Australian Government
Department of Defence
Defence Science and
Technology Organisation

Fit-for-Purpose Visualisation of Architecture to support Defence Capability Decision-Making

Kevin O'Shea, Peter Pong and Gary Bulluss

Joint Operations Division
Defence Science and Technology Organisation

DSTO-TN-1098

ABSTRACT

Making decisions across the whole portfolio of defence capability requires the integration of information about hundreds of projects and capabilities, from a number of different perspectives. Architectures have been used in the defence community for at least the last decade to address this complexity. However, traditional architecture views are often too complex for decision-makers to readily comprehend. The recently-released DoDAF 2.0 architecture framework promotes the concept of 'fit-for-purpose' views to facilitate decision support from architectural models. The work described in this paper applies a UPDM-based architecture development approach to capture capability development information with an emphasis on developing a fit-for-purpose visualisation to support decision-making. This work includes the development of prototype visualisation software to facilitate decision-support from DoDAF 2.0 architectural models.

Published by

Joint Operations Division
DSTO Defence Science and Technology Organisation
Fairbairn Business Park Department of Defence
Canberra ACT 2600 Australia

Telephone: (02) 6265 9111
Fax: (02) 6128 6332

AR-015-333
July 2012

APPROVED FOR PUBLIC RELEASE

Fit-for-Purpose Visualisation of Architecture to support Defence Capability Decision-Making

Executive Summary

Managing the entire portfolio of defence capability is a highly complex task. Senior decision-makers must understand a broad range of portfolio-level aspects, especially schedule, cost and capability. Whenever changes are made, it is critical that decision-makers are aware of the implications of these changes. This work explores the development of a decision support environment to address this problem, including a prototype solution developed as a proof-of-concept.

This work adopted the DoDAF 2.0 architecture-based approach as the framework to capture the complex data and relationships required. The Unified Profile for DoDAF and MODAF (UPDM), which provides the underlying representation in this work, is an architectural standard for modelling defence concepts, including project scheduling and capability information. The UPDM meta-model, which is compliant to the DoDAF 2.0, can also be extended to incorporate additional concepts, as was required to integrate project costing data in this task.

To support decision making, appropriate visualisation of data is critical. Decision-makers must be able to understand information by examining data in a variety of ways. DoDAF 2.0 promotes the development of 'fit-for-purpose' views to enable a complex data model to be presented in suitable ways according to decision support needs. This work has developed a prototype fit-for-purpose view called 'Program Viewer', which aims to support decision making by presenting high-level scheduling, cost and capability visualisations. Program Viewer is also interactive to enable decision-makers to understand portfolio-level implications when making changes to the data.

This work demonstrated that DoDAF 2.0 and UPDM provide a very capable framework to capture the complex data and relationships required, while facilitating the development of fit-for-purpose views for decision support. UPDM enabled rapid development of both the data model and fit-for-purpose views by leveraging existing standard metadata structures and programming interfaces. This work has achieved a proof-of-concept which demonstrates these benefits, and provides a unique decision support tool for senior decision makers. While this work recognises limitations in the current interoperability of UPDM and the dependence on quality input data, it found that given availability of reliable data, a UPDM-based model coupled with fit-for-purpose views can be an effective approach for decision support.

This page is intentionally blank

Authors

Kevin O'Shea
Joint Operations Division

Kevin O'Shea completed a Bachelor of Software Engineering (Hons.) with the Australian National University in 2010. He has since been working as a graduate in the Joint Operations Division in DSTO, providing contributions in Architecture modelling development as well as technical and software engineering support to JDSC activities.

Peter Pong
Joint Operations Division

Peter Pong completed his BE (Hons.) (1997), and M.Eng.Sc. (1999) both with the University of Queensland, and a M.Sc. (2004) in Mathematics with University of Adelaide. Currently, he is completing his PhD on Information Fusion with the University of Melbourne. He has over 10 years defence experience in the area of airborne and maritime weapon systems. His recent works include a number of Joint programs within RPDE, CDG, NCWPO and CIOG. Peter is a System Engineer working for Jacobs Australia. Currently he is the Knowledge Capture and Structural Representation specialist in JDSC/DSTO, primarily works on Architecture and information capture to support various JDSC activities. His research interests include data fusion, information uncertainty theory, and architecture development.

Gary Bulluss
Joint Operations Division

Gary Bulluss is currently the Director Discovery at the Joint Decision Support Centre, part of Joint Operations Division, DSTO. He has 28 years of experience providing scientific and professional support to the Australian Defence Organisation. This has included software support for naval tactical systems, intelligence analysis, and research into systems of systems and architecture practice. He has also undertaken secondments as the DSTO Scientific Adviser to Coastwatch and the DSTO Representative to the joint defence-industry RPDE program. Gary has a B.Sc. from the University of Sydney and a M.Sc. by research from the University of New South Wales.

This page is intentionally blank

Contents

This page is intentionally blank

List of Abbreviations

API	Application Programming Interface
COM	Component Object Model
DCDH	Defence Capability Development Handbook
DCP	Defence Capability Plan
DoDAF	Department of Defense Architecture Framework
FIC	Fundamental Inputs to Capability
IDEF	Integration Definition
IOC	Initial Operating Capability
JCA	Joint Capability Area
MODAF	Ministry of Defence Architecture Framework
ORC	Options Review Committee
PWD	Planned Withdrawal Date
UML	Unified Modelling Language
UPDM	Unified Profile for DoDAF and MODAF
YOD	Year of Decision

This page is intentionally blank

1. Introduction

Managing the delivery and operational retirement of defence force capability can be a highly complex and challenging task. Decision makers must consider the capabilities needed to meet strategic objectives, and how to deliver a force that can achieve these capabilities. Several additional factors must also be balanced, including government approvals, project interdependencies and the programmatic constraints of scheduling, budgets and resource availability.

In Australia, the primary source describing major capital acquisitions is the Defence Capability Plan (DCP). The DCP describes the projects that will be considered for government approval to contribute to the portfolio of capability that supports Australia's strategic defence objectives. With several hundred projects in the DCP, managing this portfolio of proposed major capability acquisitions can be very difficult. This is further compounded by the need to also manage projects already under acquisition and assets currently in service as part of the portfolio.

The work described in this paper identifies the need for a decision support environment to manage this complexity and present information in a form that can assist in the decision making process. This work involves a data model to capture all relevant data, coupled with an appropriate visualisation tool to comprehend the information for analysis. The work presented in this paper describes a prototype software suite, developed as a proof of concept, to show how such a decision support environment can be achieved.

The purpose of this paper is to describe:

a. The approach taken to develop a prototype, fit-for-purpose, visualisation tool to provide decision support for Defence capability development; and
b. The technical implementation for interfacing this visualisation tool with a standards-based architectural data model.

Note that while this paper contains examples which use real project names, all schedule and costing information presented is fictitious and is used only for illustrative purposes.

2. Background

Integrating a number of key concepts is required to effectively manage the delivery of a capable defence force. This is a portfolio management problem, requiring a balance between schedule, cost and capability aspects. A representation of the concepts involved and the relationships between these concepts is depicted in Figure 1. Projects in the DCP operate under several constraints, including schedule, cost, and other resource constraints. Major platforms are delivered by DCP projects and must be combined with other strategic enablers to provide an effective capability. These strategic enablers are also known as Fundamental Inputs to Capability (FIC), and include aspects such as personnel, training and infrastructure. Capabilities can then be used to perform operational tasks, under the guidance of the Defence strategic vision.

