

DYNAMISCHE CONFIGURATIE SOFTWARE

Ontwikkeling van een dynamische configuratietool voor de producten van Fluidwell



16-06-2015
N.D. van der Molen

DYNAMISCHE CONFIGURATIE SOFTWARE

16-06-2015

Auteur:	van der Molen, N.D.
Studentnummer:	2092521
Afstudeerperiode:	01-11-2014 – 16-06-2015 (75 werkdagen)
Opleiding:	ICT & Software Engineering (Deeltijd)
Onderwijsinstelling:	Fontys Hogeschool ICT Rachelsmolen 1 5900 MA Eindhoven
Afstudeerbegeleider:	Roelofsen, J.G.
Afstudeerplaats:	Fluidwell B.V. Voltaweg 23 5466 AZ Veghel
Afdeling:	Development
Bedrijfsbegeleider:	Meij, I.A., Product Manager / Technical Manager (CTO)

Getekend voor gezien door bedrijfsbegeleider:

Datum: 16-06-2015



De bedrijfsbegeleider,
Dhr. I.A. Meij

Voorwoord

Mijn naam is Nicky van der Molen. Ik ben studente aan Fontys Hogeschool Eindhoven, en volg daar de opleiding HBO-ICT & Software Engineering, in deeltijd.

Voor u ligt het afstudeerverslag “Dynamische configuratie software”, een beschrijving van mijn afstudeerperiode bij Fluidwell B.V. in Veghel. De werkzaamheden hebben plaatsgevonden in de periode van november 2014 tot medio juni 2015. Ik wil van de gelegenheid gebruik maken een aantal mensen te bedanken, zonder wie dit alles niet mogelijk was geweest.

Als eerste wil ik mijn bedrijfsbegeleider, Iljitsj Meij, bedanken, omdat hij in een voor hem ook zeer drukke periode, toch altijd de tijd genomen heeft als ik hem nodig had.

Ook wil ik mijn baas, Paul Stoop, bedanken. Hij heeft geaccepteerd dat ik een periode lang mijn normale werkzaamheden niet uit zou kunnen voeren, en mij verder waar mogelijk gesteund in het “vrij” houden van tijd voor mijn afstudeeropdracht.

Daarnaast wil ik graag mijn docentbegeleider, Jan Roelofsen, bedanken, voor zijn eindeloze geduld, en de altijd snelle reacties op vragen van mijn kant.

En ook wil ik mijn vriend, Rainer Schlönvoigt, bedanken. Hij heeft alle tirades en moodswings tijdens mijn afstudeerperiode dapper doorstaan, en me af en toe een zetje in de goede richting gegeven.

Veghel, juni 2015

Nicky van der Molen

Inhoud

Samenvatting.....	5
Summary	6
Verklarende woordenlijst.....	7
Lijst van afbeeldingen.....	8
Lijst van tabellen.....	8
Lijst van codefragmenten.....	8
1. Inleiding	9
2. Het bedrijf.....	10
2.1 Organisatie	10
2.2 Producten	10
2.3 Positie in de organisatie	11
3. De opdracht	12
3.1 Situatie.....	12
3.2 Doel	12
3.3 Opdrachtschrijving	12
4. Methoden & Tools.....	14
4.1 Methoden.....	14
4.2 Tools	15
5. Onderzoek	16
5.1 Applicatie opbouwen	16
5.2 Benodigde gegevens.....	17
5.3 Filestructuren/Database.....	17
5.4 Conclusies onderzoek.....	19
6. Ontwerp.....	20
6.1 Uitgangspunten ontwerp	20
6.2 Externe datastructuur	21
6.3 Interne datastructuur	24
6.4 GUI ontwerpen	25
6.5 Genereren GUI.....	25
6.6 Modbus driver	26
6.7 Overzichtsdiagram klassen	27

7. Realisatie	28
7.1 Implementatie ontwerp	28
7.2 Status	31
8. Conclusie en aanbevelingen	32
Evaluatie	33
Literatuur	34
Bijlage I – Project Initiatie Document	
Bijlage II – Ontwerpdocument	
Bijlage III – Modbus	

Samenvatting

Fluidwell B.V. is een Nederlands bedrijf, met het hoofdkantoor in Veghel. Ze zijn gespecialiseerd in de ontwikkeling en productie van elektronische producten en diensten voor de procesindustrie. Sinds de oprichting in 1993, heeft Fluidwell Displays (de eerste business unit) een grote verscheidenheid aan producten ontwikkeld. De eerste productlijn bestond voornamelijk uit batch controllers.

In de afgelopen 20 jaar zijn door Fluidwell veel verschillende typen displays ontwikkeld voor de weergave van data verkregen door bijvoorbeeld flow sensoren. De meeste van deze apparaten kunnen slechts met drie hardware-knoppen bediend worden. Hiermee moet ook de initiële configuratie door de klant worden uitgevoerd. Dit is een vervelend werkje. Bij de nieuwe productserie (de E-serie), wil Fluidwell daarom een configuratietool leveren. Deze tool zou ook geschikt moeten zijn voor de andere productseries.

In overleg met Fluidwell is bepaald wat er moet gebeuren: er moet een configuratietool komen, waarmee alle producten die via het Modbus-protocol kunnen communiceren, geconfigureerd kunnen worden.

De opdracht is met behulp van het Tien Stappen Plan uitgevoerd.

Allereerst is het kader van de opdracht afgebakend. Hieruit is een Project Initiation Document voortgekomen. Hierin is ook een planning voor de uitvoering opgenomen.

Na het vaststellen van de planning is begonnen met het onderzoek. De resultaten hiervan zijn gebruikt voor het ontwerp van de software. In het ontwerp is rekening gehouden met de use cases, maar ook met de onderhoudbaarheid van het uiteindelijke product. Ook zijn GUI-ontwerpen gemaakt, zodat duidelijk is waarnaar gestreefd moet worden.

