Evolving SysML and the System Modeling Environment to Support MBSE-Part 2

Sanford Friedenthal, safriedenthal@gmail.com

Introduction

This article describes a concept for a System Modeling Environment (SME) to support model-based systems engineering (MBSE), and is a follow-up to an earlier INCOSE INSIGHT article in August 2015 entitled "*Evolving SysML and the System Modeling Environment to Support MBSE* (Friedenthal and Burkhart 2015).*"*. The previous article defined the background, needs, and driving requirements for the SME. The development of the SME concept is part of an effort to derive requirements for the next generation of the Object Management Group (OMG) Systems Modeling Language (OMG SysML™), referred to as SysML v2.

The previous article notes that the transition to a MBSE approach is essential for systems engineering to meet the demands of increasing system complexity, productivity and quality, and shorter design cycles. The article also notes "MBSE emphasizes the need to create a coherent model of the system that helps to integrate other aspects of the design, including electrical, mechanical, and software. This system model provides a shared view of the system that can drive communication and coordination across the system development lifecycle (Friedenthal and Burkhart 2015)"

The following summarizes the concept for the SME in response to the needs and driving requirements defined in the previous article, and summarizes the approach to develop the requirements for SysML v2.

Role of SME in a Model-Based Engineering Environment

In Figure 1 below, each of the disciplines contributes to the development of the technical baseline of the system as part of a Model-Based Engineering (MBE) approach. Each of the disciplines use their discipline-specific models to capture and analyze different aspects of the design. The MBE environment is the overall set of tools that all disciplines use to implement a MBE approach. The SME is the part of the overall MBE environment that systems engineers use to perform MBSE and interact with other members of the development team. (Note: The scope of the SME is limited to SysML v2 modeling, but it is recognized that the SME can be extended to include many other kinds of system models and data.)

Figure 1. *The SME is part of the broader MBE environment that enables systems engineers to perform MBSE.*

The system model provides an overall description of the system that facilitates integration with the other engineering models and tools as shown in Figure 2.

