***SysML v2 RFP WG,***

The following is a summary and follow-up actions from our 1.5 day face-to-face Working Group meeting at the Orlando OMG meeting on June 21 and 23, 2016. My thanks to all who contributed. The meeting summary is posted on the Orlando meeting page at:

<http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:orlando_june_2016_meeting_presentations>

***NOTE:*** At the end of our meeting, we agreed to proceed with the SysML v2 RFP. As a result, we changed the name of the Working Group from the System Modeling Assessment & Roadmap WG to the SysML v2 RFP WG.

The following sections are included below:

* Follow-up Actions
* Background
* Meeting Summary
* Next WG Meeting in Chicago on Sept 13, 15, 2016
* Attachment A: Orlando Meeting Feedback-2016-June 21, 23
* Attachment B: SysML v2 RFP Discussion
* Attachment C: SME Concept
* Attachment D: Evolving SysML and the System Modeling Environment to Support MBSE (Aug 2015 Insight Article)
* Attachment E: Hybrid SUV Change Scenario

Hedley will provide the web and dial up information for our next WG telecon on Wednesday, July 27, at 11:00 AM ET. Our next face-to-face meeting will be at the OMG meeting in Chicago, Illinois scheduled for Tuesday and Thursday, September 13 and 15, where the focus will be on the requirements and plans for the SysML v2 RFP.

*The updated SME Concept is included in Attachment C below. Sandy requests updates by August 1, 2016 from WG members, which will be used to establish the initial baseline SME Concept.*

**FOLLOW-UP ACTIONS**

* Sandy to post meeting summary and presentation slides to the WG Wiki
* Hedley to schedule the next WG telecon for Wednesday, July 27, at 11:00 AM ET
* Concept Leads to address the feedback from the Orlando Meeting included in the Attachment A below called *Orlando Meeting Feedback-2016-June 21, 23*
* Sandy to draft the agenda for a 2 day WG meeting (Tuesday and Thursday) the week of June 21 and 23, 2016 at the OMG meeting in Orlando (refer to section on Chicago meeting below)

Sanford Friedenthal

SysML v2 Working Group Chair

safriedenthal@gmail.com

**BACKGROUND**

This WG was chartered to develop the requirements for the SysML v2 RFP, which is intended to be the next generation System Modeling Language. The WG has focused on developing the requirements and concepts for a System Modeling Environment (SME) that encompasses the system modeling language and the tools needed to support model-based systems engineering (MBSE). The SysML v2 requirements, which will be incorporated into the SysML v2 RFP, will be derived from these requirements and concepts.

The initial high level requirements for the SME are documented in the August 2015 edition of the INCOSE Insight, which has MBSE as its theme. The article is entitled *'Evolving SysML and the System Modeling Environment to Support MBSE'* and defines 7 capabilities, 8 measures of effectiveness (moe's), and 11 driving requirements for the SME, which are included in a Attachment D below. This article was the first key milestone for SysML v2 development. Establishing the baseline the SME Concept that addresses the requirements in the article is the next key milestone. The draft SME Concept is included in Attachment C below. Our next major milestone is to produce the requirements for the draft RFP for SysML v2, which can be vetted among community. This is discussed in Attachment B below.

**MEETING SUMMARY**

The objectives and agenda for our face to face meeting on June 21, 23 in Orlando are included below:

**Meeting Objectives**

1. Review the System Modeling Environment (SME) Concept Overview, Issues, and Plans
2. Discuss SysML v2 RFP Objectives, Scope, Outline, and Plans

**Meeting Agenda**

*Tuesday, June 21, 2016*

13:00 - 13:15 Introduction - Sandy Friedenthal

13:15 - 14:30 SME Concept Overview - Sandy Friedenthal

14:30 - 15:00 Break

15:00 - 16:00 MBE Change Scenario / Demo - C Muggeo, M Pfenning, P Chadzynski

16:00 - 16:30 Modeling Formalism - Jonathan Patrick

16:30 - 17:00 Analysis Concept - Manas Bajaj

17:00 - 17:30 MBEE and an Executable SE Method - Robert Karban

*Thursday, June 23, 2016*

09:00 - 09:30 Model Management - Laura Hart

09:30 - 10:00 Model Visualization - Josh Feingold, Marc Sarrel

10:00 - 10:30 Break

10:30 - 11:00 Systems Engineering Concept Model (SECM) - John Watson, Sandy F, Marc Sarrel

11:00 - 12:00 Draft RFP Requirements for Properties and Expressions - Hans Peter deKoning

12:00 - 13:00 Lunch

13:00 - 13:30 Model Construction - Ron Williamson

13:30 - 14:30 Service Requirements and the Standard API - Chas Galey

14:30 - 15:00 Break

15:00 - 15:30 SysML v2 RFP Considerations from Reston - Sandy/All

15:30 - 17:00 SysML v2 RFP Objectives, Scope, and Outline - Sandy/All

16:30 - 17:00 Next Steps - All

Each Concept Lead is responsible for a particular SME capability such as model construction, model visualization, model analysis, model management, model interoperability, and was requested to provide updates to their concept, and their open issues and plans, including the items below as it relates to their concept:

* Response to actions from the Reston WG meeting
* Effectiveness measures and driving requirements
* Limitations of current SysML
* Key features and graphic of their concept (refer to Visualization graphic)
* SME service requirements (e.g. functions) needed to support their concept
* Illustration of how their concepts supports the Hybrid SUV scenario (refer to Manas example)
* Plans for prototypes to demonstrate feasibility
* Wiki status

Some of the presentations are included on the [Orlando meeting Wiki page](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:orlando_june_2016_meeting_presentations). Other Concept Leads presented directly from their Wiki pages which can be found on links under the Working Groups section of the [System Modeling Assessment & Roadmap WG Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:sysml_assessment_and_roadmap_working_group).

The Concept Leads are requested to address the feedback from this meeting which is captured in Attachment A. They are also requested to review the SME concept in Attachment C, provide any feedback to Sandy, refine their concepts, and begin to derive requirements for the SysML v2 RFP needed to support their concepts. This includes update to the service requirements that reflect the required functionality to support the Hybrid SUV change scenarios (Attachment E). Where practical, the are asked to identify and provide actual prototypes of potential solutions to demonstrate feasibility of their concept. Also, Concept Leads should continue to maintain their concept on their Wiki page to provide visibility for broader review and vetting.

We also discussed the RFP objectives and approach (e.g., scope, outline, and plan) which is summarized in Attachment B.

