CMU 17-651 Models of Software Systems

Project 1: State Machine Modeling

Team 8: Nikita Avrelin, Artur Badretdinov, Alexandr Kozhevnikov, Artem Ostapchuk **Due: October 9, 2015**

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:
Assumption:
Action $flow\_blocked$ could happen from to reasons: problems in the line or running out of medicine.

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

2. List two traces of the pump, each at least 4 actions in length.
   Answer:

   - $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line,$
     $confirm\_settings, dispense\_main\_med\_flow, flow\_blocked, sound\_alarm, turn\_off\rangle$
   - $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line,$
     $erase\_and\_unlock\_line, clear\_rate, turn\_off\rangle$

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?
   Answer:
   In HW6 original specification model we had discrete values of medicine(0..3), that are in the machine. Every action $pump\_fluid$ decreased the number of medicine by one. When pump became empty, action $fluid\_empty$ happened. After that $ring\_bell$ happened.
   In current specification when pump is run out of liquid, $flow\_blocked$ action happens.(this action can mean two things:problems in the line or running out of medicine) Then $sound\_alarm$ action happens.
   Therefore we have two main differences:

- In the first case there is a discrete process, in the second - continuous.
- In the first case action action shows us that we are run out of medicine, in the second that this action can mean two things:problems in the line or running out of medicine.

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.
   Answer:
   No, because all of the traces from initial state lead to the $dispence\_main\_med\_flow$ action go through the set_rate. Even if there will be actions, returning machine back to the previous states ($power\_off$, for example), executed, the trace will go through the set_rates again.

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.
   Answer:
   It's not possible, because in the state $flow\_blocked$ state machine can came only to $sound\_alarm$.

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.
   Assumption:
   ?We assume that a patient is able to press the "turn off" button.
   Answer:
   $\langle plug\_in, turn\_on, set\_rate, enter\_value, press\_set, connect\_set, purge\_air, lock\_line,$
   $confirm\_settings, lock\_unit, turn\_off\rangle$

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.
   Assumption:
   ?We assume that a patient is not able to press the $unlock\_unit$ button and he doesn't know configure the pump from the start.
   Answer:
   According to the assumption, because there are no other actions, which can lead to these changes.

8. Does this version have any behavior that you feel is inconsistent with the pump specification? Could it be fixed?

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?
   Answer:
   Features modeled:

   - **Battery.** Due to the specification
   - **Battery power state.** Due to the specification, system can recognize three different states of power. First state - AC power (also we assume that state is default on the start of the system), Second state - Battery Power. (Charge is more than 4 hours), Third state - Battery Power. (Charge is less then 4 hours). And according to the level of battery, system automatically will change own state. Also this information will be showed to the user of the system. Information about 4 hours was found in specification of the infusion pump.
   - **Self-Check.** Due to the specification, after the turn on, software and hardware are applied to POST Checks (Power On Self Tests (POST)), that contains next tests (executing as one action):

     1. CPU test
     2. ROM / RAM CRC test
     3. Battery test
     4. Stuck key test
     5. Watchdog test
     6. Real Time Clock test
     7. Tone test.

     This take a period of time, but refer to articles about deathly errors of infusion pumps, this is important thing for this system. If system check is correct, than system can start the pump, in other case there are two variants: first - if the user considers that system is correct and errors aren't critical (for example, in order of emergency situations), he can start the pump; second case - he turn off the pump. Also we added periodical self-check during dispensing medicine flow, that checks correctness of the lines (such as pinched or not lines and other). Due to specification, this checking contains next tests:

     1. A RAM test shall periodically check different sections of the RAM through low-level drivers.
     2. A ROM CRC test shall periodically check different sections of the ROM through low-level drivers.
     3. A CPU test shall be performed once every 60 minutes to check the processors code register.
     4. A Communications test shall be performed during all RF/wireless communication, checking the CRC for each packet received. A packet that is dropped shall be re-transmitted at least n times.
     5. A System failure alarm shall be issued if any of the system checks fail.
   - **Blocked flow refinement.** Due to the specification, type of the error can be easily recognized by different types of alarm notifications. According to the description of the task we found that it is important to emphasize these 4 different troubles of why flow is blocked betweeen, in order to improve speed of nurse work:

     1. Medicine is ended
     2. Line is pinched
     3. Line is ended
     4. End of treatment time

     Also we assume that different actions lead to different notifications to the system, that is how nurse will distinguish difference of troubles.

