

HOMEWORK 2

Informatics 2-Software Engineering 2014/15

Ajitha Rajan

2nd October 2014

1 Overview

In this deliverable you are required to produce a detailed object oriented (OO) design of the CCS. The design must be an OO design and refined down to detailed descriptions of each operation and attribute.

The design of systems is still a bit of an art. The goal in the design process is a detailed design document that is complete enough to be a reference from which any competent software engineer (or computer scientist) can produce code, generate executable test cases, and generate test plans. Details on what we expect for the design are included in the subsection below.

Change in CCS. The customer has changed his mind on how to the engine throttle position should be interpreted. In the earlier description (in HW1), the customer stated that,

*“When the current throttle position is more (or less) than the previous throttle position by the amount $diff$, the car will accelerate (or decelerate) smoothly at $(diff/5) * max_accel$ metres/sec². Assume the max_accel for your car is 4 metres/sec². If the current throttle position is same as the previous position, then the car maintains the same speed and there is no acceleration (or deceleration). ”*

The customer has changed his/her mind and wants the acceleration to be computed according to the following,

*“The throttle position corresponds directly to speed the car will achieve (instantaneously or in a few time steps depending on terrain, previous throttle position, ...). A throttle position p corresponds to a speed of $50 * p$ km/h. Thus, position 0 corresponds to a speed of 0 km/h and a throttle position of 1.5 corresponds to a speed of 75 km/h. Thus, if the car was travelling at 100 km/h and the driver presses the start accelerating button, then the CCS, subject to other overriding conditions being false, should set the throttle position in that pulse to a value that corresponds to a speed of $(100\text{km/h} + ((2 \text{ metres/sec}^2) * 1 \text{ sec}))$. ”*

1.1 The Documents

The expected content of the documents is described below. The design document shall contain the following (at a minimum):

1. A **glossary** defining all new terms.
2. A **design overview** containing a description of all interfaces with the environment, and a description of all constraints and assumptions used in the design.
3. A **static model (UML class diagrams and class descriptions)**. This section must contain a complete class model, a high-level English description of the model, and detailed description of each class (class description, attribute descriptions, and method descriptions). You may choose to have the detailed description of each class as a separate figure rather than including all the details of each class in the complete class model. Additionally, you should include a written justification for the design you chose, including specific rationale for the decisions made in the design (for example, why your design may be better than another or why you chose to implement a specific design pattern in one of your subsystems). Use the Papyrus tool in Eclipse 4.2 to draw the UML class diagram and individual classes.
4. **Dynamic models (in this case, UML sequence diagrams)**. This section should contain a few scenarios captured as sequence diagrams illustrating the major uses of the system. Specifically, you should construct sequence diagrams for the use cases involving **start cruise control, stop cruise control, resume speed, accelerate** that you did in HW1. You may include more diagrams if you choose. Use the papyrus tool in Eclipse 4.2 to draw the UML sequence diagrams.
5. Explain how you have **addressed the feedback** (for improvement if any) provided by markers regarding requirements and use cases on your homework 1 in your design. If you have changed your requirements and use cases to address the feedback, attach the updated requirements and use cases.
6. Include a **title page with names and UUNs of team members**.
7. You should also submit a **text file named team.txt** with only the UUNs of the team members (one UUN on each line) as shown,
s1234567
s7891234

How to Submit. On the School of Informatics DiCE computer system, if your project is in a folder called *Application* (with the .pdf/.doc document and team.txt files) then you should submit it for Inf2C-SE **homework2** using the command:

submit inf2c-se 2 Application

2 Due Date

Homework 2 is due

Thursday, October 16th at Noon.

This homework is worth 30% of the total coursework.