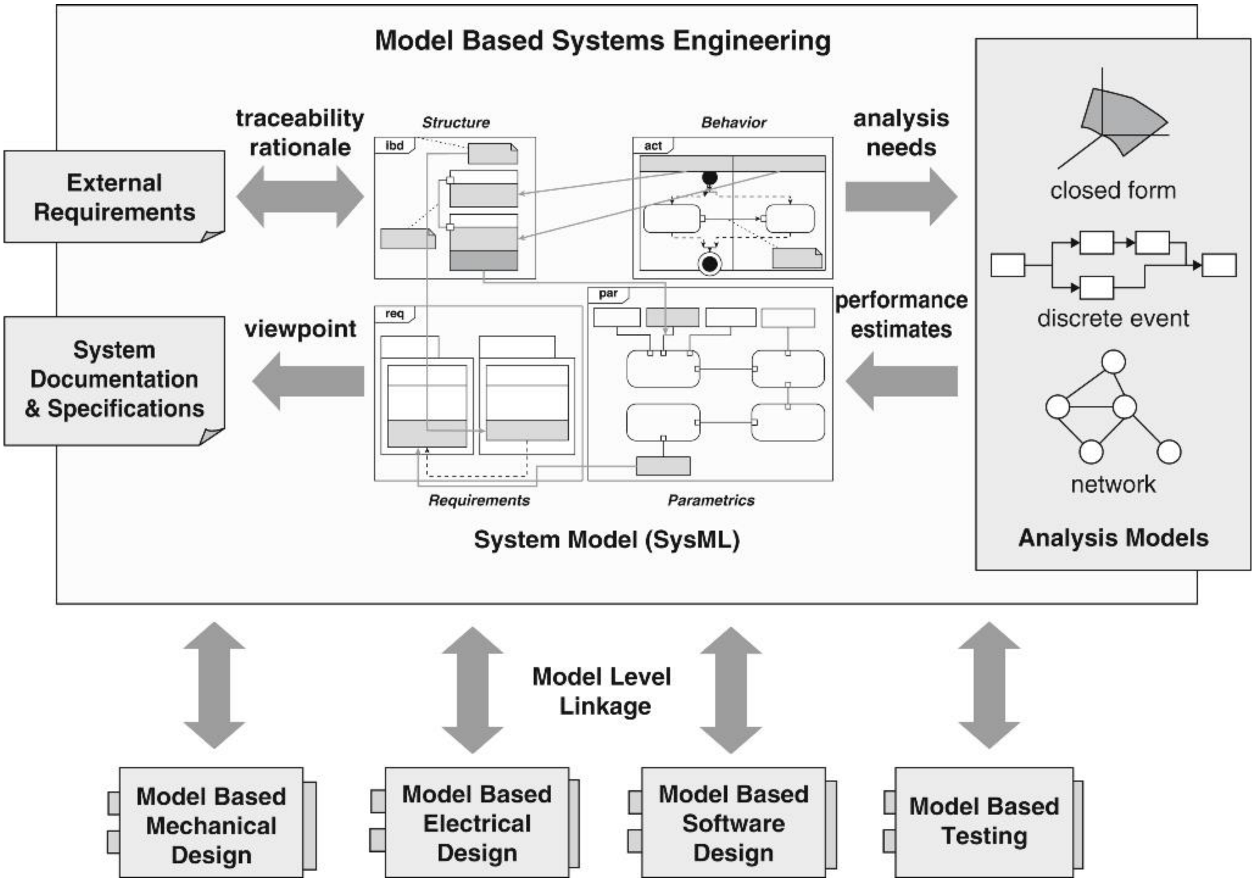

Figure 2. *The system model (Source: A Practical Guide to SysML, 3rd Edition (Friedenthal, Moore, and Steiner 2014))*

The systems engineer and others use the SME to perform MBSE as part of an overall development process to flow requirements from the mission/enterprise level to systems, subsystems, and components, and verify the components, subsystems, system, and mission requirements are satisfied. This process continues throughout the development life cycle, with the aim of delivering systems and products that meet the stakeholder needs.

SME Capabilities and Effectiveness Measures in Support of MBSE

The definition of the initial SME capabilities and driving requirements are in the August 2015 INCOSE INSIGHT article (Friedenthal and Burkhart). This article defines 7 capabilities, 8 measures of effectiveness (moe's), and 11 driving requirements the SME should support. The capabilities enable the systems engineer to perform MBSE as part of a broader model-based engineering effort, and include:

- Model construction
- Model visualization
- Model analysis

- Model management
- Model exchange and integration
- MBSE workflow and collaboration
- Extension/customization support

The effectiveness measures for the SME are used to evaluate how effectively the SME supports the above capabilities. Some of these measures are difficult to quantify, and will require further refinement. The preliminary set of 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
- **Secure:** Ability to protect all relevant data from threats
- **Usable:** Ability for stakeholders to efficiently and intuitively create, maintain, and use the model
- **Adaptable/Customizable:** Ability to extend models to support domain-specific needs

SME Services

The SME should provide services to support the following functional requirements:

- export and import structured data (for example, models, images, lists, spreadsheets)
- create, read, update, and delete model elements (including patterns)
- apply model patterns, model libraries, and reference models
- setup, validate, and execute models (for example, system models, analysis models, validation rules)
- create, read, update, delete, and execute model queries
- create, read, update, delete, and execute viewpoints
- create, read, update, and delete model metadata (for example, owner, comments, versions, status)
- manage changes to model down to the element level
- manage changes to views
- create, read, update, delete data protection controls (for example, user access permissions, roles, data rights, and markings)
- create, read, update, delete, and execute workflows and notifications
- create, read, modify, delete, and execute links between SysML models and other data
- create, read, update, delete, and execute transformations to/from SysML models

Notes:

1) A model contains model elements
2) Models can include system models, analysis models, validation rules, and queries

SME Architecture

The SME must implement the SME services to provide the functionality needed to enable systems engineers and others to evolve the system model throughout the life cycle.

Figure 3 is a view of the logical architecture of the SME. The model repository contains the data about the system, including the system model, analysis data, other metadata, and reuse libraries. The figure shows the systems engineer using a model editor with a rich interface that provides the full functionality of the SME to create, maintain, and use the system model and other data in the model repository. The systems engineer and other disciplines can also interact with the SME using a model editor with a light interface that provides the functionality needed to use and/or review the system model and other data. The user interface can present different views of the model to address different users. For example, a power subsystem engineer may view the power interfaces, and a mechanical engineer may view the system breakdown and mass allocation.

Additional functionality is provided by other logical elements of the SME to support exchange, management, and analysis of the model data, and task management to facilitate the systems engineer in implementing their systems engineering practices.

The external interface and management element enables integration between the system model and other engineering tools and models. The external interface may enable access to global task management services (for example, workflow), to support collaboration with other engineers. This includes task synchronization via notifications as the engineering work products change state. This capability is orchestrated by the system and product life cycle management environments - part of the broader MBE environment.

