***PLEASE REFER TO FEEDBACK FROM CONCEPT LEAD PRESENTATIONS STARTING ON PAGE 7***

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

The following is a summary and next steps from our full day face-to-face Working Group meeting at the Cambridge OMG meeting on September 24, 2015. The attached presentation slides from the meeting will be posted along with this summary to a link off of the WG Website at

<http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:cambridge_sep_2015_meeting_presentations> .

**FOLLOW-UP ACTIONS**

* Eldad to post meeting summary and presentation slides in the attached to the WG Wiki.
* All WG members are requested to provide feedback to Sandy by Oct 16 regarding the initial concepts
* Sandy to consolidate the feedback and make available to WG and concept leads
* Eldad to schedule telecons for concept leads to present/review individual concepts
* Concept leads to refine concepts based on feedback and include the following:
	+ Driving Requirement (refer to extract below)
	+ Limitations of current SysML modeling capability
	+ Proposed SME concept
		- Primary functions
		- Identification of alternative approaches
		- Simple example
* Concept leads to initiate a Wiki page for their concept
* Sandy & Eldad to propose agenda for 2 day WG meeting (Tuesday and Thursday) the week of December 7, 2015 at the OMG meeting in La Jolla

**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 tools needed to support model-based systems engineering (MBSE). The SysML v2 requirements will be derived from these higher level 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 (Note: these are included in the extract from the article below). The publishing of this article serves as an initial baseline and is an important milestone for SysML v2 development.

The objectives for our face to face meeting were to begin to define the concepts for each of the capabilities and associated driving requirements. The agenda included the following:

**MEETING AGENDA**
09:00 - 09:15 Introduction - Sandy Friedenthal
09:15 - 09:30 SE Use Cases - John Watson
09:30 - 10:00 Systems Engineering Concept Model (SECM) - Domain **[R1]** - John Watson
10:00 - 10:30 Systems Engineering Concept Model (SECM) - Kernel **[R1]**  - Charles (Chas) Galey
10:30 - 10:45 Break
10:45 - 11:15 Modeling Formalism **[R2]** - Yves Bernard
11:15 - 11:45 Analysis Concept **[R2]** - Manas Bajaj
11:45 - 12:00 Reserve
12:00 - 13:00 Lunch
13:00 - 13:30 Model Visualization **[R3]** - Chris Schreiber, Josh Feingeld
13:30 - 14:00 Model Construction **[R4]** - Ron Williamson
14:00 - 14:30 Model Interoperability **[R5]**and Standard API **[R6] -** Axel Reichwein
14:30 - 15:00 Break
15:00 - 15:30 Model Management **[R7]**- Laura Hart
15:30 - 16:00 Integration with PLM **[R7]**  - Uwe Kaufmann/Michael Pfenning
16:00 - 16:45 ESA System Modeling Environments - Harald Eisenmann/Hans-Peter deKonning
16:45 - 17:15 Model Execution Approach and Issues **[R2]** - Robert Karban
17:15 - 17:30 Planning and Next Steps (e.g. Work Group Coordination, Communication, Wiki's, etc.)

Note: Chris Delp presented a status update on OpenMBEE at the SE DSIG meeting on Tuesday.

Each of the concept leads had limited time to present their concepts and progress, but this provided an opportunity to see how the different concepts are coming together. The presentations are included in the attached files.

All participants are requested to send their feedback to Sandy who will consolidate the feedback and distribute to the presentors and the rest of the WG. Feel free to send directly to the presentors as well. The concept leads can begin to refine their concepts, as we begin to integrate the concepts into a cohesive SME Concept. Our goal will be to present and begin integration of the updated concepts at the La Jolla OMG meeting the week of December 7, and then baseline the SME concept by the March 2016 OMG meeting in Reston.

Eldad and I will develop a proposed agenda for the next meeting in La Jolla the week of December 7, 2015. Our current intent is to schedule a two day meeting on Tuesday and Thursday of that week, which will focus on integrating the concepts into an overall SME Concept. The two figures below represent a starting point for an integrated SME Concept.



**Figure 1. System Modeling Environment-Logical Architecture**



**Figure 2. System Modeling Environment Layered Architecture**

**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

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 transformdata with other models and structured data
* **Manageable:** Ability toefficiently 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.

