Exercise Sheet 3: Motor Controller

1 Description of Work

The aim of this exercise is to make you familiar with handling a model development with Rodin.

2 An Event-B Model: Motor Controller

2.1 Requirement Document

In this exercise, we study the model of a motor system. Figure 1 gives an overview of the system.

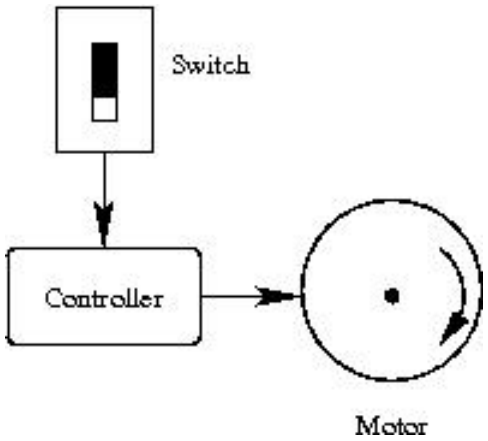

Fig. 1. A motor controller

There are a controller, a motor, and a switch.	EQP-1

The motor is either *stopped* or *working*.	EQP-2

The switch is either *on* or *off*.	EQP-3

When the switch is on, then the motor can, but does not have to change its state to working.	FUN-1

When the switch is off, then the motor can, but does not have to change its state to stopped.	FUN-2

The time diagrams in Fig. 2 and Fig. 3 illustrate these two functional requirements. The solid line is the state of the switch whereas the dotted line is the state of the motor.

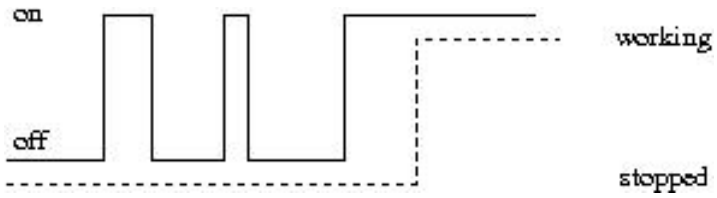

Fig. 2. Start motor

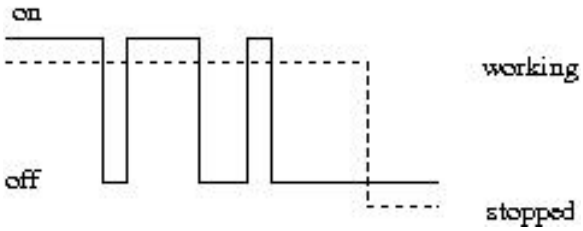

Fig. 3. Stop motor

The number of times the motor has been working so far is always less than or equal to the number of times the switch has changed from off to on..	FUN-3

2.2 Event-B Model:

There is a Rodin archive, called `03_motor1.zip`. The archive contains an Event-B model of the motor system. Import this archive, and have a look at the corresponding development.

3 Your Task

3.1 Animation:

Animate the model with `AnimB`. Play around a bit and try to understand the meaning of the several elements in the machine `m0` and the context `c0`.

3.2 Faithfulness

For each requirement, say how it is enforced by the model. For each assumption, say how it is represented in the model. One or two sentences per requirement / assumption should suffice. Example Solution:

EQP 1: The state of the motor is represented by the variable `m`, and the state of the switch by `s`. The controller is not directly represented in the model.

3.3 Proof Obligations

From the `Event-B Explorer` you can access several proof obligations. A proof obligation consists of several *hypotheses*, listed in the upper window, and a *goal*, displayed in the lower window. Informally a proof obligation states: "given that all the hypotheses hold, the goal is also true".

1. Try to find out how the various proof obligations are related to the model.
2. Open the proof obligation `motor_start/inv5/INV`. Since the smiley is red, it has not been proved. Do you think it can be proved? Explain your answer in one or two sentences.
3. Find a new invariant that will solve the problem.
4. New proof obligations should appear, and all the proof obligations should be discharged, i.e., displayed with the green smiley. Find out the reasons.

3.4 Extending the requirement document

Change the model of the motor system so that it incorporates the following additional assumptions and requirements:

When the switch is on and the motor stopped, the user cannot change the switch to off.	FUN-4

When the switch is off and the motor working, the user cannot change the switch to on.	FUN-5

When the switch is on and the motor working, the number of times the switch has been on so far equals the number of times the motor has been working.	FUN-6

Add invariants so that all the proof obligations are discharged.