Integration of M&S (Modeling and Simulation), Software Design and DoDAF (Department of Defense Architecture Framework (RT 24)

A013 - Final Technical Report SERC-2012-TR-024

Principal Investigator

Dr. Michael zur Muehlen, Stevens Institute of Technology

Team Members

Drew Hamilton, PhD, Auburn University

Russel Peak, PhD, Georgia Tech University

April 9, 2012

Contract Number: H98230-08-D-0171 DO 001 TO 002 RT 024

Report No. SERC-2012-TR-024

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
09 APR 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Integration of M&S (Modeling and Simulation), Software Design and DoDAF (Department of Defense Architecture Framework (RT 24)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Systems Engineering Research Center,Stevens Institute of Technology ,1 Castle Point on Hudson,Hoboken,NJ,07030**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Different Department of Defense communities prepare models for architecture compliance (e.g., to maintain JCIDS requirements), for simulation purposes (e.g. for performance estimates) and software engineering (e.g, for model-based code generation). Little, if any, information transfer and model reuse takes place across these communities of interest, which leads to redundant efforts, models that are out of sync, and lost domain knowledge. Differences in methods, tools and data formats are a major reason for this disconnect. The charter of RT 24 was to investigate mechanisms that could help bridge the divide between the modeling & simulation, software engineering, and enterprise architecture modeling communities.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**110**	

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

ABSTRACT

Different Department of Defense communities prepare models for architecture compliance (e.g., to maintain JCIDS requirements), for simulation purposes (e.g., for performance estimates) and software engineering (e.g, for model-based code generation). Little, if any, information transfer and model reuse takes place across these communities of interest, which leads to redundant efforts, models that are out of sync, and lost domain knowledge. Differences in methods, tools, and data formats are a major reason for this disconnect. The charter of RT 24 was to investigate mechanisms that could help bridge the divide between the modeling & simulation, software engineering, and enterprise architecture modeling communities.

This page intentionally left blank

TABLE OF CONTENTS

FIGURES AND TABLES

SUMMARY

RT-24 compared different techniques for the specification of software requirements (SysML), simulation models (Arena), and enterprise architecture (BPMN), and used the DoD Architecture Framework (DoDAF) 2.0 Meta Model as an upper ontology to relate the concepts of each language. To facilitate the interchange of models, a prototype converter between BPMN 2.0 XML and Arena was developed that allows a user to take a BPMN process model and reuse it as the basis for a discrete event simulation. In parallel, a use case consisting of news ingestion, news analysis and news distribution processes was developed and documented using typical DoDAF 2.0 views. This use case was compared to a combat medic training use case to assess the usefulness of different DoDAF views. We determined that usefulness depends on the stakeholders' intent – views that were useful for project managers were not useful for software developers, required views were deemed redundant, views that are currently not required were seen as useful for software engineering purposes. A prototypical simulation and workflow implementation of the news use case was conducted to illustrate the degree of model reuse achievable and the technical conversions that remain.

INTRODUCTION

MOTIVATION

The development of software-intensive systems combines code-oriented software engineering practices with enterprise architecture practices for the design of large-scale systems. While traditional software systems could be architected and developed using waterfall-style methods that placed emphasis on upfront analysis, architecture and design, rapid development cycles and continuous improvement of systems that are already deployed require the creation of architecture documentation as a parallel activity to technical system development. The Department of Defense (DoD) acquisition community places great rigor on proper architecture documentation to demonstrate compliance with the requirements of the Joint Capabilities Integration and Development System (JCIDS). While these models are created to demonstrate compliance with the acquisition process, their potential to drive simulation and execution are rarely realized.

Software-intensive systems are increasingly deployed in distributed scenarios, with independent, interconnected system nodes that communicate using standard protocols and message formats. This distributed nature complicates the development of test scenarios that properly reflect the topology and behavior of independent distributed components. Modeling and Simulation (M&S) technology are essential to understand the behavior of the target system and/or to evaluate various strategies for the operation of the system before it is actually built. In many cases, simulation models reflect the design of the final system in great detail and can take the place of architecture documentation. In an ideal scenario, system architecture artifacts should be directly executable and could be leveraged for simulation purposes.

The non-functional properties of software-intensive systems are typically captured in requirements management environments that are largely disconnected from the actual codebase or the architecture tools used to create conceptual models and graphical representations of the system architecture, and the simulation tools used to evaluate the behavior and performance of the target system architecture. This is a problem because developers and architects need to manually link requirements to the data in the other tools, and this process needs to be repeated every time significant changes occur in either environment.

PROJECT APPROACH

This final report reports on the results of an 18-month investigation into the synergies between enterprise architecture, modeling & simulation, and software engineering. The project proceeded in three stages.

During the first project phase, methods and techniques for system and software modeling were in focus. The project team created mappings between different modeling languages. These languages were chosen for their widespread use in the modeling and architecture community, and included open standards such as BPMN 2.0 (Object Management Group, 2011), SysML (Object Management Group, 2008), and proprietary formats, such as the simulation model format of Arena (David, Sadowski, & Sturrock, 2004).

The second phase of the research task focused on the modeling views that were required for system documentation, and useful for system design. Starting with the full set of architecture views of the DoDAF 2.0 framework (U.S. Department of Defense, 2009) the team created architecture documentation for a medical training simulator in order to determine the usefulness of DoDAF for software engineering. In parallel, the team mapped different model types of SysML to the DoDAF 2.0 views in order to determine to what extent formal models with SysML would cover the DoDAF 2.0 views.

The final phase of the project implemented prototypes to illustrate the feasibility of the findings from phase 1 and 2. In the simulation area, the team developed a converted from BPMN 2.0 XML to Arena in order to demonstrate the reuse of conceptual models for simulation purposes. In the software engineering area, the team transferred requirements models from a news use case into an executable workflow model.

FINDINGS

The project results are the following set of recommendations:

- Good software architecture requires a clear definition of underlying terms and conditions. The first step of any architecture and development project should be the definition of key vocabulary. This includes capabilities, activities, resources, and performers.
    - *Capabilities* describe the desired effects of the system under development. They can be regarded as goal statements and form the basis for DoDAF views CV-1 et al. Capabilities should be described using a phrase that includes “The system should have the ability to … [achieve a desired effect]”.
    - *Activities* are the actions that transform an input into a desired output. They are essential to the realization of capabilities. Activities can be described at different levels of granularity. At the

highest level, an activity equals a process and can be described through the access mechanisms and results of the process, e.g. in form of a service. At lower levels processes are composed of multiple activities that are in some logical and temporal order due to their input and output relationships. Activities form the basis dor DoDAF views OV-5a, OV-6c, SvcV-10c and SV-10c.
    - *Resources* represent the inputs and outputs to the aforementioned activities. They can be data or material, actors or systems. Resources form the basis of the DoDAF views DIV-1 through DIV-3, and are used in numerous other views.
    - *Performers* are the actors that are responsible for the execution of activities. This includes both humans and machines.

  It should be noted that capabilities, activities, resources and performers are not the only concepts that an architect and modeler should consider, but they form a solid foundation for additional design activies that could extend to events, network aspects, interfaces between system components and the like.

- *Not all perspectives of the DoDAF 2.0 framework are equally valuable* for software-intensive design projects. Useful perspectives include the CV-6, DIV-1, OV-6c, and SV-10a perspectives.
- *Process models* integrate concepts from many disparate architecture views and should be developed early in the system development life cycle, both to check the completeness of the architecture scope and the dynamic behavior of the intended system.
- *Simulation models* are a valuable design aid that is increasingly affordable to integrate in the software design process. Since many complex software systems are model driven (e.g., because they rely of EAI integration technology that provides models as an abstraction mechanism from code), adding simulation parameters to these models allows architects and developers to estimate system performance without building interfaces to external systems, an often costly and time-consuming activity.

PHASE 1: LANGUAGE MAPPING

The underlying assumption of this phase was that different modeling communities have a shared, but not clearly articulated view on information systems, and rely on distinct modeling languages to document their respective models.

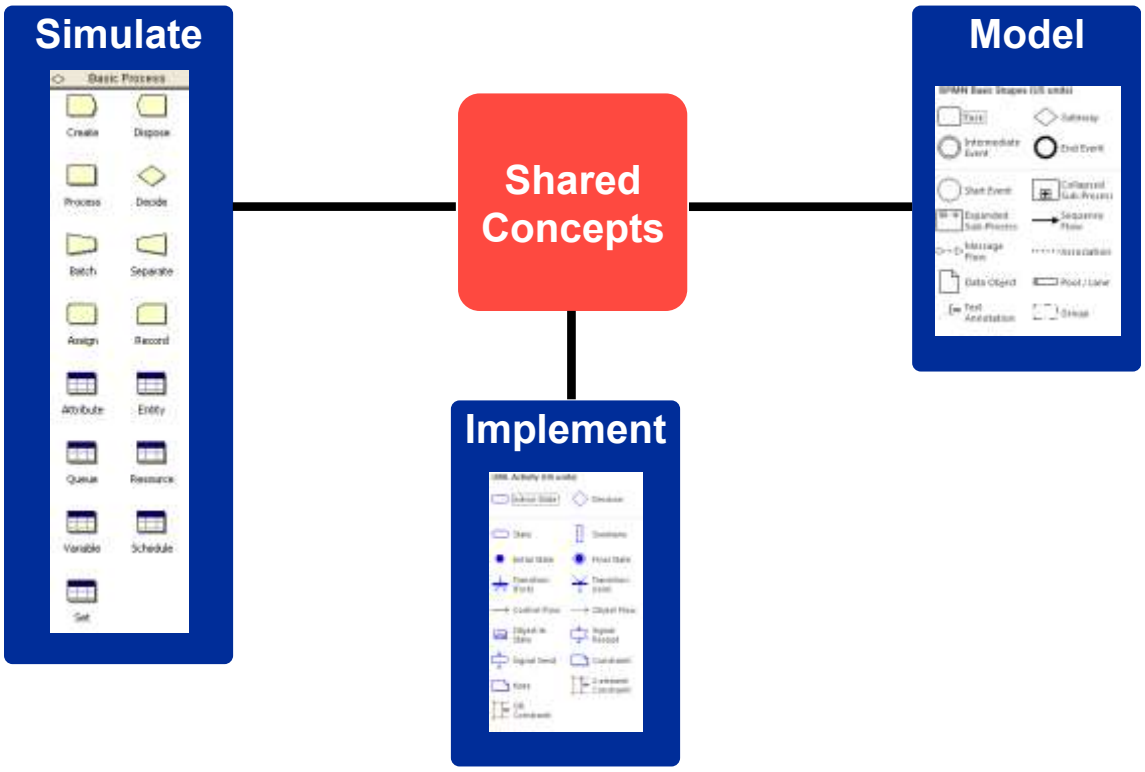

Figure 1: Shared Concepts between Modeling, Simulation & Architecture

Figure 1 illustrates this approach. Modelers, simulation experts and software developers increasingly rely on process models to capture the dynamic behavior of the system they are designing. While the implementation community typically uses UML or the more recent UML-based SysML Activity Diagrams, the process modeling community generally prefers the Business Process Model & Notation standard (zur Muehlen & Recker, 2008). The modeling languages used by the simulation community are largely determined by the toolsets that are used to create simulation models, and in the context of this research task Rockwell Software's Arena toolkit was chosen as a representative of this domain.

The result of this phase were defined mappings between the modeling languages that allowed the research team to identify shared concepts which served as the entry point for the second phase of the project: identifying core modeling constructs that are shared among all three communities. In addition, the mapping between BPMN 2.0 and Arena served as the basis for the design of a model converter designed in phase 3 of the project.

SysML Diagram Types

SysML is a general-purpose graphical modeling language that supports the analysis, specification, design, verification, and validation of complex systems. The language is intended to help specify and architect systems and specify its components.

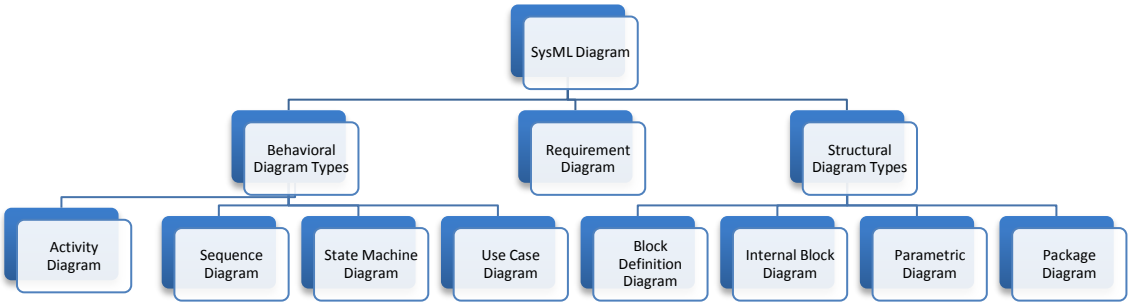

Figure 2: SysML Diagram Types

SysML includes nine diagram types as shown in the diagram taxonomy in Figure 2. Each diagram type is summarized here, along with its relationship to the underlying or related UML diagram type:

General Diagram Types

SysML contains one general diagram type that serves as a repository for requirements information.

1. Requirement Diagram

A requirements diagram represents text-based requirements and their relationship with other (e.g. non-functional) requirements, design elements, and test cases to support requirements traceability.

This diagram type is not present in UML.

Behavioral Diagram Types

Behavioral diagram types are used to capture the dynamic behavior of a system. They focus on changes over time, and the inputs and outputs that are required by and produced by transformations. The behavior can be centered on actions (in the case of activity diagrams), on participants (in the case of sequence diagrams), on rest states between actions (in the case of state machine diagrams), or in the composition of actions (in the case of use case diagrams). In essence, even though all four diagram types allow a modeler to describe changes over time, the different focal points result in different representations.

2. Activity Diagram

The SysML activity diagram represents behavior in terms of the ordering of actions based on the availability of inputs, outputs, and control, and shows how the actions transform inputs to outputs.

This diagram type is a modification of the UML Activity diagram.

3. Sequence Diagram

The SysML sequence diagram represents the behavior in terms of a sequence of messages exchanged between parts.

This diagram type is the same as a UML sequence diagram.

4. State Machine Diagram

The SysML state machine diagram represents behavior of an entity in terms of its transitions between states, where the state transitions are triggered by events.

This diagram type is the same as a UML state machine diagram.

5. Use Case Diagram

The SysML use case diagram represents functionality in terms of how a system or other entity is used by external entities (i.e. actors) in order to accomplish a set of goals.

This diagram type is the same as a UML use case diagram.

STRUCTURAL DIAGRAM TYPES

Structural diagram types are used to capture the components of a system and their relationships. They focus on the composition of the system elements and capture dependencies and interfaces.

6. Block Definition Diagram

The SysML block definition diagram represents structural elements called blocks, and allows for their composition and classification.

This diagram type is a modification of a UML class diagram.

7. Internal Block Diagram

The SysML block diagram represents interconnections and interfaces between the parts of a block (defined in the block definition diagram).

This diagram type is a modification of a UML composite structure diagram.

8. Parametric Diagram

The SysML parametric diagram represents constraints on property values, such as `F = m * a`, and is used to support engineering analysis.

This diagram type is not present in UML.

9. Package Diagram

The SysML package diagram represents the organization of a model in terms of packages that contain model elements.

This diagram type is the same as a UML package diagram.

MAPPING BETWEEN BPMN, SYSML, AND ARENA

In order to determine similarities between the models created by requirements engineers, software developers, and simulation engineers, shared concepts of the dominant modeling languages in each domain need to be identified and a cross-domain mapping needs to be established.

The comparison of modeling methods has a long history in Information Systems and Computer Science (see e.g. (Abowd, Allen, & Garlan, 1995)). For instance, a popular benchmark for the expressiveness of modeling languages is a mapping to the representation model of the *Bunge-Wand-Weber* ontology (Wand & Weber, 1993). This evaluation has been performed for process modeling languages ranging from ANSI Flowcharts to BPMN (Rosemann, Recker, Indulska, & Green, 2006), as well as for general systems analysis and design techniques (Opdahl & Henderson-Sellers, 2002).

In the context of this project, however, the BWW ontology provided a less than ideal benchmark, because it offers a very high level of abstraction from software engineering modeling constructs, and may this lead to an oversimplification of the language mappings. Instead, the research team decided to perform a point-to-point mapping of SysML activity diagrams, BPMN, and the proprietary Arena modeling language. Three members of the research team performed independent pairwise mappings, which were then consolidated. Each member represented specific subject matter expertise in one of the three languages to be mapped, in order to provide an anchor point for the comparison.

The high-level results of the mapping and subsequent analysis were as follows:

- *SysML activity diagrams* provide basic constructs to develop process models with organizational responsibility (swimlanes), decisions (gateways), and parallelism. These models are useful in closed environments, when the system under consideration operates without the influence of environmental factors.

- *Business Process Model & Notation* models provide a rich set of constructs for organizational processes, including extensive facilities for handling environmental events and rudimentary facilities for handling data (in form of persistent data stores and transient data objects that serve as hooks to data models that are described outside the actual BPMN model).
- *Arena* provides specialized facilities to describe operational processes, including the creation, batching, unbundling, binding, freeing, consumption and disposal of resources. Arena's strengths lie in the detailed description of resource behavior and capacity, something that both SysML activity diagrams and BPMN are lacking.

Overall, the SysML activity diagram modeling elements represented a subset of the BPMN modeling elements. The project team felt that in the presence of BPMN no good use case for the use of SysML activity diagrams exists, unless the architecture tool does not provide this facility. The relationship between BPMN and Arena is more nuanced. While BPMN allows for a process description with higher fidelity, the resource modeling capabilities of Arena are not matched by BPMN. To what extent BPMN models can serve as the basis for Arena simulation models required additional study, and was thus examined through a prototype that is described later in this document.

CAPABILITIES, ACTIVITIES, RESOURCES, PERFORMERS

Each architecture project should contain an integrated glossary (DoDAF 2.0 view AV-2). While in many projects this glossary is an afterthought, generated via a macro by the modeling software at the end of a design cycle, our research indicates that a core glossary at the start of a project can provide a common ground for project participants, and serve as a scoping device as much as a validation tool for the models that are subsequently generated. The four key concepts that should be defined at the outset of an architecture project are:

- Capabilities: What is the system designed to achieve? Understanding desired capabilities of the system allows architects and designers to consider and evaluate different system designs. Too often the design of a system is marked by a focus on a single way to achieve a desired objective, where alternative designs might provide the same capability easier/cheaper/more versatile. Focusing on capabilities first allows project participants to explore design alternatives. The resulting DoDAF models are CV-1 through CV-6.
- Activities: What needs to happen in order to provide the desired capabilities? Systems transform inputs into outputs, and activities are the steps that perform these transformations. Understanding the linkage between capabilities and activities allows a system architect to prioritize the development of certain processes, and developing a dependency chart

that shows how each system process contributes to the overall system objective. The resulting DoDAF models are OV-6c and SV-10c.

- Resources: What information is necessary to perform the required activities? Activities require inputs and produce outputs, and these are described in form of resources. Resources are described using the DoDAF views DIV-1 through DIV-3.
- Performers: Who is responsible for the performance of an activity or the provisioning of a desired capability? Identifying performers helps system designers develop concepts for access rights, organizational support, notification systems, behavioral incentives etc.

PHASE 2: REQUIREMENTS & COMMON CORE

Phase 2 of the research task focused on the modeling views that were required for system documentation, and useful for system design. Starting with the full set of architecture views of the DoDAF 2.0 framework (U.S. Department of Defense, 2009) the team proceeded with a two-pronged approach. In the first approach we created architecture documentation for a medical training simulator using all available DoDAF 2.0 views in order to determine which of these views the architects perceived as useful. In the second approach, we mapped the different model types of SysML to the DoDAF 2.0 views in order to determine to what extent formal models with SysML would cover the DoDAF 2.0 views. Finally, we compared our findings with the DoDAF views required by the JCIDS process, in order to arrive at a recommended development sequence for architecture views.

DETERMINING USEFUL DODAF VIEWS

The DoDAF 2.0 framework contains more than 50 views that can be used to capture system architecture information. For the design of software-intensive systems not all of these views are of equal importance. The development of the CIMT case study models provided the basis for a qualitative assessment of DoDAF 2.0 views.