Tijdens de realisatie is begonnen met het opbouwen van de data-structuur binnen en buiten het programma. De externe datastructuur bestaat uit een soort database van configuratiefiles, waarin parameter- en weergave gegevens staan opgeslagen. De voor de tool benodigde gegevens moeten van een aantal verschillende locaties worden verkregen. Vanuit deze informatie is vervolgens de user interface gerealiseerd. Hierna is het mogelijk gemaakt een configuratie vanaf de PC in te laden, of op de PC op te slaan. Als laatste is ook de communicatie met de unit geïmplementeerd.

Een aantal medewerkers van binnen Fluidwell en daarbuiten zullen de applicatie testen op gebruiksvriendelijkheid en betrouwbaarheid. Zij zullen een aantal kerntaken (openen bestaande configuratie, opslaan configuratie, configuratie naar unit schrijven) uitvoeren. De resultaten hieruit zullen nog verwerkt worden. Wanneer de applicatie is goedgekeurd, zal deze in gebruik worden genomen.

Binnen de afstudeerperiode is een applicatie gerealiseerd die aan de eisen van de opdrachtgever voldoet, maar nog wel grondig getest moet worden.

Summary

Fluidwell B.V. is a company located in the Netherlands, specifically in a town called Veghel. They have specialised in the development and production of electronic products and services for the process industry. Since founded in 1993, Fluidwell Displays (the first Business Unit) has developed a large number of different products. The first product line consisted mainly of batch controllers.

Over the past twenty years, Fluidwell has developed many different types of displays, for displaying the data retrieved from flow sensors. Most of these displays need to be configured with only three buttons. This also involves the initial configuration that the customer needs to do in order to be able to use the product in their work environment. This can be tedious, especially when multiple devices need to be supplied with the same settings.

For their new product series, the E series, Fluidwell wants to supply a configuration tool. This tool should enable the customer to quickly configure one or multiple devices. The tool should also work with the other (existing) product series. After consulting with Fluidwell, it became clear that every one of their devices, capable of Modbus communication, should be suitable for configuration via the new tool.

First, research into the assignment and wishes of the company was done. The duration of the project, and available hours were determined. A summary of this start-up period is given in de Project Initiation Document.

After this, research was done into various aspects of the project. How can Fluidwells wish of getting a very flexible, but lightweight and easy to maintain configuration tool be realised? The output of this research, was used as input for the software design. In the design, both the use cases, but also maintainability weighed in heavy when making decisions. A few GUI designs were made, to support some of the design choices.

Once the research and design were completed, it was time to implement the design. First, the data-structure, both internal and external (a database, consisting of different types of configuration files) was set up. The data the program needs, needs to be read from a number of different files. With the information found in these locations, it is possible to create the user interface. After this, it was made possible to load configurations from and save configurations to the PC. Lastly, communication to and from the device was made possible.

After the tool is completed, a number of people inside of Fluidwell, but also 'strangers', will test the application for usability and reliability. They will perform a number of key interactions with the tool, and provide feedback. The results of this will still be taken into consideration before releasing the application to the public.

The result of this project is an application that complies with the requirements, but still needs thorough testing.

Verklarende woordenlijst

ASCII	American Standard Code for Information Interchange, gebruikt in het Modbus-protocol
Batch controller	Een apparaat dat de dosering van bijvoorbeeld vloeistoffen of vaste stoffen bestuurt
C++	Een programmeertaal gebaseerd op C
Compiler	Een computerprogramma dat het geschreven programma vertaalt van een hogere programmeertaal naar assembleertaal of machinecode
CRC	Cyclic Redundancy Check, om de geldigheid van een boodschap verzonden via Modbus/RTU te bepalen
CR/LF	Een ASCII karaktercombinatie die een newline aangeeft
Delphi	De software-ontwikkelomgeving voor de objectgeoriënteerde taal Pascal
Display	Een apparaat voor de weergave van data verkregen door bijvoorbeeld flow sensoren
E-serie	De nieuw te lanceren productserie van Fluidwell B.V., en de aanleiding voor dit project
F-serie	De meest succesvolle productserie van Fluidwell B.V.
GUI	Grafische User Interface, het onderdeel van het programma waarmee de gebruiker interactie heeft
IDE	Integrated Development Environment, een code editor
LRC	Longitudinal Redundancy Check, om de geldigheid van een boodschap verzonden via Modbus/ASCII te bepalen
Modbus	Een serieel communicatie protocol, ontworpen voor gebruik met Programmable Logic Controllers, voor het eerst gepubliceerd in 1979
PID	Project Initiatie Document
Pointer	Een C++ variabele die geen object o.i.d. bevat, maar een geheugenadres, en dient als een verwijzing naar een ander object
RAD	Rapid Application Development, een omschrijving voor een manier van applicatie ontwikkeling waarbij een grafische UI-builder centraal staat
RTU	Remote Terminal Unit, gebruikt in het Modbus-protocol
Totalizer	Een apparaat voor het registreren van totalen van een bepaalde functie of meting
TSP	Tien Stappen Plan, richtlijn voor het uitvoeren van een afstudeeropdracht
Unit	Wanneer binnen dit verslag aan een 'unit' wordt gerefereerd, wordt hiermee een product van Fluidwell bedoeld
VCL	Visual Component Library, de GUI library die geïntegreerd is in C++Builder, waarvan de componenten zijn geschreven in Object-Pascal

