

# A Model-Driven Visualization Tool for Use with Model-Based Systems Engineering Projects

Kathryn Trase  
NASA Glenn Research Center  
21000 Brookpark Road  
Cleveland, OH 44135  
kathryn.trase@nasa.gov

Eric Fink  
Ohio Aerospace Institute  
22800 Cedar Point Road  
Cleveland, OH 44142  
ericwfink@gmail.com

**Abstract**— Model-Based Systems Engineering (MBSE) promotes increased consistency between a system’s design and its design documentation through the use of an object-oriented system model. The creation of this system model facilitates data presentation by providing a mechanism from which information can be extracted by automated manipulation of model content. Existing MBSE tools enable model creation, but are often too complex for the unfamiliar model viewer to easily use. These tools do not yet provide many opportunities for easing into the development and use of a system model when system design documentation already exists. This study creates a Systems Modeling Language (SysML) Document Traceability Framework (SDTF) for integrating design documentation with a system model, and develops an Interactive Visualization Engine for SysML Tools (InVEST), that exports consistent, clear, and concise views of SysML model data. These exported views are each meaningful to a variety of project stakeholders with differing subjects of concern and depth of technical involvement. InVEST allows a model user to generate multiple views and reports from a MBSE model, including wiki pages and interactive visualizations of data. System data can also be filtered to present only the information relevant to the particular stakeholder, resulting in a view that is both consistent with the larger system model and other model views. Viewing the relationships between system artifacts and documentation, and filtering through data to see specialized views improves the value of the system as a whole, as data becomes information.<sup>1</sup>

## TABLE OF CONTENTS

1. INTRODUCTION .....	1
2. SDTF AND INVEST MOTIVATION.....	2
3. SDTF AND INVEST USE CASES.....	2
4. SDTF DESCRIPTION .....	3
5. INVEST OVERVIEW .....	5
6. IMPLEMENTATION OF THE SDTF AND INVEST .....	6
7. MAINTENANCE OF THE SysML MODEL AND INVEST .....	8
8. FORWARD WORK.....	9
9. SUMMARY .....	9
REFERENCES.....	9
ACKNOWLEDGMENTS.....	10

## 1. INTRODUCTION

The advent of Model-Based Systems Engineering (MBSE) is promising to enhance the ability of systems engineers to capture system design data in a structured manner, and format that data in a variety of ways to help communicate the system design to stakeholders. MBSE enables greater consistency among the variety of design artifacts deployed in communicating the design, allowing the systems engineer to focus on the design and limit the emphasis on design documentation.

The transition from a traditional document-centric systems engineering approach to a model-based approach will not be immediate. The learning curve associated with MBSE tools, the emphasis of system design processes on document artifacts, and the relative immaturity of integrated MBSE tools each contribute to the difficulty in overcoming the inertia of current systems engineering practices. Existing MBSE tools have limited ‘out-of-the-box’ integration capability with other design tools and environments, which limits the ease at which MBSE techniques can be applied to current projects.

The SysML [1] Document Traceability Framework (SDTF) and associated Interactive Visualization Engine for SysML Tools (InVEST) were developed by the authors to ease the transition from traditional systems engineering methods to MBSE techniques. The SDTF provides a structure for integrating existing design documentation within a SysML model, and InVEST automates the process of summarizing model content in a format that is more intuitive to the user. Integrating existing systems documentation within the system model allows previous work to be fully leveraged without committing the design team to full MBSE implementation, and the automated production of various model extractions facilitates the use of the model by the design team.

Section 2 of this paper summarizes the authors’ motivation to develop the SDTF and InVEST. Section 3 describes several use cases of this tool. Section 4 presents the SDTF from which visualizations are created, described in Section 5. Section 6 discusses the implementation of the SDTF in a model and the details of InVEST’s design. Section 7 summarizes maintenance of the SysML model and InVEST. Forward work is discussed in Section 8, and Section 9 summarizes the effort.

<sup>1</sup>U.S. Government work not protected by U.S. copyright  
978-1-4799-1622-1/14/\$31.00 ©2014 IEEE

## 2. SDTF AND INVEST MOTIVATION

For a complex system under development using traditional systems engineering processes, hundreds of design documents and verification reports from test activities or analyses, are generated. System reliability was of particular interest to the project that initiated the SDTF and InVEST development. A wide variety of documentation and information is required to prove the reliability or flight readiness of a system, including specifications, test procedures and data, analysis reports, manufacturing procedures, process controls, and design standards. Communicating the source of reliability estimates (or any other technical parameter) to stakeholders when there may be hundreds of standalone source documents becomes challenging.

Many different avenues of investigation might be followed through documentation to sufficiently demonstrate to the system user the soundness of the design approach, acceptable failure mode mitigation, and the adequacy of verification and validation efforts. A list of reference or supplemental documents is generally provided in a test report or analysis summary, but this list may be incomplete or outdated as documentation is updated over the course of the design.