Figure 3 shows an overview of the architecture views designed as part of the CIMT case study. All views were designed using Word, Excel, UML/SysML, and in case of the SV-2 view a Python script that is documented in the appendix of this report.

CIMT is a training simulator for combat medics. Development of CIMT began with a description of the desired system capabilities (in the Capability Views), an integrated glossary (AV-2), followed by software-centric views, such as data models, process and state charts, as well as system rule descriptions.

Figure 3: CIMT DoDAF 2.0 Views (Overview)

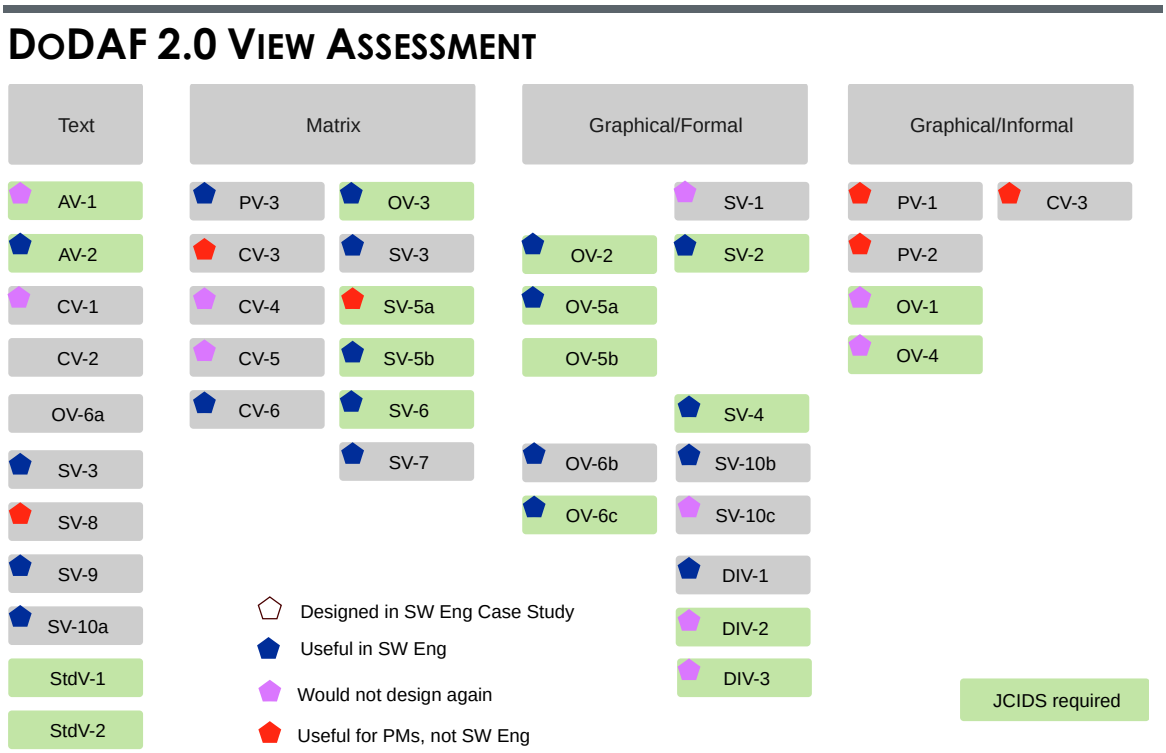

Figure 4: DoDAF 2.0 Views

Figure 4 shows an overview of architecture views as defined in the DoDAF 2.0 specification (U.S. Department of Defense, 2009).[1] Views that are required by the JCIDS process are marked in green. Views marked with a pentagon were created as part of the CIMS prototype. The different DoDAF views were assessed by the project team and classified in two areas:

- Views that were regarded as useful for software-intensive systems design are marked in blue. These views were either regarded as useful for requirements capture, or useful as a precursor for code generation. Views ranked as useful were: AV-2, DIV-1, PV-3, CV-6, OV-2, OV-3, OV-5a, OV-6b, OV-6c, SV-2, SV-3, SV-4, SV-5b, SV-6, SV-7, SV-9, SV-10a, SV-10b.
- Views that were regarded as useful for project management purposes, but not for the software engineers are marked in red. These views are: PV-1, PV-2, CV-3, SV-5a, SV-8.

[1] Note that due to the overlapping definitions the Systems Views (SVs) and Service Views (SvcVs) are collapsed into System Views. At the time of this writing the only difference between System and Service views is the emphasis of service-orientation in the SvcVs. We expect that future releases of DoDAF will replace the notion of System Views with Service Views. Our study results remain unaffected by this change.

A number of JCIDS-required views were found to be either duplicate, or not essential for software engineering purposes. These include AV-1, DIV-2, DIV-3, OV-1, OV-4.

PHASE 3: PROOF OF CONCEPT

The research task concluded with two implementation components. One component was the prototypical implementation of a converter from BPMN 2.0 to Arena, in order to assess to what extent process models created in BPMN could be reused as simulation models in Arena. The other component consisted of a fictitious news processing use case that was documented using DoDAF architecture views and implemented in both a simulation model and a workflow implementation.

BPMN 2.0 XML TO ARENA CONVERTER

After the language mapping in Phase 1 the team determined that both BPMN 2.0 and Arena had their respective strengths and weaknesses. A (semi-)automatic transformation of BPMN process models into Arena simulation models would minimize the duplication of model content between software architects and simulation engineers.

The BPMN 2.0 specification describes a uniform storage format for BPMN 2.0 models in form of an XML schema. The BPMN 2.0 XML schema is split in two components. The content area of the schema describes the components and flow logic of the business process (e.g. activities, sequence flows, gateways, events), while the diagram interchange area of the schema records the layout information of the process model itself (the x/y coordinates of graphical model elements, as well as their size and scaling). Not all BPMN tools implement this schema the same way. Some tools (e.g. IBM Blueworks Live and Rational System Architect) do not preserve the diagram information, but they do export the semantic content of the model. Figure 5 shows an excerpt from a BPMN 2.0 XML file that represents the start event of a process.

```
<startEvent id="sid-192C9BDE-5024-40E3-89D3-F50F93DF485E" name="">
    <extensionElements>
        <signavio:signavioMetaData metaKey="bgcolor" metaValue="#ffffff"/>
    </extensionElements>
    <outgoing>sid-340B6FDD-37E3-4ECA-8EC4-CBD89A795134</outgoing>
</startEvent>
```

Figure 5: BPMN 2.0 XML StartEvent (Excerpt)

Arena does not provide an import facility for XML. Instead, simulation models can be imported using either a Microsoft Access MDB schema, or a Microsoft Excel XLS file. The XLS file is split into 9 separate sheets: ModuleTables, RepeatGroup Tables, ModelLevels, Submodels, Connections, NamedViews, ProjectParameters, ReplicationParameters and Reports. Additional sheets are needed depending on the elements of the imported process, for instance for create, process, decision, and dispose nodes. Figure 6 shows the Arena XLS import file sheet that represents the start event listed in the previous figure.

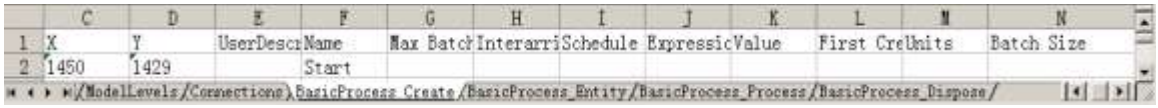

Figure 6: Arena Start Event in XLS file

The BPMN 2.0 XML to Arena converter reads a BPMN 2.0 XML file and parses it using a Java DOM construct. The convertible model elements are then written into an XLS file, which can be imported by Arena.

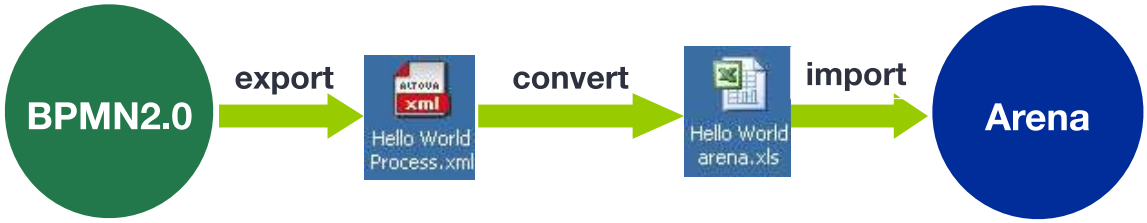

Figure 7: BPMN 2.0 to Arena conversion process

The resulting converter can import BPMN 2.0 processes that include XOR gateways (decision points) in processes. A simulation engineer can thus use a BPMN model as input for the construction of a more complex simulation model. The simulation engineer still needs to add resource capacities, arrival rates, cycle times and distributions, as well as branching probabilities. While BPMN 2.0 could be extended to accommodate these parameters, the resulting BPMN 2.0 XML would go beyond what the OMG standard specifies, hence the research team refrained from implementing proprietary extensions.

NEWS CASE DESCRIPTION

In order to illustrate the feasibility of the proposed model development process, the team developed a proof of concept in form of a news use case. This use case was documented in a select number of DoDAF views. The behavioral aspects of the news use case were then transferred into different development environments to illustrate the feasibility of a top-down development approach.

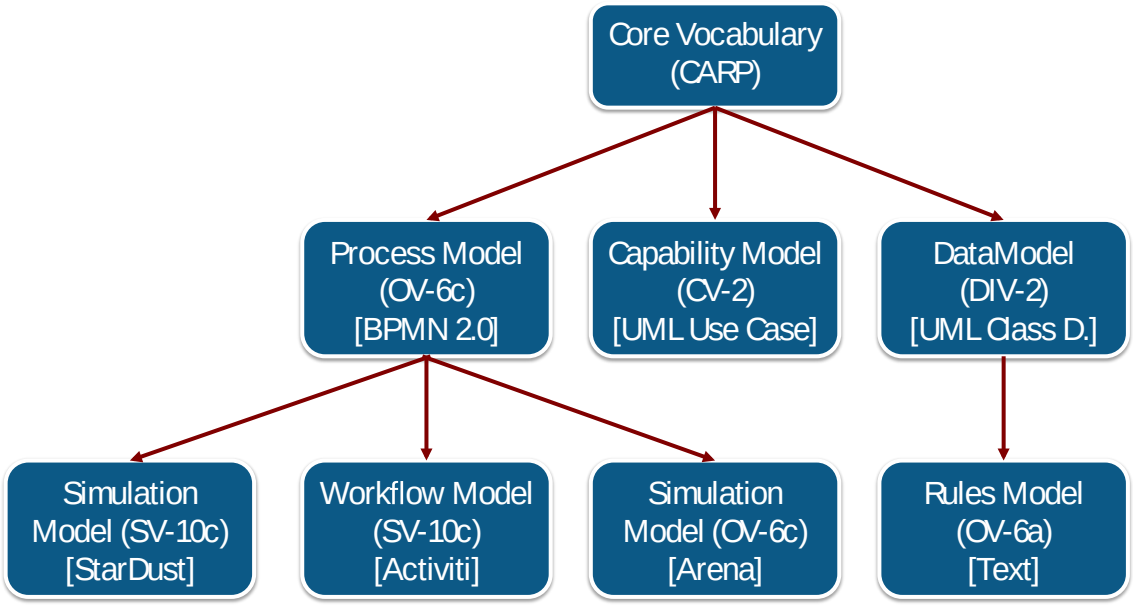

Figure 8: Model Transfer for News Use Case

The development of the use case followed the recommended sequence of Capabilities, Activities, Resources, and Performers. As a first step, a glossary of terms was constructed using an Excel template. Using this template each term in the glossary could be classified as a Capability, Activity, Resource, or Performer. Using the mapping table provided by the DoDAF 2.0 specification, a cross-reference to the DoDAF views referencing each of the four constructs was created, allowing an architect a quick assessment a) which architecture views might contain a newly defined concept, and b) given an architecture model, whether its terms were defined in the glossary. Figure 9 shows a screenshot of the Excel table.

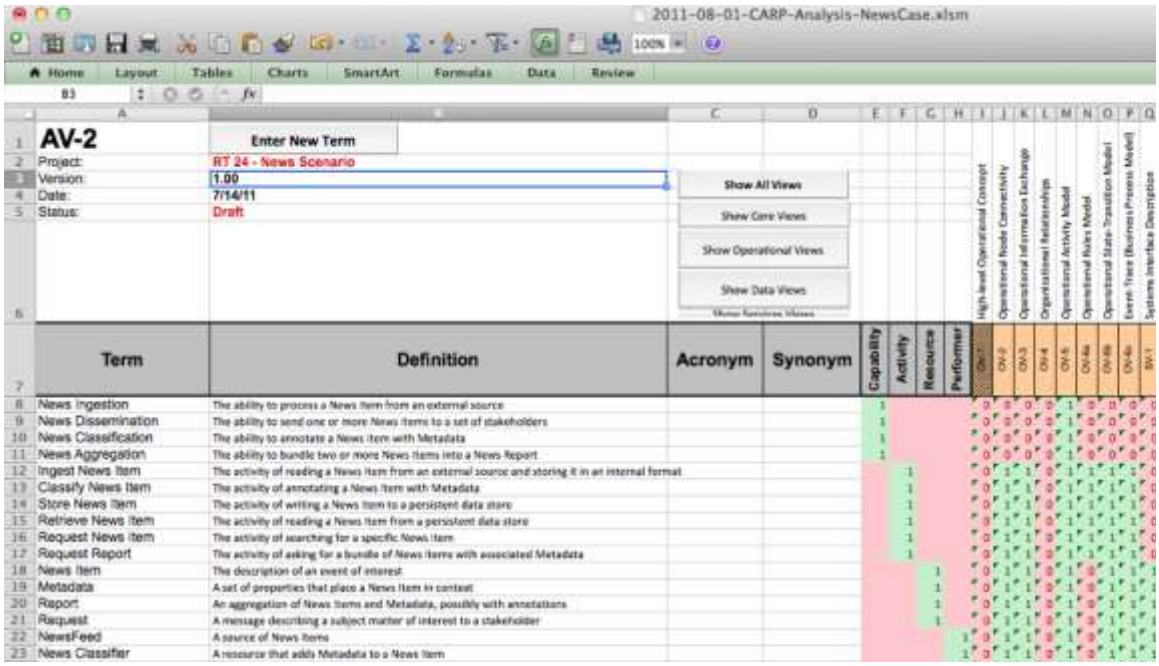

Figure 9: CARP vocabulary/glossary for News Use Case

Based on the glossary the first developed architecture view of the case study was the Capability Overview (CV-2). The team chose a UML/SysML Use Case diagram as the appropriate representation. The use case diagram contained both participants (actors) and activities (processes) defined in the AV-2.

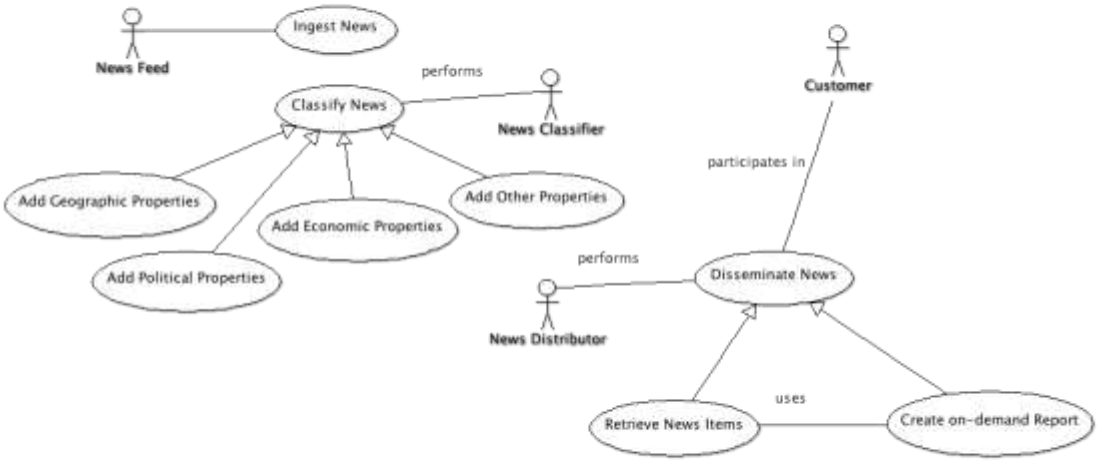

Figure 10: CV-2 (Use Case) of News Use Case

The use case diagram allowed the team to extend the concepts defined in the AV-2 with further sub-processes and actors. The next step in the architecture development was the design of the concept of operations (OV-1). The team chose a BPMN 2.0 high-level process diagram for this purpose.

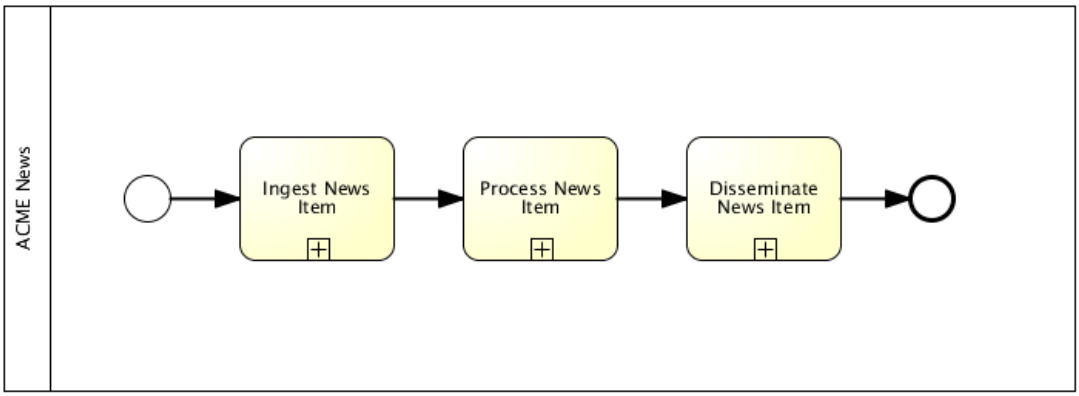

Figure 11: News Use Case Milestone Process Diagram (OV-1)

The milestone diagram depicted in Figure 11 shows the major phases of the system – news ingestion, news processing, and news dissemination. It does not include decisions or detailed performers, but through the capabilities of most process modeling tools each of the three steps could be linked to a more detailed representation at the sub-process-level.

Following the OV-1, a number of detailed process descriptions were developed, namely the OV-6c diagrams for Ingest News Item, Process News Item, and Disseminate News Item. These diagrams add the detailed roles (which had been defined as performers in the AV-2), data objects and data stores (which had been defined as resources), and branching gateways as well as decisions (which were new). In this phase, the team iterated between the process representations and the AV-2. In practice, a software architect would add elements to the AV-2 table in order to maintain consistency.

