***SysML v2 RFP Working Group Summary***

***Reston, VA***

***March 21-23, 2017***

The following is a summary and follow-up actions from our 3 day face-to-face Working Group meeting at the OMG meeting in Reston, Virginia on March 21-23, 2017. My thanks to all who contributed. The meeting summary and presentations posted on the Reston meeting page at:

<http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations>

**Note:** The [presentation](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) links in this summary take you to the presentation section of the Reston meeting page where you can download the presentations.

Hedley will provide the web and dial up information for our next WG telecon on Wednesday, April 5, at 11:00 AM ET, where we will review the meeting results and discuss the status of the follow-up actions and plans.

*We received excellent feedback from our initial requirements review. Based on the estimated effort to address this feedback and fill the gaps in specifying the requirements, the proposed RFP issue date will be extended one meeting cycle to December, 2017. In order to meet this schedule, our plan is to issue an initial draft at the June OMG meeting, and a high quality draft RFP at the September, 2017 meeting. I request your support to help achieve these important milestones by addressing the actions below.*

**ACTIONS**

* Sandy to post meeting summary and presentation slides to the WG Wiki
* Hedley to provide web session for the next WG telecon on Wednesday, April 5, at 11:00 AM ET
* All to review this meeting summary and provide comments

***The following actions are from the meeting. The primary action is to refine the requirements per the requirements guidelines and prepare a draft RFP by the June 5 OMG meeting.***

* Sandy and John to provide an example of a data model requirement and a service requirement
* Robert to generate SysML v2 RFP requirements in web viewable format using OpenMBEE
* Robert to provide a link to OpenMBEE web viewable document with usage instructions
* Jonathan to update the formalism requirements in the model
* Jonathan to consider adding a requirement to provide a textual concrete syntax
* Jonathan to consider adding a requirement to apply logic consistently throughout language
* Manas, Sandy, and John to review and update analysis pattern and associated analysis concepts
* Hans Peter to update the structure modeling requirements.
* Hans Peter to update requirements for properties and expressions.
* Hans Peter and Manas to propose requirements for standard expression languages
* Marc to update the requirements for modeling interfaces.
* Marc to refer to signal flow and physical interaction profile in the RFP as a reference
* Sandy to initiate the Behavior modeling working group to update the issues and compelling example and develop the behavior concepts and requirements
* John Watson to prepare a draft of the common core concepts to present for review at an upcoming telecon.
* Manas to include a requirement for a standard way to connect to the server and a standard discovery service.
* Manas to update the API Requirements based on the findings from the demonstration.
* Manas to consider prototyping a more complex service to query the model and perform an engineering analysis to reason about the system
* Manas and Ed to clarify whether the platform independent model defines operations on the meta-classes in the SysML metamodel (Note: the metamodel is developed by the submitters)
* Ron to update model construction service requirements with pre and post conditions. Send requirements to Geoffrey Biggs and Andy Ko for review.
* Chris to update model visualization service requirements with pre and post conditions
* Laura to update model model management metadata and service requirements with pre and post conditions
* **Action:** Laura will update the model management concept diagrams to reflect the above
* Hedley to confirm the following minimum required workflow and collaboration metadata/concepts to support the above service requirements: [User, Role, Task, Start/End Date, Model Scope, Metric]
* Hedley to update the workflow and collaboration service requirements with pre and post conditions
* John Watson to capture other requirements in the SysML v2 RFP draft for review.
* Goeffrey Biggs to propose key concepts and requirements to integrate SysML v2 with the safety and reliability, and security (TBD) requirements. Initial discussions focused on including the cause-effect relationship in SysML v2.
* Identify potential usability criteria to incorporate into SysML v2 RFP
* Sandy to draft agenda for the SysML v2 RFP WG meeting in Brussels the week of June 5.

**The following sections are included below:**

* Background
* Reston Meeting Summary
* SysML v2 RFP planning
* Next WG Meeting the week of June 5, 2017 in Brussels, Belgium

Sanford Friedenthal

SysML v2 Working Group Chair

[safriedenthal@gmail.com](mailto:safriedenthal@gmail.com)

**BACKGROUND**

The SysML v2 RFP WG was formally initiated on July 23, 2016 at the end of the System Modeling Assessment & Roadmap WG meeting in Orlando, Florida. This concluded an approximate year-long effort to establish a baseline concept for a System Modeling Environment (SME). The SME is the environment that systems engineers interact with to perform model-based systems engineering activities, and the SME concept is used to help derive requirements for SysML v2 as shown in Figure 1.



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

The initial high level requirements for the SME are documented in the August 2015 edition of the INCOSE INSIGHT. 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 to support the specification, design, analysis, and verification of systems. A second article was published in the December, 2016 edition of the INCOSE INSIGHT entitled *'Evolving SysML and the System Modeling Environment to Support MBSE – Part 2'.* This article summarizes the baseline SME Concept in response to the requirements in the earlier article. Both articles are available under the Articles section of the SysML v2 RFP Working Group Wiki at:

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

The SysML v2 RFP defines the requirements for the SysML v2 specification, which is then implemented by tool vendors. The overarching objectives for SysML v2 are to enhance support for MBSE by improving the precision and expressiveness, interoperability, and usability of SysML v2 over SysML v1. The SysML v2 RFP includes requirements to represent system behavior, structure, parametrics, and requirements similar to the requirements for SysML v1, along with requirements that support additional system modeling concepts not explicitly included in SysML v1. The SysML v2 RFP also includes service requirements to support model construction, model visualization, model analysis, model management, and workflow and collaboration. An example of service requirements are the services to create, update, and delete model elements to construct the system model.

Submission teams will develop the SysML v2 specification in response to the SysML v2 RFP requirements. The modeling concepts and associated requirements will be satisfied in the SysML specification by both a SysML metamodel and a profile of UML. The combination of a metamodel and a profile enable a broader range of vendor implementations. The metamodel supports implementation of the system concepts without imposing some of the constraints imposed by UML, while the profile supports implementation of the system concepts that is more closely aligned with SysML v1 implementations. The service requirements in the RFP will be satisfied by specifying a standard API model in the SysML v2 specification. The standard API model facilitates interoperability by enabling external tools, plugins, and user interfaces to access the system model using standard service requests. Tool vendors will then implement the SysML v2 metamodel and/or profile, and the standard API model.