Configuration management systems and document repositories are widely used to track documentation and enforce version control, but these systems can be cumbersome for the user searching for information. Typically stored in conventional file folder structures, the compartmentalized nature of these repositories prevents an independent reviewer from organically exploring the system design by following his or her individual line of reasoning. These repositories may interface to some extent with requirements management tools or engineering design tools, but generally provide only a hyperlink between the systems, and do not provide additional opportunities for data exchange.

Requirements traceability is heavily emphasized in the systems engineering discipline, [2] and robust design approaches integrate requirements verification and validation in the traceability process, but these practices are not adequate for supporting reliability estimates. Verification traceability matrices will establish the linkages between documentation and the requirement(s) verified by the described test or analysis, but are typically not well integrated with reliability estimation techniques. Verification of a few reliability requirements may include hundreds of test summaries, worst case analyses, and failure mode mitigation analyses for complex systems.

A document traceability framework is necessary to provide a formal mechanism to organize and structure system documentation into a web of information that would facilitate the independent design review process. The objective of this effort was to develop a generic framework that could be applied to any project, including those already

under development, and was not specific to a particular project and its documentation. The document traceability framework will facilitate communication of the system design to a variety of stakeholders during their review of system flight readiness.

### *Blending MBSE Constructs with Software Reporting Tools*

MBSE advertises the ability to extract information from any variety of model viewpoints relevant to a subset of stakeholders, while simultaneously maintaining consistency between all viewpoints for all stakeholders. An ultimate goal of extracting these views is the generation of documentation from the model itself. [3], [4]

In their current form, many modeling tools are complex and require specialized training to not only view and navigate the model, but also to understand the model and extract information from it. In most workplaces beginning to implement MBSE methodologies and processes, there are reservations against immediate, total transition to MBSE because of the investment required and nebulous return on investment. [5]

For these reasons, MBSE adopters must initially find ways to blend MBSE practices with both existing system documentation, and traditional document-centric systems engineering, to facilitate a gradual transition to MBSE. Previous work has identified the continued place for documentation within the MBSE philosophy. [6]

A document traceability framework that incorporates the system documentation with a MBSE model can effectively integrate traditional systems engineering deliverables and a MBSE model in a manner that is value-added and not redundant. The use of a document traceability framework in conjunction with software developed to extract information from the model will aid customers of model data in evaluating the system design. The framework provides the relationships needed for a variety of information extraction viewpoints, and the software generates the viewpoint in the selected visualization format and links the viewpoint back to the document repository. A software tool that exports and summarizes data from the MBSE model in familiar formats will enable reviewers to explore the integrated documentation from a variety of perspectives.

## 3. SDTF AND INVEST USE CASES

A SysML framework within a modeling tool enables automation of the data validation process, which would otherwise be a largely manual process. Use of a software reporting tool offers non-modelers the ability to participate in the validation and review process, and leverage the automated visualization capabilities made possible by a system model. InVEST generates visualizations from the model, thus maintaining consistency among individual visualizations so the customer can access design information uniformly along any line of reasoning. In any visualization, InVEST should provide means of directing the reviewer to

the related documentation (through the use of hyperlinks to a document repository, e.g.).

Organizations deploying MBSE have noted that not every team member must be intimately familiar with the model, or even know how to model [7], but existing tools do not provide enough functionality to adapt model views to meet a particular stakeholder's needs. InVEST should extract and reformat model information in a manner that enables a tiered approach to data interaction: provide the stakeholder with information at the appropriate level of detail.

Example use cases of this combined approach include the following:

*Has every failure mode of a given component been investigated and appropriately mitigated?*

The SDTF should allow automated summaries of analyzed failure modes and provide the user insight into the effectiveness of applied mitigation techniques. InVEST should collate and visualize the automated summaries from the modeling tool in a manner that communicates the relative level of completion of failure mode studies to the reviewer.

*What design documentation is available for this specific system component?*

The SDTF should enable listing of all design documentation associated with a given system component. This use case might facilitate a component expert in finding the pertinent information needed for his or her review. InVEST should tabulate the design documentation characteristics in an environment that is more readily familiar to a general user than a system modeling tool.

*Several components are made of the same material, which recently underwent a specification update: Which analyses require revision?*

The SDTF should incorporate information relating to component materials, the components made of those materials, and the corresponding reports discussing material analysis. InVEST should filter and present component information and documentation based on the selected material, and conversely, present material information based on the selected component.

*What documentation provides the verification and/or validation evidence of a given requirement?*

The SDTF should enable design documentation to interface with SysML requirements blocks and relationships. The visualization tool should assist reviewers in interpreting the

state of requirements verification and validation completeness.

*What standards and policies were followed in a given analysis?*

The SDTF should provide traceability from analyses back to the standards and policies which prescribed the acceptable methods or parameters used in the analysis. The reporting tool should enable the reviewer to assess the application of standards and policies at the system level, and review the system elements that have been designed in accordance with these standards.

## 4. SDTF DESCRIPTION

The SDTF provides a mechanism for integrating traditional engineering documentation within the physical system model, requirements model, failure modes, and the design environment relating to standards and project plans. A series of stereotypes were created to differentiate between the various extensions of SysML blocks. Figure 1 depicts the SysML framework in block notation. The following subsections will describe the four major classes and their properties.

