Lecture Design and Analysis of Distributed, Interacting Systems (DIS)

Assignment 2:

*Solutions to the assignment sheets must be submitted in groups of two or three students. (No submissions by just one student—they will not be graded!)*

Issued on April 19, 2013
To be submitted on April 28, 2013, via e-mail!

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

**Exercise 1:** Requirements Engineering (6 Marks)

Consider a railway crossing as sketched in Figure 1. Trains drive along the track from left to right where they pass sensors installed in the track bed. These sensors detect trains approaching, entering or exiting the crossing and communicate the respective events to a controller. The controller is responsible for closing and opening the barriers of the crossing to prevent cars from entering the crossing.

![Figure 1: Sketch of the crossing.](image-url)

a) Elaborate scenarios that describe what you think can, must, or must not happen in the crossing system. (Be creative and think of how you would expect a crossing to work!) Give at least four positive and at least two negative scenarios that you think are helpful to communicate what can, must, or must not happen in the crossing system. Elaborate more, if you need more for the subsequent tasks. (You can hand in the additional scenarios, but don’t need to. Clearly highlight the scenarios that are the additional ones).

b) Perform an environment analysis and draw a context diagram (as shown in the lecture). Name the components and state which components belong to the software part or non-software part of the system, and which components belong to the relevant context of the system. Furthermore, include all the actions in the system and its context, and document what components shares which actions. Denote if actions are shared, unshared, controllable, or uncontrollable (as explained in the lecture). Make a table in addition to the context diagram if the diagram becomes too cluttered otherwise.
c) Give examples for liveness and safety properties in the crossing system. Describe them in informal text. Which properties belong to the system requirements \((R)\), the software specification \((S)\), or are domain knowledge \((K)\)? Give at least one safety and one liveness property that fall into each category. This should result in six properties. Explain briefly for each category why the properties fall into the respective category \(S, K,\) or \(R\).

**Exercise 2:** Finite State Processes (2 Marks)

Give the textual Finite State Process (FSP) definition for the following Labeled Transition Systems. You can check your own results using the LTSA tool. (This is also what we will do. ☺)

Download and installation instructions for the LTSA tool: [http://www.doc.ic.ac.uk/ltsa/](http://www.doc.ic.ac.uk/ltsa/).

a)

![Diagram](image1)

b)

![Diagram](image2)

**Exercise 3:** Modeling with Labeled Transition Systems (6 Marks)

Consider that for some system, we model a process for the software controller of the system and another process for a non-software part of the system, or its context. In the lecture, we learned that some actions are controllable and some are uncontrollable (by the software). However, using LTS, we can easily model a process for the software that blocks uncontrollable actions or enforces a particular order on the uncontrollable actions. Vice versa, we can easily model a process for the environment that blocks actions that are controllable by the software or enforces a particular order on them. This does not seem adequate. Can you think of rules for modeling with LTS where this is avoided?

a) Give an example two processes, one for the software controller of a system and another process for a context or non-software part of a system, where the software
process can block an uncontrollable event or enforce an order on uncontrollable events. (You may refer to the crossing example above.)

b) Describe modeling rules by which you think the problem described in the exercise text can be avoided. Describe them as precisely as possible.

c) Give an example where these rules/guidelines are applied.

**Exercise 4:** Checking Safety Properties with LTSA (4 Marks)
The LTSA tool provides a simple mechanism to specify and automatically check safety properties. Consider the following process.

\[
\text{NO\_SUBSEQ\_A} = (a \rightarrow \text{AOCC}), \\
\text{AOCC} = (b \rightarrow \text{NO\_SUBSEQ\_A} | a \rightarrow \text{ERROR}).
\]

This process describes, intuitively, that there must not be two subsequent occurrences of \(a\) without an intermediate occurrence of \(b\). In the LTSA tool, \(\text{ERROR}\) represents a special kind of “bad” deadlock state, which should be avoided. Now, consider for example that we compose \(\text{NO\_SUBSEQ\_A}\) with the process \(\text{P1}\):

\[
\text{P1} = (a \rightarrow b \rightarrow \text{P1}), \\
\big| \big| \text{P1\_NO\_SUBSEQ\_A} = (\text{P1} \mid | \text{NO\_SUBSEQ\_A}).
\]

In the composed process, the deadlock state is avoided. However, if we instead consider the composition with a process \(\text{P2}\) as follows

\[
\text{P2} = (a \rightarrow \text{Q}), \\
\text{Q} = (a \rightarrow \text{Q} | b \rightarrow \text{P2}), \\
\big| \big| \text{P2\_NO\_SUBSEQ\_A} = (\text{P2} \mid | \text{NO\_SUBSEQ\_A}).
\]

then the composed process \(\text{P2\_NO\_SUBSEQ\_A}\) has a deadlock state. By saying “Check-Safety” in the LTSA tool, the LTSA tool will report a shortest trace that leads to the deadlock state, which is in this case “\(a, a\)”. You may copy the above process definitions into the LTSA tool and try it out.

Now consider the following safety property for the production cell example introduced in the lecture: “*If the controller orders ArmA to move to the table (aMoveToTable), then ArmA will not arrive at the press (aArriveAtPress) unless the controller orders ArmA to do so (aMoveToPress)*.”

a) Model the safety property as an LTS/FSP in the LTSA tool and compose it with the process \(\text{MOVEMENT\_ARMA}\) shown in Exercise 2.a). Does \(\text{MOVEMENT\_ARMA}\) satisfy the property? Model the safety property in such a way that, if \(\text{MOVEMENT\_ARMA}\) satisfies the property, the behavior of \(\text{MOVEMENT\_ARMA}\) will not be changed. Submit your FSP definition of the process that models the safety property and briefly argue why it will not change the correct behavior of \(\text{MOVEMENT\_ARMA}\).