Employing UAF Inter-Domain Traceability for Performance and Effectiveness Evaluation

Ahmad Alghamdi
dept. Systems Engineering and Operations Research
George Mason University
Fairfax, United States
Email: aalgham8@gmu.edu
ORCID: 0000-0003-3542-3882

Mohammadreza Torkjazi
dept. Systems Engineering and Operations Research
George Mason University
Fairfax, United States
Email: mtorkjaz@gmu.edu
ORCID: 0000-0001-8866-3009

Arturo J. Davila-Andino
dept. Systems Engineering and Operations Research
George Mason University
Fairfax, United States
Email: adavilaa@gmu.edu
ORCID: 0000-0003-4115-2003

Abbas K. Zaidi
dept. Systems Engineering and Operations Research
George Mason University
Fairfax, United States
Email: szaidi2@gmu.edu

***Abstract*—We propose a step-by-step Model-Based Systems Engineering (MBSE) process for the creation and simulation of an executable Unified Architecture Framework (UAF) model for evaluation purposes. The roll-up of Technical Performance Measures (TPMs) to Measures of Effectiveness (MOEs) is necessary for such a process, and has not been documented for the UAF. This paper is the first attempt to address this gap by demonstrating how interdependencies between these technical measures can be traced across the domains of a UAF architecture according to the ISO/IEC/IEEE 15288:2015 standard, the guidelines from the INCOSE Systems Engineering Handbook, and the UAF Enterprise Architecture Guide. The proposed process employs traceability and parametric diagrams within the UAF to produce an executable model that aids in evaluating the effectiveness of a system's architecture. Additionally, we describe how to build a simulation within the UAF to assess a parametric diagram containing random values of TPMs. The process identifies UAF views, their constituent model elements, and the relationships that are required to build this model. We also present an illustrative example of a forest firefighting system to demonstrate the implementation and effectiveness of the proposed process. This paper is intended as a resource for systems engineering practitioners.**

***Index Terms*—Unified Architecture Framework (UAF), Architecture Evaluation, Technical Measures, Trade Study, Effectiveness Evaluation, System Architecture**

I. INTRODUCTION

Systems architectures are created using a variety of views and viewpoints provided by architecture frameworks such as the Department of Defense Architecture Framework (DoDAF) and Unified Architecture Framework (UAF). The UAF assists system architects in better describing the complexities existing within the architectures of enterprises or Systems of Systems (SoSs) [1], [2]. An SoS offers a range of capabilities to fulfill purposes that cannot be addressed by individual constituent systems [3]. Measures of Effectiveness (MOEs) is a typical measurement for evaluating the effectiveness of an SoS architecture in delivering such capabilities [2], [4].

Each viewpoint in an architecture framework corresponds to a specific domain, e.g., operational and resource domains. The UAF facilitates trade-offs between different viewpoints by providing traceability [5] and parametric analyses. By incorporating both structural and behavioral views, parametric diagrams can help with the evaluation and comparison of different alternative architectures for a system during a trade study [6]. Parametric diagrams have been used previously in ISO/IEC/IEEE 15288:2015 compliant trade studies for architectures described in Systems Modeling Language (SysML) [7], [8]. However, the process for connecting technical measures such as MOEs, Measures of Performance (MOPs), and Technical Performance Measures (TPMs) in the UAF has not been documented. Since each category of these technical measures comes from a distinct UAF viewpoint, as shown in Fig. 1, understanding the traceability between various viewpoints is necessary before making the connection between them. The traceability also aids systems architects in creating one integrated architecture model that enables engineering analysis, trade studies, and simulations [9].

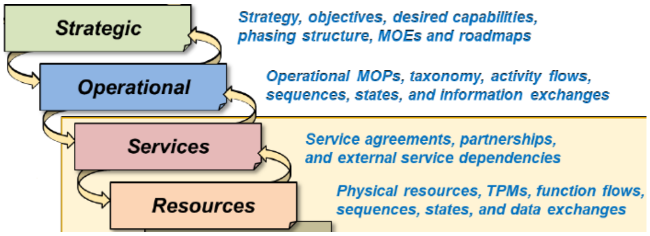


Fig. 1: Interdependency between UAF domains [10]

A trade study can be defined as a decision-making process during system design and development [11]. Such decision-making is challenging since there are many objectives, requirements, and uncertainty. A trade study is then a process of comparing these alternatives and choosing the one that will best serve the interests of all stakeholders involved [5], [11]. Trade studies are mostly employed in the system development stage of the systems' life cycle [12]. Therefore, there is a need to demonstrate how the connection between technical measures can be implemented within the UAF executable model that creates the foundation for conducting a trade study analysis.

In this paper, we are investigating the following: 1) how traceability between technical measures can be identified using UAF viewpoints, and 2) how this can provide us with an executable UAF model that allows the calculation of technical measures for architecture evaluation. To address these concerns, we demonstrate an MBSE process that employs UAF traceability and parametric diagrams according to the ISO/IEC/IEEE 15288:2015 standard and the guidelines from the INCOSE Systems Engineering Handbook [4], [13]. We also provide an illustrative case study of the proposed approach. This will help practitioners to use the UAF to show the connection of technical measures and create a UAF executable model.