Changes occurring in any of the conceptual areas will have implications for many other aspects. For example if there is a shortage of sufficiently trained personnel to operate equipment, there may be a limitation on capability available to fulfil operational tasks.

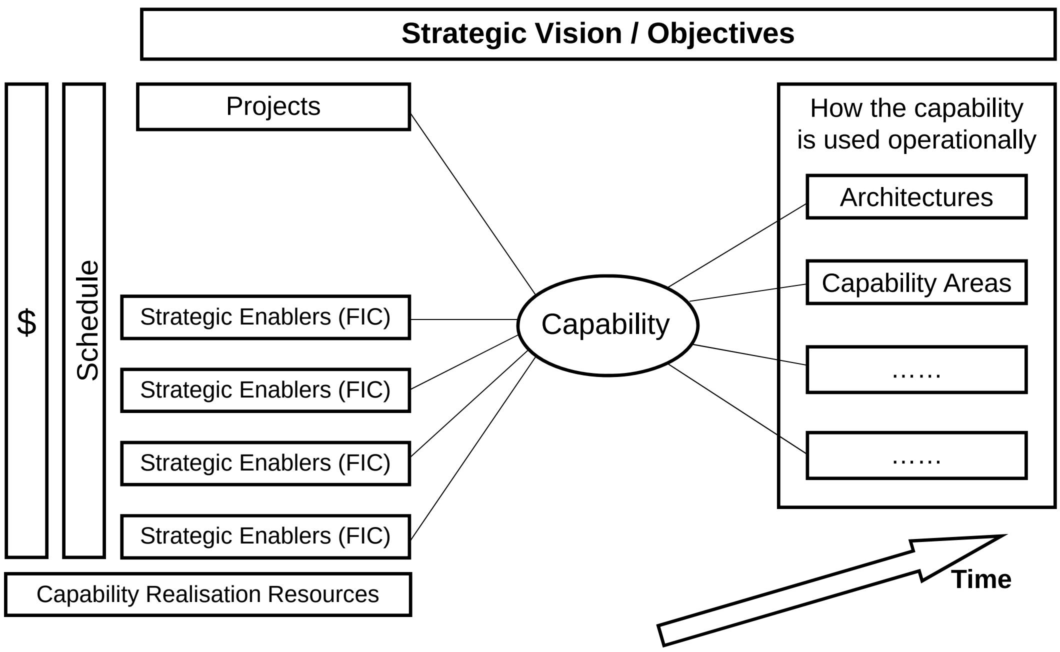

Figure 1 Defence concepts and relationships

In this work, capability is viewed as the ability to achieve an operational effect (DCDH, 2011), for example combat or lift. Capability management includes ensuring capability delivery is aligned with strategic priorities. A common perspective in Defence is to attempt to maintain continuous availability of major capabilities over time, for example maintaining an ongoing amphibious lift capability. This is because many projects in the DCP are replacement projects intended to acquire a new platform to fulfil the same capability role as an existing asset scheduled for retirement. If there are any gaps caused by these transitions, it may be necessary to either adjust project schedules or add new projects to provide an interim capability.

Schedule considerations include managing the ability to meet key milestones, and managing interdependency relationships between projects. In Australia, key milestones in each Defence project include the Year of Decision (YOD) where the project is approved by government, the Initial Operating Capability (IOC) where the capability delivered by the project first becomes available, and the Planned Withdrawal Date (PWD) where an asset is scheduled to withdraw from service. Aligning the PWD of a retiring asset with the IOC of a replacement project is a common scheduling requirement to avoid capability gaps.

More broadly, there are many different types of interdependencies between projects which must be satisfied in order to successfully deliver effective capabilities. This work focuses primarily on schedule interdependencies, capturing sequencing constraints between milestones in different projects. However there remain other types of interdependencies which are yet to be addressed in this work. For example, there are often projects which rely on other enabling projects that provide essential inputs in order to effectively realise their intended capabilities. Most importantly, changes must be managed very carefully, especially when interdependencies exist. When the milestones of a project are shifted, there can be many consequences for that project as well as other interdependent projects. For example, the cost profile of the project will often change, the delivery date of the capability may change, and the interdependency relationships may require additional changes in other projects. This work focuses on providing decision makers with an awareness of these implications when changes are being considered.

3. Overview of Prototype Solution

This work developed a prototype software suite to provide decision support across schedule, cost and capability aspects. The approach taken involves the combination of two main components – a data model to capture the complex data and relationships required, and a fit-for-purpose visualisation that extracts the data and presents information in a form suitable to support decision making. This relationship between these two components is depicted in Figure 2.

Figure 2 Data Model and Fit-for-Purpose Visualisation

In this paper the term 'data model' is defined as an abstract model that organises and documents data and relationships as a platform for communication between stakeholders. While a number of data model notations were examined in this work, it became apparent that most available authoritative data sources to support decision making in the Defence community are often ad hoc and do not conform to any specific data modelling standard such as Integration Definition (IDEF) (IDEF0, 1998) or the Unified Modelling Language (UML). Some exceptions exist, but their schemas are often developed for a specific purpose rather than following an international best practice.

This work implements the data model by leveraging the Unified Profile for DoDAF and MoDAF (UPDM) architectural modelling standard (UPDM, 2011) to represent the DoDAF framework, and then attempts to fit all data into this framework. Aligning with the guidance of DoDAF (DoDAF, 2009) and MODAF (MODAF, 2010), DoDAF 2.0 provides a number of standard views to represent the data while also promoting the development of additional fit-for-purpose views to address different stakeholder needs. For example, senior decision makers often prefer high-level summaries of data and the ability to perform sensitivity analysis. The latest version of UPDM also includes the concept of projects and capability over time, which is ideal to meet the needs of this work.

The concept of fit-for-purpose views is very important for this work. Different stakeholders will have a variety of responsibilities, and thus a number of tailored views are often required. It is important however that all of these views are representing data from the same underlying data model on which decisions are being made. This work leverages the concept of Defence Capability Situational Awareness (DCSA) illustrated in Figure 3 to express the integration of data into a common model that can be examined from a number of different perspectives. Managing the entire Defence portfolio requires a large amount of data that is beyond the cognitive ability of any single individual. A data model to capture this complexity, coupled with fit-for-purpose visualisations, is the key in this work to providing effective decision support.

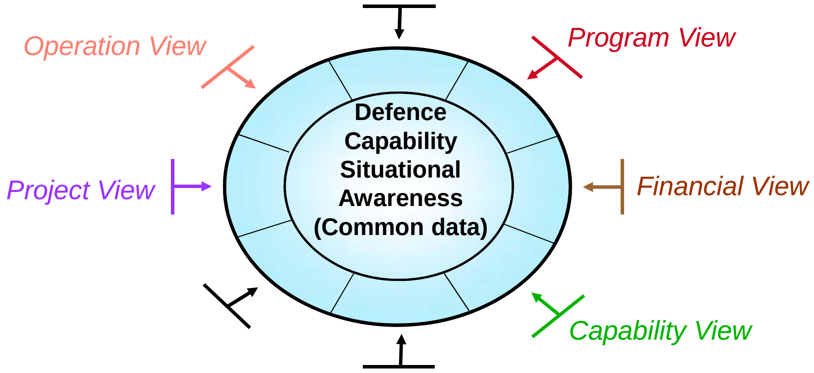

Figure 3 Defence Capability Situational Awareness model

This work has successfully developed a prototype fit-for-purpose view named Program Viewer to visualise data from the UPDM-based data model. Program Viewer is able to extract schedule, cost and capability data from the model and present this data in a suitable form to support senior decision makers. Section 5 of this report describes the features included and their rationale regarding decision support. Section 6 explains the technical implementation details of Program Viewer and the interface with the data model.