**MEETING SUMMARY**

The Reston meeting is the third dedicated SysML v2 RFP Working Group meeting following the RFP kickoff in Orlando. A meeting was also held on January 29, 2017 at the INCOSE International Workshop in Torrance, California. The summaries from each meeting can be found on the meeting links available from the SysML v2 RFP WG Wiki above. The objectives and agenda for the Working Group meeting on March 21-23, 2017 in Reston, Virginia are included below:

**Meeting Objectives**

* Review Initial Draft Requirements
* Formalism requirements
* Systems Engineering Concept Model (SECM) / Data model requirements
* API requirements
* Service requirements
* Usability requirements
* Review/update plans and model-based approach for SysML v2 RFP

**Meeting Agenda**

*Tuesday, March 21, 2017*

09:00 - 10:15 Usability req’ts working session - S. Friedenthal, B Muth, T Ngoon

10:15 - 10:45 Break

10:45 - 12:00 Usability req’ts working session (cont) - Open Discussion

12:00 - 13:00 Lunch

13:00 - 13:30 Meeting Overview - S Friedenthal

13:30 - 14:15 Model-based specification approach - J Watson/R Karban

14:15 - 14:45 Break

14:45 - 15:00 Declarative Specification Approach - Keith Butler

15:00 - 15:45 Formalism requirements update - J Patrick

15:45 - 17:00 Prototype API presentation and demonstration – Manas Bajaj

*Wednesday, March 22, 2017*

09:30 - 10:30 Draft service requirements - Concept Leads

Model management – L Hart

Model construction – R Williamson

Workflow & collaboration – H Apperly

Model visualization – C Shreiber

12:00 - 13:30 Lunch

13:30 – 13:45 Prototype interface & services for Safety Modeling – G Biggs

13:45 - 14:15 Systems Engineering Concept Model (SECM) Status Update - J Watson

14:15 - 14:45 Break

14:45 - 15:30 Structure Concepts and draft RFP req’ts - Hans Peter de Koning

15:30 – 16:15 Discussion on approach to data model requirements - All

16:15 - 17:00 Interface Concepts and draft RFP req’ts - Marc Sarrel

*Thursday, March 23, 2017*

09:00 - 09:45 Interface Concepts and draft RFP req’ts– Marc Sarrel (cont.)

09:45 - 10:30 Analysis service and data requirements – Manas Bajaj

10:30 - 11:00 Break

11:00 - 12:00 Behavior Concepts (preliminary discussion only)

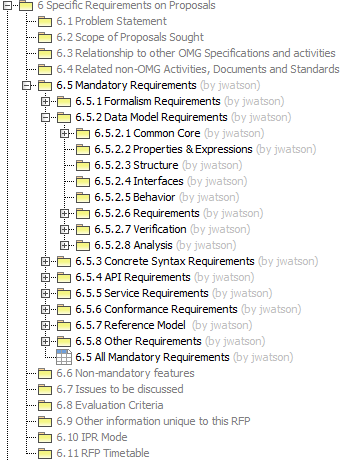
12:00 - 13:00 Lunch

13:00 - 13:45 Requirements Concepts and draft RFP requirements – Sandy F

13:45 - 14:30 Requirements for SysML/UML Interoperability – Ed Seidewitz

14:30 - 15:00 SysML v2 RFP Planning - Sandy/All

**SysML v2 RFP requirements overview and status.** This meeting focused on reviewing the initial requirements for the SysML v2 RFP. The proposed organization for the mandatory requirements in section 6.5 of the RFP are highlighted in the following Figure 2.



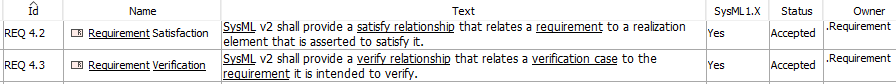
**Figure 2.** Organization of Mandatory RFP Requirements in Section 6.5

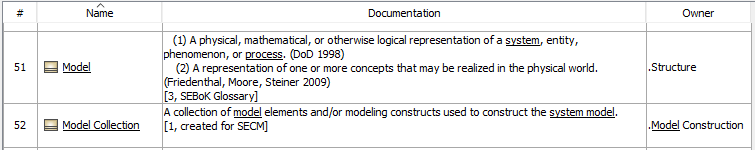
This meeting focused on the review of the initial requirements to represent the systems modeling concepts, the standard API and associated service requirements, and the formalism requirements. In addition, a facilitated working session was conducted to discuss usability requirements for SysML v2. Many of the requirements are captured in the model but are still in an early state of maturity. The intent is to publish a web viewable document of the initial requirements in the near term.

**Model-based specification.** John Watson and Robert Karban [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) a status of the model-based specification approach to generate the SysML v2 RFP. in his presentation (model-based rfp approach R2). John summarized the model content that will be used as a source for generating the RFP requirements. The RFP content includes:

* Text-based requirements
* Glossary containing list of terms and their definitions and reference sources for the definitions
* Concept model representing key system modeling concepts
* Working examples

There are currently 140 requirements and 150 glossary terms in the model. John is also continuing to evolve the modeling guidelines for the model content. Some examples of the requirements and definitions of terms are included below.





Robert Karban described the approach to auto-generate the RFP from the model. The approach leverages the open-source modeling environment called OpenMBEE developed at JPL. The OpenMBEE provides a web interface and tool neutral model repository to view and edit the model, and to generate reports from the model. A modeling tool must provide an interface to OpenMBEE in order to access the model repository.

The OpenMBEE uses its Doc/Gen application to generate the reports. The following link to the DocGen User’s Guide was provided by Ivan Gomes from JPL:

<https://github.com/Open-MBEE/mdk/blob/support/2.5/manual/DocGen%20User's%20Guide.pdf>

**Requirements guidelines.**

Over the course of the meeting, there were several discussions on how to specify the RFP requirements. The following are proposed updates to the requirements guidelines for both the data model requirements and the service requirements:

*Data model requirements guidelines*

* Abstract the requirement to the level that reflects the concept without over constraining the metamodel and profile implementation in the SysML v2 specification.
* Use terms that reflect concepts in the Systems Engineering Concept Model (SECM) and that are defined in the SECM glossary. Wherever practical, the requirement should reflect the real world concept that is being modeled (e.g., the abstract semantics).
* Each requirement should include an evaluation as to whether it is addressed by SysML v1, or whether it is a significant modification of an existing concept, or a new concept.
* If the requirement is a significant modification to SysML v1 or is a new requirement not addressed directly by SysML v1, the requirement should refer to source material which identifies the issue and/or need the requirement is intended to address. This may include a concept in the SEBoK, or an issue with current SysML modelling. In some cases, the issue may be associated with a usability issue related to productivity or understandability rather than a capability that cannot be modeled.
* Include reference to compelling examples
* Cross check the requirements against the original requirements SysML v1 requirements which are contained in the [UML for SE RFP](http://syseng.omg.org/UML_for_SE_RFP.htm) (<http://syseng.omg.org/UML_for_SE_RFP.htm>)

