


Article

# Architecture Design Space Generation via Decision Pattern-Guided Department of Defense Architecture Framework Modeling

Zhemei Fang <sup>1,2,\*</sup> , Xuemeng Zhao <sup>1,2</sup> and Fengyun Li <sup>3</sup>

<sup>1</sup> School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China; m202173115@hust.edu.cn

<sup>2</sup> National Key Laboratory of Science & Technology on Multi-Spectral Information Processing, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>3</sup> Wuhan Second Ship Design and Research Institute, Wuhan 430205, China; lfynevermore@126.com

\* Correspondence: zmfang2018@hust.edu.cn

**Abstract:** The importance of architecture design keeps increasing as the complexity of systems and system-of-systems (SoSs) continues rising. While the architecture frameworks such as the Department of Defense Architecture Framework (DoDAF) are commonly used to guide architecture design, many perspectives are still hindering their effective use. Instead of generating a set of architecture description models probably only for satisfying the milestone review, the architecture frameworks should be used to organize the vague or incomplete information, identify and formulate the decision-making problem, and guide the architecture decision-making. Unfortunately, the decision points are hidden in the architecture models and the ambiguity often leads to a confusion of whether the architecture models are built incorrectly due to the lack of modeling experience or the lack of adequate decision analysis. Therefore, this paper identifies the key decision points and decision types during the architecture model development based on the DoDAF. Plus, this paper proposes a set of decision patterns and a guide to their use to provide qualitative decision analysis for developing architecture models and generating alternatives. An illustrative example to anti-submarine SoSs demonstrates the process of applying the decision patterns to the DoDAF model's development and the generated architecture alternatives.

**Keywords:** architecture design space; decision pattern; department of defense architecture framework; system-of-systems



**Citation:** Fang, Z.; Zhao, X.; Li, F. Architecture Design Space Generation via Decision Pattern-Guided Department of Defense Architecture Framework Modeling. *Systems* **2024**, *12*, 336. <https://doi.org/10.3390/systems12090336>

Academic Editors: Gregory S. Parnell and Eric Specking

Received: 22 July 2024

Revised: 24 August 2024

Accepted: 27 August 2024

Published: 31 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Architecture is vital at the conceptual phase to reduce both the developmental and operational complexity of complex systems and system-of-systems (SoSs). The role of architecture has been emphasized in 1991 when the U.S. Navy developed its vision for combat systems in 2030 [1]. As the complexity of systems and SoSs keeps increasing, architecture design has received significant attention in the most recent two decades, especially in the defense, aerospace, and systems engineering domains. The ISO/IEC/IEEE 42010:2011 [2] defines architecture as the structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.

The methods on architecture design over the past two decades can be categorized into two primary directions [3]. The first direction focuses on developing architecture descriptions that portray a system/SoSs from different dimensions, each of which describes the key elements from a specific viewpoint (e.g., an operational viewpoint, or a system viewpoint). Many architecture frameworks have been released to lay out the viewpoints and the elements in each viewpoint. An exemplified, one is the U.S. Department of Defense Architecture Framework (DoDAF). Many defense-related institutions adopt the DoDAF

for guiding architecture model development. For instance, DoDAF models are required products of acquisition reviews by the U.S. Joint Capabilities Integration and Development System (JCIDS). The architecture models are usually evaluated by simulating the corresponding executable models, following the seminar work of Levis and Wagenhals [4]. The second direction focuses on the optimization-based methods for formulating the SoSs architecture selection problems mathematically and optimizing the architecting decisions. The decisions include, but are limited to, design option selection, system composition, activity planning, function allocation, etc.

It is interesting that most studies on these two directions are disconnected [3,5,6], and thus both sides receive complaints. While the latter one is often complained about over the over-simplification or even distortion of the original problems to obtain reasonable results from the algorithms, the former one has been questioned a lot in practice on the meaning of developing the architecture models since all the elements of different views seem to be already given [7,8]. In fact, architecture design should be an integrated process of qualitative analysis and quantitative analysis. Architecture frameworks should be used to organize vague or incomplete information, identify and formulate the decision-making problem, and guide the architecture decision-making instead of generating a set of architecture description models probably only for the completing of assignments by superiors. Unfortunately, the decision points are hidden in the architecture models and the ambiguity often leads to a big confusion of whether the architecture models are built incorrectly due to the lack of modeling experience or the lack of adequate decision analysis.

Therefore, using the DoDAF as an exemplified framework, this paper aims to identify the decision points and apply a set of decision patterns to guide the development of architecture models by clarifying the known information and candidate options. The key contribution of this paper is twofold: (1) it proposes an integrated process of model development and qualitative decision analysis via decision patterns to facilitate the DoDAF-based architecture model development; and (2) the proposed decision patterns-guided DoDAF modeling method can help generate a set of architecture alternatives and a corresponding architecture design space.

This paper contains five sections. Section 2 introduces the background and related studies. Section 3 proposes the decision patterns to support DoDAF-based architecture modeling. In Section 4, we apply the decision pattern guided-DoDAF modeling method to an ASW SoS example. Section 5 concludes the entire paper and indicates future work.

## 2. Background and Related Work

Architecture frameworks are defined as “conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders” [2] in ISO/IEC/IEEE 42010:2011. After the U.S. DoD released the DoDAF, the British Ministry of Defense Architecture Framework (MODAF) and the NATO Architecture Framework (NAF) were launched as well. The DoDAF and the MODAF were merged into a combined metamodel in the Unified Profile for DoDAF/MODAF (UPDM). To further support other frameworks, the Object Management Group (OMG) released the first version of the Unified Architecture Framework (UAF) in 2017.

The DoDAF has been an exemplified architecture framework in the domain. In order to develop the SoSs architecture models using the DoDAF, Levis et al. [9,10] proposed a series of architecture design methods, including a structured method, an object-oriented method, and a service-oriented method. Subsequent to Levis' work, many studies follow a similar pattern to conduct the DoDAF-based architecture design. Since the DoDAF does not provide a specific method to support architecture design, institutions and researchers around the world have developed many different processes and methods to guide architecture model development. Piaszczyk [11] presented a methodology for Model-Based Systems Engineering (MBSE) utilizing the guidelines provided by the DoDAF. The proposed method helps derive the operational, functional, system, and physical requirements from the system model represented with DoDAF views. Williams and Stracener [12] identified the steps

for tailoring DoDAF 2.0 for the development of a program organizational architectural framework (POAF). Specifically, the authors examined and determined the information required for different stakeholders to make decisions. Later, the authors [13] proposed an analytical model to support the decision-making process of the owner viewpoint in a POAF. Handley [14] proposed a six-step design methodology for fit-for-purpose human views and the method focuses on data collection and relationship identification. Amisshah and Handley [15] developed a process including reference modeling, conceptual modeling, executable modeling, and evaluation for DoDAF-based data-centric architecture modeling. Shaked and Reich [16] introduced a Process-Oriented Viewpoint Engineering framework (PROVE) to plan complex development efforts as well as improve the coordination of multiple development efforts in SoS development scenarios. After the OMG published the UAF, many researchers turned their attention to the UAF. Abhaya [17] developed a UAF-based MBSE method to build an SoS model. Liu et al. [18] developed a top-down military SoS design using MBSE based on the UAF. Bankauskaite et al. [19] proposed an automated trade study analysis process for the SoS architecture developed in the UAF models. Martin and O'Neil [20] proposed an enterprise architecture guide for the UAF, and this guide later became one of the OMG's official supportive documents for the UAF [21].

Overall, the studies aimed to either directly apply the DoDAF or the UAF to an example SoS following a general modeling procedure to obtain architecture products, or tailoring the DoDAF or the UAF to a specific use such as supporting program organization development, planning development efforts, adding a fit-for-purpose view (e.g., human, owner), and so on. How to develop architecture models for an example SoS still receives limited attention. While it is true that the DoDAF and the UAF are used to collect architecture data to make informed decisions, the data are not directly available. What is worse, the relationships between architecture data/models and architecture decisions are not clear. As a result, in practice, even though the architecture design method (including modeling procedures) and the modeling tools are available, the decision-makers (e.g., the SoS architect, project manager) are still confused about the purpose of architecture model development, what should be included in the architecture models, how to deal with the ambiguous information during modeling, how to use the developed architecture models, when and how to make decisions, etc.

A number of researchers work on providing decision support methods to develop architecture models based on the DoDAF or other frameworks. Sohn et al. [22] propose to generate architecture solutions by utilizing a fuzzy-semantic similarity measure to identify the appropriate views using case-based reasoning. Zhang et al. [23] developed a method to support architecture model development towards optimal architectures. Specifically, the authors propose a semi-automatic optimization method that covers the clustering of operational activities and service identification and improvement to support the SvcV-5 (Operational Activity to Services Traceability Matrix) construction. Dai et al. [6] developed a decision support framework with portfolio optimization and dependency analysis to support alternative generation and modeling. Fang et al. [24] proposed a set of decision patterns and associated mathematical formulations to guide DoDAF modeling, based on the decision pattern concept developed by Selva, Cameron, and Crawley [25]. However, these studies often analyze a particular perspective to support DoDAF modeling instead of providing a systematic process to guide architecting and modeling.

Meanwhile, another group of researchers aim to use DoDAF-based architecture models to guide the generation of SoS design alternatives or support the SoS analysis. This process often belongs to design space exploration, a process to analyze several "functionally equivalent" implementation alternatives, which meets all design constraints in order to identify the most suitable design solutions based on quality metrics [26,27]. Early work by Griendling and Mavris [28] proposed an architecture-based approach to identify SoS alternatives. Franzen et al. [29,30] apply architecture frameworks to break down the SoS needs and use ontologies and description logic reasoning to narrow down the design space. Guariniello et al. [31] used DoDAF models to provide information for a set of analytical

methods for SoS development. Ali et al. [32] proposed a holistic approach for architecture design space characterization by integrating decision alternatives in functional, physical, and allocational design spaces. Huff et al. [33] explored the links between DoDAF-based models and an integer linear program for rigorously evaluating system vulnerability.

The idea is also quite popular recently in the field of MBSE that concerns the system architecture and does not particularly employ the DoDAF. Specking et al. [5,34] integrated the architecture description models and trade-off decisions or optimization by introducing and combining the descriptive, predictive, and prescriptive analytics in the early design phase. Specking and Parnell [35] later explored the MBSE capabilities and integrated models needed to perform decision trade-off analysis. Timperley et al. [36] investigated the potential to use MBSE for design exploration and employed the concept of generative design [37] and optimization to rapidly generate and assess new designs. Bussemaker et al. [38] proposed a method for the automated generation of architecture candidates by capturing architecting decisions related to function assignment and component structure, and quantitative evaluation strategies. She et al. [39] presented an optimization-based layout design method of an unmanned aerial vehicle radar cabin using KARMA language based models. Paape et al. [40] proposed a specification language for an automated design space exploration process including the steps of specifying, modeling, simulating, and evaluating a design. Overall, the focuses of these studies are often on the appropriate use of MBSE models to guide decision-making, instead of developing an integrated and interactive bidirectional process for architecture design.