### *SystemPart Stereotype*

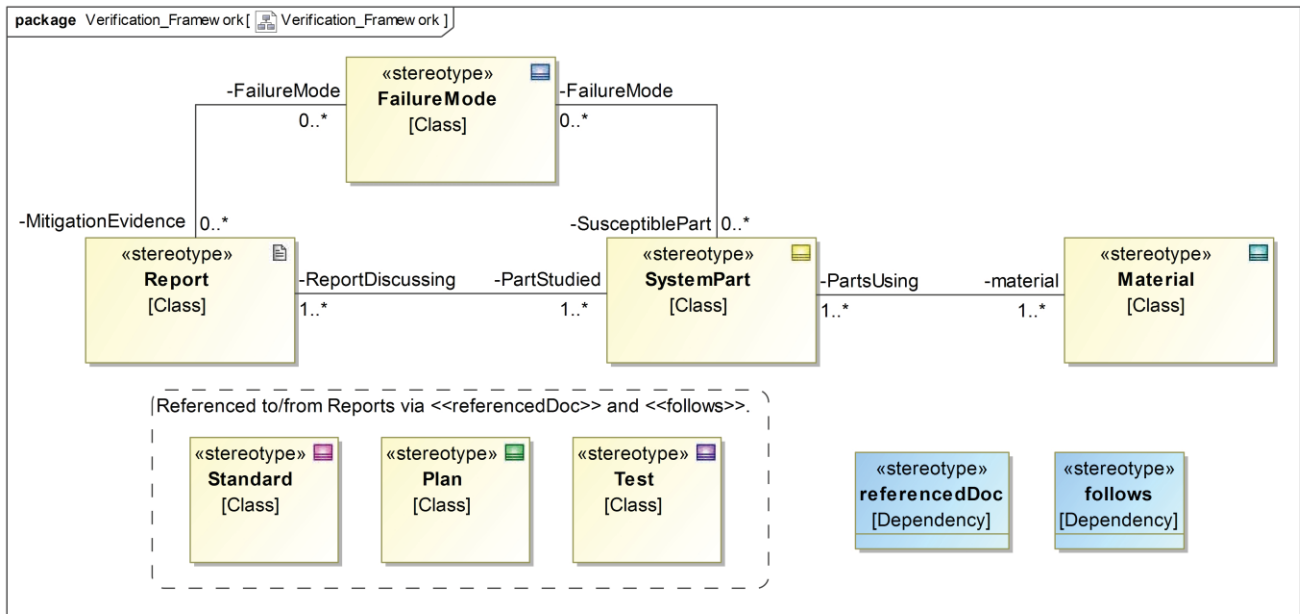
The *SystemPart* stereotype represents the component and assembly levels of the system. Properties of the *SystemPart* include the component's drawing number, the drawings revision, and a textual description of the component's function within the system.

### *Report Stereotype*

The *Report* stereotype represents an individual design document, such as an analysis, trade study, or test report. For each individual document, a variety of document metadata are recorded, including the assigned document number and its revision. The document abstract is incorporated as a property of the *Report* stereotype for later use in InVEST, enabling the independent reviewer to assess the report's relevance to his/her line of reasoning. Since a complex system may be developed by any number of supporting organizations, the team that authored the report is also noted in a property. *Reports* are categorized as describing a test or analysis, and rated according to level of completion: planned, draft, or baseline.

### *FailureMode Stereotype*

Since the primary motivation for the framework's development is effectively communicating the reliability of the system, the *FailureMode* stereotype was created to enable the reviewer to assess reliability documentation that discusses the mitigation steps taken to avoid those failures.



**Figure 1: SysML Document Traceability Framework**

### Material Stereotype

The *Material* stereotype represents the material from which a given component is manufactured. Material identification numbers and specifications are included as properties.

### SysML Associations Between Stereotypes

Associations between the four major stereotypes described above are included to provide additional sorting and filtering capabilities to the SysML framework. *SystemParts* are matched to their *Material* through an association. An association between a *SystemPart* and *FailureMode* pairs system components to the failure modes they are susceptible. Since multiple components may have the same failure mode, and a given component may have several failure modes, both ends of this association permit multiple entries. An association between a *SystemPart* and *Report* links system documentation to the components studied in the paper. A given failure mode may be discussed in a report for each susceptible component; an association between *Report* and *FailureMode* stereotypes creates these links.

### SDTF Referenced Document Stereotypes

Three additional classes are included in the SDTF to incorporate traceability of design documentation to design standards, project plans, and test plans. The *Standard* class represents design and construction standards or policies followed during the system design. These imposed standards may reflect design requirements on the system. The *Plan* stereotype represents plans implemented for project management that outline the design approach. Test events captured in a test plan or procedure are represented by the *Test* stereotype. Data produced by execution of tests

is later analyzed and captured in reports, represented by the *Report* stereotype previously discussed.

