## Introduction

Timed Rebeca model-checking tool (RMC) is used to analyze a Timed Rebeca model. First, you should use RMC to map the input Timed Rebeca model to a set of C++ files. Then, generated C++ files should be compiled and run. Running the C++ files will produce the result of model checking. The result includes the number of states and transitions in the generated state space, the status of deadlock-freedom and deadline misses in your model. In the following parts, we explain how RMC can be used for the analysis of a Timed Rebeca model.

## How to generate C++ files

All the required libraries for generating C++ files are included in a Java executable Jar file "rmc-2.5.0-SNAPSHOT.jar". You can run this file with JRE 1.6 or upper using the following parameters.

| Parameter | Example | Description |
|---|---|---|
| -s --source | -s /home/test/model.rebeca<br>-s myModel.rebeca | Location of Timed Rebeca source file |
| -o --output | -o /home/test/modelFolder<br>-o myModelFolder | Target of generated C++ file |
| -v --version | -v 2.1 | Features of Timed Rebeca is enabled in version 2.1. |
| -x | -x | Exporting the state space in XML format in "statespace.xml" file. |
| -e --extension | -e TimedRebeca | Enables model checking of Timed Rebeca models |
| -h | -h | Print the parameters description |

Here is an example of using RMC for analyzing a Timed Rebeca model:

java -jar rmc-2.5.0-SNAPSHOT.jar -s model.rebeca -o output-folder -v 2.1 -e TimedRebeca -x

The output of the above command is a set of C++ files. These files are stored in output-folder.

## How to execute the generated C++ files

The resulting C++ files can be compiled using any distribution of C++ compilers. In the following example we use g++ to compile the generated C++ files and set the compilation output file to "executable".

g++ *.cpp -w -o executable

Running "executable" file generates model checking results for this model.