intended to help solve.

11.1 Verification of Models

Verification involves assessing models against standards, methodologies, and requirements across all tools in use. It checks that the models comply with relevant data standards or modelling languages and follow the agreed-upon methodology for building the model. Modern modelling tools can help with this by enforcing formal language rules and data standards, sometimes even enforcing team-defined methodology rules.

Tool support greatly aids verification, but it is not foolproof or complete. Modelling tools will usually enforce compliance with the implemented modelling language and data standards automatically, and some can be configured with custom rules to enforce specific modelling decisions defined in the team's methodology. This can catch many errors—for instance, the tool might prevent an illegal connection or flag elements that violate naming conventions. However, no tool can enforce every rule or decision. Not all tools allow fully customisable rule definitions, and many methodological guidelines (especially project-specific conventions or nuanced best practices) cannot be encoded as automatic checks. There will always be modelling decisions agreed upon in the methodology that rely on the modellers' understanding and discipline to apply consistently.

Verification must ensure that every model element is traceable to a system requirement or user need, not just syntactically correct. Without this traceability, a model may be internally consistent but fail to solve the intended problem.

Verification also includes verification of the data input into the model. Is the data entered correctly? If taken from another data source, is the data correctly mapped to the correct elements in the model? For system architecture and design models, has the correct language element been chosen for the type of data ingested? For physics-based and mathematical analysis models, are the correct equations being used?

Where models are executable or involve a simulation, verification should include executing the model as soon as its feasible and meaningful, ensuring the outputs are as expected for the entered inputs, rather than relying solely on the complex, system-wide verifications.

Models should undergo peer review by a suitably experienced specialist who is familiar with the data, language, and methodology being applied.

11.2 Validation of Models

Validation assesses the models against the problem to be solved, and against the real world model.

Recognising that all models approximate the real world to some extent, validation needs to focus on whether the model is accurate enough for its purpose. This requires a clear understanding of what the model is trying to achieve. It should also consider whether the model aligns with the

system-level objectives, ensuring it meaningfully supports the broader goals and operational context of the system.

Validation is likely to find aspects of the model that could be made more accurate, and stakeholder engagement and review can often lead to suggestions for how the model could be changed or “improved” to align with their understanding. Making such changes should be done carefully. If they do not contribute to the model’s purpose, such changes might make the model more *accurate* but will not make it more valid. This can quickly lead to exaggerated, poor performing models that are difficult to maintain, more resource-intensive, and delay delivery due to increased V&V overheads.

Validation should look at the model from the perspective of the problem and confirm it meets its need.

For executable models and simulations, the model can be validated against the real world by comparing the output of the simulation to measured data in the real world. By confirming results are within appropriate error ranges, confidence in the model is improved and it can be more confidently applied to scenarios that might not be able to be replicated in the real world.

12 Model Curation and Reuse

Models supporting digital engineering in Defence can be very useful to solve a particular problem, to structure data, and improve communication. There are opportunities for data from models to be exchanged to inform other work in Defence, and for models to be incorporated into federated environments to reuse instead of rebuilding model content.

These opportunities come at a cost. Developing models can be expensive, particularly developing verified, validated models with the level of detail necessary to support physics-based and mathematical analysis. The maintenance and configuration management of models for their ongoing reuse, especially these detailed models, can introduce significant overhead.

The third principle of the Modelling and Simulation (M&S) Policy concerns M&S asset management. It states that models should be actively managed to support interoperability and reuse. This assessment should include whether the investment in reusability is worth the investement.

In assessing future reusability, there are a few points, which should be considered:

- The purpose of the model,
- How to assess maturity of models, and
- How models will be curated and stored.

12.1 Model Purpose: Should It Be Reused?

Not all models need to be enduring. In assessing a model's purpose, the model's usefulness over time should be considered and captured as part of its methodology.

Some models will form the basis for an ongoing design and sustainment effort. These can provide valuable insights into the reason for design decisions and inform future replacement or obsolescence analysis. These models should be maintained across a project's life cycle.