The remainder of this paper is organized as follows: Section 2 examines how technical measures from different UAF viewpoints are interconnected and discuss the need for an executable model and simulation in architecture evaluation. Section 3 describes our proposed inter-domain traceability process for UAF architecture evaluation. Section 4 demonstrates the step-by-step implementation of the proposed process in a hypothetical case study. Lastly, Section 5 concludes our work and highlights future research directions.

II. BACKGROUND

A. Technical Measures Derivation

Engineering concerns, e.g., cost, performance, and reliability, can be expressed by technical measures, i.e., MOEs, MOPs, and TPMs. Technical measures are used for verification and validation purposes in systems engineering practices. According to the INCOSE handbook [4], MOEs, MOPs, and TPMs are used in the system analysis process to provide data to the decision management process for selecting the most efficient design alternative for a system.

Fig. 2 illustrates how these three technical measures are interdependent. TPMs and MOPs are a system's verification criteria that should be traceable to MOEs which are the validation criteria for a system. According to the INCOSE handbook [4], MOEs are from the system user's viewpoint and indicate how well a system achieves the mission objectives. They are the metrics to measure overall operational success criteria (e.g., performance, safety, etc.) for a designed system. MOPs should be derived from MOEs and ensure that the designed system fulfills the requirements that are necessary to satisfy them. TPMs need to address the MOPs and determine how well a system's elements work by focusing on the technical parameters of the elements. Eq. 1 and 2 show these interdependencies.

$$MOEs = f(MOPs, Requirements) \tag{1}$$

$$MOPs = f(TPMs) \tag{2}$$

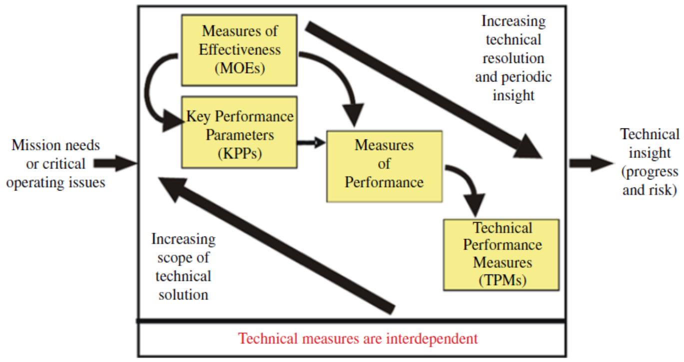


Fig. 2: Interdependency of technical measures [4]

MOEs, MOPs, and TPMs are not limited to one-to-one relationships in which an MOP is a function of one TPM, and similarly, an MOE is a function of one MOP. In fact, their relationship can be of type many-to-many, as shown in Fig. 3. To understand these relationships, assume that an enterprise has an MBSE team of 3 experts. In the first scenario, the TPM of *"Time spent on projects"* for an MBSE expert affects two MOPs of the team, e.g., *"Total time spent on projects"* and *"Productivity"*. Similarly, the MOP of *"Total time spent on projects"* affects two MOEs, e.g., the enterprise's *"Revenue"* and *"Reputation"*. Thus, one TPM influences two MOPs, and one MOP affects two MOEs. In the second scenario, assume that three TPMs of *"Experience level"*, each corresponding to an MBSE expert, impacts the MOP of *"Design quality"*. Similarly, two MOPs of *"Design quality"* and *"Design details"* impact the MOE of *"Customer satisfaction"*. Hence, three TPMs influenced only one MOP and two MOPs influenced only one MOE. This simple example also shows how TPMs can be used as inputs to calculate MOPs, which can then be used to compute MOEs.

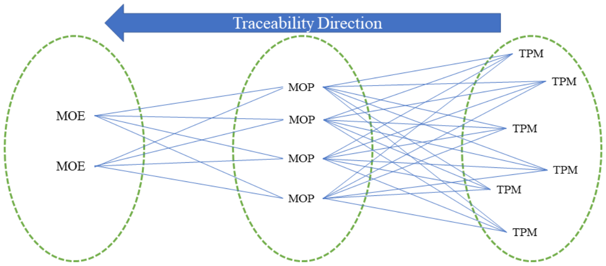


Fig. 3: Interconnection between MOEs, MOPs, and TPMs

According to the UAF Enterprise Architecture Guide (EAG) [10], [14], Fig. 1, MOEs, MOPs, and TPMs are defined in the

