**SSW 567 - Assignment 4**

**Group 5**

Miguel Camacho

Ed Chang

Harmony Sullivan

**Assignment Description**

Part 1:

Create an executable model using the tool YAKINDU.

You can do this either for the garage door opener example in the prior lecture, or the microwave oven in the previous exercise. Alternatively, if you are really feeling frisky, you can come up with your own statechart example that use concurrency.

What did you think of this tool? Is it worthwhile?

Part 2:

Have an inspection of the triangle requirements document below.

1. Use the 2nd requirements checklist in the lecture. Assign each of your reviewers a reading role. Who has what role?
2. Document the results as described in the lecture/inspection documents, e.g. create an inspection report.
3. Create a traceability matrix from the requirements document to the test cases you’ve run (I’d suggest numbering/naming all of your test cases to make this easy)
4. Analyze/ graph the relationship between number of test cases and requirements. Are all covered? Do some have many?
5. If you have any requirements that haven’t been tested, create a test case, run it, and debug etc. as required
6. Reflect upon your results. Anything interesting?

Part 3:

Create a problem statement for your triangle “system.”Make sure you include all the factors in the ARB material.

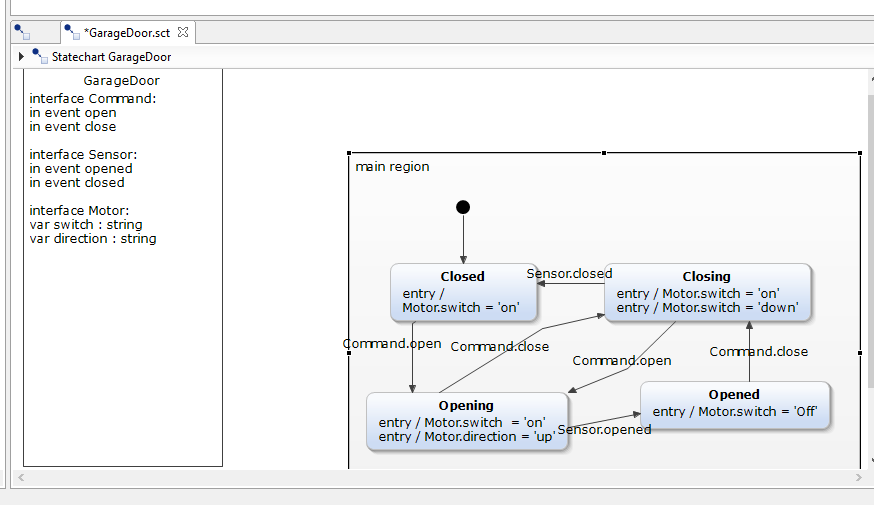
Requirements:

1. Request Input:
   1. The program will request 3 numbers, which we call A, B, and C, which are the sides of a possible triangle
2. Verify the Input:
   1. The 3 inputs must be numbers, greater than 0 and not too big.
   2. Otherwise, output error message and stop
3. Verify that the sides form a legal triangle
   1. If A + B >= C or B + C >= A or C + A >= B then a legal triangle
   2. Otherwise, output error message and stop
4. Determine if the triangle is a right triangle
   1. If A^2 + B^2 = C^2 or B^2 + C^2 = A^2 or C^2 + A^2 = B^2, then a right triangle.
   2. Due to the issues of numerical precision, = in this case means within 1%
5. Determine if the triangle is equilateral, isosceles, or scalene
   1. If the sides are all equal, then the triangle is “Equilateral”
   2. If two side are equal, but not three, then the triangle is “Isosceles”
6. Print out the type of triangle.
   1. If it is a right and Isosceles – Print “Right Isosceles Triangle”
   2. If it is a right and Scalene – Print “Right Scalene Triangle”
   3. If is a right and Equilateral – Print “Equilateral Triangle”
   4. If it is not a right and Isosceles – Print “Isosceles Triangle”
   5. If it is not a right and Scalene – Print “Scalene Triangle”
7. Stop

**Results**

**Part 1**

Yakindu model



**Part 2**

1. Roles
   1. Who has what role?
      1. Miguel = Test Engineer
      2. Harmony = Developer
      3. Ed = Moderator
2. Inspection Report
   1. Preparation Time of Reviewers:
      1. Reviewer 1- 2.5 hrs
      2. Reviewer 2 - 1 hrs
      3. Reviewer 3 - 0.5 hrs