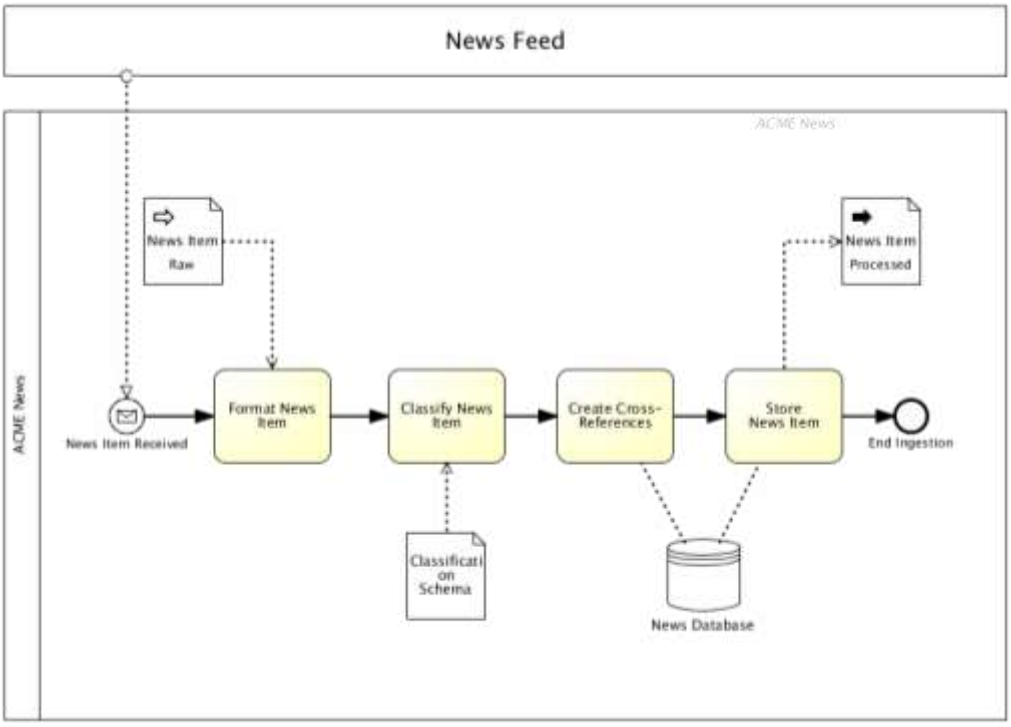

Figure 12: News Use Case News Ingestion Process Diagram (OV-6c)

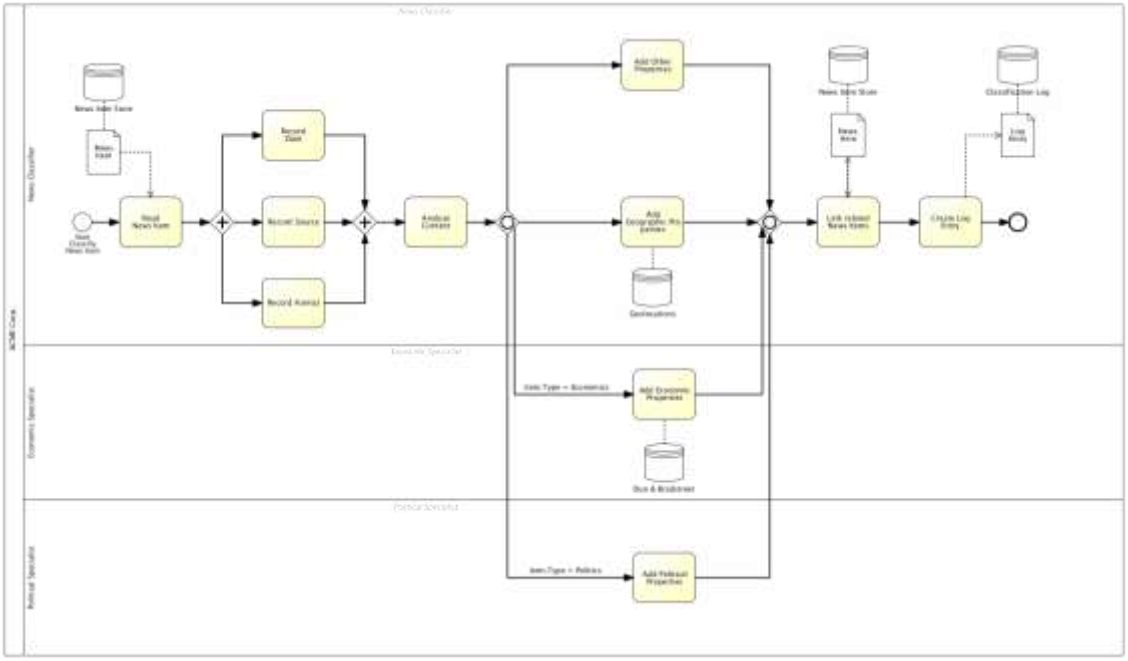

Figure 13: News Use Case News Classification Process (OV-6c)

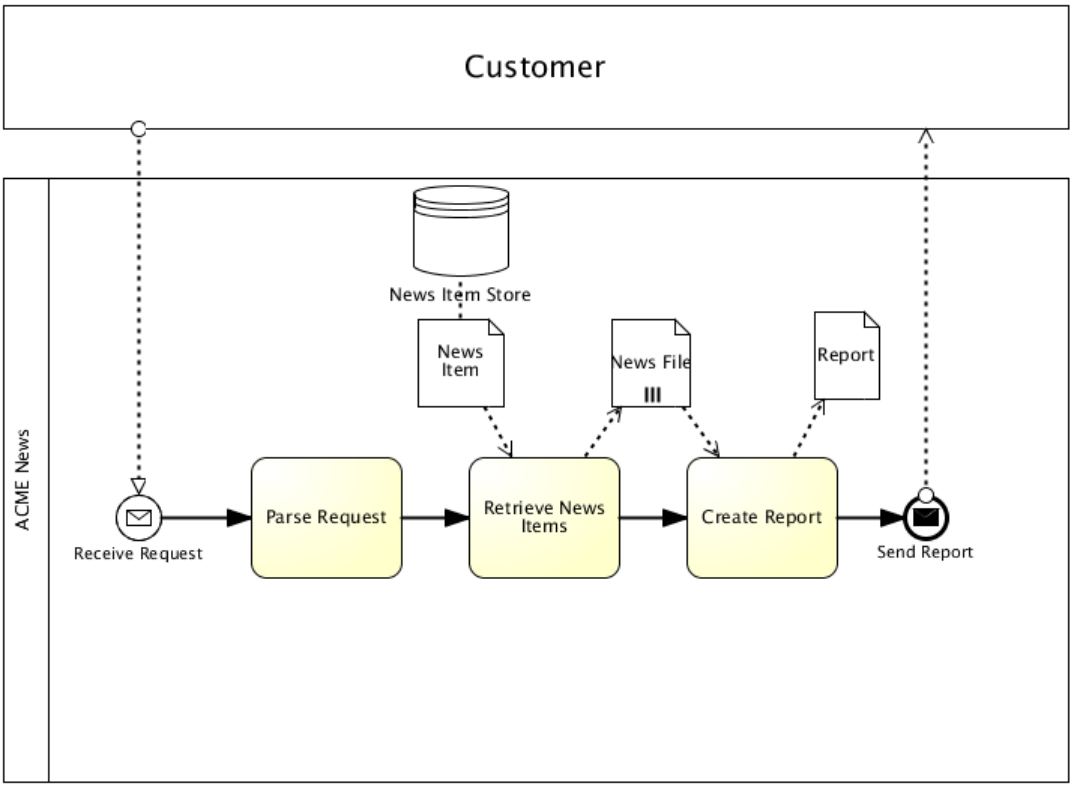

Figure 14: News Use Case News Dissemination (On Demand) (OV-6c)

It became apparent in the design of the OV-6c models that additional detail in terms of data descriptions was necessary to make the process models executable. In response an integrated data model (DIV-2) was developed using a UML class diagram (SysML Block Diagram).

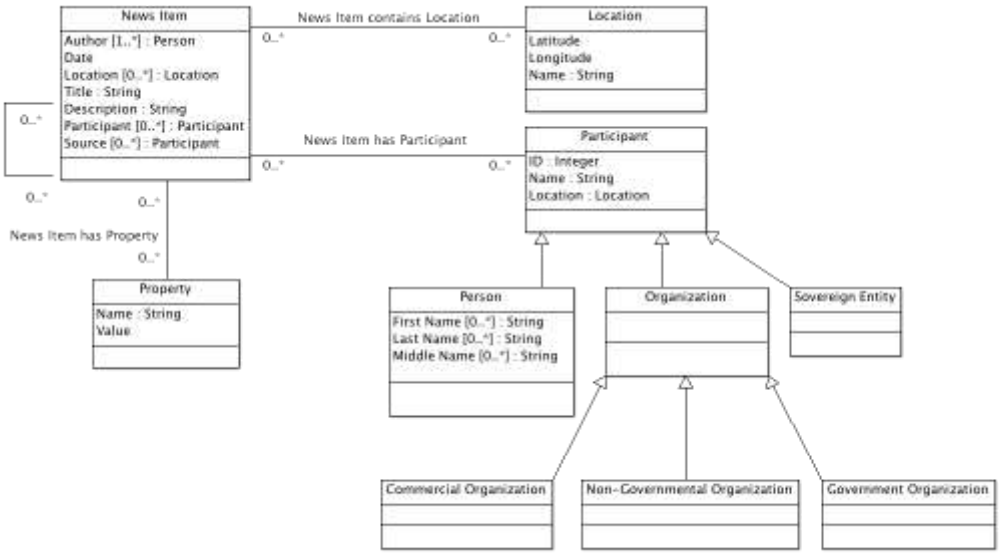

Figure 15: News Case Conceptual Data Model

The creation of these requirements-centric models illustrated the applicability of the Capability-oriented modeling strategy – at each stage of the development process the models maintained consistency with the central glossary, and the modeling tasks were supported by the availability of key (anchor) terms that represented key modeling elements.

NEWS CASE IMPLEMENTATION

In order to take the news case from concept to implementation, two prototypical, process-oriented implementations were created, both based on free open source software platforms.

The first implementation was based on the Eclipse Stardust Business Process Management System. Stardust was chosen because it has an industry-strength simulation component and thus could serve both as a workflow and simulation showcase.

While Stardust supports BPMN as the visual modeling style for processes, its data format is based on the older XML Process Definition Language (XPDL), which required a re-modeling of the existing BPMN diagrams. While this remodeling created moderate effort, it illustrated that processes at the workflow level require additional steps that are typically not captured at the requirements engineering level. Figure 16 illustrates one of the processes in Eclipse Stardust. Some differences to the BPMN process in the previous section are the existence of an additional task (Store News Item), that was not part of the abstract process representation, as well as the distinction between “Raw” news items and processed news items. While the high-level process only considers one type of news item, the workflow level has to consider transformations of data formats as

well as detailed mappings between the data items and the tasks that process them.

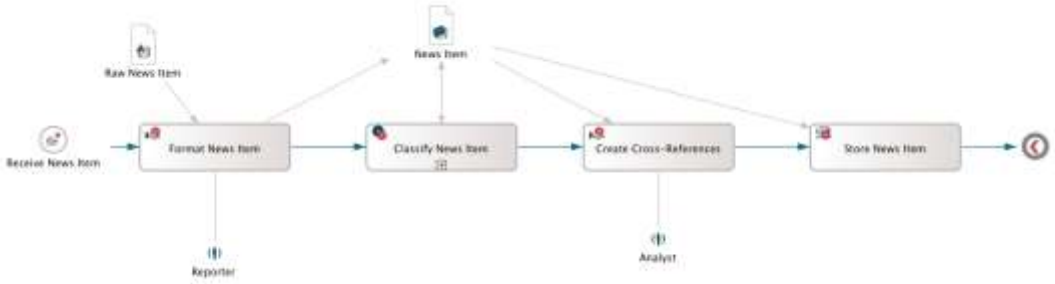

Figure 16: News Use Case implemented in Eclipse Stardust

A side effect of the StarDust implementation was the creation of a formal organization model (OV-4), which was required for proper workflow execution.

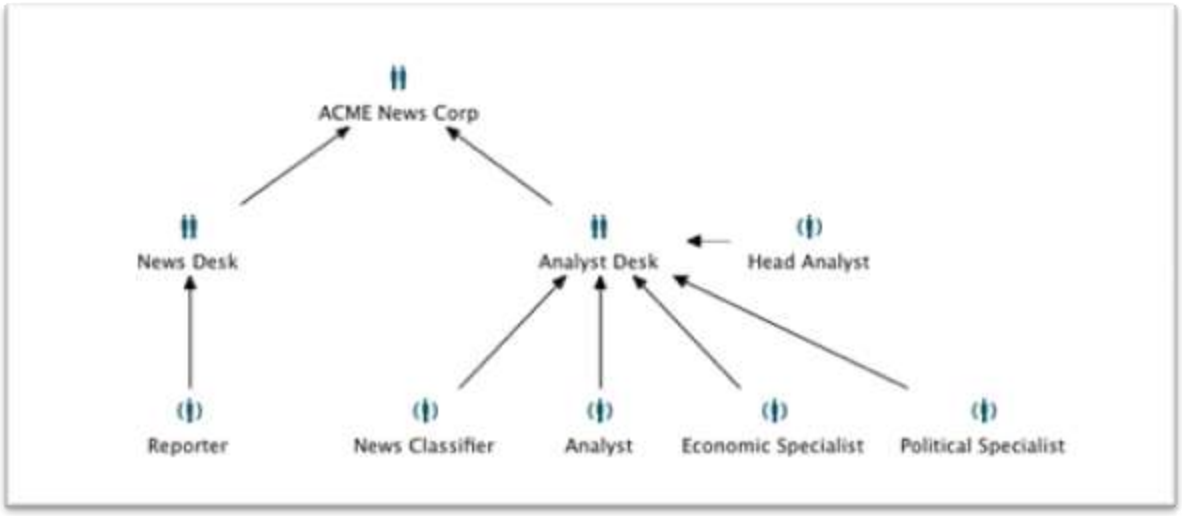

Figure 17: News Use Case Organization Model

In order to simulate the news use case, additional information about the simulation scenario was added to the model. This information was not captured during the requirements gathering phase, but would need to be added for a simulation run independent of the platform used. The questions asked during the creation of a simulation scenario proved to be excellent catalysts for a conversation between stakeholders, modelers and implementers.

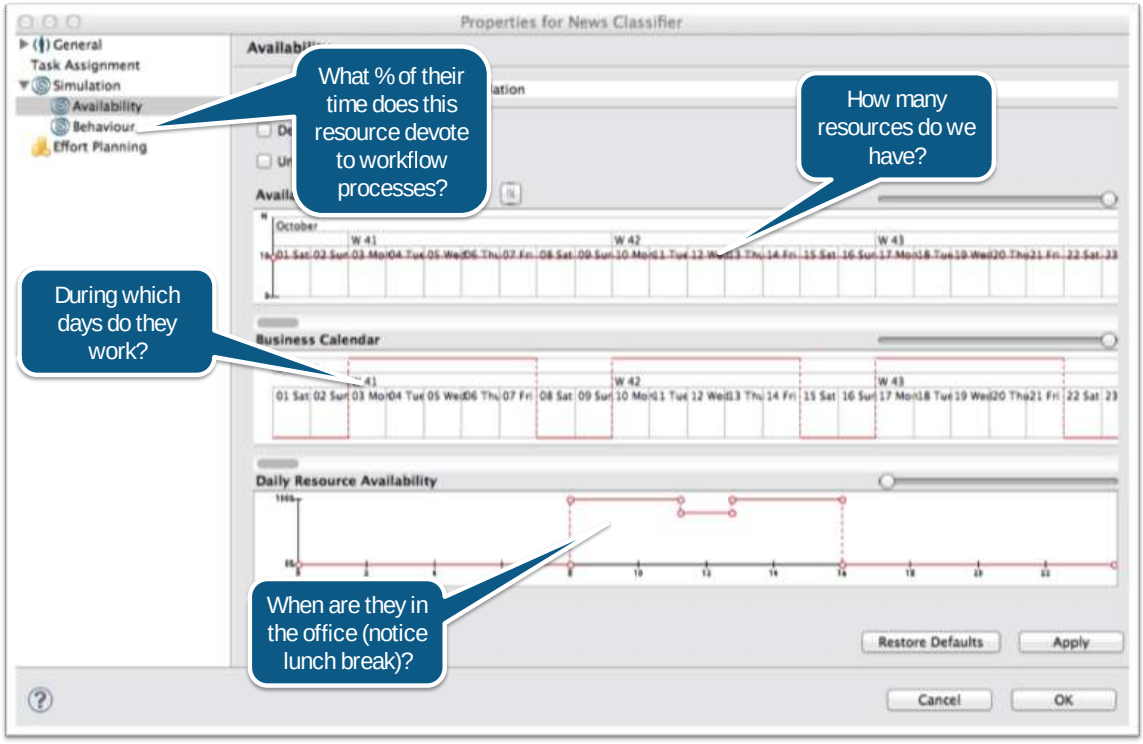

Figure 18: Sample Questions about Resources for Simulation Purposes

The second implementation was based on the Alfresco Activity Business Process Management System. Activiti was chosen because it ingests the native BPMN 2.0 XML format, which meant that the process models implemented in the previous phase could be imported directly into the Activiti design environment.

Importing the BPMN 2.0 XML into Activiti did not create any issues. However, in order to create an executable workflow, additional modifications to the XML code were necessary. In particular, data persistency required the manual creation of a JDBC bridge class to store the received news items for further processing. Furthermore, creating form fields for user-supported tasks required the manual addition of “formProperty” fields to the XML code, as illustrated in Figure 19.

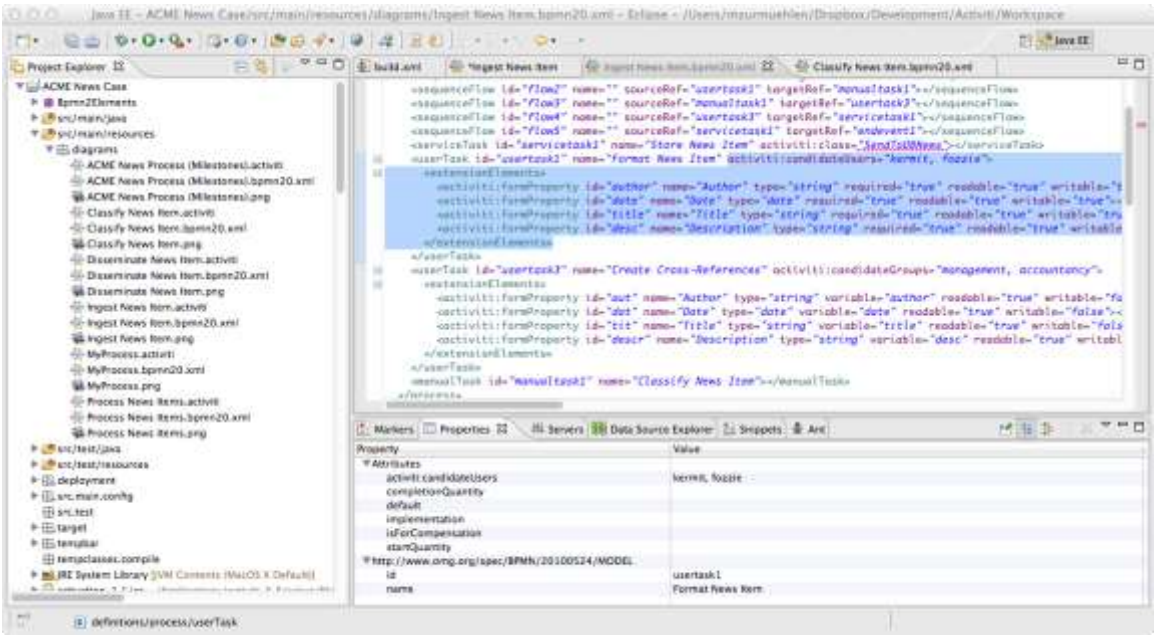

Figure 19: Adding "formProperty" fields to the BPMN XML in Activiti

OUTLOOK

The research task successfully demonstrated that the development of software-intensive systems based on a small set of core models is feasible. In particular, the use of a central glossary for architecture and design activities proved highly useful. The DoDAF AV-2 should therefore be one of the first architecture views to be developed, and not remain an automatically generated afterthought.

Creating requisite process and data models, as well as use case descriptions can facilitate the transition from requirements engineering to simulation, and implementation. The transition from design to implementation, however, is not seamless. Implementation models require a higher degree of fidelity in terms of data description, service interfaces, and user interfaces.

The reliance on individual models (e.g. BPMN) allows a designer to capture some of this information, but not all of it. Differences in tool-specific standard implementations further hamper the seamless transition of model information. Nevertheless, with the prototype converter from BPMN to Arena we were able to demonstrate that much information can be salvaged without requiring a re-creation of information.

Future work should investigate whether the language-independent mappings of model concepts from the AV-2 into different modeling languages could provide a step toward a more seamless integration of requirements engineering, software engineering, and modeling & simulation. The DoDAF 2.0 Meta Model is intended to provide a semantic model for architecture information. Its application in practice is still unproven, and should be subject to further study.

APPENDICES

APPENDIX A: SYSML 1.1 TO DODAF V2.0 MAPPING (SUMMARY)

The purpose of this section is to understand the details of each of the DoDAF viewpoints (eight viewpoints further refined to 52 models) from DoDAF's perspective and to provide a mapping of where SysML, a general purpose graphical modeling language, fits within the realm of any given view (52 models).