In summary, the SoSs architecture design should be an integrated process of architecture modeling and architecture decision-making. The architecture models based on the DoDAF or the UAF capture the key aspects of an architecture. Decision analysis, both qualitative analysis and quantitative analysis, is necessary and crucial to capture these highly interconnected aspects. The lack of decision support for architecture description models' development could lead to modeling failures such as wrong models, incomplete models, and obsolete models. However, the possible decision points and decision problems involved in the architecture models are often ambiguous. Thus, based on our previous work [24], taking the DoDAF as an exemplified architecture framework, this paper identifies the key decision points and decision types, and proposes a set of decision patterns to provide qualitative decision analysis for developing architecture models and generating architecture design spaces of alternatives that are represented by DoDAF models. Considering architecture design space generation as a step in the design space exploration process, this paper proposes a novel integrated and interactive process to support architecture modeling and architecting decision-making at the same time.

### 3. Decision Patterns for DoDAF Modeling

DoDAF includes eight viewpoints in total and the most frequently used ones are the capability viewpoint (CV), operational viewpoint (OV), system viewpoint (SV), services viewpoint (SvcV), and the data and information viewpoint (DIV). To support the modeling of the views (models) under these viewpoints, we identify the decisions points hidden in the models, and consider the downselecting, assigning, partitioning, permuting, and connecting patterns to guide the modeling. Table 1 lists the decision points and patterns in the frequently used DoDAF models.

The downselecting pattern is for choosing a subset of entities from a set of candidate entities. The partitioning pattern concerns the partition of a set of entities into subsets that are mutually exclusive and collectively exhaustive. The assigning pattern is used for assigning each entity from one set to a subset of entities from the other set. The permuting pattern aims to solve an ordering or permutation of a set while the connecting pattern concerns connecting entities (i.e., nodes) by using edges in a graph.

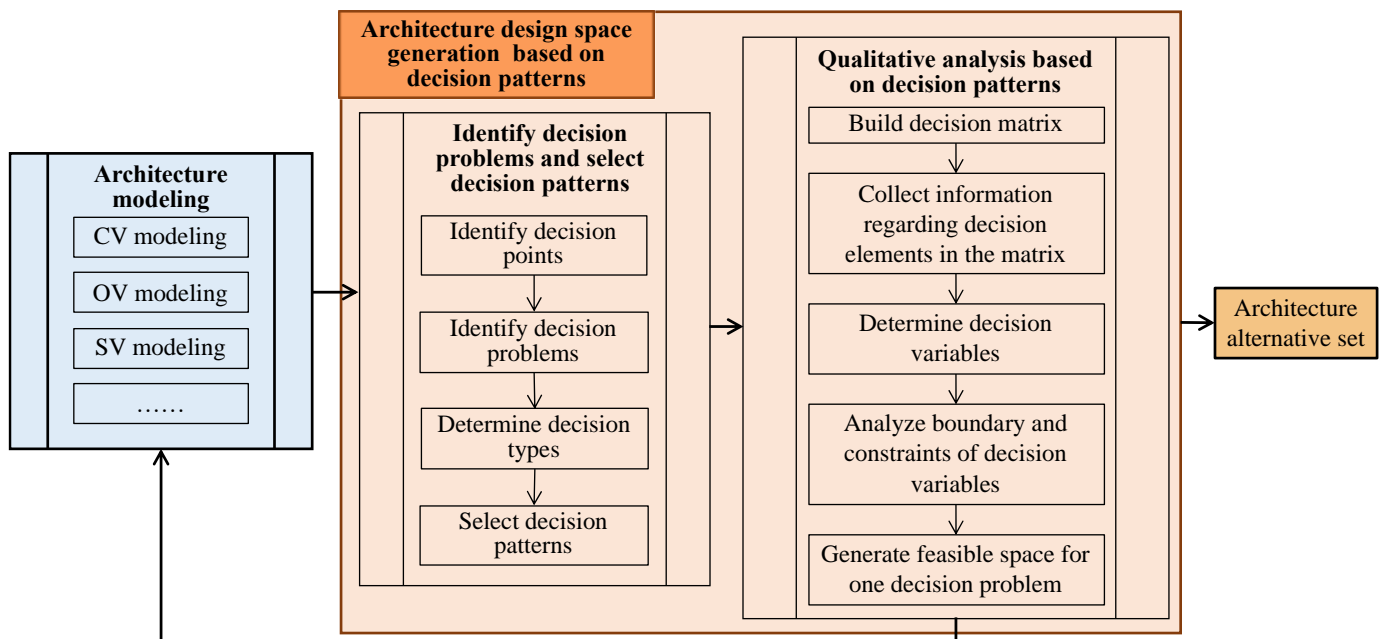
The five decision patterns are able to cover the fundamental decisions in the listed DoDAF models. The tasks needing decision support can be grouped into four types: (a) identifying the entities including capabilities, activities, systems, functions, etc.; (b) allo-

cating one set of entities, such as activities, systems, etc., to another; (c) building connections between entities such as the capability, system, and service without order; (d) building connections between entities such as the activity and function with order. Identifying entities requires downselecting and partitioning decisions. The allocation process requires assigning decisions. The connecting tasks require connecting and permuting decisions, respectively, based upon the requirements.

**Table 1.** Decision points and patterns in selected DoDAF models.

DoDAF Models	Decision Points	Decision Patterns
CV-2: Capability Taxonomy	Identifying capabilities; Decomposing capabilities; Identifying metrics	Downselecting, Partitioning
CV-6: Capability to Operational Activities Mapping	Allocating capability to activities	Assigning
OV-5a: Operational Activity Decomposition Tree	Identifying activities; Decomposing activities	Downselecting, Partitioning
OV-5b: Operational Activity Model	Connecting activities in order	Permuting, Connecting
SV-1: Systems Interface Description	Identifying systems; Connecting system components	Downselecting, Partitioning, Connecting
SV-2: Systems Resource Flow Description	Connecting systems	Permuting, Connecting
SV-4a: Systems Functionality Description	Identifying functions; Decomposing functions; Allocating functions to systems	Downselecting, Partitioning, Assigning
SV-4b: Systems Functionality Flow Description	Connecting function activities in order	Permuting, Connecting
SV-5b: Operational Activity to Systems Traceability Matrix	Allocating activities to systems	Assigning
SV-7: Systems Measures Matrix	Identifying metrics	Downselecting

The decision patterns are used to guide the decision-making in the DoDAF-based architecture design. The process to use the decision patterns is shown in Figure 1.



**Figure 1.** Decision space generation process using decision patterns.

### 3.1. Downselecting Pattern

The downselecting pattern is used to deal with the decision-making problem of “selecting entities from a candidate entity set to form a subset”. A candidate entity set  $E = \{e_1, e_2, \dots, e_i, \dots, e_N\}$  is given in advance.  $e_i$  denotes an element in set  $E$  and  $i$  indicates the element index.  $N$  is the quantity of elements.  $x_i$  is a variable indicating whether element  $e_i$  is selected; in other words, whether the selected subset  $F$  include the element  $e_i$ . The decision matrix  $X$  for the downselecting pattern is given in (1).

$$X = [x_1, x_2, \dots, x_i, \dots, x_N], x_i = \begin{cases} 1, & e_i \in F \\ 0, & e_i \notin F \end{cases}, i \in \{x \in R^+ | 1 \leq x \leq N\}, \quad (1)$$

Take SV-1 as an example. Entity refers to systems. Different types of systems could generate different sets of entities, such as a set of detection systems, a set of weapon systems, a set of sonar array systems, and a set of torpedoes.

The downselecting pattern aims to pick out the subset with a good synergy effect between elements. When applying the downselecting pattern, the modelers need to consider the net impact of the positive and negative interactions between elements. The net impact of interactions demonstrates the SoSs trait of emergence. Positive interaction means that the interaction between two elements creates a positive synergistic effect. The positive effect could either be a new capability that cannot be gained by any element alone, or by the improved performance of an existing capability. Negative interaction means that the interaction between two elements leads to a bad synergy effect (e.g., performance degradation). An architecture alternative is expected to have a high positive synergy effect and a low negative synergy effect.

Another perspective is the capability redundancy, which means that different entities in a subset have the same capability. Low capability redundancy is preferred when applying the downselecting pattern.

### 3.2. Partitioning Pattern

The partitioning pattern is used to deal with “partitioning a collection of entities into mutually exclusive and collectively exhaustive subsets”. A candidate entity set  $E = \{e_1, e_2, \dots, e_i, \dots, e_N\}$  is also given in advance. Again,  $e_i$  denotes an element in set  $E$  and  $i$  indicates the element index.  $N$  is the quantity of elements.  $x_{iF}$  is a variable indicating whether element  $e_i$  is selected for subset  $P$  with index  $d$ . The decision matrix  $X$  for the partitioning pattern is given in (2).

$$X = \begin{pmatrix} x_{11} & \dots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{i1} & \dots & x_{iD} \end{pmatrix}, x_{id} = \begin{cases} 1, & e_i \in P_d \\ 0, & e_i \notin P_d \end{cases}, \quad (2)$$

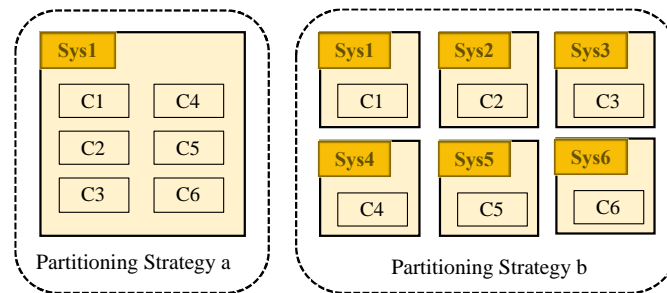
$$i \in \{x \in R^+ | 1 \leq x \leq N\}, d \in \{x \in R^+ | 1 \leq x \leq D\},$$

$D$  is the total number of subsets. The decision matrix  $X$  of the partitioning pattern is quite similar to that of the downselecting pattern, except for the mutually exclusive and collectively exhaustive constraints. That is, each row of the decision matrix must only have one 1 while the rest are set as 0, and each column has at least one 1.

Take SV-1 as an example. A combat ship system could be partitioned into a command and control (C2) system, a detection system, a communication system, and weapon systems. That is because the subsystems in the C2 system, including the target identification system, threat evaluation system, and task planning system, all serve the same purpose of command and control. The subsystems are mutually exclusive and collectively exhaustive.

