Massif Simulink Ecore Documentation

BME-FTSRG

January 29, 2015

1 Introduction

In this document we include a detailed documentation of the Simulink Ecore model. The documentation is generated directly from the metamodel definitions using model-to-text code generation. This approach has multiple advantages, (i) the documentation is available when viewing the metamodel, (ii) the EMF code generator includes the documentation in the JavaDoc of the model code and (iii) any changes in the metamodel will appear in the documentation by simply running the model-to-text code generator.

Structure

The following is a short description of the structure used in documenting the metamodels.

EPackage

The documentation section of each metamodel begins with a short introduction explaining the rationale behind the given metamodel and lists the main concepts that are identified and represented by the elements in the metamodel. This introduction is stored in the main EPackage that contains the elements of the metamodel. Finally, the namespace prefix and URI used when persisting instance models of the metamodel are listed.

Elements

After the metamodel introduction, each element in the Package has a subsection which starts with the description of the element, followed by various element properties and tables depending on the type of the element.

EClass There are two properties and three tables that can appear for an EClass element. The **Abstract** property specifies that instance models cannot contain EObjects that are instances of this EClass, instead subtypes should be used. The **Interface** property specifies that no implementation is generated for the EClass. Finally, the list of **Supertypes** is given.

The tables list the (i) **attributes**, (ii) **references** and (iii) **operations** of the EClass, if there are any. Each table has three columns, the first contains the name of the element (colored blue if derived), the second a list of properties and the third any documentation that is given. Apart from clarifications and rationale, this documentation part can also include validation rules that should be satisfied by instance models (with violations appearing as either errors or warnings).

Some properties are common to the tables, these are the **Type (T)** and the **Cardinality** of the element and whether the values are **Unordered** or **Not unique** (the default values **Ordered** and **Unique** are not shown). In the case of operations, the type and cardinality refers to the return value.

The properties specific to features (attributes and references) are **Non-changeable** (cannot be modified directly), **Volatile** (value not stored in a field), **Transient** (value not persisted), **Unsettable** (has Unset value) and **Derived** (value is computed), again, only non-default values are listed.

The property specific to attributes is **Identifier**, which is true if the value of the attribute identifies the EObject.

The reference-specific properties are **Opposite (Op)** (the name of the reference in the target that will refer back to the source), **Containment** (true, if target EObject of this reference is persisted inside source) and **Container** (Containment is true for opposite of reference). Finally, the property specific to operations is **Parameters** (any number of parameters with type and cardinality).

EEnum   There is only one table that appears additionally to the element description. The table lists the **literals** of the EEnum and contains three columns, the name, value and documentation of the literal.

2 The Simulink Ecore model

The Simulink EPackage represents the core concepts of Matlab Simulink models with a strong focus on the structure and less focus on the behavior, simulation and layout specific details.

The five main concepts (easily recognised by SimulinkElement and its subtypes) are the following:

- Elements on a Simulink model that can be identified and named are subtypes of SimulinkElement, which stores the identifier as a SimulinkReference element.

- The root element, SimulinkModel, stores the file path and version for the original Simulink system it represents to help in handling changes in the represented system.

- The Simulink model contains a hierarchy of Block elements that may have properties and specify a source block from a Simulink library. The source block is set if the internal structure and behavior of the block is defined by a library block.

- The communication between blocks is done through Port elements, that can be either output or input. Each port is represented by a PortBlock inside the block.

- The output ports are connected to input ports using Connection elements that can be one-to-one single connections or one-to-many multiconnections.

EPackage properties:
   Namespace Prefix: `simulink`
   Namespace URI: `http://hu.bme.mit.massif/simulink/1.0`

2.1 Block

This EClass represents the basic building block of Simulink systems. Each block has properties, ports and can refer to a source block that was used as the template from a library to create the element.

The properties are not a map, so the block may have multiple properties with the same name, or even same name-value pair.

The different type of ports are accessible through computed filtered lists.