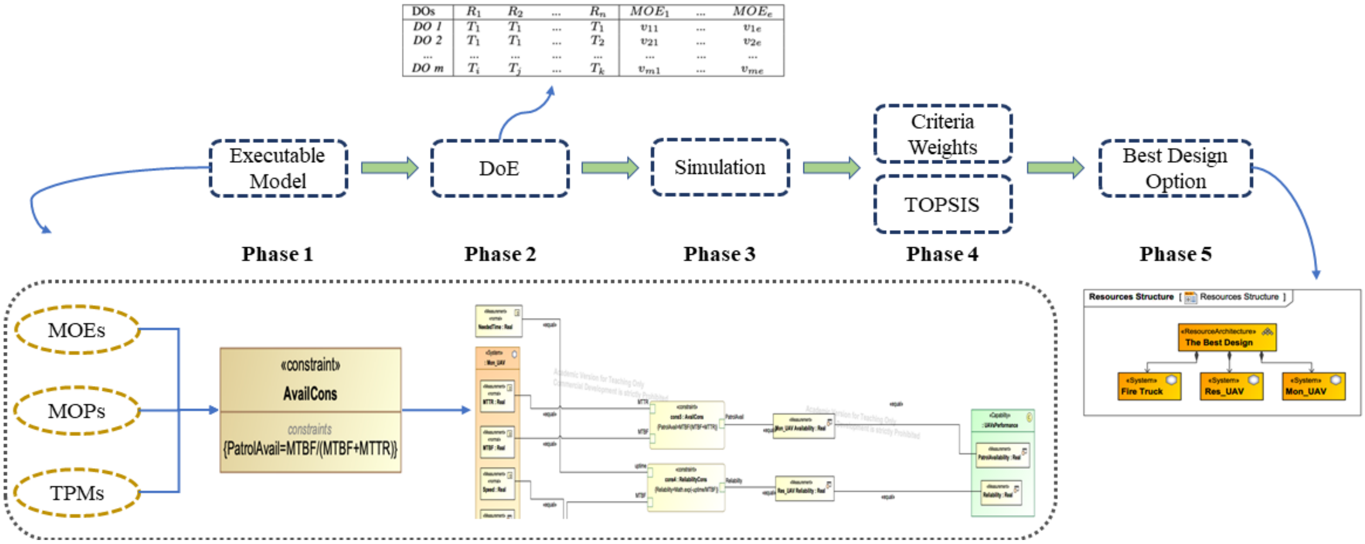


Fig. 4: Steps of the proposed methodology

UAF strategic, operational, and resources viewpoints respectively. MOEs measure how well a capability meets the needs of stakeholders, MOPs quantify the performance of operational performers, and TPMs assess the technical performance of resources. Capabilities are exhibited by operational performers which are implemented by resources. Therefore, not only do the elements of each viewpoint need to be traced, but their measures also need to be traced for consistency and evaluation purposes.

B. *Executable Model and Simulations*

As discussed in [15], an executable model needs to be created for a designed architecture to enable evaluation of its performance. To address this need, parametric diagrams support verification and validation by incorporating structural and behavioral views to evaluate the performance of a system's architecture using its technical measures [6], [16]. Simulating views of a system's architecture model enables architects to better explore designs and conduct analyses without deploying the actual system [9]. Thus, the cost of defect detection in the later stages of a system's life cycle can be greatly reduced by performing such model analyses [17], [18].

Furthermore, some studies such as [5] consider deterministic values for technical measures when performing a trade study. Whereas, these measures might be stochastic random variables in real-world settings such as speed and time. Therefore, it is necessary to incorporate Monte Carlo simulations in parametric diagrams. Studies such as [19]–[21] propose the use of external simulation software tools such as ModelCenter and MATLAB Simulink to accomplish such tasks. However, the latest improvements in systems engineering tools now offer more advanced simulation execution features to evaluate such diagrams within the modeling environment [22].

According to Kaslow et al. [23], few studies have been conducted to illustrate how to define the relation between technical measures in parametric diagrams. In this paper, we demonstrate a process that helps in understanding how to create an executable UAF model leveraging the traceability between technical measures across UAF viewpoints. The diagrams in this study were created using the Cameo Enterprise Architecture software tool. Cameo Simulation Toolkit was also employed to run Monte Carlo simulations for the parametric diagram.

III. PROPOSED METHODOLOGY

A trade study methodology for UAF architectures has been formulated by the authors of this paper [24]. Fig. 4 provides an overview of this methodology. In this paper, we only describe the details of capturing the relationships among technical measures across different viewpoints and using the resulting executable views (first and third phases in Fig. 4) for the evaluation of these measures.

A. *Traceability & Executable Model*

The first phase is concerned with designing the executable UAF model that is the foundation of a trade study. For creating an executable model in UAF, we recommend the following three steps. First, the required UAF elements along with their technical measurements should be defined, i.e., capabilities with their MOEs, operational architecture with MOPs, and potential resources with their TPMs. Second, the required constraints should be defined to calculate MOPs and MOEs given TPMs. Third, and finally, a parametric diagram should be created to include all measurements and calculations. Fig. 4 depicts this process.

1) Step 1: The only UAF views that are required for the first step are *"Strategic Taxonomy"* from the strategic viewpoint, *"Operational Taxonomy"* from the operational viewpoint, and *"Resources Taxonomy"* from the resources viewpoint. *"Strategic Taxonomy"* illustrates capabilities and MOEs. *"Operational Taxonomy"* shows a potential generic *OperationalArchitecture* element for the system and its MOPs. However, an architect can also employ multiple *OperationalPerformer* elements and define their corresponding MOPs separately. And finally, *"Resources Taxonomy"* demonstrates all possible resources and their TPMs to be used in a candidate operational architecture in order to fulfill the system capabilities.

2) Steps 2: After creating the required elements, system architects need to figure out how MOEs can be calculated given TPMs. To understand the interdependencies between technical measures, traceability matrices should be created for tracing the defined elements in the first step according to the UAF Domain MetaModel (DMM) [10], as shown in Fig. 5.