   Features leaved:

   - In case of battery: actually we can have higher price of devision in real life. In that case alarm sound should arise specified level of battery charge.

- In Infusion Pump Description document we can have from one to four lines which can be connected sequentially, due to the projects task 2 specification describes single-line infusion pump.
- Specification assume that such settings like start of treatment time, infusion rate, bolus amountor, amount of medicine to be infused can be assigned. But such opprortunities are not reflected due to the selected level of abstraction.
- Different types of alarm sound according to stop of treatment time and error occured are not specified.
- We removed enter the parameters, cancel of set entered data - thus, we leave detailed entering values. We decided it on our level of abstraction, because in our assumption all actions (for example, set of parameters) do this things.
- We decided to merge some actions, that go one after another (for example such as connect set, purge air and lock line into one). It helps us easily to describe all electrical failures during the pump work, pump could be switched to battery modes from all states. As a drawback our state machine would have unchangeble time between merged actions, but we assumed it would be constant.

2. Were there ambiguities in the English description of the infusion pump that your specification resolves?
   Answer:
   In infusion pump description we have ambiguity about possibility of patients modification of pump settings and in the same time is rare in a ICU to lock pump settings . For this, we make an assumption that patient cant use Unlock unit button (only caregiver or doctor can use it).
   In primary specification had ambiguity of the pump behaviour in case if electricity goes off, to solve this obscurity, in new specification in case of electricity loss, pump will automatically switch to the battery.
   To prevent uncertainties related to possible interruptions in the middle of the operations (such as switch from electricity to battery), we make an assumption that we have short-time actions (such as milliseconds).
   Silenced alarm is not zero volume but quite volume.
   According to requirements from "Generic Infusion Pump Hazard Analysis and Safety Requirements", we include assumption that pump will work on low battery level less than 4 hours.
   Refer to description, electrical failure may occur causing the pump to switch to battery operation. Thus, we make an assumption that AC power can be unavailable.
   To start the pump it should be plugged in to AC power and turned on, only after that we can switch to the battery power.
   In primary specification had ambiguity of the pump behaviour in case if electricity goes off, to solve this obscurity, in new specification in case of electricity loss, pump will automatically switch to the battery. In addition, there is no description about a "Unit lock" function, description doesn't name the way how it goes. Also there is no decription of the power-on-self-test, so we adde it into the specification.

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.
   Answer:

   - If AC power will fail, the Infusion Pump will switch to the battery and will continue to work. Pump checking power state all the time for level of the battery (in case of less than 4 hours, it will be notificate about it, according to description).
   - The alarm will sound if flow will be blocked and caregiver will know for reason of blocking (for example, pinch). Reasons of blocking defined by state.
   - Doctor can adjust pump settings like flow rate and concentration of the drugs.
   - Pump has ability to be checked before the using. Pump will inform the nurse about errors described above. She could thinks that the model has non-critical errors and continue work even the pump has errors.

4. Does your model say what happens if the power goes out in the middle of operation?
   Answer:
   According to specification we assume that fluids are served by impulse from power, in our specification there is

one impulse for one dispension. According to developed specification, line voltage is checked before upcoming dispension has begun.

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?
   Answer:
   One of most frequent problems is incorrect calibration or maintenance of the pump [JCAHO Environment of Care News Jan 2003, p.5-8]. In our model we try to exclude this by using self-checking. This process executes on the start of the pump and also repeats in some time interval and check hardware for potential malfunctions. Thus self-check can exclude problem that described at article about a Class 1 recall of infusion pumps with broken spring for pump measurements [http://www.youhavealawyer.com/blog/2008/01/12/infusion-pump-recall].

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?
Answer:
Difficulty will be to add extra states for each line at the setup and at the pump and set correct sequences of states. It will be changed at setup (because we should setup all 4 lines) and in the process of the pump (check for each of the lines). Thus, we should check all the lines during this processes and in case when one of flows is blocked, we should somehow show, in what line the problem occured. In other case, the pump will not fulfill the [Generic Infusion Pump Hazard Analysis and Safety Requirements] requirement to avoid any dangerous situation. In addition, we have newxt requirement: ”Once the patient is stabilized, the pump will be de-activated one channel at a time.”, says us the pump desription. Therefore, in addition to power off the pump we should provide ability to de-activate each line sequently.