Lijst van afbeeldingen

Figuur 1; F-serie	10
Figuur 2; E-serie	10
Figuur 3; Organogram Fluidwell, bewerkt van [3]	11
Figuur 4; F-serie vooraanzicht	12
Figuur 5; TSP van Kempen en Keizer met aanpassing van Stap 7 [2]	14
Figuur 6; Voorbeeld oud configuratiefile	18
Figuur 7; Gekozen filestructuur	19
Figuur 8; Voorbeeldpagina 'Communication'	20
Figuur 9; Relatiediagram configuratiefiles	21
Figuur 10; Unit description file	22
Figuur 11; Tabsheet description file	22
Figuur 12; Parameterset description file	22
Figuur 13; Dependency description file	23
Figuur 14; Language file	23
Figuur 15; Opstart file	23
Figuur 16; Configuratie file	24
Figuur 17; Datastructuur	24
Figuur 18; Ontwerp configuratiepagina 'Analog Output'	25
Figuur 19; Structuur GUI klassen	26
Figuur 20; Overzichtsdiagram klassen, versie begin realisatie	27
Figuur 21; Weergave van de GUI zoals deze er in de UI-builder uit ziet	28
Figuur 22; Dynamisch gegenereerde GUI (h = 160px; b = 300px)	29
Figuur 23; Instelling K-factor in test-unit	30
Figuur 24; Screenshot Configuration tool runtime	31

Lijst van tabellen

Tabel 1; Verschil Modbus/ASCII en Modbus/RTU	26
--	----

Lijst van codefragmenten

Codefragment 1; Genereren GUI-elementen met VCL	29
Codefragment 2; Voorbeeld afvangen click-event op dynamisch gegenereerde button	30

1. Inleiding

Fluidwell is een Nederlands bedrijf, met het hoofdkantoor in Veghel, gespecialiseerd in de ontwikkeling en productie van elektronische producten en diensten voor de procesindustrie. Sinds de oprichting in 1993, heeft Fluidwell Displays (de eerste business unit) diverse producten ontwikkeld. De eerste productlijn bestond voornamelijk uit batch controllers.

In de afgelopen 20 jaar zijn door Fluidwell veel verschillende typen displays ontwikkeld voor de weergave van data verkregen door bijvoorbeeld flow sensoren. De meeste van deze units worden met slechts drie hardware-knoppen bediend. Hiermee moet ook de initiële configuratie door de klant worden uitgevoerd. Wanneer meerdere units van dezelfde configuratie moeten worden voorzien, wordt dit erg tijdrovend. Bij de nieuwe productserie (de E-serie), wil Fluidwell daarom een configuratietool leveren. Deze tool zou ook geschikt moeten zijn voor de andere productseries (de B-, D-, F- en N-serie). In totaal gaat het om ca. 50 basismodellen, waarvan meer dan 100.000 varianten mogelijk zijn.

In overleg met Fluidwell is bepaald wat er moet gebeuren: er moet een configuratietool komen, waarmee alle producten die via het Modbus-protocol kunnen communiceren, geconfigureerd kunnen worden. De opdracht is met behulp van het Tien Stappen Plan uitgevoerd.

Toen de requirements vastgesteld waren, is begonnen met het onderzoek. In deze periode is onderzocht op welke manier het mogelijk is een dynamische applicatie te ontwikkelen waarmee alle producten van Fluidwell eenvoudig en snel geconfigureerd kunnen worden.

Aan de hand van de onderzoeksresultaten is ervoor gekozen de applicatie volledig dynamisch op te bouwen. Dit houdt in dat er in principe geen product-specifieke informatie in de tool is vastgelegd. Hierdoor is het mogelijk bestaande en nieuwe producten met dezelfde applicatie te configureren, zonder dat er een applicatie-update nodig is.

Na het onderzoek is een ontwerp van de applicatie en de bijbehorende database gemaakt. De database bestaat uit een aantal losse files, waarin alle benodigde informatie staat vastgelegd. Aan de hand van dit ontwerp is vervolgens de applicatie gerealiseerd.

In hoofdstuk 2 wordt informatie gegeven over het bedrijf. Vervolgens wordt in hoofdstuk 3 de opdracht in detail omschreven. In hoofdstuk 4 komen de gebruikte methoden en tools aan bod, waarna in hoofdstuk 5 de resultaten van het uitgevoerde onderzoek zijn opgenomen. Vervolgens wordt het productontwerp in hoofdstuk 6 besproken. Hoofdstuk 7 zal ingaan op de implementatie van het ontwerp, en hoofdstuk 8 bevat de conclusie en aanbevelingen.

2. Het bedrijf

De opdracht wordt uitgevoerd voor Fluidwell B.V. In dit hoofdstuk wordt kort uitgelegd wat voor bedrijf Fluidwell is, welke producten zij produceren, en hoe de afstudeerder binnen de organisatie past.

2.1 Organisatie

Fluidwell is een Nederlands bedrijf met het hoofdkantoor in Veghel. Onder leiding van directeur Eric Rath heeft het bedrijf zich ontwikkeld tot marktleider op het gebied van flowcomputers. Het bedrijf bestaat uit drie 'business units'; Fluidwell Displays, Fluidwell Instruments en ProcessMonitor.

Sinds de oprichting van het familiebedrijf in 1993, heeft Fluidwell Displays (de eerste business unit) verschillende productlijnen ontwikkeld. De eerste productlijn (300-serie) bestond voornamelijk uit batch controllers.

Fluidwell is gespecialiseerd in de ontwikkeling en productie van elektronische producten en diensten voor de procesindustrie. Inmiddels bestaat het bedrijf uit ca. 50 medewerkers, waarvan 42 op het hoofdkantoor in Veghel. Verder zijn er een aantal mensen actief in de Verenigde Staten, Duitsland en Australië. [4]

2.2 Producten

Fluidwell beschikt over een breed en flexibel product portfolio, bestaande uit standaard producten, welke te verdelen zijn in een aantal product ranges: flow computers, flow rate/totalizers, batch controllers, indicators en monitoring systemen, met heel veel opties. Er kunnen hierdoor veel verschillende versies geproduceerd worden.

Fluidwell Displays ontwikkelt apparaten (displays), die kunnen worden aangesloten op een flowmeter. De flowmeter is bijvoorbeeld aangesloten op een pijpleiding. Deze meter stuurt signalen (pulsen, stroom, spanning, etc.) naar de display, die vervolgens het signaal meet, telt en conditioneer. De Fluidwell display laat dan de meetdata zien. De meest succesvolle productserie van Fluidwell Displays is de F-serie. Verdere productseries in het portfolio zijn de B-, D-, en N-serie.



