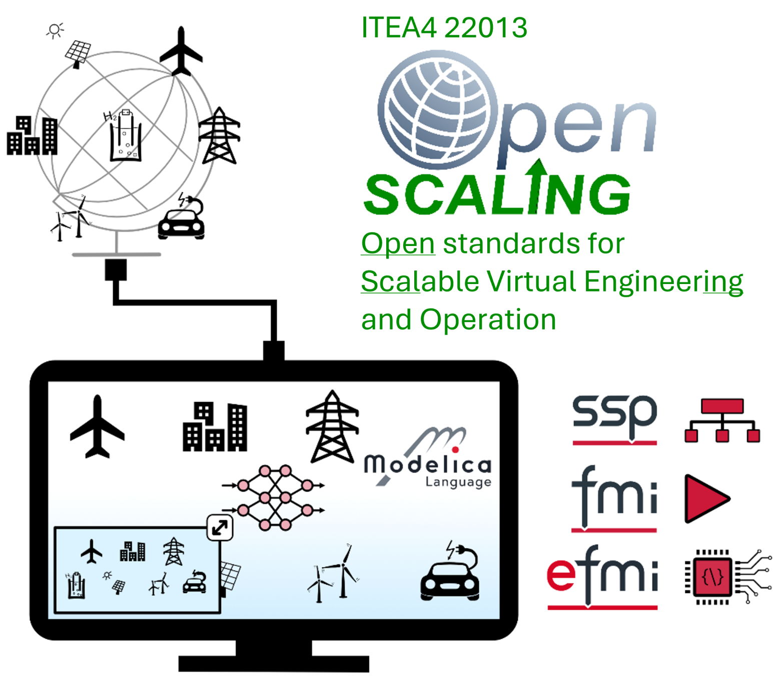
Contents



ITEA4 22013
Open
SCALING
Open standards for
Scalable Virtual Engineering
and Operation
Modelica Language
ssp
fmi
efmi

Deliverable D2.1: Uncertainty Quantification Draft

Version 0.1 (September 30, 2025)

This document presents a draft proposal for defining *Uncertainty Quantification* in a standardized and tool-independent way. It is being developed as part of the OpenSCALING project under the ITEA4 program. The goal is to establish a new Modelica Association (MA) project which will produce a version 1.0 of this document as an MA standard and maintains it in the future.

This prototype draft is being worked on actively, and thus is subject to change without notice.

Project Acronyms

Acronym	Description
DAE	Differential-Algebraic Equation
FMI	Functional Mock-up Interface (standard for model exchange and co-simulation, see https://fmi-standard.org/)
FMU	Functional Mock-up Unit (an instance of an FMI component, see https://fmi-standard.org/)
HVAC	Heating, Ventilation and Air Conditioning
LSS	Large-scale System
Modelica	Standard for modelling of cyber-physical systems, see https://modelica.org
ODE	Ordinary Differential Equation

From Full Project Proposal

ID	Type	Description	Due Month	Access
D2.1	SW	Modelica, FMI, eFMI, SSP enhancements for UQ	M18	Public

1. Introduction

1.1. Intent of this Document

This document defines a tool-independent data model and related XML/JSON file formats for describing Uncertainty Quantification (UQ) experiments. XML schema files are located in the schema directory. Script xml-to-json.js is provided to automatically convert an XML UQ definition into a compact JSON representation.

These file formats facilitate seamless exchange and execution of UQ workflows for models parameterized in a manner similar to the FMI standard, particularly targeting FMI, SSP and Modelica models. It is expected that tools will support importing and exporting these files to perform uncertainty quantification as defined in these files. Optionally, the XML file may also be stored in the `extra` directory of an FMI component in a *backwards compatible way* (= compliant tools can ignore definitions that they do not support).

The initial scope is intentionally narrow: uncertainties are described at the parameter level and analyses are limited to *forward uncertainty propagation* and *global sensitivity analysis* of potentially *nonlinear* dynamic models. Future versions will be expanded to include *validation against measurements* and *predictive-capability assessments*. The standard is designed for cross-layer reuse; for example, component suppliers can include UQ metadata together with their model (e.g., inside an FMU `extra` folder), while system integrators can compose these UQ artefacts in larger studies. Two examples - a DC motor and an industrial mobile valve - illustrate the format and its intended uses.

Non-normative text in section 2 and section 3 is given in square brackets in italic font: *[Especially examples are defined in this style.]*

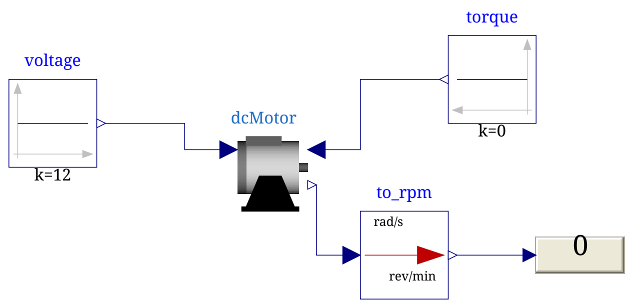
[Note: Although the document refers to major versions of the FMI, SSP, and Modelica Language standards, everything described in this document also applies to all subsequent minor versions. For further information on compatibility, see section Versioning and Layered Standards in the FMI 3.0 specification, as well as [Pre2013]]

1.2. Example DC-Motor

The following example demonstrates how to associate uncertainty information with parameters of the Modelica model `DCMotorUQExample.ScenarioDCMotor` in the file examples/dc_motor/DCMotorUQExample.mo and how to perform multiple simulations using Monte Carlo methods to estimate the uncertainty of the motor speed. This procedure is commonly referred to as *forward Uncertainty Quantification (UQ) experiment*, see, e.g. Wikipedia.

