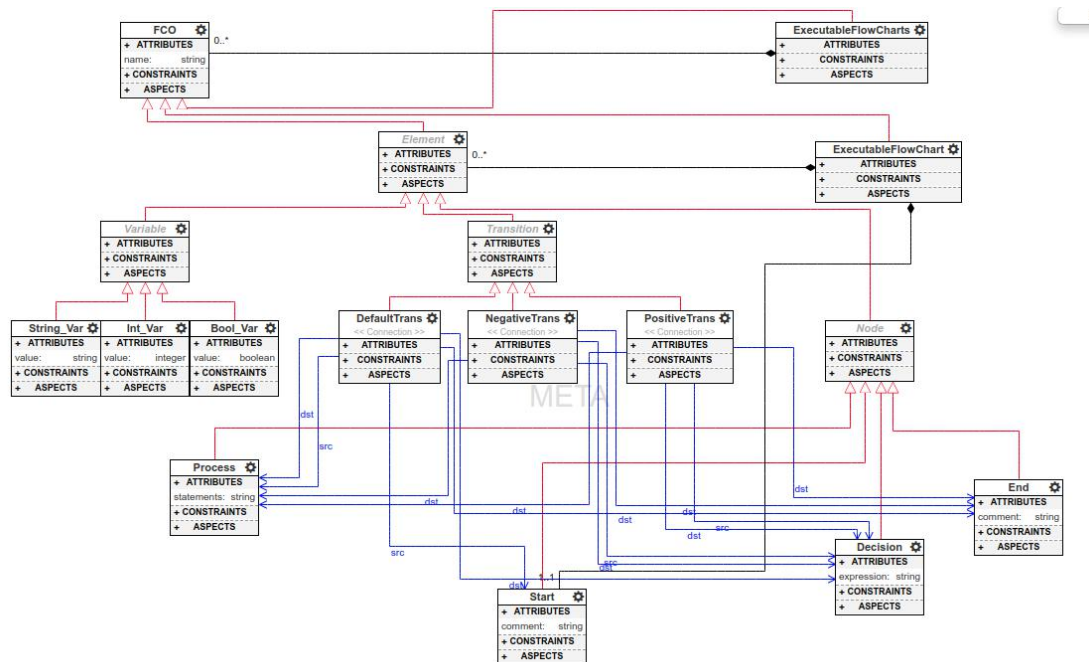


# Executable Flowchart in WebGME

## 1. Meta model



There are three types of elements in the meta model:

- Variable**: the elements that changes their value during the execution and indicate status of the whole chart. There are 3 kinds of variables: int, boolean and string
- Node**: the points of the execution. There are 4 kinds of nodes. Start is the unique one in a chart. Processes have the attribute 'statements' which is a string of Javascript statements which manipulate the variables. Decisions have the attribute 'expression' which is a Javascript true/false expression. During the execution, if the expression result is true, it go to the positive out flow, otherwise to the negative out flow.
- Transition**: the elements that connect one node to another. There are 3 kinds of transition. Default transition is used for out flow of a process. Negative and positive transitions are for decision out flow based on the result.

## 2. Visualizer

### 2.1 Validity check

In the meta model, only the type of out/in flows of a node are defined, but there is not constraint of number of out/in flow of a node. Before rendering the visualized graph, the validity of the flowchart is checked first. If there is something wrong with

the flowchart (e.g. multiple out flow from one process), there will be error messages showed in the widget.

The visualizer is only applied to Executable Flowchart instances. If it is opened under another type of component, nothing will be showed.

## **2.2 Elements and positioning**

The elements showed in the visualizer are same as those defined in the 'Component' panel. For the variables and nodes, the positions in the visualizer are same as their positions in the 'component'. For transitions, there is different layout logic that makes the lines look better ordered.

The variables are showed with their names and values. The nodes are showed in different shapes depend on their type. Black arrows are default transition, red ones are negative transition and green is for the positive.

## **2.3 Flowchart Execution**

There are two buttons in the widget: reset and trigger. Initially, the trigger button is disabled while variables input slots are available to change the variable in it. You need to click reset to enable the trigger button. When the trigger button is clicked for the first time after it is enabled, the chart is under execution and the variable slots are disabled. You need to enter the values before execution (after reset, before trigger).

During the execution, the current node's edge is highlighted as red. Trigger button will move the state from current node to the next. If a variable is changed in the current node's execution, the variable in the slot will change after the trigger is clicked.

When trigger is clicked when the current node is an end node, the execution will be terminated and the trigger button will be disabled. You can reset, enter the values, and trigger to start again. You can interrupt the execution at any point by clicking reset button to start from the beginning again.

## **3. Code Generation Plugin**

### **3.1 Code Generation**

The plugin name is 'FlowchartCode'. A file named 'simulator.zip' is generated after the plugin runs correctly. There is an index.html with corresponding Javascript file for it which is the main simulation file. Browsing the html file requires Internet connection because the js file need to load online JQuery source.

If the flowchart is not designed correctly, there will be error from the plugin and no file will be generated (similar to the validity check of the visualizer).

### **3.2 Simulation**

Similar to the visualizer, you can click trigger button to move to the next node and input the values of the variables to variable slots. The difference is there is a start button instead of reset. You need to click start button before any execution. Start button will go to the start node and wait for next step to be triggered. In each step, current node and variable information will be printed on the large scroll bar in the middle. Values in the variable slots won't change during the execution. The onclick event of start button will abort current execution, flush the scroll bar output, and read the values in variable slots for a new execution.