1 Plant and Requirement models

This section discusses the files in *1. Plant and Requirement models*. In this folder, 3 folders and a number of files are given. In the folder *Plant_Requirements*, the file *Pumpcellars.cif* contains the plant model and the requirements model. The folder *GUI* contains the file defining the Graphical User Interface and the folder *SimulationModel* contains the simulation files.

1.1 Folder *Plant_Requirements*

The plant contains 5 main components, the 3 pumpcellars and 2 traffic tubes. Each of these components contains the automata, as described in Section III [1] of the paper, these are defined in the folder *Definitions*. Next, some requirements are stated, note that these are not all the requirements. Additional requirements are stated in the component definitions in the folder *Definitions*.

An actuator automaton is modeled as follows in the CIF3 language:

```
1 plant Actuator:
2         controllable c_on, c_off;
3         location Off:
4                 initial;
5                 marked;
6                 edge c_on when x=1 do x:=2 goto On;
7         location On:
8                 edge c_off goto Off;
9 end
```

The equivalent automaton is given in Figure 1.1. The automaton contains 2 controllable events: *c aan* and *c_off*, which are listed behind keyword controllable. Two locations (also referred to as states) are created, a state '*On*' and a state '*Off*'. State '*Off*' is a marked state and the initial state. Keyword *when*, on the edge of event '*c_on*', denotes a guard function, stating that this edge can only be taken if $x = 1$. Keyword *do* denotes an update function, stating that upon taking this edge variable x should be set to 2. Other components are modeled in a similar fashion.

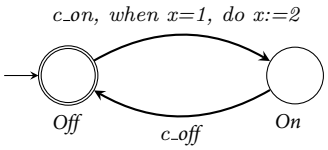

Figure 1.1: actuator automaton

Requirements can be stated in two ways, displayed below. Both types of requirements contain an event, which can be disabled, and a condition for which the event should be disabled. In requirement 1, an event needs a certain condition to be enabled, this condition can refer to variables or one or more states of automata. In requirement 2, a condition, defined in a similar fashion, disables one or more events.

```
1 requirement Event     needs    Condition;
2 requirement Condition disables Event;
```

1.2 Folder *SimulationModel*

Next to plant and requirement models, Folder 1 also contains simulation files in folder *SimulationModel*. These files are used to simulate the waterlevel in the pumpcellars, based on the pumps and rain. The

[1] Roman numberals are used to refer to sections of the paper, arabic numerals are used to refer to sections of this appendix.

simulation outputs sensor values based on the simulated water level. The simulation controls an SVG file, *Pumpcellars.svg*, to visualize the simulation.

The water level of a pumpcellar is modeled using a continuous variable *waterlevel*. The derivative of this variable is defined based on the amount of rain water entering the pumpcellar and the number of pumps that are filling or emptying the pumpcellar. Such a continuous variable is defined as follows:

```
1 cont VariableName = InitialValue der
2         if      ConditionA : DerivativeA
3         elif    ConditionB : DerivativeB
4         else DerivativeC
```

Here the variable *VariableName* has the initial value *InitialValue* and a derivative value of *DerivativeA*, *DerivativeB*, or *DerivativeC*, depending on conditions *ConditionA*, and *ConditionB*.

Based on the simulation, sensor events are enabled or disabled. Moreover, for visualization, entities of the SVG file can be adjusted as follows:

```
1 svgout id SVGtag attr Attribute value
2         if ConditionA: ValueA
3         else ValueB
```

The the attribute *Attribute* of the component in the SVG file with id *SVGtag*, is set to *ValueA* or *ValueB* depending on condition *ConditionA*. Attributes of SVG components are, for example, its color or its opacity.

1.3 Folder *GUI*

In folder *GUI*, the buttons are linked to the SVG file, such that they can be clicked during the simulation. The click of such a button triggers an event in the simulation. The GUI is contructed using SVG input statements. The SVG input statement below states that when clicking the component with id *SVGtag*, event *Event* is triggered.

```
1 svgin id SVGtag event Event
```

1.4 Other files

The file *synthesis.tooldef2*, can be used to run data-based synthesis in CIF3 on the plant and requirement model. The resulting supervisor is given in the file *Supervisor.cif*.

The CIF3 simulator can be used by running the file *Simulation.tooldef2*, which runs a simulation of the system controlled by the synthesized supervisor. The GUI and visualization are implemented using the SVG file.