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
AV-1	Overview and Summary Information	Describes a Project's Visions, Goals, Objectives, Plans, Activities, Events, Conditions, Measures, Effects (Outcomes), and produced objects.	None (Structured textual format)	No equivalent exists. Create a structured text document with the specified information.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
AV-2	Integrated Dictionary	An architectural data repository with definitions of all terms used throughout the architectural data and presentations.	None	No equivalent exists. Any element created within any of the architecture models should be clearly defined as per the minimum taxonomy information guideline. This data can be extracted via a SysML generated report.
CV-1	Vision	The overall vision for transformational endeavors, which provides a strategic context for the capabilities described and a high-level scope.	None (Textual descriptions)	No equivalent exists. Create a structured text document with the specified information (goals, desired outcomes, measureable benefits).
CV-2	Capability Taxonomy	A hierarchy of capabilities which specifies all the capabilities that are referenced throughout one or more Architectural Descriptions.	None. Selection must support the representation of a structured/hierarchal list; and may be textual, tabular, or graphical.	Block Definition Diagram Develop a supporting table that can be exported from SysML detailing the associated attributes and measures.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
CV-3	Capability Phasing	The planned achievement of capability at different points in time or during specific periods of time. The CV-3 shows the capability phasing in terms of the activities, conditions, desired effects, rules complied with, resource consumption and production, and measures, without regard to the performer and location solutions.	None (Structured tabular format).	No equivalent exists. Create a table with the specified information - rows representing Capabilities (derived from the CV-2 Capability Taxonomy model) and columns representing phases (from CV-1 Vision model).
CV-4	Capability Dependencies	The dependencies between planned capabilities and the definition of logical groupings of capabilities.	None (Graphical approach).	Block Definition Diagram
CV-5	Capability to Organizational Development Mapping	The fulfillment of capability requirements shows the planned capability deployment and interconnection for a particular Capability Phase. The CV-5 shows the planned solution for the phase in terms of performers and locations and their associated concepts.	None (Structured tabular format).	No equivalent exists. Create a table with the specified information - appropriate organizational structure represented by one axis, and the capabilities by the other axis.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
CV-6	Capability to Operational Activities Mapping	A mapping between the capabilities required and the operational activities that those capabilities support.	None (Matrix/Tabular approach)	No equivalent exists. Create a table with the specified information - rows representing the Capabilities and the columns representing the Operational Activities.
CV-7	Capability to Services Mapping	A mapping between the capabilities and the services that these capabilities enable.	None (Matrix/Tabular approach)	No equivalent exists. Create a table with the specified information - rows representing the Capabilities and the columns representing the Operational Activities.
DIV-1	Conceptual Data Model	The required high-level data concepts and their relationships.	None.	Block Definition Diagram
DIV-2	Logical Data Model	The documentation of the data requirements and structural business process (activity) rules. In DoDAF V1.5, this was the OV-7.	Class and/or Object Diagrams	Block Definition Diagram
DIV-3	Physical Data Model	The physical implementation format of the Logical Data Model entities, e.g., message formats, file structures, physical schema. In DoDAF V1.5, this was the SV-11.	Class and/or Object Diagrams	Block Definition Diagram

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
OV-1	High-Level Operational Concept Graphic	The high-level graphical/textual description of the operational concept.	None (consists of a graphical executive summary for a given Architectural Description with accompanying text)	Use Case Diagram XML report (use case descriptions) ***Support Artifacts:*** Text Document. Graphic (Powerpoint, Paint, …etc)
OV-2	Operational Resource Flow Description	A description of the Resource Flows exchanged between operational activities.	None	Block Definition Diagram
OV-3	Operational Resource Flow Matrix	A description of the resources exchanged and the relevant attributes of the exchanges.	None (Matrix development required – tabular format)	No equivalent diagram. Matrix to be developed using SysML tool and OV-2 model.
OV-4	Organizational Relationships Chart	The organizational context, role or other relationships among organizations.	None	Block Definition Diagram
OV-5a	Operational Activity Decomposition Tree	The capabilities and activities (operational activities) organized in a hierarchal structure.	None	Block Definition Diagram
OV-5b	Operational Activity Model	The context of capabilities and activities (operational activities) and their relationships among activities, inputs, and outputs; Additional data can show cost, performers, or other pertinent information.	Integration Definition for Function Modeling (IDEF0) or Class Diagrams	Activity Diagram

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
OV-6a	Operational Rules Model	One of three models used to describe activity (operational activity). It identifies business rules that constrain operations.	None (The OV-6a should be presented in a textual format in the English language.)	No equivalent diagram. A model (table) is to be developed that will feed into other models (mainly OV) created using the SysML tool. The OV-6a model should be created using a format that can easily be imported to the SysML tool e.g. XML. OV-6a should be traceable from OV-1.
OV-6b	State Transition Description	One of three models used to describe operational activity (activity). It identifies business process (activity) responses to events (usually, very short activities).	Statechart diagram	State Machine Diagram
OV-6c	Event-Trace Description	One of three models used to describe activity (operational activity). It traces actions in a scenario or sequence of events.	Any modeling notation (e.g., BPMN) that supports the layout of timing and sequence of activities along with the Resource Flow exchanges that occur between Operational Activities/Locations for a given scenario.	Sequence Diagram and/or Activity Diagram

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
PV-1	Project Portfolio Relationships	It describes the dependency relationships between the organizations and projects and the organizational structures needed to manage a portfolio of projects.	None.	Block Definition Diagram
PV-2	Project Timelines	A timeline perspective on programs or projects, with the key milestones and interdependencies.	None (Gantt Chart)	No equivalent diagram. Gantt Chart to be developed.
PV-3	Project to Capability Mapping	A mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability.	None (Matrix / Tabular representation)	No equivalent diagram. Matrix to be developed.
SvcV-1	Services Context Description	The identification of services, service items, and their interconnections.	None	Block Definition Diagram and/or Internal Block Diagram
SvcV-2	Services Resource Flow Description	A description of Resource Flows exchanged between services.	None	Block Definition Diagram
SvcV-3a	Systems-Services Matrix	The relationships among or between systems and services in a given Architectural Description.	None (Matrix development required)	No equivalent diagram. Matrix to be developed using SysML tool and SvcV-1 model.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SvcV-3b	Services-Services Matrix	The relationships among services in a given Architectural Description. It can be designed to show relationships of interest, (e.g., service-type interfaces, planned vs. existing interfaces).	None (Matrix development required)	No equivalent diagram. Matrix to be developed using SysML tool and SvcV-1 model. Matrix provides input to SvcV-10a, SvcV-10b, and SvcV-10c.
SvcV-4	Services Functionality Description	The functions performed by services and the service data flows among service functions (activities).	Taxonomic Service Functional Hierarchy and/or Data Flow Diagram	Block Definition Diagram and/or Internal Block Diagram
SvcV-5	Operational Activity to Services Traceability Matrix	A mapping of services (activities) back to operational activities (activities).	None (Matrix development required)	No equivalent diagram. Matrix to be developed using the SysML tool, the OV-5a and OV-5b model diagrams, and the SvcV-4 model diagram.
SvcV-6	Services Resource Flow Matrix	It provides details of service Resource Flow elements being exchanged between services and the attributes of that exchange.	None (Matrix development required, traceability needed back to OV-2 and OV-3)	No equivalent diagram. Matrix to be developed using SysML tool, OV-3 model, and SV-4 model, complete with traceability back to OV-2 and OV-3.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SvcV-7	Services Measures Matrix	The measures (metrics) of Services Model elements for the appropriate time frame(s).	None (Matrix is typically a table listing user defined measures (metrics) with a time period association.)	No equivalent diagram. Matrix is to be developed using the SysML tool and the SvcV-1 model diagram.
SvcV-8	Services Evolution Description	The planned incremental steps toward migrating a suite of services to a more efficient suite or toward evolving current services to a future implementation.	None (Graphical timeline)	No equivalent diagram. An evolutionary timeline (graphical accompanied by a textual description) is to be developed. It should detail the structure of each resource, using similar modeling elements as those used in SvcV-1. Interactions which take place within the resource may also be shown.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SvcV-9	Services Technology & Skills Forecast	The emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future service development.	Can be presented in a table, timeline, or a Herringbone diagram.	No equivalent diagram. An evolutionary timeline (with a tabular, timeline, or a herringbone diagram format) is to be developed. New technologies and skills are tied to specific time periods, which can correlate against the time periods used in SvcV-8 milestones and capability phases.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SvcV-10a	Services Rules Model	One of three models used to describe service functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.	None (The SvcV-10a should provide a listing of the complete set of rules with a reference to any models that they affect.)	No equivalent diagram. A model (table) is to be developed that will feed into other models created using the SysML tool. The SvcV-10a should provide a listing of the complete set of rules with a reference to any models that they affect. The SvcV-10a model should be created using a format that can easily be imported to the SysML tool e.g. XML.
SvcV-10b	Services State Transition Description	One of three models used to describe service functionality. It identifies responses of services to events.	Statechart Diagram	State Machine Diagram
SvcV-10c	Services Event-Trace Description	One of three models used to describe service functionality. It identifies service-specific refinements of critical sequences of events described in the Operational Viewpoint.	Sequence Diagram	Sequence Diagram

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
StdV-1	Standards Profile	The listing of standards that apply to solution elements.	None	No equivalent exists. Create a structured text document.
StdV-2	Standards Forecast	The description of emerging standards and potential impact on current solution elements, within a set of time frames.	None	No equivalent exists. Create a structured text document.
SV-1	Systems Interface Description	The identification of systems, system items, and their interconnections.	None	Block Definition Diagram and/or Internal Block Diagram
SV-2	Systems Resource Flow Description	A description of Resource Flows exchanged between systems.	None	Block Definition Diagram
SV-3	Systems-Systems Matrix	The relationships among systems in a given Architectural Description. It can be designed to show relationships of interest, (e.g., system-type interfaces, planned vs. existing interfaces).	None (Matrix development required)	No equivalent diagram. Matrix to be developed using SysML tool and SV-1 model.
SV-4	Systems Functionality Description	The functions (activities) performed by systems and the system data flows among system functions (activities).	Taxonomic Service Functional Hierarchy and/or Data Flow Diagram	Block Definition Diagram and/or Internal Block Diagram

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SV-5a	Operational Activity to Systems Function Traceability Matrix	A mapping of system functions (activities) back to operational activities (activities).	None (Matrix development required)	No equivalent diagram. Matrix to be developed using the SysML tool, the OV-5a model diagram, and the SV-4 model diagram.
SV-5b	Operational Activity to Systems Traceability Matrix	A mapping of systems back to capabilities or operational activities (activities).	None (Matrix development required)	No equivalent diagram. Matrix to be developed using the SysML tool, the OV-5a and OV-5b model diagrams, and the SV-1 model diagram.
SV-6	Systems Resource Flow Matrix	Provides details of system resource flow elements being exchanged between systems and the attributes of that exchange.	None (Matrix development required, traceability needed back to OV-2 and OV-3)	No equivalent diagram. Matrix to be developed using SysML tool and SV-4 model with focus on data flows across boundaries only, complete with traceability back to OV-2 and OV-3.
SV-7	Systems Measures Matrix	The measures (metrics) of Systems Model elements for the appropriate timeframe(s).	None (Matrix is typically a table listing user defined measures (metrics) with a time period association.)	No equivalent diagram. Matrix is to be developed using the SysML tool and the SV-1 model diagram.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SV-8	Systems Evolution Description	The planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation.	None (Graphical timeline)	No equivalent diagram. An evolutionary timeline (graphical accompanied by a textual description) is to be developed. It should detail the structure of each resource, using similar modeling elements as those used in SV-1. Interactions which take place within the resource may also be shown.
SV-9	Systems Technology & Skills Forecast	The emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future system development.	Can be presented in a table, timeline, or a Herringbone diagram.	No equivalent diagram. An evolutionary timeline (with a tabular, timeline, or a herringbone diagram format) is to be developed. New technologies and skills are tied to specific time periods, which can correlate against the time periods used in SV-8 milestone.

DoDAF v2.0 Model	DoDAF v2.0 Model: Name	DoDAF v2.0 Model: Brief Description	DoDAF Recommended Diagram / Tool	SIT Recommended SysML 1.x Diagram
SV-10a	Systems Rules Model	One of three models used to describe system functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.	None (The SV-10a should provide a listing of the complete set of rules with a reference to any models that they affect.)	No equivalent diagram. A model (table) is to be developed that will feed into other models created using the SysML tool. The SV-10a should provide a listing of the complete set of rules with a reference to any affected models. The SV-10a model should be created using a format that can easily be imported to the SysML tool e.g. XML
SV-10b	Systems State Transition Description	One of three models used to describe system functionality. It identifies responses of systems to events.	Statechart Diagram	State Machine Diagram
SV-10c	Systems Event-Trace Description	One of three models used to describe system functionality. It identifies system-specific refinements of critical sequences of events described in the Operational Viewpoint.	Sequence Diagram	Sequence Diagram

Table 1: Summary – DoDAF V2.0 Mapping to SysML

APPENDIX B: SYSML 1.1 TO DODAF 2.0 MAPPING (DETAILED)

ALL VIEWPOINT

The All Viewpoint (AV) details the overarching aspects of architecture context that relate to all views. The AV DoDAF-described models capture the scope of the architecture and where the architecture fits in relationship to other architectures. Another use of the All Viewpoint is for the registration of the architecture to support the net-centric goals of making architectural descriptions visible (discoverable).

AV-1 Overview and Summary Information

The AV-1 DoDAF-described model describes a project's visions, goals, objectives, plans, activities, events, conditions, measures, effects (outcomes), and produced objects. More specifically, the overview and summary information contained within the AV-1 provides executive-level summary information in a consistent form that allows quick reference and comparison between Architectural Descriptions. The written content of the AV-1 content describes the concepts contained in the pictorial representation of the OV-1. Each Architectural Description has a rationale that governs the selection of the Models used and the scope of the underlying models as a result of employing the 6-Step Architecture Development Process. The AV-1 DoDAF-described Model is intended to describe the decisions made throughout that process.

In the case of the AV-1 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. It is suggested however, that the AV-1 model should be created in a structured textual format.

AV-1:

- There is no equivalent / applicable SysML diagram that can help in the creation of the AV-1 DoDAF described model.
- The AV-1 model should be created in a structured textual format and it

should document the following descriptions: Architectural Description Identification; Scope; Purpose and perspective; Context; Status; Tools and File Formats Used; Assumptions and Constraints; and, Archtecture development schedule including start date, development milestones, date completed, and other key dates.

Additional Information:

The intended usage of the AV-1 DoDAF-described model:

- Scope the architecture effort
- Provide context to the architecture effort.
- Define the architecture effort
- Summarize the findings from the architecture effort.
- Assist search within an architecture repository.

AV-2 Integrated Dictionary

The AV-2 DoDAF-described model presents an architectural data repository with definitions of all terms used throughout the architectural data and presentations. More specifically, the AV-2 presents all the metadata used in an architecture. An AV-2 presents all the data as a hierarchy, provides a text definition for each one and references the source of the element (e.g., DoDAF Meta-model, IDEAS, a published document or policy). Data elements need to be uniquely identified and consistently used across all viewpoints, models and views within the Architectural Description. These populated views should include notes on any unique definitions used and provide a mapping to standard definitions, where possible.

In the case of the AV-2 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. No guidance is provided for the construction of AV-2. It does however indicate the minimum requirements for the taxonomies of capabilities, resource flows, activities, performance parameters, performers, skills, standards, and triggers/events.

AV-2:

- There is no equivalent / applicable SysML diagram that can help in the creation of the AV-2 DoDAF described model.
- Any element created within any of the architecture models should be clearly defined as per the minimum taxonomy information guideline. This data can then be extracted via a SysML generated report for the purpose of generating the AV-2 model.

CAPABILITY VIEWPOINT

The capability viewpoint articulates the capability requirement, delivery timing, and deployed capability. More specifically, the Capability Models describe capability taxonomy and capability evolution. The Capability Models included within DoDAF are based on the program and capability information used by Portfolio Managers to capture the increasingly complex relationships between interdependent projects and capabilities.

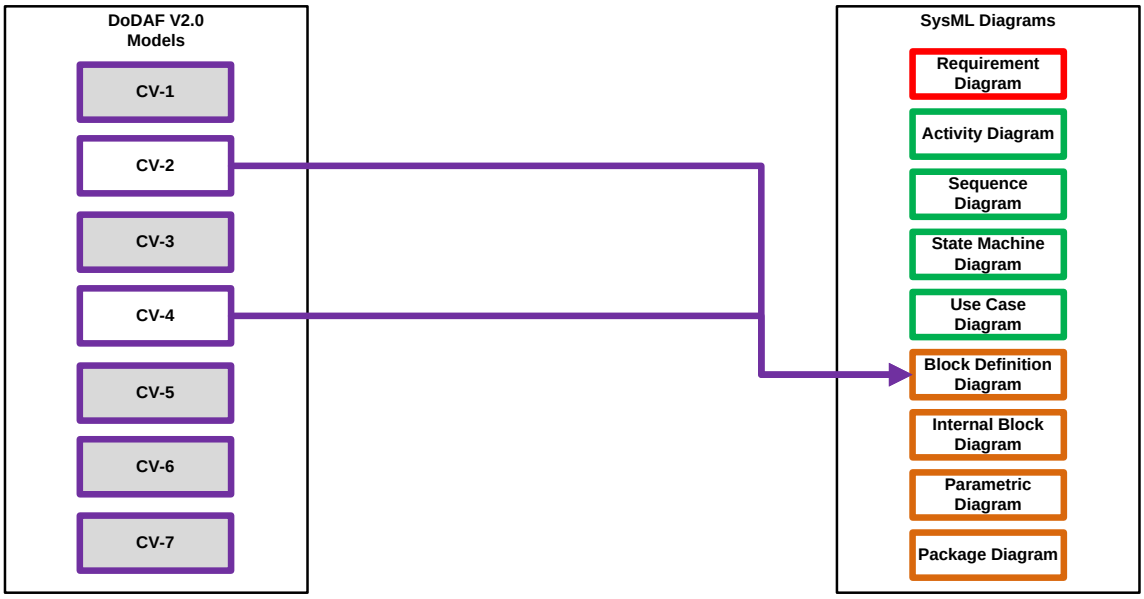

Figure 20: Capability View to SysML Mapping

CV-1: Vision

The CV-1 DoDAF-described model presents the overall vision for

transformational endeavors, which provides a strategic context for the capabilities described and a high-level scope. More specifically, the CV-1 provides a high-level scope for the Architectural Description which is more general than the scenario-based scope defined in an OV-1.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does suggest that the CV-1 model may be primarily textual descriptions of the overarching objectives of the transformation or change program that the Enterprise is engaged in. Of key importance is the identification of goals, together with the desired outcomes and measurable benefits associated with these.

CV-1:

- There is no equivalent / applicable SysML diagram that can help in the creation of the CV-1 DoDAF described model.
- The CV-1 model should be created in a structured textual format and it should capture the following information at a minimum: the identification of goals, together with the desired outcomes and measurable benefits associated with these.

Additional Information:

The intended usage of the CV-1 DoDAF-described model:

- Communication of the strategic vision regarding capability development.

CV-2: Capability Taxonomy

The CV-2 DoDAF-described model presents a hierarchy of capabilities, which specifies all the capabilities that are referenced throughout one or more Architectural Descriptions. More specifically, the CV-2 is structured as a hierarchy of capabilities, with the most general at the root and most specific at the leaves. At the leaf-level, capabilities may have a measure specified, along with an environmental condition for the measure. The CV-2 is used to capture and

organize the capability functions - required for the vision set out in the CV-1 Vision. CV-2 does not specify how the capabilities are to be implemented.

CV-2:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the CV-2 model for the following reasons:
        - Block definition diagrams facilitate relationship definition between the blocks, resulting in an easy way to depict the hierarchal relationship between the capabilities.

Additional Information:

The intended usage of the CV-2 DoDAF-described model:

- Identification of capability requirements.
- Capability planning (capability taxonomy).
- Codifying required capability elements.
- Capability audit.
- Capability gap analysis.
- Source for the derivation of cohesive sets of user requirements.
- Providing reference capabilities for architectures.

CV-3: Capability Phasing

The CV-3 DoDAF-described model presents the planned achievement of capability at different points in time or during specific periods of time (associated

with the phases - see CV-1 Vision model). The CV-3 shows the capability phasing in terms of the activities, conditions, desired effects, rules complied with, resource consumption and production, and measures, without regard to the performer and location solutions. More specifically, the CV-3 provides methods to identify gaps or duplication in capability provision and may be used to envisage the need for interventions in projects (to fulfill a capability gap) or to represent current plans (the availability of capability according to their delivery timescales).

