Rigorous Modelling of Quantities for Model-Based Systems Engineering

Hans Peter de Koning (DEKonsult), Amsterdam, The Netherlands

MODELSWARD22 | 6-8 Feb 2022 | Virtual Event
Outline

- Background
- Motivation
- Desirable Characteristics and History
- Vocabulary and Semantic Foundation
- Semantic Data Model
- Capturing ISQ as well as SI and US Customary Units in SysML v2
- Examples in SysML v2 – Demonstration with pilot implementation
- Wrap-up
My background
Space systems engineering at ESA
OMG SysML version 2 Development

Request for Proposal
- Including SE Concept Model
- Started 2015
- Language RFP released Dec 2017
- API RFP released May 2018

SysML v2 RFP WG
- Initially two submission teams – since end 2018 merged into one
- Started Jan 2018 on language (KerML & SysML)
- Started June 2018 on API & Services
- Agile workflow with monthly sprints
- Initial submission of language & API Spec’s delivered to OMG on 17 Aug 2020
  (3 parts: KerML, SysML language, API & Services)
- Includes working open-source pilot implementations and training material with examples
  - Monthly releases on GitHub

SysML v2 Submission Team (SST)
- Major stakeholders’ review held in Feb 2021
- Revised submissions delivered to Aug & Nov 2021 for review
- Monthly releases of pilot software, specifications and training material
- Final submission to OMG scheduled for May 2022

Finalization Task Force
- Finalization of the specifications
- Monthly releases of pilot software, spec’s and training material
- Expected production tool developments (both COTS and open-source)
Motivation

- Any engineering model – naturally – uses a lot of (physical) quantities
  - For properties, variables, parameters, etc.
- MBSE – being systems engineering – always involves coordination across multiple disciplines, life cycle stages, and organizations
- MBSE models therefore need a solid basis to represent quantities and their measurement references (units, scales, coordinate frames)
- Even stronger: MBSE needs a rigorous semantic model of quantities and measurement references to avoid costly mistakes and enable integration
- Important MBSE use case: (repeated) integration of contributed subsystem models into a higher level system model
  - Often models from different disciplines / organizations use different units, scales, coordinate frames
  - Need for reliable, automated method to rebase the integrated model on single set of units, scales, coordinate frames
  - E.g., for interface definition and verification, for conversion to analysis / simulation tools
All quantities in an MBSE model must have explicit semantics!

measurement unit ... and more!
(In)Famous case: Mars Climate Orbiter crash (1999)

Root cause was interface problem:
Change of impuls (m\(\cdot\)\(\Delta v\)) unit mismatch
- spacecraft software used N\(\cdot\)s (as required)
- ground control software used lbf\(\cdot\)s (violating ICD)
factor 4.45 difference
Desirable Characteristics

- Grounded in formal logic
  - SysML v2 metamodel is expressed in Semantic MOF (SMOF)
  - Uses same first order logic foundation as RDF/OWL2

- Grounded in established international standards
  - Should provide digitalized libraries of ISQ, SI and US Customary Units

- Support scalar & vector & tensor quantities

- Support coordinate frames and transformations

- Provide intuitive, compact syntax for quantity expressions

- Enable automated quantity value conversion to different unit / scale

- Enable automated quantity expression checking
  - Compatible quantity dimensions i.e. dimensional analysis ... and more

Most appeared as requirements in SysML v2 RFP
Major issue in SysML v1: quantity value property type combines quantity type and unit... so, change of unit forces change of type!
Implementations in other languages

- Many languages (and tools) support measurement units and ‘dimensions’
  - Mathematica, Maple, Modelica, Matlab, MathCAD, ...
  - Packages for Python, Java, Scala, C, C++, C#, Javascript, Julia, ..., mostly open source

- Some ontologies and schemas exist
  - QUDT (http://qudt.org/) – originally a spin off from SysML v1 QUDV (around 2009)
  - FIBO Quantities and Units (https://spec.edmcouncil.org/fibo/ontology/FND/Quantities/QuantitiesAndUnits/) adapted from SysML v1 QUDV
  - QUOMOS (https://www.oasis-open.org/committees/quomos/) – started but never completed
  - UnitsML (https://www.unitsml.org/) by NIST

- So, what is the big deal?