*Service requirements guidelines*

* Specify the function that a user or external software application requests to operate on the model
* The inputs and outputs of the function should be consistent with the SECM terms and definitions
* Define pre-conditions and post-conditions that reflect the state of the model prior to and after the function is performed
* Include reference to how this function supports a system modeling tasks where practical

**Action:**Sandy and John to provide an example of a data model requirement and a service requirement

**Action:** Robert to generate SysML v2 RFP requirements in web viewable format

**Action:** Robert to provide a link to OpenMBEE web viewable document with usage instructions

**Specific requirements.** The following summarize the specific requirements that were presented and discussed. The requirements are addressed in the following order:

* Formalism requirements
* Data model requirements
* Requirements & verification
* Analysis
* Properties and expressions
* Structure
* Interfaces
* Behavior
* Common core
* API requirements
* Service requirements
* Model construction
* Model visualization
* Model analysis
* Model management
* Workflow & collaboration
* UML interoperability
* Other requirements
* Usability

**Formalism requirements**. Jonathan Patrick and Conrad Bock [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) the formalism requirements which can be found under the presentation section of the Formalism WG site at <http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:sysml_v2_model_formalism_working_group>

The presentation included an introduction to language definition, the formalism requirements and benefits, and some additional requirements under consideration.

The formalism defines how the language is defined in terms of its syntax, semantics, vocabulary, and interchange format. The abstract syntax specifies the grammar including the basic constructs of the language analogous to verbs and nouns, and the rules for constructing legal sentences/statements. The concrete syntax specifies the symbols for these constructs and how they can be presented. The interchange/API specifies how computers read and write the language. The vocabulary defines the terms that reflect specific uses of the constructs defined in the abstract syntax that are captured in model libraries, analogous to a dictionary of terms. The semantics specify the meaning of the terms in the domain that is being represented.

The formalism is intended to enable the language specification to express the domain concepts that can be interpreted unambiguously by humans and computers. This requires a uniform syntactic and semantic interpretation to ensure consistent use of terminology and rules of grammar (well formedness), and consistent interpretation of meaning. The domain concepts and associated semantics should be defined in the language to support the analysis needed to answer questions related to system specification, design, and verification.

There are currently 8 requirements. The first 7 requirements were presented at the previous meeting, and one requirement has been added since then. The first requirement is that ‘SysML 2.0 shall have a declarative semantics expressed in mathematical logic and/or other semantics with a translation to declarative semantics in mathematical logic.’ The newly added requirement is ‘SysML 2.0 syntax shall be specified in a subset of SysML 2.0.’ This requirement is similar to specifying MOF as a subset of UML. The requirements and their benefits can be found in Jonathan’s presentation.

The general approach that is being proposed is that the language is defined in terms of a small set of terms and their base semantics, and that model libraries are used to extend the terms and associated semantics to reflect the domain concepts. A control node used in activity diagram serves as an example to illustrate the approach. Currently, each type of control node such as a fork node, join node, decision node, and merge node is defined with their own unique semantics. In SysML v2, the general concept of a control node may be specified along with its semantics. The specific control nodes for a fork node, join node, decision node, and merge node would be specified in a model library, with semantics that are specialized from the control node semantics. The language would include the meta semantics for specializing semantics.

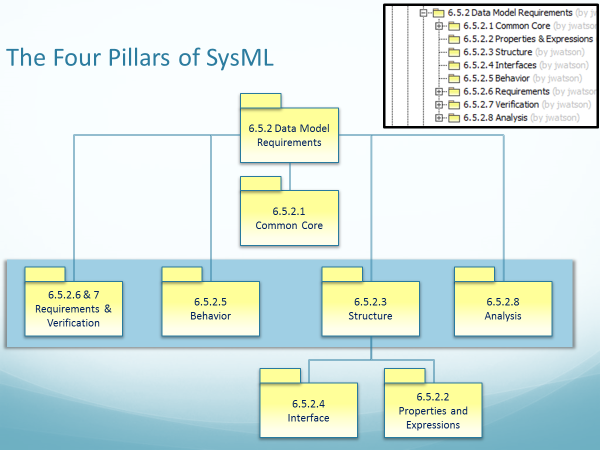
Sandy raised a question as to whether requiring a textual syntax for SysML v2 is a formalism requirement, and whether the consistent use of logic expressions is a formalism requirement.

**Action.** Jonathan to update the formalism requirements in the model

**Action**. Jonathan to consider adding a requirement to provide a textual concrete syntax

**Action.** Jonathan to consider adding a requirement to apply logic consistently throughout language

**System Engineering Concept Model (SECM) and data model requirements.** John [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) an overview of the SECM status and associated data model requirements as shown in the figure below. This set of system modeling concepts is similar in scope to SysML v1, which includes support for modeling structure, behavior, parametric, and requirements, often referred to as the 4 pillars of SysML. The SysML v2 concepts also include additional concepts related to verification, analysis, and other concepts beyond what is in SysML v1. A summary of the data model requirements is included below.



**Figure 3.** Organization of SysML v2 Modeling Concepts

A major emphasis for SysML v2 is to ensure improved integration of the domain concepts in part by applying consistent patterns to represent these concepts. The application of patterns can specialize a core set of concepts and associated semantics in a consistent way. For example, the concepts of precedence can be applied consistently in different behavior representations (e.g., activities, state machines, sequence diagrams), and decomposition can be performed consistently for behavior and structure. Logic expressions such as AND, OR, NOR, and NOT can also be applied consistently throughout the language to facilitate reasoning about the system model.

**Requirements and verification concepts.** Sandy [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) the requirements and verification concepts that were developed by the SysML v2 requirements working group. This work builds on the work performed by the requirements working group for the SysML v1.5 RTF that was led by Rick Steiner. In SysML v1.5, the SysML requirement was refactored to enable both text based requirements, and more precise statements of requirements, commonly referred to as property based requirements.

