ISBN for SETE2018: 978-1-925627-15-2

Supporting capability integration and understanding of Programs and Projects through integrated information visualisation

Christopher French
Shoal Engineering Pty Ltd
chris.french@shoalgroup.com

Steven Heard
Clearz Defence
steven.heard@clearz.com.au

ABSTRACT

In 2016 Defence, as a result of the First Principles Review, introduced the Capability Life Cycle (CLC) and Program Management construct to develop and manage Defence Capabilities for the Joint Force of the Future. The introduction of Programs is a step towards managing the integration and interoperability aspects at a Capability level in addition to the Project / Product level. As the operationally required integration and interoperability demands between multiple capability elements increases, so to does the complexity. Issues impacting one element are felt across the entire operational space. With Defence's introduction of the Program construct, there is a new avenue to drive the integration and interoperability aspects towards success across the multiple Products, Projects and Programs.

The Program brings together strategic, management and design activities that move and evolve as the needed capabilities change, projects are realised, and products go through their operating life into disposal. Dealing with the interaction that could occur, and does occur amongst these elements becomes a challenge of managing, understanding and communicating complexity. The various types of information, from disparate sources, need to be generally aligned with one another, with a sense of codification and an accepted common language.

This paper captures the methodologies, approaches and realised systems used to inform decision makers and Program owners of complex information and complex information changes. By developing architectures and facilitating codification of information, the information was aligned and brought together. Through modern visualisation techniques, the richness of information is presented to stakeholders seeking to inform decisions, help those involved with the Programs focus their effort and understand the ramifications of changes from or on others.

INTRODUCTION

In early 2016, Australian Defence changed the way there were going to manage projects and deliver capability in the evolving, increasingly integrated Operational environment. These changes, described within the Interim Capability Life Cycle documentation (Defence, 2016), brought the focus on delivering effective capability outcomes using fit-for-purpose approaches for the specific projects of interest. One of the mechanisms introduced was that of the Program; an intermediate capability layer between Portfolio and Projects. The Program is defined as "a group of related existing Products, projects and activities in a coordinated way to optimise the capability outcome within allocated resources: (Griggs, 2017), where it provides an enduring mechanism to describe the capabilities needed and facilitate their delivery by aligned projects.

The concept of the Program has been applied throughout other large-scale organisations (Gorod et al, 2008). While the concept is not new, there are always challenges that present itself when seeking to understand the System-of-Systems behaviours in each unique environment. Defence seeks to build the Program construct within the existing Portfolios and Projects, with minimal disruption to business-as-usual. This involves a bottom-up approach to Program development and integration, with immediate value from the changes being sought. The value of adding the Programs comes from increasing the ability of Projects and Products to integrate and interoperate

PROGRAM CONTEXT

The 'Program' concept

The Australian Program encapsulates the individual projects, as well as supporting elements of aligned business and operational aspects. This includes legacy systems currently in place, upcoming systems to be conceived and introduced into service, business activities and specific events. The Program is enduring through time, whereas elements within it may enter and exit, and have various levels of significance over time. Within Australian Defence a Project is defined as "a unique, finite, multidisciplinary and organised endeavour to realise discrete changes to the capability managed by a Program", and a Product defined as "the whole of life of an asset that contributes to capability such as a facility, major platform, major information and communications technology application or fleet of equipment" (Defence, 2017). A representation of the Program can be seen in Figure 1 below.

