

1.16 Framework for Architecture Trade Study using MBSE and Performance Simulation

Framework for Architecture Trade Study using MBSE and Performance Simulation

Jessica Ryan
George Washington University
Jryan82@gvmail.gwu.edu

Dr. Shahram Sarkani, Ph.D., P.E.
Professor of Engineering
Management and Systems
Engineering

Dr. Thomas Mazzuchini D.Sc.
Professor of Engineering
Management and Systems
Engineering and Decision Science

Abstract. Increasing complexity in modern systems as well as cost and schedule constraints require a new paradigm of system engineering to fulfill stakeholder needs. Challenges facing efficient trade studies include poor tool interoperability, lack of simulation coordination (design parameters) and requirements flowdown. A recent trend toward Model Based System Engineering (MBSE) includes flexible architecture definition, program documentation, requirements traceability and system engineering reuse. As a new domain MBSE still lacks governing standards and commonly accepted frameworks. This paper proposes a framework for efficient architecture definition using MBSE in conjunction with Domain Specific simulation to evaluate trade studies. A general framework is provided followed with a specific example including a method for designing a trade study, defining candidate architectures, planning simulations to fulfill requirements and finally a weighted decision analysis to optimize system objectives.

1 NOMENCLATURE

2 INTRODUCTION

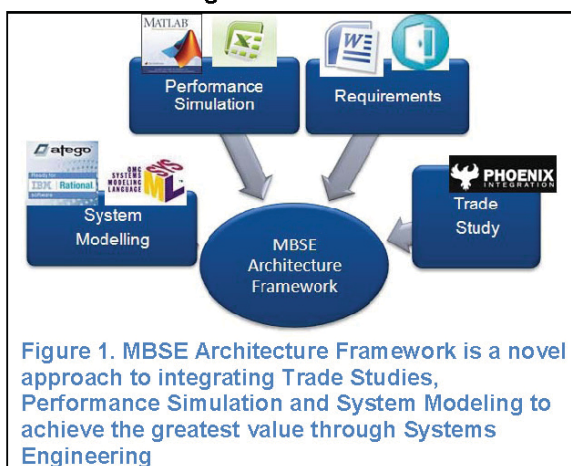
The foundation of MBSE is an object oriented design process which uses heterogeneous modeling techniques to capture system architecture, relationships, requirements and constraints [1, 2]. In many aspects MBSE complements the classical System Engineering approaches, (i.e. Waterfall, Standard 'Vee', Spiral), however document-centric processes are replaced with models which offer traceability, various viewpoints and a central repository for design information [3]. By taking advantage of the object oriented structure of MBSE, and improvements in simulation tool interoperability, a novel framework is proposed which optimizes Architecture Trade Study through MBSE and Performance Simulation integration.

Section 2 is a literature and concept review of the influencing factors in this framework. Section 3 provides a framework and demonstrates a potential application of the framework using the available tools, processes, and methods common to MBSE. A trade study is presented which varies the design parameters of a data recording system given requirements associated with cost and performance. The fundamental architecture is composed of a receiver, an analog to digital converter (A/D), a high speed buffer and a solid state data device (SSR). Section 4 is a discussion of the results, showing the benefits of the Architecture Framework using MBSE and offering a path forward for additional research.

2.1 MBSE

As defined by INCOSE "Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems" [4]. MBSE attempts to optimize the design, implementation, delivery and operation of a system throughout its entire lifecycle through modeling techniques as opposed to a standard document-centric approach.

