CST315 Software Enterprise: UML Activity Diagrams (Fall 2015 Revision)

In this module we started UML, and the first diagram we discussed was the Activity diagram. Activity diagrams are UML's version of a *flowchart*, they are typically used early in problem understanding to help business analysts visually represent 1) process flows, and/or 2) the responsibilities of various actors in a system. In this lab you will work to understand and develop Activity diagrams.

Objectives:

1. Understand and apply flow modeling to translate interaction design and behaviors.
2. Practice UML syntax and semantic "tools".
3. Be able to create more refined models assigning system responsibilities to actors and system components via Activity diagrams and swimlanes.

Tools:

- *Visual Paradigm, Community Edition*
  - *URL: http://www.visual-paradigm.com/download/community.jsp.*
  - *VP is a commercial level tool for generating UML diagrams. We will be using VP in this class since the free community version includes all the features we need and it supports Windows, Linux, and Mac.*
  - *Please note that this tool requires registering on the VP website to receive an activation key.*

    -or-

- *Microsoft Visio*
  - *As a student at ASU, you receive access to many Microsoft products as part of the DreamSpark program. For those interested in using Microsoft Visio for UML, we have uploaded a document that covers downloading, extracting, and installing Visio - see the Software tab on BlackBoard.*

Task 1 (individual): Reading Activity Diagrams [30 total points]

See next page.

Task 2 (individual): Creating Activity Diagrams [50 total points]

Model a High-Blood-Pressure (HBP) monitor used in the home that takes systolic and diastolic blood pressure readings, plus heart rate (pulse) and submits them to a cloud service. Clinicians and end users can see reports on the uploaded data in real-time, but not insurance companies.

Create a **single** UML Activity Diagram for this system that models user workflows above. (The short description does not include all of the details required to construct your Activity Diagram - we are relying on you to fill in the amount of detail appropriate for the model. If want, you may google for a commercial product to use as an example.)

1. The diagram should attempt to model a HBP monitor system – many interpretations are fine. [10 points]
2. The diagram should be complex enough to be useful – at *least* four actions with appropriate control flow elements. [10 points]
3. UML notation rules must be obeyed! [30 points].

Task 3 (individual): Adding Swimlanes [20 total points]

In this task, you will introduce swimlanes into your diagram from Task 2:

1. Refactor your Task 2's layout into appropriate columns or rows. Do not add or remove elements! [10 points]
2. Add swimlane notation to the diagram to allocate steps to different actors or system components. [10 points]

Submission

- Task 1's answers, and the edited image, should be added into a Word document.
- Task 2 is done in an UML editor. Export to a JPG and add it into the document.
- Task 3 is done in an UML editor. Export to a JPG and add it into the document.
- Name your document as <asurite>_adlab.docx.
- As always, make your work presentable. Follow the instructions here carefully, ensure your images are readable, use proper grammar, spelling, and punctuation, and express your opinions in a clear, articulate style.

Task 1: Reading Activity Diagrams [30 points]

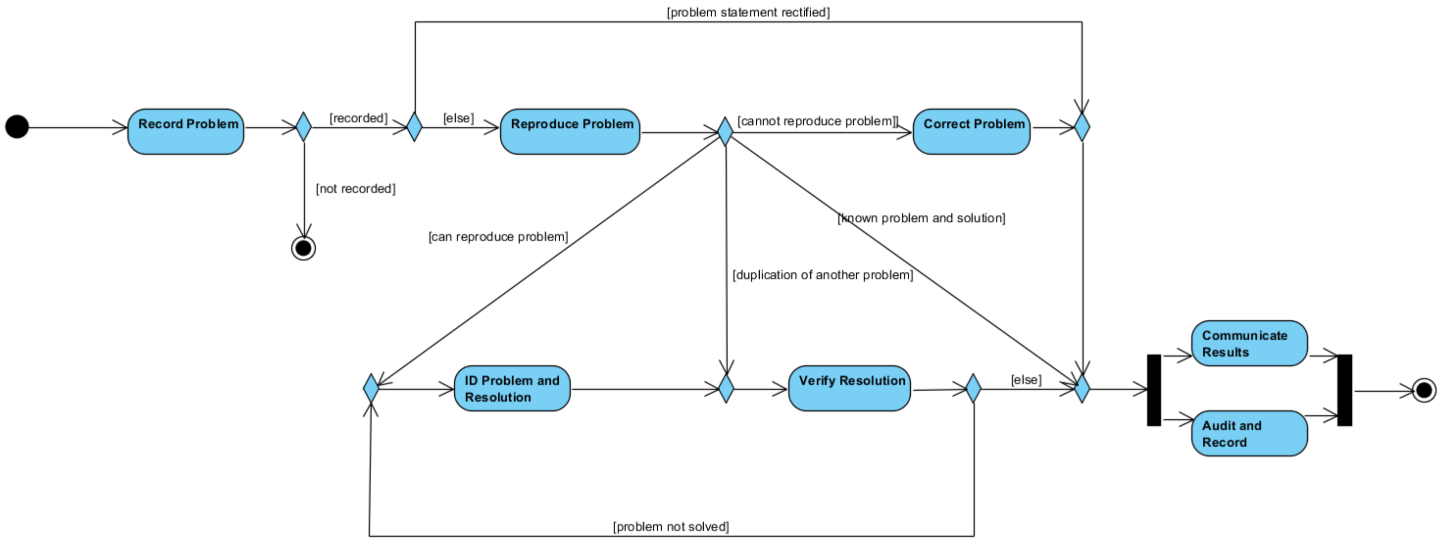

Figure 1 Trouble Ticket Activity Diagram

The above figure is an example of an UML activity diagram for the process of dealing with a trouble ticket in an organization. Using your knowledge of UML notation, answer the following questions:

1. What is the number of actions? [3 points]
2. What is the number of states? [3 points]
3. What is the number of guard conditions? [3 points]
4. What is the number of transitions? [4 points]

For the following questions, download a copy of the image from: cst315_umlbehavior_3_practice_v1_diagram.png. Edit this image in an image editor, and:

5. Draw a box around all *decision* nodes. [4 points]
6. Draw a circle around all *merge* nodes. [4 points]
7. Draw a large “X” through all final nodes. [3 points]
8. Draw a single vertical line (“|”) through all initial nodes. [3 points]
9. Draw an oval around all fork bars. [3 points] (Hint: careful with notation!)

Please draw with a color so we can see your work!