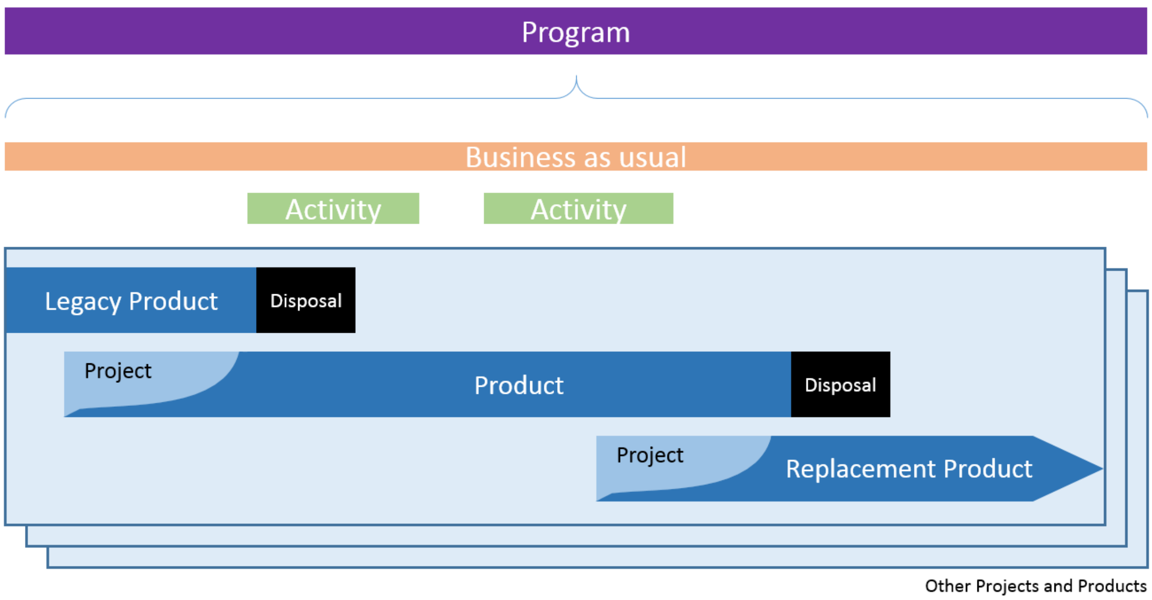

Figure 1. The elements which constitute the Program concept

The Program concept as an application of Systems of Systems Engineering, is not new. Bullus and O'Shea (2011) described an approach for Australian Defence to understand how capability evolves over time, and a means for ensuring and measuring said capability. The introduction of a Programs-based approach to delivering Australian capability outcomes, is an acknowledgement of the importance of System-of-System concepts to that delivery.

Enabling work

With Projects and Products already in place, the introduction of Programs had to be developed using a bottom-up approach. The definition of the Program emphases the nature of the capability being provided by the Program, Projects and Products over the management activities needed to ensure the successful delivery. The development approach to defining the interactions of the Programs with each other, and the Projects within the Programs comes from past research. Pratt et al (2015) acknowledge the uniqueness of the Australian Defence environment based on the organisational, strategic, political and cultural environments. The lack of directed or acknowledged System-of-systems (OSD, 2008) have not been considered to exist previously within the Australian Defence environment (Pratt 2015). The introduction of the Program construct suggests movement in this area. Furthermore, Cook and Unewisse (2017) have developed an approach, and associated recommendations for the implementation of SoS methodologies within the Australian Defence context. This approach seeks to implement widely regarded SoS concepts, such as the Simplified Wave Model Description (Dahmann et al. 2011). To understand the approach needed to define an initial Program model, and to enable some of the above captured principles, a Capability Design approach was used.

Program Integrating Operational Concept (PIOC)

The Capability Design approach, as initially described within the Defence Capability Development Handbook (Defence 2014), was applied to conceptualise and specify a model-based Program definition capability. This allows for the Program to be considered and structured against the intended purpose. Initial effort was to facilitate the capture and presentation of the Program Integrating Operational Concept (PIOC) (Defence 2017). The conceptual design effort sought to define the operational concept of the 'Program development capability' and the function and performance requirements and performance characteristics for the Program-level modelling capability system. The information would then be communicated through the PIOC, as required by Australian Defence.

The resultant system for the PIOC is a complex information model, architectured to demonstrate defined relations between elements, organised into information classes, which define attributes of Programs - including the program of interest as well as related programs. The information model and rules for inclusion are strictly defined to ensure consistency and coherency across various Programs, and between Programs and constituent Projects and Products. This approach also enables information within one Program to be leveraged on another Program due to the established structure. The approach leveraged techniques as described by Harvey et al (2014) and Liddy et al (2012).

Use of DoDAF 2.0

To develop an information structure useful for Program definition, Program information capture and PIOC publication; knowledge of established architectures (Robinson 2010) was combined with Australian Defence capability definition methodologies and established architecture frameworks. A common framework used in Australia, is the Australian Defence Analytical Framework 2.0 (AusDAF 2.0). AusDAF uses an DoDAF and MoDAF foundation, and thus for the development of the Program model structure, DoDAF 2.0 was selected. The expectation was that modifications would be made to align the Australian Project frameworks, with that of the Program. Analysis of the key areas of influence and initial use-cases of the framework were undertaken to determine what was needed from the initial instantiation of the framework. Given that the PIOC is intended to focus on Integration and Interoperability Analysis as well as Operational Analysis, these were areas of focus for the initial development.

With integration and interoperability of the Programs, Projects and Products a key analysis area, the framework was designed to support and facilitate this using common information sources between individual instantiations. As Program teams need to communicate with other Program teams, Project teams need to communicate with other Project teams, and Program teams need to communicate with their constituent Project teams, the framework focused on enabling this and clearly defining the integration and interoperability dependencies and agreements. To achieve this a consistent context was needed to be set for all Programs. The Organisational Context would serve this purpose.

Use of the SCMILE framework

Previous work within Defence, has resulted in the introduction of the SCMILE Framework (Lowe et al 2017), a method for capturing dependency information on Programs, Projects and Products within Australian Defence. It was agreed that the SCMILE Framework would be the prescribed means of identifying, categorising and rating interactions between different Programs, Projects and Products. Engagement with the developers of the SCMILE Framework resulted in adaptation for implementation, with the logic of the framework embedded within the Program construct.

Role of Operational Analysis

Operational Analysis, in line with the Australian Capability Definition process, is required at the Program-level to ensure alignment of the Program-level capability definition and subordinate Project-level capability definition. Thus, the Program model was architected to enable capture of information in the operational domain, in a manner consistent with existing project-level capability models (as described in Robinson, 2010). This Operational Analysis enables elicitation of the Operational Needs of the Program-level capability, with the model capturing these needs and their traceability to operational activities and capability scenarios. Through robust traceability it can be understood how individual capability instantiations contribute to the Operational Scenarios, and thus derived Needs.