1.2.1. Model

The top-level Modelica object diagram of this model is shown in the next figure:



The `voltage` component provides the constant supply voltage, the `torque` component provides the constant mechanical load and the motor angular velocity is provided as an output signal and displayed in revolutions per minute. The component `dcMotor` has the following parameter menu:

Parameters

R	0.1	Ω	resistance
L	1e-4	H	inductance
c_m	0.01	N·m/A	electrical (back-EMF) motor constant
c_g	0.01	N·m/A	mechanical (torque) motor constant
J	5e-5	kg·m²	inertia of the rotor
d	1e-4	N·m·s/rad	friction

The motor starts from rest and accelerates under a supply voltage `voltage.k = 12V` until it reaches a steady-state speed. The goal is to determine the uncertainty in the motor angular velocity (`dcMotor.omega`), given uncertainties in certain motor parameters. To achieve this, the parameter uncertainties and a simple Monte Carlo simulation setup are provided in either of the two files:

- XML file examples/dc_motor/dc_motor_setup.xml
- JSON file examples/dc_motor/dc_motor_setup.json (= automatically generated from the XML file with script scripts/xml-to-json.js).

1.2.2. Uncertain Parameters

An essential part in the XML file is the definition of the parameters that shall be used in the Monte Carlo simulation. This definition is a *superset* of element <ssv:ParameterSet> of the SSP standard. In particular, stochastic distributions of uncertain parameters can be described, besides providing new values of parameters:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Example to define parameters that are used for the whole operating domain -->

<uq:ExperimentSetup
    name="ScenarioDCMotor_ForwardUQ"
    xmlns:uq ="http://openscaling.org/UQ1/UncertaintyQuantification"
    xmlns:ssv="http://ssp-standard.org/SSP2/SystemStructureParameterValues"
    xmlns:ssc="http://ssp-standard.org/SSP2/SystemStructureCommon">

  <uq:ParameterSet> <!-- Super set of SSP parameters -->
    <uq:Parameters>
      <uq:Parameter name="dcMotor.d"> <!-- SSP Parameter -->
```

```
        <ssv:Real value="1e-4" unit="N.m.s/rad"/>
      </uq:Parameter>

      <uq:UncertainParameter name="dcMotor.R" source="Estimated" sourceInfo="From
project XYZ">
        <uq:Normal mu="0.1" sigma="0.01" unit="Ohm"/>
      </uq:UncertainParameter>

      <uq:UncertainParameter name="dcMotor.L" source="Computed" sourceInfo="From
project XYZ">
        <uq:Uniform minimum="1.05e-4" maximum="1.15e-4" unit="H"/>
      </uq:UncertainParameter>
    </uq:Parameters>

    <uq:Units>  <!-- From SSP without changes -->
      <!-- ... -->
    </uq:Units>
  </uq:ParameterSet>

</uq:ExperimentSetup>
```

- The namespace `uq:` is used for all new elements introduced by this document.
  The namespaces `ssv:` and `ssc:` are from the SSP standard.
- Parameter `dcMotor.d`, gets a new value with element `uq:Parameter`, which is a SSP parameter definition. Alternatively, values of this parameter and of other parameters could be included from an SSV-file.
- Parameters `dcMotor.R` and `dcMotor.c_m` are defined by a *normal* and a *uniform stochastic distribution*, respectively. This means that during the Monte Carlo simulation, random values are picked for every simulation run according to the defined distribution and to the selected sampling method. With attributes `sources` and `sourceInfo` information about the *source* of the uncertainty information is given.

1.2.3. Setup of Experiment

With further elements the setup of the Monte Carlo experiment is defined:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Example to define Forward Uncertainty Quantification -->

<uq:ExperimentSetup
    name="ScenarioDCMotor_ForwardUQ"
    xmlns:uq ="http://openscaling.org/UQ1/UncertaintyQuantification"
    xmlns:ssv="http://ssp-standard.org/SSP2/SystemStructureParameterValues"
    xmlns:ssc="http://ssp-standard.org/SSP2/SystemStructureCommon">

  <uq:ExperimentSet>
    <uq:ForwardUncertaintyQuantification>
```

```
      <uq:ParameterSet>
        <!-- Definition of parameters -->
      </uq:ParameterSet>

      <uq:ObservedVariables>
        <uq:ObservedVariable name ="dcMotor.omega"
                             distributionApproximation="Normal"/>
      </uq:ObservedVariables>

      <uq:DesiredResults
          data      ="dc_motor.mat"
          summary   ="dc_motor_result_summary.xml"
          scope     ="Trajectory"
          mean      ="true"
          standardDeviation="true"
          histogram ="true"
          pdf       ="true"
          cdf       ="true"
          quantile  ="true">
        <uq:Percentiles level="0 0.05 0.5 0.95 1.0"/>
        <uq:SobolIndices order="2" total="true"/>
      </uq:DesiredResults>

      <uq:SimulationMethod stopTime="0.5" interval="0.01" tolerance="1e-4"
solver="DASSL">
        <ssc:Annotation type="com.3ds.uq">
          <uq:KeyValue key="translation.pedanticModeForCheckingModelicaSemantics"
value="true"/>
          <uq:KeyValue key="output.storeVariablesAtEvents" value="true"/>
        </ssc:Annotation>
      </uq:SimulationMethod>

      <uq:SamplingMethod>
        <uq:LatinHypercube numberOfSamples="1000"
                           randomNumberGenerator="MT19937-64"
                           randomNumberSeed="10">
          <ssc:Annotation type="com.3ds.uq">
            <uq:KeyValue key="FurtherParameter" value="FurtherValue"/>
          </ssc:Annotation>
        </uq:LatinHypercube>
      </uq:SamplingMethod>
    </uq:ForwardUncertaintyQuantification>
  </uq:ExperimentSet>
</uq:ExperimentSetup>
```

