




Name	Description	Placeholder
File	An icon to represent the file which contains the model information. Should be a composite of a file and the package icon.	
Package	Top-level of our architecture, represents a container inside which everything else exists.	
Service Set	A set of service elements, should look like a collection / bundle / set.	
Component	Represents an executable, runtime application.	
Service	Building block of the architecture, represents a collection of states and messages. Typically a gear or set of gears.	
Internal Event	Represents a message which originates from inside the application and stays within the application.	
Message	Represents a message which originates from inside the application and stays within the application.	
State Machine	A state machine represents a series of states which the service can exist in. Typically these are modeled as a series of circles with curved lines connecting them.	
State	A state is one of a set of possible states which a service can be in. Typically modeled as a circle or box with lines into and out of it for transitions.	
Transition	Represents the connector between two states with a start point and end point.	
Loopback Transition	Special transition which is a loop.	
Push Transition	Special transition which pushes a value onto a stack.	None
Pop Transition	Special transition which pops a value from a stack.	None
State Machine Runner	Represents a thread which is excuted to process messages from the state machine.	
Action	Generic action which may change the state of the service.	
Send Action	Action which results in a Message being sent from the application to another application.	
Entry Action	Action which occurs upon entry into a given State.	
Exit Action	Action which occurs upon exit from a given State.	
Action Set	A collection of actions which occur together.	
Condition	Conditions represent guards and questions on actions, could be represented by a question mark or a shield as well.	

And Guard	Represents the logical operation AND. Should look like the C/C++ implementation of the operator “&&”	&&
Or Guard	Represents the logical operation OR... should look like the C/C++ implementation of the operator “ ”	
Not Guard	Represents the logical operation NOT... should look like the C/C++ implementation of the operator “!”	!
Declared Fields	Collection of Field objects.	
Field	Building block of all messages, fields represent programming elements like byte, short, string, blob, etc.	
Upper Limit	A maximum value for a field.	↑
Lower Limit	A minimum value for a field.	↓
Type	Type is used to define fields which may have different values at runtime. Some form of the letter “T” is acceptable as long as it fits the look and feel of the icon set.	T
Dimension	Dimension is the size of an Array element. Should be some form of the “[]” notation used in programming languages which fits the look and feel of the icon set.	[]