**Sanford Friedenthal Comments**

**MBSE Use Cases - John Watson**

a) show how the 7 SME Capabilities support the MBSE Use Cases

**Systems Engineering Concept Model - John Watson, Chas Galey**

a) Identify limitations of current SysML Concept Model

b) Define the language customization services (refer to SysML v2 services spreadsheet)

**Modeling Formalism Concept - Yves Bernard:**

*a) Add explicit reference to driving requirement #2.*

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.

a) identify what the current SysML formalism is and its limitations

b) elaborate the evaluation criteria related to ambiguity, conciseness, usability, availability of reasoners,

- Yves started this by defining precise, usable, efficient

c) Identify alternative formalisms such as fUML, predicate logic, temporal logic, and their pros and cons

- Yves started this with the fUML formalism

d) show an example of how the modeling formalism could be applied to assess the impact of a requirement change and an example of how it is applied to perform model checking, and how it addresses the above criteria

- Yves started this with the allocation and interruptible region examples

e) provide an example of how the formalism might be specified in the RFP

- Yves started this with the collaboration and mapping approach

f) define model analysis services related to model checking, query, and validation (refer to SysML v2 services spreadsheet)

**Analysis Concept - Manas Bajaj**

a) State need for SysML to support seamless integration with analysis.

b) identify limitations of current SysML analysis support

c) Clarify criteria for effective support for analysis (how do we measure this capability?)

d) Elaborate use case of how a systems engineer would apply SysML 2 to perform different kinds of analysis (trade study, sensitivity analysis, verification by analysis **with pbr**, ..) using built in solvers and external solvers. Also, show how different analysts would start with their analysis such as a free body diagram and use SysML 2 to integrate with the design model without having to modify their paradigm.

e) Add a need for SysML 2 to distinguish different probability distributions

- unit to unit variation (e.g., measuring the weight of 1000 different systems)

- parameter variation of a single unit due to environmental changes (e.g., the outside temperature may have a probability distribution causing a distribution on the length due to thermal expansion)

- other

d) in slide #2,

What is the difference between Data Structure and Type (isn't a 2D or 3D Vector a Type?)

Date is a measure of Time

Clarify Geographic map as a Type.

f) Include requirements for QUDV and expanded libraries including Imperial/English units

g Clarify how FMI fits in? Should SysML support FMI direclty?

h) Clarify how the System Analysis Metamodel would be leveraged by SysML v2 (part of the conceptual data model?)

i) Define the model analysis services (refer to SysML v2 services spreadsheet)

**Model Construction - Ron Williamson**

a) Identify limitations of current SysML construction?

- this is partially described on slide 15

b) refine the moe's for model construction

c) Include more emphasis on simpler model construction using Wiki style text and table entry.

d) Enable simpler model construction in early stages of design (e.g. connect parts on block diagram without concern for types).

e) Include requirement to transform/map input data into SysML elements

f) define the model construction services (refer to SysML v2 services spreadsheet)

**Model Visualization - Chris Schreiber, Josh Feingeld**

a) Slide 3 - R3.8 : Add text as another form of non-diagrammatic visualization

b) Identify limitations of current SysML visualization

c) define/refine the moe's for model visualization

d) Relate model view controller concept to current view and viewpoint concepts with the goal of supporting highly flexible view construction

e) Clarify that SysML v2 should include a standardized list of default views analagous to current SysML diagrams, but more robust (e.g. a graphical view of the block diagram that can be displayed in 2.5D with iconic representation and filter capability)

f) Include the need for a highly flexible graph visualization with filter and zoom capability such as displaying the electrical power interfaces, data interfaces, or thermal interfaces across the system

g) Define the model visualization services (refer to SysML v2 services spreadsheet)

**Model Management - Laura Hart**

a) Identify limitations of current SysML model management

b) define/refine the moe's for model management

c) Update slide on related initiatives to include OMG SysML v2, INCOSE Model Management WG, INCOSE German Chapter MBSE/PLM Integration WG

d) Define the model management services (refer to SysML v2 services spreadsheet)

**Model Management /PLM Integration - Uwe Kaufmann, Michael Pfennig**

a) Identify limitations of current SysML model management and PLM integration

b) define/refine the moe's for model management and PLM integration

c) define the metadata required for model management in support of MM requirements

d) Define the model management services in support of MM requirements (refer to SysML v2 services spreadsheet)

**Model Interoperability and Standard API - Axel Reichwein**

a) Identify limitations of current SysML model interoperability

b) define/refine the moe's for model interoperability

c Identify standards that should be supported for model execution interoperability (e.g., FMI)

d) Define the interoperability services (refer to SysML v2 services spreadsheet)

e) Assess the viability of defining an API that supports the services in the SysML v2 services spreadsheet

**Yves Bernard comments**

**MBSE Use Cases - John Watson**

1. The objective of this activity should be described. This will help in specifyingthe stop criteria for therefinement of the use case which is missing as well.