Some models are built to be modular. They are developed for the express purpose of reducing rework and being reused in higher-level models.

Other models, particularly some detailed analytic models, might be helping to analyse a specific problem at a point in time. Perhaps they are informing a trade-off decision or analysing specific contingency mission variants. These types of models might be useful to assist with a solution, but once that complete, they provide little use. Those models should be archived for future reference but provide little benefit outside their specific analysis and should not be maintained.

Knowing a model's purpose, defining it, and making that available is important if a model is to be reused. Potential users of a model need to understand its purpose before reusing it to ensure it can also serve their purpose and does not approximate the real world in a way that is detrimental to their analysis.

If there is value in making a model available for reuse, a key consideration is when it is considered mature enough for others to rely on it.

12.2 Understanding Model Maturity

The concept of "maturity" of models can be difficult to define. With models approximating the real world, it can be ambiguous when they are considered "complete." There is often more that *can* be done to improve models, but these efforts might not provide additional value for the problem the model is intended to solve.

For some models, determining a "good enough" threshold is straightforward. For example, the physical design model of a component might be considered mature when it is cleared for manufacture by a design authority. Other models, like those used to support the design of a product through its life cycle, are intended to evolve over time, and it is much less clear when those models are "mature."

12.2.1 Approach to Measuring Maturity

Ultimately, a mature model is a model that answers the questions of your analysis with a sufficient level of confidence in its results. This essentially can be simplified to having a verified, validated model.

Various metrics have been proposed to measure model maturity, particularly for monitoring progress over time. Sometimes means of assessing against these metrics can be built into modelling tools for ease of reporting. These metrics might include some as shown from Figure 36:

- **Model size:** This is generally a poor metric to use on its own, as it correlates more closely with the amount of work put into the model, rather than its fitness for purpose. Additional, unnecessary work will lead to bigger models that might be **less** useful and more burdensome to maintain.
- **Model volatility:** This focusses on the rate of change to the model. Assuming the rate of effort being applied is similar, a more stable model indicates it may be more mature.
- **Model quality:** This is difficult to measure quantitatively, but indicators like adherence to modelling standards, team adherence to agreed methodologies, and whether model outputs match what is expected based on its inputs can be useful indicators.
- **Model validity:** This can also be difficult to measure quantitatively but does focus on the model's purpose. A valid model is a useful model.

Figure 36: Model Maturity Metrics

12.2.2 Application in Practice

How the maturity of a particular model will be assessed will be a decision for the specific team to decide as part of its methodology.

For some it might be model approval by a design authority, for others it might be particular metrics, and others might employ a mix.

For models that support multiple stages of a product's life cycle, different aspects of the model might mature with different stages of that life cycle. Which parts of the model are expected to mature at what time, and which metrics might be useful to measure maturity might be defined differently for different life cycle stages.

The following process, from Measuring MBSE Model Maturity – A Library Supported Approach, is a useful approach teams can employ. It is intended for Model-Based Systems Engineering (MBSE) models but can be applied more broadly. The high level steps of this are shown in Figure 37.