The partitioning pattern also needs to consider the positive and negative interactions. If the entities that have a positive synergy effect are partitioned into two different subsets, the positive effect might be largely decreased. Likewise, if the entities with a negative synergy effect are put into one subset, the negative effect might impact the architecture

performance. That is, we should neither put all the entities in one basket nor over-partition the entities. Different partitioning strategies impact the architecture attributes and cost. The cost refers to the life-cycle cost including development, operations, and maintenance. The architecture attributes refer to the architecture performance, evolvability, robustness, flexibility, and so on. Figure 2 demonstrates two completely different partitioning strategies. “Strategy a” generates a big system, Sys1, including all the components {C1, C2, C3, C4, C5, C6}. “Strategy b” generates six systems, including Sys1, Sys2, Sys3, Sys4, Sys5, and Sys6, while each system has one component, respectively.



**Figure 2.** Two different partitioning strategies.

Since Sys1 in “Strategy a” has six components, its development cost and development time could both be large due to the interdependency between components. Meanwhile, if all the components have a shared function, “Strategy a” could reduce the capability redundancy and in turn the development cost. However, it might also reduce the system reliability and robustness if the shared function fails. Additionally, the integration of components might help facilitate the interactions to improve performance, or may cause resource (e.g., computation resource, power) competition to degrade the performance. The decision-makers need to strike many tricky balances during the decision-making process.

On the contrary, the distributed “Strategy b” can help avoid the negative interactions between components, and thus reduce the development cost. The cooperation cost during operation might increase though.

Overall, an architect needs to consider the following three aspects when applying the partitioning pattern.

- The net impact of positive and negative synergy effects on the architecture traits and cost;
- The impact of capability redundancy on the performance and cost;
- The impact on development, operation, and maintenance cost.

### 3.3. Assigning Pattern

The assigning pattern is used to deal with the decision problem of “assigning each entity of a set to the entity of another set”. Two entity sets  $E^1 = \{e_1, e_2, \dots, e_i, \dots, e_N\}$  and  $E^2 = \{e'_1, e'_2, \dots, e'_j, \dots, e'_M\}$  are given.  $e_i$  and  $e'_j$  denote elements in set  $E^1$  and set  $E^2$ , respectively.  $i$  and  $j$  indicate the element index in set  $E^1$  and set  $E^2$ , respectively.  $N$  and  $M$  means the quantity of elements in set  $E^1$  and set  $E^2$ , respectively.

The entity in set  $E^1$  can be allocated to the entity in set  $E^2$ . The variable  $x_{ij}$  indicates whether  $e_i \in E^1$  is assigned to  $e_j \in E^2$ . The decision matrix  $X$  is shown in (3).

$$X = \begin{pmatrix} x_{11} & \dots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \dots & x_{ij} \end{pmatrix}, x_{ij} = \begin{cases} 1, & e_i \in E^1 \text{ is assigned to } e_j \in E^2 \\ 0, & e_i \in E^1 \text{ is not assigned to } e_j \in E^2 \end{cases} \quad (3)$$

$i \in \{x \in R^+ | 1 \leq x \leq N\}, j \in \{x \in R^+ | 1 \leq x \leq M\}$ ,

In the assigning pattern, an entity  $e_i$  in set  $E^1$  has to be assigned to at least one entity in set  $E^2$ . An entity  $e'_j$  in set  $E^2$  has to be assigned by at least one entity in set  $E^1$ . That is, any row or column in the decision matrix  $X$  should not be all zeros.

### 3.4. Permuting Pattern

The permuting pattern is used to deal with the decision problem of “connecting entities with order”. A set of entities  $E = \{e_1, e_2, \dots, e_i, \dots, e_N\}$  and a set of position indexes  $Q = \{q_1, q_2, \dots, q_k, \dots, q_M\}$  are given.  $e_i$  denotes an element in set  $E$ ,  $i$  indicates the element index and  $N$  is the quantity of elements.  $q_k$  denotes an element in set  $Q$ ,  $k$  indicates the position index and  $M$  is the quantity of positions. The decision variable  $x_{ik}$  denotes whether entity  $e_i$  is assigned to position  $q_k$ . The positions refer to space, time, etc. The decision matrix  $X$  is given in (4).

$$X = \begin{pmatrix} x_{11} & \dots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{i1} & \dots & x_{iM} \end{pmatrix}, x_{ik} = \begin{cases} 1, & e_i \text{ is allocated to } q_k \\ 0, & e_i \text{ is not allocated to } q_k \end{cases} \quad (4)$$

$i \in \{x \in R^+ | 1 \leq x \leq N\}, k \in \{x \in R^+ | 1 \leq x \leq M\}$ ,

Take OV-5b as an example. If more than one operational activity is assigned to one position from the temporal perspective, i.e., a column in the decision matrix  $X$  has more than one “1”, these operational activities are parallel. As shown in Figure 3, OpAct3 and OpAct6, OpAct2 and OpAct3, and OpAct4 and OpAct6 are all parallel activities. Correspondingly, the 1st, 3rd, and 5th columns of decision matrix  $X$  in (5) have several “1”s.

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

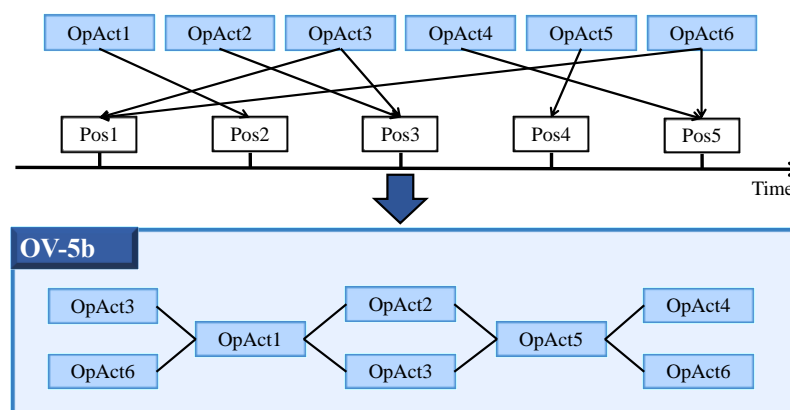


Figure 3. Permuting decisions in OV-5b model.

The permuting decision often appears in the development of OV-5b, SV-2, and SV-4b models. The operational activity sequences are determined primarily by the operational rules. The operational rules refer to the conditions and rules to start or end an operational activity. For example, target information is required to conduct striking activity. An operational activity usually requires resource consumption, thus the activity sequence is almost equivalent to the sequences of resource occupation and release. Take the operational activities in Figure 3 as an example. If the activities of OpAct2 and OpAct3 consume the same type of limited resources, OpAct2 and OpAct3 need to be allocated to successive positions instead of the same position. If OpAct3 needs to use the target information generated by



OpAct1, OpAct3 has to be located after OpAct1. That is, satisfying the operational rules is the top priority of determining the permuting sequence of operational activities.

### 3.5. Connecting Pattern

The connecting pattern is used to deal with the decision problem of “connecting entities in a set with different connecting strategies”. Given an entity set  $E = \{e_1, e_2, \dots, e_i, \dots, e_N\}$ , in which  $e_i$  denotes an element,  $i$  indicates the element index and  $N$  is the quantity of elements. The decision variable  $x_{ij}$  indicates whether two entities  $e_i$  and  $e_j$  are connected with strategy  $r$ , and the resultant decision matrix  $X$  is as in (6).

$$X = \begin{pmatrix} x_{11} & \dots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \dots & x_{ij} \end{pmatrix}, x_{ij} = \begin{cases} 0 & e_i \text{ is not connected to } e_j \\ a_1 & e_i \text{ is connected to } e_j \text{ using strategy 1} \\ \dots & \dots \\ a_r & e_i \text{ is connected to } e_j \text{ using strategy } r \\ \dots & \dots \\ a_R & e_i \text{ is connected to } e_j \text{ using strategy } R \end{cases} \quad (6)$$

$$i, j \in \{x \in R^+ | 1 \leq x \leq N\}, r \in \{x \in R^+ | 1 \leq x \leq R\},$$

$a_r$  refers to the connection strategy and  $R$  denotes the total number of connection strategies. The connection strategy is related to the connection purpose. A typical one is the communication mode between systems, such as simplex communication, duplex communication, etc.

During DoDAF model development, the connecting decisions are often required for determining resource flows between entities and can sometimes be simplified as communication decisions. Take SV-2 as an example. The common communication structures include a star topology, a tree topology, and a mesh topology, as shown in Figure 4.

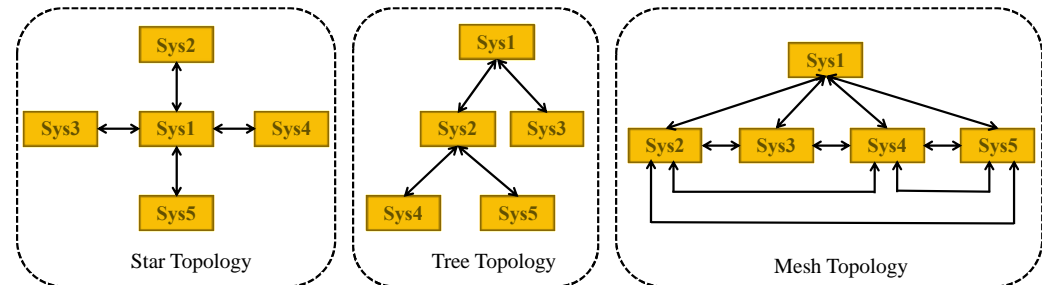


Figure 4. Three types of communication networks.

In the star topology, the hub system (Sys1) is connected to all the other systems (Sys2–Sys5). In the tree topology, systems are connected in a layered format like a tree. In a mesh topology, all systems (Sys1–Sys5) are connected like a network. Link16 adopts this mesh structure. The star topology has a critical hub node which could easily encounter targeted attack and lead to low robustness. The tree structure suffers from similar issues. The mesh topology is better in terms of latency and robustness, but its technical difficulty results in higher cost and lower scalability. There also exists new techniques to increase communication efficiency [41]. Overall, the choice of communication network depends on the aspects of connection cost, latency, reliability, robustness, scalability, and so on.

