Model based systems engineering: a methodology for collaborative requirements engineering

Chanaka Aluwihare[1], Michael Waite[2] and Christopher French[3]

[1]Defence Materiel Organisation, Land System Division, Land Engineering Agency

[2&3]Aerospace Concepts Pty Ltd

Abstract: Incomplete and incorrect requirements lead to a high probability of sub-optimal system solutions being implemented, which may not satisfy *End User* needs and expectations. In order to address requirements completeness, systems engineering provides a framework to support the elicitation of *End User* needs. Model Based Systems Engineering (MBSE), a subset of systems engineering, enables *End User* needs to be captured and linked to the originating operational policy, doctrine and guidance along with the analysis that led to their elicitation, thereby providing rationale traceability. MBSE enables effective capability system design, integration and sustainment through an integrated life cycle view of the entire capability. An effective Integrated Project Team (IPT), with the requisite knowledge and skills, is needed to provide authoritative information for capture via MBSE enabling effective implementation of this approach. When the information obtained via MBSE is baselined, the resulting information model is effectively a *'Single Source of Truth'* for the system, linking capability requirements back through solution independent functionality, the structure, and *End User* needs through analysis to initial high level guidance. This article discusses how an IPT can generate various stakeholder requirements for a sought *End User* need and how to verify the requirements rationale and completeness using MBSE.

Keywords: capability requirements, model based systems engineering, systems engineering.

1 Introduction

The foundation of any Defence engineering activity that seeks to provide solutions to the Australian Defence Force (i.e. *End User*) needs is the existence of valid, traceable, complete and endorsed requirements. Requirements are statements that identify an operational, support, functional, performance, interface, constraint or process for which an engineered design solution is sought. Requirements need to be validated by the *End User* as part of acceptance in order to determine the level of project success. The importance of unambiguous requirements and their influence on risks to projects that acquire materiel for the *End User* is evident from reviews [1], [2], & [3] into Defence procurement activities. Requirements definition in rapidly changing threat environments and areas with swift advances in technology is challenging, especially when noting the complexity of integrated man-made systems for Defence.

In order to manage the complexity within requirements definition, systems engineering provides a framework for comprehensively defining the problem. The framework enables requirements definition to be resolved without oversimplification. Systems engineering involves considering a problem in its entirety, ensuring initial definition of *End User* needs and the required functionality of a solution system by documenting requirements, and then proceeding with system design and validation [4]. The systems engineering standards [5] & [6] define “what to do” with respect to the processes for developing systems, whereas the capability maturity models [7] & [8] and the associated assessment method [9] provide a means to assess the “effectiveness” of the processes in place, which enable system development. Integrated product and process development [8] is a systematic method and management technique that integrates the system development activities through the use of multidisciplinary teams [10], the core of which are Integrated Project Teams (IPTs) with the empowerment to plan and complete set tasks. Most importantly, systems engineering considers the business, operational and technical needs of all stakeholders with the goal of providing an engineered solution that satisfies stakeholder expectations and *End User* needs.

Organisations help realise systems that satisfy given problems by utilising systems engineering processes that align with their goals and objectives. The outputs of these established organisational systems engineering processes consist of the documents that define the system requirements and design solution. Some example documents generated in this 'document-centric' approach are concepts of operations, system architectures, system requirements, performance measures, interfaces descriptions, decision registers and configuration baselines.

Instead of the document focus with textual requirement statements, it is possible for each stakeholder and their role to be combined in an information model. The information model uses specific scenarios, the operating environments, expected performance, effectiveness, constraints and quality factors in which their envisioned system will operate. This Model Based Systems Engineering (MBSE) is the formalised application of modelling to support system requirements, design, analysis, verification and validation activities. It begins in the conceptual design phase and continues throughout development and later life cycle phases [11]. In essence, MBSE is the model-centric implementation of a suite of fully integrated processes [12] for determining system requirements, which are linked and traceable.