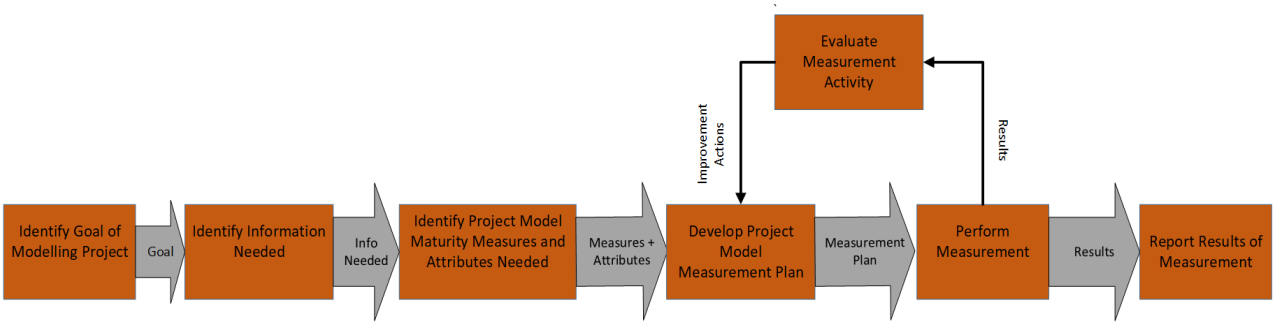

Figure 37: Model Maturity Assessment Method for Use on Projects (From Ref B83)

12.3 Model Curation and Storage

Part of the challenge in making models available for reuse is how they are stored long term. Heavily reused models may be considered master data, which should be managed appropriately in accordance with the Defence Master Data Management Framework.

Options for storing models will often be driven by the choice of tools used, as different tools will store data in different ways. Some tools create files that can be stored in the Defence Objective Records Management System or another kind of data management system. Others create databases, either of individual models or as a collaborative repository storing many models. Databases must be appropriately managed and sustained to ensure data integrity.

As with any digital record, models and model data can be lost. Capability Managers should maintain centralised records of models under their scope and responsibility that are available for reuse. These should clearly capture information about each model, including its scope, purpose, and currency.

Models not intended for reuse should be archived and stored in accordance with any applicable data-retention policies in use by the team. A description of the model and its purpose should be recorded and stored either in or with the model.

13 Model Configuration Management and Governance

To rely on models and model data as authoritative sources of truth, a level of assurance and governance needs to be applied to establish and maintain confidence. This is particularly true of models intended to be reused across the enterprise. Models that exist as part of a federated environment may have many interdependencies with other models, which can cause conflicts if changes to those models are not managed appropriately.

A single, blanket approach to model configuration management and governance is not appropriate, especially as Defence focusses on increased speed to capability. There are different approaches to managing models available, and the most appropriate should be selected as part of the methodology of a modelling effort. Different approaches might be applied to different kinds of models or aspects of models within the one project.

13.1 Scaled Configuration Management of Models

For small-scale, low-risk models with clearly bounded scope and minimal system dependencies, the level of configuration management rigor may be proportionally reduced. However, the model itself shall still be treated as a configuration item with controlled version identification and traceability. In such cases, formal review and approval may focus primarily on the generated artefacts (such as Operational Concept Document (OCD) and Function and Performance Specification (FPS)), provided the underlying model version used to produce those artefacts is uniquely identified, reproducible and baselined. Model changes should remain traceable even if a simplified approval workflow is applied.

This can be a useful technique when digital engineering is in the **Crawl** stage. The developers of the model can focus on updating the model freely to quickly demonstrate its value to the task. As the effort moves into the **Walk** and **Run** stages, models that are central to a product's design will likely require an approach that is more robust.

13.2 Configuration Management of Model Changes

For models that are central to system architecture, heavily reused across stakeholders, interconnected with other domain models, or that influence verification, certification, safety, security, mission performance, or contractual baselines, the model itself shall be places under formal configuration management. In this approach, the model is treated as a controlled configuration item with formally established baselines, and proposed changes are assessed, reviewed, and approved prior to incorporation as shown in Figure 38. Change evaluation shall consider technical impact, downstream dependencies, and alignment with system requirements. Version history, change rationale, and approval records shall be retained to preserve traceability and maintain digital thread integrity.

Although modelling tools may provide capabilities such as versioning, comparison, branching, and merging to support these workflows, effective configuration management depends on defined governance, change authority, and impact assessment processes rather than tooling alone. While this level of control can introduce additional process overhead and may slow iterative development, it is appropriate for models that are central to the system architecture, heavily reused across projects or stakeholders, contribute to verification or certification evidence, support mission safety or security-critical functions, or form part of contractual baselines. In such cases, maintaining controlled evolution of the model is essential to preserve system integrity and digital thread continuity.

