CMU 17-651 Models of Software Systems

Project 1: State Machine Modeling

Garlan **Due: October 5, 2016**

The purpose of this first project is to give you experience in modeling a realistic system as a state machine. The example that we will use is the Infusion Pump. A general description of an Infusion Pump can be found in the General Project Documents folder on Blackboard.

You should carry out this project in your assigned team. Make sure that everyone in the group contributes to the overall effort. Each team should submit a single write-up of the project, due at the beginning of class on the project due date. We have posted a template for a group project write-up under the Course Resources $>$ LaTeX section on Blackboard.

Task 1 (20 points):

A sample description of a *simplified version* of an infusion pump, written in FSP, is provided with this project document. This model describes a pump with only one infusion channel, and leaves out many of the features that a real infusion pump would have, as outlined in the general infusion pump description.

Your first task is to understand this specification. Read through the specification to make sure you understand what it is specifying. Then read it into LTSA and check its behavior using the LTSA simulation capabilities. Once you are familiar with it, answer the following questions in your project write-up:

1. What is the alphabet of the state machine?
   $\{change\_settings, clear\_rate, confirm\_settings, connect\_set, dispense\_main\_med\_flow, enter\_value, erase\_and\_unlock\_line, flow\_blocked, flow\_unblocked, lock\_line, lock\_unit, plug\_in, press\_cancel, press\_set, purge\_air, set\_rate, silence\_alarm, sound\_alarm, turn\_off, turn\_on, unlock\_unit, unplug\}$

2. List two traces of the pump, each at least 4 actions in length.
   $\langle plug\_in, turn\_on, set\_rate, enter\_value\rangle$
   $\langle plug\_in, turn\_on, turn\_off, un\_plug\rangle$

3. In contrast to the specification of an infusion pump in homework 6, how does this specification model the fact that the pump might run out of liquid?

   This specification models the fact that the pump might run out of liquid the same way it models the blocked line. In either case the system will raise the alarm.

4. Is it possible to ever dispense medication without setting the rate? Why or why not? If your answer is yes, provide a trace that justifies your answer.
   No, it is not possible to dispense medicine without setting the rate. This is because once the user turns on the pump using the action $turn\_on$, they only have the options to either unplug, $turn\_off$ or set_rate. So in order to move ahead and start the infusion, they must set the rate using the action set_rate.
   A trace depicting this:
   $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line, confirm\_settings, lock\_unit, dispense\_main\_med\_flow, dispense\_main\_med\_flow, ...\rangle$

5. Is it ever possible for the flow to become blocked and have the alarm not sound at all? Why or why not? If your answer is yes, provide a trace that justifies your answer.
   No, it is not possible for the flow to become blocked and have the alarm not sound at all. This is because once the flow is blocked using the action $flow\_blocked$, there is only one possible action which is sound alarm.
   A trace depicting this:
   $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line, confirm\_settings,$
   $lock\_unit, dispense\_main\_med\_flow, flow\_blocked, sound\_alarm\rangle$

6. If the pump is locked and dispensing, without unlocking or becoming blocked, will the pump ever stop dispensing? If your answer is yes, provide a trace that justifies your answer.
   Yes, the infusion could stop dispensing once it is locked and dispensing if it is turned off or unplugged. However, it is not turned off or unplugged, it would could dispensing.
   The following is a trace depicting it being turned off:

   $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line, confirm\_settings,$
   $lock\_unit, turn\_off\rangle$

7. If the pump is locked and dispensing, is it possible for the *patient* to alter the medicine he is receiving? If your answer is yes, provide a trace that justifies your answer.
   The answer is yes. The following is a trace depicting:
   $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line, confirm\_settings,$
   $lock\_unit, dispense\_main\_med\_flow, unlock\_unit, change\_settings, confirm\_settings, dispense\_main\_med\_flow\rangle$

8. Does this version have any behavior that you feel is inconsistent with the pump specification? Could it be fixed?
   One inconsistency could be the fact that the only way to stop the infusion pump is to turn off the pump, which is inconsistent with the intended design of the pump.
   If we were to add a Hold function that could pause the infusion, and then the pump could move to a Hold state.

Task 2 (75 Points):

For the second part of your project you should develop a more-complete FSP specification of the infusion pump (40 points). To do this you can use the sample of Task 1 as a starting point, or you can start with your own model. Here are some guidelines to keep in mind as you develop your FSP specification:

- You do not need to model the human user of the pump.
- Restrict your specification to a single-line infusion pump.
- You are free to pick the level of abstraction for this specification. Your specification should be detailed enough, however, to answer the questions posed below.
- Be sure to document your specification adequately, and choose meaningful action and process names for readability.
- Your specification should not use parallel composition.

The full specification should be attached to your project write-up. Answer the following questions (35 points); for each, briefly explain why you answered it in the way you did, based on *your* model. If your model does not address this issue, explain why. (Note: Reference certain parts of your specification that address features, ambiguities, or errors in each question.)

1. Which aspects of the pump did you choose to model, and which did you choose to leave out?
   We took the existing fsp model as a start point and added the following features:

   (a) Added hold button.
   (b) Added better error handling: in particular, distinguish between blocked line and out of liquid.
   (c) TODO: add more.

2. Were there ambiguities in the English description of the infusion pump that your specification resolves?

3. State four general properties that your pump guarantees (for example, the alarm will always sound if a line becomes clogged), and say briefly why it is guaranteed.

4. Does your model say what happens if the power goes out in the middle of operation?

5. Referring to the additional documentation about the infusion pump on the general project description section of the web site, consider the errors noted about realistic pumps. Which of these types of errors can be illustrated with your pump? Does your pump exclude some of them from happening?

   (a) If you have entered a rate which is outside the limits, or a VTBI which is greater than the maximum, when you press Run, you will see the message: "Preset Limits Exceeded" and an alarm sounds. The Rate or VTBI display also changes to the one used before you programmed the infusion.
   (b) If an occlusion is detected, the Run light remains lit, but the Pumping indicator stops.
   (c) If an occlusion alarm occurs, immediately clamp the line to the patient. Then inspect the fluid pathway to determine what has caused the obstruction.
   (d)

Task 3 (5 points):

How difficult would it be using the current subset of FSP to create a 4-line pump? What would have to change in your specification?