Abstract for MBSE2020 Workshop

Title: "What to Expect from SysML Version 2?"

Author(s): Hans Peter de Koning (hanspeter.dekoning@dekonsult.com)

Affiliation: DEKonsult, Amsterdam, The Netherlands (retired from ESA per 31 Dec 2019)

In the ongoing adoption of Model-Based Systems Engineering (MBSE) the Systems Modeling Language (SysML) standard from the Object Management Group (OMG) plays a major role, as it is de facto the only global standard for MBSE. Since SysML version 1.0 was released in 2008 and has increasingly been deployed in industry and government agencies across many industry sectors to support the development of complex systems. In addition in 2017 it was adopted as the ISO/IEC 19514 standard. SysML is also used in the European space sector, e.g. on the ESA projects e-Deorbit, Euclid, PLATO and Mars Sample Return.

In the last 10 years the standard has seen a number of gradual upgrades from version 1.2 in 2010 – that marked the start of real industrial use – to the current version 1.6 release in November 2019 [1]. SysML version 1 is strongly based on Unified Modeling Language (UML) v2, and therefore inherits a number of concepts and standardization patterns from this object-oriented software engineering standard. This can be considered both a strength and a weakness. A strength because it meant that mature UML tools could be adapted with reasonable investment to support SysML, and provided good coverage for software-intensive systems. However, a weakness too, because the software engineering heritage created barriers for the uptake by systems engineers without a strong software engineering background. A general complaint on SysML v1 is that the learning curve is too steep, and that the language unnecessarily complicates modelling a number of key systems engineering concepts, such as interface connections between nested components.

This was also acknowledged at OMG and in 2015 work was started on collecting user needs and requirements for a RFP (Request for Proposal) for SysML version 2. The goal was to ensure that all lessons learned from the initial years of industrial usage would be taken into account. Since such a major overhaul of the standard can only be afforded every now and then, this preparation was taken very seriously and performed by a working group with broad representation by end-users from different industry sectors (large and small enterprises, government agencies, research institutes, academia) as well as SysML tool vendors over the course of 2 years. It resulted in two extensive RFPs: one for the SysML v2 language itself [2], and another one for the Application Programming Interface (API) and Services [3]. All discussions and prototyping that went into the preparation can be found at [4].

Since the beginning of 2018, a team of more than 100 experts from around 60 organisations -- the so-called SysML v2 Submission Team (SST) -- has been developing the second version of SysML. This work is culminating into the first full public release – for both the language and the API and Services – planned for September 2020.

The current presentation will provide an overview of the new and enhanced capabilities of SysML version 2, including but not limited to:

1. New simplified SysML meta-model, which is founded on a minimal set of key concepts.
2. The new normative and informative model libraries including the upgraded way of handling quantities, units and scales, which now have a very rigorous underlying information model, that also allows for automated unit / scale conversion, which is important when integrating models coming from different partners. Also basic geometric modelling is supported to represent e.g. the specification of enveloping shapes for system components.
3. The new textual notation, including a standardized and very powerful expression and constraint language, as well as the upgraded graphical notation (diagrams), and the integral, flexible viewpoint

/ view capabilities. These overcome many limitations of the SysML v1 Block Definition Diagrams and Internal Block Diagram, and provide much better and more precise ways to define interfaces and connections, also in deeply nested structures. Then there is the integrated approach to model behaviour (activities, functional architecture, time-based sequences, finite state machines, 4D lifecycle objects), both in precise textual notation and in diagrams that can be mixed and matched, to answer the needs of particular domains.

4. Support for variant modelling and product line engineering built into the language and/or connectable to external variant modelling tools.
5. The much improved support for integrating SysML v2 models with external analysis and simulation paradigms and tools, founded on much more precise execution semantics.
6. The prototype implementations of the textual and graphical language, on the Eclipse Modeling Framework as well as in Jupyter Notebooks.
7. The new API and Services that provide a much better and richer capability to interact with SysML models than SysML v1 XMI files. The technology neutral API specification allows for both static whole model transfers and simultaneous dynamic interaction of many client tools with one or more SysML repositories. In the current SST prototype implementation, a REST, an OSLC, and a Java API are supported.
8. The way compatibility with SysML v1 is ensured via a SysML v2 profile as well as via the new API.

The presentation will highlight how the learning curve is expected to be reduced for systems engineers. Time will be dedicated to explain the new so-called usage-focused modelling approach, which allows to directly model deeply nested architectures, in a way that feels more natural for most systems engineers. This new capability is in addition to the SysML v1 "definition/type first" approach, and still maintaining ways to ensure a rigorous modular architectures. The same definition / usage and composition patterns are consistently applied throughout all aspects of the language, for requirements, structure, behaviour, interfaces, parametrics, constraints, verification.

Finally an outlook on the deployment schedule of SysML v2 will be provided.

The author has been a member of the SysML v1 task forces since 2009 as well as the SysML v2 RFP working group, and is an active member of the Submission Team for SysML v2.

References

[1] Systems Modeling Language v1.6, OMG, November 2019, https://www.omg.org/spec/SysML/1.6/

[2] Systems Modeling Language (SysML®) v2 Request For Proposal (RFP), OMG, December 2017, https://www.omg.org/cgi-bin/doc.cgi?ad/2017-12-2

[3] Systems Modeling Language (SysML®) v2 API and Services Request For Proposal (RFP), OMG, June 2018, https://www.omg.org/cgi-bin/doc.cgi?ad/2018-6-3

[4] OMG SysML v2 RFP Working Group Wiki at http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:sysml_assessment_and_roadmap_working_group

[5] Hans Peter de Koning, "Progress on SysML v2", 13th ESA Workshop on Avionics, Data, Control and Software Systems (ADCSS2019), November 2019, ESA/ESTEC, https://indico.esa.int/event/323/contributions/5057/attachments/3756/5215/11.55_-_Progress_on_SysML_v2.pdf

[6] General information on the OMG Systems Modeling Language (SysML), see http://www.omgsysml.org

[7] General information on MBSE across all industry sectors, INCOSE/OMG MBSE Wiki at http://www.omgwiki.org/MBSE/doku.php