4. Data Model

UPDM inherently supports the modelling of the concept of projects and capability. The use of this standard means there is no need to define metadata structures and relationships in an ad-hoc manner. This standard also promotes data interchange and interoperability, which is important for interfacing with other data models and views. This work incorporates a subset of the UPDM-based meta-model shown in Figure 4 to capture the required data and relationships. Expanding this subset to leverage more of the extensive UPDM meta-model enables efficient future development, when modelling of other concepts is required.

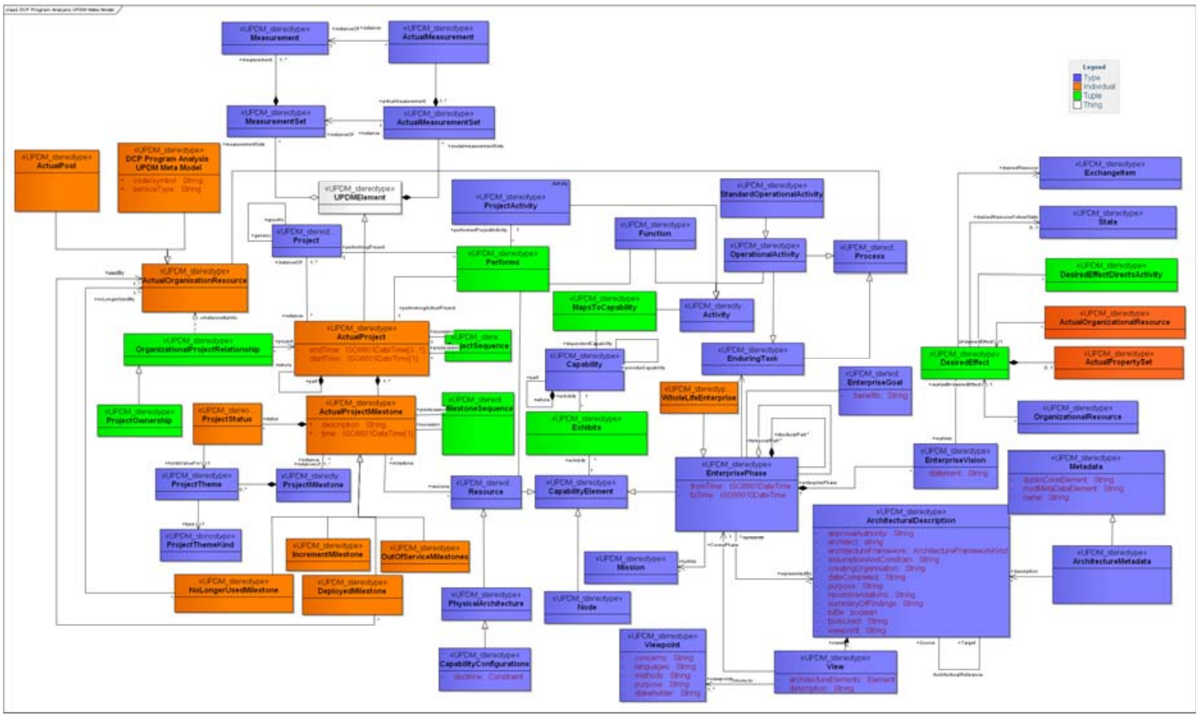

Figure 4 DCSA UPDM-based Meta-Model

All elements from the UPDM meta-model which are used in this work are shown in Figure 4. All attributes and associations of these elements are defined in the meta-model. Projects are defined as instances of *ActualProject* and are associated with their relevant *ActualProjectMilestone*s to capture scheduling data. These milestones can then be associated with a physical *resource* that the project delivers (for example an asset - Platform), and that *resource* can be associated with relevant *Capability* that it provides. The time period in which the *resource*, and thus the *Capability*, is available is based on the time period between the *IncrementMilestone* and *OutOfServiceMilestone* attached to the project delivering that *resource*.

All Defence projects appearing in the DCP, as well as under-acquisition projects and in-service assets, have been modelled as instances of *ActualProject*. Noting that *ActualProject* objects can be composed of other *ActualProject* child objects, this concept is used to capture a hierarchical structure across the Defence portfolio. Portfolio, program, and project levels can be modelled using this hierarchical structure. Each project is associated with its relevant *ActualProjectMilestone*s. These milestones include the various government approval dates, in addition to the capability delivery milestones such as the Initial Operating Capability (IOC) of the assets being acquired by that project. The IOC date defines when the capability will be initially ready for operational use, and thus it is modelled as an *IncrementMilestone* as intended by the UPDM meta-model. Similarly the Planned Withdrawal Date (PWD) is the date when assets are expected to begin withdrawal from service, making it suitable for an *OutOfServiceMilestone* to mark the end of the provision of that capability. *CapabilityConfiguration* is used to represent the physical asset delivered by a project, which is available during the time between the attached *IncrementMilestone* and *OutOfServiceMilestone* as depicted in Figure 5.

Capability taxonomies can have several different perspectives, including strategic, operational and preparedness focuses. The challenge here is trying to integrate multiple perspectives for multiple audiences. Fortunately, UPDM has sufficient flexibility to support capability mapping at multiple levels of abstraction. The recursive structure of Capability compositions in UPDM, as depicted in Figure 5, allows the construction of many-to-many relationships between capabilities while also maintaining the traceability with other domain-specific data such as operational activities, projects and systems.

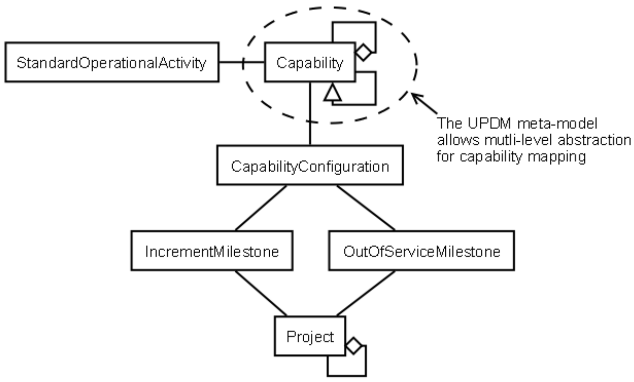

Figure 5 Capability relationships in UPDM

While UPDM offers an extensive coverage of Defence related information, some critical information such as costing is not included in the standard UPDM framework. This work acknowledged costing as an important aspect in capability decision making, however defining a comprehensive cost meta-model extension to UPDM would require significant effort. Instead, this work leveraged the *measurement* concept in UPDM which allows additional properties to be associated with existing UPDM elements. The advantage of this approach is that there is no need to add complexity into the original UPDM meta-model and therefore the unavoidable UPDM deviation is kept to a manageable size. Using the *measurement* concept is also very versatile because it can be associated with other UPDM elements at any level of abstraction, and then data can be aggregated up for portfolio-level decision making. This enables an appropriate level of abstraction to be used depending on the availability of data and the required modelling fidelity.

