***System Modeling Assessment & Roadmap WG,***

The following is a summary and follow-up actions from our two day face-to-face Working Group meeting at the Reston OMG meeting on March 15 and 17, 2016. My thanks to all who contributed. The meeting summary is posted on the Reston WG meeting page at:

<http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_mar_2016_meeting_presentations>

The following sections are included below:

* Follow-up Actions
* Background
* Meeting Summary
* Orlando Meeting
* Reston Meeting Feedback-2016-March 15,17
* SysML v2 RFP
* SME Concept
* Evolving SysML and the System Modeling Environment to Support MBSE (Aug 2015 Insight Article)

Hedley will provide the web and dial up information for our next WG telecon on Wednesday, April 20, at 11:00 AM ET. Our next face-to-face meeting will be at the OMG meeting in Orlando, Florida tentatively scheduled for Tuesday and Thursday, June 21 and 23, where we will continue to develop the concepts, requirements, and plans for SysML v2.

**FOLLOW-UP ACTIONS**

* Sandy to post meeting summary and presentation slides to the WG Wiki.
* Hedley to schedule the next WG telecon for Wednesday, April 20, at 11:00 ET
* Concept Leads to address the feedback from the Reston Meeting included in the section below called *Reston Meeting Feedback-2016-March 15, 17*
* Capability Leads to review the SME concept in the section below called *SME Concept* and provide any updates to Sandy
* Capability Leads to continue to refine their concept which should include:
  + 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
* Concept leads to maintain their Wiki page for their SME concept
* Axel to host a meeting with the participating tool vendors including Hedley, Chas, and Josh to discuss the level of specification for the SME service requirements to ensure they do not overly constrain their implementation
* Sandy to establish an interface core team to evovle the requirements for the Interface Concepts
* 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

**BACKGROUND**

This WG was chartered to develop the requirements for the SysML v2 RFP, which is intended to be a next generation System Modeling Language. The WG is in the early stages of 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 will be derived from the SME 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), and 11 driving requirements for the SME, which are included in their own section below. The publishing of this article served as an initial baseline and is an important milestone for SysML v2 development. Our next milestone is to baseline the SME Concept, which is also summarized below.

**MEETING SUMMARY**

The objectives for our face to face meeting were to review the concepts for each Systems Modeling Environment (SME) capability, and begin discussions on the SysML v2 RFP approach. The SME concept will be a key input for deriving the requirements for the SysML v2 RFP.The agenda included the following:

**MEETING AGENDA**   
**Tuesday, March 15, 2016**

09:00 - 09:30 Introduction - Sandy Friedenthal

**Concept reviews**

09:30 - 10:30 Modeling Formalism [R2] - Yves Bernard

10:30 - 11:00 Break

11:00 - 12:00 Analysis Concept [R2] - Manas Bajaj

12:00 - 13:00 Lunch

13:00 - 14:00 Model Construction [R4] - Ron Williamson

14:00 - 15:00 Model Visualization [R3] - C Schreiber, J Feingold, M Sarrel

15:00 - 15:30 Break

15:30 - 16:00 Model Management - Laura Hart

16:00 - 16:30 Integration with PLM [R7]  - U Kaufmann, M Pfenning

16:30 - 17:30 MBSE Collaboration & Workflow - Hedley Apperly

**Thursday, March 17, 2016**

09:00 - 09:15 Introduction & Recap

09:15 - 10:15 Systems Engineering Concept Model (SECM)  [R1] - John Watson

10:15 - 10:45 Break

10:45 - 12:00 Property Model  - Concept Model Core Team

12:00 - 13:00 Lunch

13:00 - 14:30 Model Interoperability [R5]  and Standard API [R6] - Axel Reichwein

13:00 - 14:30

14:30 - 15:00 Break

15:00 - 16:30 SysML v2 RFP Approach and Scope

16:30 - 17:00 Next Steps - All

Each Concept Lead presented their concepts and status. Some of the presentations are included on the [Reston meeting page](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_mar_2016_meeting_presentations). Other Capability Leads presented directly from their Wiki pages which can be found on links under the Working Groups on 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 the meeting which is captured in a section below, and continue to refine their concepts. This includes defining effectiveness measures and driving requirements for their concept, which address the limitations of current system modeling environments with SysML to support MBSE. They should update the service requirements to reflect the required functionality\*, and show how their concept supports the Hybrid SUV change scenarios, and where practical, provide actual prototypes of potential solutions to demonstrate feasibility of their concept. Also, Concept Leads should maintain their concept on their Wiki page to provide visibility for broader review and vetting.

