**OME Compatibility With Simile**

Version: PreRelesase

Last Revised: June 18, 2015

Author: Patrick Wingo

For OME to be of any initial use, the projected needed to target models that already existed. Both reference models were written and executed in the Simile modeling framework (see the Literature Review section for a brief description of Simile). Since all model update expressions and model components beyond the basic universal System Dynamics pieces were Simile-centric, it is pertinent to reimplement behaviors and structural pieces that the models rely on to work. What follows is a description of how the Simile's model structure meshes with that defined by OME, the extent of OME model expression function compatibility with that found in Simile, and a few functions defined for OME's expression syntax which have no equivalent in Simile.

Table 1 shows how Simile model components map to their equivalents in OME. The compatibility between Simile and OME components has fluctuated over time, with the biggest shift occurring when focus on supporting dynamic allocation and deallocation of submodel instances was reduced in priority in favor of ensuring that the static model behavior in the reference models would be sufficiently supported within a reasonable timeframe. Increasing compatibility with Simile, as well as adding support for System Dynamics models that originate from other environments, are goals for future development.

|  |  |  |  |
| --- | --- | --- | --- |
| **Simile Component** | **Supported?** | **OME Equivalent** | **Notes** |
| Alarm | No | IterConditional | Implementation was started, but never completed. |
| Compartment | Yes | StateVar |  |
| Condition | Yes | Spawner | Functionality consolidated into Spawner class. |
| Creation Process | Partial | Spawner | Functionality consolidated into Spawner class. Functionality is present, but likely broken; has not been tested in a while. |
| Destruction Process | Partial | Spawner | Functionality consolidated into Spawner class. Functionality is present, but likely broken; has not been tested in a while. |
| Event | No | N/A | Discrete Event injection is not supported. |
| Flow | Yes | Flow |  |
| Ghost | Yes | EvalAlias | Ghosts are mapped to EvalAliases, which largely function the same way. This component is only used for data representation and is note evaluated during runtime. |
| Immigration Process | Partial | Spawner | Functionality consolidated into Spawner class. Functionality is present, but likely broken; has not been tested in a while. |
| Influence | Yes | Influence |  |
| Reproduction Process | Partial | Spawner | Functionality consolidated into Spawner class. Functionality is present, but likely broken; has not been tested in a while. |
| Role Arrow | Partial | SubmodelAssoc | Functionality is present, but has not been thoroughly tested. |
| Squirt | No | N/A | Discrete Event injection is not supported. |
| State | No | N/A | Discrete Event injection is not supported. |

**Table 1.** . The list of model components are taken from Simile 6.x.

OLE-object

**Table 1 (Continued).**

Since OME is expected to understand Simile-derived System Dynamics models, most, if not all expression functions in Simile would need to be reimplemented in OME. This was done mostly through trial-and-error, with testing primarily focused on the functions used in reference models. Table 2 outlines the degree of compatibility between Simile and OME expression functions.

OLE-object

**Table 2.** .

**OLE-object**

**Table 2 (Continued).**

OLE-object

**Table 2 (Continued).**

The following are a handful of OME-specific expression functions which have no equivalent Simile implementations.

**valuesFromInstances(<variable>,<inds>)** - Retrieves a list of values for **<variable>** populated with the value from the parent model instances pointed to by the indexes in the list **<inds>**.

**getAsTable(…) -** Takes a variable number of arguments and returns a list object which has packaged the values.

**upgroup(<variable>,<level>) -** Retrieve a list of values for **<variable>** by up-scoping **<level>** number of submodels. Primarily used to reconcile operations on model components that exist at different submodel depths

**omecleanup() -** Deallocation function which should not be called directly.