- The element `ObservedVariables` specifies the variables for which results are to be computed and stored. Optionally, the attribute `distributionApproximation` indicates that the result at a given time point should be approximated by a Normal distribution.
- The element `DesiredResults` specifies which results should be computed for all observed variables and where the resulting data should be stored. In this example, the results are written

to the file `dc_motor.mat` (a tool-specific result file). Alternatively, a directory name may be used if the results are distributed across multiple files.

- The element `SimulationMethod` defines the integration method to be used for the simulations. Optionally, tool-specific setup data can be provided through the `Annotation` element, for example by using key/value pairs defined with the standardized `KeyValue` element.
- Finally, the element `SamplingMethod` specifies the sampling method of the Monte Carlo simulation along with all required method parameters.

More details of this example and the results of the Monte Carlo simulation experiment are shown in [Baj2025].

1.2.4. Operational Domain

It is also possible to define a *set of Monte Carlo experiments* for selected points in the operational domain (see [Ros2025] for a discussion of operational domain descriptions). For the motor example, this could be defined in the following way:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Example to define parameters for an operational domain where
     a set of parameters is associated to a set of ExperimentPoints
-->

<uq:ExperimentSetup
    name="ScenarioDCMotor_ForwardUQ"
    xmlns:uq ="http://openscaling.org/UQ1/UncertaintyQuantification"
    xmlns:ssv="http://ssp-standard.org/SSP2/SystemStructureParameterValues"
    xmlns:ssc="http://ssp-standard.org/SSP2/SystemStructureCommon">

  <uq:ParameterSet> <!-- Super set of SSP parameters -->
    <uq:Parameters>
      <!-- Parametes that hold for all operating points -->
    </uq:Parameters>

    <uq:OperationalDomain>
      <uq:Axes>
        <uq:Axis name="voltage.k" unit="V"/>
        <uq:Axis name="torque.k" unit="N.m"/>
      </uq:Axes>

      <!-- The boundary of the operational domain
           Possible values: ConvexHull, Box, UnitHyperCube
      -->
      <uq:Boundary>
        <uq:ConvexHull>
          <uq:Point coordinates=" 1 0" />
          <uq:Point coordinates=" 1 5" />
          <uq:Point coordinates="10 0" />
          <uq:Point coordinates="10 5" />
```

```
      </uq:ConvexHull>
    </uq:Boundary>

    <!-- For every point, a Monte Carlo simulation
         shall be carried out
    -->
    <uq:ExperimentPoints>
      <uq:PointSet id="PS1">
        <uq:Points>
          <uq:Point id="P1" coordinates="9 0"/>
          <uq:Point id="P2" coordinates="5 0"/>
          <uq:Point id="P3" coordinates="1 0"/>
        </uq:Points>

        <uq:Parameters>
           <!-- Parameters that hold for PointSet PS1 -->
        </uq:Parameters>
      </uq:PointSet>

      <uq:PointSet id="PS2">
        <uq:Points>
          <uq:Point id="P4" coordinates="9 1"/>
          <uq:Point id="P5" coordinates="5 1"/>
          <uq:Point id="P6" coordinates="1 1"/>
        </uq:Points>

        <uq:Parameters>
           <!-- Parameters that hold for PointSet PS2 -->
        </uq:Parameters>
      </uq:PointSet>
    </uq:ExperimentPoints>
  </uq:OperationalDomain>

