CMU 17-651 Models of Software Systems

Project 1: State Machine Modeling

Garlan **Due: November 11, 2018**

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?

   The current specification dispenses medicine but does not check if the medicine has run out. It only raises the alarm when the line is blocked.

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 based on the User Manual and Existing Recalls that were logged for the Infusion Pump:

   (a) Hold Button: Once a system is dispensing, it can only be turned off by pressing Hold. After pressing Hold, you can resume, turn off or unplug.

   (b) Power Backup: On power up, system checks to see if the machine needs to turn on the backup battery supply to support the case of a loss of power.

   (c) Self-Check: On power up, the system checks to make sure that it is in proper working condition, if it is not the pump will fail and shut itself off.

   (d) Dosage Control: After the Lines are Locked in Setup, a value between 0..X can be selected to limit the amount of fluids delivered.

   (e) Disable Power Off: If the pump is currently dispensing, the pump cannot be manually/accidently turned off. This feature was added based on recalls that were logged for the infusion pump.

   (f) Audio alarm for Blocked Line: If a line becomes blocked an audio alarm will go off until the line becomes unblocked.

   We did not include the following features in the fsp model:

   (a) Audio alarm: The model only accounted for one scenario to sound the alarm, we did not account for Limits Exceeded, Power Failure, System Check, Pump Complete, etc.

   (b) Quiet pump mode: The model does not include the setting for quiet pump mode when setting up the machine.

   (c) Low battery alarm: The model does not sound an alarm when the battery runs low, instead it turns on the backup battery and continues to pump until complete.

   (d) Press the Hold key to silence the alarm: This model not not include a way to silence the alarm other than unblocking the line that was blocked.

   (e) Secondary Infusion: The model only accounts for having the infusion pump dispense one liquid. The manual states that a secondary infusion can be setup in addition to the primary infusion.

   (f) Dosage Control Bag Fill Capacity - The model only accounts for a fluid amount to be entered and decremented as the fluid pumps, it does not account for the fill capacity of the bags that the pump pulls the fluids from.

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

   In general the manual is not very exhaustive regarding the functions that are available in each mode, the fsp diagram is much more concise about what is allowed in each state.

   One example of that is that power_off is a one step function (tapping the power_off button) unless we are in the INFUSION state, were you need to press hold followed by turn_off.

   The manual takes 120 pages to describe the operation of the device were the spec is less than 3 pages, seems like the long manual was confusing to people so the OEM had to produce a shorter version of 18 pages.

To the best of our knowledge the manual does not specify how to setup a second infusion operation after the first one ends. In our state machine is clear that after receiving the done action, the user can decide to SETUP another INFUSION operation.

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.

   The following properties are guaranteed by the specification of the PUMP:

   1. An alarm will sound if the line is blocked while the pump is performing an infusion.
   When the pump is performing an infusion, it remains in the INFUSION state. When the flow of medicine is blocked, the pump sounds an alarm and transitions to the INFUSION_BLOCKED state. The pump remains in the INFUSION_BLOCKED state until the flow is unblocked. When the flow is unblocked, the pump returns to the INFUSION state. Whenever the flow of medicine is blocked during infusion (as modeled with the flow_blocked action) the sound alarm action is triggered.

   2. Once the pump is performing an infusion, it can only be turned off from the HOLD state.
   For the INFUSION state, there is a new action defined press_hold. The press_hold action triggers the pump to go into a HOLD state from which you can unplug, turn_off or press_resume the infusion. The actions to turn off the pump were removed from INFUSION, to prevent users from directly turning off the pump without first placing it on hold.

   3. The pump will automatically fail over to battery power in the event of a power failure during infusion.
   For the INFUSION state, there is an action defined for power failure called autoswtich. A power failure triggers the pump to switch to a battery power. The system checks whether there is a sufficient battery charge at setup. If it is not sufficient, it switches to battery and continues. Then it checks again while dispensing whether there is sufficient charge. If there is not a sufficient charge, it switches to battery power.

