

Event-B Support for OpenComRTOS applications

Group C

Miguel Costa Fábio Sousa

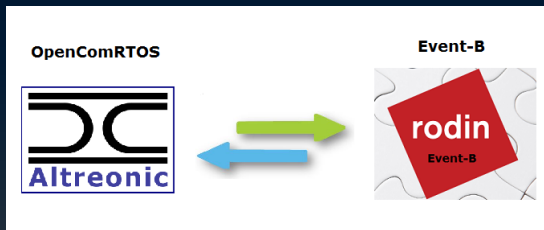
University of Minho
MFES 2013/2014

July 3, 2014

- **Final Goal**
- Reminder
- From the Last Milestone
- LTL Properties
- Convert Rodin to Visual Designer
- Bidirectionality
- From Visual Designer to Rodin
- Demonstration
- Conclusion and Future Work

Final Goal

The main goal of this project is to have a bidirectional connection between OpenComRTOS Visual Designer and Rodin platform.



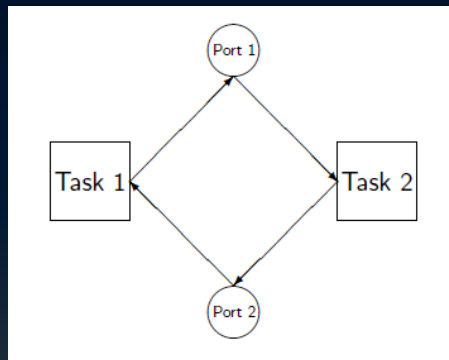
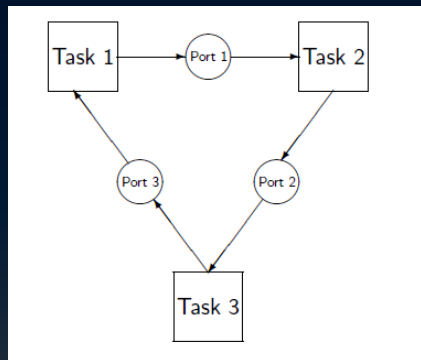
By modelling elements from OpenComRTOS Visual Designer undesired behaviours have been found regarding the connections between tasks and their methods of synchronisation.

The platform used to the Event-B language was Rodin.

- Final Goal
- **Reminder**
- From the Last Milestone
- LTL Properties
- Convert Rodin to Visual Designer
- Bidirectionality
- From Visual Designer to Rodin
- Demonstration
- Conclusion and Future Work

Reminder

Analyses of Diagrams



Two small examples that did impulse our project development.

Reminder

Code Generators

As stated in the previous milestone code generators were not working correctly.

The code generators available at the moment are:

- Tasking Event-B
- B2C
- EB2ALL

- Final Goal
- Reminder
- **From the Last Milestone**
- LTL Properties
- Convert Rodin to Visual Designer
- Bidirectionality
- From Visual Designer to Rodin
- Demonstration
- Conclusion and Future Work

From the last Milestone we set the follow objectives:

- Add other properties to be checked using LTL;
- Observe the behaviour and capabilities of our tool;
- Investigate other possibilities in converting Rodin into OpenComRTOS Visual Designer.

- Final Goal
- Reminder
- From the Last Milestone
- **LTL Properties**
- Convert Rodin to Visual Designer
- Bidirectionality
- From Visual Designer to Rodin
- Demonstration
- Conclusion and Future Work

All the necessary conditions to be checked are related to the behavior of a system communication, in other words is necessary to use Temporal Logic.

All the necessary conditions to be checked are related to the behavior of a system communication, in other words is necessary to use Temporal Logic.

To check such properties was intended to use ProB plug-in in Rodin, but were encountered some barriers:

All the necessary conditions to be checked are related to the behavior of a system communication, in other words is necessary to use Temporal Logic.

To check such properties was intended to use ProB plug-in in Rodin, but were encountered some barriers:

- *(Almost)* Non-Existent documentation.

All the necessary conditions to be checked are related to the behavior of a system communication, in other words is necessary to use Temporal Logic.

To check such properties was intended to use ProB plug-in in Rodin, but were encountered some barriers:

- *(Almost)* Non-Existent documentation.
- All attempts to check properties result in the same type of counter-example. The Set be empty.

LTL Properties

Tools capabilities

An abstract example of property that would be needed to check is:

LTL example

$$G ((\forall t . \text{TaskStatus}(t) == \text{wait}) ==> \\ F(\text{event}(\text{PortSync with } t)))$$

LTL Properties

Tools capabilities

An abstract example of property that would be needed to check is:

LTL example

$$G ((\forall t . \text{TaskStatus}(t) == \text{wait}) ==> \\ F(\text{event}(\text{PortSync with } t)))$$

But without the use of LTL properties, the undesired behaviour that is possible to predict is the "full lock" of the system. Which is represented by the nonexistence of event that can be triggered.