**NEXT WORKING GROUP MEETING IN CHICAGO on September 13, 15, 2016**

The following are tentative topics for the SysML v2 RFP Working Group meeting in Chicago, Illinois on Tuesday and Thursday (September 13 and 15, 2016):

1. SME Concept Update

2. Service Requirements Review

3. Formalism Requirements

4. Systems Engineering Concept (SECM) Review

4. RFP Development Plan

* Outline
* Organizing for the RFP Development
* Milestones

5. Other (TBD)

**Attachment A**

**Orlando Meeting Feedback-2016-June 21,23**

**All Concept Leads are requested to:**

* Ensure feedback from Reston meeting on March 15, 17, 2016 is addressed. This feedback is included in the Meeting Summary on the [Reston Meeting Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_mar_2016_meeting_presentations).
* Review SME concept in Attachment C including the SME key features section, align Wiki content, and provide updates to Sandy by August 1
* Define service requirements based on elaborated Hybrid SUV scenario and coordinate with Axel to capture integrated service requirements on his Wiki (you copy/reference)
* Maintain Wiki page for SME concept
  + Effectiveness measures and driving requirements
  + Limitations of current SysML
  + Key features of the concept
  + Service requirements (e.g. functions)
  + Illustration of how their concepts supports the Hybrid SUV change scenarios
  + Plans for prototypes to demonstrate feasibility
* Present proposed draft requirements for RFP at September meeting, along with a graphic of your concept (refer to Visualization concept as an example which will be posted to visualization Wiki)

**In addition to the above, the following are specific actions for each Concept Lead:**

**Model Formalism Concept:** Jonathan Patrick

* Establish a concise definition for model formalism
* Identify, review, and compare the pros and cons of alternative formalisms such as MOF, OWL, IDEAS, Algebra of Relationships, and Relational formalisms as the potential formal foundations for SysML.
* Clarify how constraint languages such as OCL relate to the formalism.
* Define formalism use cases and requirements for both executing engineering analysis models and support for logical inferencing (coordinate with Manas on requirements and use cases for executing engineering analysis)
* Ensure the formalism is balanced with usability considerations to ensure the formalism does not overly restrict the user to accomplish their task.

**Model Analysis Concept:** Manas Bajaj

* Integrate considerations from Property and Expressions Concept Model
* Coordinate with Formalism WG (Jonathan) to ensure formalism use cases and requirements support engineering analysis and execution

**Model Construction Concept:** Ron Williamson

* Coordinate with visualization team (Chris, Josh, Marc) to understand the implications of the interactive nature between model construction and visualization on usability and the associated requirements on SysML v2

**Model Visualization Concept:** Chris Schreiber

* Post updated visualization concept graphic to Wiki
* Update Model View Controller description and ensure consistency with current SysML and ISO 42010 definitions of View and Viewpoint
* Update concept to highlight the narrative view which combines text, tabular and graphic representations
* Coordinate with Ron (Model Construction) to understand the implications of the interactive nature between model construction and visualization on usability and the associated requirements on SysML v2

**Model Management Concept:** Laura Hart

* Update the model and demo of the Change Management Scenario presented by Christian Muggeo, Michael Pfenning and Pawel Chadzynski and post to Model Management Wiki

**MBSE Collaboration & Workflow Concept:** Hedley Apperly

* Not presented

**Model Interoperability and Standardized API:** Axel Reichwein

* Update template for specifying service requirements
* Coordinate with other Concept Leads to capture integrated service requirements based on elaborated Hybrid SUV scenario
* Coordinate with other vendors on the team to define the general requirements for the standard API
* Consider limitations of OSLC and the need for event services (per Chris Delp)
* Confirm that the proposed exchange format is an XML serialization of RDF
* Clarify the concept for migrating SysML v1 model to SysML v2 models

**Systems Engineering Concept Model (SECM):** John Watson

* Update the SECM needs for Requirements and Verification
* Coordinate with other SECM WG's (property & expression, interfaces, ...) to begin integration of SECM-RFP Model and the RFP requirements

**Systems Engineering Concept Model (SECM) - Property & Expressions Concept:** Hans Peter de Koning

* Review and update the model and requirements with the other members of the team
* Capture rationale as part of requirements
* Add support for probabilities
* Determine what is the appropriate boundary between requirements and reference implementation
* Evaluate OMG FIBO per Matthew Hause for potential applicability

**Systems Engineering Concept Model (SECM) - Interface Concept:** Marc Sarrel

* Present draft requirements at the September OMG meeting

**Attachment B**

**SysML v2 RFP DISCUSSION**

During the previous WG meeting in March in Reston, we began discussion on the approach to the SysML v2 RFP. We continued the discussion on the approach during the last part of the Orlando meeting, and refined the approach as follows:.

1. The RFP should not constrain the implementation to a profile of UML.

2. The RFP will require the following elements to be included in the SysML specification to support the SME capabilities:

* SysML Metamodel
* SysML Profile
* Model Libraries (e.g., QUDV)
* Concrete Syntax
* Standard API
* Reference Model (used for demonstrating conformance level)
* Conformance Test Cases
* User Interface Guidelines (e.g., criteria, prototypes, screen mockups,)

3. The initial minimum scope for the RFP will include the metamodel and profile that support the user concepts that are addressed by the current SysML specification, along with the standard API.

4. The RFP process will encourage prototypes and reference implementations to validate the requirements for the SysML v2 Specification. Some examples that have been demonstrated to this working group include:

* [Open MBEE](https://github.com/Open-MBEE) - Chris Delp, Robert Karban (TMT executable model)
* PLM/MBSE Integration Scenario - Christian Muggeo, Michael Pfenning, Pawel Chadzynski
* SysML with STK integration - Nerijus Jankevicius
* Adaptation of Tom Sawyer Visualization - Josh Feingold
* [***SysML Extension for Physical Interaction and Signal Flow Simulation RFC***](http://www.omg.org/cgi-bin/doc?mantis/2016-03-10) - Conrad Bock

**SysML v2 RFP Plan:** A preliminary proposed plan for the RFP development is as follows:

Aug 2015: Driving Requirements (INCOSE MBSE Themed Insight Article)

June 2016: System Modeling Environment Concept (Draft)

June 2016: RFP Objectives, Scope, and Outline (Draft)

Sept 2016: Integrated Service Requirements (Draft)

Sept 2016: Systems Engineering Concept Model (Properties & Expressions, I/F’s)

Dec 2016: Systems Engineering Concept Model (Requirements & Verification)

Dec 2016: Initial Presentation to ADTF

Jan 2017: Presentation for INCOSE IW

Mar 2017: Systems Engineering Concept Model (Integrated Behavior & Structure)

June 2017: Draft RFP and Presentation to ADTF

Sept 2017: Reviews Complete

Sept 2017: Issue RFP (ADTV vote, AB vote, start TC vote)

We reviewed the following RFP objectives to serve as the introduction to the SysML v2 RFP, along with the preliminary outline for the mandatory requirements in section 6.5 of the RFP:

**Objective of this RFP**

<This RFP specifies the requirements for the next generation of the OMG Systems Modeling Language (OMG SysML™ v2) that are intended to address many of the limitations of the current version of OMG SysML™ v1.5. The goal for SysML v2 is to specify a **standard modeling language** and **standard API,** that is to be implemented by a system modeling environment, to enable the effective application of model-based systems engineering (MBSE). SysML v2 will also include standard extension mechanisms to further extend and adapt the modeling language and API to meet the unique needs of different domains.

The modeling language expresses the core concepts required to precisely specify a system and its elements (i.e., the system model). The system model provides shared views of the system that can be realized by detailed design models which span many different engineering disciplines, such as electrical, mechanical, and software design. The modeling language includes a semantic foundation to support integrity checking of the system model, and to integrate with standard-based solvers to support the execution and analysis of the system model. The modeling language provides a graphical and textual syntax to enable the system model to be understood and used by diverse stakeholders.

This RFP requires the modeling language to be specified as a SysML metamodel that is not constrained by UML, and a SysML profile of UML. In addition, this RFP requires selected model libraries, such as those needed to represent systems of units and quantity kinds (e.g., QUDV).

The RFP requires an API that provides a standard set of service requests to access and operate on the system model. The service requests are implemented by the system modeling environment to support model construction, model visualization, model analysis, model management, model interoperability, and workflow and collaboration. The API is specified using open standards such as web-based interoperability standards.