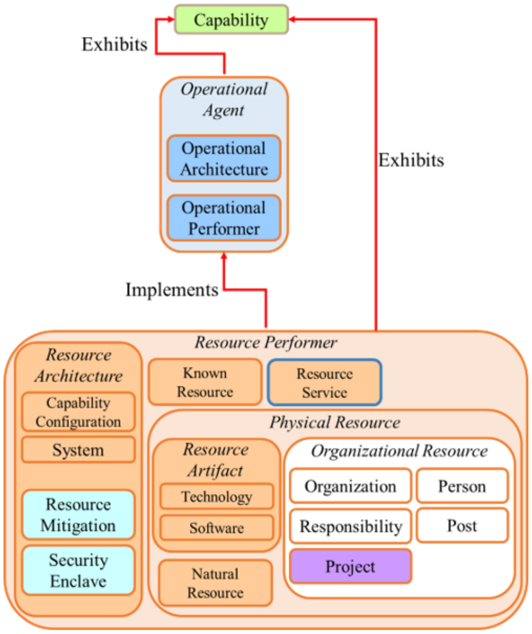


Fig. 5: Interdependencies between UAF viewpoints [10]

Traceability matrices assist with tracing back the resources and operational performers to the capabilities. UAF provides various traceability views for strategic, operational, and resources viewpoints. Architects can employ *Operational Performers to Capabilities Mapping* from the *Operational Traceability* view to show what operational performer exhibits which capability. Moreover, *Implementation Matrix* from the *Resources Traceability* view can be used for illustrating what resource implements which operational performer. As a result, TPMs can be traced back to MOPs which then be traced back to MOEs.

Upon identifying all interdependencies, equations associated with *Constraint* blocks are defined for doing the calculations. The UAF provides the architect with only *Constraint* property, however, it can be typed by a SysML constraint block, as shown in Fig. 6. The constraint blocks can be created under the *Operational Constraints* view in the UAF *Operational* viewpoint. To define the equations, architects can investigate experts' and engineers' opinions on how various TPMs can be used to calculate MOPs and how various MOPs will be employed in calculating MOEs.

3) Step 3: Once the system architecture is obtained from *step 1* and the constraint properties for calculations are identified in *Step 2*, it is time to create the executable model of the system using a UAF parametric diagram. We need a parametric diagram to evaluate our entire architecture and perform calculations for all MOEs, MOPs, and TPMs. According to [10], [14], the parametric diagram can be created for either the *OperationalArchitecture* or *ResourceArchitecture* elements in UAF. Both provide architects with the ability to include all technical measurements. Thus, the choice is left to the architect. Fig. 6 illustrates a generic UAF parametric diagram created for an *OperationalArchitecture*.

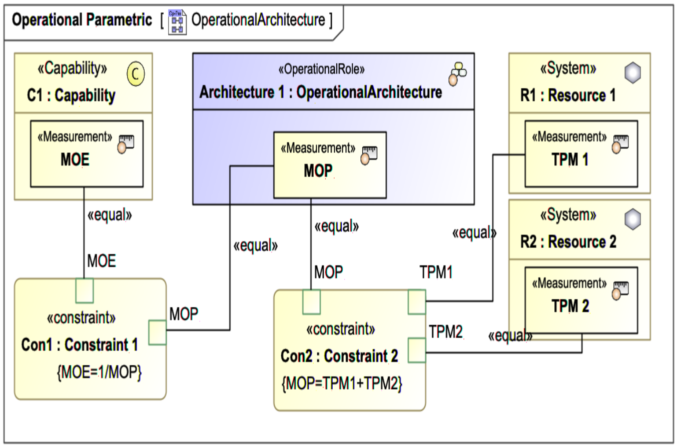


Fig. 6: UAF parametric diagram

B. Simulation

The parametric diagram needs to be executed to calculate the values of the response variables, i.e., MOEs. If all TPMs are deterministic values, a single computation of the parametric diagram is sufficient for obtaining the MOE values. However, when all or some TPMs are random variables, the Monte Carlo simulation can be employed to estimate these values.

To perform a simulation in UAF, we use a package in which we create a *SimulationConfig* diagram that builds the simulation environment for our parametric model. This diagram provides system architects with the Monte Carlo simulation setting. If the parametric model is created for the *OperationalArchitecture*, then the simulation should be created and run for this element. *Histogram* and *CSVExport* elements can also be used for visualization and saving the output results from the software tool.

IV. CASE STUDY

We demonstrate our proposed process by applying it to an illustrative example and then discussing our results. The case study involves modeling a small-scale system that can accomplish a forest firefighting mission. The objective of this case study is how traceability between technical measures can be demonstrated between the UAF viewpoints and be used in an executable model to simulate the system. The high-level operational concept for this system is depicted in Fig. 7.

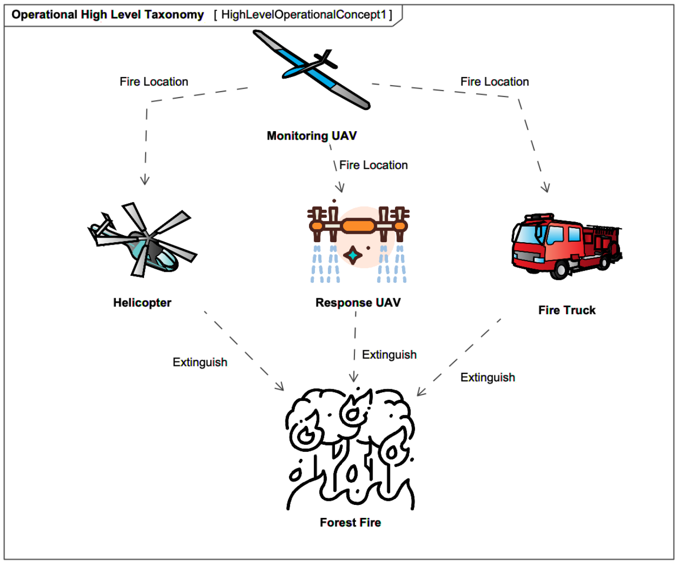


Fig. 7: Operational high-level taxonomy diagram for the forest firefighting system

A. Traceability & Executable Model

1) Step 1: Start by creating UAF *Taxonomy* views for the *Strategic*, *Operational*, and *Resources* viewpoints to capture the technical measures associated with each one shown in Figs. 8, 9, and 10.

