jan.kleinsorge [☺] tu-dortmund.de

Exercises to
Cyber-physical System Fundamentals
Summer term 2014

StateChart-Tutorial

(2 Points)

To be solved in the session on May, 14th.

1 Tutorial (2 Points)

1. Inflate the template file (“tutorial.zip”) that you find on the remote drive (**locally**!). In the folder “vs”, open “visualState.vnw”.

2. Figure 1 shows the title screen of *VisualState*. Select “Modeling” to open the graphical state chart editor.

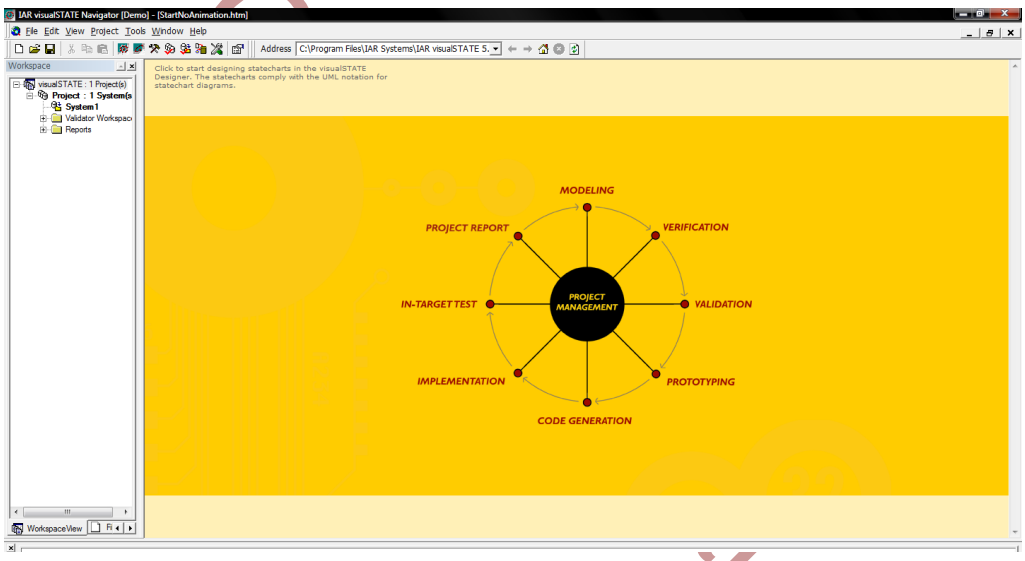

Figure 1: Title Screen

3. Figure 2 shows some design elements of the editor. To construct a state chart select *initial state*, *state* and *transitions* from the left menu bar.

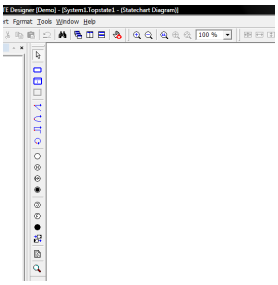

Figure 2: visualState *Designer*

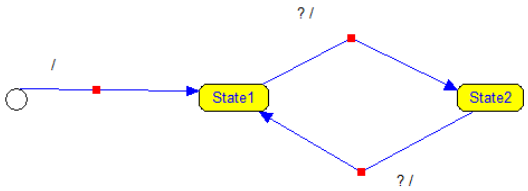

Figure 3: A simple state chart

4. Construct the state chart as depicted in figure 3. Note how this “raw” chart does not model guards, event sensitivity or side-effects yet. (**Hint:** Although state chart editors differ in their syntax for the specification of aforementioned elements, the ones on the left-hand side of the slash (“/”) commonly define preconditions, while ones on the right are postconditions. Note how the arc form the initial state yields no precondition.)

5. To modify a rule, double-click the label (“?/”) above a transition arc. The new dialog window is similar to figure 4. In the left item list, you find different rule elements that can be modified. Once an element is selected on the left, the list on the right shows the feasible options for the specific selection. Double-click an option to add it to the rule.

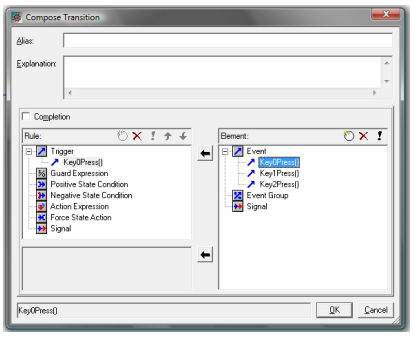

Figure 4: Transition Dialog

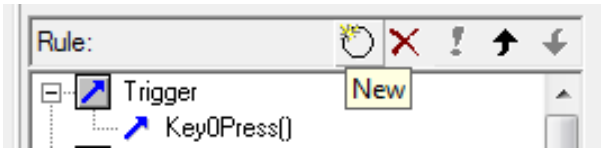

Figure 5: Create a new variable

6. Modify the rule of an arc such that its “Trigger” (precondition) is the event “Key0Press” as shown in the figure. In addition, add an “Action” (side-effect) to the same transition. Select the function “writeLine0(VS_INT number): VS_VOID” which prints an integer value in the first line of the display.

7. Without a function argument, the rule is ill-formed. Add a new integer variable to your model which is supposed to be printed. Select the rule “Action Expression” add press “New” as depicted in Figure 5. In the element list, select “Internal Variable” and press “New” to open a dialog akin to figure 6. Label the variable and select the data type “VS_INT”. The remaining fields on this dialog are designated to documentation, the definition of arrays and range-restricted values, all which is not relevant here.