**Systems Engineering Concept Model - John Watson**

1. It’s not clear to me how we can use the UML4SE RFP model as it is to day since:
	1. it can hardly be mapped to the current SysML implementation (cf. diagram on slides 16 and 17 for instance).
	2. It’s not clearhow the SysML v2RFP model could “leverage” it since it’s more likely to be a complete reformulation of many of the SE concepts.
2. Clarify the respective roles/usages of UML4SE RFP model and Dave Oliver SE Conceptual model

**Concept Model Kernel–Chas Galey**

1. As described this “kernel” appears to be a metamodel for the definition of the domain which implies that it is not at the same metalevel. Therefore, the term of “kernel” (as opposed to “body”) is not convenient to describe it since implies something which is at the same metalevelthat the “body” around it.
2. Reference to metamodel levels and “libraries” are debatable and seem to convey implementation choices that are premature here. They should be removed.
3. Why does “Block” belong to the “Kernel” and not to the “Domain”? What the rationale for this choice? To me, and as it is today this “kernel model” provides more confusion than help.

**Analysis Concept - Manas Bajaj**

1. The concept of “sematic transformation” should be clarified since it can convey various (and opposed) meanings.
2. I agree that all these mathematical operation and geometry concepts can be useful but I’m not convinced they shall be part of SysML. They are other already existing language that are very convenient for them and could rather improve the SysML capabilities to integrate other languages. Otherwise the risk is to make SysML always bigger withoutbeing able to cover all the needs anyway.
3. I think need#5 is already possible with 1.4 thanks to the enhancement of the view and viewpoint concepts
4. Need#6 can easily be covered outside SysMLtoday. What would be theadded value to add this to SysML natively?
5. Need#7 relate to process modeling rather than to system modeling there are already language compatible with SysML (e.g. SPEM) which can address it. See also MARTE which provides means to add annotations to a design model in order to provide interfaces to specific analysis methods. Such facilities could be provided by normative or non-normative extensions.

**Model Construction - Ron Williamson**

1. “EADS” is now named “Airbus Group”
2. For “Capture metadata regarding update” and “Report differences” see also the TIWG stuff. The interest for these is not limited to SysML.
3. For “Non functional Requirements” see also MARTE

**Model Visualization - Chris Schreiber, Josh Feingeld**

1. “EADS” is now named “Airbus Group”

**Model interoperability & standard API Requirements -Axel Reichwein**

(no remark)

**Model Management - Laura Hart**

1. Clarify that MLM should provide means to identify (and then manage) information shared among models (either from the same domain or not) but that, by itself, the identification of such information is beyond the scope of MLM

**Model Management /PLM Integration - Uwe Kaufmann, Michael Pfennig**

(no remark)

**Emerging standard for MBSE environment – Harald Eisenmann**

(no remark)

**Experiences from Developing Concurrent Multi-Disciplinary MBSE - Hans Peter de Koning**

(no remark)

**Model-Based Engineering Environment for Model-Based Systems Engineering - Christopher L Delp**

(no remark)

**Experiences on building executable models, approach and issues – Robert Karban**

1. Difficult to follow without the discourse
2. Justification for some claims are not clear (e.g. “Cannot reference value properties”)
3. Improve precision and rigor in the expression of limitations and observations (last slide)

**Tao Yue and Shaukat Ali’s Comments**

**MBSE Use Cases - John Watson**

* Is there any intention to propose a use case specification/modeling methodology in SysML 2? If yes, Simula is very interested in contributing to this activity in the future. Proposing such a methodology is related to the capability of Model Construction and almost all of the key effectiveness measures.
* Is there any intention to visualize use case specifications as activity diagrams in SysML?
* In the literature, there are some works that focus on generating tests from requirements. Is there any intention to include some capability in SysML?

**Analysis Concept - Manas Bajaj**

* Is the WG planning to define a set of analyses to be enabled in SysML 2? Based on our experience that different analyses require different technologies such as solvers. Do we have any criterion defined to select which analyses to be enabled and which technologies to enroll? For example do we have a plan to integrate an OCL solver for enable e.g., test data generation? If yes, perhaps one of such criterion should be the time performance of such an OCL solver.
* Since Probability is mentioned in the requirements, perhaps we need to elaborate a bit more on this concept and how it will be potentially useful for enabling analyses. By following the same trend (Measurement), do we have a plan to include other types of measurements such as Ambiguity, and Vagueness (Fuzziness and NonSpecificity). If we go a bit out of this scope, perhaps we can consider “Uncertainty” as a modeling aspect of SysML 2. We define Uncertainty as a situation whereby a belief agent lacks confidence in a belief statement. We recently have done some work on understanding/classifying uncertainty in the context of Cyber Physical Systems. If anyone is interested, we can definitely provide the Technical Report and give a talk about it in the Dec. OMG meeting.