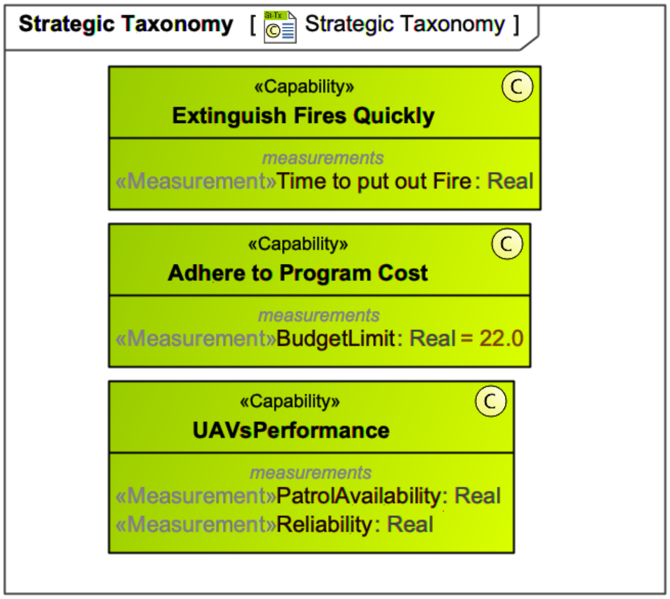


Fig. 8: Strategic taxonomy view

2) Step 2: Figs. 11 and 12 illustrate the traceability between the defined elements across the three aforementioned viewpoints. Then, equations for constraint properties are identified to assist with computing MOEs given TPMs as shown in Fig. 15.

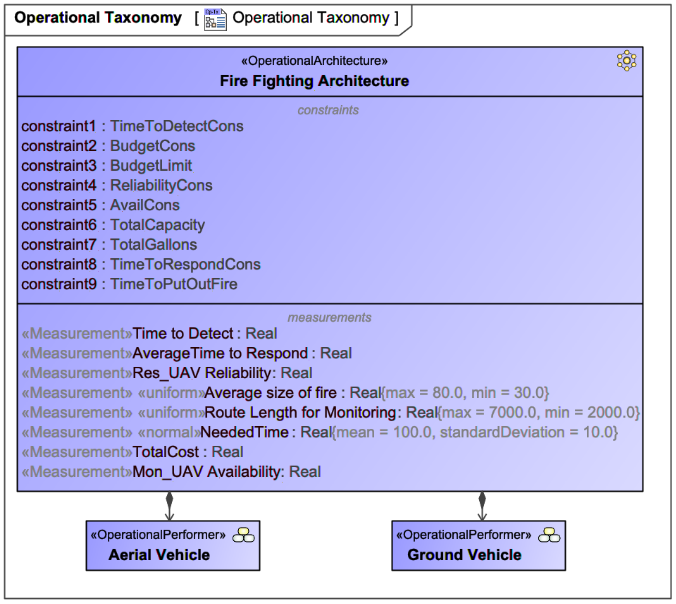


Fig. 9: Operational taxonomy view

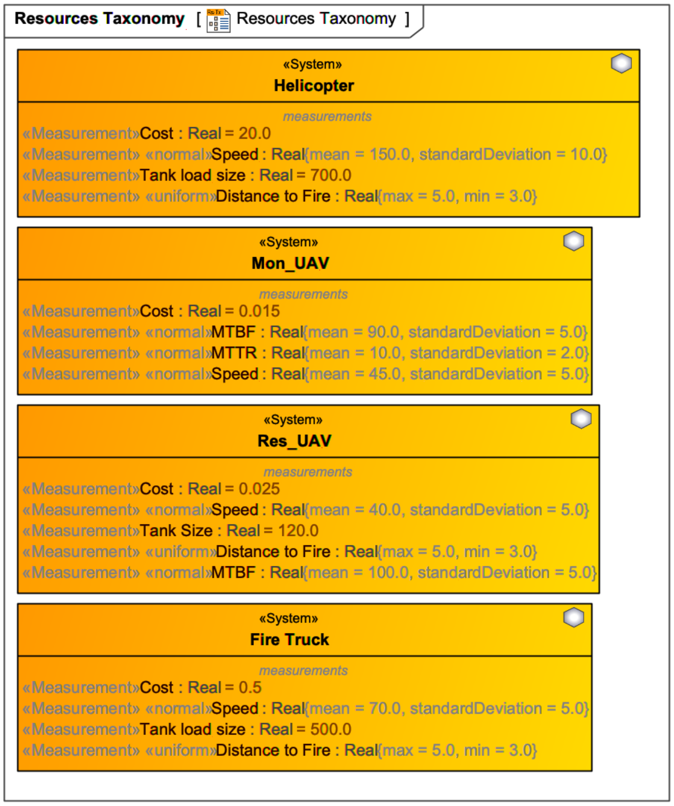
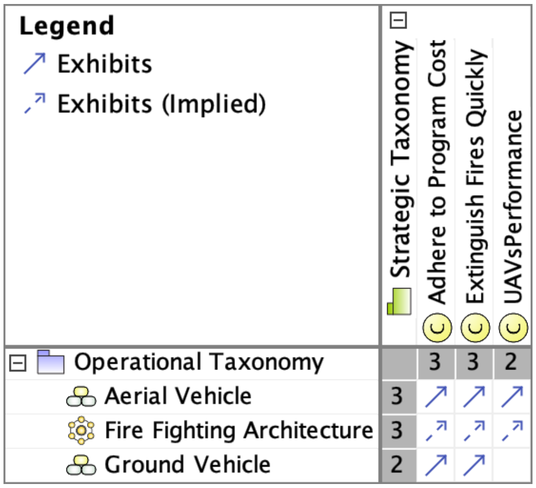