Parameters of the system that provide the maximum benefit vary depending on the actual stakeholder and can range from reduction of cost, reduction of schedule,



increased performance, sustainability, reliability, etc [5]. As each stakeholder (customer, end user, designer, etc..) has different viewpoints in relation to the system, multiple diagrams and associations exist to address system aspects [6, 7]. Modelling of interfaces at multiple levels of abstraction aids understanding system complexity Underlying links ensure communication of the current design which is important for configuration control, insight into design assumptions, and justification of requirements allocation.. Success from the software engineering domain using UML motivated the profile extension to SysML which now includes all of UML as well as a Requirement Block Diagram and Parametric Block Diagram [8, 9]. SysML is an extension of UML based on the following four fundamental pillars: Structure, Behavior, Requirements, Parametrics. The example presented in this paper is designed using Artisan Studio SysML Profile, however this design decision is independent of the fundamental principles of the framework. The scope of this paper is limited to the usage of this language, for further description of the semantics and definitions refer to [10].

2.2 Domain Specific Simulation

Domain specific simulations are used to evaluate behavior of the system for a given set of inputs, typically requiring custom tools which can execute various system responses (continuous or non-continuous). While SysML does offer features such as automatic code generation and requirement traceability automation, SysML is not an executable language per se. The purpose of SysML is to describe the system (including architecture, requirements, associations, driving function) but is not necessarily designed to execute or evaluate simulations tied to the system behavioral response. However, because of its meta-language base, SysML can be extended to interact with domain specific tools [6, 11]. Tool interoperability remains a great challenge when modeling system parameters. Most standard tools sets (ie Matlab, Excel, Agilent ADS) offer APIs which allow various levels of interaction with the models. Co-

simulation between SysML and common industry performance analysis tools such as Simulink and Modelica is an active topic of research [12, 13]. Co-simulation can be executed in 2 manners : 1) code generation from SysML, post simulation with Domain Specific code 2) graph transformation [14, 15]. In either case, supporting custom code to bridge the gap between Domain Specific tools and SysML is required.

2.2.1 Analysis of Alternatives

Trade studies are used during an Analysis of Alternatives in a manner to determine the best system architecture [16]. Modeling and simulation are often used during an Analysis of Alternative (AoA) as a cost-effective means to assess design trades and understand their impact on system response. A reform in general defense acquisition has led to a greater appreciation for system maturity which can be rated as a System Readiness Level [17]. This paradigm shift from Performance-Based Acquisition to Capabilities-Based Acquisition shows the complexity in requirements analysis [18]. AoA can be used to assess architectures representing various System Readiness Levels factors thereby supporting Capabilities Based Acquisition.

3 FRAMEWORK

This framework provides a methodology which uses MBSE to guide the structure of Performance Simulation and the efficiency of Trade Studies to define System Architecture. To date, several framework exists for MBSE including IBM Telelogic Harmony-SE, INCOSE Object-Oriented Systems Engineering Method (OOSEM), IBM Rational Unified Process for Systems Engineering (RUP SE), Vitech Model-Based System Engineering (MBSE) Methodolog, and JPL State Analysis (SA), yet there in no uniformly accepted standard [3]. While literature exists to support integration of system modelling through SysML to domain specific tools, a framework does not yet exist to provide utility

in architecture trade study using analysis of several models.

Trade Studies are improved through identifying the driving input parameters and desired system requirements. Requirements analysis metrics are used to assess each trade response and verify the ability to meet the traced requirements.

Performance simulation benefits from abstraction as parameters including inputs, outputs, and driving interdependencies (i.e. number of bits $\langle nBits \rangle$ for an A/D) are identified for each block (hardware component). In an object-oriented fashion, these parameters are abstracted such that any A/D would have a given set of parameters. A particular instance of an A/D has a discrete value for each parameter (relationship between a class versus instance). After inputs, output, interdependencies, and purposes for each simulation are identified, various algorithms can be tested. Defining simulation interfaces gives flexibility to later increase the fidelity of the model, or change methods by which the outputs are determined (ie various methods of SNR calculation).

3.1 Process

The framework follows a process which emphasizes requirements analysis, trade study definition, and performance simulation to determine the optimal architecture. The following sections refine each stage of the framework process presented in Figure 2, including a proof-of-concept example. A trade study of a data recording system is evaluated, with screenshots to help clarify the tools and processes used in each stage.