This RFP also requires the SysML v2 specification to include additional guidelines to support tool implementations. This includes test cases and a reference model to enable vendors and users to demonstrate and evaluate conformance with the SysML v2 specification, and user interface guidelines to promote usability of the modeling tools. >

This RFP solicits proposals that include the following:

* <SysML metamodel>
* <SysML profile>
* <Selected model libraries>
* <SysML concrete syntax>
* <SysML API>
* <SysML conformance test cases>
* <SysML reference model>
* <SysML user guidelines>

For further details see Section 6 of this document.

**SysML v2 RFP and Specification Development.** As noted above, the System Modeling Environment (SME) is intended to provide the modeling capabilities to enable model-based systems engineering (MBSE), and will be used to help derive requirements for the SysML v2 RFP. The RFP requirements will result in a SysML v2 specification that is implemented by SysML vendors. The following figure highlights the approach to develop SysML v2 vendor implementations as part of a SME that can enable MBSE.

**Figure 1.** SysML v2 Specification Development Approach

The MBSE Use Cases are used to identify capabilities required of the System Modeling Environment (SME). The service requirements and systems engineering concept model (SECM) needed to support the SME capabilities and MBSE Use Cases are used to specify the requirements in the SysML v2 RFP. The SysML v2 Specification will be developed by one or more submission teams in response to these requirements. The submission teams submit their solution via the OMG specification adoption process. The selected and approved SysML v2 Specification is then implemented by tool vendors to enable the System Modeling Environment. Multiple vendors and multiple vendor products may be required to provide full support for all SysML v2 services, but they will use the standard API and data model.

Test cases and a reference model are used to demonstrate the level of conformance of the SysML v2 Implementation to the specification as indicated in the figure below. The reference model is imported into the modeling environment. Services are requested of the environment via the standard SysML API. The expected response to the service is compared with the actual response to determine conformance with the specification. An example test case may be to generate a view of a requirements traceability matrix.



**Figure 2.** Demonstrating conformance level to the SysML v2 specification

**Attachment C**

**System Modeling Environment (SME) Concept**

**Introduction.** The following describes the preliminary System Modeling Environment (SME) Concept which will be used to derive the requirements for the SysML v2 RFP.

In Figure 3 below, each of the disciplines contributes to the development of the technical baseline of the system as part of a Model-Based Engineering (MBE) approach. The MBE Environment is the overall set of tools that all disciplines use to implement the MBE approach. The System Modeling Environment (SME) is the systems view of the MBE Environment. The SME is used by the systems engineers and others to perform MBSE, and evolve the system model in the broader context of MBE. The SME also enables other members of the development team to interact with the systems engineer.

**Figure 3**. The System Modeling Environment provides a systems perspective of the broader Model-Based Engineering (MBE) environment that enables systems engineers to perform MBSE

The system model provides an overall description of the system that helps integrate the models and associated design information from the other engineering disciplines as indicated in Figure 4.

Figure 4. System Model Facilitates Integration of Engineering Data (Source: John Watson)

The systems engineer performs the MBSE process as part of an incremental 'Vee' development process to flow down requirements from the mission/enterprise level to systems, subsystems, and components, and verify the components, subsystems, system and mission requirements are satisfied. This continues throughout the development process. Some example SE use cases are described on the [SE Use Cases Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:use_case_working_group).

An important part of the engineering process is the ability to effectively manage change. In fact, the entire development process can be viewed as an incremental change process. The MBE change scenario involves the entire development team to support problem identification and assessment, change impact assessment, and change implementation. The change process often varies with the phase of the life cycle. For example, a change process is different in the conceptual design phase, than it is in the preliminary and detailed design phase, integration and test phase, and the post deployment phase. Figure 5 shows a high level view of the interactions among the different tools used by different discipline engineers to support the change process.

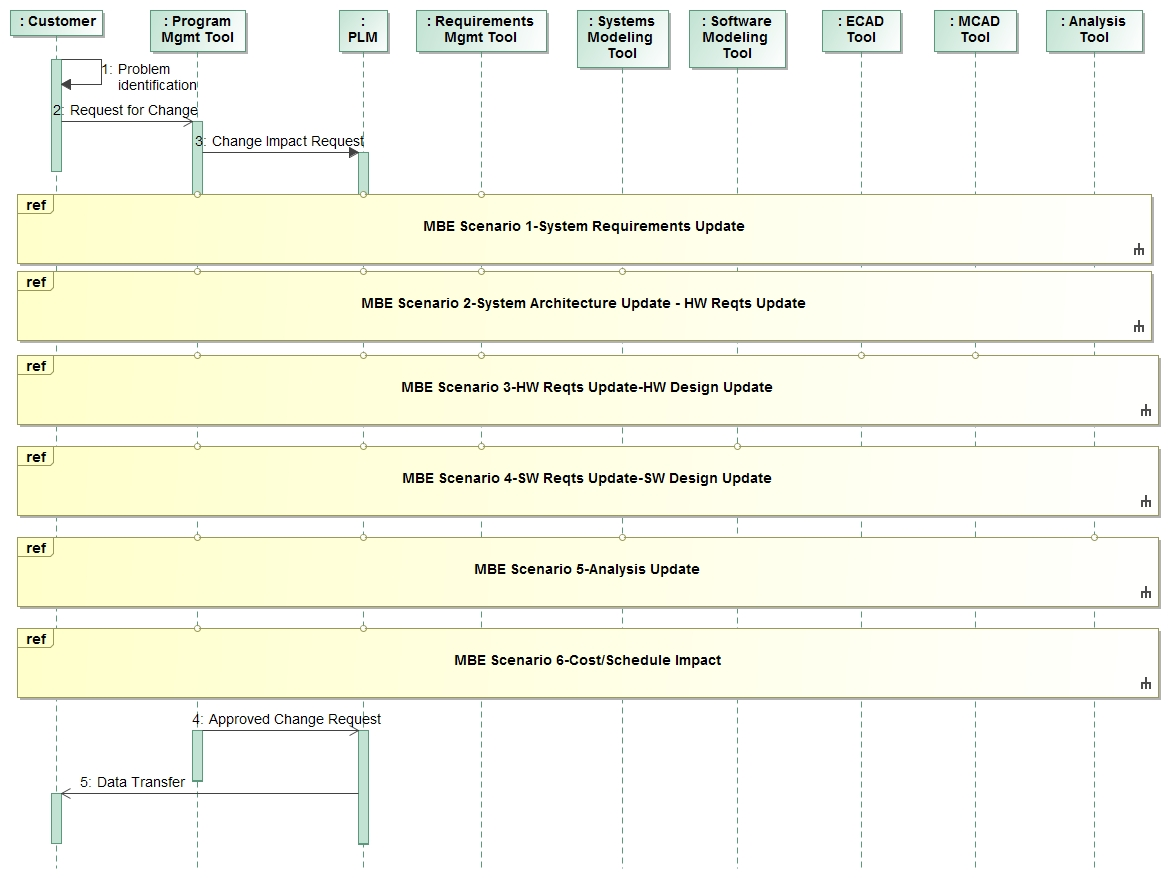


Figure 5. A MBE Change Scenario requires interactions among multiple disciplines

A representative preliminary MBE Change Scenario in three different phases of the life cycle of a Hybrid SUV is highlighted in Appendix B to help derive the functional requirements that the System Modeling Environment (SME) must support.

**SME capabilities.** The initial SME capabilities and driving requirements were originally defined in the August 2015 INCOSE MBSE Themed Issue in an article entitled *'Evolving SysML and the System Modeling Environment to Support MBSE'*. The article defines 7 capabilities, 8 measures of effectiveness (moe's), and 11 driving requirements for the SME, which are included in Attachment D.