In addition, the logical architecture includes a development environment to further extend and customize the SME to support different domain and program needs, and ensure the customized environment continues to be interoperable with the rest of the MBE environment.

Finally, there is a practices repository that stores the systems engineering and modeling practices that are implemented by the users of the SME. The local task manager is intended to facilitate the systems engineer and others to perform these practices. The SME can also be used to support process modeling to define and update systems engineering and other organizational processes.

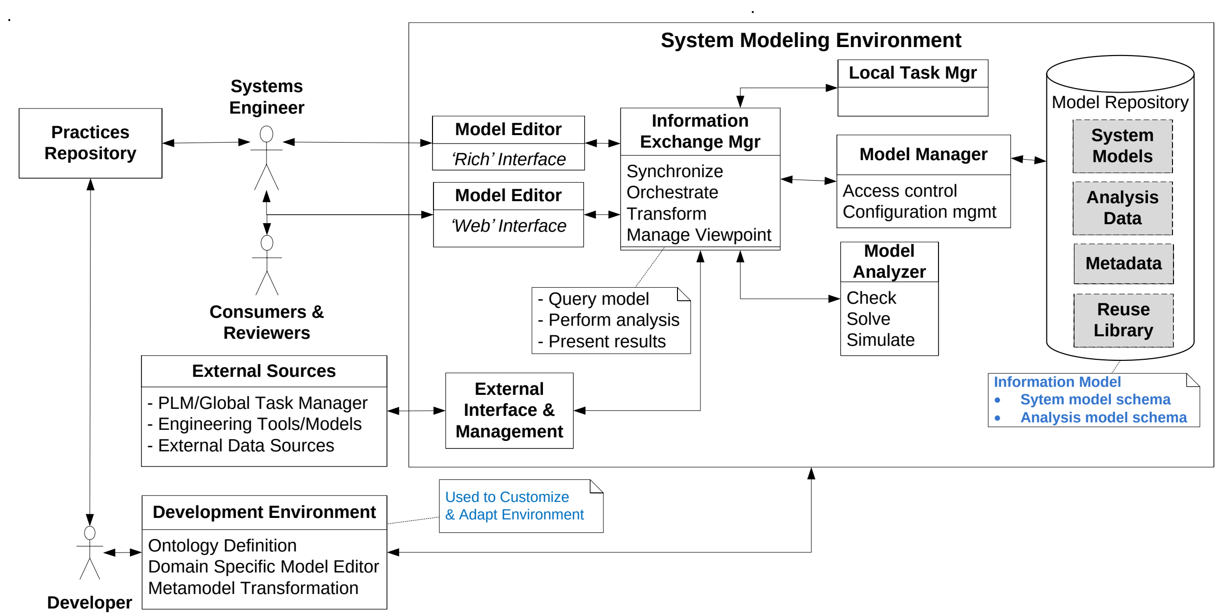

Figure 3. *System Modeling Environment-Logical Architecture*

The logical components from Figure 3 are allocated to different layers of the SME architecture as shown in Figure 4. The platform layer provides the basic hardware and operating system infrastructure. The data layer stores the model data in the repository. The services layer contains a set of applications that implement the services needed to support the SME capabilities (for example: model construction, visualization, analysis, management). An application program interface (API) requests these services. The graphical user interface (GUI) provides the interface for the users, and one or more adapters enable other tools to access the API to request these services. Finally, there is an extension layer that enables the development environment to modify and extend the SME data model, services, and interfaces.

