

Modelica Models in SSP

Dag Brück

Dassault Systèmes, Lund, Sweden, dag.brueck@3ds.com

Abstract

This is a proposed optional extension for SSP 2.0 that defines how Modelica models can be referenced in SSP. It specifies the mapping of key Modelica concepts to SSP, which necessitates a few small extensions. The purpose is to broaden the scope of SSP to embrace the more powerful modeling concepts of Modelica, for environments that can support it.

Keywords: Modelica, SSP, extension, standardization

1 Introduction

Using System Structure and Parameterization (SSP) (Mai et al. 2019) as the high-level architecture description combined with parameter sets, there is a natural desire to be able to reference not only FMUs but also Modelica component models in SSP. Modelica (Elmqvist 2014; MLS36 2023) offers strong modeling capabilities and usually yields a more efficient and accurate simulation of complex systems compared to component-based modeling with FMUs.

Given the example in Figure 1, We see two control-oriented blocks defined as FMUs, and then a large and more complex physical model with tightly-coupled elements that, experience has proven, are not well represented by interconnected FMUs. In this example, the hybrid driveline uses pre-defined electrical, mechanical and hydraulic components from Dymola's VeSyMA

library. Templates with replaceable components offers great flexibility for exploring design alternatives. Because of Modelica's inherent properties with equations, acausal connections, etc., efficient simulation code can be generated without sacrificing modular flexibility.

However, SSP has strong capabilities for combining the system topology with meta-data (Heinkel et al. 2022), multiple parameter sets, name mapping and basic simulation setup (Hällqvist et al. 2021). In total, it yields an attractive high-level package for a credible simulation experiment (Heinkel and Steinkirchner 2022).

SSP is attractive for packaging interconnected simulation modules with parameter data into a single unit that is suitable for simulation. For many tools, connection of causal FMUs are sufficient, but as the example shows, structures that are more complex can be efficiently manipulated and simulated if SSP is extended to the Modelica domain. This proposal adds those capabilities, without adding undue complexity to tools that chose to forego the benefits of Modelica. The purpose of this proposal is to provide a minimal solution for mapping Modelica models into an SSP context, without any attempt to cover advanced Modelica concepts.

The scope of this document is to define how components and connectors can be specified as Modelica models, and the mapping of Modelica modifiers to an SSP syntax. Further constraints are presented in the Discussion.

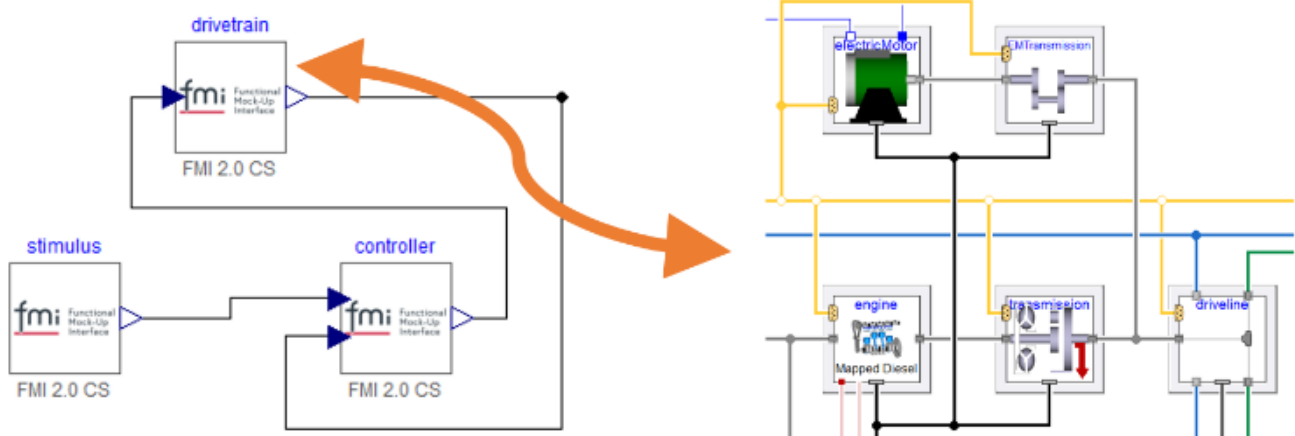


Figure 1. SSP system description using a complex component with Modelica representation.

2 Identification of the new SSP format

Within the constraints of the existing version of the SSP specification, it is possible to make extensions using a layered standard. Such a layered standard is limited to adding annotations and new media (MIME) types.