Specific consideration was applied to establishing known information classes within the reference architectures. These could be used as persistent elements between different levels of information, and provide a way to link individual Program and Project models. Analysis was undertaken to establish the commonly defined information elements, which could exist across multiple instantiations of the information set. Examples of elements common across Programs include Program and Project descriptions, existing mission systems, and higher-level Defence organisational structures. From a practical perspective, the approach used would enable dependency analysis to occur individually within the various constructs, and integrated together into a wider picture.

Inclusion of risk-based approach to design

The approach described within the CLC handbook, instructs future Australian Defence acquisitions to utilise a risk-based approach to design. The model was architected such that risks could be related to any element of the modelled Program definition. Capturing the dependency information in the Program model as individual elements allowed for risks, which are also captured as model elements, to be identified and associated with those dependencies at the appropriate level of fidelity. The dependencies and associated risks were then reported in the PIOC reports directly from the modelled information.

Program-level modelling framework

The overall approach was developed with an emphasis on System-of-Systems Engineering principles, with the initial implementation designed to be further developed and expanded over time. At the time of development, the role and use of the Program space within Defence was being developed and further understood, and thus evolving at a rapid rate. As such, consideration was given as to how the modelled information could be further leveraged without rework by the various Programs in the future. Consequently, the Program modelling framework was developed in close collaboration with related concurrent research being undertaken by Defence (and described in Cook, 2017).

A hihg-level view of the the Program-level modelling framework schema is shown in Figure 2.

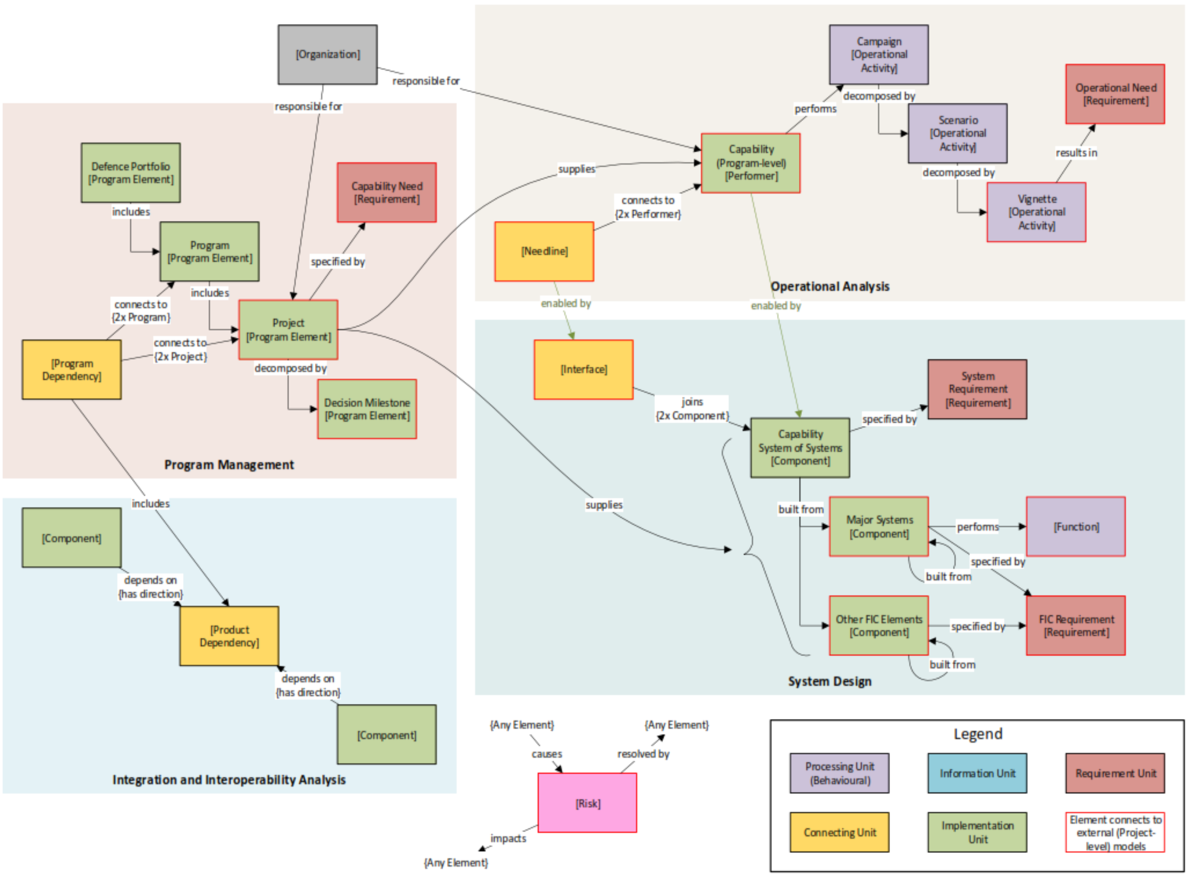

Figure 2. Program-level modelling framework schema

The established structure is focused on four distinct, but inter-related areas:

1. Program Management
2. Integration and Interoperability Analysis
3. Operational Analysis
4. System Design

As with all capability designs, the realisiation of successful capability lies in supporting and enabling the key elements. For the model-based Program definition capability, the supporting elements to be defined were:

- The Defence processes regarding how the Program will be used and enabled
- The enabling tools to facilitate and enable the Programs
- The training and information to be made available to personnel, to increase their competency when working in a System-of-Systems environment.