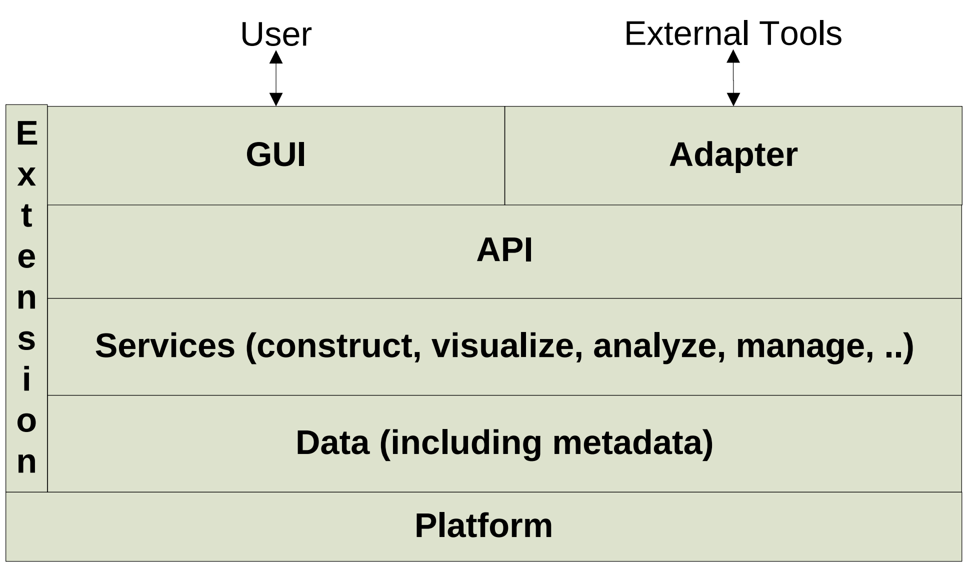

Figure 4. *System Modeling Environment Layered Architecture*

SME data model

The SME provides the capability to model systems using a systems modeling language with a precisely defined systems engineering vocabulary. A Systems Engineering Concept Model (SECM) is used to capture the key concepts, and is a primary input to help specify the requirements for the SysML v2 metamodel, profile, and model libraries. The concepts in the SECM are intended to reflect industry standards for systems engineering that include 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 and others provide the basis for defining the high level concepts which can be further elaborated to support the requirements for SysML v2. An extract from a preliminary version of the SECM-2015 Industry Reference is shown in Figure 5.

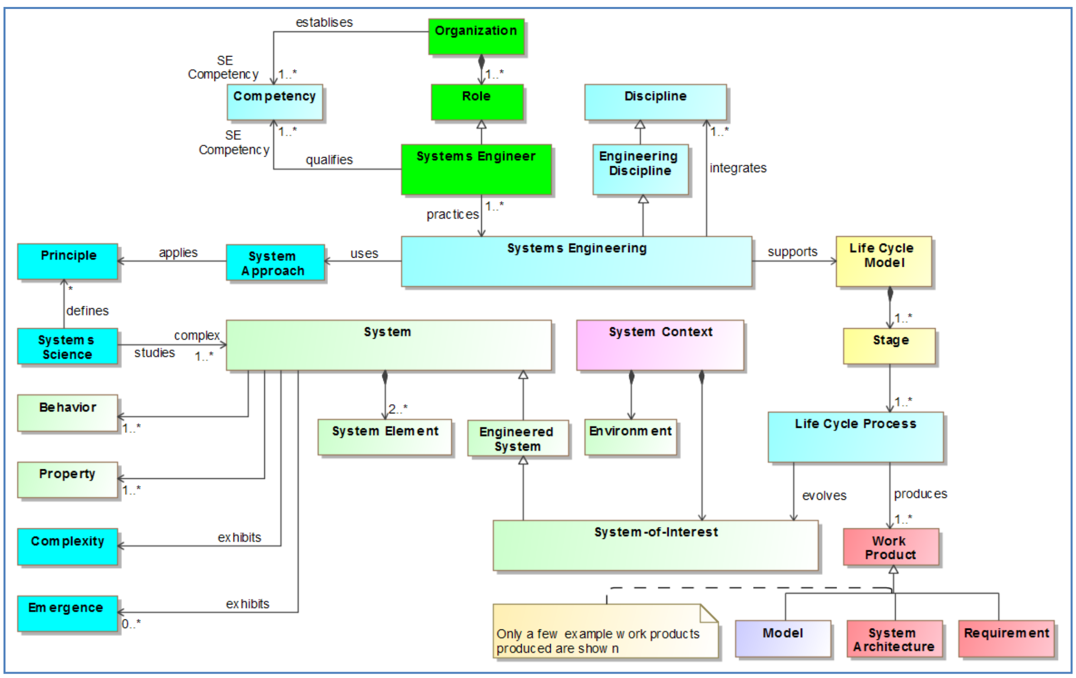

Figure 5. *Core SEBoK Concepts.*
(Extract from draft SECM-2015 Industry Reference. Used with permission)

There are many other concepts in the industry reference model beyond what is in Figure 5. The intent is to align SysML v2 with the concepts in the industry standards. However, the scope of SysML v2 is not intended to address the full scope of the industry reference model, and also may include additional concepts that are not explicitly in the industry reference model. The initial focus for SysML v2 will be on the concepts directly related to the specification, design, analysis, and verification of systems.