\*Note: Please refer to Ron Williamson's [Model Construction Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_model_construction_focus_area) for excellent examples of [service definitions](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_model_construction_focus_area#services_-_model_construction) and how the model construction capability concept supports the [Hybrid SUV change scenario](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_model_construction_focus_area#change_scenario_example_-_model_construction_focus).

**RFP DISCUSSION**

During the last 2 hours of the meeting, we began discussion on the approach to the SysML v2 RFP which included the following general considerations.

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

2. The RFP will require the following to support the SME capabilities:

* Metamodel
* UML Profile
* Model Libraries
* Concrete Syntax
* Standard API
* Reference model (used for demonstrating conformance level)
* User Interface Guidelines
* Standard extensions mechanisms

3. Test cases will be included along with a reference model to demonstrate the level of conformance of the SysML v2 Implementation to the specification as indicated below.



4. There was considerable discussion as whether to proceed with a single RFP or incremental RFP's. An incremental RFP could include separate RFP's for various incremental capabilities such as incremental improvements to behavior, structure, requirements, parametrics, or perhaps incremental RFP's to support different capabilities such as visualization capabilities. Hedley opposed incremental RFP's, and felt this would dilute the business case for SysML v2. However, some compromise recommendations included:

* Issue separate RFP's for the data model and the standardized API
* Include an initial scope similar to current SysML, and defer additional scope to future RFP's.

5. The vendor inputs included the following:

* Hedley prefers a single RFP
* Chas wants tool interoperability, and to understand the impact of services on their implementation
* Josh wants reliable access to the SysML data whether it is exported as XMI or accessed via restful services with RDF
* All wanted clarification between the API and the services

**ORLANDO MEETING**

The following are tentative topics for the Orlando, Florida meeting on Tuesday and Thursday (June 21 and 23, 2016)

1. SME Concept Update

2. RFP Approach

* Scope (supported MBSE processes, data model, and service requirements)
* RFP outline
* Roadmap
* Other (as time permits)
  + RFP responsibilities (e.g., service req'ts, SECM, ..)
  + Constraints
  + Related standards
  + Conformance
  + Issues to be addressed

3. Other Topics

* Proposed Plan for an RFP for a Safety Profile of SysML - Geoff Biggs
* TBD

**Reston Meeting Feedback-2016-March 15,17**

**Model Formalism Concept:** Yves Bernard

1. Update your definition of requirement to correspond to the definition used in chapter 16.1 of the SysML spec as follows: "*A requirement specifies a capability or condition that must (or should) be satisfied*."

2. Consider the following which may result in new or modified requirements and or evaluation criteria:

* Emphasize the need for SysML v2 to be computer interpretable.
* Clarify the need for both a textual notation similar to a programming language such as ALF and graphical notation for the formalism.
* Clarify that the SME should make you aware of inconsistencies, and provide options for resolving them.
* Clarify that the formalism should provide a precise and concise extension mechanism (similar to stereotypes).
* Clarify that the use cases for model checking should include well formedness checking such as for interface compatibility.
* Clarify the need to be able to apply varying degrees of formalism. For example, the language should not overly constrain the user when developing concepts, recognizing that concepts are formed throughout the life cycle and not just early on. This is partially addressed by differentiating the level of precision from the level of detail, but may also allow for relaxing formalism constraints based on user need.
* Continue to refine the usability criteria and how the formalism can impact usability. As an example, some system engineers struggle to understand how to organize a model which involves the concepts of namespace and containment. The criteria should enable us to measure and assess how well a concept aligns with the way different representative users think. Consider the following representative users:
  + A new SysML modeler who may be experienced with other models such as Simulink
  + An experienced SysML modeler
* Identify alternative formalisms and their pros and cons based on the requirements and evaluation criteria.
* Set up a Wiki page for the formalism concept and approach.

**Model Analysis Concept:** Manas Bajaj

1. Clearly distinguish between *logical analysis* and *quantitative analysis* in the concept. These two types of analysis impose different requirements on the SME. The logical analysis can impose requirements on the language formalism to support model checking for well formedness, such as interface type compatibility, and for making logical inferences about the design such as an impact assessment of a requirements change. On the other hand, the quantitative analysis can impose requirements on the property and expression concepts, and on the approach to integrating the SysML model with other engineering analysis models and tools.