  <uq:Units>
    <!-- Units used above -->
  </uq:Units>
 </uq:ParameterSet>

</uq:ExperimentSetup>
```

- The operational domain is seen as an n-dimensional space with n axes. In the motor case, the axes are the excitation voltage `voltage.k` and the external disturbance torque `torque.k`.
- Element `Boundary` defines the boundary of the operational domain of the motor with respect to these two axes. It is defined here by the convex hull of the given points. Defining this domain is optional and is used to document the desired operational domain for which the motor model should be valid.
- For every point defined in element `ExperimentPoints`, a Monte Carlo simulation shall be performed. Typically, every such point is inside the operational domain defined in the previous element. Every Point has a unique identification `id` that is used to identify the results for this

point.

The parameters `voltage.k` and `torque.k` can be seen as uncertain parameters that cannot be characterized by stochastic distributions due to insufficient knowledge. This type of uncertainty is known as epistemic uncertainty (see, for example, [Dan2020]).

1.3. Example Industrial Mobile Valve

This example describes a forward UQ experiment carried out on a work section of a Parker Hannifin mobile hydraulic valve [Par2024].

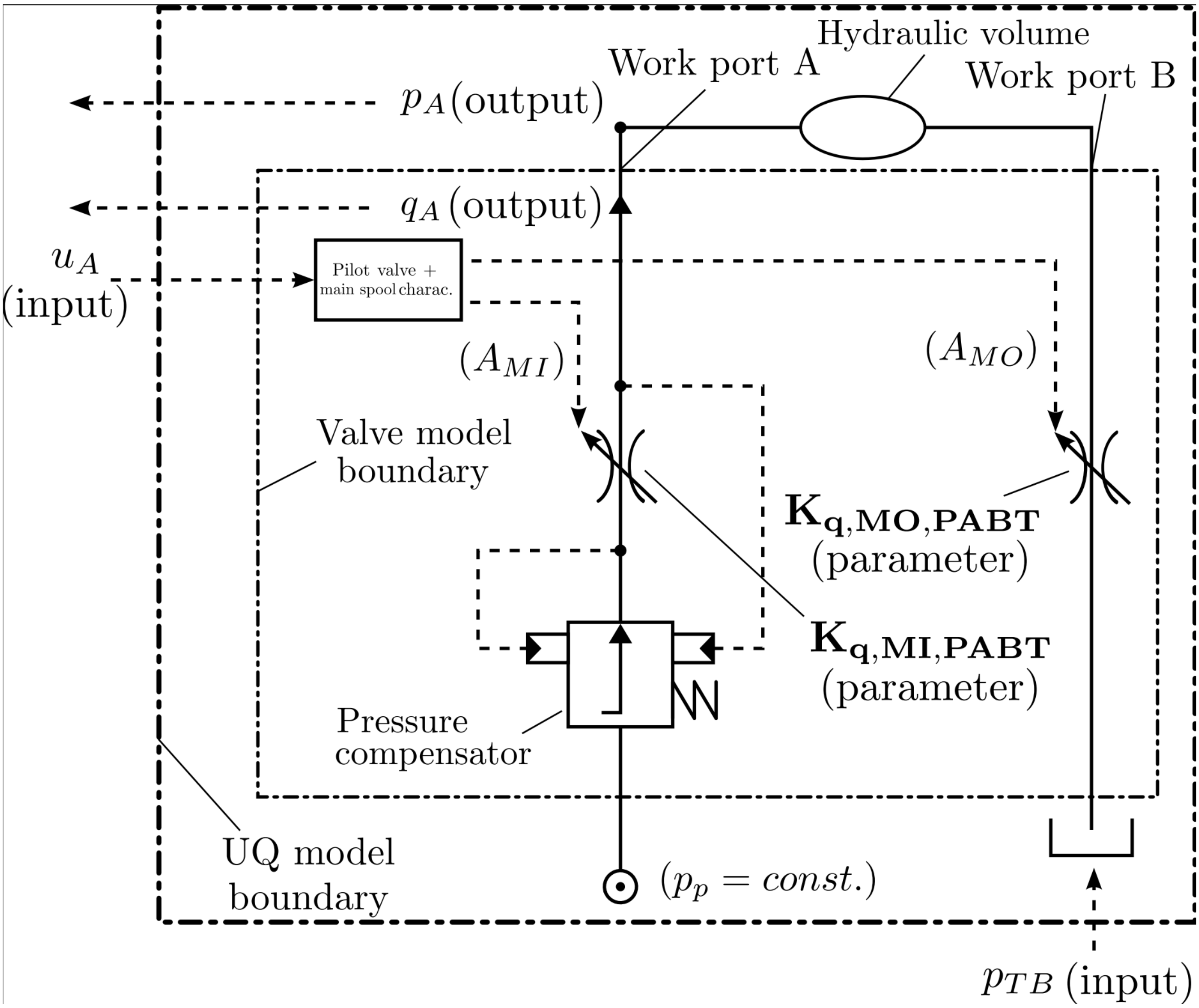
1.3.1. Model

This use case is based on a single work section of a proportional, load-sensing, pre-compensated mobile directional control valve from the Parker L90LS series [Par2024]. The valve's core function is to deliver a hydraulic flow that is proportional to an input electrical current, regardless of variations in pump and load pressures.

In a typical simulation scenario, the valve's work ports are connected to a closed hydraulic volume and supplied with constant pump pressure. The flow and counterpressure responses are evaluated for different input currents.

Manufacturing tolerances introduce uncertainty in parameters such as the discharge coefficient. Therefore, a Uncertainty Quantification (UQ) study is performed to assess the effect of these parameter uncertainties on valve behavior. This example demonstrates how UQ metadata can be packaged with the valve model using the proposed standard format.

The next figure shows the schematics of the model of the example mobile valve, which is packaged as an SSP component in file examples/mobile_valve/mobile_valve.ssp:



1.3.2. Experiment Setup

We are interested in how uncertainty in the valve gains, Kq,MI,PABT and Kq,MI,PABT propagate to uncertainty in the supplied flow (qA) and counter pressure (pA) outputs. The operational domain is assumed to be 0-900 uA control current (uA) and 0-20 bar supply pressure (pTB). This is an operational domain in two axes, i.e. a box, but more complex convex hulls are possible with multiple axes and boundary coordinates. The uncertainties of the valve gains are normally distributed and also the supplied flow and counter pressure are expected to be normally distributed. To establish the distribution of qA and pA, 10 different points in the operational domain are defined. For each of these, 1000 simulations will be done according to the defined "Simulation method" and with varying valve gains. Based on this, the two distributions can be established.

The uncertain parameters of the mobile valve and the experiment setup are defined in the XML file examples/mobile_valve/mobile_valve_setup.xml as follows:

```
<?xml version="1.0" encoding="UTF-8"?>

<uq:ExperimentSetup
  name="ScenarioMobileValve_ForwardUQ"
```

```xml
  xmlns:uq="http://openscaling.org/UQ1/UncertaintyQuantification"
  xmlns:ssv="http://ssp-standard.org/SSP1/SystemStructureParameterValues"
  xmlns:ssc="http://ssp-standard.org/SSP1/SystemStructureCommon">