The SME provides the following capabilities:

* model construction
* model visualization
* model analysis
* model management
* model exchange and integration
* support for MBSE collaboration and workflow
* extension/customization support

**SME Architecture.** The SME must provide the functionality to evolve the system model throughout the life cycle in support of MBSE.

The SME includes the following elements:

* SysML modeling language and tools
* Re-use libraries (e.g., models, practices, ..)
* Integrations with other engineering models and tools
* Extension and customization facilities

Figure 6 provides a view of the logical architecture of the SME. The model repository contains the data about the system, including the system model, analysis data, and other metadata. The figure shows the systems engineer interacting with the SME to define and evolve the system model in the model repository. The systems engineer and other disciplines can interact with the SME through a model editor that provides the full functionality of the SME (i.e., rich interface), or through a lighter weight interface (i.e., web interface) that supports a specific viewpoint. Other consumers and reviewers of the system model typically use the web interface with domain specific views to review and update the model. Additional functionality is provided by other logical elements of the SME to support exchange, management, and analysis of the model data, and task management that facilitates the Systems Engineer in implementing the SE practices defined in the practices repository.

The external interface and management element enables direct interaction between the other engineering tools and models and the system model. The external interface also provides global task management (e.g., workflow) to support collaboration with other engineers, which includes task synchronization through notifications as the engineering work products change state. This capability is facilitated by the system and product life cycle management environments that are also part of the MBE Environment.

In addition, the figure identifies the development environment used to further extend and customize the environment to support different viewpoints and program needs. This enables domain specific extensions of the environment to address the broad set of uses of the SME. The SME must provide standard extension mechanisms to maintain interoperability of the environment.

Finally, there is a practices repository that stores a set of practices that are used by the Systems Engineer to support MBSE. The local task manager facilitates the implementation of these practices.



**Figure 6. System Modeling Environment-Logical Architecture (7/10/16)**

Figure 7 shows the layers of the SME architecture using a fairly traditional layered architecture. The platform provides the basic hardware and operating system infrastructure. The data layer stores the model data in the repository. The services layer provides a set of services that support the SME capabilities including model construction, visualization, analysis, and management. A standardized application program interface (API) is used to access these services. The graphical user interface provides the interface for the users, and one or more adapters enable other tools to access the API to request the services. Finally, there is an extension layer that enables the development environment to modify and extend the SME data model, services, and interfaces.

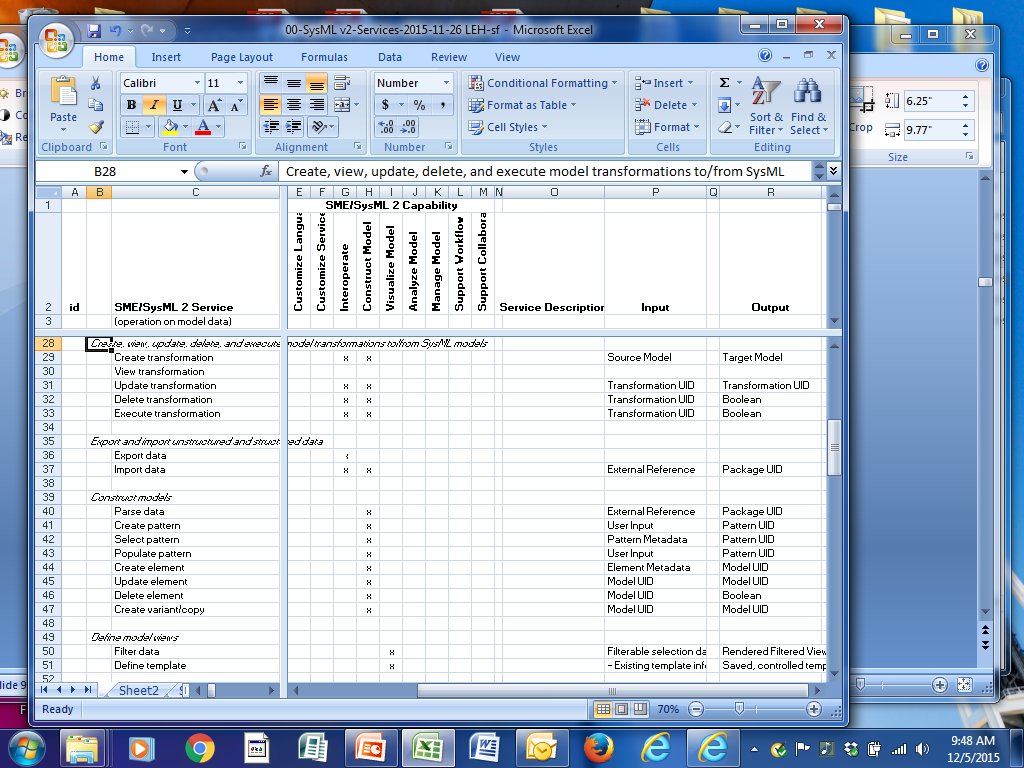


**Figure 7. System Modeling Environment Layered Architecture**

The categories of services to support the SME capabilities include:

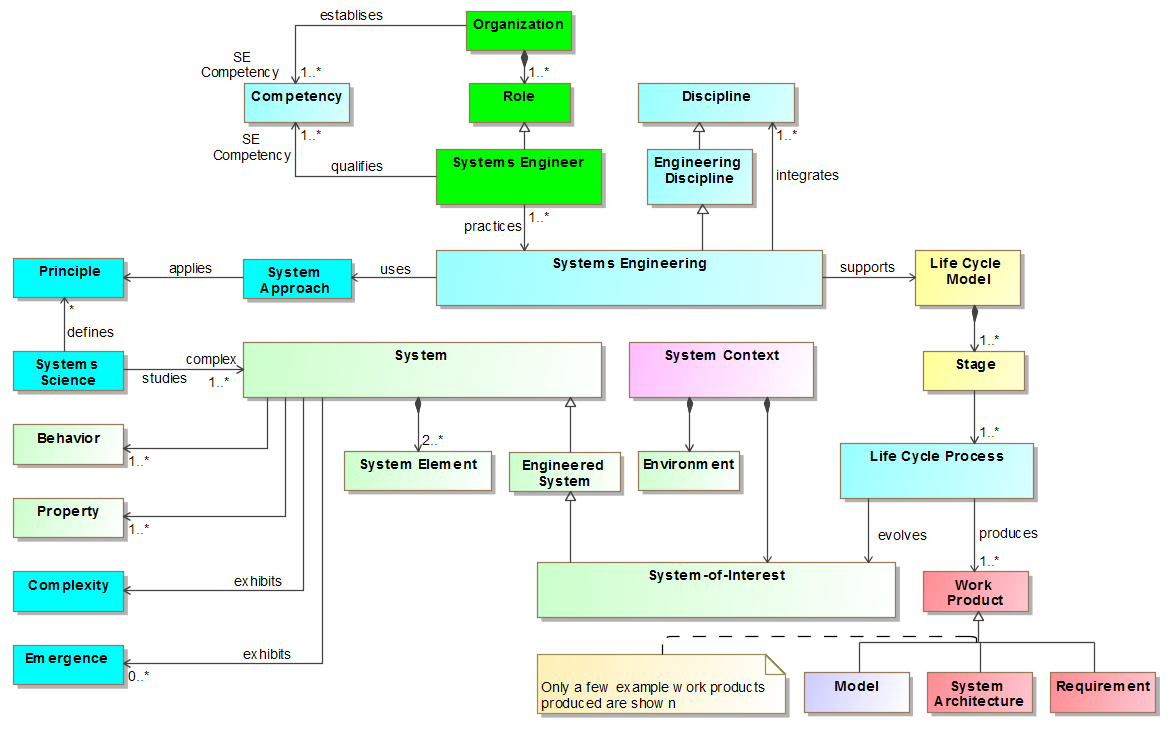
* create, read, modify, and delete links between tools
* create, read, update, delete, and execute transformations to/from SysML models
* export and import unstructured and structured data
* create, update, delete, and execute model elements and patterns
* create, read, update, and delete viewpoints
* create, read, update, delete, and execute model queries to support visualization and analysis
* generate and manage views
* create, read, update, delete, and execute model validation rules to validate input data & model
* create, read, update, delete, and execute analytical models
* manage model metadata
* manage changes to model
* create, read, update, delete, and execute workflows and notifications

A partial list of services is shown in the spreadsheet in FIgure 8 along with their inputs and outputs. This functionality will be used to help specify the API.