5. Program Viewer Functionality and Rationale

While a data model can effectively capture complex data and relationships, appropriate views of this data are essential in order to understand the information and make decisions. UPDM provides a number of standard views offered by DoDAF and MODAF, however it is sometimes necessary to develop additional fit-for-purpose views to address all stakeholder needs. Program Viewer is a fit-for-purpose view developed as part of this work which is designed to extract and present data from the UPDM-based data model in ways that support decision making for Australian Defence capability development.

As previously stated, this paper contains examples which use real project names, all schedules and costing information presented is fictitious and is used only for illustrative purposes.

5.1 Schedule and Implications of Change

The functional design of Program Viewer aims to present information in the most appropriate format for senior decision makers to gain a holistic view of important high-level details. Program Viewer presents schedule information using sophisticated Gantt chart views that most senior decision makers are familiar with. The display and interactions are tailored with specific consideration for Australian capability development decision making.

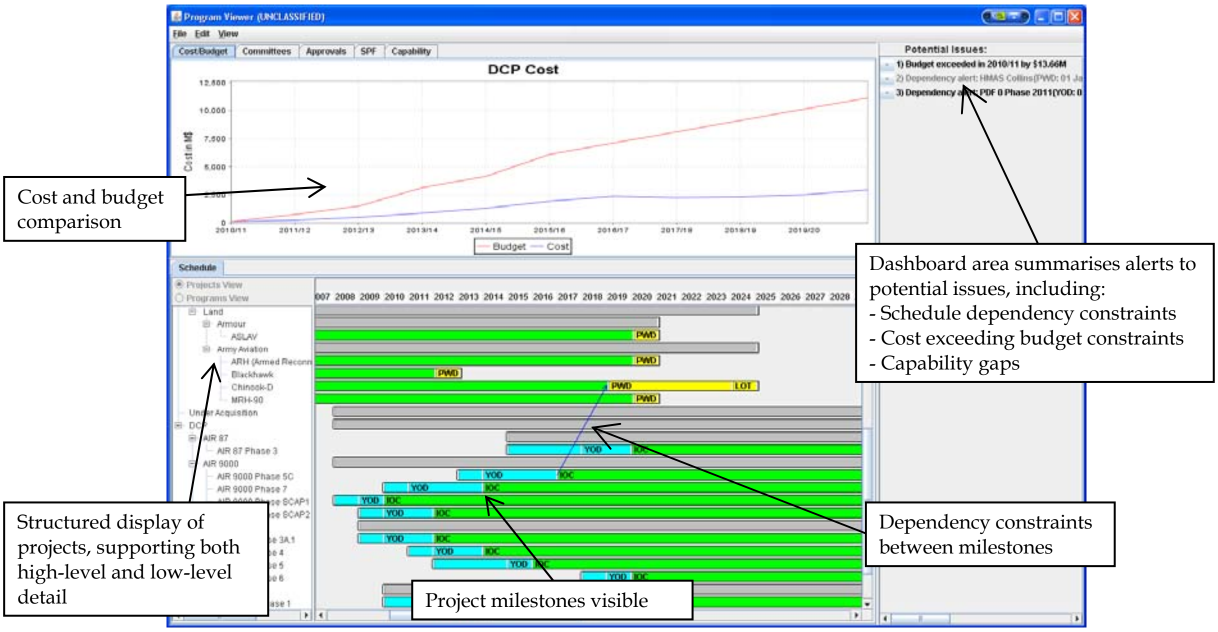

Figure 6 Program Viewer user interface (all data is fictitious)

The Program Viewer user interface depicted in Figure 6 shows the Gantt chart representation of schedule data. This screenshot presents the major features included, however the data shown is fictitious for the purpose of this report. Since the data needs to include all of the projects and milestones in the entire Defence portfolio, this view required careful design to concisely present all the important information. The Gantt chart in Program Viewer has a hierarchical structure to facilitate decomposition of the portfolio into programs and projects. The major milestones of each project are positioned inside each timeline bar for a concise display. The individual segments between milestones are coloured to reflect the status of projects during these phases – for example the initial government approval periods for a project are coloured blue, the period where the capability being delivered by a project is available is coloured green, and the period where the capability is withdrawing from service is coloured yellow. This concise visualisation facilitates quick understanding of high-level information needed for decision making.

Program Viewer supports visualisation and analysis of scheduling dependencies between projects. These are specified as sequencing constraints between milestones, which is an abstraction of the *MilestoneSequence* concept in UPDM. As discussed in Section 2, this concept can be used to capture interdependency constraints between projects. For example potential capability gaps can be visualised by enforcing the Initial Operating Capability (IOC) of a project to precede the Planned Withdrawal Date (PWD) of any assets being replaced by that project. When a platform and its subsystems are being delivered by different projects, this mechanism can also help to align the IOC dates of these projects so they will be ready to enter into service at the same time.

A key feature of Program Viewer is to allow users to make changes to the data and provide information about the potential implications of these changes. Program Viewer enables users to shift project milestones by dragging them left or right on the screen. Upon making changes, programmatic constraints are automatically checked to identify areas which may have been affected by the changes. These areas include capability issues, budget constraints and schedule interdependencies. If any new potential issues are identified by these changes, alerts are presented in red text on the right hand side of the user interface. Similarly, if issues are fixed or improved by these changes, green text is presented to reflect these improvements. An example of Program Viewer demonstrating this functionality is depicted in Figure 7, where a user has shifted a milestone and subsequently broken a schedule interdependency constraint. In this example, a YOD milestone is initially shifted to the right, labelled by step 1 in Figure 7. This causes the following IOC milestone to also shift to the right, breaking the dependency constraint labelled in step 2. An alert of this constraint is then presented to the user, as labelled by step 3 in Figure 7. The dependency constraint in this example captures a replacement relationship, where existing assets are being replaced by new assets delivered by a DCP project. Breaking this constraint will indicate a gap during this transition, which may cause a reduction in available capability during this time. Similarly, green text is shown to indicate any improvements made to the data, such as reducing over-spending or fixing a broken dependency constraint. This functionality demonstrates the ability of Program Viewer to provide decision makers with an awareness of the implications of changes.

