YaFSM UsersGuide

by Jörg Kreuzberger

24.03.10

Inhaltsverzeichnis

1Usage..3
2XML File...3
3Usage of the generator..3
4Compilation...3
5The YaFsmViewer..4

1 Usage

The YaFSM generator is a perl script bundle that is used to generate

- code
- descriptions
- images

from a single xml based FSM description.

To use it simple call the main perl script along with the command line arguments.

The complete argument list could be printed with the –man or –help command line option

```
YaFsm.pl –verbose
```

2 XML File

Input for the YaFSM generator is a xml description of the state engine. Details on the xml tags are given in Apppendix xx. You can edit the file with any text editor you like or you could use the YaFsmEditor (under development).

To verify the xml file you can test it agains the given xsd schema file ,that can be found in the xml directory.

On Linux you could use xmllint, on windows e.g. xmlstarlet

```
xmllint -s Yafsm.xsd <yourfsm.xml>
```

On Linux you could use xmllint, on windows e.g. xmlstarlet

3 Usage of the generator

An example for use of the generator is given in src/perl/CmakeLists.txt

```
perl -f YaFsm.pl --fsm=<xml-file> --genview --gencode
```

This will generate the source code and viewer input files into subdirectories of the current directoy.

You can also give the name of the output directory via cmd line. To list all available options just call

```
perl -f YaFsm.pl --help
```

4 Compilation

The generated source code will not compile if used out of the example build tree, cause some include and source code files are missing. These files must be copied „by hand“ or by our build environment to the generated source code or e.g linked by a library. This has the advantage that you could use our own implementation and framework for the timer and event handling. Provided is an implementation for Qt4 (full featured) and a standard C++ implementation (no timers and events could be used in the fsm)

5 The YaFsmViewer

The provided YaFsmViewer files are generated into a seperated directory. This directory contains several files for each state hierarchy level

- <state>.dot:

  Generated dot file for the state, use to generate html maps and image files with Graphviz dot

- <state>.html:

  Html file with a map definition to display in the viewers QWebKit View

- <state>.<GeneratedImageType>:

  Generated image file used by the map, default are SVG image files have full zooming capability in the viewer

To view the files in the viewer, just open the generated file „index.txt“ in the viewer or start the viewer with the file path from commandline.