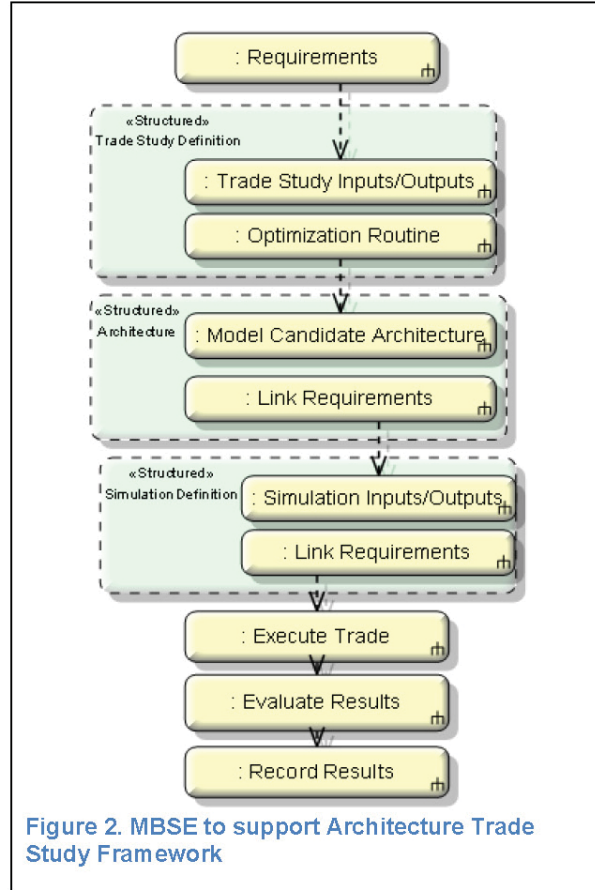


Figure 2. MBSE to support Architecture Trade Study Framework

3.2 System Requirements

Similar to conventional System Engineering Vee process, the initial stage is the identification of stakeholder needs which represent system requirements. System requirements must be gathered and documented, which is often maintained in a requirements repository. In this example we used DOORS which allows for traceability between requirements in terms of associations, derivation and flow-down (see Figure 3). Automatic synchronization allows requirements stored in DOORS to be imported and synchronized to requirements objects in SysML (see Figure 4). SysML offers a requirements diagram to show hierarchy and association of such requirements objects. Additionally these requirements are objects in the model they can be explicitly added to Block Definition Diagrams (bdd) or Constraint Diagrams where they can be linked to identify relationships. This linkage provides

traceability, documentation and extends communication to designers to describe why particular design decisions have been identified.



Figure 3. Requirements captured in DOORS

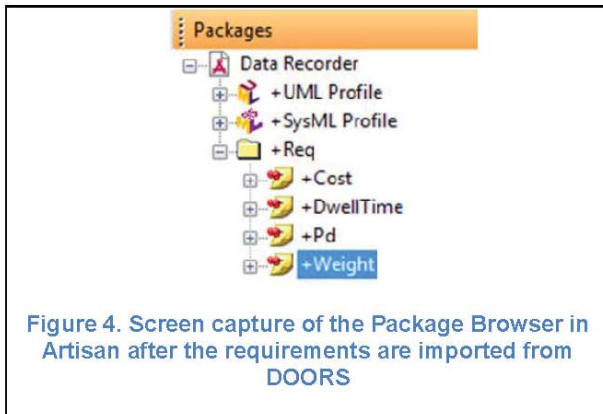


Figure 4. Screen capture of the Package Browser in Artisan after the requirements are imported from DOORS

3.3 Trade Study

Following definition of system requirements, the focus shifts to the motivation and evaluation of the Trade Study. The trade study scope must be identified and documented, encompassing all valid architecture variations and scenario alternatives. From an operational research technique, factors of the trade study can be identified as Objectives, Constraints, Decision Variables or Parameters. Inputs to the study are Decision Variables or Parameters. Requirements can be describes as Objectives (i.e. must maximize dwell time) or Constraints (i.e. must not have a buffer input rate overflow) [16]. This division of requirements allows the optimization routine to select the system that best satisfies the minimum goals of the stakeholders, while giving insight into effective system parameters which can lead to the greatest value. Inline with the Capabilities-Based acquisition trend, this

optimization supports identification of the '80 percent solution' [19].