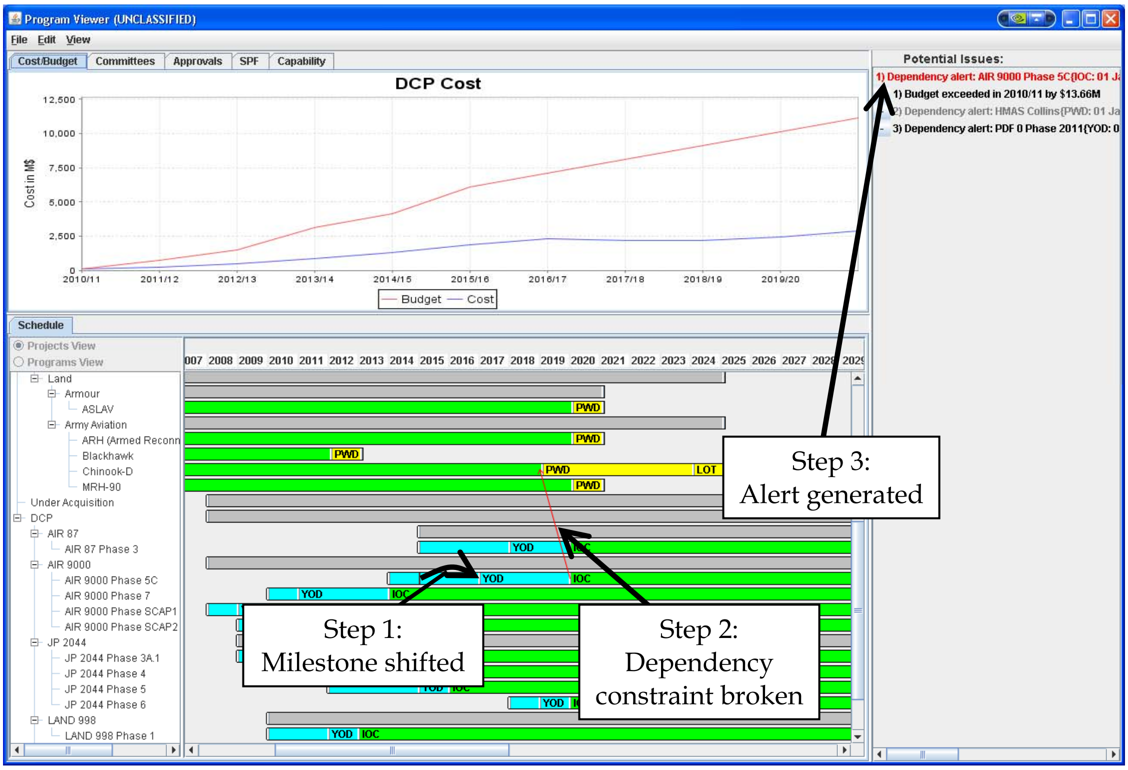

Figure 7 Example issues caused by making changes in Program Viewer

Once decision makers are satisfied with changes made, the updated data can be written back into the model directly from within the Program Viewer user interface. This completes the full abstraction from the underlying data model. Decision makers using Program Viewer do not require any knowledge of UPDM or the underlying data model, but can benefit from the advantages gained by using UPDM for flexible and rapid development.

5.2 Committee Approvals

All projects appearing in the DCP must be submitted to government and various Defence committees for approval. This process introduces a resource management problem because committees have limited capacity for considering all projects. Each project must pass several committee approvals, and the approval requirements can vary depending on the project (DCDH, 2011). Project scheduling must consider the alignment of these committee approvals as a constraint on when the capability can be delivered. Any project that is unable to pass a scheduled committee will be delayed, which could result in a number of programmatic issues.

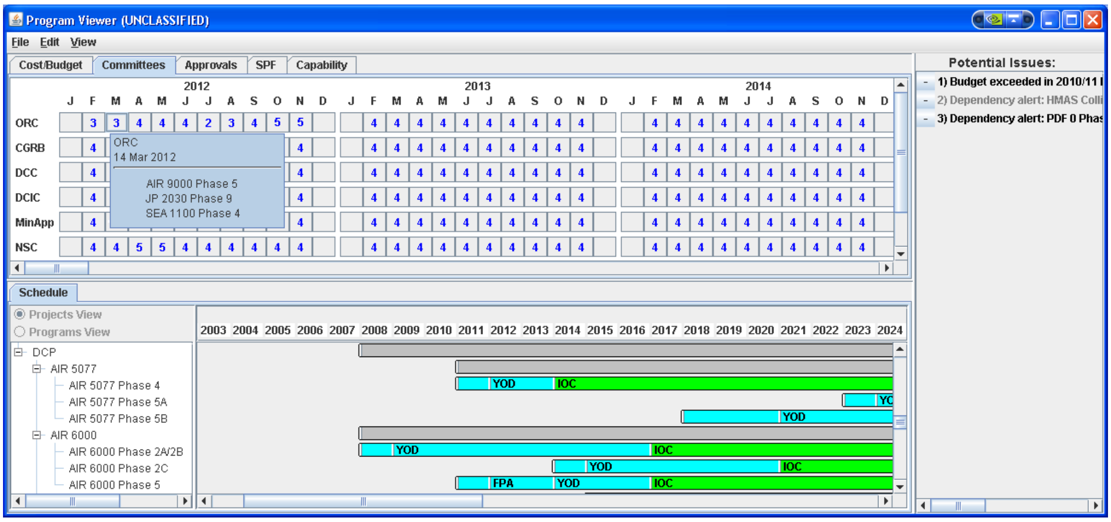

Figure 8 Committee visualisation example in Program Viewer

Program Viewer includes a simple committee-loading visualisation (Figure 8). Every project includes milestones reflecting their committee submission schedule. Program Viewer aggregates the numbers of projects scheduled for each committee into a table. For example in Figure 8 the Options Review Committee (ORC) for March 2012 shows the number 3, representing three projects scheduled for that committee. In addition, users may hover the mouse over any committee number to see which projects are scheduled for that committee, as shown in Figure 8. This view therefore provides an overview of committee approval loads coupled with an ability to look into the details for any particular committee.

The total number of projects scheduled for each committee must be balanced in order to minimize the risk of project slippage. However, identifying these risks is a challenging analytical task. Simply comparing only the numbers of scheduled projects as depicted in Figure 8 may be misleading. For example, several projects can be considered at the same time if they are similar or closely related, whereas complex projects will inevitably take longer to approve. Quantifying the complexity of a project could include many factors, such as the cost of the project or the number of interdependencies with other projects. However it is challenging to present analytical insights of this data in an accurate and useful manner in all cases.

5.3 Cost

It is essential for this work to incorporate the financial aspect as one of the major factors for decision making. Sophisticated financial modelling can be very complex however, requiring significant time and effort to implement. Instead, this work uses high-level summarised data which is easier to implement and is also appropriate for senior decision makers.

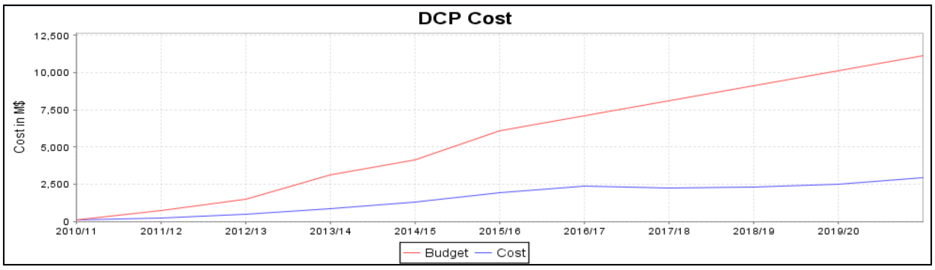

Figure 9 Cost visualisation example in Program Viewer

Program Viewer presents a chart comparing total planned costing against available funds for each financial year, as shown in Figure 9. The blue line labelled 'Cost' represents the aggregation of planned spending spreads across all projects in the DCP, and the red line labelled 'Budget' represents the total funding available for these projects. Comparing these two lines will indicate any financial years where the planned spending exceeds the planned budget. If a decision is made to shift project approval milestones, the planned spending spread associated with that project is also shifted. This basic cost modelling approach is deemed sufficient for this work. This also demonstrates that cost can indeed be included in a UPDM model associated with projects and capability data, as discussed in Section 4. With this proof of concept, there is potential to incorporate more sophisticated cost models as required.

5.4 Capability

A key focus in this work is managing the sustainment of capability over time. The scheduling of projects in the DCP will determine the availability of the capability being delivered by these projects. The challenge is to manage these schedules to minimize gaps in capability.