   4. The pump will not dispense more medication than the set dosage amount (if limit is set).
   When the pump has a dosage limit established during setup, a state variable is set which tells the INFUSION state to monitor for the maximum dosage. When the maximum dosage is met, the pump stops dispensing medicine and goes to a special done state. The special done state cannot be exited unless the pump is turned off or the user confirms they wish to dispense more medication to ensure the pump is not accidentally restarted.

4. Does your model say what happens if the power goes out in the middle of operation?
   The pump does not specify what happens if power goes out in the middle of operation.The pump talks about turn_off action at various states, but it does not mention how to address the loss of power in between operations.

   We have enhanced the model to include a battery backup operation. Once pump is turned on we check if it has sufficient charge or not. If it has enough charge it continues with the flow. If not, it turns to battery mode, and then continues the flow. Similarly when a power failure occurs,, it turns on battery mode and resumes operation from the state where failure occured.

   Examples traces:

   $\langle plug\_in, turn\_on, charge\_sufficient, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line\rangle$

$\langle plug\_in, turn\_on, charge\_not\_sufficient, auto\_switch, set\_rate, enter\_value, press\_set,$
$connect\_set, purge\_air, lock\_line\rangle$

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?

   We found 3 clear recalls for the infusion pumps:

   (a) FDA Recall: The machines, made by Baxter International and known as Colleague volumetric infusion pumps, shut down for a variety of software, wiring and design reasons. One of the most basic problem is that the "On/Off" key is so close to the "Start" key that nurses may inadvertently turn the machines off when they actually intend to begin drug therapy, the agency said.

   (b) FDA Recall retrieved from https://www.tennesseelawblog.com/fda_recalls_medtronic_infusion a defective pump motor has been known to stall, which means the delivery of vital drugs will suddenly stop without notice.

   (c) FDA Recall retrieved from https://www.youhavealawyer.com/blog/2008/01/12/infusion-pump-recall/ The hospital infusion pump recall was issued because the Alaris pumps could contain misassembled occluder springs, which is used to control the flow of medication. The springs could be bent, broken, nested or missing, which may result in too much medication or fluid being delivered. The FDA has classified the action as a Class 1 recall, since use of the defective Alaris/Medley infusion pumps involve a reasonable probability of serious injury or death.

       Overinfusion is difficult for the hospital or medical provider to detect. The defective infusion pump springs could work intermittently, and there is no warning or alert that the device is delivering too much of the fluid.

We fixed number one by requiring the user to select hold before attempting to turn off the pump while in the INFUSION state, else, just tapping turn_off will shutdown the pump.

We could have fixed number 3 by adding a flow sensor to control the amount of medicament that is being dispensed and if it was more or less than what is was expected then have the system raise an alarm and stop the dispensing procedure.

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?

In this project, we have only one line in pump, but if we had 4-lines in pump, we could give a range to every lines in specification easily. When we give a range, LTSA automatically creates parallel lines which have same behaviour with one line in pump.There is one pump machine , so general actions for all lines like plug_in and unplug does not need a range, but for the actions which specific to the lines, like check_charge, auto_switch needs a range to be able to represents different lines.

```
const N = 4
range R = 0..N

PUMP = UNPLUGGED,
UNPLUGGED =(
plug_in -> POWER_OFF
),

POWER_OFF =
(
turn_on[i:R] -> PUMP_CHARGE
),

PUMP_CHARGE=
(
check_charge[i:R] -> SETUP[ParamsNotSet][LineUnlocked]
|
check_charge[i:R] -> BACKUP_SUPPLY
),

BACKUP_SUPPLY =
(
auto_switch[i:R] -> POWER_OFF),
SETUP[params:ParamsStateT][lineLock:LineLockStateT] =
(
unplug -> UNPLUGGED
|
).
```