- Final Goal
- Reminder
- From the Last Milestone
- LTL Properties
- **Convert Rodin to Visual Designer**
- Bidirectionality
- From Visual Designer to Rodin
- Demonstration
- Conclusion and Future Work

Convert Rodin to Visual Designer

Was suggested by our co-supervisor to try to make the transference of the models made in Rodin to another application, such as Atelier B and thus try to generate C code directly from there.

Unfortunately Atelier B don't allow the importation of Rodin models, because Atelier B don't support Event-B.

- Final Goal
- Reminder
- From the Last Milestone
- LTL Properties
- Convert Rodin to Visual Designer
- **Bidirectionality**
- From Visual Designer to Rodin
- Demonstration
- Conclusion and Future Work

Bidirectionality

The bidirectionality would be accomplished by transforming OpenComRTOS code to Event-B code and vice-versa.

Unfortunately, in the time this project was developed, is not possible to create the bridge between this two applications, because the conversion from Rodin to Visual Designer was not possible.



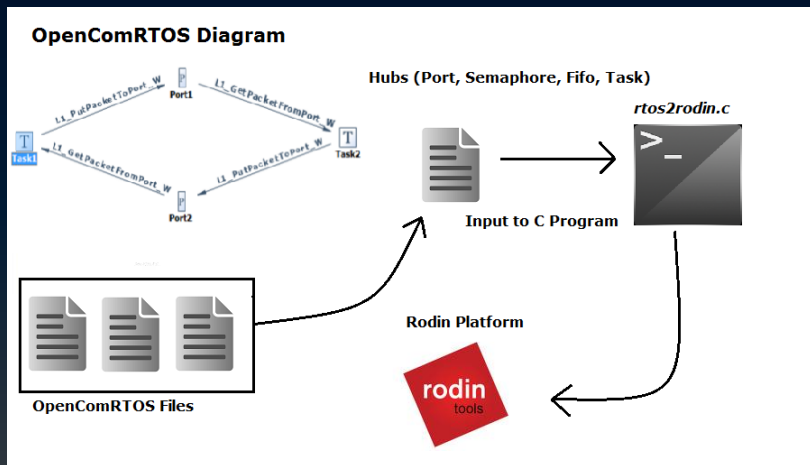
- Final Goal
- Reminder
- From the Last Milestone
- LTL Properties
- Convert Rodin to Visual Designer
- Bidirectionality
- **From Visual Designer to Rodin**
- Demonstration
- Conclusion and Future Work

To allow the conversion of OpenComRTOS Visual Designer projects into Rodin models two programs have been created.

- 1 *rtos2rodin* (C language program)
- 2 *data_script* (Bash script)

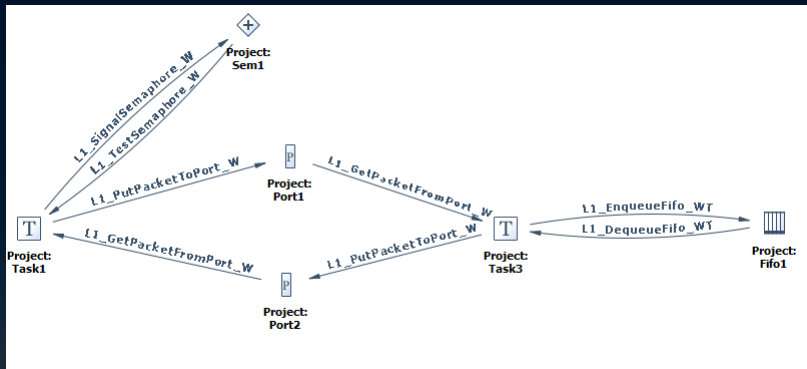
From Visual Designer to Rodin

Functional Behaviour



- Final Goal
- Reminder
- From the Last Milestone
- LTL Properties
- Convert Rodin to Visual Designer
- Bidirectionality
- From Visual Designer to Rodin
- **Demonstration**
- Conclusion and Future Work

Demonstration



This is the diagram of the example that will be used in the demonstration.

- Final Goal
- Reminder
- From the Last Milestone
- LTL Properties
- Convert Rodin to Visual Designer
- Bidirectionality
- From Visual Designer to Rodin
- Demonstration
- **Conclusion and Future Work**

We were unable to create the bridge between this two applications because of the lack of working plug-ins at the time of this project.

What remains to be done:

- Model the remaining Hubs.
- Add the feature to recognise the connection names and system calls from XMLs created by OpenComRTOS Visual Designer.
- Overcome barriers founded in Rodin Plug-ins and the work that comes with it.

Special Thanks

This milestone presentation goes beyond the conclusion and future work.

We have added this extra section to thank Altreonic for giving us the opportunity of working in such an interesting project. We also want to thank our supervisor (José Nuno Oliveira) and co-supervisor (Leonel Braga) for the amazing effort of following our project so closely and for giving us really good tips and a really good help to the development of this project.

¿ Questions ?

