Design and Analysis of Distributed Interacting Systems
Summerterm 2013
Assignment 9

Note: Solutions can be submitted in groups of up to three students.

To be submitted on July 4, 2013, 8am via e-mail.

Submissions and further questions to Raphael Pham:
Raphael.Pham@inf.uni-hannover.de

Exercise 1. The Play-Out Algorithm (8 Marks)
Figure 1 shows an MSD specification for the lecture’s RailCab example. env is the environment object and detectingRC, current, warnedRC are the system objects. The idea is the following: when a RailCab detects an obstacle that is blocking the tracks, it must inform its current track section control and report the position of the obstacle. The track section control must in turn report the obstacle and its position to another RailCab (the warned RailCab). (The example is somewhat artificial for the purpose of this exercise. Also for sending the information of an obstacle position, we would normally use a message parameter, but we skip this for simplicity.)

a. (5 Mark) There are different ways that play-out can execute system messages in reaction to the environment event in this example. Especially, it may be that play-out gets “stuck”. This can happen if there are enabled executed messages, but executing any of them would lead to a safety violation. This is called a safety violating execution of the play-out algorithm. List the different play-out reactions. Either list the sequence of message events that lead to a state where the play-out algorithm waits for the next environment event to occur or where the play-out algorithms is stuck. If a cold violation occurs, indicate which event causes it. (Hint: There is more than one execution where play-out would not get stuck. You can abbreviate the message events with oD, hO, hW, oAP, oP.)

b. (3 Marks) Extend the MSD specification by another MSD that will guarantee that play-out will avoid the safety violating execution.
Figure 1: An MSD specification for reporting obstacles on the train tracks.