The developed monitor module must support monitoring of a program behavior at runtime.

The module has an interface for importing a state-space file (\*.statespace) into the module. The state-space file represents a state transition diagram of the respective program with the defined structure [source-\*]. The module is initialized by the state-space file and produces an output at console (+logging) by observing input signal data. The input signal data feeds to the module using a given port on the computer where the module operates. The module performs monitoring algorithm on input signal data and produces output.

The **output** has the following structure:

[system\_time(computer\_clock)] [executionTime] [present\_state(state\_id)] [input] [next\_state(state\_id)] {[variable\_name: value], [variable\_name:value], …}

The **input** signal data has the following structure:

[event\_name(messageserver\_title)]

[source-\*].

<transitionsystem>

<**state id**="value[int]\_value[int]" atomicpropositions="[string]" >

<rebec name="rebec\_name[string]">

<statevariables>

<**variable name**="rebec\_name[string].variable\_name[string]" type="">**value** [int or boolean]</variable>

</statevariables>

<queue>

<message arrival="value[int]" deadline="value[int]" sender="rebec\_name[string]"> event\_name[string]() </message>

</queue>

<now> value[int]</now>

<pc></pc>

<res></res>

</rebec>

</state>

<transition source="value[int]\_value[int]" destination="value[int]\_value[int]" **executionTime**="value[int]" shift="valur[int]"> <**messageserver** sender="rebec\_name[string]" owner=" rebec\_name[string]" **title**="event\_name[string]"/></transition>

</transitionsystem>

The **monitoring algorithm** on [source-\*].

1. Initiate and identify present state (initial state id="1\_0"),
2. Determine next state using input signal data,
3. Publish output on console (+logging),
4. Set next state as present state, then go to step.2.

The step.2 refers to a transition tag(s) in state-space file. The tag(s) represents which message server title matches the input signal data.

The identified transition(s) determines next state id from present state id. The input signal data cause a change on a value of state variable(s) corresponding to the present state id. If input signal data does not cause a change on a value of state variable(s) of present state id, the next state id is same as the present state id.

In the step.3, in addition to the state id(s), the name and value are of the state variable(s) are written in output. Step.4 is responsible for setting an indicator for the present state and repeating the monitoring at step.2.