  <uq:Model
    name="ScenarioMobileValve"
    file="ParkerSimPack.ssp"
    modelResourceMetaData="mobile_valve.srmd" />

  <uq:ExperimentSet>
    <uq:ForwardUncertaintyQuantification>
      <uq:ParameterSet>
        <uq:Parameters>
          <uq:UncertainParameter name="KqMI_PABT_gain" source="Estimated"
            sourceInfo="Parker Hannifin's Hydraulic Valve Systems">
            <uq:Normal mu="1" sigma="0.1" minimum="0.8" maximum="1.2" unit="-" />
          </uq:UncertainParameter>
          <uq:UncertainParameter name="KqMO_PABT_gain" source="Estimated"
            sourceInfo="Parker Hannifin's Hydraulic Valve Systems">
            <uq:Normal mu="1" sigma="0.1" minimum="0.8" maximum="1.2" unit="-" />
          </uq:UncertainParameter>
        </uq:Parameters>

        <uq:OperationalDomain>
          <uq:Axes>
            <uq:Axis name="uA" unit="mA" />
            <uq:Axis name="pTB" unit="bar" />
          </uq:Axes>

          <uq:Boundary>
            <uq:ConvexHull>
              <uq:Point coordinates="0 0" />
              <uq:Point coordinates="0 20" />
              <uq:Point coordinates="900 20" />
              <uq:Point coordinates="900 0" />
            </uq:ConvexHull>
          </uq:Boundary>

          <uq:ExperimentPoints>
            <uq:PointSet id="PS1">
              <uq:Points>
                <uq:Point id="P1" coordinates="345 1" />
                <uq:Point id="P2" coordinates="355 1" />
                <uq:Point id="P3" coordinates="375 1" />
                <uq:Point id="P4" coordinates="395 1" />
                <uq:Point id="P5" coordinates="415 1" />
                <uq:Point id="P6" coordinates="440 1" />
                <uq:Point id="P7" coordinates="465 1" />
                <uq:Point id="P8" coordinates="485 1" />
                <uq:Point id="P9" coordinates="505 1" />
                <uq:Point id="P10" coordinates="525 1" />
```

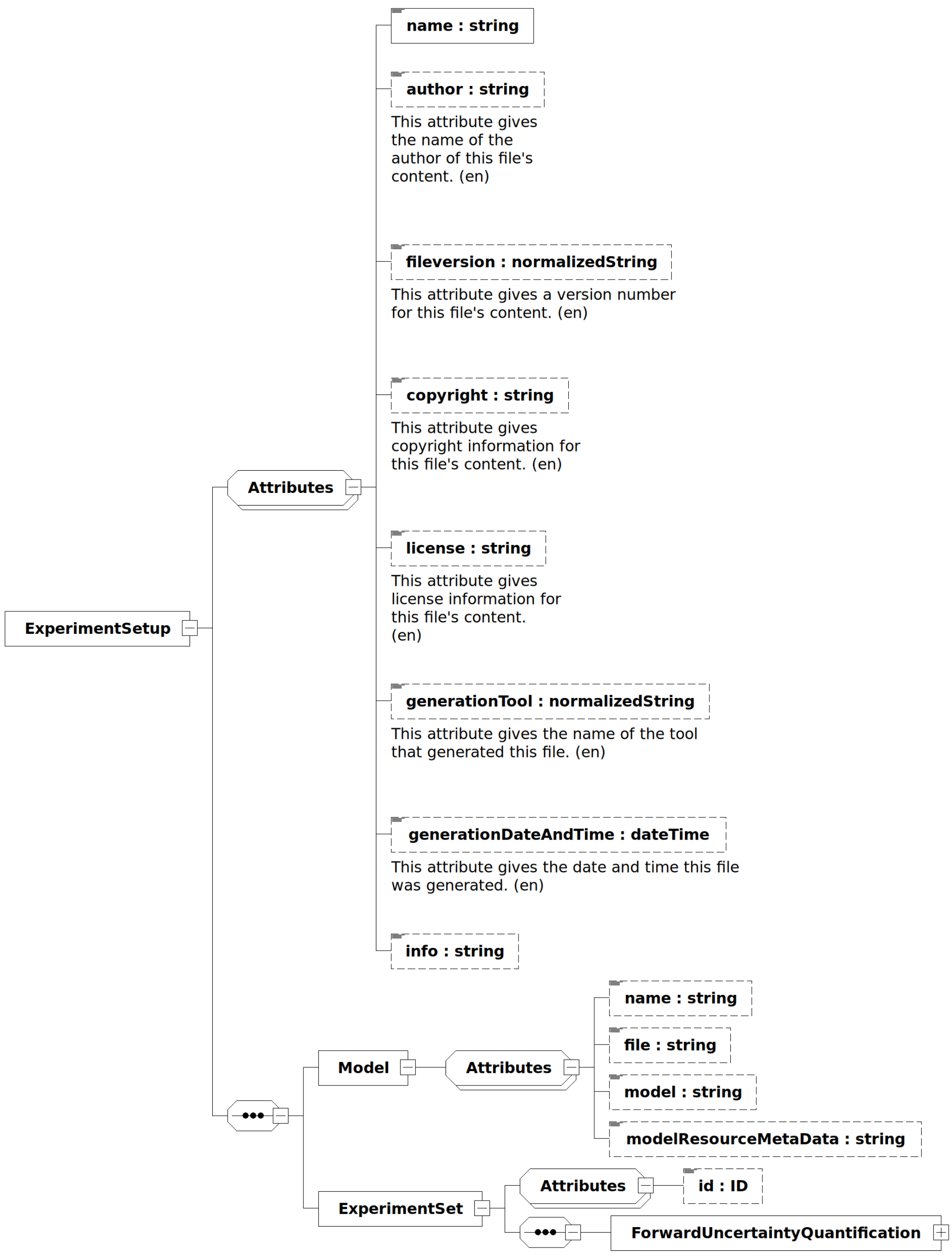
```xml
              </uq:Points>
            </uq:PointSet>