EClass properties:
   Supertypes: `SimulinkElement`

References		
Name	Properties	Documentation
enabler	**T:** `Enable` **Cardinality:** [0..1] **Non-changeable** **Volatile** **Transient** **Derived**	The values are computed by filtering Enable ports from the values of the ports feature.
inports	**T:** `InPort` **Cardinality:** [0..*] **Non-changeable** **Volatile** **Transient** **Derived**	The values are computed by filtering Inports from the values of the ports feature.
outports	**T:** `OutPort` **Cardinality:** [0..*] **Non-changeable** **Volatile** **Transient** **Derived**	The values are computed by filtering Outports from the values of the ports feature.
parent	**T:** `SubSystem` **Cardinality:** [0..1] **Container** **Op:** `subBlocks`	
ports	**T:** `Port` **Cardinality:** [0..*] **Containment** **Op:** `container`	
properties	**T:** `Property` **Cardinality:** [0..*] **Containment**	
sourceBlock	**T:** `Block` **Cardinality:** [0..1] **Non-changeable** **Volatile** **Transient** **Derived**	The value of the reference is computed by finding the element with the same name and qualifier as stored in the sourceBlockRef. **Validation** • WARN: Cannot find block specified by source block reference.
sourceBlockRef	**T:** `LibraryLinkReference` **Cardinality:** [0..1] **Containment**	
trigger	**T:** `Trigger` **Cardinality:** [0..1] **Non-changeable** **Volatile** **Transient** **Derived**	The values are computed by filtering Trigger ports from the values of the ports feature.

Table 1: References of the Block EClass

2.2 BusCreator

This EClass represents a bus creator block that bundles the signals on its inports into a bus on its outport.

See `http://www.mathworks.com/help/simulink/slref/buscreator.html`

EClass properties:
    Supertypes: `BusSpecification`

2.3   BusSelector

This EClass represents a bus selector block that separates the signals from the bus received on its inport into its outports. Since it is possible to select only some of the signals and even embedded signals from a bus inside the bus, mapping entries (BusSignalMapping) are used to indicate which outport (mappingTo) selects which signals originating from a given outport (mappingFrom).

The outputAsBus attribute is true if the selected signals are bundled into a bus and placed on a single outport.

See `http://www.mathworks.com/help/simulink/slref/busselector.html`

EClass properties:
    Supertypes: `BusSpecification`

Attributes		
Name	Properties	Documentation
`outputAsBus`	**T:** `EBoolean` **Cardinality:** [0..1]	

Table 2: Attributes of the BusSelector EClass

References		
Name	Properties	Documentation
`busCreator`	**T:** `BusSpecification` **Cardinality:** [0..1]	This reference points to the creator of the incoming bus signal. Either a bus creator or a bus selector with outputAsBus = true. Always determined as backward navigation on signals is deterministic.
`mappings`	**T:** `BusSignalMapping` **Cardinality:** [0..*] **Containment** **Op:** `selector`	

Table 3: References of the BusSelector EClass

2.4   BusSignalMapping

This EClass represents a signal mapping entry in the BusSelector to define which signals are selected from a bus.

Attributes		
Name	Properties	Documentation
incomplete	**T:** EBoolean **Cardinality:** [0..1]	A true value indicates that the map-pingFrom OutPort is not the real source of the selected signal, when the actual source is not part of the model.
mappingPath	**T:** EString **Cardinality:** [0..1]	

Table 4: Attributes of the BusSignalMapping EClass

References		
Name	Properties	Documentation
mappingFrom	**T:** OutPort **Cardinality:** [0..1]	**Validation** • ERROR: Selected port is not an outport of a signal bundled in the bus • ERROR: Multiple mappings from the same outport
mappingTo	**T:** OutPort **Cardinality:** [0..1]	**Validation** • ERROR: Selected port is not an outport of selector • ERROR: Multiple mappings to the same outport
selector	**T:** BusSelector **Cardinality:** [0..1] **Container** **Op:** mappings	

Table 5: References of the BusSignalMapping EClass

2.5 BusSpecification

This EClass is an abstract supertype for blocks that handle bus signals. A bus signal is used for bundling a set of signals into one signal to reduce the number of ports and connections required in the model.

EClass properties:
Abstract
Supertypes: Block

2.6 Connection

This EClass represents the connection between Block elements in order to transfer data from an outport to one or more inports.

