

IIA1319: Software Engineering

Assignment 2: Analysis

Isak Skeie, 245362

Contents

Contents.....	2
1 Introduction	3
1.1 Task description	3
2 Application Analysis.....	4
2.1 Requirements	4
2.1.1 <i>Use Case Diagram</i>	4
2.1.2 <i>Domain Model</i>	5
2.2 Use Case Document	5
2.2.1 <i>System Sequence Diagram</i>	7
3 Development process.....	8
4 Summary.....	9
Appendices.....	10

1 Introduction

The purpose of this assignment is to get familiar with analyzing software applications. This is done by using parts of the Unified Process. A way to standardize the requirements of development process. For a more transparent workflow, both for colleagues and clients.

1.1 Task description

Two plants are going to be connected. This is done by having a buffer tank in between the two plants. A control system for managing the level in the tank is needed. Parameters for the buffer tank level control is provided by a custom application given by the teacher.

SCE 1306 OOA of Control System Assignment

File Help

Student info 27feb-22 17:00:19

SCE student name Isak Skeie

Student number 245362 12 Semester Year 2022

Input pumps	9	Capacity	30,8 m ³ /h
Output Pumps	3	Capacity	81,4 m ³ /h
Vessel Size	4182,3 m ³	Display	Level
PV Valve	In Use		
Vol. Set Point	50,2 %	Protocols	Both
Low Alarm	29,8 %	High Alarm	71,4 %

Get Assignment Information

Figure 1-1: Parameters for the application

2 Application Analysis

By mapping the requirements, and then elaborate the with drawings. Makes it clear how the different modules are interconnected and how develop the application.

2.1 Requirements

- It's important that the control system uses both Protocol 1 and Protocol 2. This is specified by the application provided with the assignment.
- Level control is essential for this control system to be functional. In detail, this means that the controller needs to be able to control the input pumps and monitor the output pumps. From this it needs to calculate the level with a certain accuracy.
- The control system needs an option to recalibrate the buffer tank level. Either with an algorithm or manually.
- Alarm Indication is a vital part of this application, if the controller is operational this needs to be functional. It's also necessary for the alarm to reach the right person when it occurs.
- The operator of this plant needs a GUI for the controller, if the controller is working, this needs to show the correct information generated by the controller. It must also be possible for an operator to override control of the pumps.
- The security valve should work if inlet pumps work.