Optimization Output Parameters	
Objective	Goal
total_cost	min
maxDwellLength	max
Constraint	
bf_InError	FALSE
bf_OverflowErr	FALSE

Figure 6. Defining the output parameters for optimization.

Figure 5 shows the inputs to the trade study for this example. Input parameters that can vary in the trade study include the component model type(s), the number of components (i.e. number of buffers can vary from 1-2), and dwell time. Trade study inputs (and supporting parameters associated with the selection of those inputs) are entered into ModelCenter for execution of the trade study.

Inputs	Trade Space Inputs		
	Options		
amp	AMP_001	AMP_002	
a2d	A2D_001	A2D_002	A2D_003
buffer	BF_001	BF_002	
nbuffers	1-3		
ssr	SSR_001	SSR_002	
nssr	1-10		

Figure 5. Defining the trade space

Also entered into ModelCenter are the objectives and constraint requirements identified in Figure 6. The DARWIN algorithm optimizer analyzes simulation outputs against set Objectives and Constraints when selecting a final design. In this example we attempt to minimize cost and maximize dwell time (Objectives), so long as a buffer overflow has not occurred or A/D sampler ate to buffer mismatch has not occurred (Constraints).

Optimization Output Parameters	
Objective	Goal
total_cost	min
maxDwellLength	max
Constraint	
bf_InError	FALSE
bf_OverflowErr	FALSE

Figure 6. Defining the output parameters for optimization.

3.4 Architecture

After identifying what the system needs to do, and how it will be used, potential architectures can be modeled. In an object oriented fashion each block will have critical parameters which can be defined on a per-block basis. Figure 7 is an example of a Block Definition Diagram of the data recorder system. In this viewpoint we want to look at all of the components of the system therefore attribute information is hidden. By examining the system in various levels of detail, each diagram and viewpoint can serve a different purpose. Each distinct viewpoint uses information from a central data repository, thereby establishing consistency amongst objects in the model.

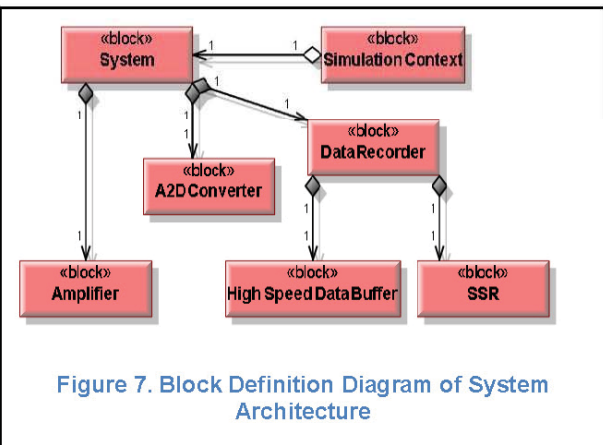
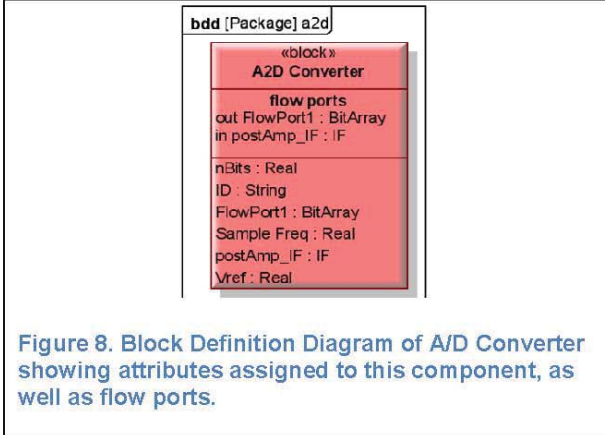


