The University of Melbourne

SWEN90010: High Integrity Software Engineering

Assignment 3

Due Date: 11:59pm, Sunday 11 May, 2014

1 Introduction

This handout is the assignment 3 sheet. The assignment is worth 15% of your total mark and is done in pairs (the same pairs as assignment 2).

The aim of this assignment is to specify a formal model of the ICD system, and to produce a fault-tolerant design for this. The assignment evaluates your ability to apply specification and design techniques to engineer a safety-critical system.

2 Your tasks

The tasks for the assignment are listed below:

1. In the subject repository, a Sum specification of the manual mode for the ICD system is available (in the file `ICD.sum`). Extend this specification to model the requirements of the ICD system, including those new requirements introduced as part of your solution to assignment 2.

   This task requires you to introduce "modules" called `ClosedLoop` and `ICD`. The `ICD` module should automatically calculate the impulse from the measured heart rate. Your task is to use the supplied functions to specify the `ICD` behaviour, and then to bring the four modules together (Heart, HRM, ImpulseGenerator, and ICD) in the `ClosedLoop` module.

   A file (`axioms.sum`) of axiom definitions has been provided in the repository, which can be included into your specification and which you may find useful for specification of the `ICD` module. These include:

   (a) a function for calculating the average of a sequence of integers; and

   (b) a function for sorting a set of integers into a list.

   NOTE: Your Sum specification is NOT required to be 'animatable' with Possum. Not all first-order logic predicates are decidable, so by consequence, not all Sum specifications will be executable. It is quite likely that some of the operations in your Sum specification will not execute in a timely manner, even though they look correct. My advice is to back your own understanding and not get concerned with the animation. As part of marking, the specifications will not be animated through Possum, but you are quite welcome to use Possum to check your solution.

2. In assignment 2, you derived a hazard log. Applying fault-tolerant techniques from lectures, modify the ICD architecture from assignment 1 (outlined in Figure 2 in the assignment 1 handout) to mitigate those hazards that you believe can be mitigated using fault-tolerant techniques.

Show your design as a high-level architecture (at a similar level of detail to that in assignment 1) and describe the (new) algorithms used for achieving fault tolerance in the design.

3. Justify the decisions that you have made in your design. In particular, link each decision back to a hazard or set of hazards from your hazard log, and to the relevant theory in the notes.
4. Briefly list any new system requirements that must be added to handle the new fault tolerant design.

3 Criteria

Criterion	**Description**	**Marks**
Sum specification [6 marks]		
Correctness	The specification correctly models the requirements.	3 marks
Completeness	All requirements have been specified.	1 mark
Abstraction and modelling	The specification is broken into smaller parts and composed in a correct and sensible manner.	2 marks
Fault-tolerant design [5 marks]		
Architecture	A correct architecture has been chosen and designed.	2 marks
Algorithms	The correct algorithms have been chosen for the design.	1 mark
Justification	The justification is sound and clear. All fault-tolerant aspects have been justified.	1 mark
Requirements	The fault-tolerant aspects of the design are sensibly addressed by the new requirements.	1 mark
Overall [4 marks]		
Clarity	The specification and design are clear and succinct.	2 marks
Formatting	The specification follows the rules in Appendix A.	2 marks
Total		15 marks

4 Submission

Submit the assignment using the submission link on the subject LMS. Go to the SWEN90010 LMS page, select *Assignments* from the subject menu, and then select *View/Complete* from the *Assignment 3 submission* item. Following the instructions, upload a PDF file containing:

1. Your Sum specification. Please ensure that this is printed in a fixed-width font, such as `Courier`.
2. Solution to questions 2, 3, and 4.

Only *one* student from the pair should submit the solution, and the submission should clearly identify both authors.

Late submissions Late submissions will attract a penalty of 1 mark for every day that they are late. If you have a reason that you require an extension, email Tim *well before the due date* to discuss this.

Please note that having assignments due around the same date for other subjects is not sufficient grounds to grant an extension. It is the responsibility of individual students to ensure that, if they have a cluster of assignments due at the same time, they start some of them early to avoid a bottleneck around the due date. The content required for this assignment was presented before the assignment was released, so an early start is possible (and encouraged).

5 Academic Misconduct

The University misconduct policy applies to this assignment. Students are encouraged to discuss the assignment topic, but all submitted work must represent the individual's understanding of the topic.

The subject staff take plagiarism very seriously. In the past, we have successfully prosecuted several students that have breached the university policy. Often this results in receiving 0 marks for the assessment, and in some cases, has resulted in failure of the subject.

Appendix

A Specification format rules

Like code, the layout of code has a strong influence on its readability. As such, you are expected for format your specification well using the following rules:

- Every axiom or schema must contain a comment at the beginning explaining its behaviour. In particular, any assumptions should be clearly stated.
- Constants and variables must be documented.
- Variable names must be meaningful.
- Significant parts of operation schemas or schema declarations must be commented.
- Consistent indenting must be used; e.g. variable declarations and predicates must be indented from the schema name declaration.
- Lines must be no longer than 80 characters. You can use the Unix command "`wc -L *.sum`" to check the maximum length line your Sum file.