Presenting capability over time can be achieved by standard views available in UPDM. The DoDAF-based standard CV-3 Capability Phasing viewpoint depicted in Figure 10 illustrates an example where 'Battlefield Lift Heavy' capability is initially supported by CH-47D Chinooks and later replaced by new Chinooks delivered by the DCP project AIR 9000 Phase 5C. CASE 1 in Figure 10 highlights a tight transition where there is little flexibility – if the AIR 9000 Phase 5C project were to suffer any delays, there would be a resulting gap as depicted in CASE 2.

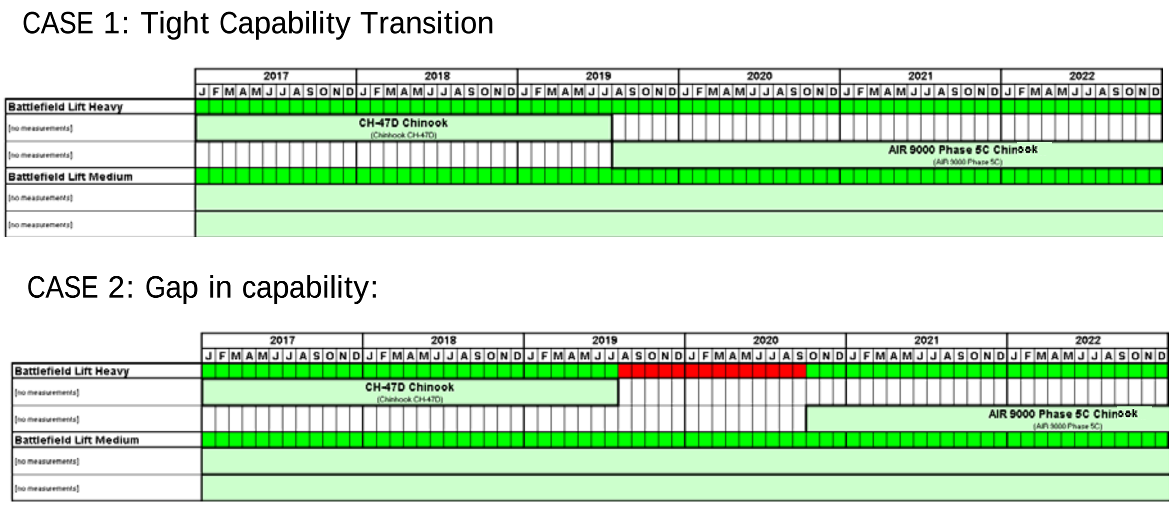

Figure 10   Standard DoDAF CV-3 capability viewpoint

While it can be an effective way to visualise capability issues, this representation scheme is insufficient for the decision support needs in this work. This view would be appropriate for a small number of capabilities and projects, whereas this work requires all Defence projects and capabilities to be captured in a single model. In this context, using CV-3 representations would become too large and cumbersome to be useful for decision making.

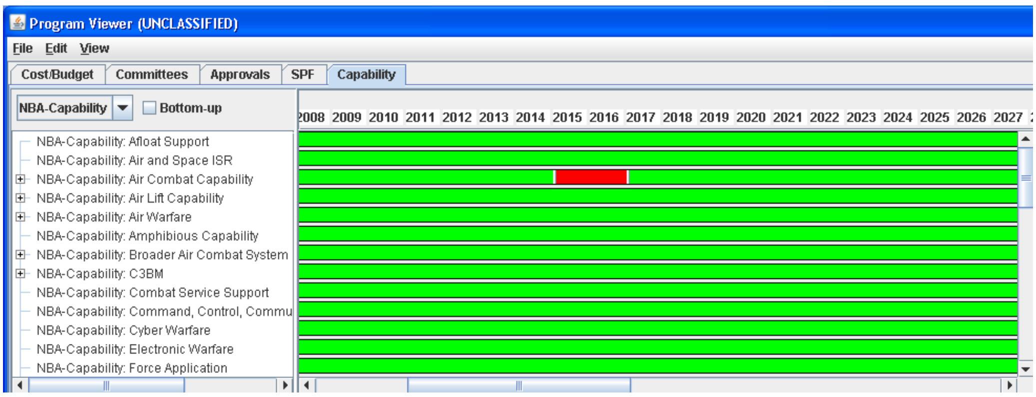

Figure 11   Capability visualisation example in Program Viewer

Program Viewer leverages the concepts in the standard CV-3 viewpoint for visualising capability realisation over time, and also introduces a number of additional features to support decision making over a large data set. The visualisation example depicted in Figure 11 demonstrates the Gantt chart representation for capability in Program Viewer. Initially this capability viewpoint shows only high-level bars depicting the availability of capability, without the lower-level force elements which contribute to the realisation of these capabilities. This presents a concise view enabling identification of capability gaps more easily than in the standard CV-3 format. Each of the bars shown in Figure 11 can be expanded to show lower-level details for particular capabilities as required. An example is shown in Figure 12 where 'Air Combat' capability has been expanded to identify the cause of the gap. This flexibility allows decision makers to see a high-level view across the entire Defence capability and also be able to see greater detail in particular areas of concern.

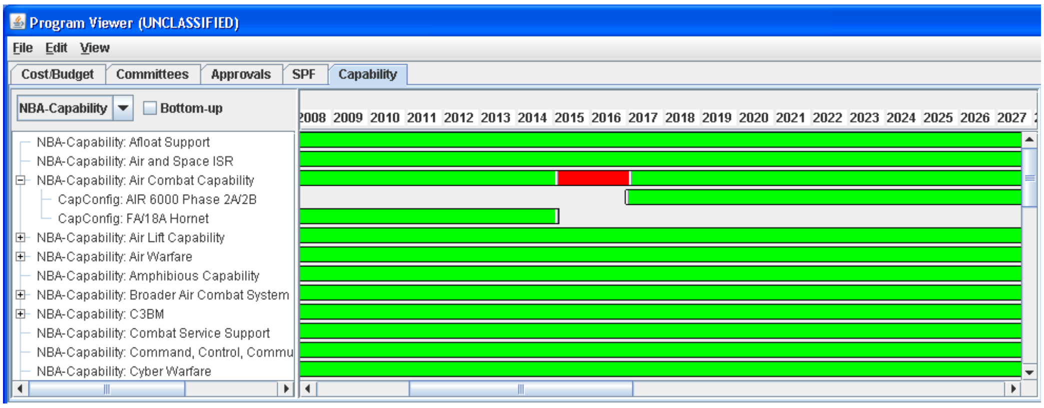

Figure 12 Capability gap example in Program Viewer

The hierarchical structure of this view enables visualisation and analysis of data at multiple levels of abstraction. This feature is important because many capability taxonomies are hierarchical, such as the Joint Capability Areas (JCA) framework used by the United States Department of Defense for functionally grouping military capabilities (Future Joint Warfare, 2012). Using the standard CV-3 visualisation would only allow a flat view of capabilities associated with force elements. Program Viewer however presents aggregated bars for the higher-level capabilities as well, which are derived from the lower-level capability bars. For example, Lift is a logistical capability to move forces and supplies, and this could be achieved by air, land or maritime support. In analysing an operational scenario, these specific types of Lift could be visualised independently if required. However if the specific type of Lift is not important, the generic concept of Lift can be visualised as the union of the lower-level capabilities for each type of Lift. This allows decision makers to analyse problems at the most appropriate level of abstraction.

Regardless of the particular capability taxonomy used, gaps in capability may be introduced when changes are made to the DCP. Program Viewer is designed to provide alerts to potential capability gaps as changes are being made. If decision makers are concerned with balancing aspects other than capability, such as cost and other resource constraints, the implications to capability may not be fully appreciated without these alerts.