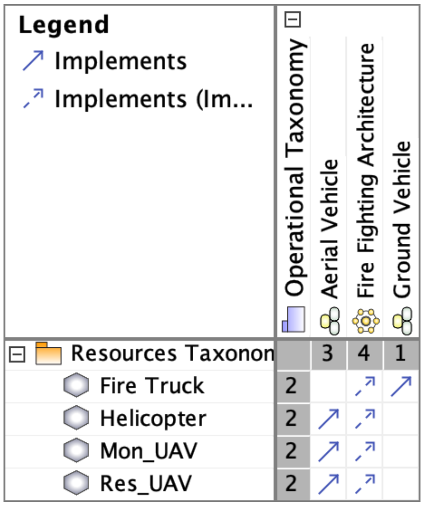
Fig. 10: Resources taxonomy view



Legend
↗ Exhibits
⇢ Exhibits (Implied)

	Strategic Taxonomy	Adhere to Program Cost	Extinguish Fires Quickly	UAVsPerformance
Operational Taxonomy		3	3	2
Aerial Vehicle	3	↗	↗	↗
Fire Fighting Architecture	3	⇢	⇢	⇢
Ground Vehicle	2	↗	↗	

Fig. 11: Operational performers to capabilities mapping matrix



Legend
↗ Implements
⇢ Implements (Im...

	Operational Taxonomy	Aerial Vehicle	Fire Fighting Architecture	Ground Vehicle
Resources Taxonon		3	4	1
Fire Truck	2		⇢	↗
Helicopter	2	↗	⇢	
Mon_UAV	2	↗	⇢	
Res_UAV	2	↗	⇢	

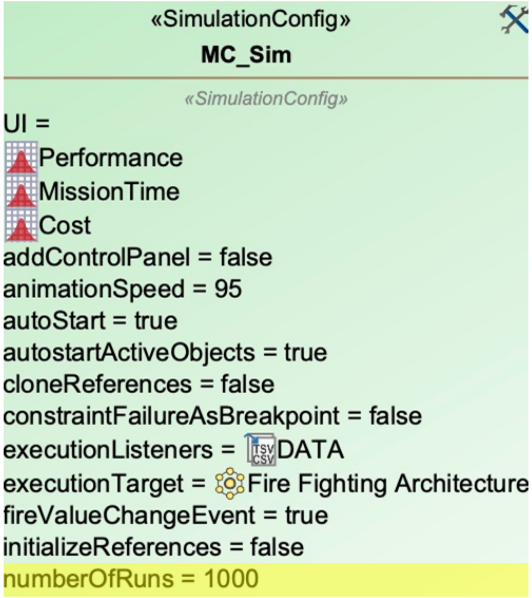
Fig. 12: Implementation matrix

3) *Step 3:* Once all constraints and the traceability between technical measures are identified, we create the parametric diagram for the *OperationalArchitecture* element as depicted in Fig. 15. In this diagram, the left side represents resources and the accompanying TPMs for them, the center shows constraints and MOPs, and the right side displays MOEs.

B. Simulation

Next, we create the simulation configuration diagram for the executable model obtained in *Step 3*. Since some of the TPMs are random variables with defined distributions, a Monte Carlo simulation with 1000 replications is run to address the uncertainty. Fig. 13 shows this simulation configuration.

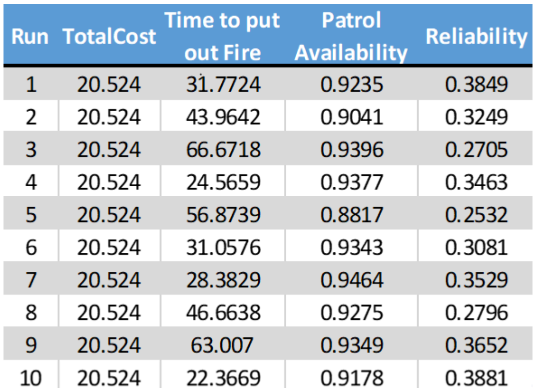
Fig. 14 shows the first ten rows of the simulation results for the firefighter system case study. As expected, the results obtained for the total cost are identical since deterministic and fixed values were used for resource costs. Whereas, the simulation results for the other MOEs show variability in numbers, reflecting the uncertainty in their corresponding TPMs.



```
«SimulationConfig»
MC_Sim
«SimulationConfig»
UI =
Performance
MissionTime
Cost
addControlPanel = false
animationSpeed = 95
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
executionListeners = DATA
executionTarget = Fire Fighting Architecture
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 1000
```

Fig. 13: Simulation configuration

Thus, the summary statistics can be gathered to compare the simulated system configurations and perform the trade study.



