University of South-Eastern Norway

IIA1319: Software Engineering

Assignment 3: Design

Isak Skeie, 245362

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

Contents

Contents **2**

1 Introduction **3**

1.1 xxx ..... Error! Bookmark not defined.
1.2 xxx etc ..... Error! Bookmark not defined.

2 Chapter title ..... Error! Bookmark not defined.

2.1 Subchapter title ..... Error! Bookmark not defined.
2.1.1 Sub-subchapter title ..... *Error! Bookmark not defined.*

3 Chapter title etc. ..... Error! Bookmark not defined.

4 Conclusion **11**

References ..... Error! Bookmark not defined.

Appendices ..... Error! Bookmark not defined.

1 Introduction

1.1 Assignment Parameters

The parameters given for this assignment as seen in Figure 1, gives a good picture of the characteristics of the systems in question. When doing the design steps for the Level control system, and the Temperature Control system. One shouldn't include these parameters. This is because they don't govern how the applications are programmed. In addition to them being susceptible for change, which creates a risk of mismatch with the design steps.

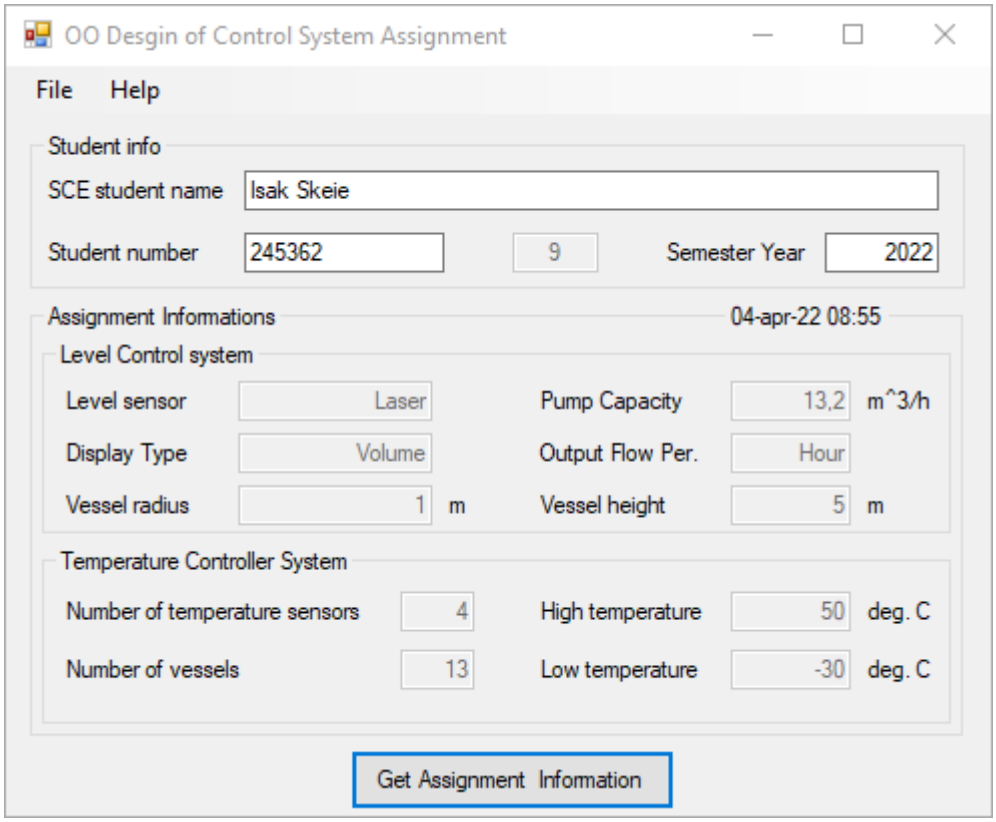

Figure 1: Parameters for the assignemnt

Confusion with temperature params

2 Level Control System

2.1 Analysis

- The most important use case from figure 1-1 in the Assignemnt paper, would be ”Control the Level”. The entire application is centered around this funcionality. The other cases would be obsolete without it. With this case being also being the reason the application is created.
- If the relation between the tank level and filled volume were to be non-linear. Additional requirements is created for the application. With a linear relation, the level could be treated as a signal passed to the next use case. With a non-linear relation additional calculations is needed before the signal is passed along.
- The use Case ”Control the Level” is defined as the most important use case. The SSD for this use case can be seen in Figure 2.