FOCUS ON VISUALISATION

The development of the PIOC results in a rich picture of how the Program will be used (the Operations), who is reliant on it (services provided) and who the Program is reliant on (services received). It was conceived that, once multiple Programs developed the information relating to the PIOC, within the previously established structure and approach, that information can be brought together and visualised.

To understand what is needed for the model-based Program definition capability, as a generic versatile methodology for Program understanding and development, the previously developed Capability Design required recasting to expand the scope and determine what was meant by “visualisation”. The initial capability description for a Program emphasised the design and understanding to allow for deployment of the model-based Program definition capability as an enduring service. The need was to abstract the capability description wider than the PIOC centric, into a whole-of-Program design, which can then be focused towards specific applications. This includes the initially captured definition which articulated the PIOC construct and understanding aspects, as well as the potential for expanding the capability as more Program attributed and information become known over time. To understand the visualisation needs of Program managers and related stakeholders, consultation and analysis was undertaken, and the model-based Program definition capability operational scenarios were expanded to include:

Following analysis and consultation, the operational scenarios were expanded to include:

1. Program Development – instantiating a Program as either a brown- or green-field exercise.
2. Program and operational capability management support – ensuring the success of the program by delivering the operational outcomes, including assurance and auditing activities
3. Program-level model development – the development of the information for population within a rich, robust model-based environment.
4. Program-level change management – the management of the program information within a wide, rich picture environment for consistency and correctness
5. Program-level model information output – the communication of program information through various outputs, such as documentation and visualisation techniques.

The above scenario development sought to aid discussion and focus attention on understanding what was needed. The described mission, as provided to the team and to guide the effort was “Help the Program Managers do their job”, and thus prudent consideration of what goes into a Program Manager’s day-to-day activities was a key input to the scenarios and the resultant identification of needs. The identified visualisation needs were then used to define the required “visualisation system” functions, that could then be further derived and realised by a software implementation.

CAPABILITY REALISATION

Introduction

Historically, a standard approach of limited engagement contracts, with fixed deliverables has resulted in the production of a range of software to meet specific client needs. Common methodologies seek to replicate the functionality of the non-software implementation as a means to increase efficiency and effectiveness of the business area. By systemising paper-based approaches there remains the risk of continuing the inherent problems associated with the previous approach as a direct replication is sought, and delivered. Through a capability development approach that seeks to realise the intended 'Operational effects', the system behaviour and functionality can be produced to achieve the intended results, which are fit-for-purpose, have a view of the future and can provide value-add information beyond the immediate outcome. This can, and should, be performed in the capability understanding phase, prior to any system implementation occurring.

By utilising software to support and facilitate process and governance changes an opportunity is created to develop a more effective and efficient system integrating software, data, processes, and governance. The consultation in support of key stakeholders was aimed to utilise current data structures and apply the model-based Program definition capability to support process and governance effectiveness.

Initial needs and constraints

An initial goal of the development effort, conducted in conjunction with end-user community, was to fully understand the current problems, processes and opportunities associated with system development of an integration and visualisation tool. An important aspect of this effort is understanding the gap between the current as-is process and outcomes in relation to the intended end state. The gap analysis conducted resulted in the following business areas for consideration:

1. The initial implementation of the Program Model had been developed to support stakeholder understanding of program intent and need. This model, developed through a systems engineering tool, had limited access by day to day users. Whilst it promulgated the PIOC information as required by the Program, Program Sponsors and the associated Program Sponsors were unable to utilise the richness and entirety of the information captured in the model. Thus, potential value was lost.
2. Visibility and understanding of information across the Defence acquisition portfolio was limited, creating issues with dependency identification lacking consistency and the appropriate level of fidelity. Without appropriate codification of how dependencies can and should be generated, the information between Programs and Projects would misaligned.
3. Introduction of the SCMILE framework to provide clarity on the services provided by each Project or Product into Defence capability aided the establishment of a common language and approach. The introduction of a new approach to dependency analysis required the appropriate stakeholders to adopt a new approach. Similarly, existent informational systems were not structured to utilise a multiple or different dependency frameworks.
4. There was an identified lack of consistency in processes, systems and governance frameworks used to support the new Program areas. The bottom-up approach leveraging existing information, processes and tools, all of which operated in similar yet different ways. Understanding how the various approaches aligned and varied was a necessary exercise. A strong governance framework would be needed to understand and assure such a complex, rich capability.