2.1.1 Use Case Diagram

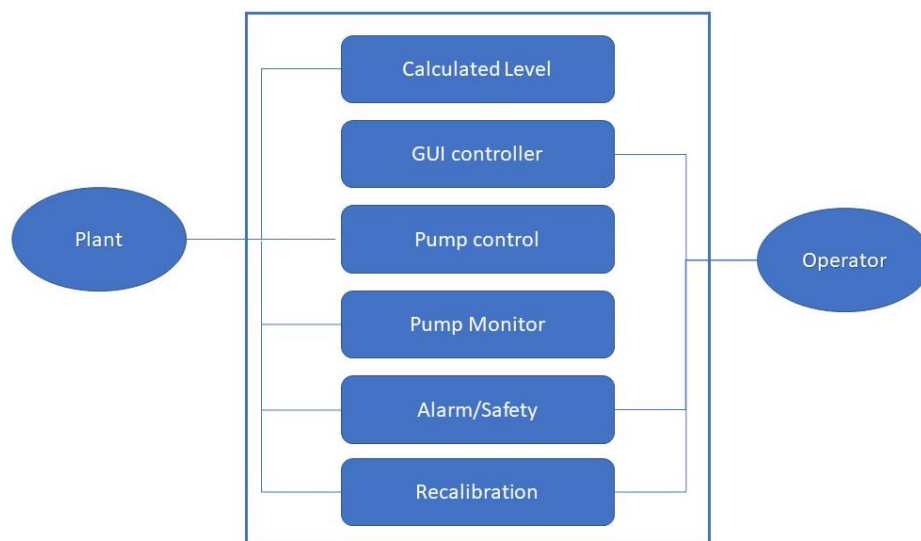


Figure 2-1: Use case diagram, depicting the requirements

2.1.2 Domain Model

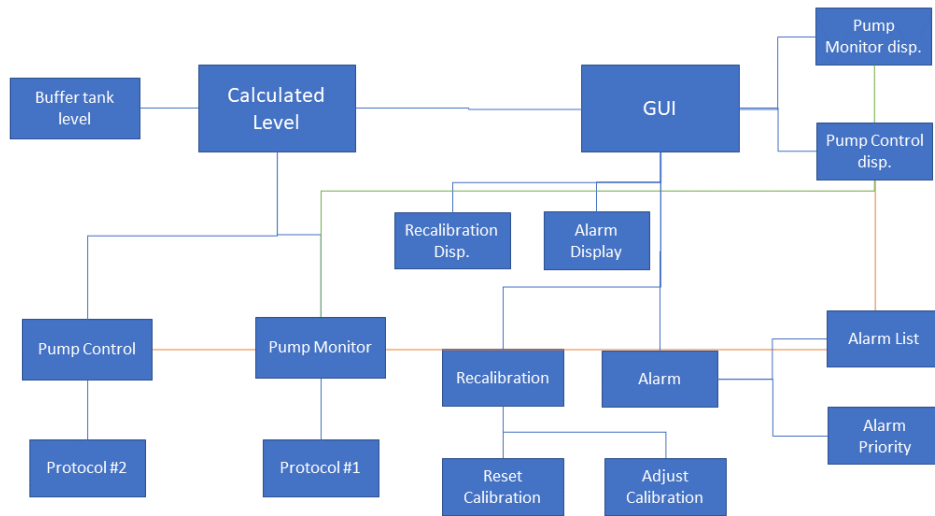


Figure 2-2: Domain model for the use case Level calculation

2.2 Use Case Document

Use Case name	Estimation of level in buffer tank
Scope	Product transfer from one plant to another
Level	Sub function
Primary actor	Pump control , GUI, Alarms
Stakeholder and interests	The people managing the factory cares for the level estimation in the buffer tank. The same does management that made the decision to invest in this upgrade
Preconditions	Communication between control system 1 and 2 must be working. Flow with respect to pumps must be precise. Alarm for high/low level must be working for safety reasons
Success Guarantee	Inlet pump must be controllable within a certain precision. The outlet pumps must be monitored with a certain precision as well.

Main Success scenario	Communication with Control system 1 over Protocol 1 must be in place, the same applies to Control system 2 over Protocol 2. A setpoint for the buffer tank level is given by the operator in the GUI. If the level in the buffer tank gets under the setpoint, the inlet pumps start pumping product. The flow with respect to the inlet pumps are accumulated over time. The outlet flow is monitored with respect to the outlet pumps. The outlet flow gets accumulated over time. Accumulated inlet flow gets subtracted by accumulated outlet flow, then level is calculated based on specifications on buffer tank and piping. The calculated level is shown in the GUI for the operator. Operator must be able to stop inlet flow when there's overflow.
Extensions	If there's issues with communication over the two protocols, inlet pump needs to stop, and operator notified in GUI. The inlet and outlet pumps could become inconsistent, depending on the product it pumps (viscosity, sediments etc.) and fatigue. Recalibration of pumps are needed. Communication with control systems could break, alarms notifying the operator is needed. Calculated level could deviate from real level over time. Recalibration, or emptying the tank to restart is necessary. GUI operations doesn't produce substantial errors for this use case.
Special Requirements	There shouldn't be a substantial latency in the calculations, communication between the protocols, nor the handling with the GUI.
Technology list	Protocol 1, Protocol 2, Internal protocol to GUI
Frequency occurrence	The frequency occurrence is based on the needed precision of the level calculation. The highest precision possible would be for the function to run on each cycle of the program.
Miscellaneous	Buffer tank desperately needs a level probe for High high level.