Because this proposal defines additional attributes, for reasons discussed below, a new version number is required. In the intermediate period until a future standardization, the SSP design group has decided that SSDs (System Structure Descriptions) using this extension shall use the version number *2.0-alpha.1*.

3 Representation of components

For SSP to support Modelica component models, the proposed encoding is:

- The component source attribute contains the path of the Modelica model. The source URI *modelica:* designates a model in the namespace of the Modelica environment.
- A media type (formerly known as a MIME type) is a two-part identifier for file formats and format contents. The media type used for Modelica is "text/x-modelica".

Possibly an optional version number of the Modelica library should be added, to support on-demand loading.

An example of such a Modelica component element is shown in Listing 1.

4 Representation of connectors

Modelica connectors for built-in input and output types are easily mapped to SSP connectors, see Table 1. Doing this mapping facilitates connections with FMUs and nested SSDs to Modelica components with fundamental connector types.

Modelica Type	SSP Type	SSP Kind
RealInput	ssc:Real	input
RealOutput	ssc:Real	output
IntegerInput	ssc:Integer	input
IntegerOutput	ssc:Integer	output
BooleanInput	ssc:Boolean	input
BooleanOutput	ssc:Boolean	output
StringInput	ssc:String	input
StringOutput	ssc:String	output

Table 1. Mapping of standard connector types in Modelica.Blocks.Interfaces to SSP.

In SSP 2.0 we expect a richer set of types because of the adaption to FMI 3. This offers the opportunity to represent additional Modelica types, for example arrays and perhaps even hierarchically structured types.

Modelica connectors of more advanced types are mapped the same way as Modelica component models.

- The connector type is `ssc:Binary`.
- The connector source attribute contains the full path of the Modelica model.
- The media type is "text/x-modelica".
- Acausal Modelica connector types are in SSP of `kind="acausal"`. This is a generalization compared to SSP 1.0.

For all connector types, the following extension is made:

- ConnectorGeometry is amended: An optional attribute `rotation` of type `xs:double` may be specified. It should be noted that this extension is of general interest to SSP, regardless of Modelica support.

Two such Modelica connectors are shown in Listing 2.

5 Representation of modifiers

Modelica modifiers are mapped to SSP parameter sets as follows:

- Literal modifiers of Modelica types Real, Integer, Boolean and String are mapped to their corresponding types in SSP.
- Other modifier expressions are mapped to `ssv:Enumeration`, with the value attribute containing the Modelica text of the modifier value. This is needed to handle expressions, which are common in Modelica models.

An example of a parameter binding is shown in Listing 3. A more complete example is shown in Listing 4.

6 Discussion

After presenting the actual proposal for extension in Sections 3-5, some of the design choices can be discussed.

6.1 The scope of the proposal

If we intend to work actively with Modelica models in the SSP context (not only display the architecture), we can assume that a full Modelica environment (Brück 2023), and hence any needed Modelica library, is available. For that reason, the SSD only needs to store references to the Modelica models.

Storing the model text of a component model and all dependent models would be a major effort (corresponding to "Save Total" in Dymola), but doable. If needed, a tool could for example store the Modelica text as an SSP annotation. It should be noted that this approach would fail for many libraries that are deployed in encrypted form, or use a license mechanism.

```

<ssd:Elements>
  <ssd:Component name="sin"
    type="text/x-modelica" source=
      "modelica://Modelica.Blocks.Sources.Sine">
    <ssd:Connectors>
      ...
    </ssd:Connectors>
    <ssd:ElementGeometry x2="-110" x1="-130"
      y1="-10" y2="10"/>
  </ssd:Component>
</ssd:Elements>

```

Listing 1. Representation of a Modelica component.

```

<ssd:Connectors>
  <ssd:Connector name="y" kind="output"
    description="Connector of Real output
    signal">
    <ssc:Real/>
    <ssd:ConnectorGeometry x="0.5" y="0.0"
      rotation="90" />
  </ssd:Connector>
  <ssd:Connector name="flange_a"
    kind="acausal"
    description="Flange of left shaft">
    <ssc:Binary mime-type="text/x-modelica"
    source="modelica://Modelica.Mechanics.Rotati
    onal.Interfaces.Flange_b"/>
    <ssd:ConnectorGeometry x="0" y="0.5"/>
  </ssd:Connector>
</ssd:Connectors>