          </uq:ExperimentPoints>
        </uq:OperationalDomain>

        <uq:Units>
          <ssc:Unit name="-">
            <ssc:BaseUnit factor="1" offset="0" />
          </ssc:Unit>
          <ssc:Unit name="bar">
            <ssc:BaseUnit kg="1" m="-1" s="-2" factor="1e5" offset="0.0" />
          </ssc:Unit>
          <ssc:Unit name="mA">
            <ssc:BaseUnit A="1" factor="1e-3" offset="0.0" />
          </ssc:Unit>
          <ssc:Unit name="L/min">
            <ssc:BaseUnit m="3" s="-1" factor="6e-4" />
          </ssc:Unit>
        </uq:Units>
      </uq:ParameterSet>

      <uq:ObservedVariables>
        <uq:ObservedVariable name="qA" distributionApproximation="Normal" />
        <uq:ObservedVariable name="pA" distributionApproximation="Normal" />
      </uq:ObservedVariables>

      <uq:DesiredResults
        data="mobile_valve.csv"
        summary="mobile_valve_result_summary.xml"
        scope="FinalTime"
        histogram="true">
      </uq:DesiredResults>

      <!-- TODO: Viktor -->
      <uq:SimulationMethod stopTime="0.5" interval="0.001" tolerance="1e-4"
solver="FixedStepMaster" />