Figure 8 shows a Block Definition Diagram of just the A/D component where all of the attributes are visible. These attributes are abstracted such that each model of A/D given in the Trade Space Inputs can have variable

parameters (ie. A2D_001.nBits = 12 bits, A2D_002.nBits = 18, A2D_003.nBits = 32 Bits). Due to abstraction principles, these instances themselves do not need to individually be modeled, as the associated instance information is documented in the trade study evaluation.



Requirements particularly linked to architecture can be visualized through association, satisfy and refine connections on SysML Block Definition Diagrams (or Parametric Diagrams). Such links allow for traceability when either the block information changes which could affect the ability to satisfy a requirement, or if the requirement changes in which case the selection of a block may need to be re-evaluated.

3.5 Simulation

Once the requirements are understood, the trade space is defined and potential architectures have been identified, simulation models can be designed. Simulation often requires domain – specific tools. The critical foundation for integrating these tools is a defined metadata language, APIs provided by the domain tools or custom interface software. By identifying the simulation inputs and outputs, complex scenarios can be shared amongst a group of engineers with increased communication and interoperability. Identifying requirements associated with each simulation tailors those simulations for specific goals, thereby limiting their scope and

clearly defining the purpose of each simulation. Again, requirements objects can be linked to blocks representing simulations in the model for traceability purposes.

Error! Reference source not found. shows a block diagram of Performance Simulations required in the Trade Study evaluation of the Data Recorder Model (note – this also includes simulations not yet implemented in the framework example). It is obvious from this example that the Performance Simulations can each have an independent focus or work together to fulfill a system requirement.

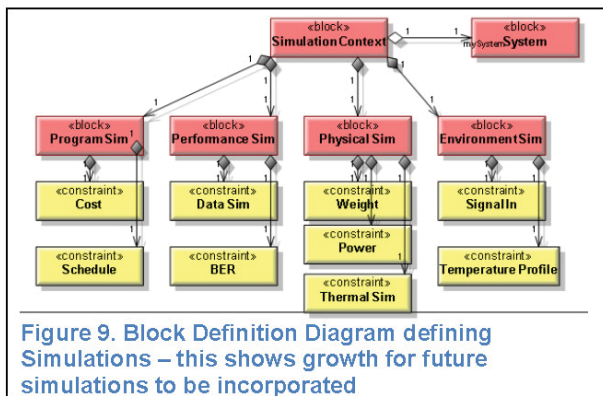


Figure 9. Block Definition Diagram defining Simulations – this shows growth for future simulations to be incorporated

Parametric Diagrams can represent the structure of the Performance Simulation. Inputs and outputs can be identified as ports on Constraints Blocks, and these inputs and outputs can be linked to attribute value fields on the architecture. By first creating a Block Definition Diagram and associating the System Block to the Simulation), then attributes from components in the System Block can be linked to Constraints Block inputs or outputs. In the 'Data Sim' Parametric Diagram figure below requirements are linked, inputs and outputs of the simulation are identified and an example exists of linking a, 'A2D Converter' block attribute 'nBits' to an input of the 'Data Sim' performance simulation. Additionally, requirements are linked to the Constraint Block, again using MBSE techniques to enhance communication and traceability.