```

Listing 2. Representation of Modelica connectors.

```

<ssd:ParameterBindings>
  <ssd:ParameterBinding
    type="application/x-ssp-parameter-set">
    <ssd:ParameterValues>
      <ssv:ParameterSet
        name="DefaultParameters"
        version="1.0">
        <ssv:Parameters>
          <ssv:Parameter
            name="peak">
            <ssv:Real value="1.1"/>
          </ssv:Parameter>
          <ssv:Parameter
            name="startTime">
            <ssv:Enumeration value="2*T2"/>
          </ssv:Parameter>
        </ssv:Parameters>
      </ssv:ParameterSet>
    </ssd:ParameterValues>
  </ssd:ParameterBinding>
</ssd:ParameterBindings>

```

Listing 3. Example of a parameter binding.

Advanced Modelica concepts such as inheritance, replaceable components/models, re-declarations and expandable connectors are intentionally left out because there is no natural mapping to SSP concepts.

6.2 Tools without Modelica capabilities

We can expect that most tools supporting SSP will not have capabilities to simulate Modelica models. Such tools can partially support SSP files that use Modelica components with little additional effort. The information to display components and their connectors, as well as connections, is identical. The tool must ignore what it cannot process, such as any component source of type "text/x-modelica". Simple editing operations are possible, assuming that the new attributes described in this document are ignored but maintained.

Note that in either case, SSPs that do not include Modelica components are completely unchanged compared to SSP 1.0. In that sense, this is an unobtrusive extension.

6.3 Enumeration expressions

The proposal to handle parameter expressions as enumerations can be questioned. Reusing the concept of enumeration values can be perceived as a creative abuse of the rules, but appears to fall within the constraints of SSP. A cleaner solution would be to introduce a new kind of value for this case, but that adds another incompatibility with SSP 1.0. A further generalization would be to use Modelica's full modifier syntax, which would allow e.g. redeclaration.

It has been proposed to generalize the allowed string for e.g. Real values to include an arbitrary expression. This is not a good idea because it defeats the possibility to syntax-check purely numeric parameters sets.

6.4 Change proposal or layered standard

When developing a proposed extension of SSP, one is faced with the choice of two approaches.

The first is to make an extension that is as close as possible to SSP 1.0 with minimum disruption for existing tools and users. SSP is designed to manage this, using the concept of Layered Standard that uses a general extension mechanism in the form of annotations. Using such annotations, it is possible to make a layered standard that can (in limited form) be processed by conforming tools that know nothing at all about new features.

The second is to make an extension proposal designed to cleanly extend SSP 1.0 into SSP 2.0 with new concepts that naturally fit into SSP. In this case, we need to add attributes that are not present in SSP 1.0 instead of using annotations. Examples are the proposed rotation attribute and the notion of acausal connectors. The drawback is that tools that strictly conform to SSP 1.0 will not be able to

read the new format, which for that reason should be identified with a unique version number.

After due consideration we have respectfully opted for the second approach, a non-layered extension. The key reason is the future growth path with a potentially wide application. If this feature will be a permanent part of SSP 2.0, we want it to be “natively” integrated and not be implemented with annotations. If we started with a layered standard based on SSP 1.0, the native representation in SSP 2.0 would require yet another migration effort. Some aspects of the proposal, such as the rotation attribute for connectors, are not inherently tied to Modelica.

Acknowledgements

The author wants to thank the members of the SSP design group, in particular Pierre Mai, Robert Hällqvist and Peter Lobner, for constructive feedback on the proposal.