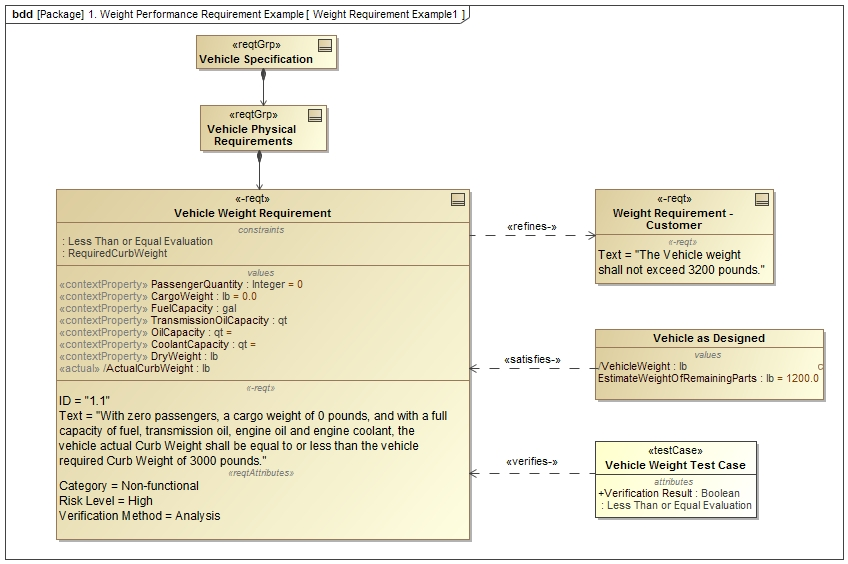
SysML v2, is intended to represent property-based requirements by enabling requirements to contain formal expressions that constrain property values, such as weightActual<1000 kilograms. More complex expressions can also be specified to impose constraints on design solutions such as a property based requirement for vehicle stopping distance as a function of speed and road conditions. The SysML v2 requirement will contain the formal expressions, the properties that are being constrained, along with the more traditional text statement. In addition, the SysML v2 requirement includes the ability to specify assumptions, such as whether the weight of oil, gas, and other fluids are included in the vehicle weight requirement. The assumptions are represented by context properties that can have values assigned or constrained.

SysML v2 also supports the concept for restricted natural language text to specify requirements in natural language more precisely. Different approaches that were introduced by Tao Yue and Shaukat Ali as part of the requirements working group show considerable promise.

In addition to the property based requirements, SysML introduces the concept of requirement group, which is a container for requirements. The requirement group is not a shall statement, but can include additional information that provides context for the requirements contained in the group.

The requirement also includes standard requirement attributes and requirement categories based on the INCOSE Handbook. Bertil Muth also reviewed the attributes, categories, and other requirement concepts to improve alignment between SysML v2 and the ReqIF requirements interchange standard.

The following weight requirement illustrates some of the requirements concepts describe above including the requirement group, property based requirements, requirement attributes and categories.



**Figure 4.** Example Weight Requirement

SysML v2 supports the concepts of requirement definition and usage to facilitate reuse of requirements. A requirement may be contained in more than one requirement group, or a generic requirement can be defined and then specialized for specific usages.

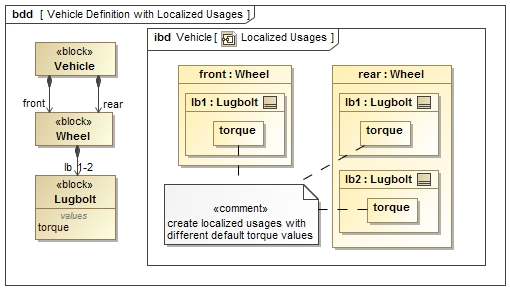
Several verification concepts are intended to integrate with the requirements concepts in SysML v2. Brian Selvy and David Haines identified core concepts to integrate with SysML v2 requirements including verification case, objectives, criteria, unit under verification, verification systems, verification data collection and evaluation activities. and verification results.

The preliminary requirements that support the above Requirement and Verification concepts are included in the presentation.

**Analysis concepts.** Manas Bajaj [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) the analysis concepts to be incorporated into SysML v2. Although SysML v1 includes parametrics, it lacks some essential concepts to facilitate analysis. These concepts include similar concepts to verification, such as analysis case, objectives, criteria, scenarios, unit under analysis, analysis model, analysis run, analysis evaluation activities. and analysis results. The initial concepts and requirements are reflected in the presentation.

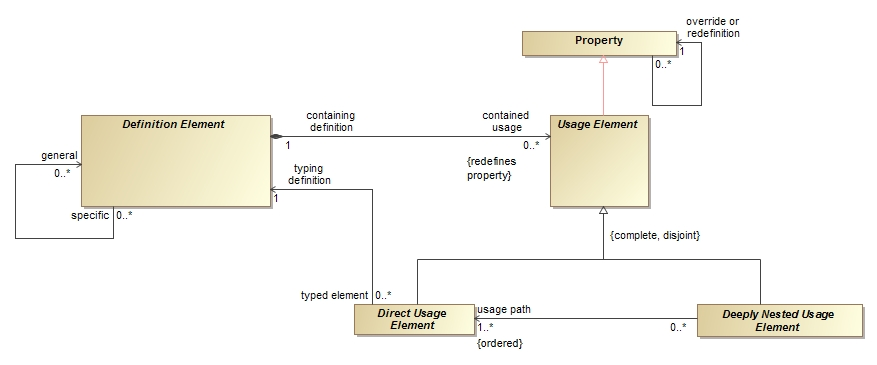
**Action:** Manas, Sandy, and John to review and update analysis pattern and associated analysis concepts

**Structure modeling concepts.** Hans Peter de Koning [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) some of the structure modeling concepts, and associated requirements.These concepts are intended to facilitate modeling deeply nested structures that represent system design configurations. The structure modeling builds on SysML v1 concepts of definition and usage (e.g., blocks and parts), but adds the concept of a deeply nested part. The issue with SysML v1 is highlighted in the figure below showing a deeply nested structure of Lugbolts that are part of Wheels that are part of a Vehicle. The challenge in SysML v1 is the inability to easily represent the structures corresponding to specific design configurations with localized values as indicated in the ibd in the figure. For example, the 2nd lugbolt on the rear wheel should with a specific torque value should be clearly identifiable in the browser. This can be done in SysML v1 but requires considerable effort and knowledge to apply redefinitions, subsetting, bound references, and other advanced features of SysML.



**Figure 5.** Specifying an unambiguous system design configuration

SysML v2 is intended to facilitate support for modeling system design configurations and individual systems. A composite structure can be defined in a definition model that captures typical variability in the structure such as part multiplicities and components with different subclasses. A product structure is defined that removes the variability in the definition model which corresponds to a specific design configuration. The design configuration can include property values that over-ride the values of the definition model and add features by further specializing the components in the definition model. Finally, an individual model specifies the structure of an individual system, such as an as-built system on the factory floor. The individual system can include property values corresponding to measured values that further over-ride the property values of the design configuration. The individual model has its own lifetime where its property values can continue to change over time. Although these concepts can be reflected in SysML v1 models, they often require considerable effort involving more advanced features of the language. The intent is to represent these concepts in SysML v2 in a more straight forward manner. A fragment of the structure concepts model is included below.



**Figure 6.** Sample of structure modeling concepts.

This same set of concepts can be applied to any decomposition, such as decomposition of behavior (e.g., functional decomposition).

There was considerable discussion on how to specify the requirements for structure modeling to avoid over-constraining the solution. A proposed starting point for the requirements is included below. The intent is to augment the requirements with a concept model, and to identify the SysML v1 limitations as described in this section.