****

**Figure 8. Service Requirements (Extract from initial draft)**

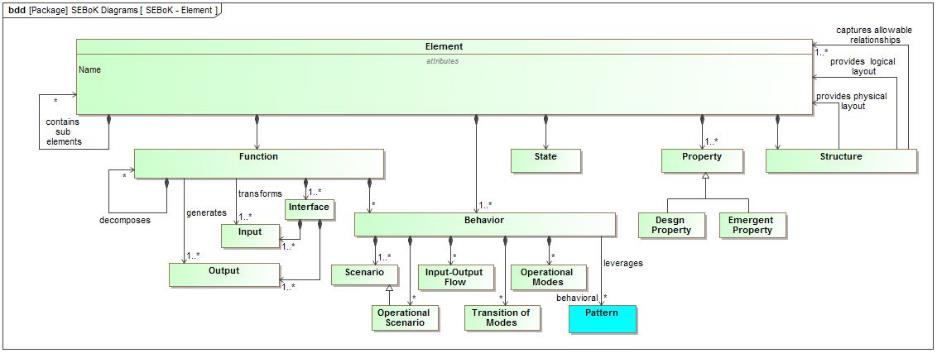
The Systems Engineering Concept Model (SECM) will be a primary input to the requirements for the SysML v2 metamodel, profile, and model libraries. The concepts in the SECM are intended to reflect industry standards for systems engineering including the Systems Engineering Body of Knowledge (SEBoK), the ISO standard for Systems and software engineering -- System life cycle processes (ISO/IEC/IEEE 15288:2015), and the INCOSE Systems Engineering Handbook v4. These sources generally provide the basis for defining the high level concepts which can then be further elaborated to support the definition of the SysML v2 language concepts. An extract from a preliminary version of the SEBoK is shown in Figure 9.

****

**Figure 9. Top Level Systems Engineering Concepts in Systems Engineering Concept Model (Extract from recent draft of SECM-2015 Industry Reference)**

There are many additional concepts in the industry reference model. The intent is to align SysML v2 with these concepts. However, the scope of SysML v2 is not intended to address the full scope of the industry reference model, and may include concepts that are not explicitly included in the reference model. The initial focus for SysML v2 will be on the System concepts, and may not include other concepts such as those related to organization.

The industry reference model is not sufficiently detailed to fully specify SysML v2 requirements. For example, there is the concept of a system property in the industry reference model as shown in Figure 10 below. However, there is no explicit description of data type, units, and quantity kinds, that must be supported by SysML v2. The high level concepts from the industry reference model will be elaborated as part of the SysML v2 requirements development effort.

****

**Figure 10. Element Concept in Systems Engineering Concept Model (Extract from recent draft of SECM-2015 Industry Reference)**

**SME Key Features:** The following are key features of the System Modeling Environment (SME) Concept. Refer to the Working Group Wiki's on the [System Modeling Assessment & Roadmap Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:sysml_assessment_and_roadmap_working_group#working_groups) for additional detail.

**General capability.** The SME enables systems engineers to perform model-based systems engineering activities, and enables others to provide inputs to or consume outputs from these activities. . The SME includes a federated data repository and applications to enable systems engineering concepts to be expressed, visualized, analyzed, managed, exchanged, and documented in support of MBSE collaboration and workflows. The environment is highly extensible to enable domain specific applications, including extensions to the language concepts, provided services, and user interface.

**SME capabilities.** The SME provides the following capabilities:

* model construction
* model visualization
* model analysis
* model management
* model exchange and integration
* support for MBSE collaboration and workflow
* extension/customization support

**SME scope.** The SME scope includes the following elements:

* SysML modeling language and tools
* Re-use libraries (e.g., models, practices, ..)
* Integrations with other engineering models and tools
* Extension and customization facilities

**SME open environment.** The SME will leverage standards from the OMG and other standards bodies where practical to provide an open environment to support interoperability and extensibility. Possible standards include:

* OMG standards - MOF, MOF Versioning, Diagram Definition, Diagram Interchange. Model to text
* Web standards (RDF, REST, SPARQL, OSLC,..)
* Other standards (FMI, ...)

**SME functionality.** The SME functionality supports the SME capabilities noted above.

* The functionality is implemented as services exposed by the standard API and/or the GUI
* Each unit of SME functionality supports the create, read, update, delete, and execution of the model, along with defined change notifications and processing when the model is updated.

**SME usability.** The SME will provide a user interface that is intuitive and efficient.

* The SME adapts its user interface to accommodate the needs of different classes of users including different levels of modeling expertise (novice to experienced), different domains, different lifecycle phases, and different levels of rigor. The SME must provide an agile modeling environment that supports the modeling need with the appropriate level of rigor for the particular class of user. In some case, the user is primarily a reviewer and/or consumer of the modeling information, where as in other cases, the user creates and/or updates the model. The SME provides user access to the model data via a simplified web-based interface that supports domain specific viewpoints, and a rich modeling interface with full functionality to create, view, modify, delete, and execute models.
* User range include novice vs. experienced, consumer vs. producer, and conceptual vs. detail design
* Role based adaptation of the user interface to accommodate different modeling roles and needs
* Simplified web based interface using tables, text, and links
* Continuous feedback is provided to the user on their model development, along with embedded help.
* Textual, tabular, graphical inputs and outputs (combination is called a narrative view)
* Support for evolving technologies such as touch screen, mobile devices, etc.
* The SysML v2 specification includes guidelines to ensure the user experience is both efficient and intuitive.

**SME quality characteristics.** The SME will provide the following quality characteristics

* Performance and scalability
* Availability
* Data protection that includes data classification and access controls (refer to model model management concept below)

**System Model.** The system model is a model of the system and its environment, and is intended to capture the essential data to support the specification of the system. The level of detail is adapted to the application, but a primary purpose is to specify the system elements, and support the integration of multi-disciplinary design data. The system model should express the concepts that are defined in the Systems Engineering Concept Model (SECM) Refer to [SECM Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_concept_model_workgroup) for details.

**Language formalism.** The language formalism provides the precise semantic foundation to provide unambiguous expression of the key concepts (refer to [Formalism WG Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:sysml_v2_model_formalism_working_group)).

* Semantic/ontological foundation which is precise enough (i.e. computer interpretable) to support deterministic results for interpretation, transformation (to model, to text), model query, logical inferences, and model checking
* Includes representation of time related properties
* Support variable formalization and enforcement levels (e.g., warnings, errors)
* Supports interoperability with other modeling languages
* Includes a standard extension mechanism

**Model analysis.** The SME will provide the capability tosupport model execution and validation, and seamlessly integrate with diverse engineering analysis models (refer to [Analysis Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:system_analysis_workgroup)).