- Almost all only support simplest case: value \cdot unit product
  - Do not support interval | ordinal | logarithmic | cyclic measurement scales
  - Limited support for quantity dimensions
  - Lacking or very limited support for vector | tensor quantities and coordinate frames

- SysML v1 QUDV already went a bit further, but was cumbersome to use in practice
Foundation – VIM and ISO/IEC 80000 (ISQ & SI)

- BIPM VIM, JCGM 200:2012, “International vocabulary of metrology”


- ISO/IEC 80000 “Quantities and Units” – International System of Quantities (ISQ) and International System of Units (SI)

- The NIST Reference on Constants, Units, and Uncertainty

  - In particular Appendix B “Conversion Factors” – links US Customary Units to SI
  - https://www.nist.gov/pml/special-publication-811
### VIM – Selected vocabulary terms (1/3)

<table>
<thead>
<tr>
<th>term</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>quantity</strong></td>
<td>property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number and a reference</td>
</tr>
</tbody>
</table>
| **kind of quantity**          | aspect common to mutually comparable quantities  
Note 1: The division of 'quantity' according to 'kind of quantity' is to some extent arbitrary.  
Note 2: Quantities of the same kind within a given system of quantities have the same quantity dimension. However, quantities of the same dimension are not necessarily of the same kind. |
| **base quantity**             | quantity in a conventionally chosen subset of a given system of quantities, where no subset quantity can be expressed in terms of the others   |
| **derived quantity**          | quantity, in a system of quantities, defined in terms of the base quantities of that system                                               |
| **quantity dimension**        | expression of the dependence of a quantity on the base quantities of a system of quantities as a product of powers of factors corresponding to the base quantities, omitting any numerical factor |
| **quantity of dimension one** | quantity for which all the exponents of the factors corresponding to the base quantities in its quantity dimension are zero               |

<table>
<thead>
<tr>
<th>term</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>quantity value</td>
<td>number and reference together expressing magnitude of a quantity</td>
</tr>
<tr>
<td>ordinal quantity</td>
<td>quantity, defined by a conventional measurement procedure, for which a total ordering relation can be established, according to magnitude, with other quantities of the same kind, but for which no algebraic operations among those quantities exist</td>
</tr>
</tbody>
</table>
| quantity-value scale      | ordered set of quantity values of quantities of a given kind of quantity used in ranking, according to magnitude, quantities of that kind  
Examples: Celsius temperature scale, Time scale |
| ordinal quantity-value scale | quantity-value scale for ordinal quantities  
Examples: Rockwell C hardness scale, Scale of octane number for petroleum fuel |
<table>
<thead>
<tr>
<th>term</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>measurement unit</td>
<td>real scalar quantity, defined and adopted by convention, with which any other quantity of the same kind can be compared to express the ratio of the two quantities as a number</td>
</tr>
<tr>
<td>base unit</td>
<td>measurement unit that is adopted by convention for a base quantity</td>
</tr>
<tr>
<td>derived unit</td>
<td>measurement unit for a derived quantity</td>
</tr>
<tr>
<td>multiple of a unit</td>
<td>measurement obtained by multiplying a given measurement unit by an integer greater than one</td>
</tr>
<tr>
<td>submultiple of a unit</td>
<td>measurement unit obtained by dividing a given measurement unit by an integer greater than one</td>
</tr>
</tbody>
</table>

Quantity dimension in ISO/IEC 80000 (ISQ) is defined using 7 base quantities: Length (L), Mass (M), Time (T), Electric Current (I), Thermodynamic Temperature (Θ), Amount of Substance (N), Luminous Intensity (J)

\[
q_{\text{dim}}(q) = L^\alpha \cdot M^\beta \cdot T^\gamma \cdot I^\delta \cdot \Theta^\epsilon \cdot N^\zeta \cdot J^\eta
\]

(where \( \alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta \) are the dimensional exponents)

Examples:
- length: \( q_{\text{dim}}(L) = L \)
- velocity: \( q_{\text{dim}}(v) = L \cdot T^{-1} \)
- energy: \( q_{\text{dim}}(E) = L^2 \cdot M \cdot T^{-2} \)
- torque: \( q_{\text{dim}}(M_Q) = L^2 \cdot M \cdot T^{-2} \)
Quantity modelling basics in SysML v2 (simplified)

