

The Shlaer-Mellor Metamodel

Here is an overview map of the model subsystems defining the Shlaer-Mellor Metamodel.

Taken together, these subsystems define executable, platform independent modeling semantics that can be expressed with a variety of notations including, but not exclusive to Executable UML and SysML.

The content here is maintained and described in the github [modelint/shlaer-mellor-metamodel](#) repository

You'll find all of the documentation in the repository wiki

Underneath each subsystem rectangle is a brief description with a list of its key classes to the right

DOMAIN MODEL

Domain

How we organize an entire system and identify subsystem and spanning model elements

System
Modeled Domain
Realized Domain
Subsystem
Element

CLASS MODEL

Class and Attribute

How we define classes and attributes

Class definition
Attributes and Identifiers
Other Attribute roles

Relationship

The different kinds of relationships among classes

Relationship definition
Types of relationships:

Association with Perspective, and Association Class
Generalization with Facet and Lineage
Ordinal

Constrained Loop

Formalization

How we use referential attributes to formalize Association and Generalization Relationships

STATE MODEL

State

How we define a state model

Lifecycle and Assigner
State
Transition
Event Reponse

Event

How an event is specified

Event Specification
Monomorphic event (normal events)
Polymorphic event

ACTION MODEL

Activity

How Activities are defined and associated with model elements

State Activity
Method
Signature
Parameter

Action

All of the Action building blocks that can be assembled to define an Activity as well as the flow dependencies forming the flow graph of that Activity

Action
Instance / Relation Action
Flow Dependency
Wave

Flow

Types of data and control that flows in an Activity

Control Flow
Instance Flow
Relation Flow
Scalar Flow

Navigation

Defines the Traverse Action and the anatomy of a Path through the Relationships of a class model

Traverse Action
Path
Hop

Signal Action

Defines the Signal Action which emits a signal to various kinds of destinations

Signal Action
Initial Signal Action
Signal Assigner Action
Interaction Signal Action
Delivery Time

Create Delete

Defines actions to create and delete an instance of a Class

Create Action
Attribute Initialization
Delete Action

External Service

How we define operations and events triggered in an external domain

External Service
External Event
External Operation

Select Action

Defines the Select Action which selects one or more instances of a Class

Select Action
Identifier Select
Many Select

Method Call

Defines an Action that invokes a Method on some instance

Method Call
Method Call Output
Method Call Parameter

Relational Action

Defines the supported Relational Actions

Restrict Action
Project Action
Restriction Condition
Rename Action
Set Action
Extend Action

Type

Type definition is outside the scope of Shlaer-Mellor semantics, but here we do register the externally defined names of types as well as supported type operations along with any invocations of those operations

Type
Type Action
Type Operation
Scalar
Table Attribute

Iteration

Defines an action that iterates through a sequence of model elements (no, these are not for loops!)

Iteration Action
Sequence
Sorting Attribute

EXECUTION

Execution

All of the other subsystem specify static model elements, but here is where execution on all of those elements is defined

With all the hard work done in those in that multitude of semantic specification subsystems, our work here is greatly simplified to the point we need only one small subsystem to define the platform independent execution mechanics.

State Machine
Instance
Activity Execution
Dispatched Event

MIT Open Source License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



October 5, 2025 / v2.0.1

Model Integration, LLC

Copyright © 2025, Leon Starr