The two types of analysis are often merged from a user perspective.  For example, some logical assertion may be tested to be true or false such as when verifying a requirement, but engineering analysis may be performed to obtain results that are compared in the boolean expression to evaluate whether the requirement is satisfied. ***Note:*** Your presentation also refers to qualitative analysis. This seems like a hybrid between logical and quantitative. It includes quantitative analysis where the parameter values are unknown, and logical analysis such as requirements coverage checks.

**Coordinate** with Yves on the development of the approach for logical analysis, Hans Peter on the Property and Expressions concept model, and Axel Reichwein on the approach to integrate the SME with other engineering analysis models.

2. As noted by Hans Peter and Rick, there needs to be more emphasis placed on verification by analysis throughout the life cycle beginning with requirements verification in the concept phase through downstream analysis of an as-built system. Sometimes the verification includes a combination of verification by analysis and testing such as with hardware/software in the loop testing.

**Model Construction Concept:** Ron Williamson

1. Consider combining the following 3 model construction use cases to illustrate the concept.

* + Constructing a model from scratch
  + Reusing an existing model
  + Importing unstructured legacy data (e.g. word, excel)

However, keep them distinct in terms of specifying the requirements on the SME.

2. Your recommendation to relate the methodology workflow to model construction is an excellent insight. Based on further discussion, we suggested that when constructing a part of the model, it should be done as part of implementing a specific practice. This may be accomplished by role based workflows that are defined in the practices repository, and are executed as wizards that apply profiles, patterns, and libraries. This approach should be coordinated with Hedley as part of the MBSE Collaboration & Workflow concept.

3. Provide a capability to perform a batch create, read, update, or delete of a set of model elements which are specified by some criteria such as importing the items as blocks in a particular package,

4. Emphasize the need to support a textual input to the model. **Coordinate** with Hans Peter and the Property and Expression Concept to provide support for a string property which captures the text, and can include element references to other parts of the model.

**Model Visualization Concept:** Chris Schreiber

1. Include a concise description of the model view controller (MVC):

* Model is the model data
* View is the presentation of that data and corresponds to a *View* in SysML
* Controller is the mechanism to transform the model data to a View, and is specified in SysML by the *Viewpoint Method*. This method defines the rules for constructing the view, and specifies the query of the model, data organization, analysis, and presentation of the results.

2. In addition to the graphical views, explicitly include the textual view in your concept. The Open MBEE/Doc Gen/View Editor concept is a good candidate for this.

3. Provide a textual query language (e.g. SPARQL)

4. Include the Bruce Douglass example to illustrate an interesting domain specific symbology

**Model Management Concept:** Laura Hart

Note: I did not capture the detailed feedback from your presentation. Please provide your notes.

1. Per Chris Delp, clarify versions versus revisions.

2. Identify which model management use cases can be incorporated into the PLM/MBSE RFI noted below.

**PLM/MBSE Integration Concept:** Uwe Kaufmann and Michael Pfenning

1. Uwe made some suggestions to refactor some of our WG's including PLM/MBSE and model interoperability. Sandy noted that we can consider the refactoring after the initial baseline of the SME concept is established. There is overlapping concepts between many of the working groups such as model construction and model visualization, and model formalism and model analysis. In addition, the PLM/MBSE integration includes overlapping concepts with model management, model interoperability, collaboration and workflow, but we will need to retain distinct focus on model management, workflow and the practices repository, and on the standardized API.

2. There is an opportunity for the model management use cases and requirements that Laura presented to be included and perhaps validated and refined through the PLM/MBSE RFI process. The required SysML v2 model management services should be provided under the general umbrella of PLM, although there are many potential PLM architectural solutions to satisfy this.

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

1. Provide the capability to associate workflows with model construction per Ron Williamson suggestion above. As a 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 wizards which apply profiles, patterns, and libraries. In addition, Hans Peter emphasized the need for the workflow for to support assignment of highly granular role based access to the data. An example is a reliability engineer or other specialty engineer that is responsible for specific properties of the system model. Roles that are defined with clear data ownership responsibilities, are associated with specific workflows and the corresponding practices, metrics, approvals, etc.

2. The metrics collection are captured throughout the workflow and provided back to the practices repository. In particular, the cost side of the equation is captured during each step of the workflow. Other metrics such as complexity can be derived from the model itself. Other value metrics may be provided at key points in a workflow such as quality metrics captured during a peer review.