The initial solution looked to address these problems through accessibility and visualisation of data already held within Defence systems. By combining data held within the Defence project management tool, CDMRT2, and the Program model data held within CORE, provided users with access to data in order to inform decisions across the Defence portfolio. By the very nature of the information held within the Defence acquisition portfolio, it was an imperative that complex information, particularly around dependencies, was displayed to allow for innate understanding of the data displayed. In line with the concepts of system development over software engineering, this allowed for the utilisation of the system as a platform for future development of interaction, analysis and machine learning algorithms.

Approach

The approach to software development can vary between projects and organisations. When working within a complicated environment such as Defence, with multiple integrated information systems, it is important to have a view of the future outcomes the system needs to achieve. With this in mind the approach for the initial implementation of the system was to establish a strong foundation to support a three-phase plan. The plan sought to deliver a comprehensive, implemented system providing analysis and decision support to Defence in line with the initial capability design.

Phase 1 sought to identification of data sources/relationships and bringing together stakeholders in order to visualise existing data across disparate systems and processes. Identification of integration steps required to display this information. Building a software framework to achieve phase 1 data visualisation and support subsequent phases of data exploitation. The main effort of this phase was to implement system foundations and engage the stakeholder community.

Phase 2 build upon the phase 1 software to identify what the data connections needed to be and how they would support evidence-based decision making across the portfolio. Developing stronger project and program linkages to systems engineering models through modern system engineering tools (modern API) and connected supporting data repositories. The aim of Phase 2 was to allow the user to identify connections across the available data to provide evidence for decision making processes using the engineered models as a framework of requirement, and provide SME input to the model data. The main effort of this phase was supporting visulaisation of data linkages for decision support.

Phase 3 utilised connected data elements in conjunction with AI and machine learning to move the program/project management into a proactive system. Optimisation algorithms and natural language processing provide users with business insights and future looking data visualisation to solve likely or known issues prior to realisation. The aim of this phase was to provide proactive analysis, problem solving, threat injection and optimisation of processes and decisions.

Technology used

The Defence ICT environment poses a range of challenges to implementing software to support Defence business. This requires a considered approach when developing tools and identifying third party software to support development. When selecting software to be utilised it is important that it not only performs on older architecture but is also highly resistant to change to support Defence needs.

The system developed to support the program model was in essence a visualisation layer on top the model developed in CORE and CDMRT2 (the Defence project management software). It is important to note that the software does not hold source information. This allowed for the tools to remain the single source of truth with the developed system providing visualisations to support decision making.

The software developed was written in C# utilising ASP.NET CORE framework supported by Angular2, vis.js and Highcharts libraries. These development frameworks allowed for use of industry standard development models whilst still being applicable to the Defence architecture.

Realised System

Following engagement with stakeholders, a realised prototype system was developed. Screenshots of the realised system, using demonstration information, are seen below.

The Risk viewpont, shown Figure 3,.seeks to communicate the risk information in CDMRT2.

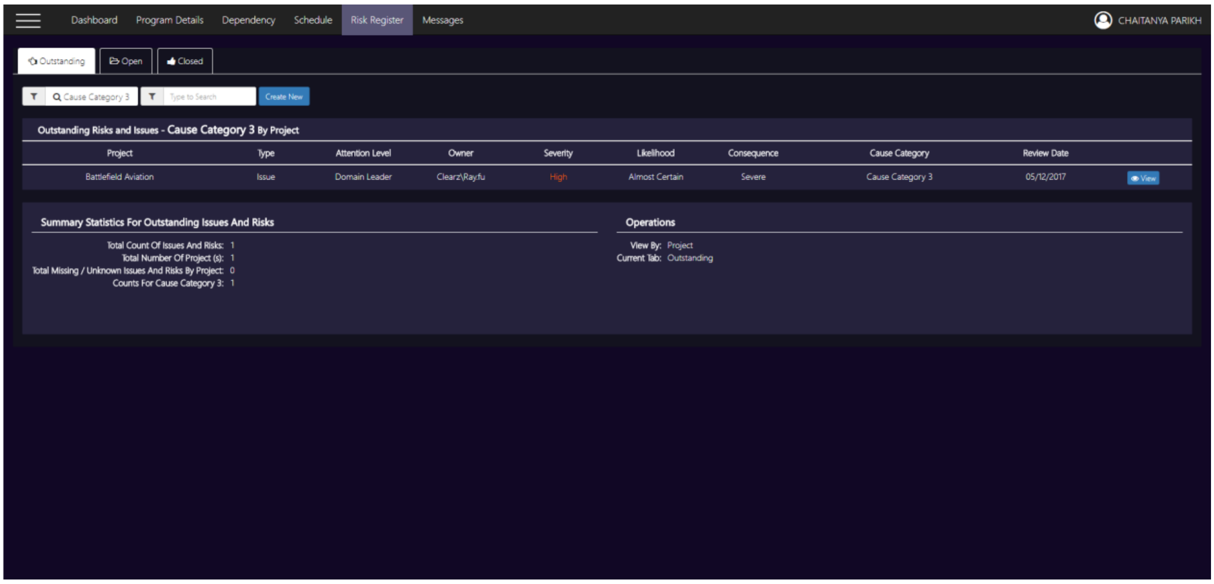

Figure 3. Screenshot of Risk viewpoint

Schedule information was presented to understand the impact from schedule changes to other, dependent Products and Projects. By moving key dates within the Project or Program timeline, a change report was generated to capture the potentially affected entities. This ensures that all relevant parties are informed of major changes and a 'whole-of-Program' approach can be taken to decision making as the necessary communication and impacts are shared. The schedule viewpoint can be seen in Figure 4.