Within Defence industry, it is widely accepted that incomplete and incorrect requirements lead to designed systems that do not solve the expected *End User* problems because these problems were never correctly stated or identified during the design and validation process. When this occurs, attempts are often made to change the operational context taking into account the system limitations. When this is not possible retrofitting (changing the system) by the *End User* can be conducted to the extent feasible at great expense. The other possibility is the absence of requirements that are clearly traceable to *End User* operational needs and the 'visualised' existence of off-the-shelf solutions that satisfy another user's requirements, which are incorrectly assumed to be identical or similar to those of the *End User*. In these instances, the basic error is the assumption that a solution exists as result of "political engineering" rather than a quantitative comparison of established solutions against properly defined *End User* requirements prior to system acceptance.

Acquisitions of high value Defence systems that are used by the *End User* typically involve huge costs and are in use for several decades. Therefore, it is imperative that all stakeholder expectations and *End User* needs are fully understood and their requirements defined. In order to develop the *End User* capability needs and the acquisition of Defence materiel, there are guides [13], [14] & [15] which use systems engineering as the basis for requirements development. These guides detail the need for coordinated inclusions and contributions from the *End User* and subject matter experts from across relevant engineering and other disciplines to work in an IPT structure to help ensure information completeness. MBSE links all of this information in a model, providing traceability and helping to ensure consistency and completeness.

In summary, the purpose of this paper is to detail how an IPT using MBSE can define capability requirements and verify that these requirements satisfy both *End User* needs and Defence objectives thereby ensuring that the acquired systems perform as expected.

2 Model Based Systems Engineering

When investigating and designing complex systems a systematic approach is needed to define and solve the problems encountered. As problems become increasingly more complex, the methodology employed to solve them must also exhibit a level of sophistication and flexibility in order that these problems can be resolved. The traditional document-based approach employed by systems engineering can only provide limited benefit because the information contained in these documents require careful examination by skilled personnel on a case-by-case basis to ensure traceability. In contrast, using a computer based model that is correctly encoded, allows for a more tailored, robust approach. This model-based approach capitalises on the skill of the engineers involved in the process, and enables them to incorporate the complex information into a single repository and to link the information in a logical manner. Simplistically, MBSE is the computer implementation of a process that allows for trained and experienced systems engineers to establish, analyse and solve systems design problems [12] using the best possible structured support. The model and its framework enables complex problems to be defined, the architectures to be established, and maintains the flexibility to address the unique sets of issues that complex systems experience. Complex systems typically contain too many points of consideration that exist within the system boundaries and therefore without a computer based model it is difficult for an individual or team of systems engineers to analyse each issue for resolution. In order to overcome these difficulties, it is possible to utilise a model-based approach which captures all relevant information within the model along with their relationships and interactions between each element. Thus, the model is the 'single source of truth' or the 'knowledge repository' containing the system life cycle information.

The innate traceability that exists when using a model-based approach shows the relationship between multiple connected elements within the model. For example, the traceability can demonstrate how changing doctrine or legislation can have a direct impact on requirements within the system. Conversely, when a system does not satisfy a defined requirement, its effects on the sought operational capability can be ascertained and decisions can be taken to modify the operational context. The model's architectural framework provides the rules and structure for capability definition, system definition, requirements elicitation, system verification and validation. One of the success factors of MBSE is a framework that is agile and flexible which facilitates the design process as opposed to influencing the design. Therefore, when using MBSE techniques it is necessary that its application facilitates the design process noting that poorly executed MBSE will provide no benefit over what would be achieved without using MBSE.

The MBSE approach brings to the forefront the hard and difficult questions which can then be addressed early in the system life cycle. Early in the life cycle when potential issues are identified, as shown in Figure 1, it takes less time and costs less to rectify the identified issues. Using a model-based approach helps to support this early problem definition and identification of potential issues [16].