***Proposed structure modeling requirements.***

**Modular unit of structure.** SysML v2 shall represent modular units of structure called Definition Elements that contain features as specified below.

* Value properties
* Ports
* Constraints
* Behavioral features
* Other (TBD)

**Note:** Each of the features is specified in other requirements.

**Reusing definition elements in a structural context.** SysML v2 shall represent usages of Definition Elements called Usage Elements in deeply nested structural contexts, and be capable of over-riding its feature multiplicity, values, and types, and adding features to this localized usage.

**Unambiguous configuration definition.** SysML v2 shall represent system design configurations with composite structures with uniquely identified Element Usages in which any variability in the structure has been resolved.

**Modeling individual systems.** SysML v2 shall represent an individual system that conforms to a system design configuration that can be uniquely identified in the real world, such as with a serial number. The individual system can override the values of the system design configuration with its measured values.

**Action:** Hans Peter to update the structure modeling requirements.

**Property and expressions.** The foundation concepts in SysML v1 for specifying quantitative characteristics and supporting engineering analysis are value properties and value types, the usage of constraint blocks that capture reusable equations and constraint parameters, and a rich non-normative model for quantities and units. SysML v2 will extend these concepts to include more robust set of value types such as arrays and vectors, and concepts related to coordinate systems and geometric shape. SysML v2 also provides improved support for probability concepts, quantities and units, and standard expression languages that can be applied to any property.

A carry over from a previous discussion centered around the requirements for defining mathematical expressions in SysML v2.   The level of support could include some combination of standardizing on a signature for the expression with its inputs and outputs, specifying full expression trees with operators and operands, and specifying one or more default standard expression languages.  SysML could also require a platform independent expression language along with platform specific bindings, to prevent SysML v2 from being locked into a particular technology.

**Action:** Hans Peter to update requirements for properties and expressions.

**Action:** Hans Peter and Manas to propose requirements for standard expression languages

**Interfaces.** Marc Sarrel [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) the interface concepts that he and Steve Hetfield have been working on. Marc identified several goals for interface modeling to model a diverse range of interfaces (e.g., electrical, mechanical, software, user). The core concept of an interface includes 2 interface ends and an interface connection as indicated below.



**Figure 7.** SysML v2 interface concept

The concepts of definition and usage apply to the interface ends and interface connection consistent with SysML v1. These concepts are being aligned with the structure modeling concepts described above to allow multiple levels of nested interfaces. SysML v2 also requires support for layered interfaces such as when modeling an OSI stack with application layers and physical layers. Another important concept is an interface agreement, which can be used to specify and constrain the interface ends. This interface agreement can specify constraints on information flow between peer layers of a protocol stack, and specify physical constraints between physical interface layers such as torque and angular rate constraints for a motor interface.

One of the key issues with SysML v1 is the inability to readily support views of different abstraction levels of complex interfaces. A desired view of a detailed multi-layer interface may show one layer of a stack and hide others, or highly detailed pinouts between physical connectors. This concern is being addressed by the visualization concepts and requirements to facilitate different viewpoints.

Brittany Friedland also [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) a usability issue with modeling connections between nested ports, which is highlighted in her presentation. A combination of features in SysML v1 including ‘onPort’ and ‘nested connector ends’ partially address this issue, but do not address some of the usability concerns.

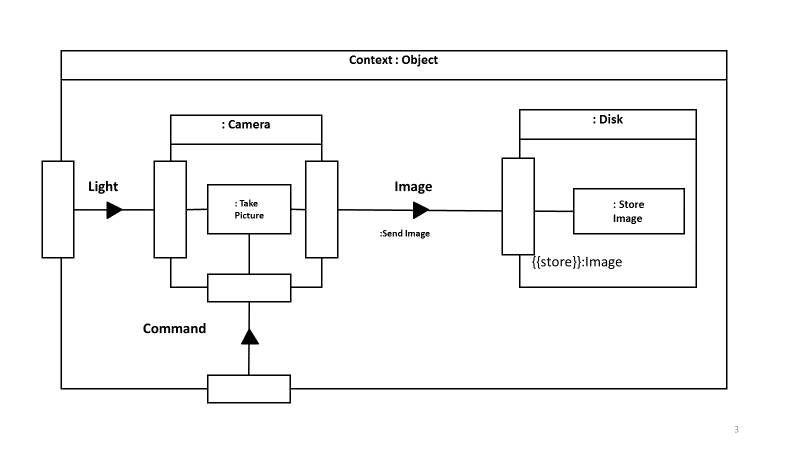
**Action:** Marc to update the requirements for modeling interfaces.

**Action:** Marc to refer to signal flow and physical interaction profile in the RFP as a reference

**Action:** Marc to resolve whether interface layer should be generalized to layer

**Behavior.** The SysML v2 RFP WG has not begun developing concepts for behavior modeling. The intent was to define and integrate the structure modeling concepts and interface modeling concepts prior to focusing on the behavior modeling concepts. Given the progress achieved with the structure and interface modeling per above, the development of the behavior modeling concepts can now begin.

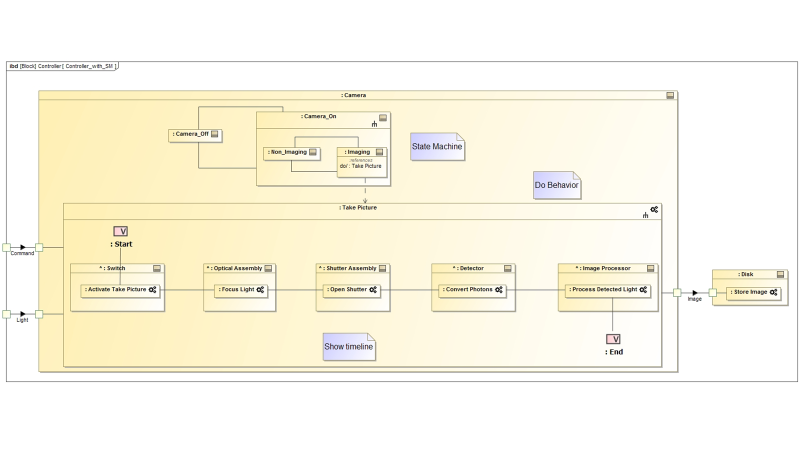
During this meeting, Sandy led a discussion to identify behavior modeling issues. To motivate the discussion, he showed a simple example of what might be expected from a behavior model in SysML v2.



**Figure 8.** Integrating Take Picture function with Structure

In this example, the Camera performs the Take Picture function that accepts Light as an input and produces an Image as an output. The Image is then input to the Store Image function of the Disk. The key requirement is that the behavior and structure should be seamlessly integrated including function inputs and outputs, and the interface between the Camera and Disk.