Figuur 1; F-serie

Een nieuw uit te brengen platform is de E-serie. Deze serie bestaat uit gebruiksvriendelijke, explosieveilige indicators en totalizers, bedoeld voor gebruik in zware omstandigheden.

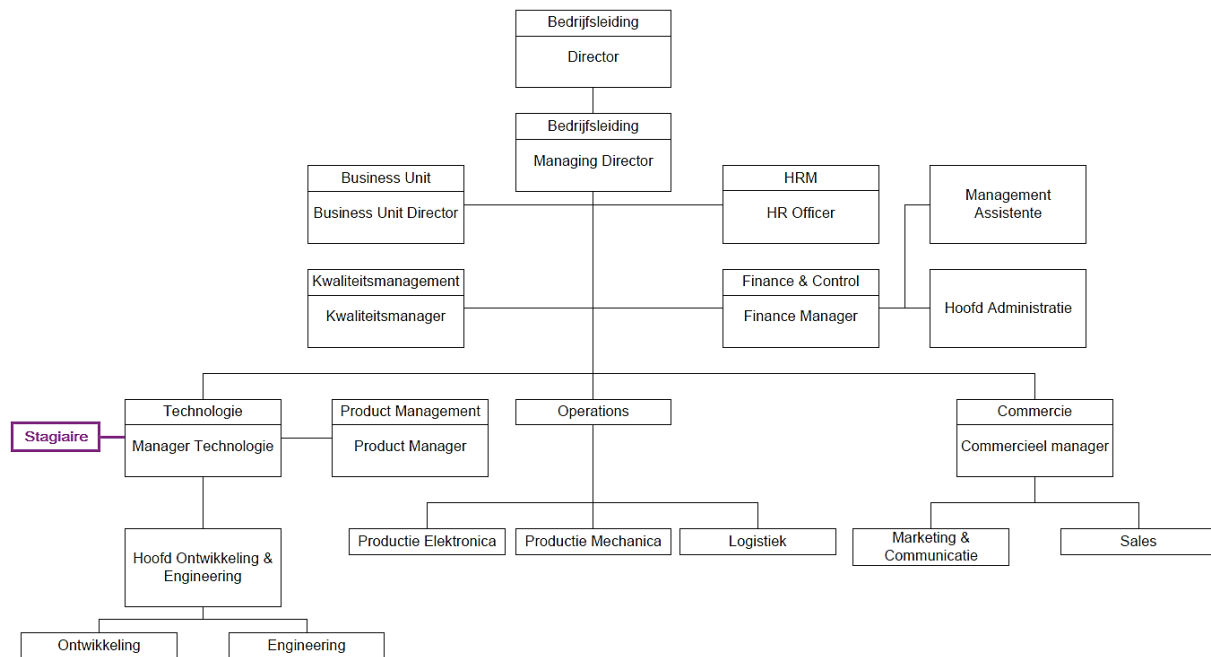
Deze opdracht zal in eerste instantie betrekking hebben op producten uit de E-serie, maar moet ook 'out of the box' werken met producten uit de andere series die via Modbus communiceren.



Figuur 2; E-serie

2.3 Positie in de organisatie

De werkzaamheden vinden plaats binnen de business unit 'Fluidwell Displays'. Onder de afdeling Technologie vallen de afdelingen Ontwikkeling en Engineering (figuur 3). Ontwikkeling is de afdeling waar de nieuwe producten ontwikkeld worden. Hieronder vallen de hard- en software engineers. Engineering maakt deze producten vervolgens klaar voor productie. De afstudeerder rapporteert aan de Manager Technologie. Dit is tevens de afstudeerbegeleider.



Figuur 3; Organogram Fluidwell, bewerkt van [3]

3. De opdracht

Het is belangrijk de opdracht duidelijk in kaart te brengen. In dit hoofdstuk wordt uiteengezet wat de huidige situatie is, wat de opdrachtgever hoopt te bereiken, en op welke manier dit wordt aangepakt.

3.1 Situatie

Op dit moment moeten de klanten van Fluidwell de apparaten één voor één met de hand configureren. De meeste van deze apparaten hebben slechts drie knoppen (figuur 4), waarmee de gehele configuratie gedaan moet worden. Doordat er geen configuratieprogramma beschikbaar is, is het ook niet mogelijk meerdere identieke units snel van dezelfde configuratie te voorzien.



Figuur 4; F-serie vooraanzicht

De nieuwste productserie, de E-serie, staat op het punt vrijgegeven te worden. Waar de F-serie nog werkt met echte knoppen, zijn deze bij de E-serie vervangen door infrarood toetsen. Het programmeren van de units wordt hierdoor niet gemakkelijker, deze zijn namelijk trager. Hierom, en om de concurrentiepositie te versterken, zal er iets gedaan moeten worden om de klant tegemoet te komen. Een concurrent van Fluidwell levert namelijk bij een soortgelijk product wel een configuratietool mee.

3.2 Doel

Het doel voor Fluidwell is een configuratieprogramma bij de producten te kunnen leveren waarmee alle producten (bestaand of nieuw) die via Modbus kunnen communiceren, geconfigureerd kunnen worden. Ook moet het mogelijk zijn een configuratie op de PC op te slaan, om zo snel meerdere units van dezelfde configuratie te kunnen voorzien.

Het ondersteunen van alle Fluidwell producten maakt het lastiger om een one-size-fits-all configuratieprogramma te ontwikkelen, dat geen grote hoeveelheid overbodige informatie bevat.