**ALL DEFECTS FOUND**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Defect # | Reviewer | Severity (H, M, L, Q) | Location | Description |
| 1 | Reviewer 1 | M | Throughout | Tradeoffs between requirements are not clear; requirements are not prioritized. |
| 2 | Reviewer 1 | H | Throughout | Multiple dimensions have not been considered, such as cost, customer value, and development risk. |
| 3 | Reviewer 1 | H | Throughout | It is not explicitly stated if all product stakeholders have provided input to the prioritization process. |
| 4 | Reviewer 1 | M | Throughout | The requirements are not realistically distributed among the priority levels. There are no priority levels. Requirements are listed in order of occurrence. |
| 5 | Reviewer 1 | H | Requirement 2 | Requirement 2 states ‘not too big’, this is vague. The variable type or bounds should be included. |
| 6 | Reviewer 1 | M | Requirement 5 | The term scalene is not defined. |
| 7 | Reviewer 1 | L | Throughout | Diagrams, algorithms, use cases, tables, or other devices are not used to reduce ambiguity where appropriate. |
| 8 | Reviewer 1 | H | Requirement 5 | Requirement 5 states to determine if the triangle is isosceles, equilateral or scalene but does not have a sub-bullet for determining if it is scalene and therefore missing the definition of scalene. |
| 9 | Reviewer 1 | H | Requirement 6 | Requirement 6 does not include a print statement for a triangle that is not a right but equilateral |
| 10 | Reviewer 1 | Q | Requirement 4 | How to test Requirement 4. ‘within 1%’? This implies that the numbers are not integers, which has not been established? Must make an assumption on how to test this, it is vague. |
| 11 | Reviewer 1 | M | Throughout | Non-functional requirements (performance, reliability, etc.) are not quantified using an appropriate scale of measure. |
| 12 | Reviewer 1 | H | Throughout | Each requirement is not externally consistent with requirements at other levels (product, business, market, etc.). This information is not provided. |
| 13 | Reviewer 1 | H | Requirement 6c | It is impossible to have a right equilateral triangle. This requirement should be removed. |
| 14 | Reviewer 2 | L | General | Tradeoffs are not defined |
| 15 | Reviewer 2 | L | General | No way to determine cost, customer or development risk |
| 16 | Reviewer 2 | L | General | Not sure if all product stakeholders are involved or have input to the prioritization process |
| 17 | Reviewer 2 | L | General | Can not determine if requirements are realistically distributed among the priority levels |
| 18 | Reviewer 2 | H | Requirement 5 | Scalene isn't defined as part of Requirement 5 section |
| 19 | Reviewer 2 | M | Requirement 2.1 | not too big' isn't clearly define as to what it should be |
| 20 | Reviewer 2 | H | General | None of the requirements use 'shall' |
| 21 | Reviewer 2 | H | Requirement 4.3 | Problem with the requirement - can't be both right triangle and equilateral |
| 22 | Reviewer 2 | L | General | No non-functional requirement can be obtained |
| 23 | Reviewer 2 | M | Requirement 2.1 | Requirement has 'and' thus making it compound - should break down to multiple requirements instead |

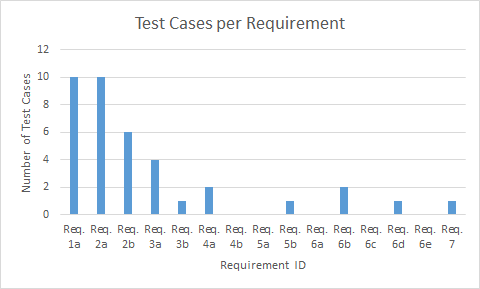
**DEFECTS WITH DUPLICATES REMOVED**

|  |  |  |  |
| --- | --- | --- | --- |
| Defect # | Severity (H, M, L, Q) | Location | Description |
| 1 | M | Throughout | Tradeoffs between requirements are not clear; requirements are not prioritized. |
| 2 | H | Throughout | Multiple dimensions have not been considered, such as cost, customer value, and development risk. |
| 3 | H | Throughout | It is not explicitly stated if all product stakeholders have provided input to the prioritization process. |
| 4 | M | Throughout | The requirements are not realistically distributed among the priority levels. There are no priority levels. Requirements are listed in order of occurrence. |
| 5 | H | Requirement 2 | Requirement 2 states ‘not too big’, this is vague. The variable type or bounds should be included. |
| 6 | M | Requirement 5 | The term scalene is not defined. |
| 7 | L | Throughout | Diagrams, algorithms, use cases, tables, or other devices are not used to reduce ambiguity where appropriate. |
| 8 | H | Requirement 5 | Requirement 5 states to determine if the triangle is isosceles, equilateral or scalene but does not have a sub-bullet for determining if it is scalene and therefore missing the definition of scalene. |
| 9 | H | Requirement 6 | Requirement 6 does not include a print statement for a triangle that is not a right but equilateral |
| 10 | Q | Requirement 4 | How to test Requirement 4. ‘within 1%’? This implies that the numbers are not integers, which has not been established? Must make an assumption on how to test this, it is vague. |
| 11 | M | Throughout | Non-functional requirements (performance, reliability, etc.) are not quantified using an appropriate scale of measure. |
| 12 | H | Throughout | Each requirement is not externally consistent with requirements at other levels (product, business, market, etc.). This information is not provided. |
| 13 | H | Requirement 6c | It is impossible to have a right equilateral triangle. This requirement should be removed. |
| 14 | L | General | Can not determine if requirements are realistically distributed among the priority levels |
| 15 | H | General | None of the requirements use 'shall' |
| 16 | L | General | No non-functional requirement can be obtained |
| 17 | M | Requirement 2.1 | Requirement has 'and' thus making it compound - should break down to multiple requirements instead |

C. Traceability Matrix

Please see separate worksheet “Traceability Matrix”.