In the following figure, the Take Picture function is further decomposed. The decomposed function shows what structural elements perform each function similar to an activity diagram with swim lanes. The Take Picture function is also seen to be a do behavior of the imaging state of the Camera state machine.



**Figure 9.** Decomposition of Take Picture function

This example highlights some of the key challenges associated with integrating structure and behavior models in SysML v1. During the discussion, the following SysML v1 issues were identified, and should be addressed by SysML v2 behavior models:

* Lack of integration between structure and behavior (e.g., flows, triggers, input to do/behavior)
* Limitations of executing behavior that is integrated with structure/ports
* Lack of integration between different kinds of behavior (seq, act, stm, uc, timing)
* Lack of integration with analysis expressions so that it is simple to create an expression anywhere in the model
* Unification of expressions and constraints
* Inability to represent structure of I/O and how they are input to different functions
* Lack of representation of timelines
* Inability to easily create and destroy relationships such as connectors
* More flexible pre/post conditions using bdd/instance diagrams (refer to Keith Butler presentation)
* Support for software architecture concepts as identified by Ron Townsend

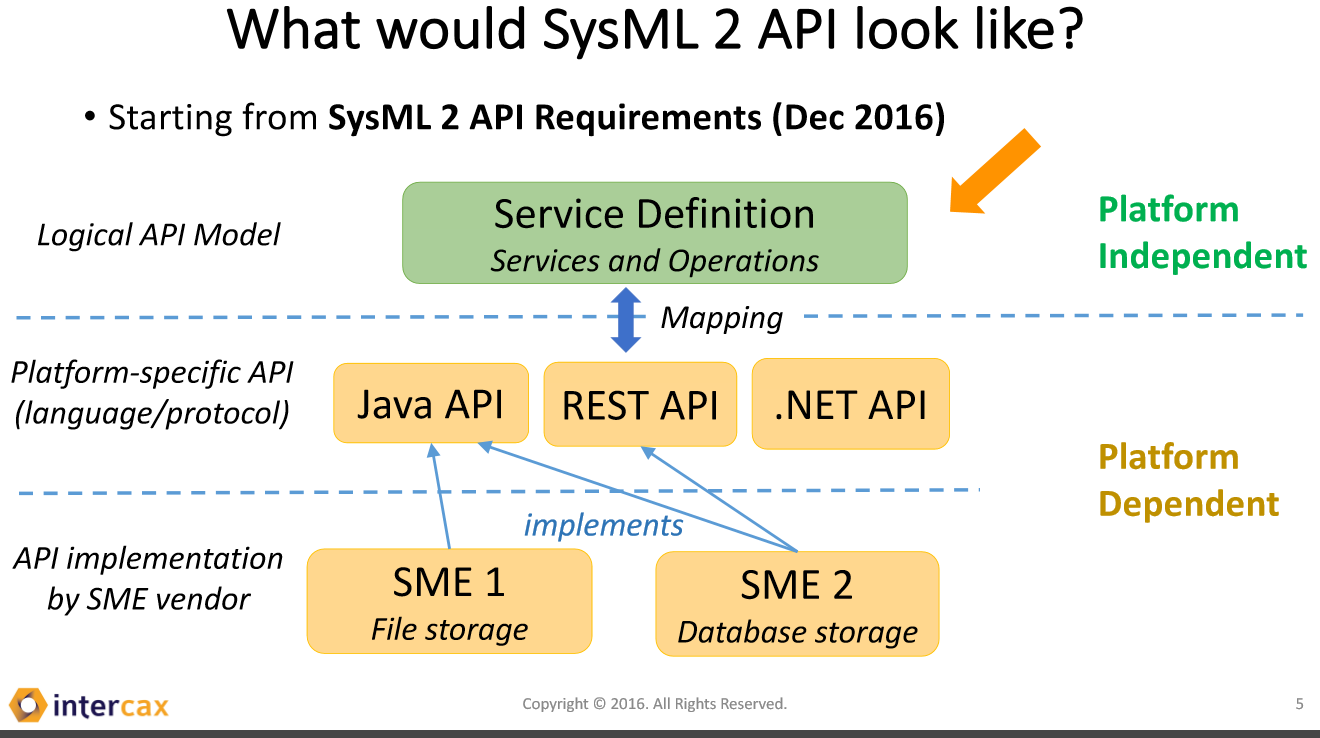
**Action:** Sandy to initiate the Behavior modeling working group to update the issues and compelling example and develop the behavior concepts and requirements

**Common core concepts.** During this meeting, we did not discuss the common core concepts. However, there has been some initial work to identify and refactor concepts that span the model. These include concepts common to all model elements and their relationships. This will be a topic of future discussion and review.

**Action:** John Watson to prepare a draft of the common core concepts to present for review at an upcoming telecon.

**API model.** The SysML v2 RFP requires a standard API model to support interoperability. SysML v2 should allow any client such as an external tool, plugin, or user interface to be able to request services that access the system model repository in a standard way. Manas Bajaj [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) an overview of the approach to develop an API model, and provided a prototype demonstration of the approach. The goal was to help validate the API requirements, and ensure a common understanding of what is expected of a submitter.

As highlighted in the figure below, the API requirement is to produce a platform independent model (i.e., logical API model) with selected platform specific bindings that are then implemented by modeling tools to enable access to the system model repository. The platform independent model provides a service definition that is consistent with the information model. The platform specific binding defines the service using a particular technology (e.g., java, web services). A formal mapping is maintained between the platform independent model and platform specific bindings.

vendors.

**Figure 10.** SysML v2 API specification approach

Manas also demonstrated this approach by prototyping an API for the Analysis Setup service with different platform specific bindings that read and write to different model repositories. Some additional services that were identified during the discussion included a standard way to connect to the server and a standard discovery service.

**Action:** Manas to include a requirement for a standard way to connect to the server and a standard discovery service.

**Action:** Manas to update the API Requirements based on the findings from the demonstration.

**Action:** Manas to consider prototyping a more complex service to query the model and perform an engineering analysis to reason about the system

**Action*:*** Manas and Ed to clarify whether the platform independent model defines operations on the meta-classes in the SysML metamodel (Note: the metamodel is developed by the submitters)

**Service requirements.** The following section highlights the concepts and initial requirements that facilitate the following system modeling capabilities:

* Model construction
* Model visualization
* Model analysis
* Model management
* Workflow and collaboration

**Note:** It was agreed that the service requirements will include pre and post conditions that refer to glossary terms. The glossary terms are preferably used in the concept model.

