# 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 repsitory

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

# CLASS MODEL

DOMAIN MODEL

Modeled Domain

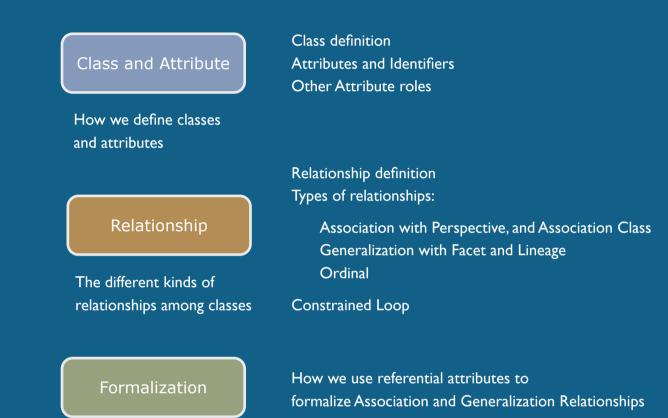
Realized Domain

Subsystem

Element

Domain

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



### STATE MODEL

State Transition Event Reponse

How we define a state model

How an event is specified

Event Specification Monomorphic event (normal events) Polymorphic event

Lifecycle and Assigner

### ACTION MODEL

State Activity Method Activity Signature

How Activities are defined and associated with model elements

Action Instance / Relation Action Flow Dependency

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

Control Flow Instance Flow Relation Flow

Scalar Flow

Types of data and control that flows in an Activity

> Signal Action Initial Signal Action Signal Assigner Action Interaction Signal Action **Delivery Time**

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

External Service

**External Service** External Event

How we define operations and events triggered in an external domain

> Method Call Method Call Output Method Call Method Call Parameter

Defines an Action that invokes a Method on some instance

Traverse Action Path Defines the Traverse Action and the

anatomy of a Path through the Relationships of a class model

> Create Action Attribute Initialization Delete Action

Defines actions to create and delete an instance of a Class

Select Action Identifier Select Select Action Many Select

Defines the Select Action which selects

Set Action

Defines the supported Relational Actions

Type Action Type Operation

Table Attribute

Scalar

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

Computation Action Computation Input

Defines the Computation Action which inputs multiple Data Flows, applies a mathematical or logical expression and outputs a single Scalar result

Read / Write Action Attribute Access

Defines actions to read and write the value of an Attribute

Control Dependency

Sequence Flow

Defines Actions that output Control Flows to enable other Actions

Iteration Action Sequence Sorting Attribute

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

Restrict Action

Project Action

Rename Action

Extend Action

Restriction Condition

# EXECUTION

State Machine



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.

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.