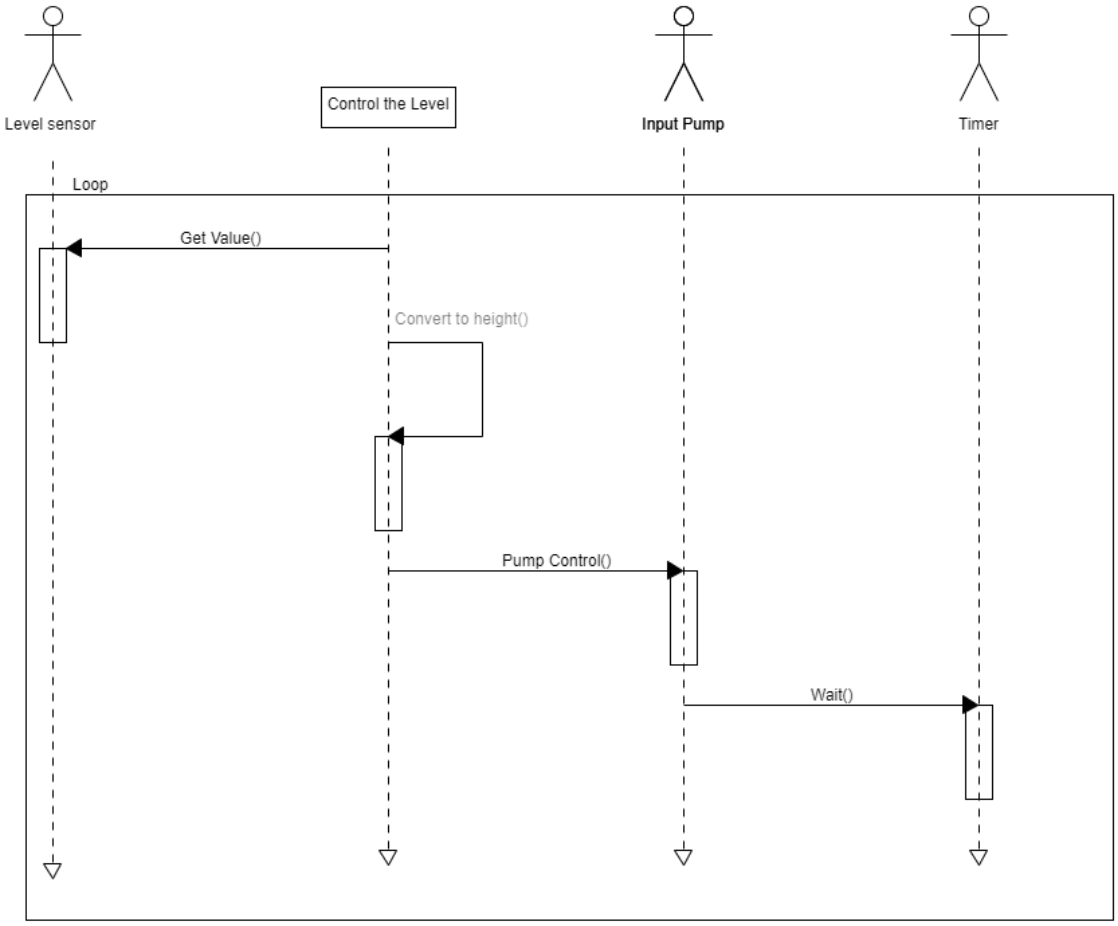

Figure 2: SSD for the use Case Control the temperature

2.2 Design

The first pattern being called is “Control the Level”. This use case doesn’t have any direct relation to the UI, that’s why this pattern cant be called a Controller pattern. It is however, a created pattern. It has assigned the responsibility of creating objects from the other classes in the diagram. The Information pattern is used by the Level Sensor class. Being the only class that gathers information the system bases its operations upon. High Cohesion is visible with in the interaction diagram. Each of the methods perform a single task, represented by the naming. Theres likely alternative sequences that could be more thoroughly elaborated, making the system somewhat less Cohesive than its potential. Based on the direction of the arrows, as well as the number of arrows, its clear that the system has low coupling. This could also be related to poorly elaborated alternative sequences. The Interaction diagram can be seen in Figure 3.

