Using *timedreb2erl* version 0.3

Department of Computer Science
Reykjavik University, Iceland.

February 24, 2012

1 Prerequisites

1.1 Installing Erlang

Before you are able to run translated code from timedreb2erl, you have to have Erlang R12B-5 or greater installed.
To install follow these simple steps:

Windows:

1. Download Erlang/OTP Windows binaries from http://www.erlang.org.
   This tutorial assumes that you are downloading R15B binaries.
2. Install Erlang by running otp_win32_R15B.exe.
3. Follow the instruction and finish the installation.
   Notice that the binaries are only available for x86, so it will install to c:\program files (x86)\erl5.9\ on x64 versions of Windows.[1]
4. (Windows XP) Go to "Control Panel" and click "System". From there go to the "Advanced" tab and click "Environment Variables".
5. (Windows 7 / Vista) Go to "Control Panel" and click "System". From there go to the "Advanced system settings" link (placed on the left of the panel) and click "Environment Variables".

[1] Is my Windows 32 or 64 bits: https://help.ru.is/index.php?/Knowledgebase/Article/View/49/1/staff-is-my-windows-32-or-64-bit

6. Add Erlang to **PATH**, by clicking on the **PATH** variable from the top list and by pressing **Edit**. There you can add the string ";c:\program files (x86)\erl5.6\bin" at the end of the variable (*note that for 32bit versions of Windows you will need to omit "(x86)"*). Note that ";" separates locations inside the **PATH** variable.

7. Click OK, to stop editing and you are finished.

1.2 Installing *timedreb2erl*

1.2.1 Obtain *timedreb2erl* binaries

Binaries for *timedreb2erl* is available from http://en.ru.is/icerose, in the "tools" section.
When downloaded, you should have a compressed ZIP archive, containing the folder "timedreb2erl-0.3". This folder has:

- bin (The binaries)
- timedreb2erl-0.3 (Support data)
- examples (Examples)

1.2.2 Installing binaries and support data

To install binaries and support data for *timedreb2erl*, simply place the folder "timedreb2erl-0.3" to C:\ (Root of your OS disk).
After that you can choose one of the following approaches to run *timedreb2erl*:

- Always use full reference c:\timedreb2erl\bin\timedreb2erl.exe, to translate models.
- Add c:\timedreb2erl\bin to **PATH** by adding the string ";c:\timedreb2erl-0.3\bin" at the end of the variable (using the method above) and be able to run timedreb2erl from anywhere.

2 Running *timedreb2erl*

After successfully implementing the steps in former chapter, you should be able to run the following commands without errors (giving that timedreb2erl.exe is in PATH):

- erl (CTRL-Z, to exit shell)
- erlc (Should give an empty output)
- timedreb2erl –help (gives possible arguments for timedreb2erl)

2.1 Arguments for *timedreb2erl*

- -s (Simulate with McErlang *This needs McErlang. This is not explained in this tutorial)
- -m (Generate a monitor to be used with McErlang. This is not explained in this tutorial)
- -r (Decides what time factor is to be used, default 1000 (1 sec). This is not explained in this tutorial)
- -e (Displays a graphical event trace while simulating)
- -o (Output directory for the translated Erlang code)

2.2 Translate models

As an example we will use the model "simplecommunication.rebeca".
To translate an timed Rebeca model, you can use the following syntax:

- timedreb2erl.exe **-o out** simplecommunication.rebeca

This will generate Erlang code to the folder **out**\.

2.3 Simulate models

After being successful in generating the Erlang code for the model you need to compile the code with the following syntax:

- cd out\ (Given that the output folder is "out")
- erlc simplecommunication.erl
- erlc rebeca.erl

Note, that you can ignore all warnings containing "unused variables". Though "Unbound" errors means that something is wrong with the model.

If compiling the model was successful (should see *.beam files), you are ready to simulate the model with the following syntax:

- erl -run simplecommunication main 1 1 -noshell

Note, that the two integers following main, are environment integers that are being passed to the model.

If the model is successfully simulated you should see the following output:

```
Type;ReceivedDate;SentDate;Failed;Rebec;MsgSrv;Parm
system;1329959668688002;1329959668688000;0;receiverObj;initial;[]
system;1329959668688003;1329959668688001;0;senderObj;initial;[]
system;1329959668688005;1329959668688004;0;senderObj;start;[]
system;1329959669780000;1329959668778000;0;receiverObj;send;[]
system;1329959676780001;1329959669780001;0;senderObj;ack;[]
system;1329959676780002;1329959676780000;0;senderObj;checkAck;[]
.
.
.
```

2.3.1 Saving simulations

Output of *timedreb2erl* is basically a CSV format, thus it is very easy to pipe the simulation results to a file, like this:

- erl -run simplecommunication main 1 1 -noshell > simple.output.txt

Then you can look at the output with your favorite editor by opening simple.output.txt.