EClass properties:
Abstract

Supertypes: `SimulinkElement`

Attributes		
Name	Properties	Documentation
`lineName`	**T:** `EString` **Cardinality:** [0..1]	

Table 6: Attributes of the Connection EClass

References		
Name	Properties	Documentation
`from`	**T:** `OutPort` **Cardinality:** [0..1] **Container** **Op:** `connection`	

Table 7: References of the Connection EClass

2.7   Enable

This EClass represents an enable port of a Block.
See `http://www.mathworks.com/help/simulink/slref/enable.html`

EClass properties:
   Supertypes: `InPort`

Attributes		
Name	Properties	Documentation
`statesWhenEnabling`	**T:** `EnableStates` **Cardinality:** [0..1]	

Table 8: Attributes of the Enable EClass

2.8   EnableBlock

This EClass represents a port block of a Enable port with an outport that can be used by blocks inside the subsystem.
See `http://www.mathworks.com/help/simulink/slref/enable.html`

EClass properties:
   Supertypes: `InPortBlock`

2.9   EnableStates

This EEnum represents the possible settings of a Enable port for specifying what happens to the states of blocks in the enabled system upon disabling.
See `http://www.mathworks.com/help/simulink/slref/enable.html`

Literals		
Name	Value	Documentation
Held	0	
Reset	1	
Inherit	2	

Table 9: Literals of the EnableStates EEnum

2.10 From

The From block accepts a signal from a corresponding Goto block, then passes it as output. The data type of the output is the same as that of the input from the Goto block. From and Goto blocks allow you to pass a signal from one block to another without actually connecting them.

See http://www.mathworks.com/help/simulink/slref/from.html

EClass properties:
    Supertypes: VirtualBlock

References		
Name	Properties	Documentation
gotoBlock	**T:** Goto **Cardinality:** [0..1] **Op:** fromBlocks	

Table 10: References of the From EClass

2.11 Goto

The Goto block passes its input to its corresponding From blocks. From and Goto blocks allow you to pass a signal from one block to another without actually connecting them.

See http://www.mathworks.com/help/simulink/slref/goto.html

EClass properties:
    Supertypes: VirtualBlock

Attributes		
Name	Properties	Documentation
gotoTag	**T:** EString **Cardinality:** [0..1]	
tagVisibility	**T:** TagVisibility **Cardinality:** [0..1]	

Table 11: Attributes of the Goto EClass

References		
Name	Properties	Documentation
fromBlocks	**T:** From **Cardinality:** [0..*] **Op:** gotoBlock	

Table 12: References of the Goto EClass

2.12 GotoTagVisibility

The Goto Tag Visibility block defines the accessibility of Goto block tags that have scoped visibility. The tag specified as the Goto tag parameter is accessible by From blocks in the same subsystem that contains the Goto Tag Visibility block and in subsystems below it in the model hierarchy.

See http://www.mathworks.com/help/simulink/slref/gototagvisibility.html

EClass properties:
   Supertypes: VirtualBlock

References		
Name	Properties	Documentation
gotoBlock	**T:** Goto **Cardinality:** [0..1]	

Table 13: References of the GotoTagVisibility EClass

2.13 IdentifierReference

This is a specific class used as a unique identifier for Simulink elements.

EClass properties:
   Supertypes: SimulinkReference

2.14 InPort

This EClass represents an inport of a Block.
   See http://www.mathworks.com/help/simulink/slref/inport.html

EClass properties:
   Supertypes: Port

2.15 InPortBlock

This EClass represents a port block of a InPort with an outport that can be used by blocks inside the subsystem.
   See http://www.mathworks.com/help/simulink/slref/inport.html

References		
Name	Properties	Documentation
`connection`	**T:** `SingleConnection` **Cardinality:** [0..1] **Op:** `to`	

Table 14: References of the InPort EClass

EClass properties:
 Supertypes: `PortBlock`

2.16   LibraryLinkReference

This is a specific class used for representing links to Simulink elements.

EClass properties:
 Supertypes: `SimulinkReference`

Attributes		
Name	Properties	Documentation
`disabled`	**T:** `EBoolean` **Cardinality:** [0..1]	Disabled links mean that the block was originally copied from a library but it was modified later.

Table 15: Attributes of the LibraryLinkReference EClass

2.17   ModelReference

This EClass represents a SimulinkModel included as a block in this model.
 See http://www.mathworks.com/help/simulink/slref/model.html

EClass properties:
 Supertypes: `Block`

2.18   MultiConnection

This EClass represents a connection between a single OutPort and multiple InPort. Each inport is connected by a SingleConnection contained by this connection.