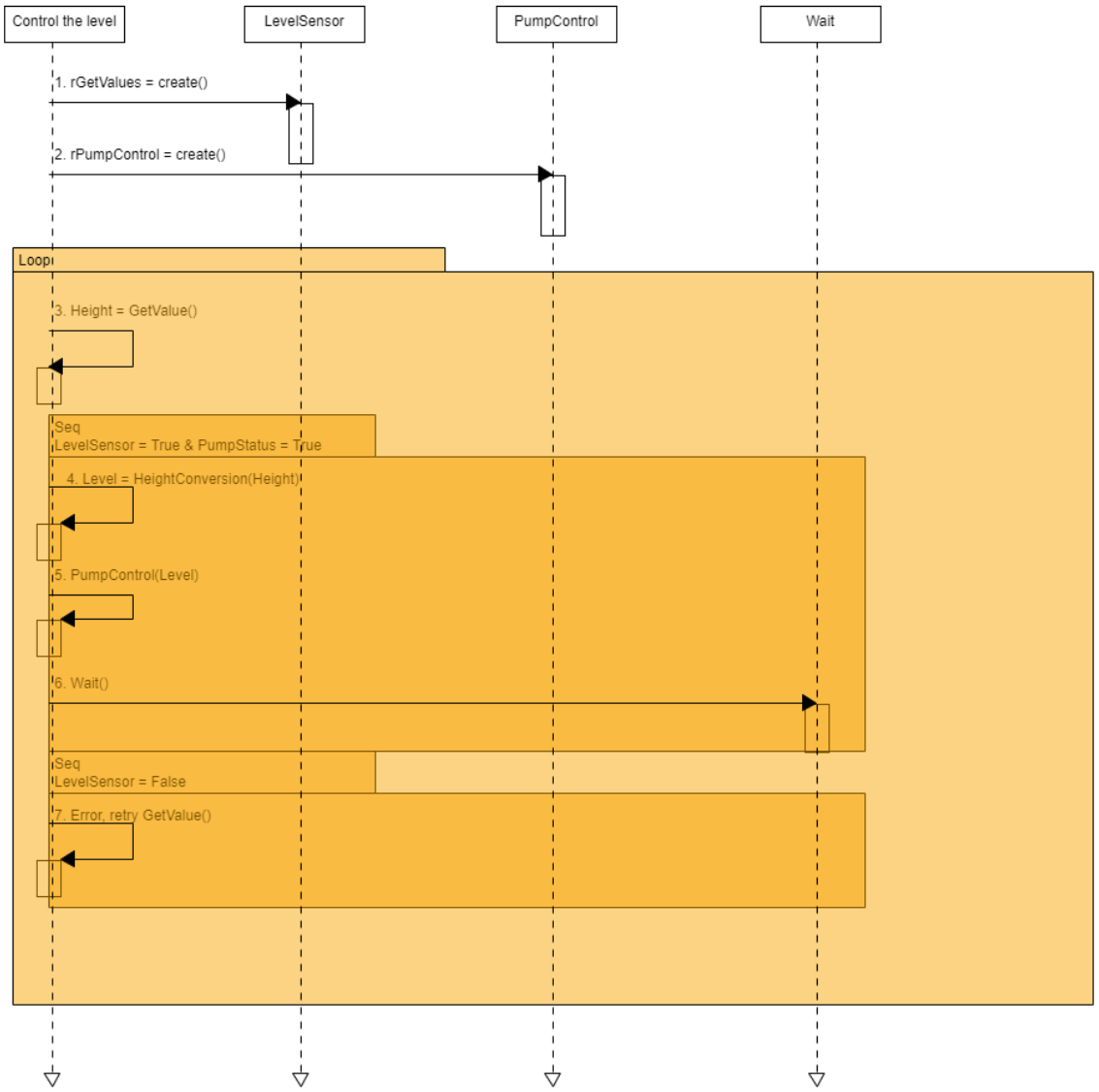

Figure 3: Interaction diagram for the use case

The SSD is more user oriented, focusing how the different use cases of an application interact with the different parties. This gives an overall structure of the data flow, where data is being created, altered and consumed. The Interaction diagram goes deeper into how each use case works. Which objects are being created to handle data, as well as how they are related to each other. The interaction diagram also handles how errors and exceptions are handled.

The Interaction diagram shows the dynamic behavior of the system, how data is handled and created to fulfill its goals. While the Class Diagram depicts more static representation of the system. How classes are related to each other, which ones are storing information, which is creating it?

The SSD gives an overview of how the application should be structured. When a structure is defined, the interaction diagrams should be created to get an overview of how the use cases

acts. The interaction diagram in combination with a class diagram should be enough to start coding the project. Starting with one use case at a time, testing and debugging it properly before moving on to the next use case.

The Class diagram is based on the Interaction diagram, and can be seen in Figure 4.