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does suggest that the CV-3 model can be presented as a table consisting of rows representing Capabilities (derived from the CV-2 Capability Taxonomy model) and columns representing phases (from CV-1 Vision model).

CV-3:

- There is no equivalent / applicable SysML diagram that can help in the creation of the CV-3 DoDAF described model.
- Tabular Representation:
    - The population of the CV-3 model in practice, tends to iterate between considering the desired capability and considering what capability is planned to be delivered. The output from this iterative approach can be a table that represents the required capability phasing.
    - The CV-3 can be presented as a table consisting of rows representing Capabilities (derived from the CV-2 Capability Taxonomy model) and columns representing phases (from CV-1 Vision model).

Additional Information:

The intended usage of the CV-3 DoDAF-described model is:

- Capability planning (capability phasing).

- Capability integration planning.
- Capability gap analysis.

CV-4: Capability Dependencies

The CV-4 DoDAF-described model presents the dependencies between planned capabilities and the definition of logical groupings of capabilities. More specifically, the CV-4 shows the capabilities that are of interest to the Architectural Description. It groups those capabilities into logical groupings, based on the need for those elements to be integrated.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does suggest that the CV-4 model can be developed using a graphical approach.

CV-4:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the CV-4 model for the following reasons:
        - Block definition diagrams facilitate relationship definition between the blocks, resulting in an easy way to depict the hierarchal relationship (or dependencies) between the capabilities.

Additional Information:

The intended usage of the CV-4 DoDAF-described model is:

- Identification of capability dependencies.
- Capability management (impact analysis for options, disposal etc.).

CV-5: Capability to Organizational Development Mapping

The fulfillment of capability requirements shows the planned capability deployment and interconnection for a particular Capability Phase. The CV-5 DoDAF-described model presents the planned solution for the phase in terms of performers and locations and their associated concepts. More specifically, the CV-5 is used to support the capability management process and, in particular, assist the planning of fielding.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does suggest that the CV-5 model can be presented as a table, with the appropriate organizational structure represented by one axis, and the capabilities by the other axis.

CV-5:

- There is no equivalent / applicable SysML diagram that can help in the creation of the CV-5 DoDAF described model.
- Tabular Representation:
    - The CV-5 can be presented as a table, with the appropriate organizational structure represented by one axis, and the capabilities by the other axis.
    - Graphical objects representing Capabilities or resources can be placed in the relevant positions (intersections) relative to these axes.

Additional Information:

The intended usage of the CV-5 DoDAF-described model is:

- Fielding planning.
- Capability integration planning.

- Capability options analysis.
- Capability redundancy/overlap/gap analysis.
- Identification of deployment level shortfalls.

CV-6: Capability to Operational Activities Mapping

The CV-6 DoDAF-described model presents a mapping between the capabilities required and the operational activities that those capabilities support. More specifically, the CV-6 provides a bridge between capability analyzed using CVs and operational activities analyzed using OVs. It identifies how operational activities can be performed using various available capability elements. The capability to activity mappings may include both situations where activities fully satisfy the desired capability and those where the activity only partially meets the capability requirement.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does suggest that the CV-6 model can be presented as a table, with the rows representing the Capabilities and the columns representing the Operational Activities.

CV-6:

- There is no equivalent / applicable SysML diagram that can help in the creation of the CV-6 DoDAF described model.
- Matrix (Tabular) Representation:
    - The CV-6 can be presented as a table, with the rows representing the Capabilities and the columns representing the Operational Activities.
    - An X, date, or phase, may indicate that the capability may be utilized in support of that activity (by the date or phase indicated) whereas a blank indicates that it does not.

Additional Information:

The intended usage of the CV-6 DoDAF-described model is:

- Tracing capability requirements to operational activities.
- Capability audit.

CV-7: Capability to Services Mapping

The CV-7 DoDAF-described model presents a mapping between the capabilities and the services that these capabilities enable. More specifically, the CV-7 provides a bridge between capability analyzed using CVs and services analyzed using SvcVs. It identifies how services can be performed using various available capability elements. The capability to service mappings may include both situations where a service fully satisfies the desired capability and those where the service only partially meets the capability requirement.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does suggest that the CV-7 model can be presented as a table, with the rows representing the Capabilities and the columns representing the Services.

CV-7:

- There is no equivalent / applicable SysML diagram that can help in the creation of the CV-7 DoDAF described model.
- Matrix (Tabular) Representation:
    - The CV-7 can be presented as a table, with the rows representing the Capabilities and the columns representing the Services.
    - An X, date, or phase, may indicate that the capability may be utilized in support of that activity (by the date or phase indicated) whereas a blank indicates that it does not.

Additional Information:

The intended usage of the CV-7 DoDAF-described model is:

- Tracing capability requirements to services.
- Capability audit.

DATA AND INFORMATION VIEWPOINT

The data and information viewpoint articulates the data relationships and alignment structures in the architecture content. DoDAF V2.0 incorporates three levels of abstraction that correlate to the different levels associated with most data models developed in support of the operations or business. These levels are: conceptual, logical, and physical.

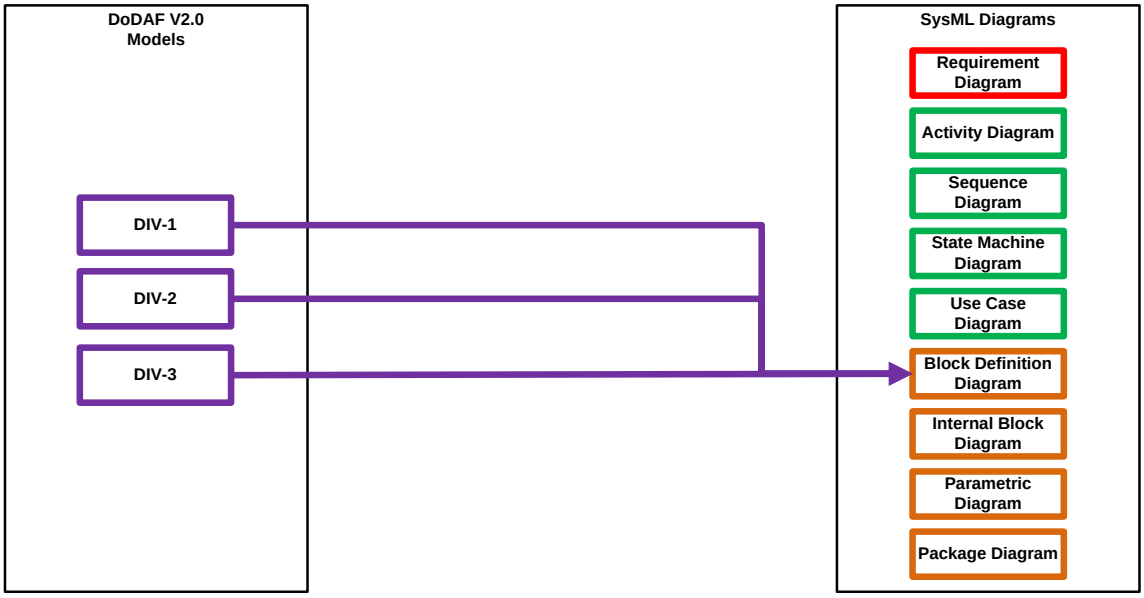

Figure 21: Data and Information View to SysML Mapping

DIV-1: Conceptual Data Model

The DIV-1, logical data model, represents the required high-level data concepts and their relationships. More specifically, the DIV-1 model is used to document the business information requirements and structural business process rules of the architecture. It describes the information that is associated with the information of the architecture. Included are information items, their attributes or characteristics, and their inter-relationships.

DoDAF v2.0 does not endorse a specific logical modeling methodology, and it also does not provide any insight into the possible construction methods for DIV-1.

DIV-1:

- *Block Definition Diagram:*
  - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
  - The block definition diagram is a suitable tool that would prove helpful in developing the DIV-1 model for the following reasons:
    - The block is the modular unit of structure in SysML that is used to define a number of items including logical abstractions.
    - The relevant requirements can be captured within a given block.
    - If we assume the business information data model (DIV-1) consists of business information entities and how they are related, the block definition diagram provides an easy methodology for depicting a DIV-1 model.
    - Within a bdd, attributes (and operations) can easily be captured and defined.

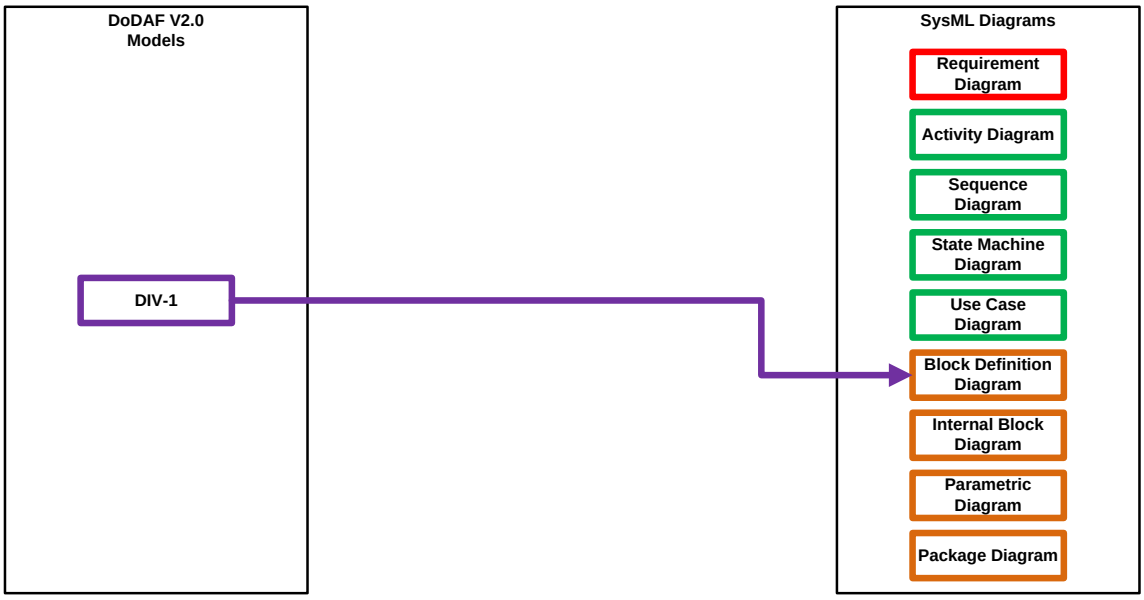

Figure 22: DIV-1 to SysML Mapping

Additional Information:

The intended usage of the DIV-1 DoDAF-described model is:

- Information requirements.
- Information hierarchy.

DIV-2: Logical Data Model

The DIV-2, logical data model, represents the documentation of the data requirements and structural business process (activity) rules. More specifically, the DIV-2 model allows analysis of an architecture's data definition aspect, without consideration of implementation specific or product specific issues. Another purpose is to provide a common dictionary of data definitions to consistently express models wherever logical-level data elements are included in the description. For the DIV-2, care should be taken to avoid hidden overlaps, where there is a semantic overlap between concepts with different entity, attribute, or domain value names.

Although DoDAF v2.0 does not endorse a specific logical modeling methodology, it does provide insight into some of the possible construction

methods for DIV-2. The appropriate way to develop a logical data model depends on the technology chosen as the main design solution (e.g., relational theory or object-orientation). For relational theory, a logical data model seems best described using an entity relationship diagramming technique. For Object-Oriented, a logical data model seems best described using Class and/or Object diagrams. In the case of this study, an object oriented approach is appropriate given SysML is the modeling language in question.

DIV-2:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the DIV-2 model for the following reasons:
        - The block is the modular unit of structure in SysML that is used to define a number of items including logical abstractions.
        - The relevant requirements can be captured within a given block.
        - If we assume the logical data model (DIV-2) consists of entities and how they are related, the block definition diagram provides an easy methodology for depicting a DIV-2 model.
        - Within a bdd, data definitions can be complex data structures.

DIV-3: Physical Data Model

The DIV-3, logical data model, represents the physical implementation format of the Logical Data Model entities, e.g., message formats, file structures, physical schema. In DoDAF V1.5, this was the SV-11. More specifically, the DIV-3 model defines the structure of the various kinds of system or service data that are utilized by the systems or services in the Architectural Description. DIV-3 is used to describe how the information represented in the DIV-2 Logical Data Model is actually implemented. DIV-3 describes data relevant at the system or service-level.

Although DoDAF v2.0 does not endorse a specific logical modeling methodology, it does provide insight into some of the possible construction methods for DIV-3. The appropriate way to develop a physical data model depends on the product chosen to instantiate the logical data model (e.g., a relational database management system [RDBMS]). A physical data schema model seems best described using an entity-relationship diagramming technique. For Object-Oriented data modeling, a physical data schema seems best described using by Class and/or Object diagrams. For other implementation technologies, such as message orientation, a reference to a message format standard might be more appropriate. In the case of this study, an object oriented approach is appropriate given SysML is the modeling language in question.

DIV-3:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove

helpful in developing the DIV-3 model for the following reasons:

- The block is the modular unit of structure in SysML that is used to define a number of items including logical abstractions.
- The relevant requirements can be captured within a given block.
- If we assume the physical data model (DIV-3) consists of entities and how they are related, the block definition diagram provides an easy methodology for depicting a DIV-3 model.
- Within a bdd, data definitions can be complex data structures.

Additional Information:

The intended usage of the DIV-3 DoDAF-described model is:

- Specifying the system/service data elements exchanged between systems and/or services, thus reducing the risk of interoperability errors.
- Definition of physical data structure.
- Providing as much detail as possible on data elements exchanged between systems, thus reducing the risk of interoperability problems.
- Providing data structures for use in the system design process, if necessary.
- Providing a common dictionary of data implementation elements (e.g., tables and records in a relational database schema) to consistently express models wherever physical-level data elements are included in the descriptions.
- Providing as much detail as possible on the system or service data elements exchanged between systems, thus reducing the risk of interfacing errors.

- Providing system and service data structures for use in the system and service design process, if necessary.

OPERATIONAL VIEWPOINT

The Operational Viewpoint articulates operational scenarios, processes, activities & requirements. The OV DoDAF-described Models re-use the capabilities defined in the Capability Viewpoint and put them in the context of an operation or scenario. The OV DoDAF-described Models can be used in a number of ways, including the development of user requirements, capturing future concepts, and supporting operational planning processes.

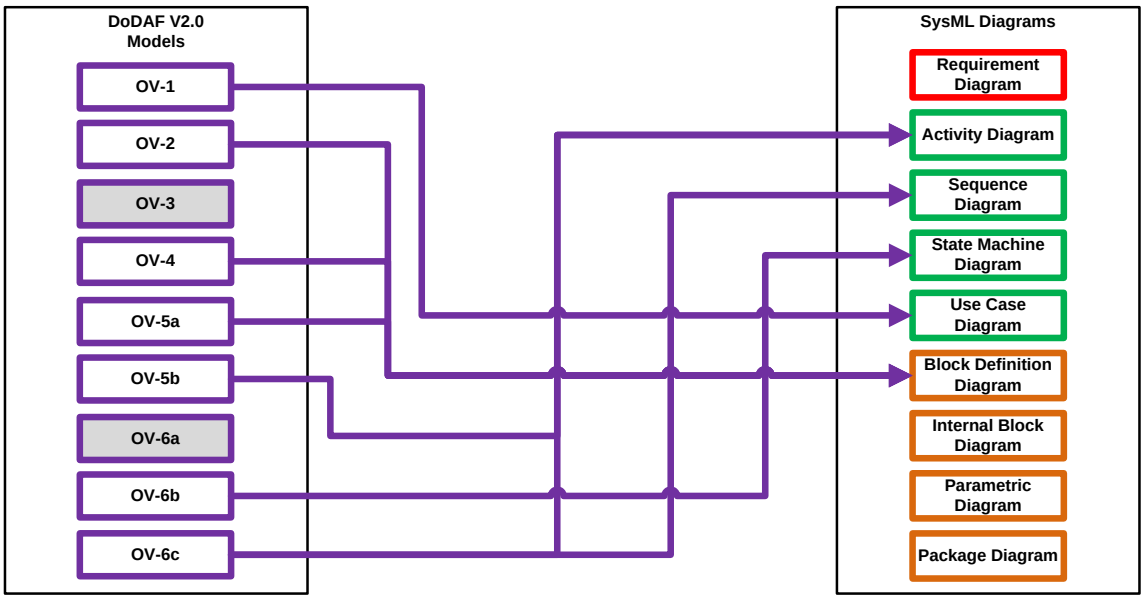

Figure 23: Operational View to SysML Mapping

OV-1: High-Level Operational Concept

The OV-1 DoDAF-described model presents a high-level graphical/textual description of the operational concept. More specifically, the OV-1 model describes a mission, class of mission, or scenario. Its purpose is to provide a quick, high-level description of what the architecture is supposed to do, and how it is supposed to do it. Its main utility is as a facilitator of human communication, and it is intended for presentation to high-level decision-makers as it conveys, in simple terms, what the Architectural Description is about and an idea of the players and operations involved.

In the case of the OV-1 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. It is suggested however, that the OV-1 model should consist of a graphical (one or more graphics, possibly a video clip) executive summary for a given Architectural Description with accompanying text.

OV-1:

- *Use Case Diagram:*
  - The use case diagram (uc) is used to model the relationship between the system under consideration or subject, its actors, and use cases. In other words, it models high-level functionality in terms of how a system or other entity is used by external entities (i.e. actors) to accomplish a set of goals.
  - The use case diagram is a suitable tool that would prove helpful in developing the OV-1 model for the following reasons:
    - In general the OV-1 model describes the business activities or missions, high-level operations, organizations, and geographical distribution of assets. The model frames the operational concept (what happens, who does what, in what order, to accomplish what goal) and highlight interactions to the environment and other external systems. This can easily be achieved with use case diagrams (one or many) and the use of the supporting artifacts listed below.
    - Use case descriptions can capture the text requirements for the OV-1, and describe the use case diagrams . These can be exported in report format from the SysML tool.
- *Additional supporting artifacts may also be required:*
  - Depending on the SysML tool employed, a high-level graphic (Powerpoint, Paint, Unity3D, Blender, Flash creator tool, etc) may need to be created to support the use case diagram.

- A structured text document should be created to describe the OV-1 model. This may be captured by the built in use case descriptions – a report can be exported from the SysML tool.

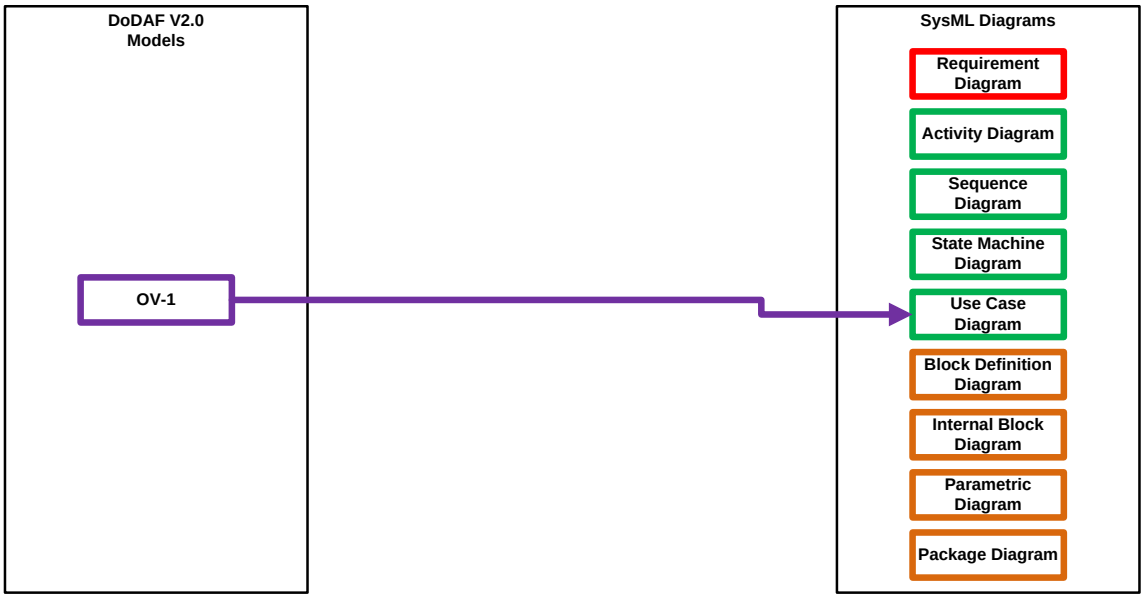

Figure 24: OV-1 to SysML Mapping

Additional Information:

The intended usage of the OV-1 DoDAF-described model is:

- Putting an operational situation or scenario into context.
- Providing a tool for discussion and presentation; for example, aids industry engagement in acquisition.
- Providing an aggregate illustration of the details within the published high-level organization of more detailed information in published architectures.