The element and system element concepts from the industry reference model are further elaborated in Figures 6A and 6B. Figure 6A describes a generic element that is further classified as system and system element in Figure 6B. However, these concepts are still not sufficiently detailed to fully specify the SysML v2 requirements. For example, there is a concept for property in the industry reference model as shown in the figure, but this concept does not include an explicit description of data type, units, and quantity kinds that must be supported by SysML v2.

The high level concepts from the industry reference model will be further refactored and elaborated as part of the SysML v2 requirements development effort.

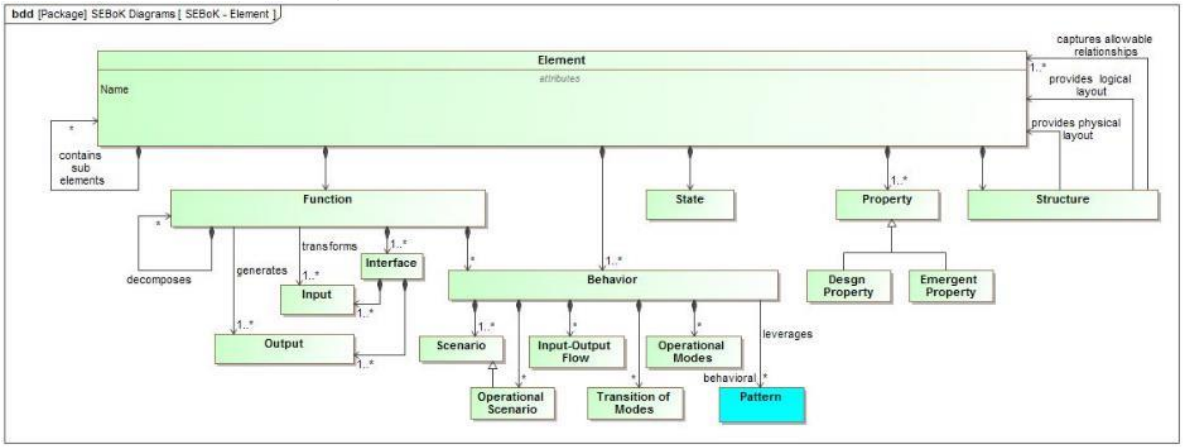

Figure 6A. *Element Concept*
(Extract from draft SECM-2015 Industry Reference. Used with permission)

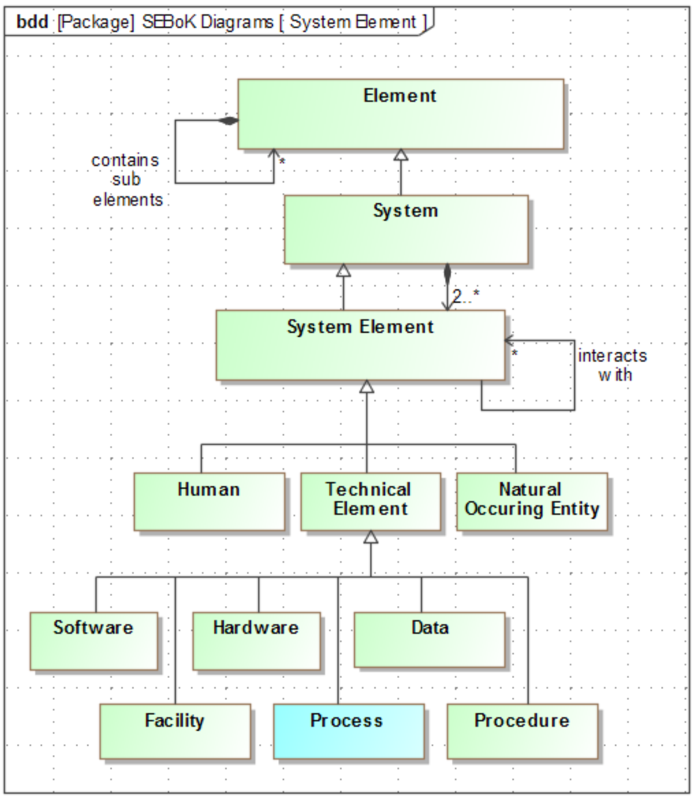

Figure 6B. *System Element Concept*
(Extract from draft SECM-2015 Industry Reference. Used with permission)

SysML v2 RFP Approach

The OMG is an industry consortium that develops modeling standards such as Unified Modeling Language (UML), SysML, and Unified Profile for DoDAF/MODAF (UPDM). The standards are subject to a rigorous development process that begins with the development of requirements that are issued in a request for proposal (RFP). Submission teams develop modeling specifications (sstandards) in response to the RFP. This adoption process is supported by multiple reviews by the applicable OMG task force, architecture board, and business committee to ensure the quality and viability of the specification. Tool vendors then implement these specifications in tools that support system, software, and domain specific modeling.