Model federation is a useful tool for splitting model content between different models, which can help reduce the configuration management burden.

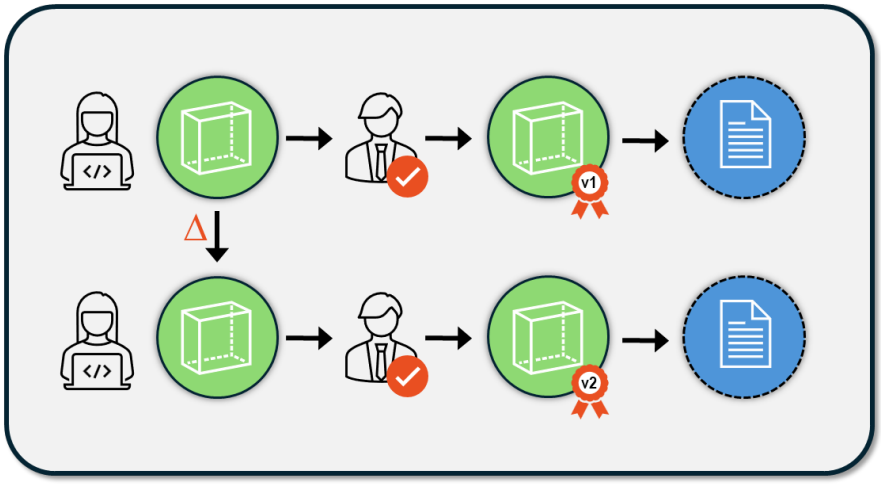

Figure 38: Configuration Management of Models

13.3 Agile Model Management

The use of models in digital engineering lends itself well to more agile approaches to managing change. Most tools can track changes as they are made, compare model versions, and roll back changes if required as shown in Figure 40.

An agile approach relies less on change analysis and approval before changes are applied, and more on peer review and short update cycles to adjust as required. Improved models and analysis can be demonstrated quickly. Rather than confidence in the model being generated by the authority of the review, confidence can be built by assessing the validity of the model's outputs.

Where a model is heavily reused across a project or across Defence, this speed of change can be counter-productive due to flow-on impacts. Teams managing other models may not be appropriately resourced to implement changes to their models accordingly.

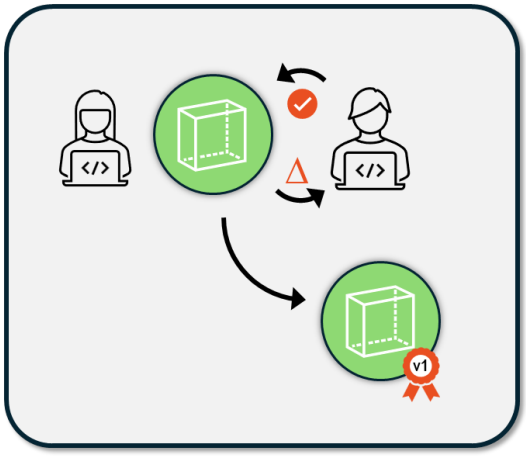

Figure 39: Agile Model Management

13.4 Configuration Management Features of Tools

Different tools and different model types have different configuration management features, which may influence the configuration management processes selected for your methodology, or the tools you elect to implement.

Some tools have very flexible version tracking built into them, which tracks version history down to individual model artefacts and supports branches and merging. They also support the creation and tracking of model variants with different configurations, with engineering workflows integrated directly into the tool. This is more common in life cycle management tools.

Other tools support version tracking only at the whole-of-model level, requiring manual comparison between model versions to determine which artefacts changed. This is somewhat less flexible for large models.