* Ability to specify analyses including input/output parameters and their relationships, analysis objectives, assumptions, fidelity, and results
* Model transformations between design and analysis models
* Ability to represent properties and expressions to support quantitative analysis
* Intuitive equation generation and solving including basic math operators and built in support for standard expression and simulation languages such as Modelica, and standard scripting languages such as Python
* Enhanced system of quantity and units libraries with support for dimensional analysis
* Ability to support simple geometric shapes, coordinate systems, coordinate transformations, and other geometric analysis
* Visualization of quantitative analysis (e.g., plots, tables, ..)
* Management of analysis inputs and results
* Support with interoperability with analytical models that leverage standards such as FMI

**Model construction.** The SME will provide the capability to enable intuitive and efficient model construction (refer to [Model Construction Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_model_construction_focus_area) and the usability section concept above and collaboration and workflow concept below).

* User input includes a combination of:
  + Create new model from scratch
  + Reuse existing model
  + Linking data to an external data source
  + Import from external data source (structured and **unstructured data**)
* User can construct model using text, tabular, or graphical input
* Support for batch operations to create, update, and delete model information
* Support for creating and leveraging reuse libraries
* Support for defining and applying modeling patterns
* Support for domain specific viewpoints
* Support for process implementation (e.g., role adaptation to execute practices which apply patterns, profiles, and libraries using domain specific interface)
* Context sensitive user help facilities
* Controlled, role based access to the model via Model Management capabilities
* Model change management supporting new model versions and variants via Model Management capabilities
* Support for collaborative, team-oriented model construction

**Model visualization.** The SME will provide the capability to support flexible and rich visualization and reporting capabilities for a broad range of model users (refer to [Model Visualization Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:model_visualization_working_group) and usability concept above).

* Graphical, tabular, and textual views of the model (narrative view may include combination of all)
* Standard symbology (i.e., concrete syntax) with defined graphical and layout styles for the standard diagram types and tables
* Highly flexible viewpoint specification and view generation including model construction views, domain specific views, version and configuration management views with domain specific graphical symbols and layout styles
* Automated view generation (rule-based)
* Interactive view generation including semantic filter, zoom, and pan capability
* Document generation
* Diagram definition standard with extensions to be considered

**Model Interoperability and Standardized API.** The SME willprovide the capability tointegrate with discipline-specific engineering tools, including hardware and software design, analysis and simulation, and verification (refer to [Model Interoperability Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_interoperability_working_group)).

* Availability of data
  + in a standardized graph-based data format
  + through a standard API
  + through a standard Web API
  + through a standard Query API to perform complex queries
* Adoption of data format and data access standards aligned with W3C standards
* Adoption of data format and data access standards aligned with the larger web community
* Merging/linking different systems engineering standards (STEP/OMG/ISO/OSLC) through Linked Data
* Recommended standards
  + **Data** of the System Modeling Environment (SME) should be available in the W3C standard [**Resource Description Framework (RDF)**](http://www.w3.org/RDF/Resource%20Description%20Framework%20(RDF))
  + **Metadata** (semantics) in the System Modeling Environment (SME) should be defined in RDF vocabularies (W3C standard [**RDF Schema**](http://www.w3.org/TR/rdf-schema/)) and shape constraints (W3C standard [**SHACL**](http://www.w3.org/TR/shacl/))
  + **Web API** of the System Modeling Environment (SME) should be RESTful and conform to the W3C standard [**Linked Data Platform**](http://www.w3.org/TR/rdf-schema/)
  + **Query API** of the System Modeling Environment (SME) should conform to the W3C standards [**SPARQL**](http://www.w3.org/TR/sparql11-query/) and [**SPARQL Protocol**](http://www.w3.org/TR/sparql11-protocol/)
  + **Exchange format** is an XML serialization of RDF
  + **Note:** include reference to secure access (e.g. https)
* Other standards to support model execution and analysis such as FMI
* Standardized formats for persistent data storage
* Migration between SysML versions using standards and transformations

**Model management.** The SME will provide the capability to manage system models as part of a heterogeneous and distributed modeling environment (refer to [Model Lifecycle Management Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:model_lifecycle_management_working_group)).

* Standardized metadata for capturing change, variant, and configuration information of the system model and related artifacts
* Integration with product and system life cycle management environment (e.g., an abstract bill of material structure that can be used to relate system specification data with detailed design and implementation data)
* Unique id for all model elements
* Version control to the model element level
* Identification of model configuration items (CI)
* Model diff from CI to element level
* Model branch and merge capabilities
* Efficient capture and logging of incremental changes
* Visualization of model change history
* Highly granular role based access controls
* Support for data protection (i.e., classification)
* Dependency management (e.g., visibility as to what elements depend on another)
* Broad support for model transformation
* Automated transformation between SysML v1 and SysML v2 and later versions

**MBSE Collaboration & Workflow.** The SME will provide the capability to execute workflows in support of MBSE, and enable collaboration with the development stakeholders (refer to [Workflow and Collaboration Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:mbse_workflow_and_collaboration_working_group)).

* As s systems engineer constructs a model, he or she follows a particular workflow that supports a particular practice from the practice repository. A  specific practice from the repository is implemented by executing role based adaptations that apply domain specific user interfaces, profiles, patterns, and libraries, with highly granular role based access to the data (e.g. clear data ownership). Others are notified of the change in state of the model and the workflow status.. Metrics are collected to support project and technical managmenent.
* Executable workflows to execute processes which bundles relevant input data
* Change notification and task synchronization with other workflows
  + Includes integration via PLM with MCAD, ECAD, Software Dev, Simulation
* Integration with practices repository
* Support for metrics collection

**Extensibility facilities.** The SME will provide the capability to enable domain specific extensions to the modeling environment that may include extensions to concepts, services, and user interfaces. Extension capabilities include:

* Standard extension mechanism for language concepts with support for aliases
* Flexible viewpoint extensions to support domains specific interfaces and view generation
* Extensible workflows that automate the execution of practices with patterns, profiles, and libraries
* Extensible validation rules
* Development of domain specific plug-ins to support the above

**Attachment D**

**EXTRACT FROM AUGUST 2015 INCOSE INSIGHT**

**Evolving SysML and the System Modeling Environment to Support MBSE**

***Future Directions for SysML***   
The OMG Systems Engineering Domain Special Interest Group (SEDSIG) chartered the System Modeling Assessment and Roadmap WG to assess how well SysML is supporting MBSE, and to develop a roadmap for SysML as part of a System Modeling Environment. The WG is beginning to identify driving requirements for the next generation of SysML and the tools that implement the language. Some of the initial capabilities and requirements are below. These are subject to further analysis, inputs, and review with the broader community.   
System modelers who perform MBSE in the broader context of Model-Based Engineering (MBE) use a System Modeling Environment (SME). This environment must provide basic capabilities that impose requirements on both the modeling language and the tools.

Some of the key capabilities for the SME include:

* Model construction
* Model visualization
* Model analysis
* Model management
* Model exchange and integration
* Support for MBSE collaboration and workflow
* Support for extension/customization (added after the article)

Some of the key effectiveness measures include:

* **Expressive:** Ability to express the system concepts
* **Precise:** Representation is unambiguous and concise
* **Presentation/communication**: Ability to effectively communicate with diverse stakeholders
* **Model construction:** Ability to efficiently and intuitively construct models
* **Interoperable:** Ability to exchange and transform data with other models and structured data
* **Manageable:** Ability to efficiently manage change to models
* **Usable:** Ability for stakeholders to efficiently and intuitively create, maintain, and use the model
* **Adaptable/Customizable:** Ability to extend models to support domain-specific concepts and terminology.