### 3.6. Combination Pattern

The combination pattern selects some of or all of the previously described decision patterns to solve a set of decision problems. Assume we have  $N$  decision problems, each of which has an alternative set  $O_i$ , as shown in (7).

$$O_i = \{o_{i1}, o_{i2}, \dots, o_{im}, \dots, o_{iM}\}, \quad (7)$$

$$i \in \{x \in R^+ | 1 \leq x \leq N\}, m \in \{x \in R^+ | 1 \leq x \leq M\},$$

$i$  denotes the index of decision problems and  $m_i$  refers to the alternative strategy index of the  $i$ th decision problem.  $M_i$  denotes the total number of alternative strategies of the  $i$ th decision problem. The decision matrix  $X$  of the combination pattern is shown in (8).

$$X = [x_1, x_2, \dots, x_i, \dots, x_N], x_i \in O_i, \tag{8}$$

$x_i$  refers to decisions of the  $i$ th decision problem and  $N$  is the number of decision problems. The combination problem is directly related to the design space generation, and the fundamental principle is decomposition first and aggregation later, as shown in Figure 5.

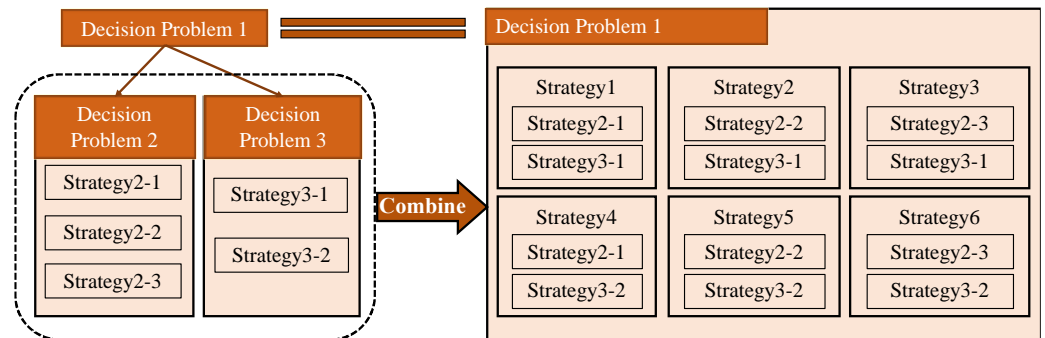


Figure 5. Combination strategies.

#### 4. Illustrative Example

We apply the decision pattern-guided DoDAF modeling approach to the architecture design for a synthetic anti-submarine warfare (ASW) SoS. The available combat resources include a combat ship and the submarine detection and attack systems. Assume that the enemy submarine (i.e., named “sub” for short) has been vaguely detected by the fixed sonar array systems in the ocean.

The ASW SoS has a limited number of combat resources. Other than the fixed sonar array systems, the other candidate ASW resources are listed in Table 2. The combat ship can carry two anti-submarine helicopters at best and each helicopter can take two light-weighted torpedoes at the same time or a sonobuoy array and magnetic anomaly detector (MAD) unmanned aerial vehicle (UAV) at the same time. The technical parameters of the resources are provided and the connection strategy can either be a star topology or a mesh topology. We can apply the decision patterns to support the decision-making during DoDAF model development.

Table 2. Available ASW resources.

Resource Type	Quantity
Combat ship	1
Anti-submarine helicopter	2
Sonobuoy	6
MAD UAV	1
Torpedo	2

##### 4.1. Initial DoDAF Models

Given the available information, the DoDAF models can be developed following the development process shown in Figure 6.

Based on initial information, we can build an initial OV-1 model as shown in Figure 7 that includes a combat ship, communication facilities, detection resources and attack resources to destroy an enemy submarine 120 nautical miles away. The attack resource is a torpedo but the quantity to be used is not known yet. The detection and communication resources are not determined either. Based on the OV-1 model, we can also extract the key

operational performers and build the OV-2 model, as shown in Figure 8. The command and control (C2), submarine detection, submarine attack, and communication nodes are easy to be identified. Since the enemy submarine is 120 nautical miles away, it is necessary to add an payload deployment node.

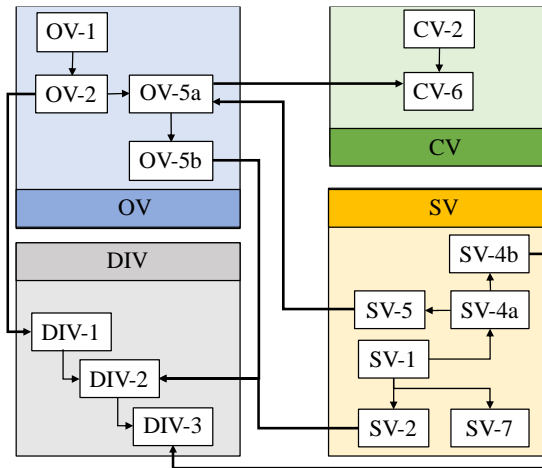


Figure 6. Development process of DoDAF models.

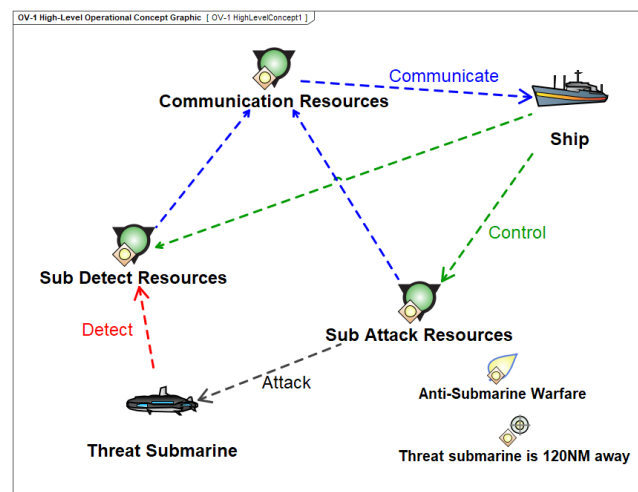


Figure 7. OV-1 model.

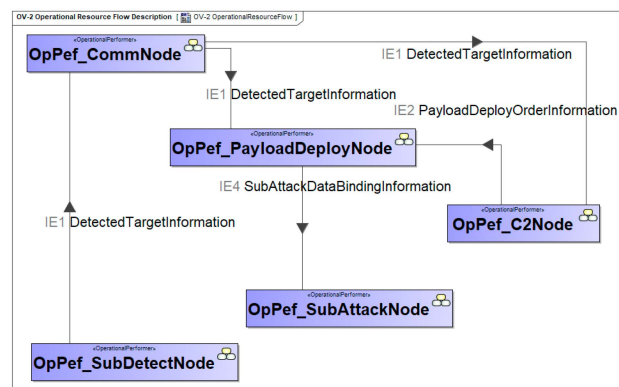


Figure 8. OV-2 model.

No decision points appear during the OV-1 and OV-2 modeling. OV-5 involves decision problems and we will come back to the OV-5 models later. Based on the available equipment resources, we can also build the SV-1 and SV-4 models. Take the torpedo system as an example. A torpedo system can be decomposed into its hull, warhead system, control system, homing system and power system in the SV-1 model, as shown in Figure 9. Figures 10 and 11 illustrate the functions of a torpedo system in the SV-4a model as well as the functional activity flow in the SV-4b model.

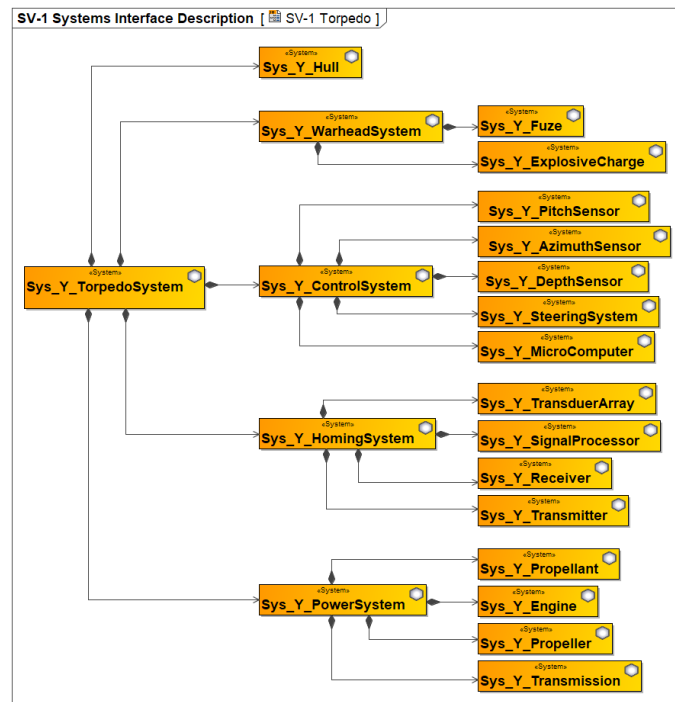


Figure 9. SV-1 model (torpedo system).

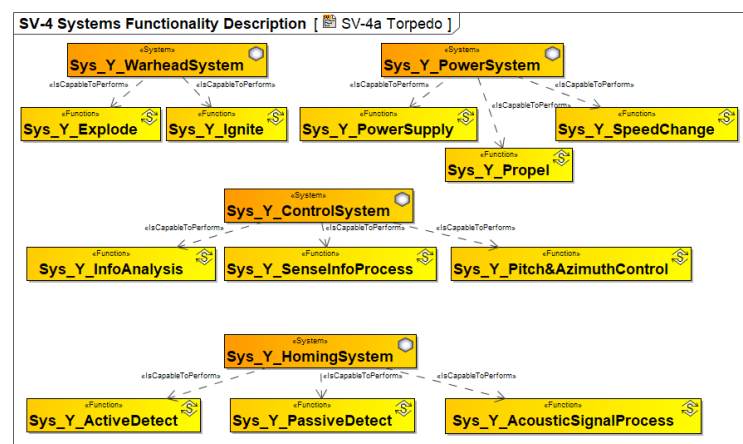


Figure 10. SV-4a model (torpedo system).

After building the SV-1 and SV-4 models of the combat ship, helicopter, torpedo, fixed sonar array system, sonobuoy array system, and MAD UAV system, we continue building the SV-2 model, OV-5 model, and SV-7 model.