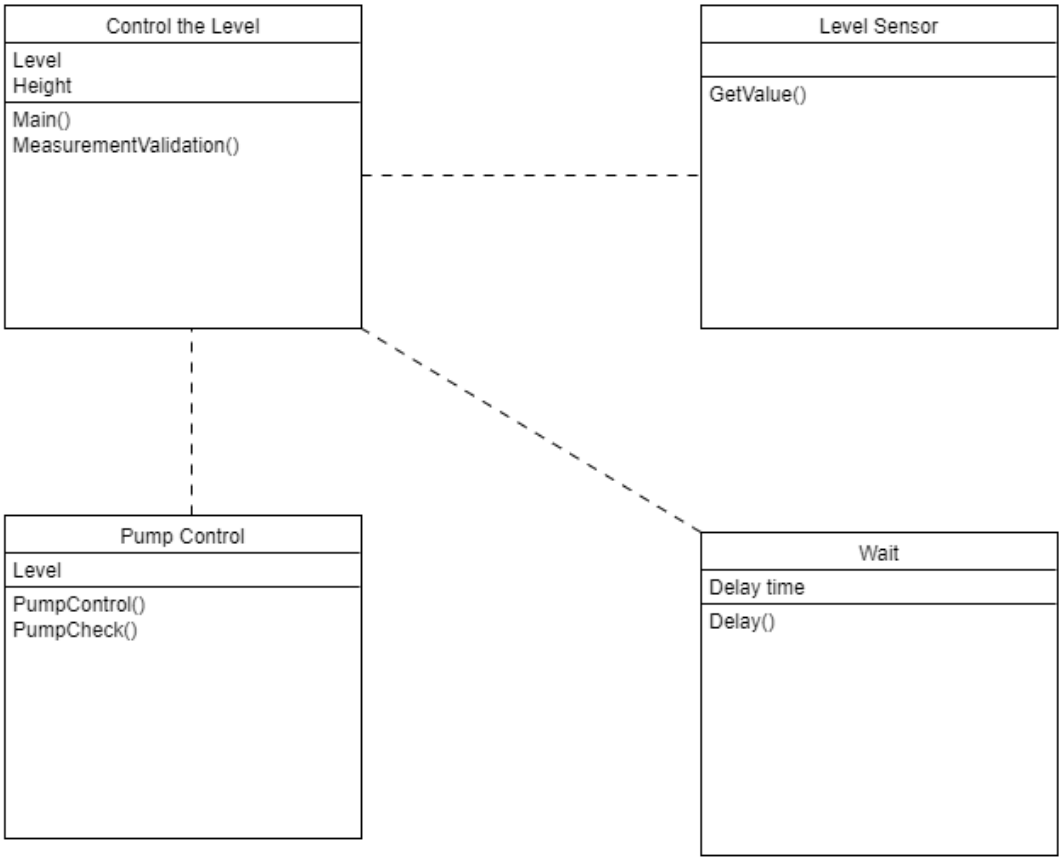

Figure 4: Class diagram for use case, Level Control

3 Temperature System

3.1 Design

With an interaction diagram visibly more complex than the one created for the level control. The design patterns are deployed to a larger degree. “Control the temperature” is the Controller pattern. Being the class to run the system when its requested from UI. For this interaction diagram theres two classes responsible for gathering data which makes them Information Experts. “Config DB” handles the configuration of the system, and “Temp Sensor” handles the temperature readings, used by the rest of the Application. With the number of Classes and objects being used, as well as the naming convention for them, its clear that High cohesion is used. The temperature control system is slightly higher coupled than the Level control system. Classes has objects instantiated from more than one class, making them more interconnected. This is related to structure of the interaction diagram, being more thorough and robust. Both “Power Up” and “ControlTheTemperature” created objects, which makes use of the created pattern.

Figure 5 shows the class diagram for Temperature Control use case. Clearly showing an increase in Coupling from the previous class diagram. Figure 6 shows the Object Diagram, Theres 13 vessels for the buffer tank, and 4 temperature sensors in each of these vessels. By making the ControlTheTemperature class an array of 13, its indicated that it controls the temperature in 13 different vessels. With TempSensor class having an array of 4, showing that four objects are created, one for each Temperature sensor.