Many tools do not have their own configuration management features built in and rely on separate specialised configuration management tools to manage the configuration of models. The files generated by the modelling tool are handled by the configuration management tool in the same way a document might be. Comparison of different versions of the model can be more difficult, as the configuration management tool may not be able to read the model file to perform such an analysis. Many vendors offer engineering ecosystems, where tools for different purposes are tightly integrated in a single vendor tools environment. These are specially designed such that configuration management components can read and process model files, or tool databases are appropriately integrated behind the scenes to provide a similar outcome.

Configuration management of model files is common for models in the physical design and specialty engineering functional area, where many different models from different tools and disciplines need to be brought together, and a common configuration management tool across all disciplines is useful. In this case, life cycle management tools often provide this capability.

Some tools do not generate “files” but instead interact with a separate database. In this case, the configuration management features of the database must be used.

Some physics-based and mathematical analysis tools rely on models that are captured in a format closer to source code than databases. Source Code Management (SCM) tools, like Git, can be particularly useful to manage models of this kind.

14 Modelling Best-Practice Tips

Focus on the Problem

Identify the specific goals for modelling:

- What problems should models solve?
- Which decisions will they inform?
- Select model types relevant to your goals.

Limit the Model Scope

Determine what level of detail is sufficient for addressing the problem.

- When you have modelled enough, stop.
- Balance the time frame of interest, analytical rigour applied, and complexity of the problem.

Build Your Models Using Authoritative Sources

Ensure models are built using authoritative data and trace the source of the data you use to create the models.

Leverage Existing Models Wherever Possible

Before you build a new model, assess whether there are models available that you can use or adapt for your purposes.

In a federated environment, keep your model to your scope.

Document Your Models

The ability to reuse models depends on the ability to clearly understand what is being modelled and how.

Configuration Management for Models

Determine and apply configuration control to models, including meta-data on the purpose of the models and how they have been used.

Plan your strategy for sustaining artefacts over time.

Make Your Models Available to Other Organisations for Their Use

This includes sharing documentation and configuration management information.

Address Data Rights and Life Cycle Accessibility

Plan for data rights, storage, and accessibility from the outset, ensuring models are usable, upgradable, and affordable.

Plan, Manage, and Tailor the Methodology

Develop a modelling strategy tailored to your organisational needs.

Define clear objectives, required model types, and expected uses.

Reuse Existing Methodologies

If performing a similar modelling effort to an existing one, reuse existing methodologies to maximise alignment in approach.

15 Conclusion

This guidebook provides foundational knowledge on models and their value to Defence. It offers practical guidance for Defence personnel, partners, and industry on leveraging models to accelerate capability development, enhance decision-making, and foster collaboration across the One Defence Capability System (ODCS).

Throughout the guidebook, the role of models as authoritative is consistently emphasised. Models—whether formal or informal, physical or abstract, descriptive or analytical—form the backbone of contemporary engineering practice. By integrating authoritative data and sources of truth, including information stored in external platforms such as 1DD, Defence ensures that its models are both robust and reliable. This integration underpins structured engineering data, enables advanced analytics, and establishes digital threads that connect strategy, design, acquisition, and sustainment.

Using and adopting open standards and interoperable methodologies, Defence guarantees that models can be reused, federated, and exchanged across organisational boundaries, supporting collaboration within Defence, with industry, and with international partners.

A central lesson is the necessity for purposeful modelling: models must be fit for purpose, scoped to address specific challenges, and constructed using authoritative data sources. The guidebook cautions against unnecessary complexity and advocates for incremental approaches (“crawl, walk, run”) to build modelling maturity and workforce capability. Configuration management and governance are critical to maintaining model integrity, particularly as models become central to decision-making and contractual relationships.

Verification and validation processes are essential to ensure models are both accurate and fit for use. Teams must invest in rigorous modelling methodologies, peer review, and continuous improvement to maintain confidence in model outputs. The guidebook also highlights the evolving role of artificial intelligence, digital twins, and advanced simulation, noting their transformative potential and the importance of careful oversight and governance.

The guidebook calls for Defence to continue advancing its modelling practices. It encourages the Defence community to share models, methodologies, and lessons learned, fostering a culture of collaboration and continuous improvement.