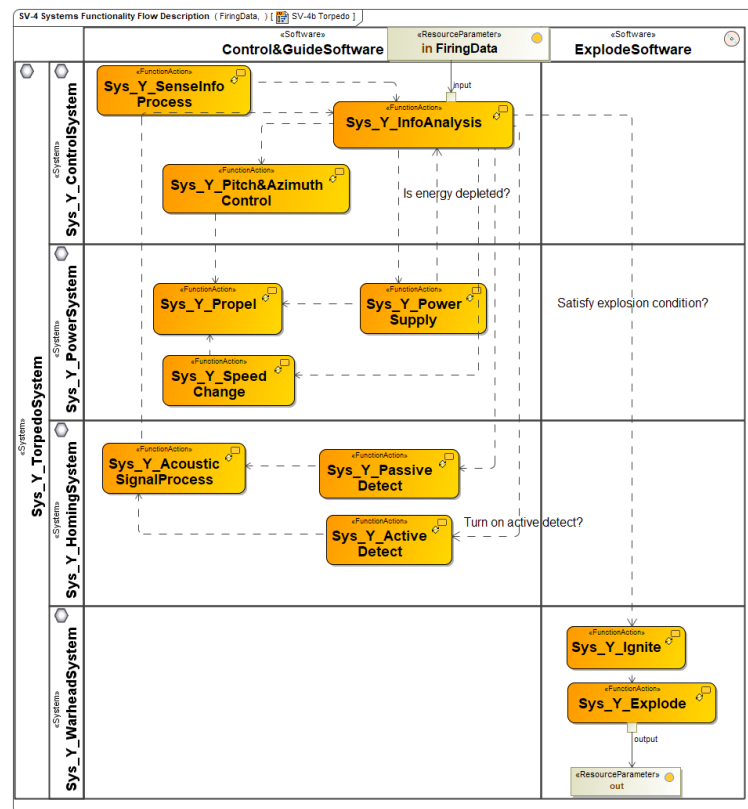


Figure 11. SV-4b model (torpedo system).

4.2. Decision Problems

Based on the initial DoDAF models, we can identify the decision points and decision problems. The pink-colored rectangles in Figure 12 demonstrate the models with decision points. These models include OV-5a, OV-5b, SV-2, SV-7, and SV-5. The corresponding decision problems and decision patterns are illustrated in Table 3.

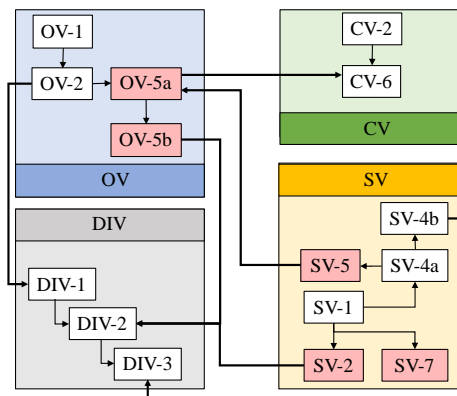


Figure 12. Decision points in DoDAF models of this example.

Table 3. Decision points and patterns in this example.

Views	Decision Problems	Decision Types	Decision Patterns
OV-5a	Identifying operational activities; Decomposing operational activities	Identifying entities	Partitioning
OV-5b	Connecting operational activities in order	Connecting entities in order	Permuting
SV-2	Determining connecting types; Connecting systems	Connecting entities	Connecting
SV-5	Allocating systems to activities	Mapping between entities	Assigning
SV-7	Identifying metrics and attributes	Identifying entities	Downselecting

4.3. Architecture Design Space Generation  
 4.3.1. Downselecting Decisions in SV-7

The SV-7 model collects the system measures of all the candidate resources, as partially listed in Figure 13. Not all the system options will be selected into the ASW SoS architecture and different selections and combinations would generate different alternatives with varying SoS performance. Since the formation of the sonobuoy array and the number of torpedoes are not determined, we will make decisions on these two problems over the SV-7 model.

#	Property Set	Performance Requirement	Measure	Metric
1	TorpedoSize	Length	0.324	metre
2	TorpedoSize	Width	2.6	metre
3	TorpedoWeight	TotalWeight	231	kilogram
4	TorpedoWeight	WarheadWeight	44.5	kilogram
5	TorpedoPower	PowerType	Thermodynamics	char
6	TorpedoPower	PowerEquipment	Piston cam machine	char
7	TorpedoVoyage	MaximumSpeed	45	knot
8	TorpedoVoyage	MinimumSpeed	28	knot
9	TorpedoVoyage	MaximumRange	20	kilometre
10	TorpedoVoyage	StandardSpeed	35	knot
11	TorpedoVoyage	TimeToReachStandardSpeed	8	second
12	TorpedoFuze	FuzeType	Trigger fuze and electromagnetic fuze	char
13	TorpedoExplosion	ExplosiveChargeType	Conventional explosives	char
14	TorpedoExplosion	ExplosiveAmount	43.9	kilogram
15	TorpedoGuidance	GuidanceType	Active-passive joint acoustic homing	char
16	TorpedoOperationDepth	MaximumDepth	305	metre
17	TorpedoRelease	TimeToEnterWater	10	second
18	TorpedoRelease	ReleaseHeight	3000	kilometre
19	TorpedoDetection	MaximumDetectRadius	1.5	kilometre
20	TorpedoDetection	ActiveDetectRange	2.5	kilometre
21	TorpedoDetection	PassiveDetectRange	6	kilometre
22	SonobuoyArray	SonobuoyArrayFormat	{Triangular, Circular}	char
23	SonobuoyArray	SonobuoyQuantity	6	int
24	TorpedoQuantity	TorpedoQuantity	{1, 2}	char

Figure 13. System measure matrix (SV-7).

There are a number of sonobuoy array formations, and the typical triangular array and circular array are selected as the optional formations. The system set  $T$  can be expressed as  $T = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$ . The elements in  $T$  represent the combat ship, sub-detection helicopter, sub-attack helicopter, fixed sonar array system, sonobuoy triangular array, sonobuoy circular array, MAD UAV, torpedo 1, and torpedo 2, respectively.

Based on the given information, the constraints on the optional systems are summarized as follows: (a) when a sonobuoy array or MAD UAV is selected, the sub-detection helicopter must be selected; (b) the fixed sonar array system must be selected; (c) only one of the sonobuoy array formations can be selected; and (d) at least one torpedo must be selected. Assume the set of selected systems is  $F$ , and the decision matrix  $X_{SV}$  is as follows.

$$X_{SV} = [1, x_2, 1, 1, x_5, x_6, x_7, x_8, x_9], x_i = \begin{cases} 1, & e_i \in F \\ 0, & e_i \notin F \end{cases} \tag{9}$$

$$x_2 = \begin{cases} 1, & x_5 + x_6 + x_7 > 0 \\ 0, & \text{otherwise} \end{cases}, x_5 + x_6 \leq 1, x_8 + x_9 \geq 1,$$

There are three options for the sonobuoy array (i.e., triangular, circular, none), two options for the MAD UAV (i.e., select or not), and two options for the torpedo (i.e., torpedo 2 select or not, since two torpedoes are the same). Thus, there are twelve selection options

in total and the selection strategy  $S_{SV}$  can be expressed as in (10). The twelve strategies can be defined using a matrix, as shown in (11).

$$S_{SV} = \{X_{SV}^1, X_{SV}^2, \dots, X_{SV}^i, \dots, X_{SV}^{12}\}, i = \{x \in R^+ | 1 \leq x \leq 12\}, \tag{10}$$

$$\begin{bmatrix} X_{SV}^1 \\ X_{SV}^2 \\ X_{SV}^3 \\ X_{SV}^4 \\ X_{SV}^5 \\ X_{SV}^6 \\ X_{SV}^7 \\ X_{SV}^8 \\ X_{SV}^9 \\ X_{SV}^{10} \\ X_{SV}^{11} \\ X_{SV}^{12} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{11}$$

### 4.3.2. Connecting Decisions in SV-2

The SV-2 model faces connecting decision problems, and it is necessary to determine whether and how to communicate between various systems. First, we need to determine the set of entities involved in the problem. It is known that the ASW SoS architecture involves six types of equipment systems, of which the number of anti-submarine helicopters and torpedoes is both 2. In the ASW SoS architecture, the tasks undertaken by the two anti-submarine helicopters are fixed but not the same, so they need to be distinguished. One is defined as a sub-attack helicopter (carrying submarine attack equipment) and the other is defined as a sub-detection helicopter (carrying submarine detection equipment). The two torpedoes have the same mission and are both carried by the sub-attack helicopter. Therefore, there is no need to distinguish them. Thus, the system set  $E$  can be expressed as  $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ . The elements in  $E$  represent the combat ship, sub-detection helicopter, sub-attack helicopter, fixed sonar array system, sonobuoy array, MAD UAV, and torpedo, respectively.

The constraints on the systems are listed as follows: (a) duplex communication is used for combat ship and helicopter communication; (b) the fixed sonar array system can only communicate with the combat ship and duplex communication is used; (c) the sonobuoy array sends information to the sub-detection helicopter only after entering water; (d) the MAD UAV can only communicate with the sub-detection helicopter and duplex communication is used; (e) the torpedoes can receive information from the helicopter before entering water, and cannot communicate with any systems after entering water.

We can write the decision matrix  $X_{SV-2}$  for the SV-2 model accordingly, as shown in (12).  $x_{ij}$  indicates the communication status between two systems.

$$X_{SV-2} = \begin{bmatrix} 0 & 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & x_{23} & 0 & -1 & 2 & 0 \\ 2 & x_{32} & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{12}$$

$$x_{ij} = \begin{cases} 0, & e_i \text{ and } e_j \text{ have no communication} \\ 1, & e_i \text{ sends message to } e_j \\ -1, & e_j \text{ sends message to } e_i \\ 2, & e_i \text{ and } e_j \text{ have duplex communication} \end{cases},$$

$$x_{23}, x_{32} \rightarrow X_{SV}, x_{23} \pm x_{32} = 0,$$

The meaning of  $x_{23}, x_{32} \rightarrow X_{SV}$  is that the values of  $x_{23}$  and  $x_{32}$  depend on the decision matrix  $X_{SV}$  described in (11). Take  $x_{23}$  as an example. If  $X_{SV}$  shows that the sub-detection helicopter is not selected (e.g.,  $X_{SV}^{11}$  and  $X_{SV}^{12}$ ), there is no communication between the sub-detection helicopter and the sub-attack helicopter; that is,  $x_{23} = 0$ .

There are four types of interactions: no communication ( $x_{ij} = 0$ ), duplex communication ( $x_{ij} = 2$ ), and simplex communication ( $x_{ij} = \pm 1$ ) in which the positive and negative sign indicates the direction. Since the values of  $x_{ij}$  and  $x_{ji}$  in  $X_{SV-2}$  are either the same or in opposite signs, only one undetermined variable remains and it concerns the communication mode between the sub-detection helicopter and the sub-attack helicopter.

Two types of communication modes are under consideration—star and mesh. If the star topology is selected, the sub-detection helicopter and the sub-attack helicopter cannot directly communicate with each other, as the decision matrix  $X_{SV-2,1}$  for SV-2 models shows in (13). If the mesh topology is selected, the sub-detection helicopter and the sub-attack helicopter use duplex communication, as the decision matrix  $X_{SV-2,2}$  for SV-2 model shows in (14).

$$X_{SV-2,1} = \begin{bmatrix} 0 & 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & -1 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{13}$$

$$x_{ij} = \begin{cases} 0, & e_i \text{ and } e_j \text{ have no communication} \\ 1, & e_i \text{ sends message to } e_j \\ -1, & e_j \text{ sends message to } e_i \\ 2, & e_i \text{ and } e_j \text{ have duplex communication} \end{cases},$$

$$X_{SV-2,2} = \begin{bmatrix} 0 & 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & -1 & 2 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{14}$$

$$x_{ij} = \begin{cases} 0, & e_i \text{ and } e_j \text{ have no communication} \\ 1, & e_i \text{ sends message to } e_j \\ -1, & e_j \text{ sends message to } e_i \\ 2, & e_i \text{ and } e_j \text{ have duplex communication} \end{cases},$$

The decision matrixes  $X_{SV-2,1}$  and  $X_{SV-2,2}$  represent different SV-2 models. The SV-2 model for  $X_{SV-2,2}$  is illustrated in Figure 14.