6. Technical Implementation

This section describes the technical implementation details of the prototype solution, including the architectural design of the fit-for-purpose visualisation Program Viewer and how the interface with the underlying data model is realised. This section also discusses the associated assumptions and technical limitations encountered during development.

6.1 Software Architecture

The high-level architecture of Program Viewer and its interaction with external components is depicted in Figure 13. Program Viewer has four major modules, namely *Model*, *Views, Data Analysis,* and *Data Manager*. The *Data Manager* provides the integration between Program Viewer and the data model. Program Viewer has been developed in the Java programming language, which provides compatibility advantages and support for efficient and robust prototyping development.

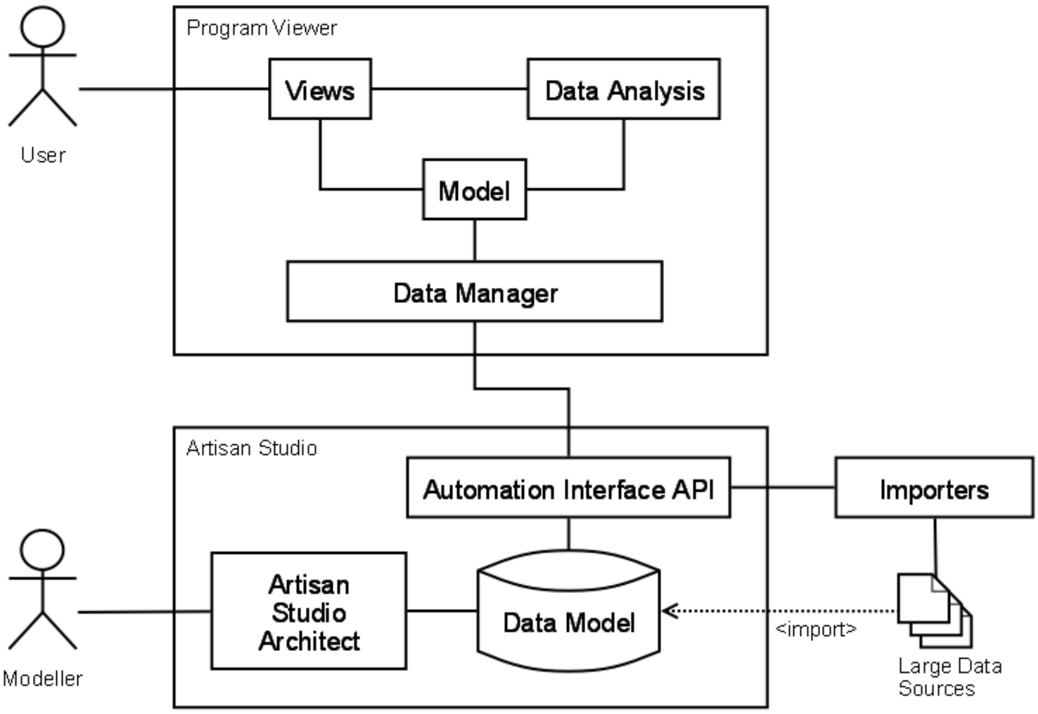

Figure 13 Architecture of Program Viewer and Data Model environment

The *Model* module resides at the centre of the Program Viewer software architecture, providing the core data structures upon which Program Viewer operates. This data includes all project and milestone information, costing data structures, systems and capabilities. The designs of these data structures are intended to match the corresponding structures they represent in the UPDM standard. This approach enables flexibility as the data model is developed, and also promotes interoperability with other UPDM models.

The *Views* module is responsible for accessing the data in the *Model* module and presenting information to decision makers. These representations are the fit-for-purpose viewpoints described in Section 5. Each of these viewpoints integrates a subset of data from the *Model* according to decision support needs. The *Views* module also relies on the *Data Analysis* module which analyses all of the constraints in the data. If the *Data Analysis* module identifies any potential issues, alerts are presented to users through the appropriate viewpoints in the *Views* module.

The *Data Manager* module is responsible for all communication with the external data model. Reading and writing data between the *Model* module and the data model is performed by the *Data Manager*. Separating the *Model* and *Data Manager* allows compatibility with different data model implementations. The only modifications required to enable Program Viewer to interface with other UPDM-based models are the reading and writing procedures in this *Data* *Manager* module.

For this work the data model has been implemented using the *Artisan Studio*™ enterprise architecture software by Atego (Atego, 2012). By employing the UPDM standard, any modelling software which is compliant with the UPDM standard could have been used in this work. Artisan Studio was chosen for its sophisticated user interface and inclusion of features such as database configuration management and programming interfaces.

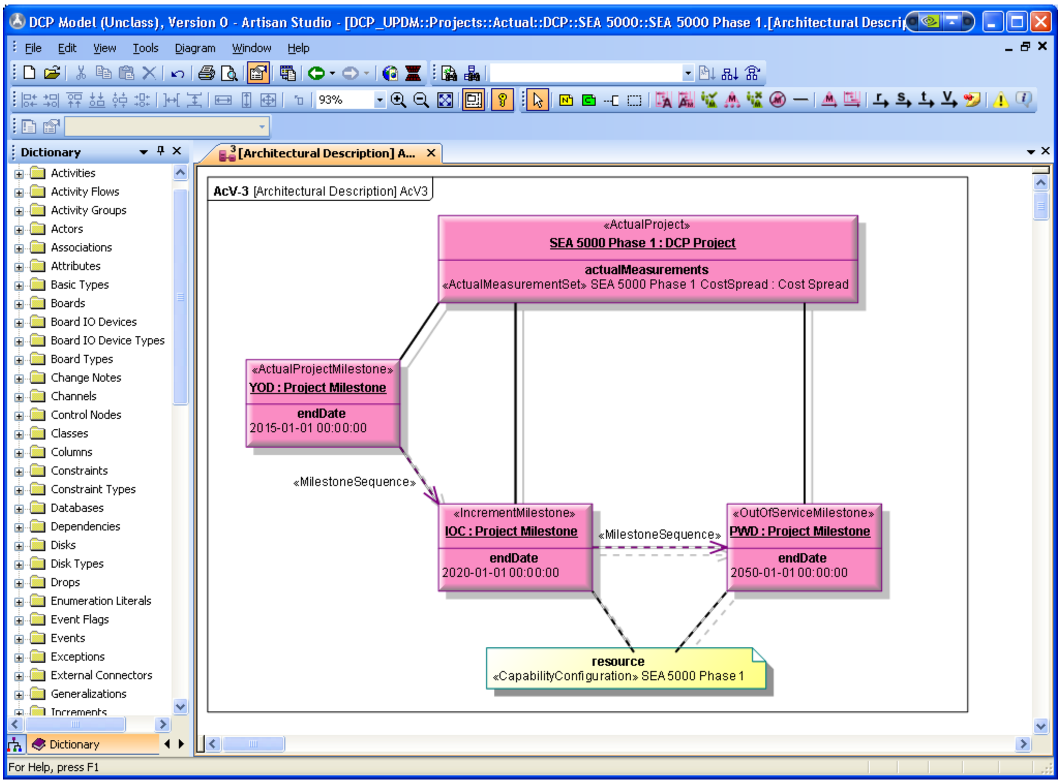

Figure 14 Artisan Studio™ *modelling interface*

The Artisan Studio user interface depicted in Figure 14 provides the environment in which the data model can be constructed. This environment allows complex data and relationships to be captured, including projects, milestones, systems, and capabilities. Once the data model is populated, the data captured by the modeller can then be presented through customised visualisations in Program Viewer. Any simple changes such as shifting milestone dates can be performed within Program Viewer by users. However if more sophisticated modelling changes are required, such as composing 'systems-of-systems' hierarchies, a modeller can build these advanced structures using the complete modelling environment.