Run	TotalCost	Time to put out Fire	Patrol Availability	Reliability
1	20.524	31.7724	0.9235	0.3849
2	20.524	43.9642	0.9041	0.3249
3	20.524	66.6718	0.9396	0.2705
4	20.524	24.5659	0.9377	0.3463
5	20.524	56.8739	0.8817	0.2532
6	20.524	31.0576	0.9343	0.3081
7	20.524	28.3829	0.9464	0.3529
8	20.524	46.6638	0.9275	0.2796
9	20.524	63.007	0.9349	0.3652
10	20.524	22.3669	0.9178	0.3881

Fig. 14: Simulation results

V. CONCLUSION AND FUTURE WORK

UAF has provided architectures for enterprises and SOSs in both commercial and governmental domains. As previously stated, the process for rolling up TPMs to MOEs has not been documented and demonstrated. This paper is the first attempt to create an executable UAF architecture based on tracing TPMs to MOEs which helps in understanding the traceability across various UAF viewpoints. In this paper, we showcased a step-by-step process to create an executable UAF model that is based on traceability between technical measures from different viewpoints. Our process also demonstrated the implementation of a Monte Carlo simulation to address the uncertain TPMs as well as deterministic ones. The proposed process is applicable to any UAF architecture including large-scale models for SoSs and enterprises. The major limitation of this work is that the simulation of different design options

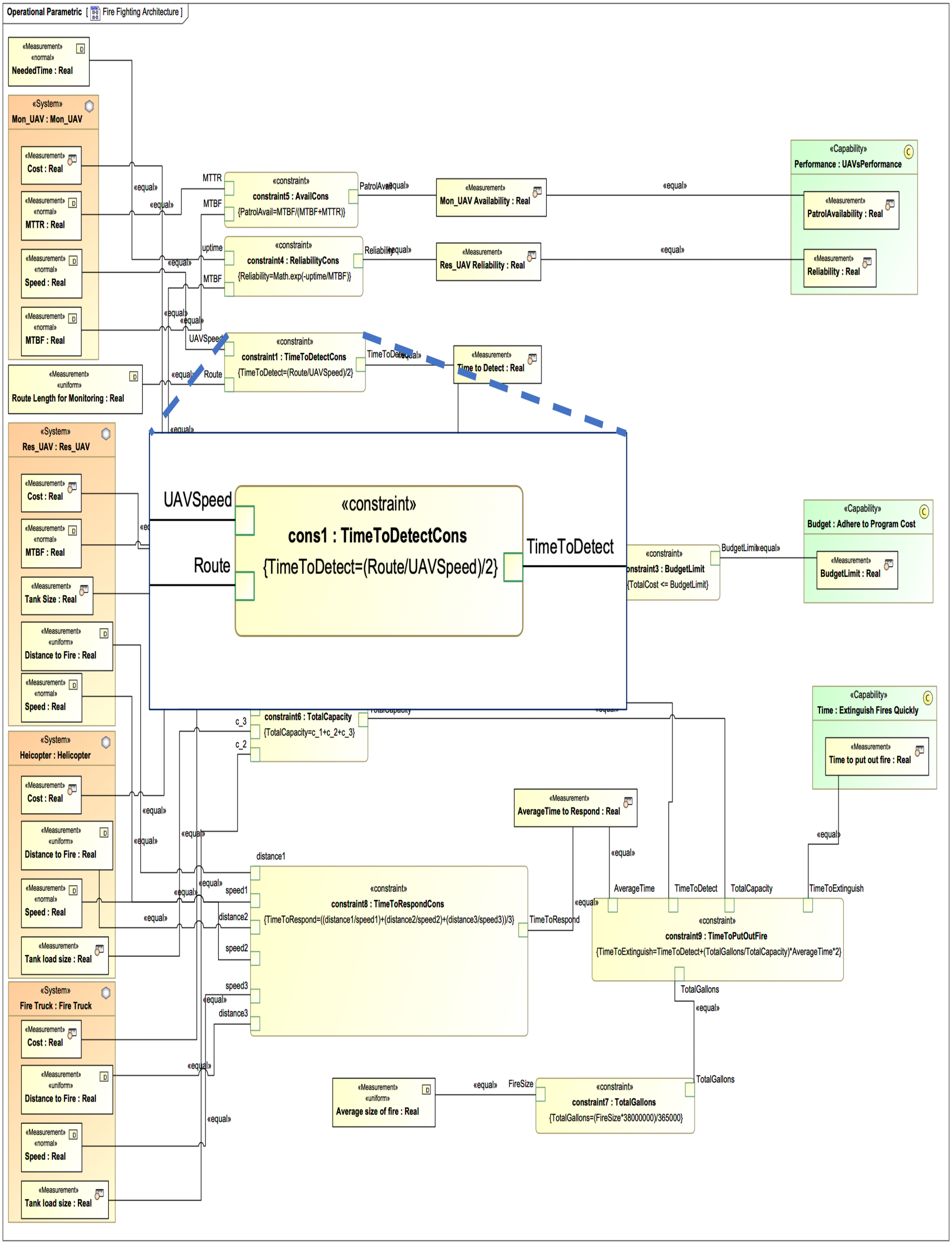


Fig. 15: Parametric diagram for the *OperationlArchitecture*

for a system's architecture can be computationally expensive using the simulation toolkit within the Cameo Enterprise Architecture software tool.