Overall, the decision matrix facilitating the SV-2 model’s development has two options,  $X_{SV-2}^1(x_{23} = 0)$  and  $X_{SV-2}^2(x_{23} = 2)$ . Thus, the solutions generated for the SV-2 model are listed in (15). Note that if the sub-detection helicopter is not selected,  $x_{23}$  is set to zero and only strategy  $X_{SV-2}^1$  can be selected.

$$S_{SV-2} = \{X_{SV-2}^1, X_{SV-2}^2\}, \tag{15}$$

#### 4.3.3. Partitioning Decisions in OV-5a and SV-5

SV-5 model development faces assigning decision problems and OV-5a model development faces partitioning decision problems. Since SV-5 and OV-5a are closely related, we can combine these two decision-making problems into one decision-making problem. That is, the partitioning systems can be divided into mutually exclusive and collectively exhaustive functions and those functions can support the operational activities. Since the operational activities are conducted by operational performers, the operational performers



are set as references for the initial function decomposition. The set of functions  $P$  is expressed as  $P = \{P_1, P_2, P_3, P_4, P_5\}$ . The elements in  $P$  represent the C2Node, SubDetectNode, SubAttackNode, CommNode, and the PayloadDeployNode, respectively.

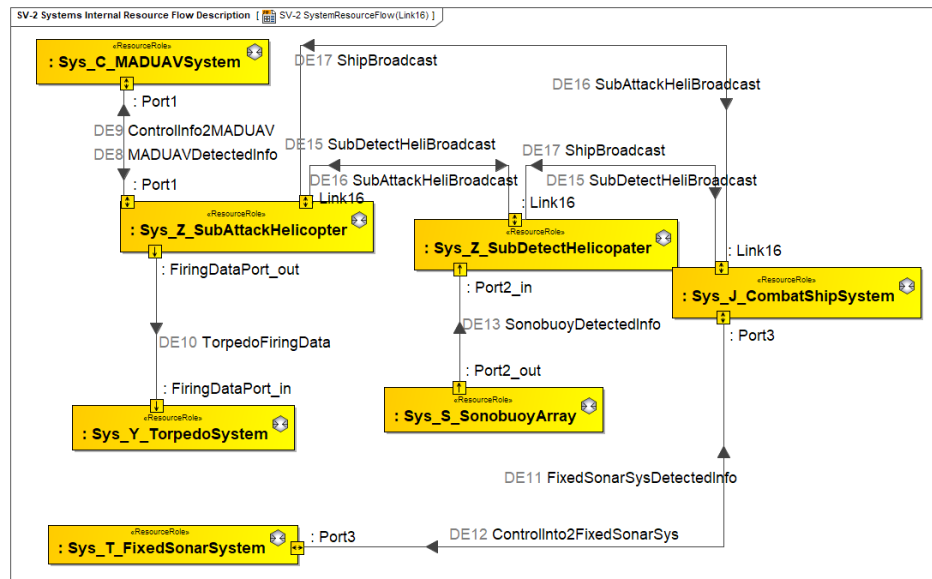


Figure 14. SV-2 model using mesh topology.

Take the function decomposition of the sonobuoy system as an example. The function set of sonobuoy  $F_{SN}$  is expressed as  $F_{SN} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}\}$ . The elements in  $F_{SN}$  represent ProvidePowerforDetection, ProvidePowerforCommunication, DetermineDepth, RetardationAfterWaterEntry, SendRadioSignal, SignalModulation, RetardationDuringFall, ShockAbsorbDuringWaterEntry, ProvideBuoyancy, Transduce, SendAcousticSignal, and ReceiveAcousticSignal, respectively, as shown in Figure 15.

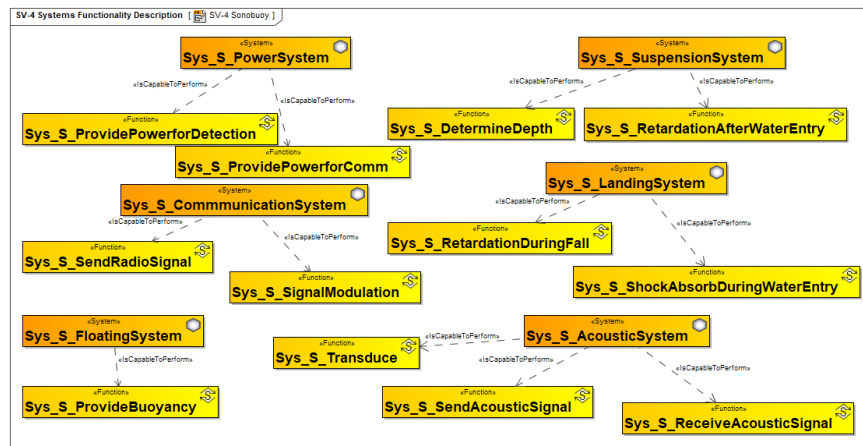


Figure 15. SV-4a model (sonobuoy system).

$x_{ij}$  indicates whether function  $f_j$  of the sonobuoy is added to function set  $P_i$ . The decision matrix  $X_{OV-5\_SN}$  concerning the partitioning decisions of the sonobuoy functions is written as in (16).

$$X_{OV-5\_SN} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, x_{ij} = \begin{cases} 1, & f_j \in P_i \\ 0, & f_j \notin P_i \end{cases} \quad (16)$$



Legend	
	Maps To Capability (Implied)
	MapsToCapability
OpAct_AttackHelicopterComm	
OpAct_AttackHelicopterPayloadRel	
OpAct_DetectHelicopterComm	
OpAct_DetectHelicopterPayloadRel	
OpAct_FixedSonarComm	
OpAct_MADUAVComm	
OpAct_MADUAVDetect	
OpAct_MADUAVFlyingToStandbyPoint	
OpAct_ShipComm	
OpAct_SonarComm	
OpAct_SonarDetect	
OpAct_SonobuoyIntoWater	
OpAct_SubAttackResDeployment	
OpAct_SubDetectProcess@Distribut	
OpAct_SubDetectResDeployment	
OpAct_TaskPlan@Distribution	
OpAct_TorpedoSearch&Attack	
Cap_AntiSubCap	

Figure 18. CV-6 model.

Based on the qualitative analysis, both OV-5a and SV-5 have one option and do not need further decision-making.

#### 4.3.4. Permuting Decisions in OV-5b

Since we already determined the connecting decisions in the SV-2 model, we are only concerned with the permuting decisions in OV-5b. According to the “sense-decide-influence” kill chain process, we set the position index set  $Q = \{Q_1, Q_2, Q_3\}$ . The elements in  $Q$  represent EarlyDetect-C2 position index set  $Q_1$ , SubDetect-C2 position index set  $Q_2$ , and SubAttack position index set  $Q_3$ . The sequence of positions represents the temporal relationship of operational activities. We decompose the subset of operational activities based on the temporal sequence, as shown in (17)–(19).

$$OP_1 = \{\text{EarlyDetect, FixedSonarComm, ShipComm, TaskPlan\&Distribution}\}, \quad (17)$$

$$OP_2 = \left\{ \begin{array}{l} \text{DetectHelicopterPayloadRelease, SonobuoyIntoWater, ...} \\ \text{MADUAVFlyingToStandbyPoint, ...} \\ \text{SonarDetect, MADUAVDetect, ShipComm, SonarComm, ...} \\ \text{MADUAVComm, DetectHelicopterComm} \end{array} \right\}, \quad (18)$$

$$OP_3 = \left\{ \begin{array}{l} \text{AttackHelicopterPayloadRelease, TorpedoSearchAttack, ...} \\ \text{AttackHelicopterComm} \end{array} \right\}, \quad (19)$$

The decomposition is different from the partitioning decisions since the latter requires a collection of mutually exclusive and collectively exhaustive subsets, while the former only requires collectively exhaustive subsets, as shown in (20).

$$OP = \{op | op \text{ denotes any operational activity in OV - 5a model}\}, \quad (20)$$

$$OP = \bigcup_{i=1}^3 OP_i, \bigcap_{i=1}^3 OP_i \neq \emptyset,$$

Based on the operational activity subset  $OP_1$ , we can determine the elements of the EarlyDetect-C2 position index set, as written in (21). Based on the operational rules of the operational activities in  $OP_1$ , the decision matrix  $X_{OV-5b_1}$  is written as in (22), then the sequence of operational activities is determined.

$$Q_1 = \{q_1, q_2, q_3, q_4\}, \quad (21)$$

$$X_{OV-5b_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, x_{ik} = \begin{cases} 1, & op_i \text{ is allocated to } q_k \\ 0, & op_i \text{ is not allocated to } q_k \end{cases} \quad (22)$$

Likewise, we obtain the decision matrix  $X_{OV-5b_3}$ , as shown in (23). Because the location of operational activity “AttackHelicopterComm” remains undetermined, we use  $x_{31}$  to represent it.

$$X_{OV-5b_3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ x_{31} & 0 & 0 \end{bmatrix}, x_{ik} = \begin{cases} 1, & op_i \text{ is allocated to } q_k \\ 0, & op_i \text{ is not allocated to } q_k \end{cases}, x_{31} \rightarrow X_{SV}, \quad (23)$$

The operational activity subset  $OP_2$  concerns SonarDetection and MADUAVDetection. We decompose  $OP_2$  into two subsets,  $OP_{2-1}$  and  $OP_{2-2}$ , representing subsets of sonar operational activities and MAD UAV operational activities, respectively, as shown in (24) and (25). The associated decision matrices,  $X_{OV-5b_2-1}$  and  $X_{OV-5b_2-2}$ , are listed in (26) and (27). We use “a” to represent the values of  $x_{11}$ ,  $x_{22}$ ,  $x_{33}$ , and  $x_{54}$  in  $X_{OV-5b_2-1}$ . The values are the same but unknown, and all are constrained by  $X_{SV}$ . Similarly, we use “a” to represent the same values of  $x_{11}$ ,  $x_{24}$ ,  $x_{35}$ ,  $x_{56}$ , and  $x_{67}$  and use “b” to represent the same values of  $x_{62}$  and  $x_{53}$  in  $X_{OV-5b_2-2}$ .