3.3 Opdrachtomschrijving

Er moet een applicatie komen waarmee de klant eenvoudig de gewenste configuratie voor een bepaald product kan instellen, ongeacht uit welke productserie. Dit moet kunnen door een product uit een lijst te selecteren, of door een product op de PC aan te sluiten. De configuratietool moet dan de default configuratie weergeven, of aanbieden de huidige configuratie uit de unit te lezen. Vervolgens moet de klant deze configuratie op zijn PC kunnen opslaan, en/of naar de unit kunnen schrijven. Voor de communicatie met de unit moet een nieuwe Modbus driver ontworpen worden, om de bestaande en verouderde driver te vervangen. Ook moet de klant een eerder op de PC opgeslagen configuratie kunnen openen, om deze vervolgens te bewerken, of naar de unit te schrijven. Idealiter kan het programma ook log data uit lezen, en dit exporteren naar csv. Er zijn echter nog geen units die dit ondersteunen.

Tevens moet de applicatie zelf nog aan een aantal eisen voldoen. De tool zelf moet geschreven worden in de programmeertaal C++. Hierdoor kunnen de Software Engineers van Fluidwell, die voornamelijk embedded software schrijven, de tool zelf onderhouden. Om deze reden moet ook gebruik gemaakt worden van de ontwikkelomgeving C++Builder (XE7). Het is ook belangrijk dat de tool onderhoudsarm is, en niet bij iedere nieuwe unit/firmware versie geupdate moet worden.

Er is vroeg in het traject een opdracht toegevoegd, namelijk het maken van een losse tool voor het uitlezen van logdata van een specifieke productserie (E-serie). Dit wordt een losse applicatie, zodat het voor minder geschoold personeel mogelijk is de logdata uit te lezen, zonder per ongeluk de configuratie van de unit aan te passen. Deze functionaliteit zal op termijn ook geïntegreerd moeten worden met het configuratieprogramma, hier dient bij het ontwerpen rekening mee gehouden te worden.

Gaandeweg is gebleken dat de logdata-functionaliteit niet op tijd in de units geïmplementeerd zou zijn, om de logdata tool tijdens de afstudeerperiode te kunnen afronden. Om deze reden is de logdata tool gerealiseerd tot er informatie nodig was die niet beschikbaar was (de structuur van de records etc.). Hierna is de aandacht volledig op de configuratietool gericht.

Tijdens het ontwerpen van de configuratietool en Modbus driver is gebleken dat er niet voldoende tijd was om ook de Modbus driver te herschrijven. Deze is wel (deels) meegenomen in het ontwerp, maar niet gerealiseerd. Er is gekozen gebruik te maken van een aangepaste versie van de bestaande Modbus driver.

Tijdens de realisatiefase is de wens voor meertaligheid, en help-functionaliteit uitgesproken. Dit is vervolgens aan de opdracht toegevoegd. Hiervoor waren aanpassingen in het ontwerp nodig.

De opdracht is aangepakt met behulp van de methode Tien Stappen Plan (TSP), aangepast voor ontwikkelopdrachten. [2] In hoofdstuk 4 wordt dit nader toegelicht.