OV-2: Operational Resource Flow Description

The OV-2 DoDAF-described model presents a description of the resource (information, funding, personnel, or materiel) flows exchanged between operational activities. It is intended to be logical and describes who or what, not how. More specifically, the OV-2 model provides a focus for the operational

requirements which may reflect any capability requirements that have been articulated but within the range of operational settings that are being used for operational architecture. The aim of the OV-2 is to record the operational characteristics for the community of anticipated users relevant to the Architectural Description and their collaboration needs, as expressed in Needlines and Resource Flows. There can be a one-to-many relationship from Needlines to Resource Flow (e.g., a single Needline in OV-2 represents multiple individual Resource Flows). The OV-2 is not a communications link or communications network diagram but a high-level definition of the logical requirement for resource exchange. In addition to Needlines, Resource Flow Connectors can be used to overlay contextual information about how the Operational Activities and Locations interact via physical flows. ***Note:*** The mapping of the Resource Flows to the Needlines of OV-2 occurs in the Operational Resource Flow Matrix (OV-3), where the identity of the individual elements and their attributes are documented.

In the case of the OV-2 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. No guidance is provided for the construction of the OV-2 model (graphic).

OV-2:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the OV-2 model for the following reasons:
        - The block is the modular unit of structure in SysML that is

used to define a number of items including system components, logical abstractions, and items that flow through the system.

- Flow port nodes can be used to represent an interaction point where different items (single / multiple items) may flow into or out of a block.
- SysML provides several mechanisms to relate activities to the blocks that perform them.

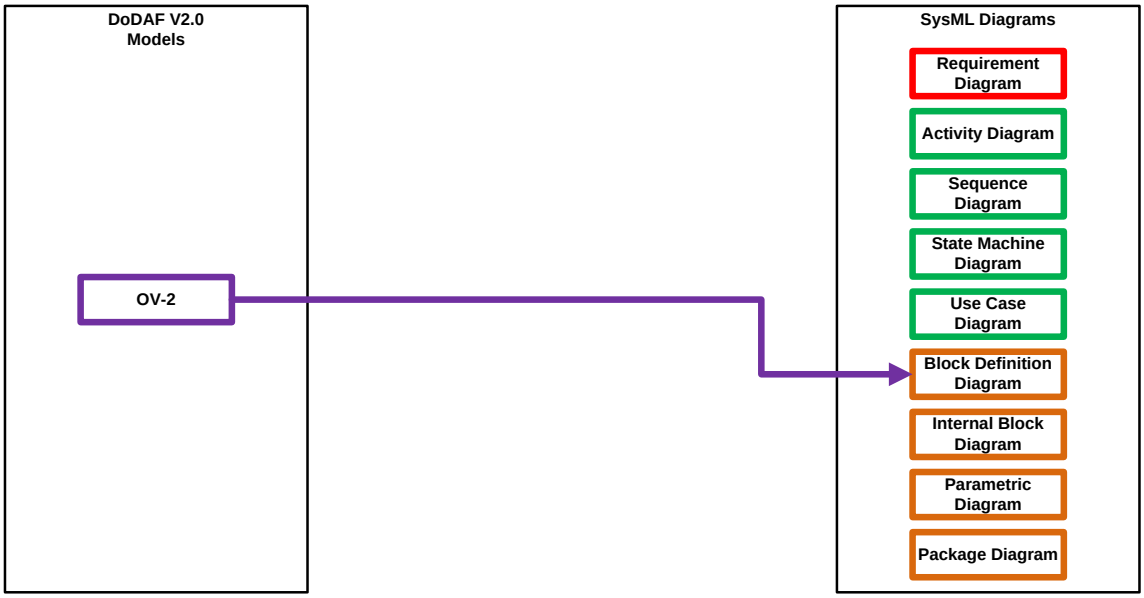

Figure 25: OV-2 to SysML Mapping

Additional Information:

The intended usage of the OV-2 DoDAF-described model is:

- Definition of operational concepts.
- Elaboration of capability requirements.
- Definition of collaboration needs.
- Applying a local context to a capability.
- Problem space definition.
- Operational planning.

- Supply chain analysis.
- Allocation of activities to resources.

OV-3: Operational Resource Flow Matrix

The OV-3 DoDAF-described model provides a description of the resources exchanged and the relevant attributes of the exchanges. More specifically, the OV-3 model identifies the resource flows that are necessary to support operations to achieve a specific operational task. This model is initially constructed from the information contained in the OV-2 Operational Resource Flow Description model. The OV-3 details resource flow exchanges by identifying which operational activity and locations exchange what resources, with whom, why the resource is necessary, and the key attributes of the associated resources. The emphasis in this model is on the logical and operational characteristics of the Resource Flows being exchanged, with focus on the Resource Flows crossing the capability boundary. ***Note:*** There is not always a one-to-one mapping of OV-3 Resource Flows to OV-2 Operational Resource Flow Description Needlines; rather, many individual Resource Flows may be associated with one Needline.

In the case of the OV-3 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it does not prescribe the column headings (interaction characteristics) or the symbols to be used in an OV-3 Matrix (tabular format).

OV-3:

- There is no equivalent / applicable SysML diagram that can help in the creation of the OV-3 DoDAF described model.
- A matrix is to be developed using the SysML tool and the OV-2 model diagram.

Additional Information:

The intended usage of the OV-3 DoDAF-described model is:

- Definition of interoperability requirements.

OV-4: Organizational Relationships Chart

The OV-4 DoDAF-described model presents an organizational context, role or other relationships among organizations. The OV-4 exists in two forms; role-based (e.g., a typical brigade command structure) and actual (e.g., an organization chart for a department or agency). More specifically, a typical (role-based) OV-4 illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in the business represented by the architecture, whereas, an actual OV-4 shows real organizations and the relationships between them. In both cases, it is possible to overlay resource interaction relationships which denote relationships between organizational elements that are not strictly hierarchical (e.g., a customer-supplier relationship). An OV-4 may be a hybrid diagram showing typical and actual organization structures.

In the case of the OV-4 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. No guidance is provided for the construction of the OV-4 model.

OV-4:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the OV-4 model for the following reasons:

- The block is the modular unit of structure in SysML that is used to define a number of items including systems, and system components.
- The bdd is used to define blocks in terms of their structural and behavioral features, and the relationships between the blocks such as their hierarchal relationship.

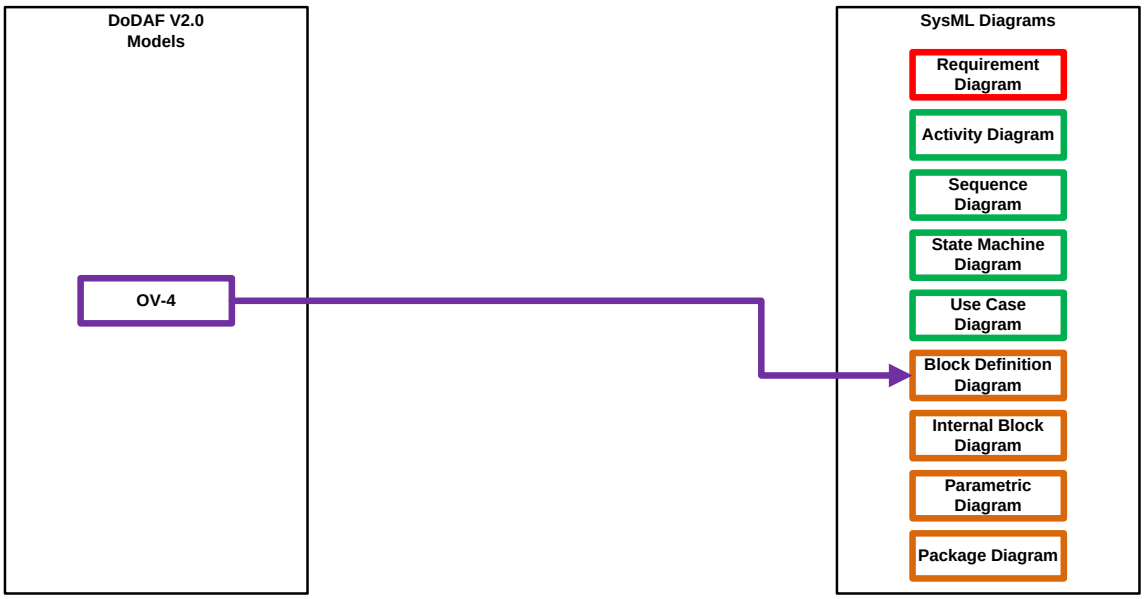

Figure 26: OV-4 to SysML Mapping

Additional Information:

The intended usage of the OV-4 DoDAF-described model is:

- Role-Based OV-4:
  - Organizational analysis.
  - Definition of human roles.
  - Operational analysis.
- Actual OV-4:
  - Identify architecture stakeholders.
  - Identify process owners.
  - Illustrate current or future organization structure.

OV-5a: Operational Activity Decomposition Tree and OV-5b: Operational Activity Model

The OV-5a and OV-5b DoDAF-described models are presented as a single package description. They describe the operations that are normally conducted in the course of achieving a mission or a business goal, for example, the operational activities (or tasks), input/output flows between activities, and to/from activities that are outside the scope of the Architectural Description. More specifically, OV-5a describes the capabilities and activities (operational activities) organized in a hierarchal structure. OV-5b describes context of capabilities and activities (operational activities) and their relationships among activities, inputs, and outputs. Additional data captured as part of Ov-5b can show cost, performers or other pertinent information. The OV-5a helps provide an overall picture of the activities involved and a quick reference for navigating the OV-5b.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does provide insight into some of the possible construction methods for OV-5b. No guidance is provided for OV5a. The OV-5b model can be constructed using Integration Definition for Function Modeling (IDEF0) or Class Diagrams.

OV-5a:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the OV-5a model for the following reasons:
        - The required notation exists for defining activity models on

block definition diagrams.

- The block is the modular unit of structure in SysML that is used to define a number of items including items that flow through the system.
- Block definition diagrams facilitate relationship definition between the blocks, resulting in an easy way to depict the hierarchal relationship between the capabilities and the activities.
- SysML provides several mechanisms to relate activities to the blocks that perform them.

OV-5b:

- *Activity Diagram:*
  - The activity diagram (act) is used to model behavior in terms of the flow of inputs, outputs, and control. An activity diagram is similar to a traditional functional flow diagram.
  - The activity diagram is a suitable tool that would prove helpful in developing the OV-5b model for the following reasons:
    - It is the principle diagram used to describe activities and their associated input/output flows (object flow) whether they are discrete or continuous.
    - The concept of "control flow" provides the capability for the modeler to indicate constraints relating to when, and in which order the actions within an activity will be executed.
    - The required notation exists for defining activity diagram structural nodes, control nodes, object and action nodes, and paths.
    - SysML provides several mechanisms to relate activities to the blocks that perform them.

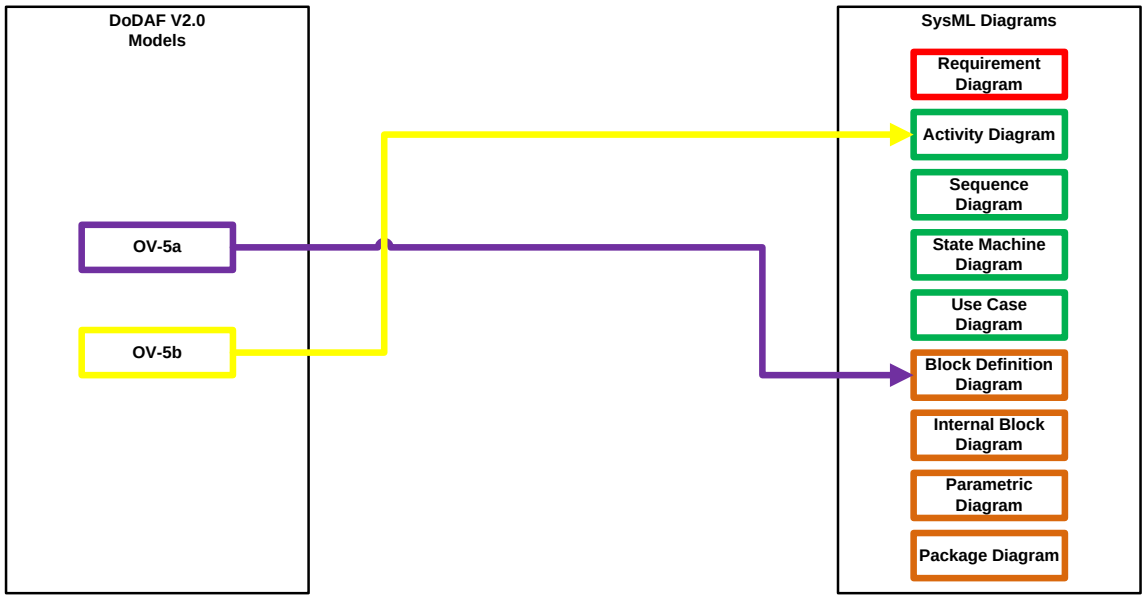

Figure 27: OV-5a and OV-5b to SysML Mapping

Additional Information:

The intended usage of the OV-5a and OV-5b DoDAF-described models is:

- Requirements capture.
- Description of activities and workflows.
- Definition of roles and responsibilities.
- Support task analysis to determine training needs.
- Problem space definition.
- Operational planning.
- Logistic support analysis.
- Information flow analysis.

OV-6a: Operational Rules Model

The OV-6a DoDAF-described model is one of three models used to describe activity (operational activity). It identifies business rules that constrain operations. More specifically, the OV-6a model specifies operational or business rules that are constraints on the way business is done in the enterprise. Operational (mission oriented) rules are statements that constrain some aspect of

the mission or the architecture. OV-6a can also be used to extend the capture of business requirements by constraining the structure and validity of DIV-2 Logical Data Model elements.

In the case of the OV-6a DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. The OV-6a model should be created in a textual format and in the English language.

OV-6a:

- There is no equivalent / applicable SysML diagram that can help in the creation of the OV-6a DoDAF described model.
- A model (table) is to be developed that will feed into other models created using the SysML tool. The OV-6a should provide a listing of the complete set of operational rules with a reference to any models that they affect.
- A rule defined in textual form OV-6a may be applied to any Architectural element defined in an OV.
- The OV-6a should demonstrate traceability back to OV-1.
- The OV-6a model should be created using a format that can easily be imported to the SysML tool e.g. XML.

Additional Information:

The intended usage of the OV-6a DoDAF-described model is:

- Definition of doctrinally correct operational procedures.
- Definition of business rules
- Identification of operational constraints.

OV-6b: State Transition Description

The OV-6b DoDAF-described model represents one of three models used to describe operational activity (activity). It describes how an Operational Activity responds to various events by changing its state. More specifically, OV-6b can be used to describe the detailed sequencing of activities or work flow in the business

process. The OV-6b is particularly useful for describing critical sequencing of behaviors and timing of operational activities that cannot be adequately described in the OV-5b Operational Activity Model.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does provide insight into some of the possible construction methods for OV-6b indicating that the OV-6b model is based on the statechart diagram.

OV-6b:

- *State Machine Diagram:*
  - The state machine diagram (stm) is used in SysML to describe the state-dependent behavior of a block throughout its lifecycle in terms of its states and the transitions between them.
  - The state machine diagram is a suitable tool that would prove helpful in developing the OV-6b model for the following reasons:
    - OV-6b is based on the statechart diagram.
    - State machine diagrams are sometimes referred to as state charts or state diagrams.
    - stm diagrams facilitate the representation of states, and transitions (triggers, guards, effects).
    - Call events within the stm facilitate the response to operational calls.

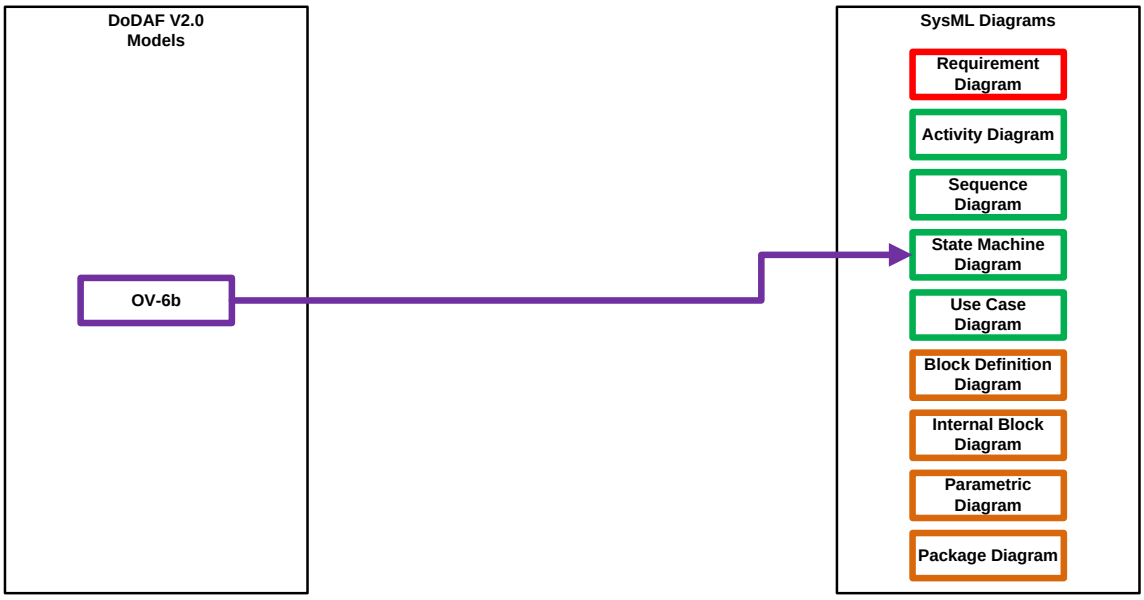

Figure 28: OV-6b to SysML Mapping

Additional Information:

The intended usage of the OV-6b DoDAF-described model is:

- Analysis of business events.
- Behavioral analysis.
- Identification of constraints.

OV-6c: Event-Trace Description

The OV-6c DoDAF-described model represents one of three models used to describe activity (operational activity). More specifically, it traces actions in a scenario or critical sequence of events and is valuable for moving to the next level of detail from the initial operational concepts. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does provide insight into some of the possible construction methods for OV-6c. The OV-6c model can be constructed using any modeling notation (e.g., BPMN) that supports the layout of timing and sequence of

activities along with the Resource Flow exchanges that occur between Operational Activities/Locations for a given scenario. Different scenarios can be depicted by separate diagrams.

OV-6c:

- *Activity Diagram:*
  - The activity diagram (act) is used to model behavior in terms of the flow of inputs, outputs, and control. An activity diagram is similar to a traditional functional flow diagram.
  - The activity diagram is a suitable tool that would prove helpful in developing the OV-6c model for the following reasons:
    - It is the principle diagram used to describe activities and their associated input/output flows (object flow) whether they are discrete or continuous.
    - It facilitates the tracing of actions in a given scenario.
    - The required notation exists for defining activity diagram structural nodes, control nodes, object and action nodes, and paths.
- *Sequence Diagram:*
  - The sequence diagram (sd) is used to represent the interaction between structural elements of a block, as a sequence of message exchanges.
  - The sequence diagram is a suitable tool that would prove helpful in developing the OV-6c model for the following reasons:
    - It is the principle diagram used to describe activities and their associated input/output flows (object flow) whether they are discrete or continuous.
    - It can easily be used to depict a sequence of events.
    - Sequence diagrams can represent the interaction between structural elements of a block, where the interaction is

between the system and its environment or between the components of a system at any level of a system hierarchy.