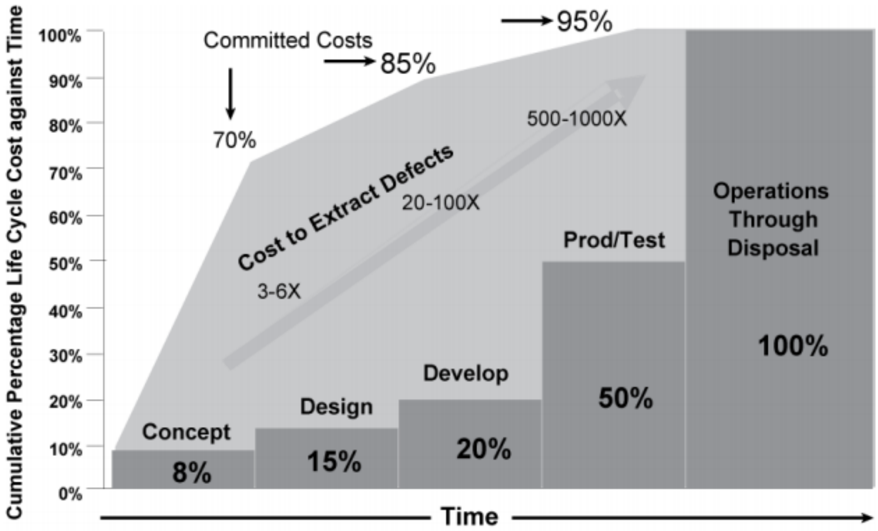

Figure 1: Committed Life-cycle Cost against Time [4]

From Figure 1, it can be seen that identification and rectification of defects early in the concept, design and development phases results in significant savings in terms of time and expense. This encourages more thought and defect identification through a robust, traceability concept design, which the MBSE approach facilitates.

Similarly, the benefits of a well implemented systems engineering effort can be seen from Figure 2, where it is noted that when the early systems engineering effort is about 15% of the total project cost, the project has a high probability of being within budget and schedule. This evidence further emphasises the need for MBSE as a basis for project success.

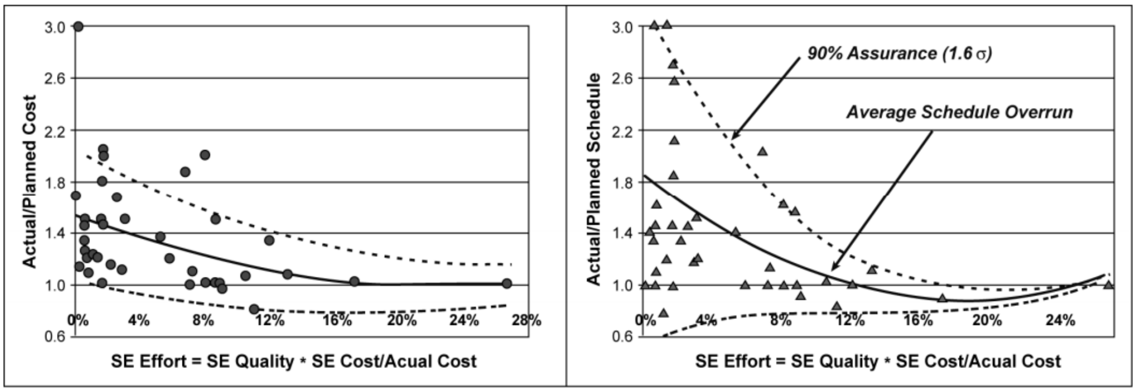

Figure 2: Cost and schedule overruns correlated with Systems Engineering effort [4]

2.1 Whole-of-System Analytical Framework

When looking to use MBSE techniques in different industries and operational contexts, care must be taken to ensure that the knowledge model is structured in a way that it reflects best practice. Therefore, when applying MBSE to Australian Defence and more specifically to capability definition, it is essential that the guidance given in the Defence Capability Definition Document Guide [13] and practices stated in the Defence Capability Development Handbook (DCDH) [14] are followed. The DCDH provides doctrine and guidance on how capability is established in the Australian Defence Force. Using a defined analytical framework, the *End User* can be assured that the operational needs have been translated into system requirements.

In order to apply MBSE techniques to the standards and practices within the Australian Defence context [16], the Defence Science and Technology Organisation developed the Whole-of-System Analytical Framework (WSAF). The WSAF has been developed using the commercially available MBSE tool VitechCORE® and it is an amalgamation of the following components:

- *Processes and Reference Model* – The structure and approaches used to shape the capability definition process within the Australian Defence capability development context.
- Experienced Staff – Trained experienced staff, skilled and capable of applying the WSAF to the capability development process.
- *Subject Matter Experts* – Utilising appropriate subject matter experts to gather information.
- *Guidance* – Defence doctrine, guidance and standards which are to be applied to a project.
- *Scripting* – Scripts automate the production of various necessary views required by DCDH.