EClass properties:
 Supertypes: `Connection`

2.19   OutPort

This EClass represents an outport of a Block.
 See http://www.mathworks.com/help/simulink/slref/outport.html

References		
Name	Properties	Documentation
modelRef	**T:** `IdentifierReference` **Cardinality:** [0..1] **Containment**	
referencedModel	**T:** `SimulinkModel` **Cardinality:** [0..1] **Non-changeable** **Volatile** **Transient** **Derived**	The value of the reference is computed by finding the model with the same name and qualifier as stored in the modelRef. **Validation** • WARN: Cannot find model specified by model reference.

Table 16: References of the ModelReference EClass

References		
Name	Properties	Documentation
connections	**T:** `SingleConnection` **Cardinality:** [0..*] **Containment** **Op:** `parent`	

Table 17: References of the MultiConnection EClass

EClass properties:
    Supertypes: `Port`

References		
Name	Properties	Documentation
connection	**T:** `Connection` **Cardinality:** [0..1] **Containment** **Op:** `from`	

Table 18: References of the OutPort EClass

2.20 OutPortBlock

This EClass represents a port block of a OutPort with an inport that can be used by blocks inside the subsystem.
    See `http://www.mathworks.com/help/simulink/slref/outport.html`

EClass properties:
    Supertypes: `PortBlock`

2.21 Port

This EClass represents the abstract supertype of block ports that are used for allowing data communication and signaling between blocks.

EClass properties:
  Abstract
  Supertypes: `SimulinkElement`

References		
Name	Properties	Documentation
`container`	**T:** `Block` **Cardinality:** [0..1] **Container** **Op:** `ports`	
`portBlock`	**T:** `PortBlock` **Cardinality:** [0..1] **Op:** `port`	

Table 19: References of the Port EClass

2.22 PortBlock

This EClass represents the abstract supertype of blocks that represent ports of a subsystem.

EClass properties:
  Abstract
  Supertypes: `VirtualBlock`

References		
Name	Properties	Documentation
`port`	**T:** `Port` **Cardinality:** [0..1] **Op:** `portBlock`	

Table 20: References of the PortBlock EClass

2.23 Property

This EClass represents properties of Block elements. Each property has a name, a type and a value. The value is stored as a character string but is validated based on the type.

2.24 PropertySource

This EEnum represents the possible sources of properties in a Simulink element.

2.25 PropertyType

This EEnum represents the possible value types of Property elements.

Attributes		
Name	Properties	Documentation
name	**T:** `EString` **Cardinality:** [0..1]	
source	**T:** `PropertySource` **Cardinality:** [0..1]	
type	**T:** `PropertyType` **Cardinality:** [0..1]	
value	**T:** `EString` **Cardinality:** [0..1]	**Validation** • ERROR: Value type is not compatible with property type.

Table 21: Attributes of the Property EClass

Literals		
Name	Value	Documentation
MASK	0	
DIALOG	1	

Table 22: Literals of the PropertySource EEnum

Literals		
Name	Value	Documentation
StringProperty	0	
IntegerProperty	1	
DoubleProperty	2	

Table 23: Literals of the PropertyType EEnum

2.26 SimulinkElement

This EClass represents the abstract supertype of elements in Simulink systems that can be identified uniquely with a fully qualified name consisting of a name and a qualifier.

EClass properties:
  Abstract

Attributes		
Name	Properties	Documentation
name	**T:** `EString` **Cardinality:** [0..1] **Non-changeable** **Volatile** **Transient** **Derived**	The value is computed from the name feature of the SimulinkReference stored in simulinkRef.

Table 24: Attributes of the SimulinkElement EClass

References		
Name	Properties	Documentation
`simulinkRef`	**T:** `IdentifierReference` **Cardinality:** [0..1] **Containment**	

<div align="center">Table 25: References of the SimulinkElement EClass</div>

2.27 SimulinkModel

This EClass represents the root of an imported Simulink system that contains blocks. The version and file path informations are stored to allow change management. The library attribute stores whether the model was a library or a regular model in Matlab.