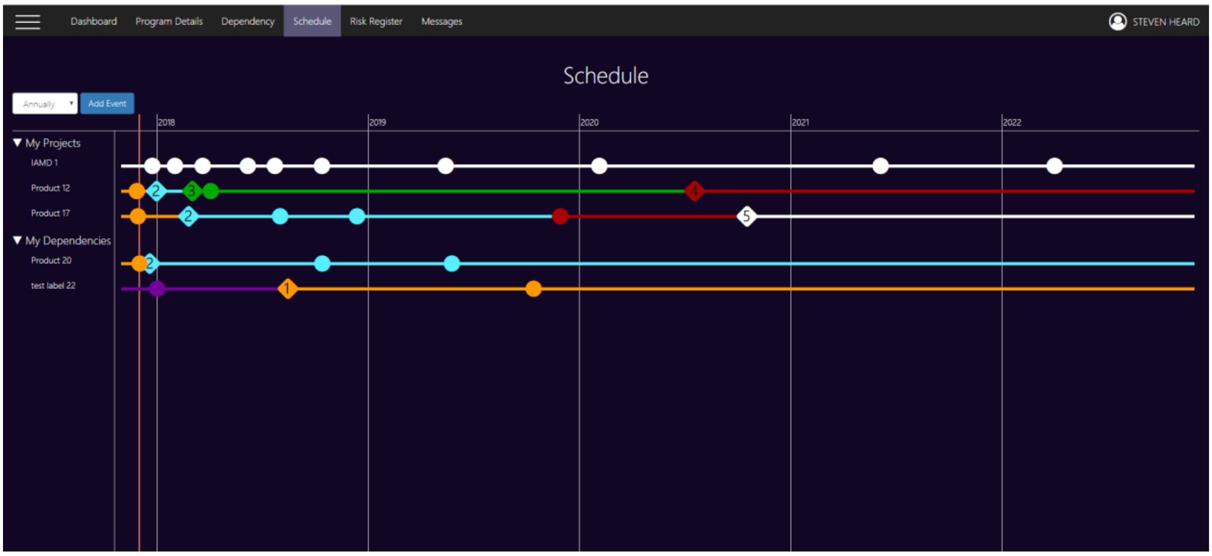

Figure 4. Screenshot of Schedule viewpoint

Dependency information was the most complex set of information intended for visualisation. The various considered Programs, Projects and Programs were 'linked' together, through the captured dependency information as defined from the SCMILE framework. This can then be filtered, grouped and added to through the tool. It included understanding secondary and tertiary dependencies on elements with natural groupings used to communicate the information easily. The methodology of information change reflects Defence process, with escalation mechanisms included. An example of the dependency viewpoints are provided in Figure 5 and Figure 6.