The guidebook will evolve over time as Defence’s use of models matures and insights from across Groups and Services are incorporated. Once established, its primary format will be a dynamic website or wiki, which can be updated and expanded as required and will be a critical part of the Digital Engineering Knowledge Hub (DE KHub).

In summary, the Digital Engineering Modelling Guidebook serves as both a reference and a call to action. By embracing its principles and integrating authoritative sources of truth, Defence can realise the full benefits of digital engineering: accelerated capability delivery, enhanced decision-making, and a resilient, collaborative enterprise equipped to meet future challenges.

Point of Contact

The point of contact for this guidebook is the CASG Engineering, Technology & Material Logistics Division (email: digital.engineering@defence.gov.au).

Annex A Acronyms, Abbreviations

Term	Definition
1DD	One Defence Data
AI	Artificial Intelligence
ALM	Asset Lifecycle Management
API	Application Programming Interface
ASDEFCON	Australian Standard for Defence Contracting
BIM	Building Information Modelling
BPMN	Business Process Model and Notation
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CC1	Creative Commons 1.0 (license)
CC3	Creative Commons 3.0 (license)
CDD	Capability Definition Documentation
CONOPS	Concept of Operations
DE	Digital Engineering
DEKHUB	Digital Engineering Knowledge Hub
DIS	Distributed Interactive Simulation
DM2	DoDAF Meta-Model
DoDAF	Department of Defense Architecture Framework
DSE	Defence Secret Environment
ELM	Engineering Lifecycle Management
ERP	Enterprise Resource Planning
FDAC	Force Design & Assurance Cycle
FIC	Fundamental Inputs to Capability
FMS	Foreign Military Sales
FPS	Function and Performance Specification
GRA	Government Reference Architecture
HLA	High Level Architecture
I&DC	Investment and Delivery Cycle
ICA	Integrated Capability Assessment
ICD	Integrated Capability Directive
IEEE	Institute of Electrical and Electronics Engineers
IFA	Integrated Force Assurance
IFC	Industry Foundation Classes

Term	Definition
ITAR	International Traffic in Arms Regulation
IVVQ	Integration, Validation, Verification, Qualification
JMD	Joint Mission Design
M&S	Modelling and Simulation
MARTE	Modeling and Analysis of Real-Time and Embedded Systems
MBSE	Model-Based Systems Engineering
MoDAF	Ministry of Defence Architecture Framework
MVC	Minimum Viable Capability
NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organization
NIST	National Institute of Standards and Technology
OCD	Operational Concept Document
ODCS	One Defence Capability System
OMG	Object Management Group
OOSEM	Object-Oriented Systems Engineering Method
OPM	Object-Process Methodology
PDU	Protocol Data Unit
PMI	Product and Manufacturing Information
RAM	Reliability, Availability, and Maintainability
SCM	Source Code Management
SCPC	Strategy, Concepts and Planning Cycle
SEBoK	Systems Engineering Body of Knowledge
SOI	System of Interest
STEP	Standard for Exchange of Product Model Data
SWaP	Space, Weight, and Power
SysML	Systems Modeling Language
T&E	Test and Evaluation
TOGAF	The Open Group Architecture Framework
UAF	Unified Architecture Framework
UML	Unified Modeling Language
V&V	Verification and Validation
XAI	Explainable Artificial Intelligence