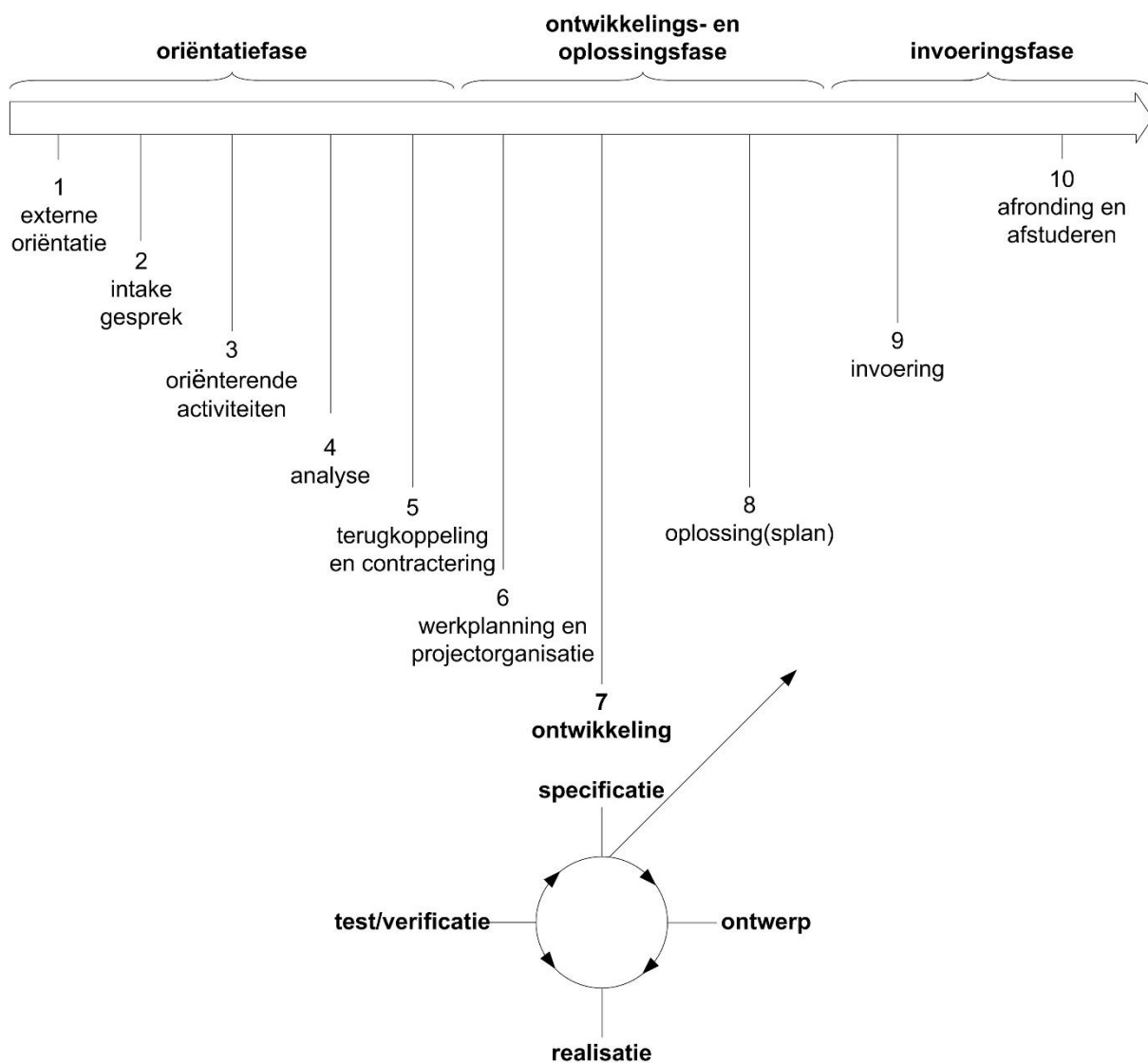
4. Methoden & Tools

In dit hoofdstuk zal kort worden beschreven van welke methoden en tools gebruik is gemaakt voor de aanpak en uitvoering van deze opdracht.

4.1 Methoden

4.1.1 Tien Stappen Plan

Er is gebruik gemaakt van een licht aangepaste versie van het Tien Stappen Plan (TSP) van Kempen en Keizer. [2] “Stap 7 – Diepteonderzoek” is vervangen door “Stap 7 – Ontwikkeling”, om het TSP geschikt te maken voor het uitvoeren van een afstudeeropdracht waar een product ontwikkeld moet worden.



Figuur 5; TSP van Kempen en Keizer met aanpassing van Stap 7 [2]

In Stap 7 zal in plaats van het Diepteonderzoek dus de Ontwikkeling van het product plaatsvinden. Dit onderdeel bestaat uit de gebruikelijke ontwikkelings activiteiten: specificatie, ontwerp, realisatie en verificatie (figuur 5). Wel zal er ondersteunend onderzoek worden verricht.

Het Project Initiatie Document (PID) is op basis van basis van de resultaten van stappen 1 t/m 6 van het TSP opgesteld. In dit document zijn zaken die betrekking hebben op de uitvoering van de

opdracht, zoals aanpak, kosten en doorlooptijd, planning en eindresultaat opgenomen. Tijdens de uitvoering van het project, is het PID als leidraad gebruikt. Het PID is te vinden in Bijlage I. De inhoud van de scriptie bestaat met name de stappen 7 en 8.

4.1.2 PRINCE2

Voor de projectplanning is gebruik gemaakt van de PRINCE2 methode. Dit is een methode voor projectmanagement, die gericht is op de fasering van een project. Elke fase moet afgesloten zijn, voor aan een volgende fase kan worden begonnen. Naast het PID, is een plan van aanpak geschreven, waarin de projectplanning is opgenomen. De gebruikte fasen zijn: verkenning, ontwerp, realisatie en testen.

4.2 Tools

De opdrachtgever heeft aangegeven dat het configuratieprogramma in C++ geschreven moet worden. Dit deels in verband met de onderhoudbaarheid door Fluidwell zelf. De reeds beschikbare tools (zoals de Modbus-driver die zij zelf gebruiken voor de initiële configuratie van de units) zijn ook in C++ geschreven. Hierbij hoort een component die afhankelijk is van de Visual Component Library (VCL), wat een product is van Embarcadero en hierdoor alleen beschikbaar is in C++Builder. Eveneens is het hierdoor alleen mogelijk de standaard compiler te gebruiken.

VCL is een framework dat door Borland is ontwikkeld, voor het gebruik met hun IDE voor Object Pascal, Delphi. De VCL is dan ook geschreven in Pascal. Toen later C++Builder ontwikkeld werd, is besloten de VCL niet naar C++ te vertalen. Bij gebruik van dit framework in C++Builder, worden bij het compileren C++ header files gegenereerd uit de Delphi componenten. VCL componenten kunnen ook in C++ geschreven worden, maar erven altijd van Object Pascal klassen. Er zal voor het genereren van de GUI gebruik worden gemaakt van de mogelijkheden die de VCL biedt. [5]

5. Onderzoek

Er is onderzocht of het mogelijk is de gevraagde applicatie te ontwikkelen. Tevens is onderzocht met welke hulpmiddelen dit gerealiseerd kan worden. De resultaten van dit vooronderzoek worden in dit hoofdstuk uiteengezet.

Om de opdracht duidelijk in kaart te brengen is eerst op verschillende manieren onderzoek gedaan. Er is uitvoerig gesproken met de opdrachtgever om de requirements vast te leggen. Een overzicht van de requirements is te vinden in Bijlage II.

Hierna is begonnen met het onderzoeken van de mogelijkheden voor de configuratietool zoals de opdrachtgever deze voor ogen heeft. Hieruit is de volgende onderzoeksvraag voortgekomen: *Op welke manier is het mogelijk een dynamische applicatie te ontwikkelen waarmee alle verschillende producten van Fluidwell (die via Modbus kunnen communiceren) eenvoudig en snel geconfigureerd kunnen worden, zonder overbodige componenten?*

5.1 Applicatie opbouwen

Deelvraag: Hoe kan de GUI/applicatie worden opgebouwd op basis van het aangesloten of geselecteerde product, zodanig dat deze alleen de voor deze unit relevante opties weergeeft?

Voor het opbouwen van de applicatie zijn een aantal mogelijkheden. Omdat het met de applicatie mogelijk moet zijn veel verschillende apparaten te configureren, bestaat het gevaar dat de gebruiker veel overbodige informatie voorgeschoteld krijgt. Om dit te voorkomen, moet hier met de opbouw van de GUI rekening mee worden gehouden. Er zijn een drietal mogelijkheden onderzocht.

