27 November 2003

Modeling Reactive Systems with Statecharts

Exercise: A Simple Air Conditioning System

The air conditioning system attempts to adjust the temperature in the room to some goal value between MINT and MAXT constants (MINT < MAXT). The goal temperature can be controlled by the user by pressing *TempUp* and *TempDown* buttons, respectively increasing and decreasing the value of temperature by one degree. The speed of air conditioning depends on the rotation speed of the fan installed in it. This speed can be controlled by pressing the *FanUp* and *FanDown* buttons, which respectively increases and decreases the rotation speed by one unit. The rotation speed of the fan should not go below MINF and should not exceed MAXF (MINF < MAXF).

While being *On* the air conditioner operates in three modes: *automatic*, *manual* and *turbo*. The rotation speed can be adjusted only while in *manual* mode, while the goal temperature may be changed at *manual* and *auto* modes. The mode is changed by pressing one of the two buttons: *AutoOn*, *TurboOn*. Upon pressing the button, the new mode is entered and the previous one is exited. As soon as the user presses one of the rotation adjustment buttons, the system enters the *manual* mode. The factory setup default mode is *auto*.

When the system enters the *auto* mode the fan speed is set to the AUTO value (where MINF $\leq$ AUTO $\leq$ MAXF). When the system enters the *turbo* mode, the fan speed is set to the MAXF value and the goal temperature is set to MAXT. However this changes do not affect the values most recently configured by user. As soon as the mode returns to the *manual* value the old values are restored.

The system is turned *on* by pressing the *PowerButton*. The system may be switched *off* at any moment during operation by pressing the *PowerButton* again. When activated again, the system returns to the mode, which was most recently active.

The air conditioner has a display, exhibiting the current value of goal temperature and the current value of fan rotation speed.

Question 1

Translate the above specification of the control algorithm to a statechart model. Assume that the statechart model will be used in an implementation put in parallel with a process controlling the heating and cooling elements. This process will read the value of goal temperature from *GoalTemp* shared variable and the speed rotation from *FanSpeedCurrent* variable. The fan rotation display can be updated by calling a *DisplayFan* function. The goal temperature can be updated by calling the *DisplayTemp* function.

```
void DisplayTemp(int);
void DisplayFan(int);
```

Question 2

Model the same algorithm without using hierarchy (nested states) and entry/exit actions of statecharts (but still using concurrency). Observe how features of statecharts affect succinctness of your model.

Question 3

Try to model the same algorithm without using neither hierarchical features nor concurrency.

Question 4

Use your model from the first or the second question as basis for (manual) implementation of the algorithm in C.