Based on the above capabilities and effectiveness measures, some of the preliminary driving requirements for the next-generation system modeling language and tools are as follows:

1. The next-generation modeling language must express the core systems engineering concepts. This requires definition of a robust data model that reflects these concepts. The requirements that drove SysML derive from the original Systems Engineering Conceptual Model, jointly developed by the INCOSE/OMG/AP233 WG requirements team. Modifications and refinements to this model will occur in light of lessons learned over the last several years, and as necessary to express the core systems engineering concepts.   
    
2. The next-generation modeling language must include precise semantics that avoid ambiguity and enable a concise representation of the concepts. SysML currently leverages the UML metamodel for much of its semantic foundations. The language must derive from a well-specified logical formalism that can leverage the model for a broad range of analysis and model checking. This includes the ability to validate that the model is logically consistent, and the ability to answer questions such as the impact of a requirement or design change, or assess how a failure could propagate through a system. The language and tools must also integrate with a diverse range of equation solvers and execution environments that enable the capture of quantitative data.   
    
3. The next-generation modeling language and tools must provide flexible and rich visualization and reporting capabilities to support a broad range of model users. SysML currently includes concepts for view and viewpoint. Tool vendors and end users have been able to apply this capability to query the model and provide flexible reporting capability. The next generation must extend this capability with advanced visualization techniques that include dynamic zoom, filtering, traversal of model relationships, and visualization of the dynamic behavior of a system, such as those provided by simulations. The modeling language must also support symbol libraries that extend well beyond the current SysML notations. In addition, the modeling environment must provide a simplified web interface to dynamically view the model from a diverse set of viewpoints.   
    
4. The next-generation modeling language and tools must enable much more intuitive and efficient model construction. It often requires several clicks to capture a core concept in a model. More streamlined and efficient user interfaces could reduce the time and effort to build and maintain a model. The ability to repeat common modeling patterns with reduced user input (e.g., table-based entry) is another capability to increase modeling productivity and understanding.   
    
5. The next-generation modeling language and tools must support MBSE in the broader context of Model-Based Engineering (MBE), where the models and tools fully integrate across discipline-specific engineering tools, including hardware and software design, analysis and simulation, and verification. All these model-based tools working together establish an environment for engineering the total system.   
    
6. The next-generation modeling language must provide a standard application programming interface (API) to provide dynamic access to the model, while providing appropriate access controls. It should also integrate with emerging platforms for managing and integrating model-based content, such as Open Services for Lifecycle Collaboration (OSLC), which is based on linked data and semantic web technology, and the Functional Mockup Interface (FMI), which provides model exchange and co-simulation capability for executable behavior models. Model transformation is another core capability of the SME that provides the ability to translate from one modeling language to another.   
    
7. The next-generation modeling language must be capable of management in a heterogeneous and distributed modeling environment. The ability to manage change to the model, where multiple users are collaborating on a single model, is challenging enough. This basic capability requires extensive branch and merge capability that includes effective means for evaluating and integrating changes from multiple users, while maintaining a history of all changes. These challenges increase when multiple models and tools are all part of the collaboration. The ability to integrate with Product Lifecycle Management (PLM) environments, which enable versioning, configuration, and variant management, is a fundamental SME requirement.   
    
8. Usability must be a primary consideration for the next-generation modeling language and tools. As noted previously, the learning curve for the SysML language and tools is quite steep The next-generation modeling language and tools must enable efficient and intuitive use by a broad range of users with diverse skills. This imposes requirements on model precision, model construction, model visualization, model management, and several other aspects of the language and tools.   
    
9. The next-generation modeling language and tools must be highly adaptable and customizable to multiple application domains. This implies that the modeling language must be extensible to address domain-specific concepts, and that the modeling tools provide flexible means for the user to enter, analyze, and visualize model data in ways that are meaningful to each domain. In addition, the SME must accommodate customization performed in a standard and rigorous way.   
    
10. To protect investments made by organizations, the next-generation modeling languages must support the migration of existing models with minimum information loss. Models must also be capable of being stored in neutral formats, retained for future access.   
    
11. The next-generation modeling language and tools must be modular and extensible to enable evolution of the above capabilities to take advantage of on-going advances in technologies, concepts, methods, and theories.   
    
 

**Attachment E**

**Hybrid SUV Change Scenarios**

**November 26, 2015**

The following are the proposed updates to the Hybrid SUV change scenarios based on our discussion at our last telecon on November 23, 2015. Some background information and the change scenarios A, B, and C are included below.

During our telecon and in your email below, you note that the change board may request a group of changes to be evaluated that may include previously deferred changes from a change log. The scenarios below make minor reference to this, but this can be further accommodated in a future version if desired.

Regards,

Sandy

**BACKGROUND**

The following Hybrid SUV change scenarios are instantiations of the 'Process Change Request' scenario that you documented in the attached *Review Document For Process Change Request V6*. The scenarios are intended to align with the top level generic process, but the scenarios have not been compared with the lower level processes.

The scenarios are intended to provide a common basis for the Concept Leads to refine their concept and derive requirements for the System Modeling Environment (SME). It is not intended to be exhaustive, but sufficient to derive a representative set of requirements.

The System Modeling Environment includes the system modeling tool, and contains or integrates with the following kinds of tools to execute the change scenarios:

* Requirements management
* System modeling
* CAD
* CAE
* Software modeling
* Performance analysis (e.g., Simulink)
* Safety analysis
* Project management
* Problem reporting
* Configuration and change management

As we discussed, the Hybrid SUV in the SysML v1.4 Specification (Annex D) emphasizes a thread for fuel efficiency. In order for this change scenario to be more generally applicable, the scenario is initiated based on different issues that surface in different phases of the lifecycle as follows:

* During the pre-PDR phase, Contracts is notified of a change in the government regulation to improve fuel efficiency on selected models by 10 % within 3 years. Specifically, the fuel efficiency for this planned vehicle model must increase from 25 mpg to 27.5 mpg.
* During the I&T phase, Quality provides test data that shows the vehicle fails to meet its fuel efficiency requirement.
* During the operational and support phase, NHSTA issues a recall to the Automobile manufacturer to address a high rate of failure of the catalytic converter that results in both high emissions and poor fuel economy

**SCENARIOS**

The change scenarios A, B, and C for the Hybrid SUV are included below.

**Note:** From the June 2016, SysML v2 WG meeting in Orlando, refer to the following presentations dealing with the change scenario (available on the Orlando meeting wiki page)

* 02A-sysml\_v2\_meeting-MBE\_Change\_Scenario-20160621-muggeo
* 02B-sysml\_v2\_meeting-MBE\_Change\_Scenario\_PoC-ARAS\_XPLM-20160621-muggeo-pfenning-chadzynski
* 04-sysml\_v2\_meeting-Hybrid SUV Scenario Elaboration for Analysis-20160621-bajaj

**Scenario A**

**Issue identification:**

During the pre-PDR phase, Contracts is notified of a change in the government regulation to improve fuel efficiency on selected models by 10 % within 3 years. Specifically, the fuel efficiency for this planned vehicle model must increase from 25 mpg to 27.5 mpg.