$$OP_{2-1} = \left\{ \begin{array}{l} \text{DetectHelicopterPayloadRelease,SonobuoyIntoWater,...} \\ \text{SonarDetect,ShipComm,SonarComm} \end{array} \right\}, \quad (24)$$

$$OP_{2-2} = \left\{ \begin{array}{l} \text{DetectHelicopterPayloadRelease,...} \\ \text{MADUAVFlyingToStandbyPoint,MADUAVDetect,...} \\ \text{ShipComm,MADUAVComm,DetectHelicopterComm} \end{array} \right\}, \quad (25)$$

$$X_{OV-5b_2-1} = \begin{bmatrix} a & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & 0 & x_{45} \\ 0 & 0 & 0 & a & 0 \end{bmatrix}, x_{ik} = \begin{cases} 1, & op_i \text{ is allocated to } q_k \\ 0, & op_i \text{ is not allocated to } q_k \end{cases}, \quad (26)$$

$a \rightarrow X_{SV}, x_{45} \rightarrow X_{SV-2},$

$$X_{OV-5b_2-2} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{48} \\ 0 & 0 & b & 0 & 0 & a & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 & a & 0 \end{bmatrix}, \quad (27)$$

$$x_{ik} = \begin{cases} 1, & op_i \text{ is allocated to } q_k \\ 0, & op_i \text{ is not allocated to } q_k \end{cases}, \quad a \rightarrow X_{SV}, \quad x_{48}, b \rightarrow X_{SV-2},$$

Take  $X_{OV-5b_2-1}$  as an example to analyze the interactions between decision matrices. If  $X_{SV}$  is set as (28), and  $X_{SV-2}$  is set as (14), then  $X_{OV-5b_2-1}$  can be written as (29).

$$X_{SV} = [1, 1, 1, 1, 1, 1, 1], \quad (28)$$

$$X_{OV-5b_2-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, x_{ik} = \begin{cases} 1, & op_i \text{ is allocated to } q_k \\ 0, & op_i \text{ is not allocated to } q_k \end{cases}, \quad (29)$$

Assume that  $O$  is the set of decision matrices as shown in (30). The decision matrix for the OV-5b model  $X_{OV-5b}$  is shown in (31).

$$O = \{X_{OV-5b_1}, X_{OV-5b_2-1}, X_{OV-5b_2-2}, X_{OV-5b_3}\}, \quad (30)$$

$$X_{OV-5b} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, x_{ik} = \begin{cases} 1, & op_i \text{ is allocated to } Q_k \\ 0, & op_i \text{ is not allocated to } Q_k \end{cases}, \quad (31)$$

Overall, the decision matrices for developing OV-5b models include  $X_{OV-5b_1}$ ,  $X_{OV-5b_2-1}$ ,  $X_{OV-5b_2-2}$ ,  $X_{OV-5b_3}$ , and  $X_{OV-5b}$ . There are no decision variables in  $X_{OV-5b_1}$  and  $X_{OV-5b}$ . Decision matrices  $X_{OV-5b_2-1}$  and  $X_{OV-5b_3}$  are constrained by  $X_{SV}$  and  $X_{SV-2}$ , but do not include decision variables. Only  $X_{OV-5b_2-2}$  contains decision variables. As shown in (27), the value of  $x_{62}$  depends on whether the MAD UAV can receive guidance information from the sub-detection helicopter after being launched. Then, the decision matrix  $X_{OV-5b_2-2}$  could have two options,  $X_{OV-5b}^1(x_{62} = 0)$  and  $X_{OV-5b}^2(x_{62} = 1)$ . Thus, the strategy set  $S_{OV-5b}$  of the OV-5b model can be expressed as in (32). When the MAD UAV is not selected for submarine detection, the value of  $x_{62}$  is 0 and only  $X_{OV-5b}^1$  can be used.

$$S_{OV-5b} = \{X_{OV-5b}^1, X_{OV-5b}^2\}, \quad (32)$$

### 4.3.5. Combination Pattern for Generating Alternative Set

Based on the above analysis, the decision problems in  $S_{SV}$ ,  $S_{SV-2}$ , and  $S_{OV-5b}$  receive 12 strategies, 2 strategies, and 2 strategies, respectively. According to the combination pattern, the ASW SoS architecture has 34 alternatives, as shown in Figure 19. Because the strategies from the three decision points are not independent from each other, the total number of alternatives is 34 instead of 48. Based on the 34 architecture alternatives, we can complete the DoDAF models (Figure 20 as an example) and continue building executable simulation models to examine the effectiveness of alternatives quantitatively.

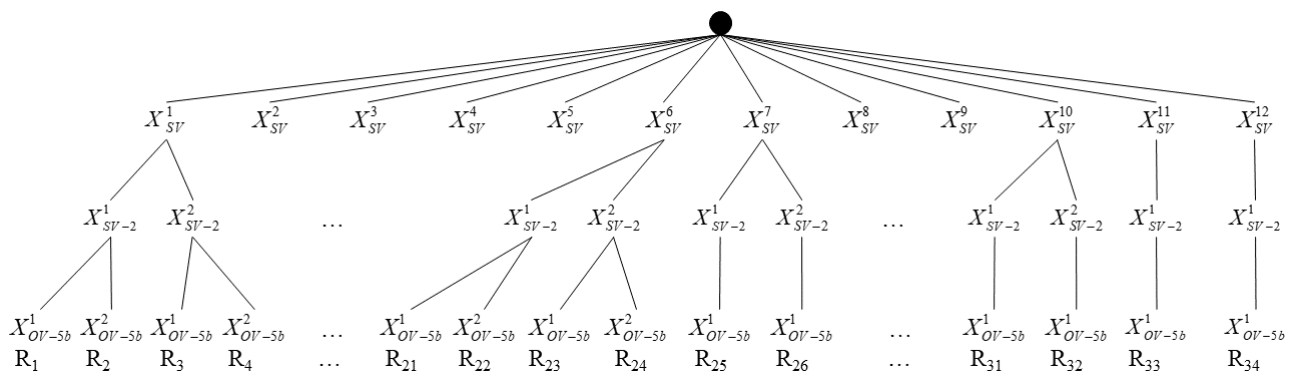


Figure 19. ASW SoS architecture design space represented by decision tree.

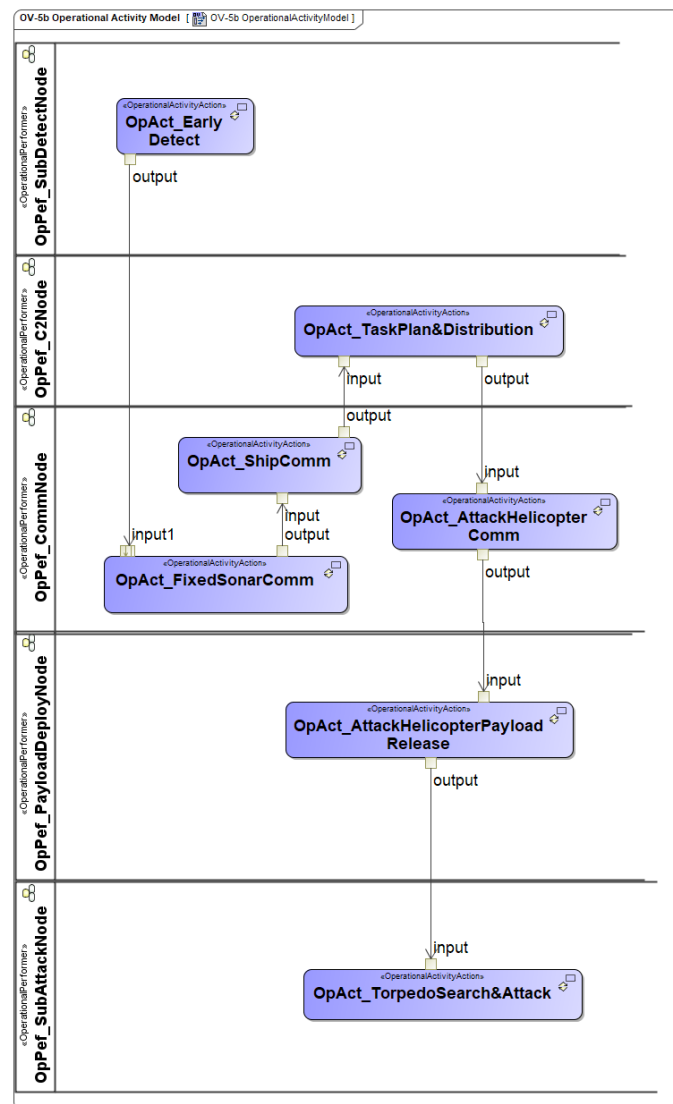


Figure 20. OV-5b model for alternative R34.

### 5. Conclusions

SoS architecture design is a process integrating architecture modeling and architecture decision-making. A classic architecture design process includes developing architecture description models, building corresponding executable simulation models, and employing the design of experiments to obtain parameters and run simulations, and eventually conducting SoS effectiveness evaluation. However, even though many elements are unknown during architecture model development, it is common that the architects (or modelers, systems engineers, etc.) continue the modeling anyway to only complete the tasks instead of using an appropriate decision-support tool to figure out the right model.

To facilitate adequate thinking during SoS architecture design, this paper developed a decision pattern-guided DoDAF modeling approach. Specifically, we identified the major decision points and decision problems during the architecture model developing using the DoDAF and developed decision patterns including downselecting, assigning, connecting, partitioning, permuting, and the combination of the previous patterns to provide qualitative decision analysis that can be integrated with DoDAF model development. After implementing the entire process, we are able to generate a set of architecture alternatives that can further be quantitatively analyzed using simulation. The process and the results are illustrated in the ASW SoS architecture design example. Compared to the traditional

design of experiments method, our method is highly integrated with the DoDAF modeling process itself.

Our proposed method aims to motivate and enforce a thorough, iterative, and interactive analysis during SoS architecture model development in practice. The method is not only useful in the SoS domain, but is also useful for the system architecture design in which the decision patterns of downselecting, connecting, assigning, etc., can be applied as well. In the short-run future, we consider developing a plug-in software to enable decision pattern-based analysis for supporting architecture model development and alternative generation, and, meanwhile, developing mathematical formulation and quantitative solutions for the decision patterns using optimization theory and decision theory if necessary. In the long run, an automated analysis using advanced artificial intelligence (AI) techniques such as large language models (LLM) and other generative AI models should have large potential to further facilitate the architecture design process. Some initial studies [42,43] have worked on translating natural language-based design documents to system modeling language (SysML) or unified modeling language (UML) models by using templates. A few other studies [44] aim to construct representation models via data, mathematical equations, and quantitative analysis results. Both aspects are useful for architecture design.