Het is mogelijk, voor ieder product een aparte GUI te maken. Deze kan dan vervolgens worden getoond, wanneer dit bepaalde product wordt gedetecteerd. Behalve dat dit het programma onnodig groot maakt, betekent dit ook dat bij het toevoegen van een nieuw product, een nieuwe GUI gebouwd moet worden, en de gebruiker het hele programma moet updaten.

Een andere mogelijkheid is het bouwen van een universele GUI, welke bepaalde elementen wel of niet laat zien, afhankelijk van het gekozen/aangesloten apparaat. Het programma wordt hierdoor groot, en moeilijker onderhoudbaar. Bepaalde parameters verschillen per apparaat iets van elkaar, waardoor er een aantal GUI componenten ontstaan die erg veel op elkaar lijken, maar net niet hetzelfde zijn.

Een derde mogelijkheid is om de GUI volledig dynamisch op te bouwen. De benodigde elementen staan dan niet in het programma opgeslagen. Aan de hand van een gekozen of gedetecteerde unit, kan dan met een database de benodigde informatie worden verkregen om de juiste parameters weer te geven.

Er is gekozen voor het dynamisch opbouwen van de GUI aan de hand van een database. De tool is hierdoor eenvoudiger onderhoudbaar. Er hoeft geen nieuwe versie van de applicatie te worden uitgebracht wanneer een nieuwe unit wordt gelanceerd. Daardoor is geen programmeur meer nodig,

want de database kan door minder geschoold personeel worden geupdate. Ook blijft de tool hierdoor kleiner.

Wel heeft dit invloed op de complexiteit van de applicatie. Dit houdt in dat de ontwikkeling langer zal duren, maar dat hier later winst wordt behaald omdat de applicatie nauwelijks onderhouden hoeft te worden.

5.2 Benodigde gegevens

Deelvraag: Wat is de beste manier om de applicatie te laten weten welke parameters voor een bepaalde unit/firmware-versie moeten worden ingesteld?

Net als voor het opbouwen van de GUI, zijn hier meerdere opties. Bij een aantal van de eerder toegelichte mogelijkheden (§5.1), zouden alle parameters hard in het programma gecodeerd staan. Dit houdt in dat er ergens een lijst moet staan waarin staat aangegeven welke unit welke parameters nodig heeft. Deze lijst kan in het programma zelf worden bijgehouden.

Een andere mogelijkheid is om deze lijst in een aparte file op te slaan, die dan door het programma wordt ingelezen. Dit heeft als voordeel dat bij een nieuwe unit niet iedere keer het hele programma opnieuw gedownload hoeft te worden door de gebruiker, maar dat het programma zelf de nieuwe lijst kan downloaden wanneer deze beschikbaar is. Wanneer er echter niet alleen een nieuw product, maar ook een nieuwe parameter wordt toegevoegd, moet deze wel in het programma worden toegevoegd. In beide gevallen wordt het programma groter, en bevat het veel informatie die waarschijnlijk niet vaak gebruikt wordt.

Het is ook een optie de externe lijst met benodigde parameters te combineren met de database die eerder is gekozen. Zowel de parameters als de lijst worden dan dus niet in het programma gedefinieerd. Hierdoor blijft de tool zelf zo klein en onderhoudbaar mogelijk. Bij een nieuw product hoeven dan alleen een paar nieuwe files gedownload te worden, maar hoeft het programma zelf niet geupdate te worden. Omdat deze mogelijkheid het meest compatibel is met de eerdere keuze, is hiervoor gekozen. Vervolgens is onderzocht op welke manier deze files het beste gestructureerd kunnen worden.

5.3 Filestructuren/Database

Deelvraag: Op welke manier kan de applicatie compatibel zijn met nieuwe firmware-versies, zonder dat hiervoor een update nodig is?

In de productieomgeving gebruikt Fluidwell configuratiefiles. Het idee onstond deze files als basisidee te gebruiken voor het vastleggen van de benodigde informatie voor het configuratieprogramma. De mogelijkheid een database zoals MySQL te gebruiken is niet onderzocht.

De parameters staan in het configuratiemenu van de units (en de manuals) gegroepeerd binnen een aantal submenu's (bijvoorbeeld "Total", "Flowrate"). Om dit voor de gebruiker zo vertrouwd mogelijk weer te geven, kan ervoor gekozen worden de verschillende submenu's op aparte tabbladen weer te geven.

```

81 [FLOWRATE2]
82 NAME=      Flowrate 2
83 FLOWUNIT2=  224,Unit,          2,mL,L,m3,mg,g,kg,ton,GAL,bbl,lb,No unit
84 FLOWINT2=   225,Interval,      2,/second,/minute,/hour,/day
85 FLOWDEC2=   226,Decimals,      2,0,1
86 FLOWK2=     227,K-Factor,      4
87 FLOWKDEC2=  230,K-Decimals,    2,0,1,2,3,4,5,6
88 FLOWPULS2=  231,Pulses,        2
89 FLOWOFF2=   232,Cut-Off,        2
90 FLOWRATE2=  588,Flowrate,      4
91
92 [FLOWRATEDIF]
93 NAME=      Flowrate Difference
94 FLOWRATEDIFFER= 58,Flowrate,  4
95
96 [FLOWRATEGUARD]
97 NAME=      Flowrate Guarding
98 FLOWMIN=    234,Flowrate Min,   4
99 FLOWMAX=    237,Flowrate Max,   4
100
101 [RATIO]
102 NAME=      Ratio
103 RATIOFORM=  85,Formula,         2,B/A,B/(A+B),B/(A-B),(B-A)/A,(B-A)/B
104 RATIOFOR2M= 86,Display Form,    2,1 : 1...,Percentage
105 RATIODEC=    88,Decimals,       2,0,1,2,3
106 RATIOFILTER= 87,Filter Value,    2
107 RATIO=       82,Current Ratio,   4
108
109 [DISPLAY]
110 NAME=      Display
111 FUNCTION=   64,Function,         2,Total,Flowrate
112 ACCSHOW=    65,Acc.Total Show,   2,Off,On

```

Figuur 6; Voorbeeld oud configuratiefile

In figuur 6 is een voorbeeld gegeven van een bestaande configuratiefile. Wanneer de GUI aan de hand van deze files gegenereerd zou moeten worden, zou hiervoor informatie moeten worden toegevoegd. Hierbij moet worden gedacht aan weergave informatie (hoe moet deze parameter worden weergegeven, radio button, input veld, etc.), ranges en tekstuele toevoegingen zoals bijvoorbeeld helpteksten. Er moet dus een nieuwe filestructuur ontworpen worden. Hiervoor zijn een aantal mogelijkheden onderzocht.