EClass properties:
  Supertypes: `SimulinkElement`

Attributes		
Name	Properties	Documentation
`file`	**T:** `EString` **Cardinality:** [0..1]	
`library`	**T:** `EBoolean` **Cardinality:** [0..1]	
`version`	**T:** `EString` **Cardinality:** [1..1]	

<div align="center">Table 26: Attributes of the SimulinkModel EClass</div>

References		
Name	Properties	Documentation
`contains`	**T:** `Block` **Cardinality:** [0..*] **Containment**	

<div align="center">Table 27: References of the SimulinkModel EClass</div>

2.28 SimulinkReference

This abstract EClass represents a reference for a Simulink element. The identifier is a fully qualified name constructed from a qualifier (the fully qualified name of the parent of the element) and a name. Since a SimulinkElement can be identified based on its fully qualified name, it is possible to reference an element by cloning and storing the reference instead of a direct link to the element itself.

EClass properties:
  Abstract

Attributes		
Name	Properties	Documentation
name	**T:** EString **Cardinality:** [0..1]	
qualifier	**T:** EString **Cardinality:** [0..1]	

<div align="center">Table 28: Attributes of the SimulinkReference EClass</div>

References		
Name	Properties	Documentation
element	**T:** SimulinkElement **Cardinality:** [0..1]	

<div align="center">Table 29: References of the SimulinkReference EClass</div>

Operations		
Name	Properties	Documentation
getFQN	**T:** EString **Cardinality:** [0..1]	

<div align="center">Table 30: Operations of the SimulinkReference EClass</div>

2.29   SingleConnection

This EClass represents a simple connections between a single OutPort and a single InPort. The outport can be identified either through the from reference or through the parent MultiConnection.

EClass properties:
   Supertypes: Connection

References		
Name	Properties	Documentation
parent	**T:** MultiConnection **Cardinality:** [0..1] **Container** **Op:** connections	
to	**T:** InPort **Cardinality:** [0..1] **Op:** connection	

<div align="center">Table 31: References of the SingleConnection EClass</div>

2.30   SubSystem

This EClass represents a Simulink block that may contain subblocks that specify its internal structure and behavior.

   See http://www.mathworks.com/help/simulink/slref/subsystem.html

EClass properties:
    Supertypes: `Block`

Attributes		
Name	Properties	Documentation
`tag`	**T:** `EString` **Cardinality:** [0..1]	

Table 32: Attributes of the SubSystem EClass

References		
Name	Properties	Documentation
`subBlocks`	**T:** `Block` **Cardinality:** [0..*] **Containment** **Op:** `parent`	

Table 33: References of the SubSystem EClass

2.31 TagVisibility

The Tag Visibility is a parameter of Goto blocks to determine the location of From blocks that access the signal.
    See `http://www.mathworks.com/help/simulink/slref/goto.html`

Literals		
Name	Value	Documentation
`Local`	0	
`Scoped`	1	
`Global`	2	

Table 34: Literals of the TagVisibility EEnum

2.32 Trigger

This EClass represents a trigger port of a Block.
    See `http://www.mathworks.com/help/simulink/slref/trigger.html`

EClass properties:
    Supertypes: `InPort`

2.33 TriggerBlock

This EClass represents a port block of a Trigger port with an outport that can be used by blocks inside the subsystem.
    See `http://www.mathworks.com/help/simulink/slref/trigger.html`

Attributes		
Name	Properties	Documentation
statesWhenEnabling	**T:** EnableStates **Cardinality:** [0..1]	
triggerType	**T:** TriggerType **Cardinality:** [0..1]	

Table 35: Attributes of the Trigger EClass

EClass properties:
    Supertypes: InPortBlock

2.34 TriggerType

This EEnum represents the possible events that can trigger the execution of a subsystem with a Trigger port.

    See http://www.mathworks.com/help/simulink/slref/trigger.html

Literals		
Name	Value	Documentation
Rising	0	
Falling	1	
Either	2	
FunctionCall	3	

Table 36: Literals of the TriggerType EEnum

2.35 VirtualBlock

This EClass represents the abstract supertype of blocks that do not explicitly affect the simulation of the Simulink system. These blocks are called virtual and are added as syntactic sugar, for example Goto and From can be used instead of a direct Connection to connect blocks.

EClass properties:
    Abstract
    Supertypes: Block