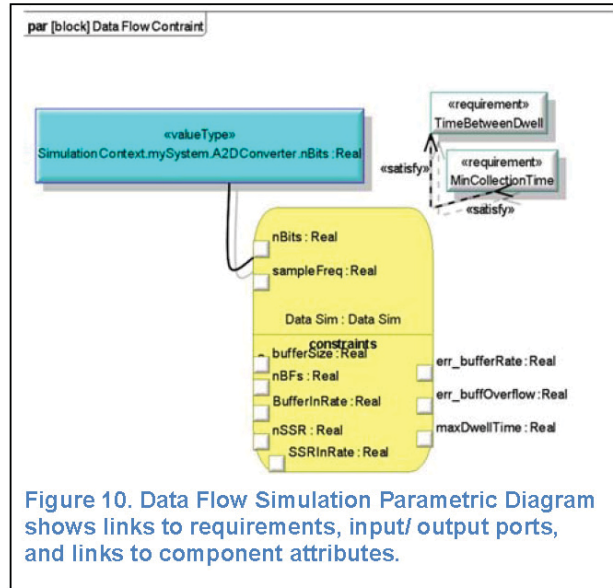


Figure 10. Data Flow Simulation Parametric Diagram shows links to requirements, input/ output ports, and links to component attributes.

3.5.1 Cost Model (excel)

The 'Cost Model' represents a specific example of a Simulation that models the cost of the system based on inputs from the Trade Study and an Inventory Model (Figure 11). Inputs driven by the Trade Study include component selection, compared to an Inventory Model which defines component stock level and cost to compute a final cost (with the added min lot buy complexity). The output computed (

	Inventory			
	In Stock	cost per	min lot buy	order time
AMP_001	1	1000	20	1
AMP_002	2	1500	30	1
A2D_001	0	4000	10	2
A2D_002	3	5000	5	3
A2D_003	2	6000	10	1
BF_001	1	300	2	2
BF_002	0	200	2	3
SSR_001	2	10000	1	4
SSR_002	3	12000	1	5

Figure 11. Inventory model representing components in stock, price, and min lot buy requirements

The output (Figure 12) which will be returned to the overall trade study is the final cost of the system for a given set of selected inputs.

	Inputs		Outputs		
	Part Id	n req	n bought	cost	schedule (mo)
amp	AMP_002	1	1	\$ 1,500	0
a2d	A2D_001	1	10	\$ 40,000	2
buffer	BF_001	2	2	\$ 600	2
ssr	SSR_001	1	1	\$ 10,000	0

Figure 12. Example of the output of the cost model given a scenario from the Trade Study inputs

3.5.2 Dwell Time (Matlab)

In this example one of the driving requirements is a given dwell time. A maximum dwell time can be computed based on the output data rate of the A/D, the input rate of the buffer, the size of the buffer memory, the output rate of the buffer and the input rate of the SDDR. Less data from the A/D increases dwell time because not as much data needs to be stored in a given time. Increasing the buffer size increases dwell time because it takes more time to fill the buffer prior to overflow. Increasing the SDDR input rate (or buffer output rate) increases dwell time because the data can flow from the buffer faster, therefore it takes longer for a buffer overflow to occur. In any of these cases, the opposite had a negative impact on dwell time. If the data rate at the output of the A/D exceeds the input rate to the buffer a 'bf_inError' occurs. If the max dwell time computed for a given architecture exceeds the requirement dwell time a 'bf_Overflow Error' occurs. Both errors are output to the Trade Study in ModelCenter (as Constraints) along with the computed 'max dwell time' (as an Objective) which will be used to optimize the dwell time versus cost.

While Matlab and Excel are tools that often communicate, the point of this framework is that by defining the trade study, modelling the simulations ahead of time, properly constraining the simulations, and defining the interdependencies, more complicated simulations can be performed and eventually additional simulations can be incorporated.

3.6 Execute & Evaluate

Finally the trade study is executed on an integrated set of models, and evaluated against the requirements. In this example the simulation models are linked together using ModelCenter. Each execution, Model Center varies the inputs of the trade study, attempting to find the best architecture to meet the needs of the optimizer routine. Again custom software could drive the execution of these simulations in concert, understanding that the driving inputs are generated from the trade study, architecture model and prior simulation output.