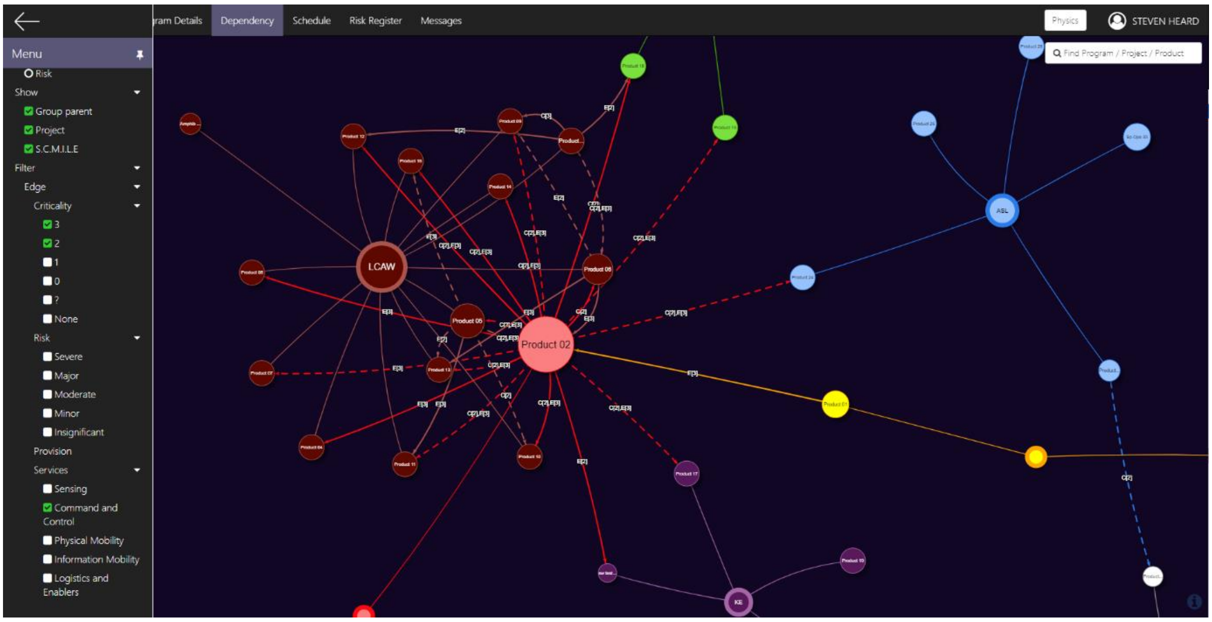

Figure 5. Screenshot of Dependency viewpoint, SCMILE framework

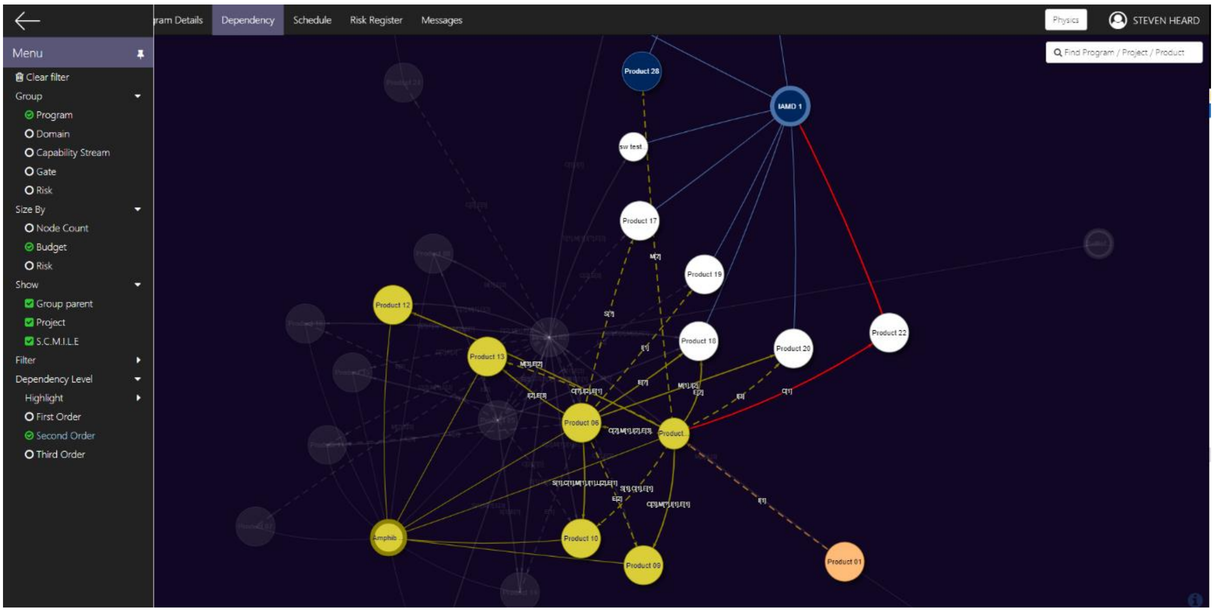

Figure 6. Screenshot of Dependency viewpoint, second order effects

A high-level, selective view of the various Program-level models, which were used to develop PIOCs were integrated within the developed system. With limited cross-communication currently, the implemented system does show the wider set of information from the Program model, which is greater than the current Project and Program management systems provide. It can be seen in Figure 7.