2.2.1 System Sequence Diagram

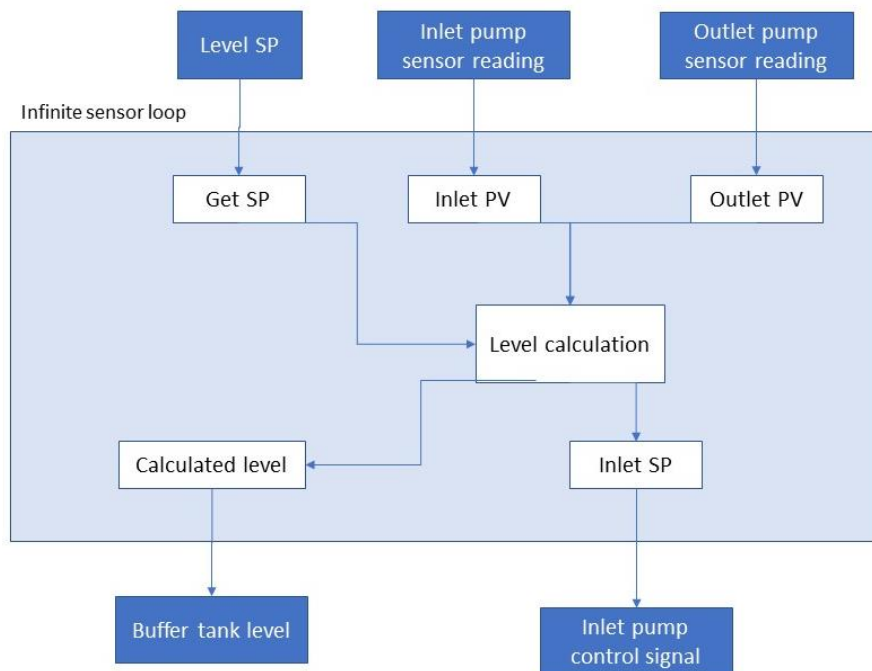


Figure 2-3: System sequence diagram for the Level calculation use case

3 Development process

With the Unified process you could start by gathering the use cases for the application, this includes level control of buffer tank, and different event handlings. Based on the use cases, it's possible to collect the requirements, as performed earlier in this report. When requirements for the controller is collected, it's possible to find out which modules to buy versus develop, and what kind of architecture to use. In this case, it could be wise to buy the modules that handles the communication with Controller 1 and 2. Otherwise, the rest should be developed, either in a Functional Block Diagram, or a C-based language.

With the use cases mapped, one could start prioritizing them and start development. In this case, you should start with the communication between the application and the controller, afterwards level estimation should be implemented. When this is done, a GUI should be made. As it confirms the workings of the system and needs to be used further on when developing the alarm and event handling functionality. Alongside the development process, it's important to have established procedures for testing the modules. This is to verify that each module work before continuing the next one.

When all the modules are done, its important test them together in a realistic environment before deploying it. With the application fully developed, documentation needs to be in order. This includes an elaborated Unified process document, as well as the version history of the different modules.

The programming for this project is simple, most of the time is going to be spent planning the project. As well as making sure the communication between different components is secure and reliable. An estimation of the time needed on this project would be around 3 weeks.

When it comes to testing the application, parts of the documentation produced from the Unified Process could be used. Specifically, the Fully Documented Use Case Document with the main success scenario, and Extensions.

4 Summary

By going through parts of the Unified Process. It's easy to see why it's needed on coding projects. It gives both the developer and the client, a clarifies picture of the development process. This is done by elaborating the end user's needs, and then categorize them, and handle them. This makes it easier to map what's needed to develop a project, as well as estimating the time that's needed.

When it comes to implementing this application. Not enough Information is provided. Specifications on the pumps are needed. As well as specifications on the pipes transporting the product. Additionally, it is not known how plant 1 supplies the buffer tank, if its continuously or in batches. This characteristics of the 1 process greatly influences the nature of the control system.

Appendices

Appendix A: “Lecture notes for object-oriented analysis, design, and programming using UML and C#”