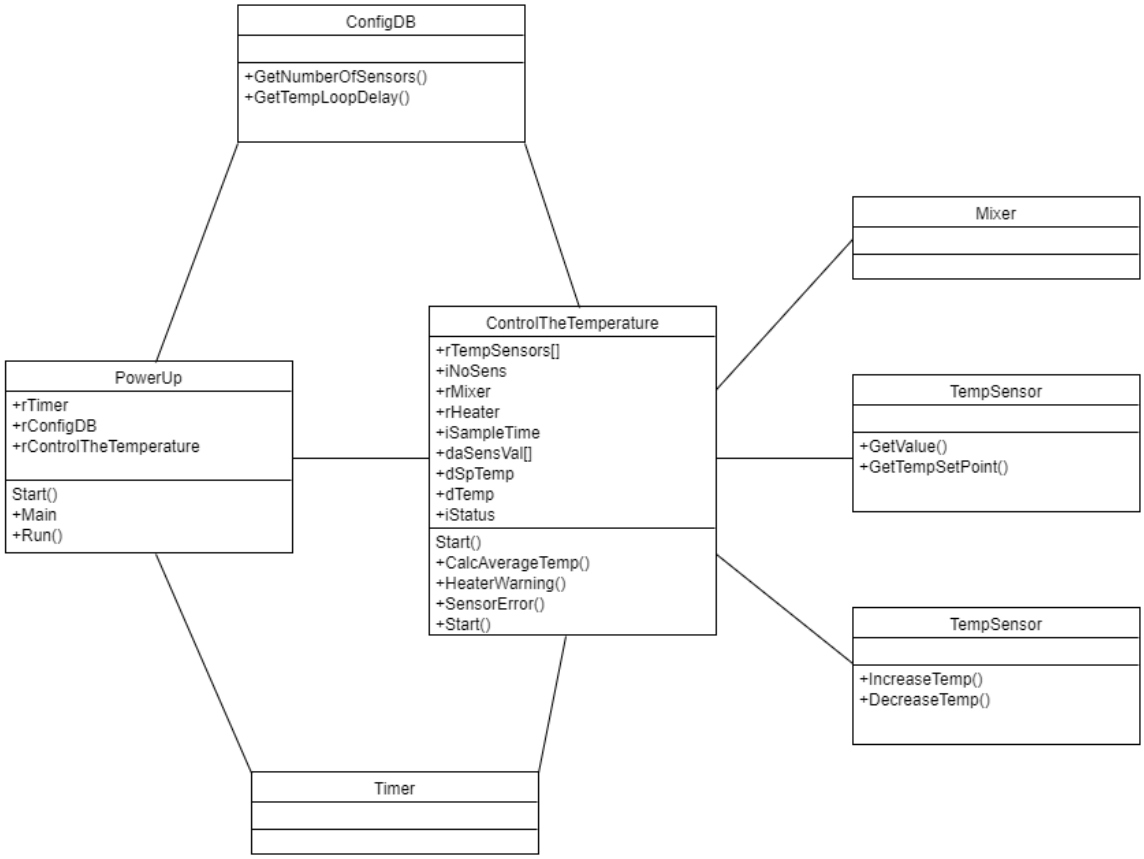

Figure 5: Class diagram for TemperatureControl

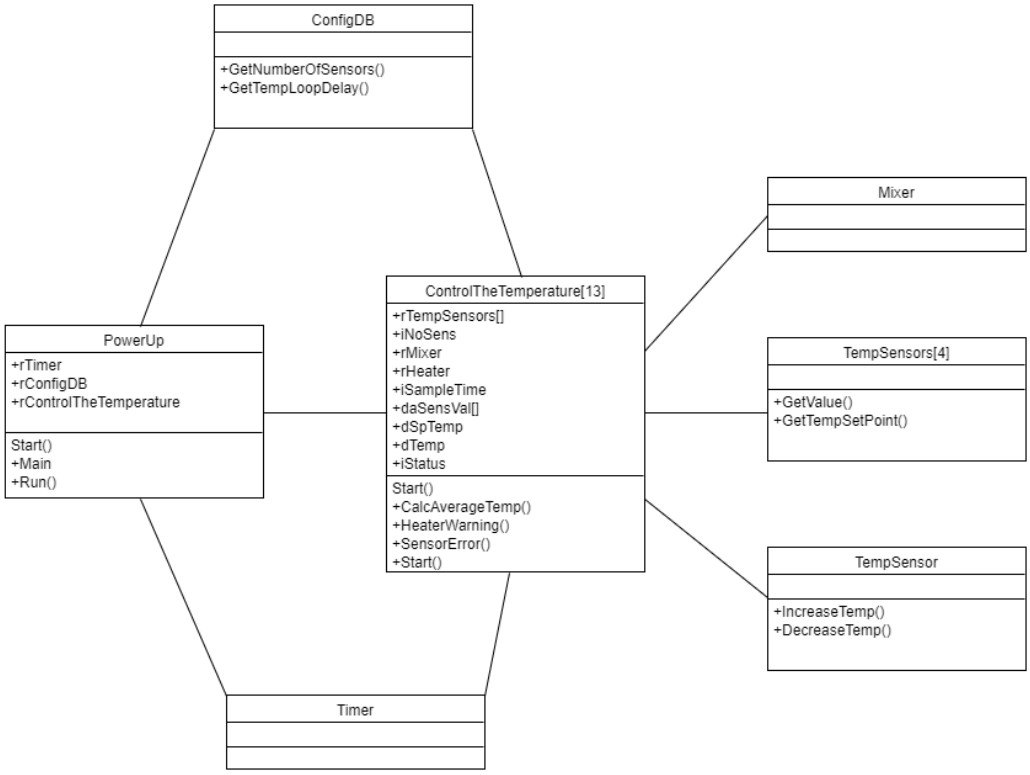

Figure 6: Object Diagram for Temperature control

4 Conclusion

The design and construction phase is highlighted as important methods to complete before coding begins. It gives the programmer an important overview of how the program should be created, forcing the programmer to take all possible events into account. It also produces good documentation, for other people to be able to develop the same application, as well for debugging purposes. The completed design steps, shows the importance of the previous steps in the Unified Process, creating the Fully dressed Use Case Document.

By creating a Interaction diagram with the Level controller, before reviewing the Interaction Diagram for the Temperature Control system. You learn how to create consistent Interactions diagrams, that meets certain criteria before moving on to the next phase of the development process.