The architectural methodology employed by the WSAF can be decomposed into five steps namely; *Scope, Bound, Carve, Match*, and *Compare*. Using these steps, the bounded and carved system can be assessed by the appropriate subject matter experts. Combined with the inherent traceability within WSAF, the systems engineer can clearly demonstrate the expected system behaviour. When executing the traceability functionality that is integral to WSAF, the overall system design robustness can be ascertained. Capturing all relevant stakeholder viewpoints and storing them in the 'knowledge repository' with traceability enables filtering of stakeholder view points and information. To ensure completeness of the 'knowledge repository' it is necessary that a wide range of stakeholders are engaged and their needs captured thereby creating an accurate system model.

3 Requirements Design

An extension of capability development is the design of requirements that adequately describe a system which meets the *End User's* needs. Incomplete and incorrect requirements can lead to designed systems that do not meet the needs of the *End User.* It is therefore necessary that a proper elicitation method is used for obtaining these requirements. The reference model used by WSAF for requirements design is given in Figure 3 which shows at a high-level the linkage between *End User's* needs (described here as "Operational Needs") to the developed system requirements.

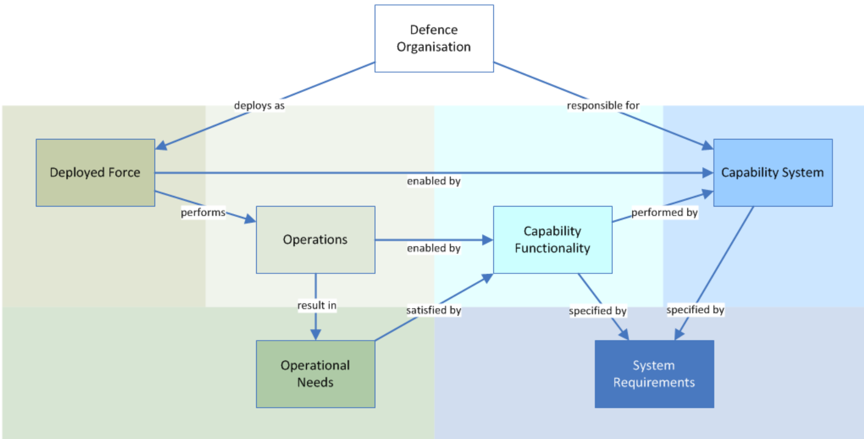

Figure 3: WSAF Requirements Design Reference Model Overview

From Figure 3, it can be seen that requirements are specified by the 'Capability Functionality' and the 'Capability System', which results in derived 'System Requirements'. Through an iterative process requirements are expressed by a function of the system and also a physical component within the system which performs that function. This iterative requirements design process is conducted until all the functions and the physical elements performing the function are identified. For *End User* capability development Figure 4 demonstrates the methodology used for capability design. This figure depicts the iteration of operational requirements and concept solution requirements until there is an acceptable concept capability design. The iterative *End User* capability design method shown in Figure 4 is facilitated by the MBSE structure and the WSAF approach.

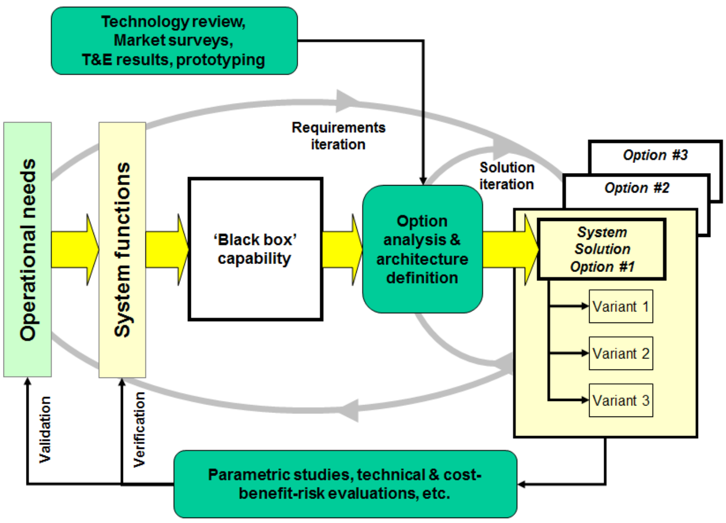

Figure 4: End User Capability Design Cycle

The capability design cycle depicted in Figure 4 is one of constant iteration, improvement, verification and validation. Throughout requirements design, the capability design cycle is used to identify gaps and to ensure that *End User* operational needs are satisfied by the system functionality. The available solution options can then be assessed against the overall system ability to perform tasks, rather than against individual features that one may believe is most crucial.