This research lays the groundwork required for performing a trade study analysis. Potential future research is to complete the proposed methodology for UAF trade studies, as shown in Fig. 4, as there is no documented methodology for a UAF trade study compliant with ISO/IEC/IEEE 15288:2015 and the INCOSE handbook.

REFERENCES

[1] A. Morkevicius, L. Bisikirskiene, and G. Bleakley, "Using a systems of systems modeling approach for developing Industrial Internet of Things applications," in *2017 12th System of Systems Engineering Conference (SoSE)*, Jun. 2017, pp. 1–6.

[2] Z. Fang, "System-of-Systems Architecture Selection: A Survey of Issues, Methods, and Opportunities," *IEEE Systems Journal*, pp. 1–12, 2021.

[3] M. Torkjazi and A. K. Raz, "A Taxonomy for System of Autonomous Systems," in *2022 17th Annual System of Systems Engineering Conference (SOSE)*, Jun. 2022, pp. 198–203.

[4] D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and T. M. Shortell, *INCOSE Systems Engineering Handbook, Fourth Edition*, 2015.

[5] J. Bankauskaite, A. Morkevicius, and R. Butleris, "Model-Based Evaluation of the System of Systems Architectures Used to Perform Trade Studies and Sensitivity Analyses," *IEEE Access*, vol. 9, pp. 114 609–114 621, 2021.

[6] R. Karban, T. Weilkiens, R. Hauber, M. Zamparelli, R. Diekmann, and A. Hein, "Cookbook for MBSE with SysML," MBSE Initiative–SE2 Challenge Team, 2011.

[7] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML*, 3rd ed., 2015.

[8] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley, Nov. 2013.

[9] J. Bankauskaite, A. Morkevicius, and R. Butleris, "Early Quality Evaluation of System of Systems Architecture Within Trade Study Process," *IEEE Access*, vol. 8, pp. 220 858–220 868, 2020.

[10] "Enterprise Architecture Guide for UAF (Informative), Appendix C, Version 1.2," 2021. [Online]. Available: https://www.omg.org/spec/UAF/1.2

[11] M. V. Cilli and G. S. Parnell, "4.3.1 Systems Engineering Tradeoff Study Process Framework," *INCOSE International Symposium*, vol. 24, no. 1, pp. 313–331, 2014.

[12] J. Bonometti and D. Beale, *The Lunar Engineering Handbook, Chapter 4: System Engineering Tools*. Auburn, AL: Auburn University. [Online]. Available: https://www.eng.auburn.edu/~dbeale/ESMDCourse/index.htm

[13] "ISO/IEC/IEEE 15288:2015(en), Systems and software engineering — System life cycle processes." [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:15288:ed-1:v1:en

[14] J. N. Martin and D. P. O'Neil, "Enterprise Architecture Guide for the Unified Architecture Framework (UAF)," *INCOSE International Symposium*, vol. 31, no. 1, pp. 242–263, 2021.

[15] L. W. Wagenhals and A. H. Levis, "Service Oriented Architectures, the DoD Architecture Framework 1.5, and Executable Architectures," *Systems Engineering*, vol. 12, no. 4, pp. 312–343, 2009.

[16] M. Hause, "The Unified Profile for DoDAF/MODAF (UPDM) enabling systems of systems on many levels," in *2010 IEEE International Systems Conference*, Apr. 2010, pp. 426–431.

[17] M. Erder and P. Pureur, "Chapter 6 - validating the architecture," in *Continuous Architecture*, M. Erder and P. Pureur, Eds. Boston: Morgan Kaufmann, 2016, pp. 131–159.

[18] M. Efatmaneshnik, S. Shoval, and K. Joiner, "System Test Architecture Evaluation: A Probabilistic Modeling Approach," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3651–3662, 2019.

[19] J. Ryan, S. Sarkani, and T. Mazzuchi, "Leveraging Variability Modeling Techniques for Architecture Trade Studies and Analysis," *Systems Engineering*, vol. 17, no. 1, pp. 10–25, 2014.

[20] R. Kawahara, D. Dotan, T. Sakairi, K. Ono, H. Nakamura, A. Kirshin, S. Hirose, and H. Ishikawa, "Verification of embedded system's specification using collaborative simulation of SysML and simulink models," in *2009 International Conference on Model-Based Systems Engineering*, 2009, pp. 21–28.

[21] J.-P. Brunet, H. Sohier, M. Yagoubi, M. Bisquay, P. Lamothe, and P. Menegazzi, "Simulation architecture definition for complex systems design: A tooled methodology," in *10th Complex Systems Design & Management (CSD&M) conference*, Paris, France, Dec. 2019.

[22] "Cameo Simulation Toolkit - CATIA - Dassault Systèmes®." [Online]. Available: https://www.3ds.com/products-services/catia/products/no-magic/cameo-simulation-toolkit/

[23] D. Kaslow, B. Ayres, P. T. Cahill, and L. Hart, "A model-based systems engineering approach for technical measurement with application to a CubeSat," in *2018 IEEE Aerospace Conference*. Big Sky, MT: IEEE, 2018, pp. 1–10.

[24] M. Torkjazi, A. Alghamdi, A. Davila-Andino, and A. K. Zaidi, "A holistic approach for system of systems trade study," 2022, [Manuscript in preparation]. Department of Systems Engineering and Operations Research, George Mason University.