### SDTF Dependency Stereotypes

Two specializations of a dependency were created to aid the modeler in linking design documentation to other documentation, including *Reports*, *Standards*, *Plans*, and *Tests*. The *referencedDoc* dependency stereotype establishes references between two design documents, be they other *Reports*, *Plans*, or *Test* classes. The *follows* dependency stereotype links the *Standard* class to *Reports*, and the arrow is drawn in accordance with SysML convention: from the *Report* element to the *Standard* element.

### SDTF Integration with SysML Requirements

Existing SysML relationships, such as *satisfy* and *verify*, can be established between SysML requirements blocks and elements of this framework. Requirements imposed on the system may be derived from standards levied on the system design, or organizational requirements captured in project plans. Reports might capture testing or analyses performed to demonstrate requirements compliance, and this link can be established between the *Report* stereotype and the appropriate requirement block in the SysML model.

## 5. INVEST OVERVIEW

Visualizations help express the meaningfulness of data and enables data interpretation to produce information. As MBSE techniques and applications continue to grow, data visualization will become critical to understanding and navigating a complex system's model. In most cases, the visualization of a system is a requirement as there are a vast number of components in a modern system interacting with one another, which occasionally produce unexpected behavior. Simply viewing tables of data or countless SysML diagrams will not always enable the reviewer to understand the system's complexity or status of the design effort.

InVEST allows the user flexibility and control by providing multiple interactive data visualizations from which to choose. If modelers break model data into smaller segments to decompose the model into a variety of model views, new levels of comprehension and control can be attained. Enabling the visualizations to be interactive, rather than the typical, static SysML diagram, allows the user to control the content they wish to see in a format that is more relevant to their purpose.

Recent efforts have explored dynamic generation of SysML diagrams from a model by stakeholders. Principles of effective communication through SysML diagrams, such as symbol alignment or proportionality, are automated and integrated with a user interface that allows stakeholders to generate diagrams to meet their current need. [8]

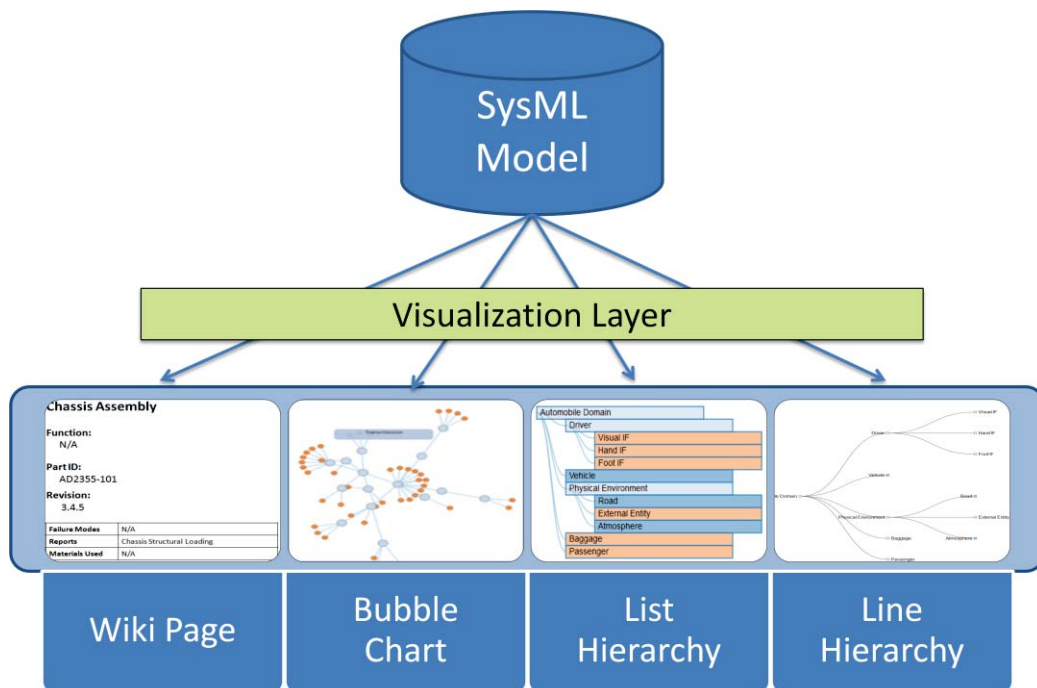
InVEST generates abstract visualizations that do not follow SysML notation, both to limit the necessary understanding of the language on the part of the user, and to summarize the current design of the system through various metrics. Stakeholder confusion with some aspects of SysML has been identified as a potential source of dissatisfaction with MBSE. [5] Abstracting slightly from SysML notations prevents the reviewer from becoming distracted by unfamiliar symbols. Statistical and numerical summaries of the system design are better communicated through data visualization techniques than SysML.

Each data visualization is generated from the system model independently of other visualizations, but remains consistent with other model views by virtue of being generated from the same model, represented in Figure 2. In InVEST's current design, data is only entered in the modeling tool and visualized in a purely read-only format.

A variety of visualizations can be produced to meet the differing needs and perspectives of stakeholders. At the highest level, summary representations of the model structure are presented in interactive, web-based images. InVEST also generates wiki pages that provide the interested reviewer with information in addition to that summarized with the interactive views. The visualizations are dashboards for exploring the system design and include hyperlinks to wiki pages with moderate detail.