D. The graph of test cases per requirements is shown below. As you can see, not all requirements were covered with our 10 test cases from week 2.



\*Note that Req. 6c is invalid

E.

New test cases below, to cover requirements that were not tested by existing test cases.

**Test Run 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| Wk4-1 | Determine if a triangle is a right triangle, within 1%. Testing requirement 4b. | 3.01, 4.01, 5.01 | [a^2 + b^2 must be greater than or equal to 99% of c^2 or less than or equal to 101% of c^2. So if a^2 + b^2 is greater than 0.99\*c^2 and less than 1.01\*c^2.]  **This is a right triangle, scalene.** | Scalene, not a right-angle triangle | Fail |
| Wk4-2 | Determine if a triangle with all sides equal is equilateral. Testing requirement 5a. | 8,8,8 | It is not a right and an equilateral (missing print statement in requirements) | This triangle is equilateral. This is not a right-angle triangle. | Pass |
| Wk4-3 | Print Right Isosceles Triangle. Testing requirement 6a. | 1,1,1.4142 | Print “Right Isosceles Triangle” | This is an isosceles triangle. This is not a right-angle triangle. | Fail |
| Wk4-5 | Print Scalene Triangle. Testing Requirement 6e. | 7,8,9 | Print “Scalene Triangle” | This triangle is scalene. | Pass |

**Test Run 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| Wk4-1 | Determine if a triangle is a right triangle, within 1%. Testing requirement 4b. | 3.01, 4.01, 5.01 | [a^2 + b^2 must be greater than or equal to 99% of c^2 or less than or equal to 101% of c^2. So if a^2 + b^2 is greater than 0.99\*c^2 and less than 1.01\*c^2.]  **This is a right triangle, scalene.** | This is a Scalene triangle and this is a right-angle triangle | Pass |
| Wk4-2 | Determine if a triangle with all sides equal is equilateral. Testing requirement 5a. | 8,8,8 | It is not a right and an equilateral (missing print statement in requirements) | This triangle is equilateral. This is not a right-angle triangle. | Pass |
| Wk4-3 | Print Right Isosceles Triangle. Testing requirement 6a. | 1,1,1.4142 | Print “Right Isosceles Triangle” | This is an isosceles triangle. This is a right-angle triangle. | Pass |
| Wk4-5 | Print Scalene Triangle. Testing Requirement 6e. | 7,8,9 | Print “Scalene Triangle” | This triangle is scalene. | Pass |

F. Reflect upon your results:

Each person found a different number of defects. Some were solely based on the checklist and sanity check of the requirements, but others also included an inspection of proper requirement writing.

Its up to reviewers discretion to categorize as H, M, L - for example customer feedback may not be a big deal for a program this size. Since the reviewers have different perspective, the same defect found by the different reviewers could be categorized quite differently.

The traceability matrix is a powerful tool for ensuring proper testing of requirements. The test cases that were written to cover requirements that were not tested by our original test cases caused us to open up the code and change it. In our opinion, the ‘requirements’ given in week 1 were the ‘baseline’ and these new requirements for week 4 are an update to that baseline. This has caused rework of the code and more specific behaviors expected of the triangle program.

**Part 3**

Our customer desires a simple way of classifying a triangle. This program makes it very simple for the user, he/she only needs to know the lengths of the 3 sides of the triangle and the calculations are done for him/her. It determines if the sides form a valid triangle, and then states whether it is a isosceles, equilateral, scalene and/or right angle. The functional requirements for the program have been summarized, describing the functions of the program in order of occurrence. The requirements are still being refined and have not yet been shared with the customer. This program will run on any Windows or Mac. At this time we have a constraint that it cannot be run on handheld devices but that is a feature that is planned in upcoming releases. The only external interface is the input from the user. This program is self-contained and does not need to connect to anything else externally.

**Lessons Learned**

The Yakindu is a useful tool as we develop our system. It allows us to visually see how the state transition is taking place and the simulation part offers us a chance to inspect the internal of the state machine throughout the process. This gives us a chance to establish the requirements necessary for our state machine up front instead wait until later stage of the software development process. In addition, the code generation part just ties the package together well. The only negative on Yakindu tool is that there is only a short tutorial available to ‘train’ the users and will require lots of trial and error before the state model can be created. In addition, we had experienced problem both installing/setup/configuring the Yakindu tool, and also experienced problem access the model after it was created.

Requirements may (and often do) change. This has a ripple effect as the requirements themselves must be re-validated, then the design of the program and test cases must be updated to account for the changes. Assumptions that were made initially are questioned and these must also be rewritten.

**Honor Pledge**

We pledge on our honor that we have not given or received any unauthorized assistance on this assignment/examination. We further pledge that we have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.