**Model construction services.** Ron Williamson presented the requirements for model constructions services. The model construction concept and requirements are captured on the Model Construction Wiki at <http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:systems_engineering_model_construction_focus_area>

The model construction concept is intended to provide a more efficient and intuitive means for model development by users with a diverse range of experience and needs. The requirements reflect the need to create, update, and delete any model element or group of model elements using textual, tabular, or graphical entry. In addition, these requirements are extended to apply to other model constructs beyond model elements that include metadata, modeling patterns, model queries, expressions, viewpoint methods, model transformations, and external links. In addition, model construction is intended to be facilitated by workflow execution mechanisms that enable a user to perform a specific systems engineering task or practice.

Some of the feedback included.

* A core requirement is to create, update, and delete a data set with 1 or more data elements
* Replace the service requirement to import and transform multiple elements in batch mode with a transaction service. This includes a requirement to read a list of structured data elements and its associated schema, and another requirement to transform the structured data to model elements.
* When creating an element or other model construct, create the element id (i.e., UUID)
* When transforming structured data, optionally set the id depending on whether one is specified
* When deleting an element, ensure the final state is consistent with the deletion semantics.
* When deleting an element, retain and reserve its uuid. (This may be a model management function)
* Add a service requirement to apply a pattern.
* Consider how to construct models through elaboration and refinement to transition from one level of abstraction to another, while preserving the earlier abstraction. (Note: this may be considered a transformation of one abstraction level to another that can be viewed in different viewpoints.)
* Consider usability issues that were identified in the usability discussion below

**Action:** Ron to update model construction service requirements with pre and post conditions. Send requirements to Geoffrey Biggs and Andy Ko for review.

**Model visualization services.** Chris Schreiber presented the requirements for model visualization services. The model visualization concept and requirements are captured on the Model Visualization Wiki at <http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:model_visualization_working_group>

The visualization concept reflects a model view controller paradigm where the controller is the viewpoint method. A primary driving service requirement is to create, read, update, delete, and execute viewpoints. Executing the viewpoint method includes querying the model and rendering the query results in a view. This concept supports dynamic visualization, semantic zoom and pan, diagram differencing, and the ability to provide diverse renderings including graphical, tabular, text, and geometric views of the model. The visualization requirements also include the requirement to map the model elements to the visualization elements and provide support for domain specific symbol libraries.

SysML v2 is required to support the 9 standard SysML v1 diagrams and a geometric view. In addition, the viewpoint method is required to be highly flexible to address a diverse range of systems engineering visualization needs. The views must also be capable of being stored in various formats, such as in standard document formats.

**Action:** Chris to update model visualization service requirements with pre and post conditions

**Model analysis services.** Manas Bajaj [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) the requirements for model analysis services. The goal is to ensure that analysis can be performed on the system model as seamless as practical. The services are intended to support full lifecycle analysis that includes both quantitative and qualitative (logical) analysis. The model analysis concept and requirements are captured on the Model Analysis Wiki at

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

The primary analysis service requirements are to Setup, Execute, Visualize, and Store the analysis, Track analysis decisions, and Query the analysis. Manas also identified several other requirements needed to support analysis, which should be covered in other areas such as visualization, model management, and properties and expressions. This includes the need for representing uuid, versioning, data structures, units and quantities, operators and functions, geometry, and model transformations.

**Action:** Manas to update analysis service requirements with pre and post conditions

**Model management services.** Laura Hart [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) the requirements for model management services for SysML v2. The goal is to manage the configuration of the system model and control the process to change the system model. In addition to the system model, the scope of model management encompasses other model constructs that are used to develop and maintain the system model, including links and transformations to other models and external data. The model management concept is highlighted in the figure below, and is elaborated on the Model Lifecycle Management Wiki at <http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:model_lifecycle_management_working_group>



**Figure 11.** Integrated System Model (ISM) Lifecycle Management

A key requirement is for each model element to have a unique identifier, and for model elements to be capable of being versioned. The Universal Unique Identifier (UUID) was proposed as a potential standard for a globally unique id. There is still discussion on how to specify the versioning requirement, and other required metadata such as timestamp and its context (e.g., when created, changed, approved). This working group will continue to identify other essential metadata to support model management as required, such as whether to include metadata regarding tool versions along with the applicable model or data source that the SysML model links to. For SysML v2, there is a requirement for the metadata to be extensible. There is also a requirement to interchange this metadata in a standard way. The overarching intent for the model management requirements are to identify what services and metadata are required to manage the system model, without over constraining where this metadata is stored, how it stored, and whether the metadata is part the SysML model.

The primary model management services are to create versions, create baseline configuration, log changes, compare differences, generate version histories, manage data protection controls such as data rights and markings, and manage user authorizations to the data. Additional services are required to support extending the metadata.

The following Issues still need to be resolved:

* Should the identifier be globally unique?
* Should a standard unique identify such as UUID be required?
* What other metadata is required (e.g., tool version and its context)?
* Should create branch, merge (including rebase) be included as service
* Define each critical term used including configuration and baseline

**Action:** Laura to update model management metadata and service requirements with pre and post conditions

**Action:** Laura will update the model management concept diagrams to reflect the above

**Workflow and collaboration services.** Hedley Apperly presented the requirements for workflow and collaboration services for SysML v2. The workflow and collaboration concept is elaborated on the Workflow and Collaboration Wiki at <http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:mbse_workflow_and_collaboration_working_group>

The Workflow and Collaboration concept includes an overarching workflow that begins with developing the practices and capturing them in a master practices repository. These practices are tailored to a particular project, and used as an input for project planning, execution, and monitoring. As part of the planning process, the tasks are assigned to roles, and roles are assigned to individuals to perform the task.

The minimum requirements for SysML v2 are to provide services that support integration with external process definition and workflow execution engines. In particular, SysML v2 should enable a modeler to understand the task and role definition in the context of a workflow, accept notification to begin a task, accept some level of guidance on the execution of the task, and provide status/metrics on the performance of the task. The specific workflow and collaboration requirements for SysML v2 are to:

* assign roles to a task
* assign users/engineers to roles
* notify user of task (e.g., start date, completion date)
* identify the model scope to be reviewed and/or updated by this task (i.e., a task change set) - optional
* provide access permissions to the model scope - optional
* provide task guidance (e.g. enable execution of practice from the practice repository that may include use of patterns, model libraries, wizards, etc.)
* provide user task status and metrics

**Action:** Hedley to confirm the following minimum required workflow and collaboration metadata/concepts to support the above service requirements: [User, Role, Task, Start/End Date, Model Scope, Metric]

**Action:** Hedley to update the workflow and collaboration service requirements with pre and post conditions

**UML interoperability requirements.** Ed Seidewitz [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) some of his initial thoughts regarding UML interoperability requirements for SysML v2. SysML v1 is a profile of UML, so that the integration between SysML and UML can be accomplished through their common syntax and semantics. In order to preserve this for SysML v2, Ed proposed that SysML v2 contain a subset that has equivalent semantics to a correspondingly identified subset of UML. He proposed that this subset be sufficient to model software architectures. Further discussion on this topic is needed to ensure SysML v2 can be used to specify software designs, and that UML models can interpret a specification in a SysML v2 model. (Note. Sandy suggests starting with a minimum requirement to specify blocks with interfaces and state machines.)