3. Chris Schreiber discussed an important need to bundle the input data when performing a task. This is a key part of the workflow since engineers spend an inordinate amount of time trying to collect the right data to do their work. Chris also noted that the data often comes from outside the model, such as the qualification data for a part.

4. Set up the Wiki for this concept

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

1. Include https in the web standards to support the security/encryption layer

2. As a starting point for the standardized API, capture common service list based on the Excel spreadsheet on your Wiki. Working Groups will include a link to the relevant services. (please refer to 3 spreadsheet attachments that have not been fully reconciled, and include the list of services Ron Williamson provided on the Model Construction wiki).

3. Generate an example API for a service to apply a pattern

4. Contact Hans Peter on the API he developed for the Concurrent Design Facility that enables high performance synchronization in a collaborative environment (50 users with updates 2 updates per minute)

5. Set up telecon with Josh, Chas, and Hedley to discuss how to specify services to avoid over constraining implementations (refer to example service specification provided by Hedley under the Documents section of our [WG Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:sysml_assessment_and_roadmap_working_group))

6. Include the interoperability figure on the Wiki

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

1. Laura will review the SECM and identify gaps with respect to concepts in the UAF Domain Model. These gaps will be considered for inclusion into the SECM.

2. Include the concept of derived model elements. Sometimes the model element is not explicitly in the model but may be derived. An example is a derived connector between two composite elements whose nested parts may have a connection. An analogy was provided with a google maps view of a city block. Although there is no specific concept of a city block in google maps, it is a useful construct for some viewpoints.

3. The need for a geometric view was identified, which may be another diagram kind.

4. John presented the interface concepts. It was observed that the interface table includes data associated with multiple layers of an interface and the transformation between them. For example, there is a logical data element, with a message in bits and bytes that corresponds to the application layer, and the characteristics of the electrical signal that corresponds to the physical layer. Include this clearly in the requirements.

5. Clearly state that the connector is distinct from the interface medium (e.g., harness, pipe)

5. Capture the limitations of current interface modeling capability in SysML including the ability to more readily generate views of different abstraction levels. There is a critical need for derived model elements such as the ability to connect 2 nested parts, and then display connections between their parent parts.

6. Following the meeting, the request was made to include provisions for alias in the concept model that enable extensions that use different labels for the same concept.

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

1. Expressions should be elaborated in the model.

2. A comment from Robert Karban following the meeting noted that there is a critical need for a property type which includes strings that can reference other model elements (refer to his email dated March 21). This property type can be used in default values and expressions to reference other model elements. For example, a default value can refer to another property or to an expression containing other element references. In this way, expressions define as strings can include element references.

3. Make provisions to associate an expression language with a string to enable it to be interpreted and parsed.

4. Hans Peter presented an elegant concept for Sampled Function Property Type. This seems to address many issues such as those raised by George Sawyer in an email following the meeting. Show how George's issues are addressed explicitly.

5. Hans Peter made a compelling argument for the need for dimensional analysis by noting that there may be different models with different units that must be reconciled. This should be stated clearly as part of the concept.