The SysML v2 RFP will specify the requirements for the next generation of the OMG SysML™ v2. SysML v2 is intended to address limitations of the current version of SysML (v1.4) to enable the effective application of model-based systems engineering (MBSE), with particular emphasis on improving precision, interoperability, and usability of the modeling language. The SysML v2 RFP will require the specification of a standard modeling language and a standard application-programming interface (API) that will be implemented by modeling tools. These tools are part of the SME.

The SysML v2 will express the core concepts required to precisely specify a system, its elements, and its environment (the system model) at levels of fidelity appropriate for the application. The system model will be used to specify system elements that can be realized by detailed design models that span many different engineering disciplines, such as electrical, mechanical, and software design (Note: the discipline-specific detailed design models are outside the scope of SysML). The modeling language will include a semantic foundation to support integrity checking of the system model, and to integrate with standard-based solvers to support the execution and analysis of the system model. The modeling language provides a graphical and textual syntax to enable the system model to be understood and used by diverse stakeholders.

This RFP will require the modeling language to be specified as a SysML metamodel that is not constrained by UML, and a SysML profile of UML. In addition, this RFP will require selected model libraries, such as those needed to represent systems of units and quantity kinds. A tool vendor may choose to implement either the meta-model or the profile, or both, along with selected libraries.

The RFP requires an API to be specified that provides a standard set of service requests to access and operate on the system model. An example is a service request to zoom in and out of the model to enhance visualization. The service requests are implemented by the system modeling environment to support model construction, visualization, analysis, management, interoperability, and workflow and collaboration. The API will be required to use open standards such as web-based interoperability standards.

The RFP also requires test cases and a reference model to enable vendors and users to demonstrate and evaluate conformance with all, or selected part of the SysML v2 specification. The RFP also requires that the SysML v2 specification include standard extension mechanisms to further extend and adapt the modeling language and API to meet the unique needs of different domains. Finally, the RFP will request user interface guidelines to promote usability of the modeling tools.

Summary

The OMG adopted the OMG Systems Modeling Language (http://www.omgsysml.org/) in 2006. Since its adoption, SysML enabled broad recognition and increased adoption of MBSE practices across industry. Systems engineers learned much from this experience, including the strengths and weaknesses of SysML as a language, and the benefits and challenges in adopting and applying MBSE. This experience provides a basis for developing the next generation of SysML v2.

An effective MBSE approach requires a robust modeling language, tools, methodology, and a workforce with the requisite skills. The SME enables systems engineers and others to apply their domain knowledge and skills to specify, analyze, design, and verify systems using a model-based approach. This article summarized a proposed concept for a SME that is being used to derive the requirements for SysML v2.

The SME concept supports a set of modeling capabilities to enable systems engineers to perform MBSE that include model construction, visualization, analysis, management, interoperability, and workflow and collaboration. The SME concept enables different kinds of users to interact with the model in ways that are adapted to their needs.

The requirements for SysML v2 support this concept by specifying both a standard data model and a standard API. The data model basis is industry standards that evolved since the requirements for SysML v1 were issued. In addition, the API is a key enhancement of SysML v2 over SysML v1 that provides a standard means to access the system model, and interact with the SysML v2 tools to improve interoperability.

The requirements for SysML v2 are under development by an OMG working group sponsored by the OMG Systems Engineering Domain Special Interest Group (SE DSIG). The members of this group include representatives from tool vendors, end users, and academia. The Working Group maintains a Wiki with detailed information on the SME concept and the requirements for SysML v2.

The development of SysML v1 specification resulted from a collaborative effort between members of the OMG, INCOSE, and the ISO STEP AP-233 Working Group. Similarly, the development of SysML v2 is intended to be a collaborative effort that engages the broader systems engineering community. Those interested in contributing to this effort should contact the authors of this article.

References.

[1] Friedenthal, S., and R. Burkhart. 2015. "Evolving SysML and the System Modeling Environment to Support MBSE" *INSIGHT* (*August 15 Volume 18 Issue 2, Pg 39-42*)

[2] Friedenthal, S., A. Moore, and F. Steiner. 2014. "A Practical Guide to SysML, 3rd Edition" Elsevier/Morgan Kaufmann (FIGURE 18.1, Pg 508)