### *Interactive Views*

To summarize detailed SysML model data in an environment that is much simpler than the typical modeling tool interface, InVEST exports and adapts the data to create various interactive displays. Each display is created using



**Figure 2: Visualization Overview**

web programming languages which interface with raw SysML model data. Included in these visualizations are a line hierarchy, list hierarchy, and bubble chart, which help users explore the model (system design) without requiring knowledge of a modeling tool.

The line and list hierarchies provide the same capability, but in different formats, to explore the component assembly hierarchy: assemblies are click-able nodes that expand to reveal their subassemblies and components. Color coding indicates whether a component is a leaf element or has subcomponents. These interactive images convey a sense of the system's assembly hierarchy and allow the reviewer to "drill down" on the component of interest. The bubble chart extracts a value property from components and assemblies (e.g., probability of failure or mass), and generates a "bubble" for each component and assembly that is scaled to the relative value of that property. The bubble chart allows the reviewer to understand at a glance the system's distribution of that property (i.e., which components are the most massive). For ease of navigation between visualizations, the images are hyperlinked together in a single web interface.

Users are not hindered by complicated software menus or difficult-to-learn, new modeling programs. InVEST visualizations are designed from the ground up, using D3 [9], to be intuitive, described in more detail in the next section.

#### *Wiki Pages*

In addition to generating a variety of data visualizations from the SysML model, InVEST also generates content for importing into a wiki environment. A "Wiki" is a space for informative web pages within a general scope. Wiki spaces can supply a collaborative space, enabling the project team to assist in the creation and addition of information to the Wiki remotely for later incorporation into the SysML model. This environment includes: a textual, functional description of each component and assembly provided by an expert in the field; a tabulated section summarizing component properties (e.g., probability of failure or mass); the documentation that provides the technical details and source of these values; and an optional descriptive image. Unlike the data visualizations described previously, the wikis provide an entirely editable interface.

The wiki interface is unique among the data visualizations in that it connects with a secure server to hyperlink related documents to the wiki environment. Hyperlinks to each document in the document repository are stored in the SysML model and exported to the wiki environment, where they direct the wiki user to the appropriate repository location. Documentation stored in the repository is securely maintained and access controlled, and the wiki environment is similarly restricted, as appropriate.

## **6. IMPLEMENTATION OF THE SDTF AND INVEST**

### *SysML Model Setup*

This tool was originally developed to support the Department of Energy and NASA's joint Advanced Stirling Radioisotope Generator (ASRG) Flight Project (which has since been terminated in favor of continued Stirling technology development) team and its potential space mission customers. As a power generator, the ASRG would be an element within a larger power subsystem of a spacecraft. The ASRG consists of two Stirling engines, synchronized by a controller unit. [10]

Since this tool was developed for the existing ASRG Flight Project, a large number of design documents were already created and in multiple revisions. Existing ASRG design documentation was referenced to develop the SysML model, such as system specifications, drawing trees, assembly trees, failure modes and effects analyses, and materials usage lists. ASRG design descriptions, system architectures, and design conventions captured in the documentation were closely mimicked in the top-level SysML model to maintain a sense of familiarity to the project team as they interact with the visualizations generated with InVEST.

A top-level SysML model of the ASRG component structure was a first step necessary to provide a hierarchy around which documentation could be related. Drawing trees formed the basis from which the *SystemPart* component and assembly hierarchy was established. *FailureMode* blocks were created to mirror the failure modes summarized in the system's Failure Mode Effects and Criticality Analysis. Materials usage lists provided data necessary to create the *Material* blocks. The top-level SysML model of the ASRG's Stirling engines consisted of approximately 60 *SystemParts*, 20 *FailureModes*, and 40 *Material* blocks. Subsystem specifications and exports from requirements management tools provided data for importing into the modeling tool, as necessary.

Configuration management personnel were consulted to provide a comprehensive listing of the design documentation available for the ASRG Flight Project, often referencing content prepared for system lifecycle reviews. The *Report* blocks were generated from this documentation listing. Referenced document tables within existing documentation provide the necessary data to establish the *follows* and *referencedDoc* dependencies within the model. The model of the ASRG Flight Project includes nearly 400 technical documents that summarize and analyze the system design.

### *InVEST Software Description*

At a high level, InVEST creates a pipeline from modeling software to a web interface. By linking multiple languages together, InVEST allows data to be shown in a user-friendly

environment without a license to modeling software. While free model viewing tools exist, InVEST abstracts the relevant subsets of model data into a variety of interactive visualizations that do not require prior knowledge of SysML. Multiple views with simple interactive controls are supplied to the user. InVEST may display the information of an entire model, or a single branch of a model.

InVEST utilizes JavaScript, Data Driven Documents (D3) [9], Hyper-Text Markup Language (HTML), Velocity Template Language (VTL) [11], and JavaScript Object Notation (JSON) [12] to create interactive data visualizations, wiki environments, and reports, summarized in Figure 3. D3 is a data visualization library for use with Javascript. D3 can support very simple to very complex data visualizations, and includes functionality to interact with the data visualizations it creates. By containing data in a JSON structure, the data is read by D3 using Javascript.