**SysML v2 RFP.** 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



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

**System Modeling Environment (SME) Concept**

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

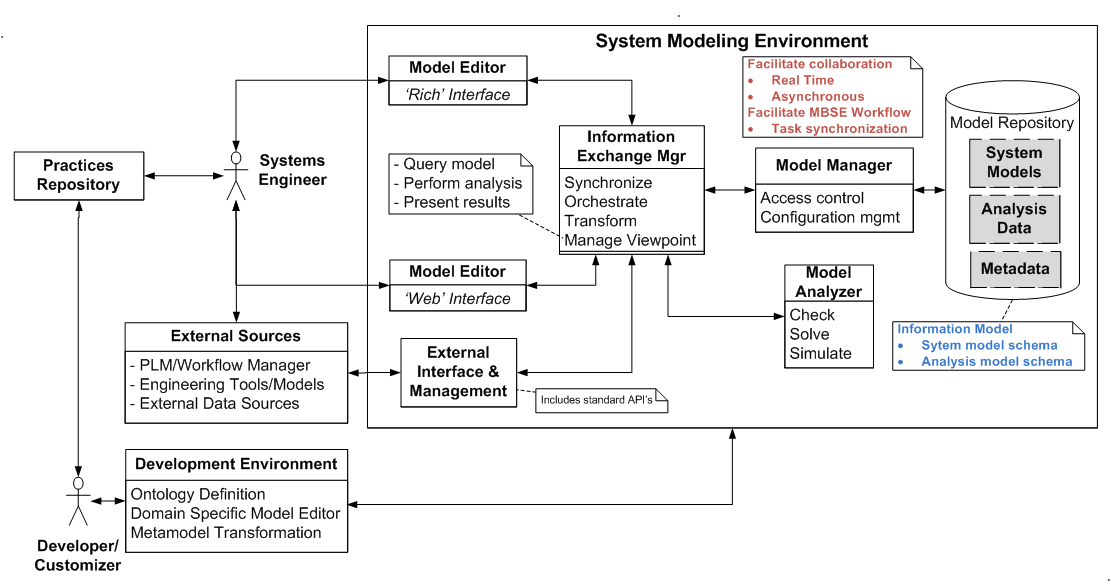
In Figure 3 below, each of the disciplines contributes to the overall management and technical baseline of the system under development as part of a Model-Based Engineering (MBE) approach. The System Modeling Environment is the systems view of the MBE Environment, and is used by the systems engineers and others to perform MBSE and evolve the system model in the broader context of MBE. The system model provides an overall description of the system that helps integrate the models and associated design information from the other engineering disciplines.

**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

Figure 4 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. In addition, there is an external interface to enable direct interaction between the system model and other engineering tools and models. Additional functionality is provided by other logical elements to support exchange, management, and analysis of the model data, as well as support for collaboration and workflow.

In addition, the figure identifies the development environment used to further extend and customize the environment. This is essential to enable domain specific extensions of the environment to address the broad set of uses of the SME. However, this environment provides a standard extension mechanism to maintain interoperability of the environment.

Finally, there is a practices repository that stores a set of practices that are used by the workflow engine to support MBSE.



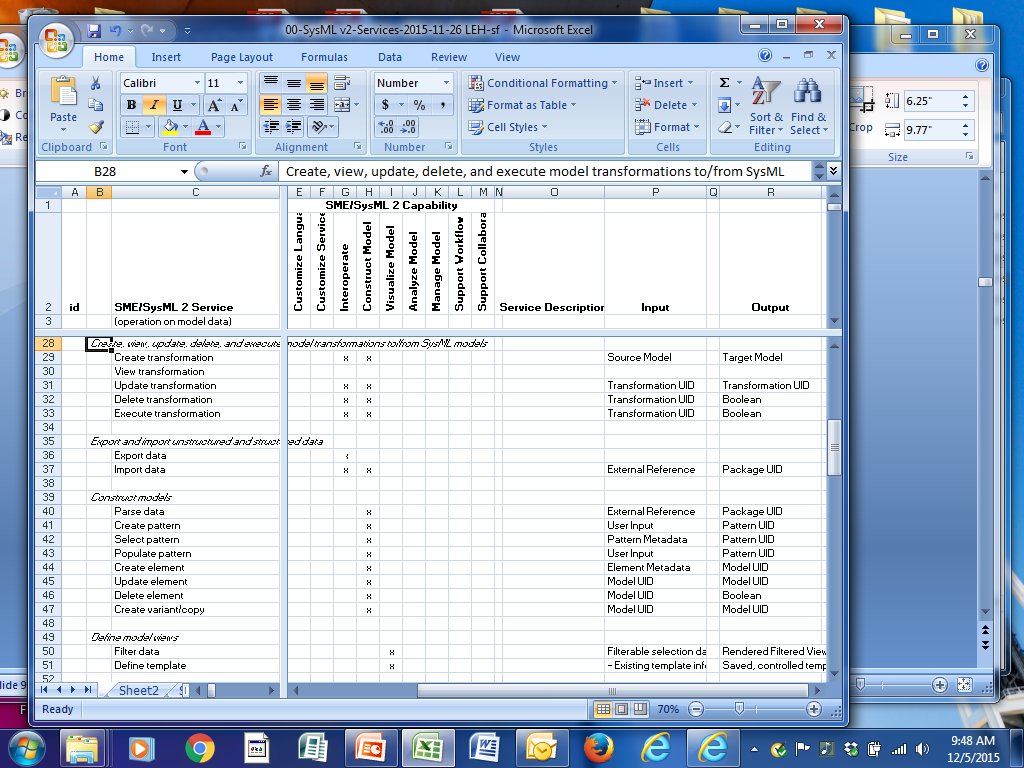
**Figure 4. System Modeling Environment-Logical Architecture**

Figure 5 shows the layers of the SME architecture using a fairly traditional architecture layering approach. 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. A standardized application program interface (API) is used to access these services. The graphical user interface and one or more adapters may then leverage 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 5. System Modeling Environment Layered Architecture**

A partial and preliminary list of services that support the SME capabilities is shown in FIgure 6. This functionality will be used to help specify the API.