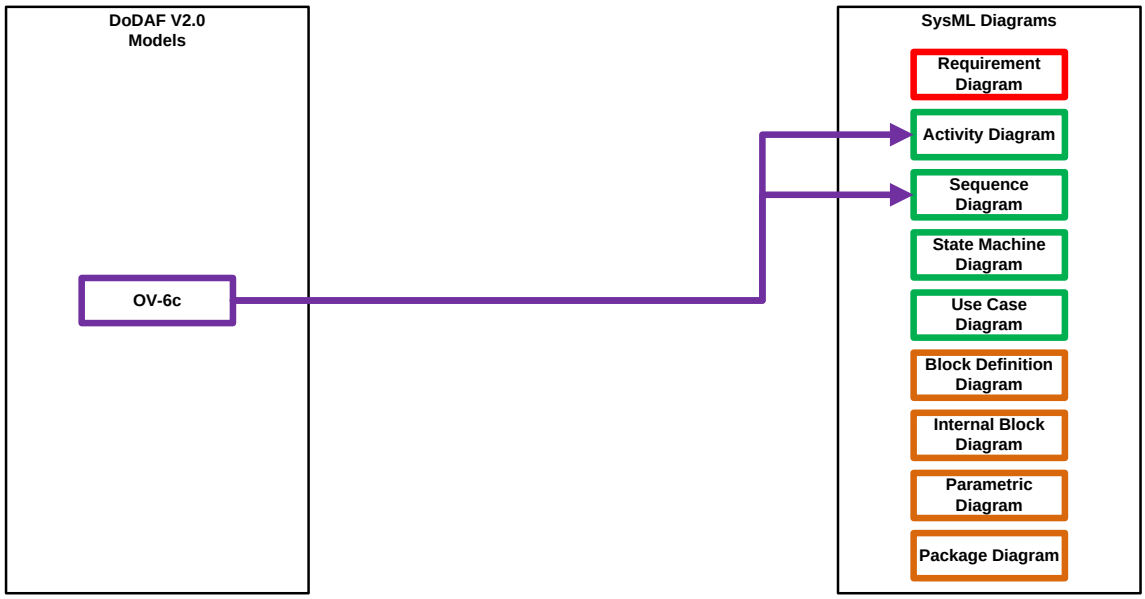

Figure 29: OV-6c to SysML Mapping

Additional Information:

The intended usage of the OV-6c DoDAF-described model is:

- Analysis of operational events.
- Behavioral analysis.
- Identification of non-functional user requirements.
- Operational test scenario.

PROJECT VIEWPOINT

The project viewpoint describes the relationships between operational and capability requirements and the various projects being implemented. It also details dependencies between capability management and the Defense Acquisition System process. The Project Models can be used to answer questions such as:

- What capabilities are delivered as part of this project?
- Are there other projects that either affect or are affected by this project? To what portfolios do the projects or projects belong?
- What are the important milestones relative to this project? When can I expect capabilities to be rendered by this project to be in place?

PV-1: Project Portfolio Relationships

The PV-1 DoDAF-described model describes the dependency relationships between the organizations and projects and the organizational structures needed to manage a portfolio of projects. More specifically, the PV-1 enables the user to model the organizational structures needed to manage programs, projects, portfolios, or initiatives. The PV-1 provides a way of describing the organizational relationships between multiple acquisition projects or portfolios, each of which are responsible for delivering individual systems or capabilities.

DoDAF v2.0 does not endorse a specific activity modeling methodology, and it also does not suggest any suitable development/construction approach for this model.

PV-1:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their

hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.

- The block definition diagram is a suitable tool that would prove helpful in developing the PV-1 model for the following reasons:
  - Block definition diagrams facilitate relationship definition between the blocks, resulting in an easy way to depict the hierarchal relationship (or dependencies) between the organizations and projects and the organizational structures needed to manage a portfolio of projects.

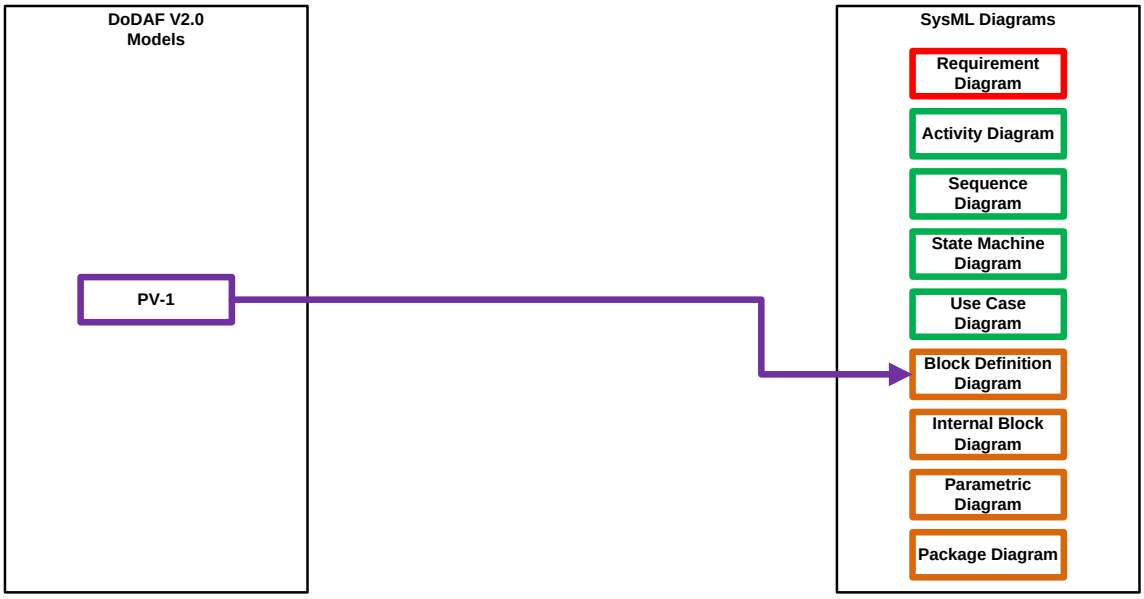

Figure 30: PV-1 to SysML Mapping

Additional Information:

The intended usage of the PV-1 DoDAF-described model is:

- Program management (specified acquisition program structure).
- Project organization.
- Cross-cutting initiatives to be tracked across portfolios.

PV-2: Project Timelines

The PV-2 DoDAF-described model provides a graphical timeline perspective on programs or projects, with the key milestones and interdependencies. More specifically, the PV-2 is intended primarily to support the acquisition and fielding processes including the management of dependencies between projects and the integration of DoD 5000.1 Defense Acquisition System policies to achieve a successfully integrated capability. However, the PV-2 is not limited to the acquisition and fielding processes. The information provided by the model can be used to determine the impact of either planned or unplanned programmatic changes, and highlight opportunities for optimization across the delivery program. It may sometimes be convenient to use a PV-2 timeline model for other purposes, e.g., to show temporal relationships between transformation initiatives at the strategic-level or for technology roadmapping. Use of PV-2 should support the management of capability delivery and be aligned with the CV-3 Capability Phasing model, if one exists.

In the case of the PV-2 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology; however, it does indicate the PV-2 can be represented using a Gantt chart that displays the entire lifecycle of each project, together with dependencies between them.

PV-2:

- There is no equivalent / applicable SysML diagram that can help in the creation of the PV-2 DoDAF described model.
- A Gantt Chart (or similar diagram) is to be developed using an appropriate project management tool e.g. Microsoft Project. It should display the entire lifecycle of each project, together with dependencies between them.

Additional Information:

The intended usage of the PV-2 DoDAF-described model is:

- Project management and control (including delivery timescales).

- Project dependency risk identification.
- Management of dependencies.
- Portfolio management.

PV-3: Project to Capability Mapping

The PV-3 DoDAF-described model provides a mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability. More specifically, the PV-3 supports the acquisition and deployment processes, including the management of dependencies between projects and the integration of all relevant project and program elements to achieve a capability. The analysis can be used to identify capability redundancies and shortfalls, highlight phasing issues, expose organizational or system interoperability problems, and support program decisions, such as when to phase out a legacy system.

In the case of the PV-3 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology; however, it does indicate the PV-3 can have a tabular representation.

PV-3:

- There is no equivalent / applicable SysML diagram that can help in the creation of the PV-3 DoDAF described model.
- Matrix (Tabular) Representation:
    - The PV-3 can be presented as a table, with the rows representing the Capabilities and the columns representing the programs, projects, portfolios, or initiatives.
    - An X, date, or phase, may indicate that the capability may be utilized in support of that program, project, portfolio, or initiative (by the date or phase indicated) whereas a blank indicates that it does not.

Additional Information:

The intended usage of the PV-3 DoDAF-described model is:

- Tracing capability requirements to projects.
- Capability audit.

SERVICES VIEWPOINT (ALSO: SYSTEMS VIEWPOINT)[2]

The services viewpoint articulates the performers, activities, services, and their exchanges providing for, or supporting, DoD functions.

SvcV-1 Services Context Description

The SvcV-1 DoDAF-described model represents identification of services, service items, and their interconnections. More specifically, it focuses on the Resource Flow and the providing service. The primary purpose of a SvcV-1 model is to show resource structure, i.e., identify the primary sub-services, performer and activities (functions) and their interactions. SvcV-1 contributes to user understanding of the structural characteristics of the solution.

In the case of the SvcV-1 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it does not provide insight into some of the possible construction methods. It does however indicate that if a single SvcV-1 is not possible, the resource of interest should be decomposed into multiple SvcV-1 models.

SvcV-1:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove

[2] At the time of this writing the Services and Systems viewpoint in DoDAF 2.0 were identical, except for the moniker "System" for the SV and "Service" for the SvcV architecture perspectives. The mappings and recommendations for the services viewpoint are identical for the systems viewpoint. For this reason the systems viewpoint is not listed separately.

helpful in developing the SvcV-1 model for the following reasons:

- The block is the modular unit of structure in SysML that is used to define a number of items including the type of system (or service), and the items (resources) that flows through the system (or service).
- Within a bdd, the notation for depicting interfaces exists.

- *Internal Block Diagram:*
  - The internal block diagram (ibd) is used to describe the internal structure of a block in terms of how its parts are interconnected.
  - The internal block diagram is a suitable tool that would prove helpful in developing the SvcV-1 model for the following reasons:
    - As with a bdd, systems (or services), subsystems (or subservices), flows, and interfaces can easily be depicted with an internal block diagram.
    - The ibd will ensure that the decomposition from the higher level bdd will reach the appropriate level of detail.

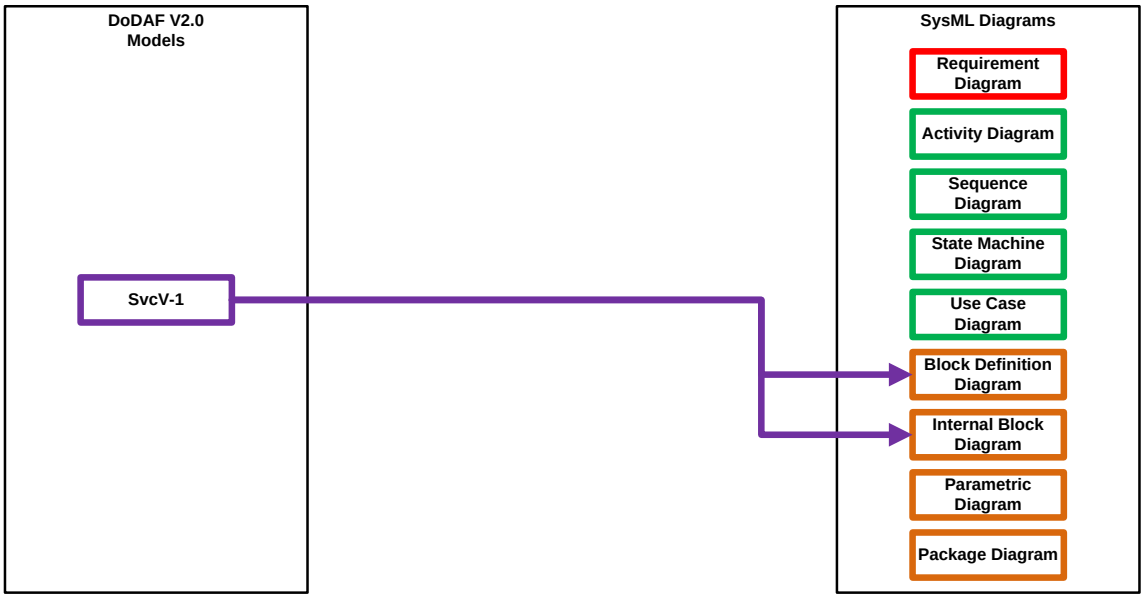

Figure 31: SvcV-1 to SysML Mapping

Additional Information:

The intended usage of the SvcV-1 DoDAF-described model is:

- Definition of service concepts.
- Definition of service options.
- Service Resource Flow requirements capture.
- Capability integration planning.
- Service integration management.
- Operational planning (capability and performer definition).

SvcV-2 Services Resource Flow Description

The SvcV-2 DoDAF-described model represents a description of resource flows exchanged between services. It comprises of services, their ports, and the resource flows between those ports. More specifically, it is used to give a precise specification of a connection between services. This may be an existing connection, or a specification for a connection that is to be made for a future connection.

In the case of the SvcV-2 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it does not provide insight into some of the possible construction methods.

SvcV-2:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the SvcV-2 model for the following reasons:

- The block is the modular unit of structure in SysML that is used to define a number of items including the type of system (or service), and the items (resources) that flows through the system (or service).
- Within a bdd, ports are structural features that describe the points at which a block interacts with other blocks. Flow ports are relevant to the SvcV-2 model, they specify what can flow in and out of blocks. Item flows can be used to describe what flows on the connectors between ports.

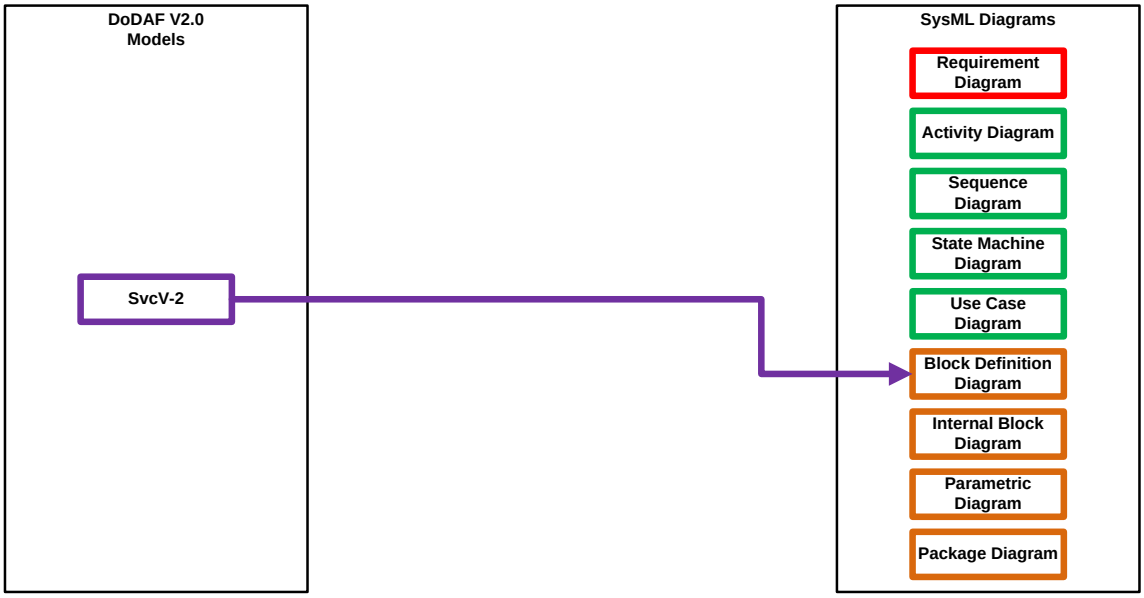

Figure 32: SvcV-2 to SysML Mapping

Additional Information:

The intended usage of the SvcV-2 DoDAF-described model is:

- Resource Flow specification.
- Each SvcV-2 Model can depict:
  - Which ports are connected?
  - The producing Services that the ports belong to.
  - The Services that the Service Resource Flows are consumed by.
  - The definition of the Service Resource Flow in terms of the

physical/logical connectivity and any protocols that are used in the connection.

SvcV-3a Systems-Services Matrix

The SvcV-3a DoDAF-described model represents the relationships among or between systems and services in a given Architectural Description. It can be organized in a number of ways to emphasize the association of system-to-service interactions in context with the architecture's purpose. More specifically, the SvcV-3a model provides a tabular summary of the system and services interactions specified in the SvcV-1 Services Context Description for the Architectural Description. The matrix format supports a rapid assessment of potential commonalities and redundancies (or, if fault-tolerance is desired, the lack of redundancies). The suite of SvcV-3a models can be organized in a number of ways (e.g., by domain, by operational mission phase, by solution option) to emphasize the association of groups of resource pairs in context with the Architectural Description purpose.

In the case of the SvcV-3a DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology; however, it does indicate the SvcV-3a is generally presented as a matrix, where the System and Services resources are listed in the rows and columns of the matrix, and each cell indicates an interaction between Systems and Services if one exists.

SvcV-3a:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-3a DoDAF described model.
- A matrix is to be developed using the SysML tool and the SvcV-1 model diagram.

Additional Information:

The intended usage of the SvcV-3a DoDAF-described model is:

- Summarizing system and service resource interactions.
- Interface management.
- Comparing interoperability characteristics of solution options.

SvcV-3b Services-Services Matrix

The SvcV-3b DoDAF-described model represents the relationships among services in a given Architectural Description. It can be designed to show relationships of interest, (e.g., service-type interfaces, planned vs. existing interfaces). More specifically, the SvcV-3b model provides a tabular summary of the services interactions specified in the SvcV-1 Services Context Description for the Architectural Description. The matrix format supports a rapid assessment of potential commonalities and redundancies (or, if fault-tolerance is desired, the lack of redundancies). The suite of SvcV-3b DoDAF-described Models can be organized in a number of ways (e.g., by domain, by operational mission phase, by solution option) to emphasize the association of groups of resource pairs in context with the Architectural Description purpose. It is important to note that one usage of the Service-Service Matrix (SvcV-3b) can support a net-centric (service-oriented) implementation in describing the interactions between producing services and consuming services. ***Note:*** This model is useful in support of net-centric (service-oriented) implementation of services as an input to the SvcV-10a Services Rules Model, SvcV-10b Services State Transition Description, and SvcV-10c Services Event-Trace Description, implemented as orchestrations of services.

In the case of the SvcV-3b DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology; however, it does indicate the SvcV-3b is generally presented as a matrix, where the Services resources are listed in the rows and columns of the matrix, and each cell indicates an interaction between Services if one exists.

SvcV-3b:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-3b DoDAF described model.
- A matrix is to be developed using the SysML tool and the SvcV-1 model diagram.
- This matrix provides input to the SvcV-10a Services Rules Model, SvcV-10b Services State Transition Description, and SvcV-10c Services Event-Trace Description, implemented as orchestrations of services.

Additional Information:

The intended usage of the SvcV-3b DoDAF-described model is:

- Summarizing service resource interactions.
- Interface management.
- Comparing interoperability characteristics of solution options.

SvcV-4 Services Functionality Description

The SvcV-4 DoDAF-described model represents the functions (activities) performed by services and the service data flows among service functions (activities). More specifically, the SvcV-4 model develops a clear description of the necessary data flows that are input (consumed) by and output (produced) by each resource, ensures that the service functional connectivity is complete (i.e., that a resource's required inputs are all satisfied), and ensures that the functional decomposition reaches an appropriate level of detail. It also provides detailed information regarding the allocation of service functions to resources, and the flow of resources between service functions. It is important to note that one usage of the SvcV-4 can support a net-centric (service-oriented) implementation in describing the producing services and consuming services.

In the case of the SvcV-4 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology; however, it does not provide insight into some of the possible construction methods indicating that either a

Taxonomic Service Functional Hierarchy or a Data Flow Diagram are suitable methods for depicting a SvcV-4.