InVEST's process begins with extracting raw model data. This data is pulled from a model using VTL. VTL places placeholder variables within a document that will be filled with model data. InVEST does use some logical processing within VTL, but never expands in complexity beyond basic logical operators such as "If, Then" or Boolean equal. Logical processing was a necessary step to ensure links between child and parent data were upheld and represented correctly.

The VTL code places the exported data into a JSON shell. VTL syntax is placed within literal JSON commands to insert data into a JSON structure. The stereotypes of the SDTF are referenced within VTL code to extract the information stored in the SysML model. The translation from model to code is recursively performed by placing child elements (both SDTF classes and properties) within parent element's (SDTF classes) JSON structure. The VTL logic checks each element for its existing properties, determining the children section of the JSON structure.

The entirety of the fused VTL and JSON code is stored within an HTML document as a JavaScript variable to permit D3 to manipulate that data. A Javascript string variable is used to hold the JSON string.

Using the D3 library, model data is extracted from the JSON shell and manipulated using standard JavaScript and D3 commands. The JSON code stored within the string variable in Javascript is parsed using a built-in Javascript JSON parse function. The parsed JSON code can be looped through and manipulated by D3. [9] The result is an HTML document created directly from the source SysML model.

InVEST will link the report functionality from the modeling tool directly to a clean output in the form of an HTML file. HTML files can be opened by any web browser and interacted with or saved.

Once opened, a user is greeted a menu to navigate through the various InVEST environments previously described. Each view shares the same data but represents it from a different perspective, or view. With the click of a button, the user is transported to their desired view of data, where they may interact and navigate animated menus, or read in-depth descriptions and reports in a wiki environment.

#### Deploying InVEST in the Organization

Given the hundreds of existing documents prepared for the ASRG Flight Project, the population of the STDF for the ASRG is an ongoing team effort. Two SysML modelers created the basic *SystemPart* hierarchy and tables within the modeling tool were utilized to allocate *FailureModes* and *Materials* to each *SystemPart*. When known, documents were related to the appropriate *SystemPart* and *FailureMode* blocks, per the SDTF, and these properties were exported to the wiki space.

Having completed some background work to establish the wiki space and populate each component or assembly's page with available data, system experts will be provided

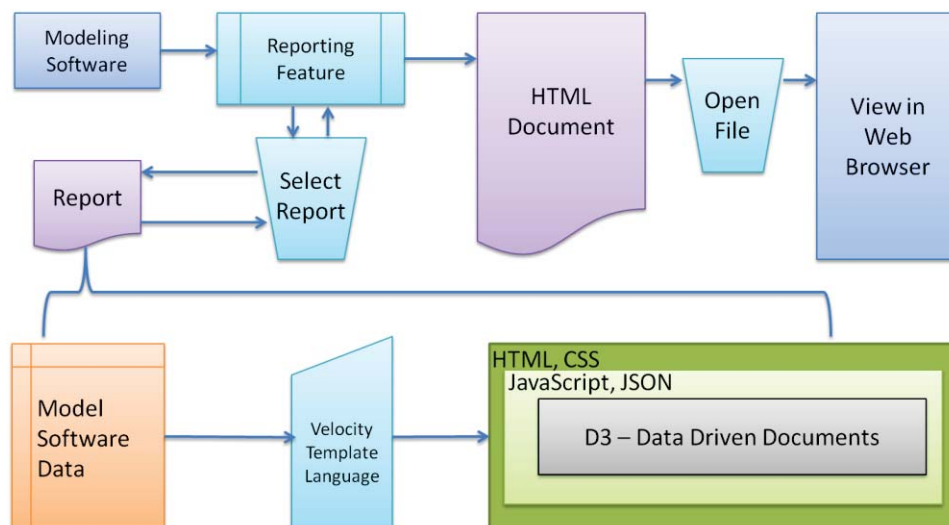


Figure 3: InVEST Data Manipulation Process Summary

access to the wiki space to collaboratively edit wiki content. The experts will help ensure the data summarized on the wiki page is complete and up-to-date, and provide additional input on the reliability parameters of relevance to this particular system. The ASRG SysML model will be updated following any revisions provided by the system experts, adapting the approach taken in [6].