References

- Brück, Dag (2023). "SSP in a Modelica Environment" in Proc. 15th International Modelica Conference, Aachen, Germany.
- Elmqvist, Hilding (2014). "Modelica Evolution - From My Perspective" in Proc. 10th Modelica Conference, Lund, Sweden.
- Heinkel, Hans-Martin, P. R. Mai, R. Aue, J. Bou, C. Bühler, C. Franke and A. Puetz (2022). "SRMD format and classifications for metadata". https://gitlab.setlevel.de/open/processes_and_traceability/traceability_data/-/blob/main/SETLevel_SRMD_and_classifications_for_metadata.pdf.
- Heinkel, Hans-Martin and K. Steinkirchner (2022). "Credible Simulation Process". https://gitlab.setlevel.de/open/processes_and_traceability/credible_simulation_process_framework/-/blob/main/Credible-Simulation-Process-v1-2.pdf.
- Hällqvist, Robert, R. C. Munjulury, R. Braun, M. Eek and P. Krus (2021). "Engineering Domain Interoperability Using the System Structure and Parameterization (SSP) Standard" in Proc. of the 14th International Modelica Conference, Linköping, Sweden.
- Mai, Pierre R. et al. (2019). "System Structure and Parameterization". <https://ssp-standard.org/publications/SSP10/SystemStructureAndParameterization10.pdf>.
- MLS36 (2023). "Modelica – A Unified Object-Oriented Language for Systems Modeling, Language Specification Version 3.6".

```

<?xml version="1.0" encoding="UTF-8"?>
<ssd:SystemStructureDescription fileversion="4.0.0" generationDateAndTime="2023-08-15T14:16:21Z"
  generationTool="Dymola by Dassault Systemes" name="CoupledClutches" version="2.0-alpha.1" >
  <ssd:SystemDescription description="Drive train with 3 coupled clutches" name="CoupledClutches">
    <ssd:Connectors>
      <ssd:Connector description="Frequency to invoke clutch1" kind="parameter" name="f">
        <ssc:Binary mime-type="text/x-modelica"
          source="modelica://Modelica.Units.SI.Frequency"/>
        </ssd:Connector>
        ...
      </ssd:Connectors>
      <ssd:Elements>
        <ssd:Component name="sin2" source="modelica://Modelica.Blocks.Sources.Sine" type="text/x-
          modelica">
          <ssd:Connectors>
            <ssd:Connector description="..." kind="output" name="y">
              <ssc:Real/> <ssd:ConnectorGeometry x="1.05" y="0.5"/>
            </ssd:Connector>
          </ssd:Connectors>
          <ssd:ElementGeometry rotation="270" x1="-40" x2="-20" y1="30" y2="50"/>
          <ssd:ParameterBindings>
            <ssd:ParameterBinding type="application/x-ssp-parameter-set">
              <ssd:ParameterValues>
                <ssv:ParameterSet name="DefaultParameters" version="1.0">
                  <ssv:Parameters>
                    <ssv:Parameter name="amplitude"> <ssv:Real value="1"/> </ssv:Parameter>
                    <ssv:Parameter name="f"> <ssv:Enumeration value="f"/> </ssv:Parameter>
                    <ssv:Parameter name="phase"> <ssv:Real value="1.570796326794897"/>
                  </ssv:Parameter>
                </ssv:Parameters>
              </ssv:ParameterSet>
            </ssd:ParameterValues>
          </ssd:ParameterBinding>
        </ssd:ParameterBindings>
      </ssd:Component>
      ...
    </ssd:Elements>
    <ssd:Connections>
      <ssd:Connection endConnector="flange_a" endElement="J1"
        startConnector="flange" startElement="torque"> </ssd:Connection>
      ...
    </ssd:Connections>
    <ssd:ParameterBindings>
      <ssd:ParameterBinding type="application/x-ssp-parameter-set">
        <ssd:ParameterValues>
          <ssv:ParameterSet name="DefaultParameters" version="1.0">
            <ssv:Parameters>
              <ssv:Parameter name="f"> <ssv:Real value="0.2"/> </ssv:Parameter>
              <ssv:Parameter name="T2"> <ssv:Real value="0.4"/> </ssv:Parameter>
              <ssv:Parameter name="T3"> <ssv:Real value="0.9"/> </ssv:Parameter>
            </ssv:Parameters>
          </ssv:ParameterSet>
        </ssd:ParameterValues>
      </ssd:ParameterBinding>
    </ssd:ParameterBindings>
    <ssd:SystemGeometry x1="-140" x2="140" y1="-100" y2="100"/>
    <ssd:Annotations>
      <ssc:Annotation type="com.3ds.ssp">
        <smmd:ModelicaMetaData>
          <UserAnnotation key="description" value="Drive train with 3 coupled clutches"/>
          <UserAnnotation key="name" value="CoupledClutches"/>
          <UserAnnotation key="version" value="4.0.0"/>
          <UserAnnotation key="versionDate" value="2020-06-04"/>
          <UserAnnotation key="revisionId" value="6626538a2 2020-06-04 19:56:34 +0200"/>
        </smmd:ModelicaMetaData>
      </ssc:Annotation>
    </ssd:Annotations>
  </ssd:System>
  <ssd:DefaultExperiment startTime="0" stopTime="1.5"/>
</ssd:SystemStructureDescription>

```

Listing 4. CoupledClutches example from MSL with certain parts removed for brevity.