3.1 Requirements Documents

One core tenet of system development and design is the engagement of core stakeholders and subject matter experts throughout the design life cycle to gain pertinent information which is structured, traceable and consistent. Therefore, it is in a project's best interest to work closely with the *End Users* who would ultimately be using the system. Engaging *End Users* in a formal workshop setting is the preferred method for determining operational needs and associated functionality expected of a system. Adherence to this methodical process increases the likelihood of obtaining correct requirements and more importantly requirements endorsement. WSAF, which by design follows the mandated DCDH process, will produce the required requirements documents namely; the Operational Concept Document (OCD) and the Function and Performance Specifications from the model.

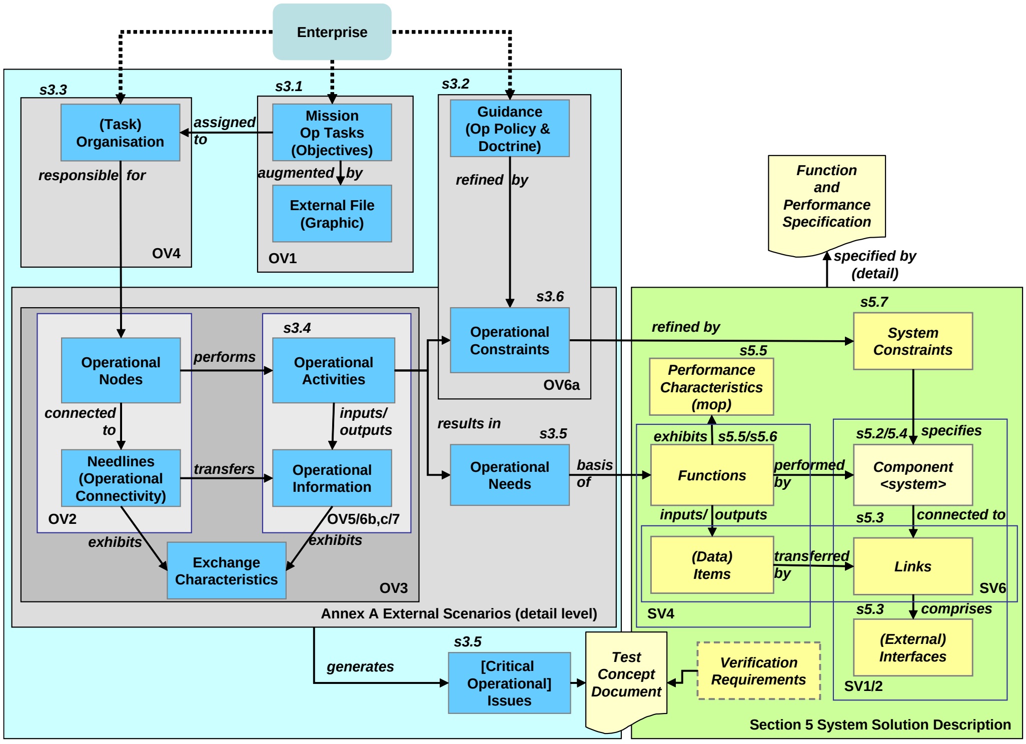

Figure 5: Relationship between OCD Section and DODAF Artefacts

For instance, Figure 5 shows the relationship between the various OCD sections [17] and the mandated Defence Architecture Framework [18] views. When building the model the various architecture views and OCD sections are formed, not explicitly, but as an inherent by-product of good MBSE practice incorporated within WSAF. Therefore, by following the framework in the WSAF, a project will organically develop the architecture views and OCD sections when identifying the problem and designing the solution. This allows the engineer to focus on problem solving with confidence noting that the resultant requirements documents are viable. In addition, the WSAF contains a number of reports which can be generated via executing in-built scripts that enables confirmation of information completeness and traceability.