**Author Contributions:** Conceptualization, Z.F., X.Z. and F.L.; methodology, Z.F.; modeling, X.Z.; validation, X.Z., F.L. and Z.F.; writing—original draft preparation, Z.F.; writing—review and editing, F.L.; visualization, X.Z.; supervision, Z.F.; project administration, Z.F.; funding acquisition, Z.F. and F.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant No. 62103158.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Duren, B.G.; Pollard, J.R. *Combat Systems Vision 2030 Combat System Architecture: Design Principles and Methodology*; Naval Surface Warfare Center Combat Systems Department: Dahlgren, VA, USA, 1991.
2. *ISO/IEC/IEEE 42010:2011(E)*; Systems and Software Engineering—Architecture Description. ISO/IEC/IEEE: New York, NY, USA, 2011.
3. Fang, Z. System-of-Systems Architecture Selection: A Survey of Issues, Methods, and Opportunities. *IEEE Syst. J.* **2022**, *16*, 4768–4779. [[CrossRef](#)]
4. Levis, A.H.; Wagenhals, L.W. C4ISR Architectures: I. Developing a Process for C4ISR Architecture Design. *Syst. Eng.* **2000**, *3*, 225–247. [[CrossRef](#)]
5. Parnell, G.S.; Shallcross, N.J.; Specking, E.A.; Pohl, E.A.; Phillips, M. Role of Decision Analysis in MBSE. In *Handbook of Model-Based Systems Engineering*; Madni, A.M., Augustine, N., Sievers, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; pp. 119–150.
6. Madni, A.M.; Boehm, B.; Erwin, D.; Moghaddam, M.; Sievers, M.; Wheaton, M. Implementing a MOSA Decision Support Tool in a Model-based Environment. In *Recent Trends and Advances in Model-Based Systems Engineering*; Dai, M., Guariniello, C., DeLaurentis, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; pp. 257–268.
7. Giachetti, R.E. Evaluation of the DoDAF Meta-Model's Support of Systems Engineering. *Procedia Comput. Sci.* **2015**, *61*, 254–260. [[CrossRef](#)]
8. Hughes, T.; Tolk, A. Orchestrating Systems Engineering Processes and System Architectures within DoD: A Discussion of the Potentials of DoDAF. *J. RMS Syst. Eng.* **2013**, *13*, 13–19.
9. Wagenhals, L.W.; Shin, I.; Kim, D.; Levis, A.H. C4ISR Architectures: II. A Structured Analysis Approach for Architecture Design. *Syst. Eng.* **2000**, *3*, 248–287. [[CrossRef](#)]
10. Wagenhals, L.W.; Levis, A.H. Service Oriented Architectures, the DoD Architecture Framework 1.5, and Executable Architectures. *Syst. Eng.* **2008**, *12*, 312–343. [[CrossRef](#)]
11. Piaszczyk, C. Model Based Systems Engineering with Department of Defense Architectural Framework. *Syst. Eng.* **2011**, *14*, 305–326. [[CrossRef](#)]
12. Williams, J.L.; Stracener, J.T. First Steps in the Development of a Program Organizational Architectural Framework (POAF). *Syst. Eng.* **2012**, *16*, 45–70. [[CrossRef](#)]
13. Alblawi, A.; Stracener, J.; Williams, J. Decision-making in the Program Organization Architecture Framework-Owner View (POAF-OV). In Proceedings of the 2016 Annual IEEE Systems Conference (SysCon), Orlando, FL, USA, 18–21 April 2016.

14. Handley, H.A.H. A Design Methodology for Fit-for-Purpose Human Views. *Syst. Eng.* **2016**, *19*, 498–509. [[CrossRef](#)]
15. Amissah, M.; Handley, H.A.H. A Process for DoDAF based Systems Architecting. In Proceedings of the 2016 Annual IEEE Systems Conference (SysCon), Orlando, FL, USA, 18–21 April 2016.
16. Shaked, A.; Reich, Y. Designing Development Processes Related to System of Systems Using a Modeling Framework. *Syst. Eng.* **2019**, *22*, 561–575. [[CrossRef](#)]
17. Abhaya, L. UAF (Unified Architecture Framework) Based MBSE (UBM) Method to build a System of Systems Model. In Proceedings of the 31st Annual INCOSE International Symposium, Virtual Event, 17–22 July 2021.
18. Liu, N.; Wang, J.; Zhang, Y.; Li, D.; Ju, M. Top-Down Military System-of-Systems Design using MBSE based on UAF: A Case Study. In Proceedings of the International Conference on Complex Systems Design & Management, Beijing, China, 30–31 October 2023.
19. Bankauskaite, J.; Morkevicius, A. Towards an Automated UAF-based Trade Study Process for System of Systems Architecture. In Proceedings of the 30th Annual INCOSE International Symposium, Virtual Event, 20–22 July 2020.
20. Martin, J.N.; O’Neil, D.P. Enterprise Architecture Process Guide for the Unified Architecture Framework (UAF). In Proceedings of the 31st Annual INCOSE International Symposium, Virtual Event, 17–22 July 2021.
21. Object Management Group. *Enterprise Architecture Guide for UAF: Appendix C*; Object Management Group: Needham, MA, USA, 2021; Volume 31, pp. 242–263.
22. Sohn, M.; Jeong, S.; Kim, T.; Lee, H.J. Development supporting framework of architectural descriptions using heavy-weight ontologies with fuzzy-semantic similarity. *Soft Comput.* **2016**, *21*, 6105–6119. [[CrossRef](#)]
23. Zhang, X.; Luo, A.; Mao, Y.; Lin, M.; Kou, Y.; Liu, J. A Semi-Automatic Optimization Design Method for SvcV-5 in DoDAF 2.0 based on Service Identification. *IEEE Access* **2020**, *8*, 33442–33460. [[CrossRef](#)]
24. Fang, Z.; Jin, W. Exploring Decision Patterns for Supporting DoDAF based Architecture Design. In Proceedings of the 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Prague, Czech Republic, 9–12 October 2022.
25. Selva, D.; Cameron, B.; Crawley, E. Patterns in System Architecture Decisions. *Syst. Eng.* **2016**, *19*, 477–497. [[CrossRef](#)]
26. Hegedus, A.; Horvath, A.; Varro, D. A Model-driven Framework for Guided Design Space Exploration. *Autom. Softw. Eng.* **2015**, *22*, 399–436. [[CrossRef](#)]
27. Kang, E.; Jackson, E.; Schulte, W. An Approach for Effective Design Space Exploration. In Proceedings of the Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems, Redmond, WA, USA, 31 March–2 April 2010; pp. 33–54.
28. Griendling, K.; Mavris, D. An Architecture-based Approach to Identifying System-of-Systems Alternatives. In Proceedings of the 5th International Conference on System of Systems Engineering, Loughborough, UK, 22–24 June 2010.
29. Franzen, L.K.; Staack, I.; Krus, P.; Jouannet, C.; Amadori, K. A Breakdown of System of Systems Needs using Architecture Frameworks, Ontologies and Description Logic Reasoning. *Aerospace* **2021**, *8*, 118. [[CrossRef](#)]
30. Franzen, L.K.; Staack, I.; Jouannet, C.; Krus, P. An Ontological Approach to System-of-Systems Engineering in Product Development. In Proceedings of the Aerospace Technology Congress, Stockholm, Sweden, 8–9 October 2019.
31. Guariniello, C.; Fang, Z.; Davendralingam, N.; Marais, K.; DeLaurentis, D. Tool Suite to Support Model Based Systems Engineering Enabled System-of-Systems Analysis. In Proceedings of the 2018 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2018.
32. Raz, A.K.; Kenley, C.R.; DeLaurentis, D.A. System architecting and design space characterization. *Syst. Eng.* **2018**, *21*, 227–242. [[CrossRef](#)]
33. Huff, J.; Medal, H.; Griendling, K. A Model-based Systems Engineering Approach to Critical Infrastructure Vulnerability Assessment and Decision Analysis. *Syst. Eng.* **2019**, *22*, 114–133. [[CrossRef](#)]
34. Specking, E.; Parnell, G.; Pohl, E.; Buchanan, R. Early Design Space Exploration with Model-Based System Engineering and Set-Based Design. *Systems* **2018**, *6*, 45. [[CrossRef](#)]
35. Parnell, G.S.; Shallcross, N.; Specking, E.; Philips, M. MBSE Enabled Trade-off Analyses. In Proceedings of the 31st Annual INCOSE International Symposium, Virtual Event, 17–22 July 2021.
36. Timperley, L.; Berthoud, L.; Snider, C.; Tryfonas, T.; Prezzavento, A.; Palmer, K. Towards Improving the Design Space Exploration Process Using Generative Design with MBSE. In Proceedings of the 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2023.
37. Li, H.; Lachmayer, R.H. Automated Exploration of Design Solution Space Applying the Generative Design Approach. In Proceedings of the Design Society: 22nd International Conference on Engineering Design (ICED19), Delft, The Netherlands, 5–8 August 2019.
38. Madni, A.M.; Augustine, N.; Sievers, M. MBSE in Architecture Design Space Exploration. In *Handbook of Model-Based Systems Engineering*; Bussemaker, J.H., Ciampa, P.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; pp. 589–629.
39. She, S.; Lu, J.; Wang, G.; Ding, J.; Hu, Z. Model-Based Systems Engineering Supporting Integrated Modeling and Optimization of Radar Cabin Layout. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Nantes, France, 5–9 September 2021.
40. Paape, N.; van Eekelen, J.; Reniers, M. A specification language for automated design space exploration of production systems. *Procedia CIRP* **2023**, *119*, 1023–1028. [[CrossRef](#)]
41. Schoeberl, M.; Khodadad, E.; Lin, S.; Maroun, E.J.; Pezzarossa, L.; Lee, E.A. Invited Paper: Worst-Case Execution Time Analysis of Lingua Franca Applications. In Proceedings of the 22nd International Workshop on Worst-Case Execution Time Analysis (WCET 2024), Lille, France, 9–12 July 2024.



42. Camara, J.; Troya, J.; Burgueno, L.; Vallecillo, A. On the Assessment of Generative AI in Modeling Tasks: An Experience Report with ChatGPT and UML. *Softw. Syst. Model.* **2023**, *22*, 781–793. [[CrossRef](#)]
43. Timperley, L.R.; Berthoud, L.; Snider, C.; Tryfonas, T. Assessment of Large Language Models for Use in Generative Design of Model Based Spacecraft System Architectures. 2024. Available online: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4823264](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4823264) (accessed on 25 August 2024).
44. Abdelbari, H.; Shafi, K. A System Dynamics Modeling Support System Based on Computational Intelligence. *Systems* **2019**, *7*, 47. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.