SvcV-4:

- *Block Definition Diagram:*
    - The block definition diagram (bdd) is used to define the characteristics of blocks in terms of structural and behavioral features, and the relationships between the blocks, such as their hierarchal relationship. Extensions to the block definition diagram are used to define parametric constraints and also to show a hierarchal view of activities.
    - The block definition diagram is a suitable tool that would prove helpful in developing the SvcV-4 model for the following reasons:
        - The block is the modular unit of structure in SysML that is used to define a number of items including the type of system/service (resource), and the items (data) that flows through the system/service.
        - Within a bdd, ports are structural features that describe the points at which a block interacts with other blocks. Flow ports are relevant to the SvcV-4 model, they specify what can flow in and out of blocks. Item flows can be used to describe what flows on the connectors between ports.
- *Internal Block Diagram:*
    - The internal block diagram (ibd) is used to describe the internal structure of a block in terms of how its parts are interconnected.
    - The internal block diagram is a suitable tool that would prove helpful in developing the SvcV-4 model for the following reasons:
        - Within an ibd, ports are structural features that describe the points at which a block interacts with other blocks and are used to connect the internal parts of a higher level block.

Again, flow ports are relevant to the SvcV-4 model, and item flows can be used to describe what flows on the connectors between ports and parts.

- The ibd will ensure that the decomposition from the higher level bdd will reach the appropriate level of detail.

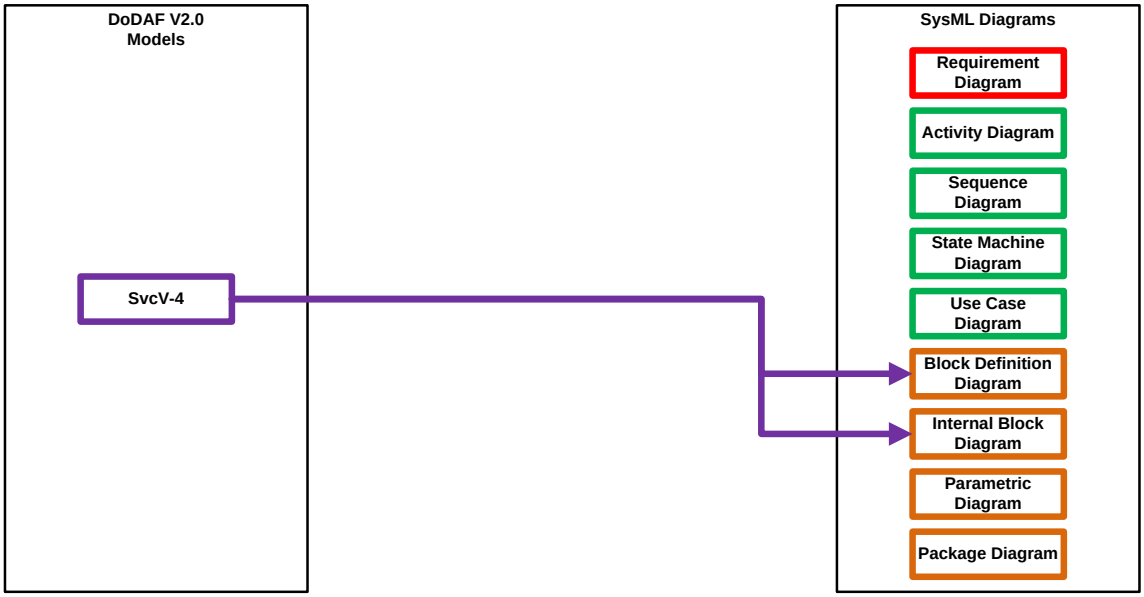

Figure 33: SVcV-4 to SysML Mapping

Additional Information:

The intended usage of the SvcV-4 DoDAF-described model is:

- Description of task workflow.
- Identification of functional service requirements.
- Functional decomposition of services.
- Relate human and service functions.

SvcV-5 Operational Activity to Services Traceability Matrix

The SvcV-5 DoDAF-described model provides a mapping of services (activities) back to operational activities (activities). More specifically, the SvcV-5 model presents the mapping of service functions (and, optionally, the capabilities and performers that provide them) to operational activities and thus identifies

the transformation of an operational need into a purposeful action performed by a service solution. During requirements definition, the SvcV-5 plays a particularly important role in tracing the architectural elements associated with system requirements to those associated with user requirements. The relationship between operational activities and service functions can be expected to be many-to-many (i.e., one activity may be supported by multiple systems, and one system may support multiple activities).

In the case of the SvcV-5 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology; however, it does provide guidance for the creation of this matrix. The SvcV-5 is generally presented as a matrix of the relationship between service functions and activities. The SvcV-5 can show requirements traceability with Operational Activities on one axis of a matrix, the System Functions on the other axis, and with an X, date, or phase in the intersecting cells, where appropriate. An alternate version of the tabular SvcV-5 model can allow the implementation status of each system to be shown (Refer to http://cio-nii.defense.gov/sites/dodaf20/services-5.html for further details).

SvcV-5:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-5 DoDAF described model.
- A matrix is to be developed using the SysML tool, the OV-5a and OV-5b model diagrams, and the SvcV-4 model diagram.

Additional Information:

The intended usage of the SvcV-5 DoDAF-described model is:

- Tracing service functional requirements to user requirements.
- Tracing solution options to requirements.
- Identification of overlaps or gaps.

SvcV-6 Services Resource Flow Matrix

The SvcV-6 DoDAF-described model provides details of service resource flow elements being exchanged between services and the attributes of that exchange. More specifically, the SvcV-6 model specifies the characteristics of the service resource flows exchanged between services with emphasis on resources crossing the service boundary. The SvcV-6 focuses on the specific aspects of the service resource flow and presents the service resource flow content in a tabular format. In addition, this model is useful in support of net-centric (service-oriented) implementation of services. In a net-centric implementation, not all the consumers are known and this model emphasizes the focus on the producer and service resource flow.

In the case of the SvcV-6 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it does not prescribe the column headings in a SvcV-6 Matrix. It merely indicates that a tabular format is required. In addition, it should be noted that the focus of SvcV-6 is on how the Service Resource Flow exchange is affected, in service-specific details covering periodicity, timeliness, throughput, size, information assurance, and security characteristics of the resource exchange. In addition, for Service Resource Flow of data, their format and media type, accuracy, units of measurement, applicable system data standards, and any DIV-3 Physical Data Models are also described or referenced in the matrix.

SvcV-6:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-6 DoDAF described model.
- A matrix is to be developed using the SysML tool, the OV-3 model diagram, and the SvcV-4 model diagram.
- Traceability is also needed back to the OV-2 and OV-3 DoDAF models.

Additional Information:

The intended usage of the SvcV-6 DoDAF-described model is:

- Detailed definition of Resource Flows.

SvcV-7 Services Measures Matrix

The SvcV-7 DoDAF-described model presents measures (metrics) of Services Model elements for the appropriate timeframe(s). More specifically, the SvcV-7 model expands on the information presented in a SvcV-1 by depicting the characteristics of the resources in the SvcV-1. One of the primary purposes of SvcV-7 is to communicate which measures are considered most crucial for the successful achievement of the mission goals assigned. In addition, this model is useful in support of net-centric (service-oriented) implementation of services. It is to be expected that this model is updated throughout the specification, design, development, testing, and possibly even its deployment and operations lifecycle phases.

In the case of the SvcV-7 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. The only guidance provided relating to the creation of the SvcV-7 matrix is that it is typically a table listing user defined measures (metrics) with a time period association.

SvcV-7:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-7 DoDAF described model.
- A matrix is to be developed using the SysML tool and the SvcV-1 model diagram.

Additional Information:

The intended usage of the SvcV-7 DoDAF-described model is:

- Definition of performance characteristics and measures (metrics).
- Identification of non-functional requirements.

SvcV-8 Services Evolution Description

The SvcV-8 DoDAF-described model describes the planned incremental steps toward migrating a suite of services to a more efficient suite, or toward evolving current services to a future implementation. More specifically, the SvcV-8 model can describe historical (legacy), current, and future capabilities against a timeline. The model shows the structure of each resource, using similar modeling elements as those used in SvcV-1. Interactions which take place within the resource may also be shown. In addition, this model is useful in support of net-centric (service-oriented) implementation of services. This model can present a timeline of services evolve or are replaced over time, including services that are internal and external to the scope of the architecture.

In the case of the SvcV-8 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it does not provide guidance relating to the creation/construction of the SvcV-8 model.

SvcV-8:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-8 DoDAF described model.
- An evolutionary timeline (graphical accompanied by a textual description) is to be developed. It should detail the structure of each resource, using similar modeling elements as those used in SvcV-1. Interactions which take place within the resource may also be shown.

Additional Information:

The intended usage of the SvcV-8 DoDAF-described model is:

- Development of incremental acquisition strategy.
- Planning technology insertion.

SvcV-9 Services Technology & Skills Forecast

The SvcV-9 DoDAF-described model describes emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future service development. More specifically, the SvcV-9 model provides a summary of emerging technologies and skills that impact the architecture. The SvcV-9 provides descriptions of relevant: emerging capabilities, industry trends, predictions (with associated confidence factors) of the availability and readiness of specific hardware and software services, and current and possible future skills. In addition, the SvcV-9 model also includes an assessment of the potential impact of these items on the architecture. In addition, this model is useful in support of net-centric (service-oriented) implementation of services. ***Note:*** Given the future-oriented nature of this model, forecasts are typically made in short, mid and long-term timeframes, such as 6, 12 and 18-month intervals.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does provide insight into some of the possible construction methods for SvcV-9 indicating that it can be presented in a table, timeline, or a herringbone diagram.

SvcV-9:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-9 DoDAF described model.
- An evolutionary timeline (with a tabular, timeline, or a herringbone diagram format) is to be developed. New technologies and skills are tied to specific time periods, which can correlate against the time periods used in SvcV-8 milestones and linked to the capability phases.

Additional Information:

The intended usage of the SvcV-9 DoDAF-described model is:

- Forecasting technology readiness against time.
- HR Trends Analysis.
- Recruitment Planning.
- Planning technology insertion.
- Input to options analysis.

SvcV-10a Services Rules Model

The SvcV-10a DoDAF-described model is one of three models used to describe service functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation. The constraints are specified in text and may be functional or structural (i.e., non-functional). More specifically, the SvcV-10a model describes the rules that control, constrain or otherwise guide the implementation aspects of the architecture. Service Rules are statements that define or constrain some aspect of the business, and may be applied to performers, resource flows, service functions, service ports, and data elements.

In the case of the SvcV-10a DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology. No guidance is provided relating to the creation of the SvcV-10a model.

SvcV-10a:

- There is no equivalent / applicable SysML diagram that can help in the creation of the SvcV-10a DoDAF described model.
- A model (table) is to be developed that will feed into other models created using the SysML tool. The SvcV-10a should provide a listing of the complete set of rules with a reference to any models that they affect.
- The SvcV-10a model should be created using a format that can easily be imported to the SysML tool e.g. XML.

Additional Information:

The intended usage of the SvcV-10a DoDAF-described model is:

- Definition of implementation logic.
- Identification of resource constraints.

SvcV-10b Services State Transition Description

The SvcV-10b DoDAF-described model represents one of three models used to describe service functionality. It identifies responses of services to events. More specifically, SvcV-10b represents the sets of events to which the resources in the Activities respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action. The SvcV-10b models state transitions from a resource perspective, with a focus on how the resource responds to stimuli (e.g., triggers and events). The SvcV-10b can be used to describe the detailed sequencing of service functions described in SvcV-4 Services Functionality Description.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does provide insight into some of the possible construction methods for SvcV-10b indicating that the SvcV-10b model is based on the statechart diagram.

SvcV-10b:

- *State Machine Diagram:*
    - The state machine diagram (stm) is used in SysML to describe the state-dependent behavior of a block throughout its lifecycle in terms of its states and the transitions between them.
    - The state machine diagram is a suitable tool that would prove helpful in developing the SvcV-10b model for the following reasons:
        - SvcV-10b is based on the statechart diagram.
        - State machine diagrams are sometimes referred to as state charts or state diagrams.

- stm diagrams facilitate the representation of states, and transitions (triggers, guards, effects).
- Call events within the stm facilitate the response to operational calls.

Additional Information:

The intended usage of the SvcV-10b DoDAF-described model is:

- Definition of states, events and state transitions (behavioral modeling).
- Identification of constraints.

SvcV-10c Services Event-Trace Description

The SvcV-10c DoDAF-described model is one of three models used to describe service functionality. It identifies service-specific refinements of critical sequences of events described in the Operational Viewpoint. More specifically, the SvcV-10c model specifies the sequence in which resource flow elements are exchanged in context of a resource or service port. The components of a SvcV-10c include f functional resources or service ports, owning performer, as well as the port which is the subject for the lifeline. Each Event/Trace diagram should have an accompanying description that defines the particular scenario or situation.

Although DoDAF v2.0 does not endorse a specific activity modeling methodology, it does provide insight into some of the possible construction methods for SvcV-10c. The Services Event-Trace Descriptions are sometimes called sequence diagrams, event scenarios or timing diagrams. Sequence diagrams would therefore be an appropriate approach for developing the SvcV-10c model.

SvcV-10c:

- *Sequence Diagram:*
  - The sequence diagram (sd) is used to represent the interaction between structural elements of a block, as a sequence of message

exchanges.

- The sequence diagram is a suitable tool that would prove helpful in developing the SvcV-10c model for the following reasons:
  - It is the principle diagram used to describe activities and their associated input/output flows (object flow) whether they are discrete or continuous.
  - It can easily be used to depict a sequence of events.
  - It can easily be used to specify the sequence in which resource flow elements are exchanged in context of a resource or system/service port.

Additional Information:

The intended usage of the SvcV-10c DoDAF-described model is:

- Analysis of resource events impacting operation.
- Behavioral analysis.
- Identification of non-functional service requirements.

STANDARDS VIEWPOINT

The standards viewpoint articulates applicable Operational, Business, Technical, and Industry policy, standards, guidance, constraints, and forecasts. These sets of rules/standards can be captured at the enterprise level and applied to each solution, while each solution's architectural description depicts only those rules pertinent to architecture described. Its purpose is to ensure that a solution satisfies a specified set of operational or capability requirements.

StdV-1 Standards Profile

The StdV-1 DoDAF-described model presents a listing of standards that apply to solution elements. More specifically, the StdV-1 defines the technical, operational, and business standards, guidance, and policy applicable to the architecture being described. As well as identifying applicable technical standards, the StdV-1 also documents the policies and standards that apply to the operational or business context. With associated standards with other elements of the architecture, a distinction is made between applicability and conformance (compliance). If a standard is applicable to a given architecture, that architecture need not be fully conformant (compliant) with the standard. The degree of conformance (compliance) to a given standard may be judged based on a risk assessment at each approval point.

In the case of the StdV-1 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it also does not suggest any approach for the construction of the StdV-1 model.

StdV-1:

- There is no equivalent / applicable SysML diagram that can help in the creation of the StdV-1 DoDAF described model.
- The StdV-1 model should be created in a structured textual format and each standard listed in the model (profile) should be associated with the elements that implement or use the standard (SV-1, SV-2, SV-4, SV-6, SvcV-1, SvcV-2, SvcV-4,

SvcV-6, DIV-2, and DIV-3).

Additional Information:

The intended usage of the StdV-1 DoDAF-described model is:

- Application of standards (informing project strategy).
- Standards compliance.

StdV-2 Standards Forecast

The StdV-2 DoDAF-described model presents a description of emerging standards and potential impact on current solution elements, within a set of time frames. More specifically, the StdV-2 contains expected changes in technology-related standards, operational standards, or business standards and conventions, which are documented in the StdV-1 model. The StdV-2 model complements and expands on the StdV-1Standards Profile model and should be used when more than one emerging standard time-period is applicable to the architecture. One of the prime purposes of this model is to identify critical technology standards, their fragility, and the impact of these standards on the future development and maintainability of the architecture and its constituent elements.

In the case of the StdV-2 DoDAF-described model, DoDAF v2.0 does not endorse a specific activity modeling methodology, and it also does not suggest any approach for the construction of the StdV-2 model.

StdV-2:

- There is no equivalent / applicable SysML diagram that can help in the creation of the StdV-2 DoDAF described model.
- The StdV-2 model should be created in a structured textual format
- StdV-2 delineates the standards that potentially impact the relevant system and service elements (SV-1, SV-2, SV-4, SV-6, SvcV-1, SvcV-2, SvcV-4, SV-6, and DIV-2) and relates them to the time periods that are listed in the SV-8, SvcV-8, SV-9,

and SvcV-9.

Additional Information:

The intended usage of the StdV-2 DoDAF-described model is:

- Forecasting future changes in standards (informing project strategy).

APPENDIX C: PYTHON SOURCE CODE FOR SV-2

The following is the source code for the Python programs that were used to produce the SV-2 for the CIMT project. This source code makes use of a Python library known as NodeBox (http://nodebox.net/code/index.php/Home).

build_edgelist.py

```
#!/usr/bin/env python
import csv
f = csv.reader(open('SV-3.csv', 'rb'))
headers = ['%s:%s' % i for i in zip(f.next(), f.next())[2:]]
for line in f:
    node = '{0}:{1}'.format(line[0], line[1])
    for i, e in enumerate(line[2:]):
        if not e: continue
        print '{0},{1},{2}'.format(node, headers[i], e)
```

build_graph.py

```
import os
os.chdir('/Users/devin/Documents/ia/svn/RT24/sv2')
size(2048, 2048)
graph = ximport("graph")
g = graph.create(iterations=1000, distance=4)
for edge in open('edgelist.csv').readlines():
    node1, node2, edge_type = edge.strip().split(',')
    if node1.split(':')[0] == node2.split(':')[0]:
        w = 1
    else:
        w = 0
    g.add_edge(node1, node2, weight=w, label=edge_type)
categories = set()
for n in g.nodes:
    n.category, n.label = n.id.split(':')
    categories.add(n.category)
    n.style = n.category
colors = {
          'Assets': color(0.351, 0.236, 0.625, 1.000), #593c9f
```

```
            'Script': Color(0.251, 0.937, 0.992, 1.000), #40eefc
            'GameObject': Color(0.290, 0.189, 0.530, 1.000), #493087
            'Unity GUI': Color(0.926, 0.729, 0.614, 1.000), #ecb99c
            'Transform': Color(0.351, 0.631, 0.217, 1.000), #59a037
            'Particle Components': Color(0.814, 0.442, 0.293, 1.000), #cf704a
            'Particle Rendering': Color(0.011, 0.235, 0.874, 1.000), #023bde
            'Animation': Color(0.866, 0.542, 0.549, 1.000), #dc8a8b
            'Mesh': Color(0.011, 0.706, 0.495, 1.000), #02b47e
            'Audio': Color(0.441, 0.967, 0.157, 1.000), #70f628
            'Physics': Color(0.325, 0.549, 0.553, 1.000), #528b8d
            'Rendering Component': Color(0.722, 0.701, 0.337, 1.000) #b8b256
}
for cat in categories:
    s = g.styles.create(cat)
    s.fill = colors[cat]
g.solve()
g.draw(directed=True)
print
```

APPENDIX D: REFERENCES

Abowd, G. D., Allen, R. J., & Garlan, D. (1995). Formalizing Style to Understand Descriptions of Software Architecture. *ACM Transactions of Software Engineering and Methodology*, *4*, 319-364.

David, W., Sadowski, R. P., & Sturrock, D. T. (2004). *Simulation with arena.* McGraw-Hill Higher Education.

Object Management Group. (2008). Systems Modeling Language (SysML) version 1.1. Standards Specification Document formal/2008-11-02. Retrieved from www.omg.org/spec/SysML/1.1/

Object Management Group. (2011). Business Process Model and Notation (BPMN 2.0). OMG Specification formal/11-01-03. Retrieved from http://www.omg.org/spec/BPMN/2.0/

Opdahl, A. L., & Henderson-Sellers, B. (2002). Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model. *Software Systems Modeling*, *1*, 43-67.

Rosemann, M., Recker, J., Indulska, M., & Green, P. (2006). A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars. In E. Dubois & K. Pohl (Eds.), *Advanced Information Systems Engineering - CAiSE 2006* (Vol. 4001, pp. 447-461). Luxembourg, Grand-Duchy of Luxembourg: Springer.

U.S. Department of Defense. (2009). DoD Architecture Framework Version 2.0. Volume 1: Introduction, Overview and Concepts. Manager's Guide. *2.0*,.

Wand, Y., & Weber, R. (1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Journal of Information Systems*, *3*(4), 217-237.

zur Muehlen, M., & Recker, J. (2008). How much language is enough? Theoretical and Practical Use of the Business Process Management Notation. *Conference on Advanced Information Systems Engineering (CAiSE '08)*,.