CMU 17-651 Models of Software Systems Fall 2016

Homework #6: State Machines II and FSP

Garlan Due: 3 October 2016

1. Consider the answering machine described in HW 5. Write an FSP specification of **AnsMachine**. (For your answer, include the text of the specification and turn in a diagram drawn by LTSA as an attachment.)

2. Consider the FSP specification of a simplified version of the Infusion Pump attached at the end of this assignment. Answer the following questions:

   a. Give an example of an action-based trace that causes the infusion pump to terminate without an alarm being raised.

   b. Give an example of an action-based trace that causes the infusion pump to terminate with an alarm being raised.

   c. Is there a limit to how much medication can be administered to a patient? Explain why or why not.

   d. What happens if the nurse forgets to put any medicine in the bag (i.e., uses a refill amount of 0)?

   e. Is it possible for an alarm to sound even if the patient has received the correct and full amount of medicine?

3. Modify the FSP specification above to add two of the following capabilities, making sure to explain in your comments which capabilities you are adding.

   a. self-check at start-up

   b. confirmation of settings

   c. a start and end of treatment time

   d. other error condition detection

   e. ability to set the amount of medicine to be dispensed at each dispensing step

   f. power outage and automatic switch to backup power supply

   Submit an electronic copy of the modified FSP specification.

```
//---------------------------------------------------
//  Simple Infusion Pump
//---------------------------------------------------

//
// Set of actions that can be selected interactively to
// the animation of this model with the LTSA tool.
//
menu AnimationControlMenu = {
    plug_in, set_value[0..3], reset, fill_fluids
}

//---------------------------------------------------

//======================
// Constants and Ranges
//======================

const Max = 3 range Amt = 0 .. Max

const FillAmt = 2    // Amount in bag initially and after refilling

//=====================
// Process Definitions
//=====================

//
// Pump starts in power off state
//
PUMP = POWER_OFF,

//
// User must plug pump in before anything else can happen
//
POWER_OFF = (
    plug_in -> SETUP
),

//
// Before pump operation starts, user must enter amount of medicine to deliver
// to patient
//
SETUP = (
    set_value[deliver:Amt] -> PUMP[deliver][FillAmt]
),

//
// Main operation of pump:
//  User may reset pump at any time
//  When the pump has delivered the amount of medicine requested it goes
//      to the DONE state
//  When fluid runs out, the pump goes into an alarm state
//  Otherwise, the pump delivers one unit of medicine
//
PUMP[deliver:Amt][remaining:Amt] = (
    reset -> SETUP
    |
    when (deliver == 0)
        done -> DONE
    |
    when (remaining == 0)
        fluid_empty -> EMPTY_ALARM[deliver]
    |
    when (deliver > 0 && remaining > 0)
        pump_fluid -> PUMP[deliver-1][remaining-1]
),

//
// Error state associated with empty pump:
//  Repeatedly rings bell until user refills the pump
//
EMPTY_ALARM[deliver:Amt] = (
    ring_bell -> EMPTY_ALARM[deliver]
    |
    fill_fluids -> PUMP[deliver][FillAmt]
),

DONE = END.
```