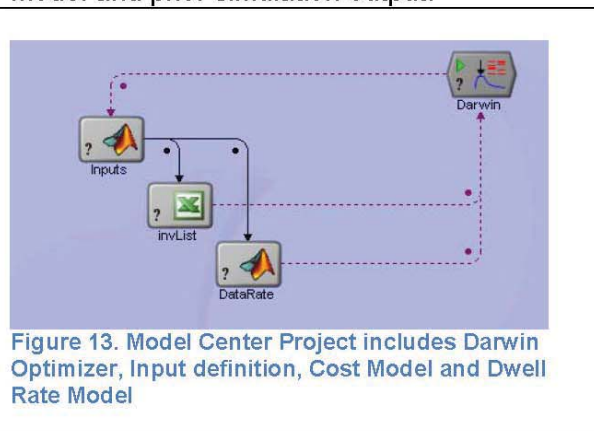
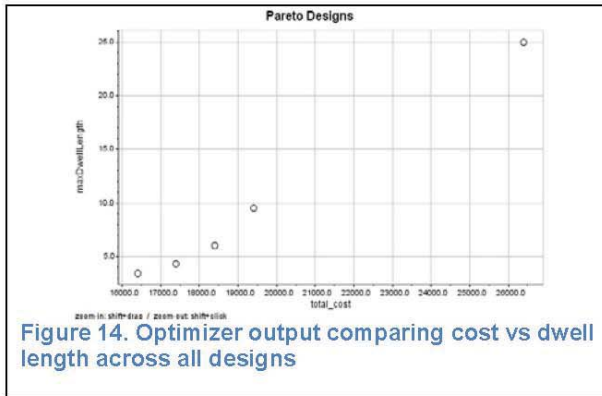


Figure 13. Model Center Project includes Darwin Optimizer, Input definition, Cost Model and Dwell Rate Model

3.7 Analyze Results

The evaluation stage includes assessing the results of the optimization routine. The optimization routine included in ModelCenter Darwin is selected for this example. Decision evaluations such as weighted matrix could also be used to evaluate the trade study results. Figure 14 is an example of the comparison of two Objective requirements (Cost, MaxDwellTime) resulting from input parameter variation. The optimal design minimizes total cost while maximizes dwell time.



After determining the optimum trade study result, (resulting in final architecture definition), the results of the study should be written back into the model for configuration control. ModelCenter displays the selected design (ie. Model.Input. A2D = 2 means that A2D_002 was selected as the converter in the optimal design). This can be supported through a Block Definition Diagram of instances. Again, custom software must be written to integrate the output of the trade study into the SysML models.

Design 1	
Objective(s)	
Model.invlst.total_cost	= 16400.0000000
Model.DataRate.maxDwellLength	= 3.4482759
Design Variables	
Model.Inputs.dwell_time	= 0.4000000
Model.Inputs.amp	= 1
Model.Inputs.a2d	= 2
Model.Inputs.bf	= 2
Model.Inputs.n_bf	= 2
Model.Inputs.ssr	= 1
Model.Inputs.n_ssr	= 1
Constraints	
Model.DataRate.bf_InError	= 0.0000000
Model.DataRate.bf_Overflow	= 0.0000000
Watch Variables	

Figure 15. Output of ModelCenter showing design which lead to the optimal requirements analysis

4 DISCUSSION

The framework proposed supports a methodology to integrate MBSE and Performance Simulation to optimize Architecture Trade Studies. Inherent in this framework is the novel approach to modeling simulations which provides definition of

simulation inputs and outputs as well as explicitly linking those simulations to requirements. Benefits of this framework include 1) current configuration of design decision 2) increased communication between system engineers for performance simulation 3) explicit traceability between trade studies and requirements 4) potential for system engineering reuse (including models, hardware, software and analysis tools). Additional case studies will give insight into the scalability of this framework and add fidelity to the integration of SysML directly with Performance Simulations (again tool interoperability). As new tools are developed, and further examples of explicit program application emerge, this framework will provide a guideline for architecture analysis through MBSE and Performance Simulation in the System Engineering user community.