ScalarQuantityValue

\[ \text{num : Number}[1] \]

ScalarMeasurementReference

\[ \text{mRef [1]} \]

Abstract generic supertype for any scalar measurement reference:
- MeasurementUnit
- MeasurementScale

Example concrete textual syntax in user model (M1)

\[ \text{attribute length : LengthValue = 1340 [mm];} \]

Note: attribute replaces value property of SysML v1
Extend taxonomy to model multi-dimensional quantities

- TensorQuantityValue
  - Can represent component-free and componentized tensors, vectors
  - mRef [1]

- VectorQuantityValue
  - mRef [1]
  - {redefines mRef}

- ScalarQuantityValue
  - mRef [1]
  - {redefines mRef}

- TensorMeasurementReference
  - Is generalization of coordinatized vector space product
  - mRefs[*]

- VectorMeasurementReference
  - Is generalization of coordinatized vector space, i.e., coordinate frame

- ScalarMeasurementReference
  - Is generalization of measurement unit and scale
  - Specifies e.g., units for all dimensions / components
New in KerML and SysML v2 are self-standing features (here AttributeUsage)

E.g. can declare reusable attribute mass : MassValue[1]; directly at package level, not owned by an Item or Part Definition (which replaces SysML v1 Block)
Q: Why Interval Scale in addition to units?

A: To support relative and absolute quantities
E.g., for time dimension: duration (relative) and time instant (absolute)

UTC timestamp = 2021-12-19T14:25:33.425321Z

UTC ‘zero reference’
0000-01-01T00:00:00Z (= 1 BC!)

My experiment t₀ = 0 [mts]

interval of duration = 6.45 [s]

My experiment t = 19.825 [mts]

Note: A simplification of course, assuming a Newtonian global clock
may need to consider different clocks, precision, other time scale without leap seconds, etc.
Also Ordinal, Logarithmic, Cyclic Ratio Scales

**Ordinal Scale**
Beaufort Wind Force
no unit, just ordering

**Logarithmic Scale**
Sound Pressure Level
symbol = dB(A) or dB_A
unit = dB
reference level = 20 $[\text{μPa}]$

**Cyclic Ratio Scale**
Rotation Angle
unit = °
modulo = 360 [$\text{°}$]

From https://en.wikipedia.org/wiki/Beaufort_scale
From https://en.wikipedia.org/wiki/Sound_pressure
Coordinatization – Coordinate Frames

- In order to quantify vectors and tensor coordinate frames are needed
- Predefined typical coordinate frames and transformations for engineering
  - 2D / 3D Cartesian, 3D Spherical, 3D Cylindrical, 2D Polar
  - Geographic coordinate system (lat, long, altitude)
  - ISO 10303-242 (STEP) axis placement
  - Transformation matrix (translation and rotation about principal axes)
- Domains can add their particular coordinate frames by extension
  - E.g., rotation via quaternions, ...
**Coordinate Frames and Placement Model**

Coordinate Frame is a specialization of `VectorMeasurementReference`

*Origin translation and basis direction vectors are specializations of `VectorQuantityValue`*

*Example CoordinateTransformation / placement for 3D Cartesian coordinate frames*
Free and Bound Vectors

- Multi-dimensional equivalent of relative and absolute (scalar) quantities are free and bound vector quantities

- For many quantity dimensions there are pairs of free and bound quantities
  - In case of a vector the quantity dimension is taken from its magnitude

<table>
<thead>
<tr>
<th>quantity dimension</th>
<th>relative / free</th>
<th>absolute / bound</th>
<th>tensor order</th>
<th>dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (T)</td>
<td>duration</td>
<td>time instant</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Energy (L²·M·T⁻²)</td>
<td>kinematic energy</td>
<td>potential energy</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Length (L)</td>
<td>displacement vector</td>
<td>position vector</td>
<td>1</td>
<td>1, 2 or 3</td>
</tr>
</tbody>
</table>

- Observation:
  Interval Scale is same as 1D bound vector space with 1D coordinate frame
  Interval Scale zero shift (offset) is same as origin translation
Operations on Free and Bound Vectors

Example 2D but holds for any N-dimensional space
Same algebra as for absolute quantity on interval scale!
E.g., temperature on Celsius or Fahrenheit scale.