**Additional requirements.** Some additional critical SysML v2 requirements that were not discussed include:

* SysML v2 compatibility with SysML v1
* Storage and interchange requirements
* Conformance levels
* Integration with safety, reliability and security requirements.

In addition, there was a brief discussion on the requirement for a common approach for execution services to be applied to execute the following:

* Query
* Analysis model/expressions
* Viewpoint method
* Transformation

**Action:** John Watson to capture other requirements in the SysML v2 RFP draft for review.

**Action:** Goeffrey Biggs to propose key concepts and requirements to integrate SysML v2 with the safety and reliability, and security (TBD) requirements. Initial discussions focused on including the cause-effect relationship in SysML v2.

**Usability requirements.** A facilitated working session was conducted on the first morning of the meeting (March 21) to discuss the usability requirements for SysML v2. Sandy opened the discussion with an introductory [presentation](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations), followed by a [presentation](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) by Bertil Muth, and then a [presentation](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations#presentations) by Tricia Ngoon. Considerable feedback was provided throughout.

Usability is generally viewed as the purview of the tool vendors. However, usability is a critical driver to the ultimate acceptance and use of SysML v2, and is fundamental to enabling more effective application of MBSE. As a result, the SysML v2 RFP must promote the need to address usability concerns.

The general approach that was identified is to establish usability criteria that can be included in the SysML v2 RFP. Submitters should reflect the criteria in the SysML v2 specification, including non-normative usability guidelines, and identify assessment methods for tool vendors to demonstrate how well their tool satisfies the criteria. There are various standards for usability and human-machine interaction standards within ISO (e,g,, ISO 25062, ISO 9231) and other standards bodies that can be leveraged to identify the relevant criteria.

A key point that Bertil raised is that we must ensure the model can address different classes of users, and that each class of user has its own concerns. A new modeler, versus an occasional modeler, versus an experienced modeler, versus a consumer of model information all have unique needs. In addition, the needs shift with the phase of development, the engineering discipline, task complexity, and other factors. The criteria noted above should be mapped to different classes of users and their needs. The criteria must enable SysML v2 models to be understandable and engaging for different classes of users, and be capable of being constructed efficiently.

Sandy noted some SysML v1 usability issues that included:

* Learning obstacles
  + Definition/Usage (e.g., part vs block, action vs activity, ..)
  + Maintaining consistency across views
    - different kinds of behavior, structure, parametrics, requirements
  + Model containment tree vs product structure
  + Unfamiliar terminology from UML legacy and SysML (e.g., proxy port)
  + Functional vs geometric concepts (*enclosed by* versus *external to*)
  + Lack of engaging/intuitive *domain-specific* visualizations (i.e., boxology)
  + Inability to express informal ideas and refine
  + Modeler ability to abstract
* Productivity obstacles in model construction
  + Complexity of user interface
  + Create/update one element at a time (too many clicks)
  + Manual diagram layout

Based on the discussion, some key aspects of the SysML v2 specification that can address usability concerns include:

* Model construction services (e.g., use of patterns)
* Model visualization services (e.g., flexible view/viewpoint)
* Customization & extensibility capabilities (use of aliases, different native languages, domain specific symbol libraries concepts)
* Consistency across language (e.g., terminology, symbols, concepts)
* Consistent workflows, modeling patterns, and user interfaces
* Ability to quickly find relevant information within the current view
* Ability to quickly navigate to related information that is not in the current view
* Setting defaults to support common uses (e.g., default multiplicity of 1)

Alan Moore distinguished the need to make the language useful, which at times, may not imply the language is easy to use. This emphasizes the point that there is inherent complexity in the systems we model, and ultimately the usefulness of the model must be weighed against the effort required to model (i.e., return on investment).

Tricia highlighted the user centered design approach that includes storyboarding, prototyping, and applying usability heuristics and criteria.

**Action:** Identify potential usability criteria to incorporate into SysML v2 RFP

**Declarative behavior specification.** Keith Butler [presented](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_march_2017_meeting_presentations" \l "presentations) an approach he and others have developed to specify a behavior using a declarative behavior specification with a start state and goal state. He provided a simple example for a buyer/seller transaction. At the beginning of the transaction, the seller owns the item and the buyer owns the money. At the end of the transaction, the buyer owns the item and the seller owns the money. This behavior can be specified by a class diagram prior to the transaction and a class diagram after the transaction showing the ownership relationships. It was noted that the class diagrams (or perhaps instance diagrams) can be viewed as pre and post conditions. The major difference is that pre and post conditions are specified as constraints. SysML v2 should enable this declarative specification by specifying the pre and post conditions as constraints on the ownership relationships. In addition, the behavior should be able to create and destroy the relationships.

**SysML v2 RFP planning.** Sandy reviewed the milestones for development of the SysML v2 RFP. Based on the work to be performed, Sandy proposed moving the RFP submission date out one meeting cycle from September to December, 2017. The proposed updated plan is included below.

Aug 2015 Driving Requirements (INCOSE MBSE Themed Insight Article)

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

Dec 2016 SME Concept (INCOSE INSIGHT Article)

Jan 2017 Presentation at INCOSE IW

Mar 2017 Initial Draft Requirements (SECM, API, Formalism)

June 2017 Initial Draft RFP for Review

Sept 2017 Updated Draft RFP and Presentation to ADTF

Dec 2017 Reviews Complete

Dec 2017 Issue SysML v2 RFP

Dec 2017 Form SysML v2 Submission Teams

The goal will be to prepare a draft RFP for review with the requirements included by the next OMG meeting in Brussels on June 5, 2017. The focus will be to develop the behavior modeling concepts and requirements, and to refine the requirements, and integrate them into the RFP for review following the next OMG meeting. The SysML v2 RFP WG leads will be requested to refine their requirements using the guidelines provided to ensure precise, complete, and traceable requirements.

**SysML v2 RFP Working Group agenda for Brussels meeting.** The SysML v2 RFP Working Group meeting will be held in Brussels, Belgium on Tuesday, June 6 and 8. The goal will be to review the RFP requirements and conduct additional working sessions to address specific issues.

**Action.** Sandy to draft agenda for the SysML v2 RFP WG meeting in Brussels the week of June 5.