#### Implementation Example

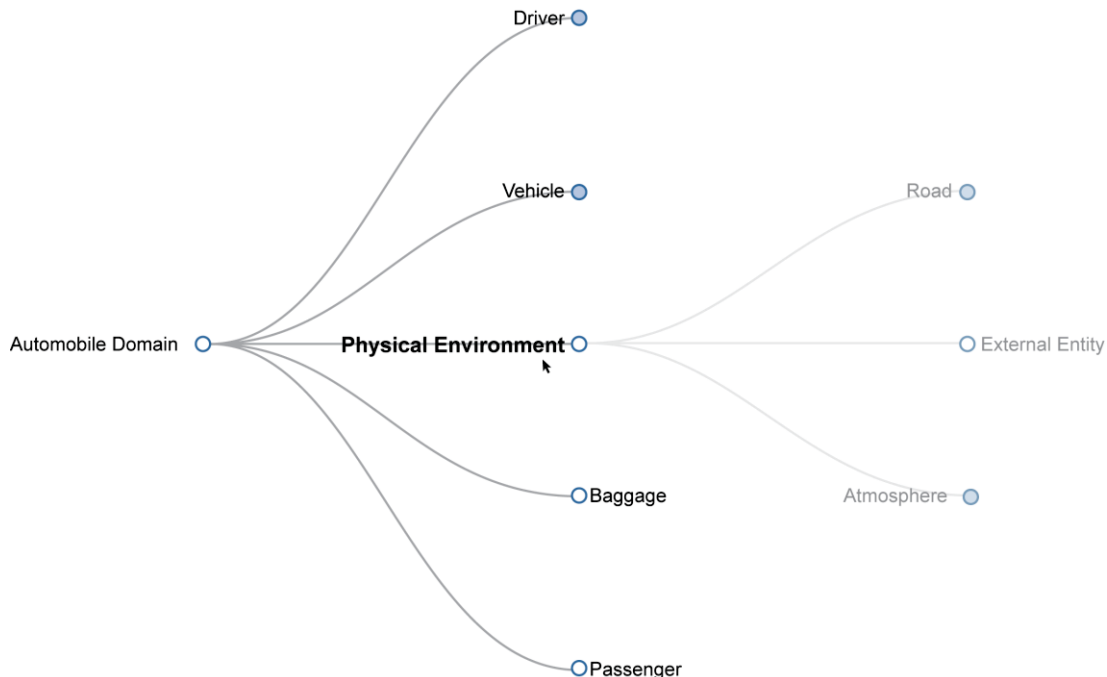
An example of a basic vehicle hierarchy, adapted from [13], using InVEST visualizations is shown in Figure 4. The Physical Environment node was expanded, revealing its subcomponents: Road, External Entity, and Atmosphere. Solid blue nodes indicate the existence of subcomponents; circles indicate leaf elements without subcomponents.

Similarly, expandable list hierarchies in Figure 5 show an alternative representation of the same hierarchical information presented in Figure 4. This alternate view provides a more consolidated summary of the system design than the line hierarchies.

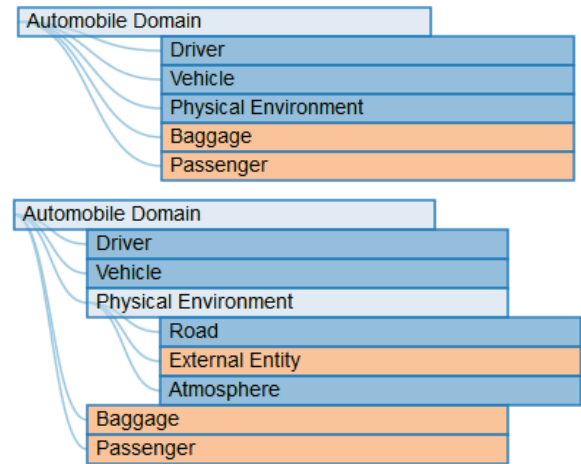
## 7. MAINTENANCE OF THE SysML MODEL AND INVEST

#### Software Maintenance

Not unlike MBSE, InVEST will change and evolve over time. Regular maintenance of InVEST will be necessary to adapt to new standards in MBSE, new capabilities of modeling software, and new needs of the end-user. Preventative maintenance measures to ensure InVEST remains error free are of the highest importance. InVEST visualizations should be regenerated after any major update



**Figure 4: Example Expansion of a Vehicle assembly using the Line Hierarchy Visualization**



**Figure 5: Example Expansion of a Vehicle Assembly using the List Hierarchy Visualization**

to the original SysML model for consistency. Given that InVEST relies on parent software libraries such as D3 and Javascript to function, any updates to these libraries may require updates to InVEST.

#### Model Maintenance

Since the SysML model of the ASRG is being developed well after it began its design lifecycle, model development will follow system development. As updates occur on the ASRG design, these changes will be reflected in the wiki environment by the system experts first, and later incorporated by modelers into the SysML model. Interactive visualizations and wiki pages will later be regenerated to fully incorporate and update existing content to reflect the design changes.

## 8. FORWARD WORK

The focus of the SDTF could be expanded to incorporate additional classes and properties, beyond the summary-level classes currently included. The SDTF was initially developed with few classes and properties for use in InVEST as a proof of concept. As characterizations, such as mass [14] or reliability, are added to the SysML model, more system parameters become available for visualizing with InVEST. Documentation can be associated with specific system properties, rather than general components (*SystemParts*) in the current SDTF, increasing the resolution of data traceability.

While InVEST serves as a functional tool and proof of concept for use with SysML modeling software, to be truly effective it must be expanded to support the majority of the end-user's needs. InVEST is a system that will benefit from integration with other collaborative modeling environments. Additional visualization modes and reporting interfaces would prove beneficial to the end-user to aid them in assessing the current state of the design. Enhancements to the level of interaction available in each visualization, such as a search capability, would better enable the user to evaluate the relevancy of the examined content to their task. Filter and sort capabilities within the web-based output of InVEST would promote model and system analytics generally unavailable within current modeling tools.

