Esterel Studio

Esterel-Studio is a IDE for Esterel . It contains a graphical interface to create Safe State Machines, an editor for Esterel and several tools, e.g., for verification.

Startup

Our license for Esterel Studio is only valid inside the domain informatik.uni-kiel.de. Therefore a client runnning Esterel Studio must have a permanent network connection inside that domain to be able to talk to the license server. An independent home installation is not possible.

To start Esterel Studio just call

> */home/esterel/bin/estudio*

and the graphical modeling tool appears.

Observe that Esterel Studio uses Esterel V7.

Examples

Examples from Esterel Technologies

The example project files delivered with Esterel Studio are located in

> */home/esterel/EsterelStudio/examples*

They are organized in three hierarchies with different grades of complexity: "Basic", "Intermediate", and "Advanced":

```
/home/esterel/EsterelStudio/examples/Basic/Abro
/home/esterel/EsterelStudio/examples/Basic/MooreMealy
/home/esterel/EsterelStudio/examples/Basic/SimpleRam
/home/esterel/EsterelStudio/examples/Intermediate/CRC
/home/esterel/EsterelStudio/examples/Intermediate/Dopar
/home/esterel/EsterelStudio/examples/Intermediate/Excel
/home/esterel/EsterelStudio/examples/Intermediate/Fifo11
/home/esterel/EsterelStudio/examples/Intermediate/ReflexGame
/home/esterel/EsterelStudio/examples/Intermediate/RegisterBank
/home/esterel/EsterelStudio/examples/Advanced/Elevator
/home/esterel/EsterelStudio/examples/Advanced/Fifo22
/home/esterel/EsterelStudio/examples/Advanced/Wristwatch
```

To try them out, copy a projects directory into a working directory of yours and start Esterel Studio:

```
 mkdir abro
 cd abro
 cp -r /home/esterel/EsterelStudio/examples/Basic/Abro/* .
 /home/esterel/bin/estudio abro.etp &
```

Locally Prepared Examples

There are two example directories prepared for use as templates for your own projects:

- /home/esterel/estudio/examples/
```
      Native compilation and execution on the host (Unix/Linux) architecture.
          o Makefile:    Input file for the make command.
          o ABRO.etp:    Project file for Esterel Studio.
          o ABRO.scg:    SyncChart file containing the ABRO machine.
          o ABRO.h:       C header file with interface declarations.
          o ABRO_data.c:    C code with functions for signal interfacing and keyboard control.
```
- /home/esterel/estudio/example-legOS/
```
      Crosscompilation for the LEGO Mindstorms system with the alternative operating system legOS.
          o Makefile:    Input file for the make command.
          o ABRO.etp:    Project file for Esterel Studio.
          o ABRO.scg:    SyncChart file containing the ABRO machine.
          o ABRO.h:       C header file with interface declarations.
          o ABRO_data.c:    C code with functions for data processing and legOS calls.
```

Compilation

The compilation of a project into textual Esterel and C code is started by

> *Project->Generate Code*

The code for the former ABRO example is placed in the following directories (relative to the working directory):

```
Default/Code/Temp/ABRO.strl
Default/Code/ABRO_strl.h
Default/Code/ABRO.c
```

It is possible to perform this task on the command line without startup of the full graphical interface:

> */home/esterel/bin/estudio --embedded ABRO.etp*

To compile the resulting Esterel file into C code execute:

> */home/esterel/bin/esterel7 -v7 -Lc -W Default/Code/Temp/ABRO.strl*

Generate VHDL Code

To Generate VHDL create a new "Configuration" with a handy name like "VHDL" and choose "VHDL" as "Target Language". Follow the *Compilation* steps.

Notes:

- due to the fact that VHDL is case insensitive, beware of naming your identifier case sensitive
- read the guide: *Generating efficient hardware with Esterel V7 and Esterel Studio*: use whenever possible **temp** and **value**-only signals to reduce the amount of registers
- If you use BlockRam feature of the FPGA board and you get strange deadlock problems, it might help to add a **NOT** to the clock input pin or toggle the behavior of the clock input from **falling** to **rising** or v.v..

Makefile for the Unix target

The Makefile in /home/esterel/estudio/example compiles esterel programs for the Unix-Target.

The following targets are available (replace ABRO with the base name of your project):

- make ABRO compiles and links the Esterel file ABRO.strl along with ABRO_data.c into an executable file ABRO.
- make ABRO.c If you are only interested in the C code generated from the Esterel file you can use this target. It just calls the Esterel compiler to produce the C code ABRO.c from ABRO.strl.

Makefile for the legOS target

The Makefile in /home/esterel/estudio/example-legOS compiles esterel programs for the legOS-Target. It is similar to the one for Unix with some additional targets for kernel and program upload into the RCX.

The following targets are available (replace ABRO-rcx with the base name of your project):

- make ABRO.lx compiles and links the Esterel file ABRO.strl along with ABRO_data.c into a binary image ABRO.lx. This image can be uploaded into the RCX to be executed.
- make ABRO.c If you are only interested in the C code generated from the Esterel file you can use this target. It just calls the Esterel compiler to produce the C code ABRO.c from ABRO.strl.
- make ABRO.dll uploads the binary image ABRO.lx into the RCX. If ABRO.lx is not existant or not up to date, then it is rebuilt.
- make kernel just uploads the kernel into the RCX (e.g. after a crash)

Documentation

Several documents are available: Manuals for EsterelStudio and textual Esterel. For copyright reasons we can not provide HTTP links to these files. Therefore they are only locally available in the directory /home/esterel/EsterelStudio/doc.

Further information can be found at [www.esterel-technolgies.de Esterel Technologies].

Back-Animation

To animate a Safe State Machine while the program is executed, the esi file format can be used.

- the input function should print the input whenever it calls a input function of the SSM
- after every step of the automaton, a ";" should be printed
- when the program is executed, select Simulation->ConnectToFile and play