4 Integrated Project Team

In order to ensure that the requirements are complete it is necessary that all stakeholder issues and expectations are understood, and this can be achieved via an IPT (or IPTs). An IPT is a temporary construct of cross-functional groups of individuals brought together under a 'Charter' (or contract) for the specific purpose of delivering a project for the *End User*. IPT membership is largely dependent on the project size, phase of development, security classification, technical challenge, and novelty of the design. Dependant on the project, it is apparent that a greater number of functional disciplines will be needed on large or technically challenging projects when compared with smaller less technically challenging projects. Notably, IPTs are the *crossroads* where budgetary and technical management intersect and is the mechanism for addressing various stakeholder concerns and issues.

To ensure that all requirements are captured, the IPT structure should include skilled membership from all relevant functional elements of a project. The IPT must work as a unit, sharing information willingly, balancing conflicting priorities and ideologies thereby defining requirements needed for successful project delivery. An IPT could be formed with both Defence and Industry personnel and would include personnel such as; a qualified program manager, *End User* representatives, engineers, logisticians, technicians, scientists, project management, finance and contracting staff. Importantly, the IPT members must possess the education, experience, knowledge, skills, and abilities necessary to perform their assigned role. The role of systems engineers in the IPT is to facilitate the elicitation of requirements from all stakeholders whilst being aware of the technologies and the commensurate disciplines required which is relevant to the project.

For technically challenging projects involving multiple stakeholders and tight budgetary and schedule constraints there is a reduced probability of the project succeeding without an IPT that is competent and committed.

5 Conclusions

The success (or failure) of Defence projects which seek to provide solutions to *End User* needs is dependent on unambiguous requirements that are complete and traceable. Systems engineering provides a framework for requirements definition, functional analysis, synthesis, verification and validation, which enables the most appropriate, balanced design to the problem environment. MBSE supports traditional systems engineering with an explicit underlying information model and adds visual modelling best practices. The MBSE process when correctly used enables verification through the traceability of requirements to operational policy, doctrine and guidance thereby assuring requirements completeness. Most importantly, for ensuring that requirements are complete, it is necessary that all relevant stakeholders are identified and each stakeholder's needs elicited. The most effective mechanism to ensure that appropriate information is gathered using this approach is an IPT, a construct with the proper membership of relevant disciplines working within a Charter that members abide. An IPT is the enabler that fosters the partnership for successful requirements definition. The systems engineering process incorporated via MBSE, which has been described in this paper, provides a basis for comprehensive requirements definition which contributes to the success of *End User* capability definition.

Acknowledgements

The authors wish to thank Mr Shane Aitken, DGLEA for his support and to Mr Shaun Wilson, CEO Aerospace Concepts Pty Ltd for his encouragement, commitment and active involvement which ensured completion of this paper.

References

1. Malcolm Kinnaird, Report of the Defence Procurement Review, 2003
2. David Mortimer, Report of the Defence Procurement and Sustainment Review, 2008
3. The Senate, Foreign Affairs, Defence and Trade References Committee, Procurement Procedures for Defence Capital Projects Report, August 2012
4. INCOSE, Systems Engineering Handbook, June 2006
5. EIA 632: Processes for Engineering a System, 1998
6. ISO/IEC 15288:2008: Systems and Software Engineering - System Life Cycle Processes
7. CMMI-SE/SW for Systems Engineering and Software Engineering, 2001
8. CMMI-SE/SW/IPPD for Systems Engineering, Software Engineering, and Integrated Product and Process Development, 2001
9. Standard CMMI^{SM} Assessment Method for Process Improvement (SCAMPI), 2001
10. U.S. Department of Energy, Integrated Project Teams Guide, DOE G 413.3-18, 2008
11. INCOSE, Systems Engineering Vision 2020, INCOSE-TP-2004-004-02, 2007
12. David Long and Zane Scott, A Primer for Model Based Systems Engineering, October 2011
13. Defence Capability Definition Document Guide, 2011
14. Defence Capability Development Handbook, 2012
15. Defence Procurement Policy Manual, 2013
16. Kevin Robinson, Demonstrating Model-Based Systems Engineering for Specifying Complex Capability, 2010
17. Australian Defence, Data Item Description, DID-ENG-DEF-OCD, V2.0
18. U.S. Department of Defense, The Department of Defense Architecture Framework (DoDAF), V2.02