Automating the process of updating the SysML model from the edited wiki content, as described in [3], and incorporating edit functionality within the visualizations, would promote consistency between the SysML model and current state of the system design. As the gap between the SysML model and current state of the system design is minimized, the SysML model becomes closer to being the 'source of truth' promoted by MBSE advocates.

## 9. SUMMARY

The SDTF and InVEST together provide a mechanism for integrating traditional system design documentation with a SysML model and generating visualizations useful to the non-modeling community. InVEST enhances the ability of the system modeler to effectively connect stakeholders to the details of the system design without requiring users to have knowledge of the modeling tool or SysML. Interactive visualizations and collaborative Wiki pages bring a design team together, and get team members involved who may not be interested in learning MBSE software.

While there is forward work to be done with InVEST, it does provide a clear proof of concept for the advancements in the way we see and use MBSE data.

## REFERENCES

- [1] *OMG Systems Modeling Language (SysML) Specification*, Object Management Group (OMG), Version 1.3, 2 June 2012.
- [2] Nair, S.; de la Vara, J.; Sen, S., "A review of traceability research at the requirements engineering conference<sup>re@21</sup>," *Requirements Engineering Conference (RE)*, 2013 21st IEEE International, vol., no., pp.222,229, 15-19 July 2013.
- [3] Delp, C.; Lam, D.; Fosse, E.; Cin-Young Lee, "Model based document and report generation for systems engineering," *Aerospace Conference, 2013 IEEE*, vol., no., pp.1,11, 2-9 March 2013.
- [4] Jackson, M.; Delp, C.; Bindschadler, D.; Sarrel, M.; Wollaeger, R.; Lam, D., "Dynamic gate product and artifact generation from system models," *Aerospace Conference, 2011 IEEE*, vol., no., pp.1,10, 5-12 March 2011.
- [5] Bone, M.; Cloutier, R., "The Current State of Model Based Systems Engineering: Results from the OMG<sup>TM</sup> SysML Request for Information 2009," *Conference on Systems Engineering Research 2010*, 17-19 March 2010.
- [6] Logan, P.; Harvey, D.; Daniel Spencer, "Documents are an Essential Part of Model Based Systems Engineering," *International Symposium, 2012 INCOSE*, 9-12 July 2012.
- [7] Bayer, T.J.; Seung Chung; Cole, B.; Cooke, B.; Dekens, F.; Delp, C.; Gontijo, I.; Lewis, K.; Moshir, M.; Rasmussen, R.; Wagner, D., "Model Based Systems Engineering on the Europa mission concept study," *Aerospace Conference, 2012 IEEE*, vol., no., pp.1,18, 3-10 March 2012.
- [8] Sindiy, O.; Litomisky, K.; Davidoff, S.; Dekens, F., "Introduction to Information Visualization (InfoVis) techniques for Model-Based Systems Engineering," *Conference on Systems Engineering Research 2013*, 19-22 March 2013.
- [9] Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. "D<sup>3</sup> Data-Driven Documents." *Visualization and Computer Graphics, IEEE Transactions*, 17.12 (2011): 2301-2309.
- [10] National Aeronautics and Space Administration, "NASA facts: Advanced Stirling Radioisotope Generator (ASRG)," [rps.nasa.gov](http://rps.nasa.gov).
- [11] Apache Software Foundation, "Velocity Template Language (VTL)," [www.velocity.apache.org](http://www.velocity.apache.org).
- [12] JavaScript Object Notation (JSON): [www.json.org](http://www.json.org).

- [13] S. Friedenthal; A. Moore; R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. Waltham, MA: MK OMG Press, 2012.
- [14] Chung, S.; Bayer, T.; Cole, B.; Cooke, B.; Dekens, F.; Delp, C.; Lam, D., “Model-Based Systems Engineering Approach to Managing Mass Margin,” *International Workshop on Systems and Concurrent Engineering for Space Applications*, 2012, 17-19 October 2012.

## ACKNOWLEDGMENTS

The authors would like to acknowledge NASA’s Radioisotope Power Systems Program for funding this work and enabling the use of MBSE techniques and tools in an innovative environment.

## BIOGRAPHY



**Kathryn Trase** received a B.S. in Industrial and Systems Engineering from the Ohio State University in 2011. She has been with the NASA Glenn Research Center for 2 years. A System Engineer in the Radioisotope Power Systems Program, she leads the program’s implementation of MBSE. She is a member of Glenn’s MBSE Working Group and Secretary of the Cleveland-Northeast Ohio Chapter of INCOSE.



**Eric Fink** received a B.S. in Computer Science, and minor in Mathematics from Edinboro University, Pennsylvania. Eric has held internship positions as a Software Engineer for Erie Insurance, as well as a member of the Radioisotope Power Systems Program at NASA GRC. He resides as president of the Edinboro Film Series, and serves as a board representative for the NWPA Film Society.