5 REFERENCES

- [1] M. R. Blackburn, "What's model driven engineering (MDE) and how can it impact process, people, tools and productivity," System and Software Consortium (SSCI), Tech. Rep. SSCI-2008002-MC, Sept 2008.
- [2] J. Axelsson. (2002, Model based systems engineering using a continuous-time extension of the unified modeling language (UML). *Systems Engineering* 5(3), pp. 165-179.
- [3] J. Estefan, "Survey of model-based systems engineering (MBSE) methodologies," Jet Propulsion Laboratory, Pasadena, California, U.S.A., 2007.
- [4] INCOSE, "IncoSE system engineering handbook," 2010.
- [5] A. L. Ramos, "Model-Based Systems Engineering: An Emerging Approach for Modern Systems," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 2010.
- [6] M. Hause, "The Unified Profile for DoDAF/MODAF Enabling Systems of Systems on Many Levels," *IEEE*, 2010.
- [7] Rhodes, D.H., and A.M. Ross, "Anticipatory capacity: Leveraging model-based approaches to design systems for dynamic futures," in *International Conference on Model-Based System Engineering*, 2009, pp. 46.
- [8] R. Rasmussen, M. Ingham and D. Dvora, "Achieving control and interoperability through unified model-based systems and software engineering," Sept 2005.
- [9] J. O. Grady. (2009, Universal architecture description framework. *Systems Engineering* 12(2), pp. 91-116.
- [10] S. Friedenthal, *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann, 2008.
- [11] C. J. J. Paredis and T. Johnson. Using OMG'S SYSML to support simulation. Presented at Simulation Conference, 2008. WSC 2008. Winter.
- [12] T. A. Johnson, "Integrating Models and Simulations of Continuous Dynamic System Behavior into SysM," 2008.
- [13] R. Kawahara, H. Nakamura, D. Dotan, A. Kirshin, T. Sakairi, S. Hirose, K. Ono and H. Ishikawa. Verification of embedded system's specification using collaborative simulation

- of SysML and simulink models. Presented at Model-Based Systems Engineering, 2009. MBSE '09. International Conference on.
- [14] D. B. Russell and W. K. McQuay. The joint modeling and simulation system a common modeling architecture for the dod. Presented at Simulation Conference Proceedings, 1993. Winter.
- [15] L. Xing-hua and C. Yun-feng. Design of UAV flight control system virtual prototype using rhapsody and simulink. Presented at Computer Design and Applications (ICCD), 2010 International Conference on.
- [16] A. P. Sage and W. B. Rouse, Eds., *Handbook of Systems Engineering and Management*. Hoboken, NJ: John Wiley & Sons, 2009.
- [17] B. Sauser, J. Ramirez-Marquez, R. Magnaye and W. Tan, "A Systems Approach to Expanding the Technology Readiness Level within Defense Acquisition," *International Journal of Defense Acquisition Management*, vol. 1, 2008.
- [18] United States Government Accountability Office, "DEFENSE ACQUISITIONS: Assessments of Selected Weapon Programs," 2010.
- [19] R. Gates, "Remarks by Secretary of Defense Robert Gates at the Army War College, Carlisle, Pa." *Office of the Assistant Secretary of Defense (Public Affairs)*, .

6 ACKNOWLEDGEMENT

I would like to acknowledge several colleagues who have in some way or another positively contributed this body of knowledge. Rebekah Ceremae assisted in supporting literature reviews. Her ability to find relevant and recent information greatly shaped the definition of the research efforts.. Sean McGervey, who is currently developing a novel the foundation for MBSE efforts at Northrop Grumman guided the research by describing issues facing current MBSE practitioners. Brad Longo assisted through custom tool development, in addition to integration of the performance simulations in Model Center.