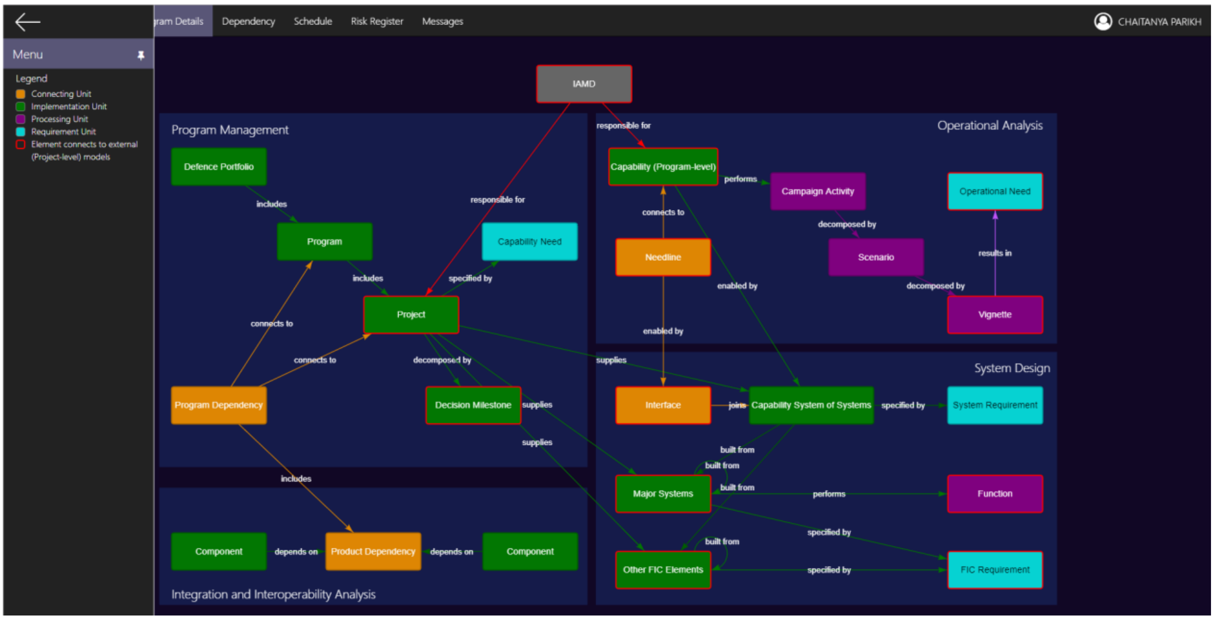

Figure 7. Screenshot of Program-model viewpoint

Initial outcomes

Following the implementation of Phase 1 above, a prototype, realised system was implemented on the Defence environment. It achieved the desired outcomes regarding stakeholder engagement and proving the viability of the approach used. Going forward, this initial implementation will now be shared with the user community to elicit immediate feedback and further progress the realised system and validate the capability-based approach to development.

CONCLUSION

Following the introduction of the Programs construct within Defence, significant effort has gone into understanding the implications across Defence. This body of work began by taking a capability-based approach to describing the concepts within the Program. The development of an information architecture, to appropriately capture the information, allowed for a greater understanding of the dependencies between projects, products and programs, as well as how those dependencies related to achieving required operational effects. The model-based Program definition capability was then successfully evolved to include information integration and visualisation.

To demonstrate immediate value of the Programs approach, and help contribute to Defence outcomes a software prototype with a whole-of-capability approach was developed. The initial implementation, allowed for communication and information sharing of complex Program-level issues and information through visualisation techniques on Defence systems. Immediate feedback received has been positive, with ongoing effort looking to expand the contribute and realise more potential from a Program-based methodology.

REFERENCES

Bulluss, G, O'Shea, K "Architecture-based Decision Support for Capability Development", DSTO MBSE Symposium 2011, Adelaide, 2011

Cook, S, Unewisse, M, "A SoS Approach for Engineering Capability Programs", 27th Annual INCOSE Internation Symposium, Adelaide, 2017

Dahmann, J., Rebovich, G., Lowry, R., Lane, J., & Baldwin, K., 2011, "An implementers' view of systems engineering for systems of systems", 2011 IEEE International Systems Conference (SysCon), pp. 212-217. IEEE.

Defence, "Defence Capability Development Handbook", Commonwealth of Australia, 2014

Defence, "Interim Capability Lifecycle Manual", Commonwealth of Australia, 2016 & 2017

Gorod, B. Sauser and J. Boardman, "*System-of-Systems Engineering Management: A Review of Modern History and a Path Forward*," in IEEE Systems Journal, vol. 2, no. 4, pp. 484-499, Dec. 2008

Harvey, D, Liddy T, "Architecting the Problem Space – Model-based User Needs Analysis", INCOSE International Symposium, Nevada, 2014

Liddy, T, Harvey, D, Waite, M, Logan, P, "Theatre of Operations", MBSE Symposium, Canberra, 2012

Lowe, D, Flahive A, Unewisse, M, "Improving Joint Force Integration by Design in the new Defence Capability Life Cycle", INCOSE International Symposium, Adelaide, 2017

Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, Systems Engineering Guide for Systems of Systems, Version v1.0, Washington, DC: ODUSD(A&T)SSE, 2008

Pratt, J, Cook, S, Unewisse, M, "A SoSE Principles Framework to Drive Australian Defence Capability Inegration", Systems Engineering Test and Evaluation Conference 2015, Canberra, 2015

Robinson, K. and Graham, D., "An improved methodology for analysis of complex capability", Systems Engineering & Test and Evaluation (SETE 2010), Adelaide, Australia, April 2010.

Robinson, K., Tramoundanis, D., Harvey, D., Jones, M. and Wilson, W., "Demonstrating MBSE for specifying complex capability", Systems Engineering & Test and Evaluation (SETE 2010), Adelaide, Australia, April 2010.

BIOGRAPHIES

Chris French is a Senior Systems Engineer working with experience in developing and understanding some of the complex capabilities within Australian Defence and Transportation sectors. He continues to lead complex projects, through the application of Systems Engineering principles to capability design. Chris graduated with a Bachelor of Engineering (Aerospace) with First Class Honours and a Bachelor of Science (Theoretical Physics) from the University of Adelaide, and is currently studying a Masters in Project Management.

Steven Heard is a Business Analyst with experience in software design and training. He has a decade of Australian Defence Force experience contributing to a deep understanding of the complex Defence environment. His uniformed operational experiences in Iraq and Afghanistan give him practical knowledge around the importance of information in the Defence context.

ISBN for SETE2018: 978-1-925627-15-2