****

**Figure 5. Service Requirements (Extract from initial draft)**

The Systems Engineering Concept Model (SECM) will be a primary input to the requirements for the SysML data model including the metamodel, profile, and 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 presented at the Reston OMG meeting on March 17 is shown in Figure 6.

****

**Figure 6. Systems Engineering Concept Model (SECM-2015 Industry Reference)**

**(Extract from recent draft)**

**SME Key Features:** The following are key features of the System Modeling Environment (SME) Concept presented to date.

**General capability.** The SME enables systems engineers and other to perform model-based systems engineering activities (refer to [SE Use Cases Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:use_case_working_group) for typical examples). It 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
* Reuse 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
* 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, different domains, different lifecycle phases, and different levels of rigor. The SME provides user access to the model data via a simplified web-based interface and a rich modeling interface with full functionality.
* Continuous feedback is provided to the user on their model development, along with on line help guidance.
* The SysML v2 specification includes guidelines to ensure the user experience is both efficient and intuitive.

**Systems Engineering Concept Model (SECM).** The SECM will capture the key concepts needed to support systems engineering (refer to [SECM Wiki](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_concept_model_workgroup)).

* The SECM is captured in a modeling language
* The SECM industry reference model is based on the SEBok, ISO 15288, and INCOSE SE Handbook
* The SECM RFP model incorporates selected inputs from the industry reference model, other industry ontologies, and related OMG specifications
* The SysML v2 specification implements the SECM RFP model in a metamodel, profile, and model libraries which provide a precise, unambiguous, concise, and intuitive expression of the SECM concepts

**Language formalism.** The language formalism provides the foundations needed to support logical analysis (refer to **TBD** Wiki).

* Semantic foundation which is precise enough 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.** SysML v2 models will provide basic solver capabilities, and seamlessly integrate with diverse engineering analysis models (refer to **TBD** Wiki).

* Support for specifying an analysis including input/output parameters and their relationships, analysis objectives, assumptions, and fidelity
* 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
* Support for standardized expression languages
* Enhanced system of quantity and units libraries with support for dimensional analysis
* Support for static analysis and simulation via built in solvers
* 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

**Model construction.** The SME will 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)).

* User input
  + Create new model from scratch
  + Reuse existing model
  + Link data to an external data source
  + Import from external data source (structured and unstructured data)
* User can construct model using text or graphical input
* Batch operations for create, update, delete
* Extensive reuse libraries
* Support for model defining and applying patterns
* User interface supports domain specific viewpoints
* Support for process implementation (e.g., wizards that 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
* Collaborative, team-oriented model construction

**Model visualization.** The SME must provide flexible and rich visualization and reporting capabilities to support 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)).

* Graphical, tabular, and textual views of the model
* 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 willintegrate 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/)
  + **Note:** include reference to secure access (e.g. https)
* Standardized formats for persistent data storage
* Other standards to support model execution and analysis such as FMI

**Model management.** The SME will becapable of managing 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
* Integrate with PLM environment
* Manage SysML and its dependencies as a graph
* Unique id for all model elements
* Model configuration item (CI)is the level the model is versioned
* Model diff hierarchical from CI to element level
* Model branch and merge capabilities
* Capture incremental changes efficiently
* Ensure automated transformation between SysML v1 and SysML v2 and later versions
* Broad support for model transformation
* Role based access controls

**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 **TBD** Wiki).

* Executable workflows which bundles relevant input data
* Process execution
* Change notification
* Practices repository
* Metrics collection
* 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 wizards which apply domain specific user interfaces and profiles, patterns, and libraries, with highly granular role based access to the data (e.g. clear data ownership). Others are notified of the change as defined by the workflow.

**Extensibility facilities.** The SME enables extension to the Meta-model, Services, and User Interface. Some examples of extensions may 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

**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.