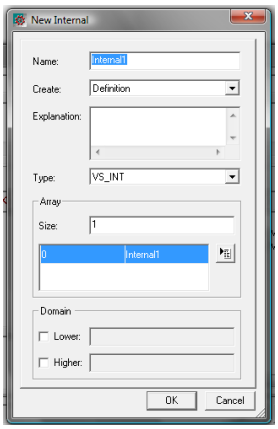

Figure 6: Create new internal variable

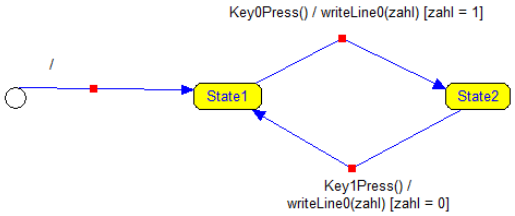

Figure 7: Finished example

8. A new “Action Expression” is now available. Double-click “X= ?” to add an arithmetic expression to the rule. In our case, assign a value to your variable. In addition, associate the variable with “writeLine0” by double-clicking the respective item in the rule list.

9. Complete the state chart to look identical to figure 7 (**hint:** change the order of action expressions, if necessary.).

10. Before actually flashing the resulting implementation on the board, the model ought to be simulated to verify its semantics. Save your project, close the window to get back to the title screen (figure 1) and select “Validation”.

11. In this simulator (figure 8), events can be triggered (double-click in the left panel), transitions (middle panel) and side-effects (right panel) can be observed. Verify that everything performs as intended.

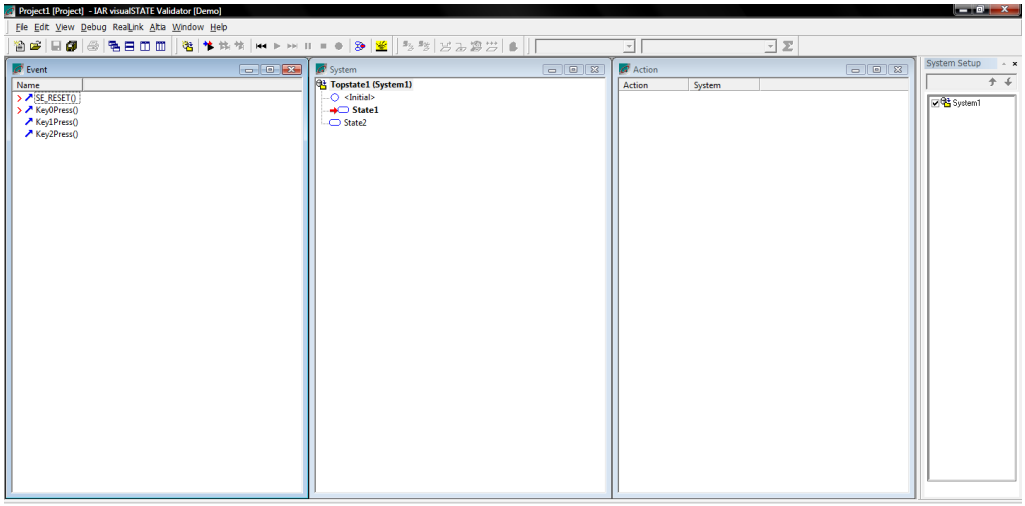

Figure 8: visualState Validator

12. Close the validator and select “Code Generation” from the title screen. This step generates an implementation of your model in the C language. Check the debug log in the output window for warnings or errors before proceeding.

13. Now close Visual State and launch the “Embedded Workbench”. In the template folder, open “ew” and then the project file. EW is basically a general purpose IDE. We use it here to compile our implementation.

14. Click “Project → Make” to compile the source files to a binary image that can be found in the subfolder “Debug/Exe” afterwards.

15. Open “Flash Magic” from the Windows program menu. Depending on your version, the dialog windows looks more or less akin to figure 9. Check “Erase all Flash + Code Rd Prot” and make the following selections:

    - *Port*: COM1
    - *Device*: ARM7 → LPC2103
    - *Interface*: NXT ICP Bridge
    - *Oscillator*: 16MHz

    The *COM* port depends on the system (COM1 usually works). If you are in a virtual machine you first have to connect through the physical device into the guest system. To do that, move your cursor to the bottom of the screen such that small control-panel pops. Under “Devices” check the appropriate device which should then be available.

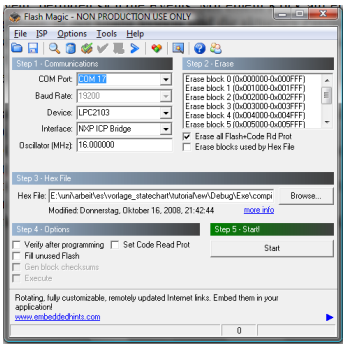

Figure 9: Flash Magic

16. Browse for the binary *DEBUG\Exe\compiler.hex* and press “Start”. UARTs are fairly error prone, so give it a few tries should the flashing process fail. Once this is done, the board is ready for testing.

General notes:

Dates and additional information can be found on the lecture website (can also be found via EWS). The assignments will be typcically be published **Mondays** on a weekly basis and have to be solved in the lab session of the same week. To pass the labs, a minimum of 50% of the total points must be achieved in the first half and the second half, respectively.