- Point \( P_1 \) in ‘free vector space’
  aka affine space or torsor

- Bound position vector 1 in A \( (v_{1A}) \)
- Origin of ‘bound’ vector space A
- Bound position vector 1 in B \( (v_{1B}) \)
- Origin of ‘bound’ vector space B
- Bound position vector 2 in A \( (v_{2A}) \)
- Bound position vector 2 in B \( (v_{2B}) \)
- Free displacement vector 1→2
  \( = P_2 - P_1 = v_{2A} - v_{1A} \) (in vector space A)
  \( = v_{2B} - v_{1B} \) (in vector space B)

Algebra:
- Free vector + free vector → free vector
- Free vector – free vector → free vector
- Bound vector – bound vector → free vector
- Bound vector + free vector → bound vector

Cannot add two bound vectors, only subtract
**SysML v2 Measurement reference model details**

- `isBound` - false declares free vector space, true declares bound vector space
- `contravariantOrder` - declare index conventions for some kinds of tensors
- `covariantOrder` -
- `mRefs` specifies measurement references for all tensor or vector components

**Array provides multi-dimensional structure**
Examples of Implementation in SysML v2

Example in Jupyter Lab Notebook (local or remote server and FireFox browser)

Example quantity attributes

Model authored in textual language

Quantity and units imported from SysML v2 libraries

On-the-fly, auto-layout visualization via built-in PlantUML/SysMLv2 visualizer
Wrap-up

- **SysML v2 well underway – has very a solid quantities and units model**
  - ISO/IEC 80000 parts 3-12 captured as semantic model libraries (700+ quantities, few hundred units)
  - NIST SP811 captured for US Customary Units with official conversion factors (300+)
  - Integrated with SysML v2 expression language

- **Currently working on complete coordinate frames and transformations**
  - Including basic geometry modelling capabilities

- **Next steps**
  - Implement automated conversion algorithm and quantity dimensions checking
  - Develop expression of uncertainties and probability distributions
## References

<table>
<thead>
<tr>
<th>Reference</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SysML v2 Submission Team (SST) public repositories on GitHub</td>
<td><a href="https://github.com/Systems-Modeling/">https://github.com/Systems-Modeling/</a></td>
</tr>
<tr>
<td>General information on MBSE across all industry sectors, INCOSE/OMG MBSE Wiki</td>
<td><a href="http://www.omgwiki.org/MBSE/doku.php">http://www.omgwiki.org/MBSE/doku.php</a></td>
</tr>
<tr>
<td>General information on the OMG Systems Modeling Language (SysML)</td>
<td><a href="http://www.omg.sysml.org">http://www.omg.sysml.org</a></td>
</tr>
<tr>
<td>Systems Modeling Language (SysML®) v2 Request For Proposal (RFP), OMG, December 2017</td>
<td><a href="https://www.omg.org/cgi-bin/doc.cgi?ad/2017-12-2">https://www.omg.org/cgi-bin/doc.cgi?ad/2017-12-2</a></td>
</tr>
<tr>
<td>Systems Modeling Language (SysML®) v2 API and Services Request For Proposal (RFP), OMG, June 2018</td>
<td><a href="https://www.omg.org/cgi-bin/doc.cgi?ad/2018-6-3">https://www.omg.org/cgi-bin/doc.cgi?ad/2018-6-3</a></td>
</tr>
<tr>
<td>Systems Modeling Language v1.6, OMG, November 2019</td>
<td><a href="https://www.omg.org/spec/SysML/1.6/">https://www.omg.org/spec/SysML/1.6/</a></td>
</tr>
<tr>
<td>Ed Seidewitz, “SysML v2 and MBSE: The Next Ten Years”, MODELS 2018 Conference, Copenhagen, Denmark, Oct 2018</td>
<td>link</td>
</tr>
<tr>
<td>Hans Peter de Koning, “SysML Version 2 Approaching Industrial Use”, ESA MBSE2021 Workshop</td>
<td>link</td>
</tr>
</tbody>
</table>