Annex B References

B1. Defence Digital Engineering Strategy 2024
B2. Digital Engineering Roadmap
B3. Digital Engineering Framework
B4. One Defence Capability System (ODCS) Manual
B5. Joint Mission Design Guide
B6. Digital Engineering 101 – Getting Started
B7. 2024 National Defence Strategy
B8. Shared Modeling Framework
B9. Shared Modeling Framework for Mission Engineering
B10. Concept Aspire
B11. Enterprise Business Management System
B12. Mission Engineering Guide – USDR&E
B13. CASG-2-Manual (E&T) 12-3-003 – Capability Definition Documentation Guide
B14. S. Fowler and E. Sitnikova, 2019, Toward a framework for assessing the cyber-worthiness of complex mission critical systems, IEEE
B15. Model Based Acquisition User Community
B16. Defence Test and Evaluation Manual
B17. CASG-2-Procedure (E&T) 12-3-001 – Operational Concept Document
B18. ASDEFCON templates
B19. ASDEFCON Strategic Materiel Draft Statement of Work
B20. DAF DTO Digital Considerations for Acquisition
B21. DAF DTO Model-based Strategic Contract Guidance
B22. DAF DTO Integrated Development Environment
B23. DAF DTO Acquisition and Sustainment Data Package
B24. US DAF Digital Building Code
B25. Defence Reliability, Availability and Maintainability Manual (RAMMAN) 3.0
B26. Modelling and Simulation (M&S) Policy
B27. MITRE ATLAS™
B28. Inquiry into the Defence Annual Report 2022-23, Chapter 5, Joint Standing Committee on Foreign Affairs, Defence and Trade
B29. Enterprise Resource Planning (ERP)
B30. What is a Model (SEBoK)
B31. Why Model? (SEBoK)
B32. Model Classification (SEBoK)
B33. A Common Basis for Digital Engineering Discussions
B34. Data Capability Portal
B35. Digital Engineering Tools Assessment
B36. Reference Data Management
B37. Master Data Management
B38. Box, G. E., & Draper, N. R. (1987). *Empirical model-building and response surfaces*.
B39. One Defence Data (1DD)
B40. Object Oriented Systems Engineering Method (OOSEM)
B41. MagicGrid Book of Knowledge
B42. INCOSE Systems Engineering Tools Database Categories

B43. UML
B44. SysML
B45. ISO 15288:2023 Systems and Software Engineering – System life cycle processes
B46. Dori, D. (2002). *Object-Process Methodology: A Holistic Systems Paradigm*. Springer
B47. DoDAF
B48. Fit-for-Purpose Visualisation of Architecture to Support Defence Capability Decision-Making
B49. NAF
B50. UAF
B51. Modeling Capabilities and CONOPS using UAF
B52. ArchiMate
B53. TOGAF
B54. Distributed Interactive Simulation (IEEE Std 1278.1 and IEEE Std 1278.2)
B55. High Level Architecture (IEEE 1516)
B56. Modelling and Simulation Government Reference Architecture for Synthetic Training
B57. Business Process Model and Notation (BPMN)
B58. Industry Foundation Classes (IFC)
B59. OpenBIM
B60. STEP (ISO 10303)
B61. OSLC
B62. CASG-2-MANUAL (E&T) 12-0-001 Materiel Engineering and Maintenance
B63. Agile SE (SEBoK)
B64. DAF DTO MBSE Guidebooks & Style Guides
B65. CASG-2-MANUAL (E&T) Function and Performance Specification Development Guide
B66. Guide to the Systems Engineering Body of Knowledge (SEBoK)
B67. Model-Based Systems Engineering (SEBoK)
B68. Representing Systems with Models (SEBoK)
B69. CATIA Magic Locking a model for editing
B70. IBM Engineering Lifecycle Management (ELM)
B71. Git
B72. Power BI
B73. IBM DOORS Suspect links
B74. Modular Open Systems Approach (MOSA)
B75. Digital Engineering Data and Data Sharing Guiding Principles
B76. Defence Data Modelling Framework
B77. Model Transformation by Example
B78. Validation In CATIA Magic Tools
B79. VITECH GENESYS Design Integrity Checker
B80. Digital Engineering Model Maturity
B81. Measuring MBSE Model Maturity – A Library Supported Approach
B82. US DoD USD(R&E) Five Foundational Principles for DEM&S Adoption
B83. DAF DTO Best Practices and Lessons Learned
B84. US DoD USD(R&E) Mission Engineering Principles