      <uq:SamplingMethod>
        <uq:PseudoRandom numberOfSamples="1000" />
      </uq:SamplingMethod>
    </uq:ForwardUncertaintyQuantification>
  </uq:ExperimentSet>
</uq:ExperimentSetup>
```

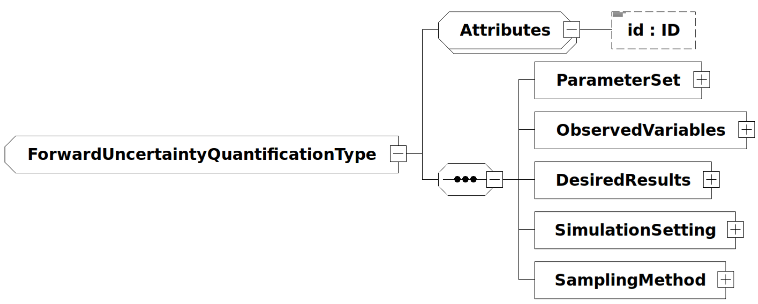
More details of this example and the results of the forward UQ experiment for the requested experiment points are provided in [Baj2025].

2. XML file format

This section defines the uncertainty quantification with XML schemas.

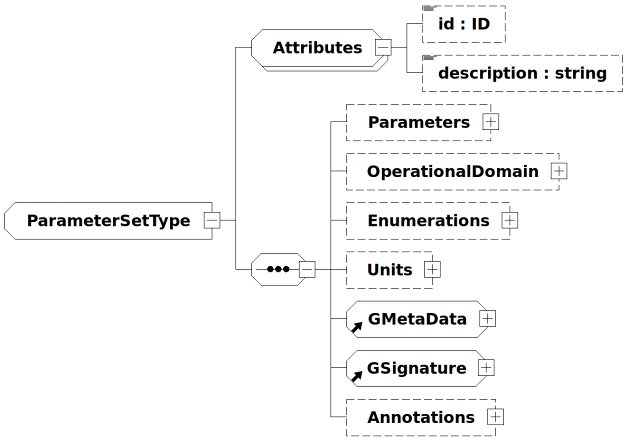
2.1. Root level

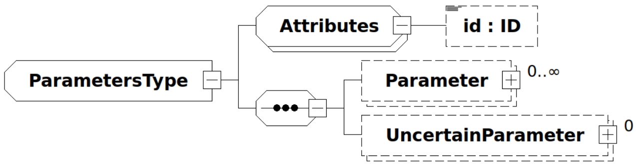




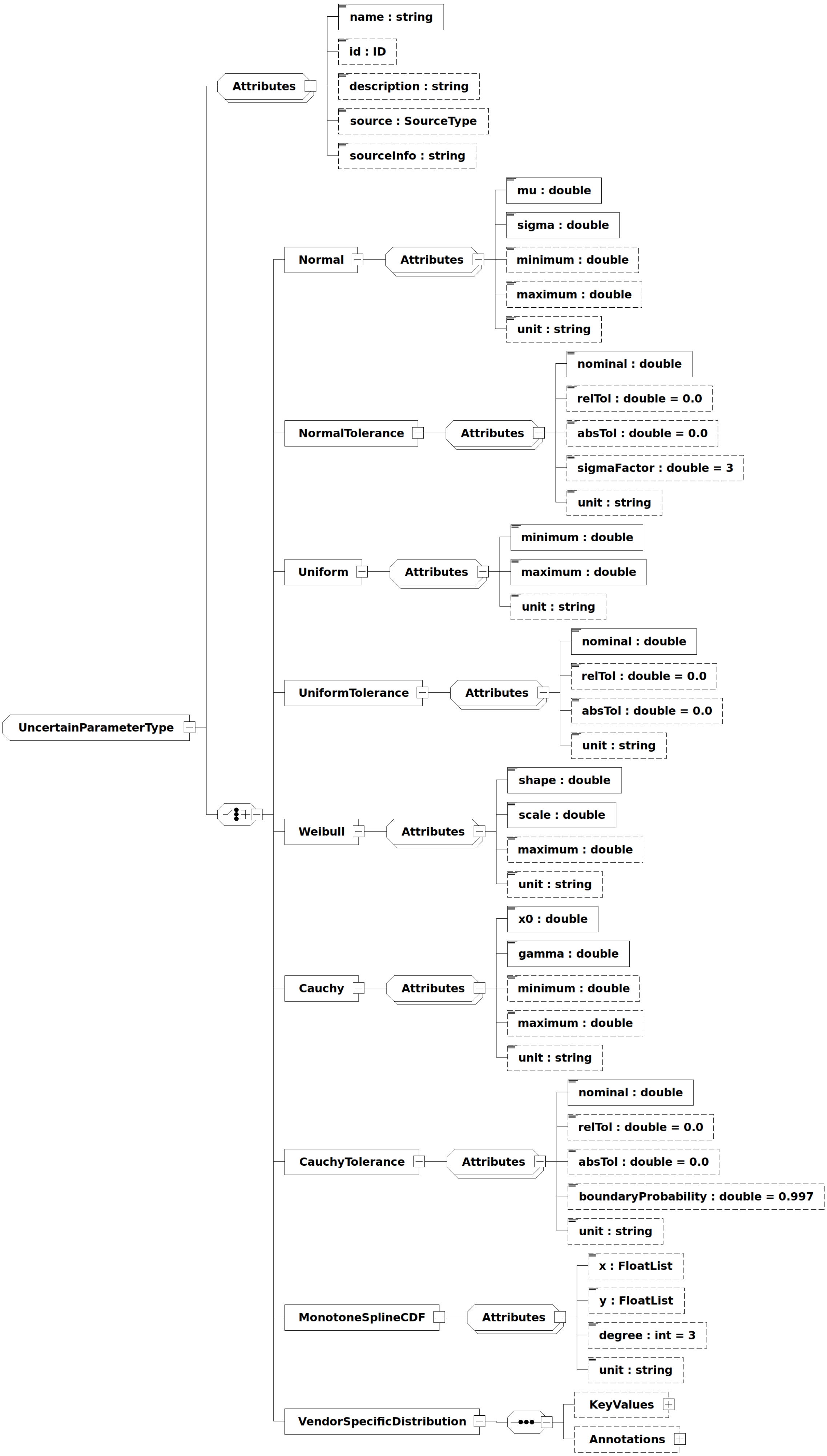
2.2. Parameter sets

2.2.1. Overall structure





2.2.2. Uncertain parameters



UncertainParameterType
Attributes
name : string
id : ID
description : string
source : SourceType
sourceInfo : string
Normal
Attributes
mu : double
sigma : double
minimum : double
maximum : double
unit : string
NormalTolerance
Attributes
nominal : double
relTol : double = 0.0
absTol : double = 0.0
sigmaFactor : double = 3
unit : string
Uniform
Attributes
minimum : double
maximum : double
unit : string
UniformTolerance
Attributes
nominal : double
relTol : double = 0.0
absTol : double = 0.0
unit : string
Weibull
Attributes
shape : double
scale : double
maximum : double
unit : string
Cauchy
Attributes
x0 : double
gamma : double
minimum : double
maximum : double
unit : string
CauchyTolerance
Attributes
nominal : double
relTol : double = 0.0
absTol : double = 0.0
boundaryProbability : double = 0.997
unit : string
MonotoneSplineCDF
Attributes
x : FloatList
y : FloatList
degree : int = 3
unit : string
VendorSpecificDistribution
KeyValues
Annotations

Uncertain scalar floating variables are described by continuous, stochastic distributions. These can be, for example, uncertain parameters as input for an experiment or distributions that variables reach in steady-state as result of an experiment.

Missing (needs to be added):
array floating variables, scalar integer variables

Three different types of distributions can be defined (for details, see image and schema/UncertainParameterType.xsd)

1. **Standard distributions**, such as the `Normal` distribution, represent widely known distributions that are defined with a few parameters. Via the *optional* attributes `minimum` and `maximum` it is possible to define lower and upper bounds on every distribution, transforming them into *truncated* distributions. The given parameters are always with respect to the underlying original, non-truncated, distribution. *[A tool can redraw a random number from which a quantile is computed, if the quantile is not in the minimum..maximum limits, or it can transform the underlying distribution to its truncated form with a simple formula, see e.g. Wikipedia, Julia function Truncated.]*
2. The distribution `MonotoneSplineCDF` defines a *Cumulative Distribution Function (CDF)* by a monotone spline that is defined by node values (x_i, y_i) with y_i the Cumulative Distribution values and a spline `degree`. It is required that all values x_i and y_i are strictly monotone increasing. If, for example, either measurements or Monte Carlo simulation CDFs are available by **tabulated values**, a monotone spline of the desired degree can be constructed for example with the open source Python function `PchipInterpolator` from `scipy.interpolate`. The corresponding *Quantile* can be determined by constructing a monotone spline using the interchanged node values (y_i, x_i) and the *Probability Density Function (PDF)* can be constructed by differentiating the montone spline of the CDF.
3. The distribution element `VendorSpecific` defines a distribution that is not (yet) standardized in this document but is available in the target tool. It allows a **target tool specific** definition.

Attributes of Distributions:

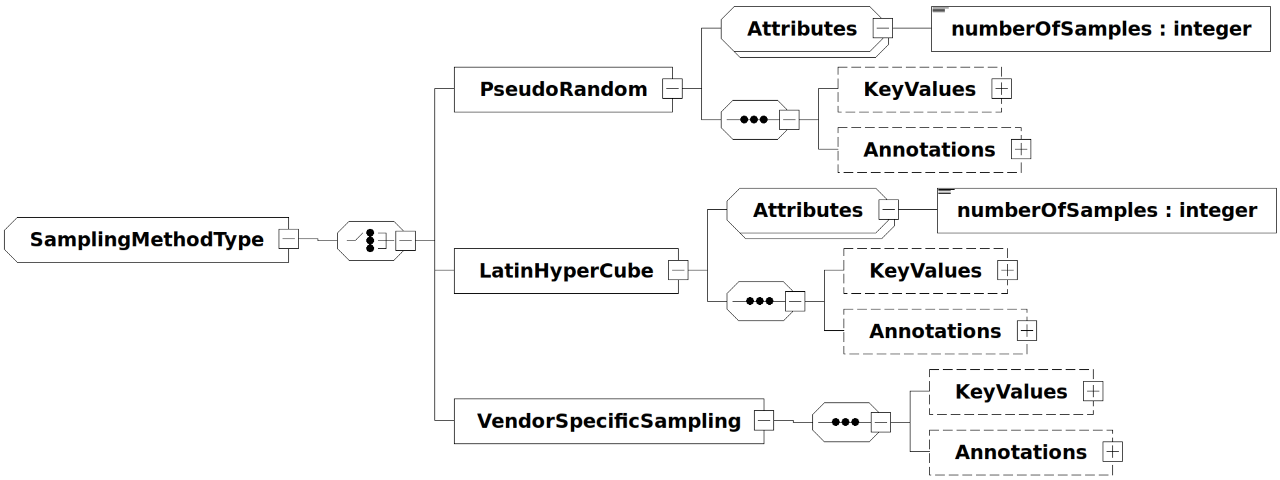
Show attributes in a table.

Test of Formulas:

PDF: $f(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

Quantile: $Q(p) = \sqrt{2}\,\sigma\,\mathrm{erf}^{-1}(2p-1) + \mu$

2.2.3. Sampling



3. JSON file format

A JSON file can be automatically generated from an XML file with the provided script scripts/xml-to-json.js using the command:

```
node  xml-to-json.js  file.xml  file.json
```

This script uses xml-js for the conversion with the xml-js options:

- compact: true,
- ignoreDeclaration: true,
- ignoreComment: true

Additional Javascript code performs further transformations

- Expands any attributes objects into their parent objects (removing the attributes layer) and placing these elements before any element on the same level
- XML string values are mapped to JSON values, such as
    - "0.2" is changed to 0.2 (number)
    - "true" is changed to true (boolean)
    - "0.2 0.3 4 5" is changed to [0.2, 0.3, 4, 5] (number vector)

4. Known Limitations of this Standard

To be reviewed

- Computational Scalability: Monte Carlo sampling methods scale poorly with model complexity and parameter count.#
- Limited UQ Methods: Standard covers forward propagation only. Missing: inverse UQ, global sensitivity analysis, etc.

- Tool Implementation Dependencies: Success requires vendor adoption of tool-specific annotations. No enforcement mechanism for consistent implementation across platforms.
- Schema Extensibility: Current data model may not accommodate future UQ methodologies or domain-specific requirements.

5. Contributors

- Amin Bajand, Linköping University, Sweden
- L. Viktor Larsson, Parker Hannifin Manufacturing, Sweden
- Lena Buffoni, Linköping University, Sweden
- Elmir Nahodovic, Dassault Systèmes, Sweden
- Robert Hällqvist, Saab Aeronautics, Sweden
- Oliver Lenord, Robert Bosch GmbH, Germany
- Hans Olsson, Dassault Systèmes, Sweden
- Martin Otter, German Aerospace Center (DLR-FK), Germany
- Antoine Vandamme, Robert Bosch GmbH, Germany
- Adrian Pop, Linköping University, Sweden

References

[Baj2025] Bajand, A. et al. (2025): **Towards a Common Standard for Uncertainty Quantification**. Modelica & FMI Conference 2025

[Dan2020] Danquah, B. et al. (2020): **Statistical Model Verification and Validation Concept in Automotive Vehicle Design**. DOI 10.1016/j.procir.2020.02.175

[Hae2023] Hällqvist, R., et al. (2023): **Toward Objective Assessment of Simulation Predictive Capability**. DOI 10.2514/1.I011153

[Ott2022] Otter, M. et al. (2022): **Towards Modelica Models with Credibility Information**. DOI 10.3390/electronics11172728

[Pre2013] Preston-Werner, T. (2013): **Semantic Versioning 2.0.0**. URL: https://semver.org/spec/v2.0.0.html

[Par2024] Parker Hannifin Corp. (2024): **Mobile Directional Control Valve – Series L90LS**. URL: https://ph.parker.com/us/en/product/l90ls-mobile-directional-control-valve/l90ls-mobile-valve (visited on 2025-03-17).

[Ros2025] Rosenlund, E. et al. (2024): **Objectively Defined Intended Uses, a Prerequisite to Efficient MBSE**. DOI 10.3384/ecp20729