**Submit change request:**

Contracts prepares a change request and submits to the program change control board (CCB). The change request is submitted via an on line form that identifies the proposed requirement change, who submitted, when submitted, and rationale for the change by referring to the modified government regulation.

**Categorize change request:**

The CCB categorizes the change as class A (significant impact with high priority). This information gets added to the change request. This change is grouped with another previous change that relates to improved fuel efficiency.

**Evaluate business impact:**

The CCB determines that this will impact the production line of this new model beginning in January, 2018. The production schedule impact is added to the change request.

**Assess architecture impact/Assess preliminary design impact:**

The first attempt of this iteration is to determine whether software changes alone can achieve the fuel efficiency by modifying the fuel control algorithms. It is determined that approximately 5% can be accomplished via an algorithm change, and the other 5% must be accomplished through hardware changes. The assessment will include whether this can be accomplished through weight reduction to account for the other 5%, which may include a combination of material and design changes.

The preliminary assessment is performed by first identifying all affected elements across the system architecture that may be impacted. The hardware and software designers use their tools to conduct a preliminary feasibility assessment of what can be accomplished. The engine control team determines that they can achieve 5% efficiency through adjustments to the gear selection algorithm (e.g. switching gears at lower rpm). However, this reduces vehicle acceleration performance. As a result, the weight must be reduced sufficiently to achieve the additional fuel efficiency and to maintain acceleration performance. Preliminary weight reductions are identified by potential design and material changes to the engine, transmission, and battery (e.g., power train). In addition, consideration is given to adding a tire pressure control feature if the fuel efficiency cannot be achieved through the weight reductions.

Several analysis are performed as part of this initial proposed change, including fuel efficiency analysis, weight analysis, and acceleration analysis. A hazard assessment is also performed to confirm the changes do not adversely impact vehicle safety.

The above architecture and design impacts along with their rationale and traceability are captured in a preliminary change impact assessment, which is referenced in the change request.

**Review change request impact assessment:**

The CCB reviews the change impact assessment and approves the decision to proceed with the change, and adds the decision, rationale, and approvals to the change request.

**Plan release and effectivity:**

A preliminary project plan to implement this change is developed which includes tasks to update the hardware design and software design and implementation, updates to verification plans, manufacturing plans, and supplier plans.

**Update architecture and specifications /Assess detail design impact:**

A trade study is performed to assess the feasibility of adding a tire pressure control feature, but this is deemed as not cost effective for this vehicle design.

Further detailed hardware and software design impact assessment is performed to determine changes to requirements to the system, element, and component level specifications. This includes changes to the system fuel efficiency requirement, the engine control software requirements, and the weight requirements for the engine, transmission, and battery. The battery is a long lead item that is acquired from a supplier, so the supplier specification must be modified along with changes to the supplier contract. Manufacturing determines that the manufacturing process and tooling will be impacted based on the hardware design and material impact assessment.

The results of this activity are proposed redlines to specifications and to updates to the architecture design which are captured in the preliminary change package. This impact is also referenced in the change request.

**Review final change estimate:**

The CCB reviews and approves the proposed preliminary change package. This approval is reflected in the updated change request.

**Update Project Plans, Cost and Schedule:**

The preliminary project plan is updated to reflect more detailed plans including cost and schedule estimates to support the detailed hardware and software design, implementation, and verification.

**Assess applicability of change to product family:** This process will be treated as a separate use case, but is included here to give it visibility.Assessing the impact of a change is captured in an existing activity diagram. see section 4.1.2 of the UC review document. This activity is called in the activity “Assess Architectural Impact” and “Update Architecture and Specification”. Both of which are called from the main UC activity.

The change process described above is applied to the product family, where the change control board represents the institutional change control board responsible for managing the evolution of the product family.

**Scenario B**

**Issue identification:**

During the I&T phase, Quality provides test data that shows the vehicle fails to meet its fuel efficiency requirement. Quality generates a problem report. The corrective action process is performed, and identifies a design deficiency in the new fuel control algorithms. This information is added to the problem report.

**Submit change request:**

Quality prepares a change request and submits to the program change control board (CCB). The change request is submitted via an on line form that identifies the proposed design change, who submitted, when submitted, and rationale for the change by referring to the problem report.

**Categorize change request:**

The CCB categorizes the change as class A (significant impact with high priority). This information gets added to the change request.

**Evaluate business impact:**

The CCB determines that this has minimal business impact but must be fixed during the I&T phase.

**Assess architecture impact/Assess preliminary design impact:**

The preliminary architecture and design impact determines that this fix can be made through a software fix. The architecture and design impacts along with their rationale and traceability are captured in a preliminary change impact assessment, which is referenced in the change request.

**Review change request impact assessment:**

The CCB reviews the change impact assessment and approves the decision to proceed with the change, and adds the decision, rationale, and approvals to the change request.

**Plan release and effectivity:**

A preliminary project plan to implement this change is developed which includes tasks to update the software design and implementation and perform additional testing at the engine control software, engine, and system level.

**Update architecture and specifications /Assess detail design impact:**

The preliminary change package includes no change to the specifications.

**Review final change estimate:**

The CCB reviews and approves the proposed preliminary change package. This approval is reflected in the updated change request.

**Update Project Plans, Cost and Schedule:**

The preliminary project plan is updated to reflect more detailed plans including cost and schedule estimates to support the detailed software design, implementation, and verification.

**Scenario C**

**Issue identification:**

During the operational and support phase, NHSTA issues a recall to the Automobile manufacturer to address a high rate of failure of the catalytic converter that results in both high emissions and poor fuel economy.

**Submit change request:**

Contracts prepares a change request and submits to the institutional change control board (CCB). The change request is submitted via an on line form that identifies the defective unit, who submitted, when submitted, and rationale for the change by referring to the NHSTA recall.

**Categorize change request:**

The CCB categorizes the change as class A (significant impact with high priority). This information gets added to the change request.

**Evaluate business impact:**

The CCB determines that replacing the faulty catalytic converter will have a very significant business impact on the vehicle under question, but also will potentially impact other vehicles currently under development, since this same catalytic converter is used on the newer vehicles. The potential impact is added to the change request.

**Assess architecture impact/Assess preliminary design impact:**

The initial assessment identifies two alternative fixes, including replacing the part with another, or modified the current part. The two alternatives are identified, and analysis is performed to assess each alternative, including fuel efficiency analysis, emissions analysis, and reliability analysis. The results are captured in the preliminary change impact assessment, and referenced in the change request.

**Review change request impact assessment:**

The CCB reviews the preliminary change impact assessment and approves the decision to proceed with the part replacement, and adds the decision, rationale, and approvals to the change request.

**Plan release and effectivity:**

A preliminary project plan to implement this change is developed which includes tasks to perform the detailed design change with the replacement part, which involves some mechanical interface changes, and procurement of the replacement part. This change will impact multiple vehicle models.

**Update architecture and specifications /Assess detail design impact:**

The results of this activity are proposed redlines to the catalytic converter specification and a minor interface requirements change to the manifold. The updates are captured in the preliminary change package. This impact is also referenced in the change request.

**Review final change estimate:**

The CCB reviews and approves the proposed preliminary change package. This approval is reflected in the updated change request.

**Update Project Plans, Cost and Schedule:**

The preliminary project plan is updated to reflect more detailed plans including cost and schedule estimates to support the detailed design and implementation changes, procurement of the new part, and verification. The detailed design implementation and integration of this change, and the management of the change, will be the responsibility of each individual Vehicle program that is impacted.