Artisan Studio provides a useful Application Programming Interface (API) called the *Automation Interface* which enables other software applications to interact with the modelling environment and underlying data model. This API is the key which enables Program Viewer to communicate with the data model. This communication occurs via the Component Object Model (COM) inter-process communication mechanism. In Program Viewer the *Data Manager* module makes use of the *Automation Interface* to read and write data in the data model, according to the actions performed by users in the fit-for-purpose views.

The *Automation Interface* also proved valuable when populating the data model with the large amount of input data required for this work. Since this work covered the entire Australian Defence portfolio, including all projects, costing and capability, it was infeasible for a modeller to manually enter all of this data through the Artisan Studio modelling interface. Instead, this work developed additional software, denoted as *Importers* in Figure 13, to read external data sources and automatically create the corresponding elements in the data model. This yields significant benefits in time and effort, particularly since these data sources are continuously changing and therefore the data model must be frequently updated to remain consistent. Automating this process is the only feasible solution with such a large input data set.

6.2 Assumptions and Limitations

Implementing the data model using UPDM relies on the assumption that UPDM is capable of supporting most of the concepts that need to be captured. This work has found UPDM to be suitable for capturing the core concepts needed for projects and capabilities. For any concepts that are not included in the UPDM standard, the meta-model would have to be extended to incorporate the data. While the Artisan Studio modelling software is able to facilitate these extensions if necessary, additional time and effort would be required.

Designing the core of Program Viewer to match the structures of corresponding concepts in UPDM introduces limitations on design flexibility. This approach is beneficial for enabling extensions as the data model is developed, however the design for decision support features can be less intuitive. For example, many projects in the DCP deliver major systems, however in UPDM a *Project* and a *System* must be related via *Milestone* objects. This means there is no immediate link between a *Project* and a *System*, and the only way to present this information to a decision maker would be to derive these links through the intermediate *Milestones*. This introduces complexity that could have been avoided if the design had not been dependent on the structure of UDPM.

UPDM as a standard is intended to support interoperability between different modelling tools, however in this work the standard interchange was found to be less effective than expected. This work experimented with XMI interchange between multiple modelling tools, including Enterprise Architect™ by Sparx Systems, Cameo™ by Magic Draw and Artisan Studio™ by Atego. The exported XMI from these tools were not identical and did not include all of the data from the model. Achieving a working standardised interchange between these tools is currently under development by the Object Meta Group (OMG) Model Interchange Working Group (MIWG). Model interchange was demonstrated at the recent DoD Enterprise Architecture Conference (Hause, 2012). Note that, with the current version of Artisan Studio™ used for this work, the *Automation Interface* was used to provide complete access to all data in the model, but this interface is dependent on Artisan Studio™ and is not a standard interchange method.

Since this work integrates a number of existing data sources, accuracy of data is a critical limitation. Data from all sources must be current and consistent in order to provide accurate decision support outputs. While this work recognises this limitation, the process of integrating data from several data sources and visualising the results can also enable decision makers to identify, and subsequently resolve, data inaccuracies. Inconsistencies may only become apparent once the data has been integrated with other related data. If this same data is also being used to support decision making in other contexts, improving the quality of this data will also help to improve decision-making outcomes beyond the scope of this work.

7. Conclusions and Future Work

This work has achieved a proof-of-concept for employing a standard architecture-based approach with fit-for-purpose views to support decision making. Beyond this proof-of-concept, the fit-for-purpose view Program Viewer developed in this work is a unique decision-support tool which is capable of providing sophisticated support to senior Defence decision makers. Although limitations still exist, this work finds that DoDAF and UPDM provide a very capable framework for developing complex data models and for facilitating decision support from these models.

A number of future extensions of this work are possible. Program Viewer may be extended to include additional analytical features to further enhance decision support. With an initial focus on high-level portfolio management, there is scope for incorporating greater fidelity to enable lower-level issues to be analysed in the context of wider portfolio implications. Use of the data model may also be extended to incorporate a wider subset of UPDM. However future extensions will inevitably depend on availability of reliable and consistent data.

8. References

Artisan Studio™, Atego, accessed 8 February 2012, URL http://www.atego.com

Atego 2012, Atego corporate website, accessed 8 February 2012, URL http://www.atego.com

Cameo™, No Magic, Inc, accessed 8 February 2012, URL http://www.nomagic.com/

DCDH 2011, Defence Capability Development Handbook, Department of Defence, Australia 2011

DCP 2009, Defence Capability Plan, Department of Defence, Australia, 2009

DoDAF 2009, Department of Defense Architecture Framework, Version 2.0, Volume 1, 2 and 3, Department of Defense, United States, 2009

Enterprise Architect™, Sparx Systems Pty Ltd, accessed 8 February 2012, URL http://www.sparxsystems.com.au/

Hause 2012, Hause, M,.and Bleakley, G., Unified Profile for DoDAF and MODAF (UPDM) - Architecture Exchange, 2012 DoD Enterprise Architecture Conference, 30 April – 3 May 2012, Miami, Florida, USA

IDEF0 1998, 1320.1-1998 IEEE Standard for Functional Modeling Language – Syntax and Semantics for IDEF0, IEEE Standard, 25 June 1998

JCCD 2012, Future Joint Warfare, Defence Technical Information Centre, Joint Concepts to Capabilities Division, accessed 8 February 2012, URL http://www.dtic.mil/futurejointwarfare/jca.htm

MODAF 2010, Ministry of Defence Architecture Framework, Version 1.2.004, Ministry of Defence, United Kingdom, 2010

UPDM 2011, Unified Profile for DoDAF and MODAF, Version 1.1, Object Management Group, May 2011.

This page is intentionally blank

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)
Fit-for-Purpose Visualisation of Architecture to support Defence Capability Decision-Making	Document (U) Title (U) Abstract (U)
4. AUTHOR(S)	5. CORPORATE AUTHOR
Kevin O'Shea, Peter Pong and Gary Bulluss	DSTO Defence Science and Technology Organisation Fairbairn Business Park Department of Defence Canberra ACT 2600 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TN-1098	AR-015-333	Technical Note	July 2012

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
2012/1052968	07-271	DGCP	19	12

13. DSTO Publications Repository	14. RELEASE AUTHORITY
http://dspace.dsto.defence.gov.au/dspace/	Chief, Joint Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS

Decision making, Visualisation, Architecture, Modelling

19. ABSTRACT
Making decisions across the whole portfolio of defence capability requires the integration of information about hundreds of projects and capabilities, from a number of different perspectives. Architectures have been used in the defence community for at least the last decade to address this complexity. However, traditional architecture views are often too complex for decision-makers to readily comprehend. The recently-released DoDAF 2.0 architecture framework promotes the concept of 'fit-for-purpose' views to facilitate decision support from architectural models. The work described in this paper applies a UPDM-based architecture development approach to capture capability development information with an emphasis on developing a fit-for-purpose visualisation to support decision-making. This work includes the development of prototype visualisation software to facilitate decision-support from DoDAF 2.0 architectural models.

Page classification: UNCLASSIFIED