Als eerste is uitgegaan van één groot file per unitgroep, waar alle voor deze groep mogelijke/gelijke variabelen in staan. Hiernaast is er een unitspecifiek file, welke dan wijst naar de variabelen die voor een bepaalde unit nodig zijn. De variabelen staan in het eerstgenoemde file gegroepeerd op tabblad. Bij een update/nieuw model moet dan eventueel het groep-file worden aangepast, en een unitspecifiek file worden gecreëerd. Een nadeel is, dat zelfs binnen dezelfde unitgroep, veel ongeschijnlijk dezelfde parameters, net weer andere eigenschappen hebben.

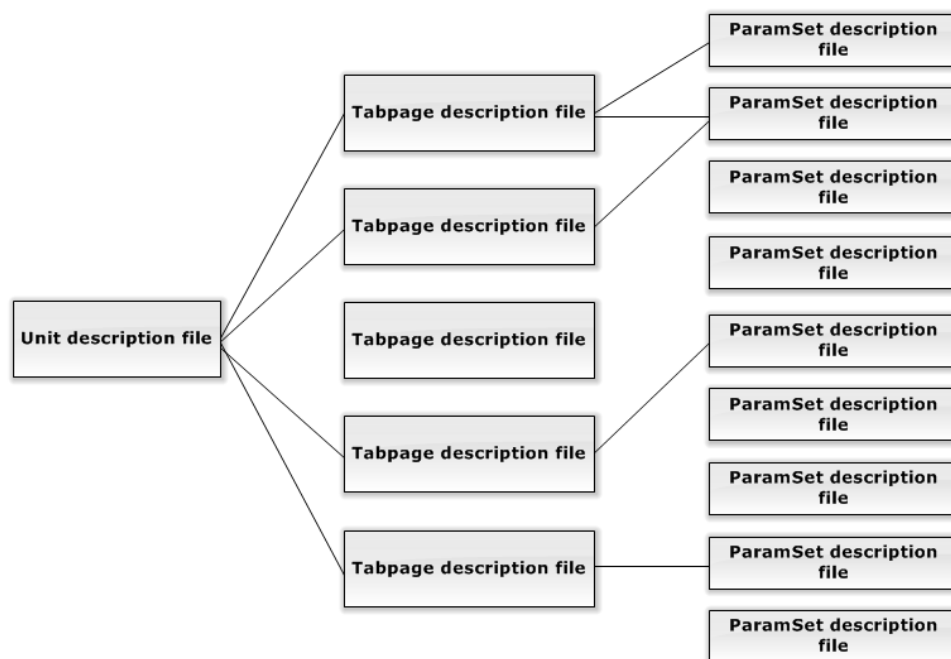
Er kan ook worden uitgegaan van één file per tabblad. Een tabblad is dan een bepaalde versie van één bepaald menu. Het unitspecifieke file wijst dan naar alle tabbladfiles die nodig zijn voor de geselecteerde unit. Het tabbladfile bevat de verwijzingen naar de benodigde parameters. De parameters staan allemaal samen in een parameterfile. Een nadeel is dat dit parameterfile erg groot kan worden, bijvoorbeeld doordat een aantal parameters meerdere keren voor moeten komen,

omdat ze net andere eigenschappen hebben. Bij een update moet eventueel een nieuw tabblad file worden gemaakt, en een unitspecifiek file worden gecreëerd.

Een derde mogelijkheid is om uit te gaan van dezelfde constructie tabbladfiles als hiervoor genoemd, maar met hiernaast files met parametersets. Het unitspecifieke file verwijst dan naar een tabbladfile, en geeft deze de gewenste parameterset mee. Het tabbladfile bevat de weergave informatie.

Een nadeel van deze structuur is dat er veel verschillende files onderhouden moeten worden. Een voordeel is dat veel van de files hergebruikt kunnen worden (bijvoorbeeld door een bestaande tabbladfile naar een nieuwe parameterset te laten wijzen).

De keuze is gevallen op de laatstgenoemde mogelijkheid. Deze optie wordt in figuur 7 toegelicht. Dit biedt betere flexibiliteit, en hoewel er meer files nodig zijn, zijn een aantal files (bijvoorbeeld de parametersets) eenvoudiger te onderhouden dan in de overige mogelijkheden.



Figuur 7; Gekozen filestructuur

5.4 Conclusies onderzoek

De wens van de opdrachtgever is om de applicatie zo klein en flexibel mogelijk te maken. Door minder informatie in de code vast te leggen, wordt de tool eenvoudiger onderhoudbaar. Er is dan niet voor iedere aanpassing een programmeur nodig.

Uit het onderzoek toegelicht in de voorgaande paragrafen, blijkt dat het mogelijk is een dynamische applicatie te ontwikkelen waarmee de producten van Fluidwell kunnen worden geconfigureerd. Er is gebleken dat het volledig dynamisch opbouwen van de applicatie (§5.1 & §5.2), en dus nagenoeg niets in de code vast te leggen, de voorkeur heeft. Dit is te realiseren door het gebruik van een externe database. Uit de onderzochte mogelijkheden in §5.3 komt naar voren dat het mogelijk is een filestructuur te ontwerpen waarmee de applicatie kan worden opgebouwd. Met deze filestructuur is het dan ook mogelijk bestanden te hergebruiken.

6. Ontwerp

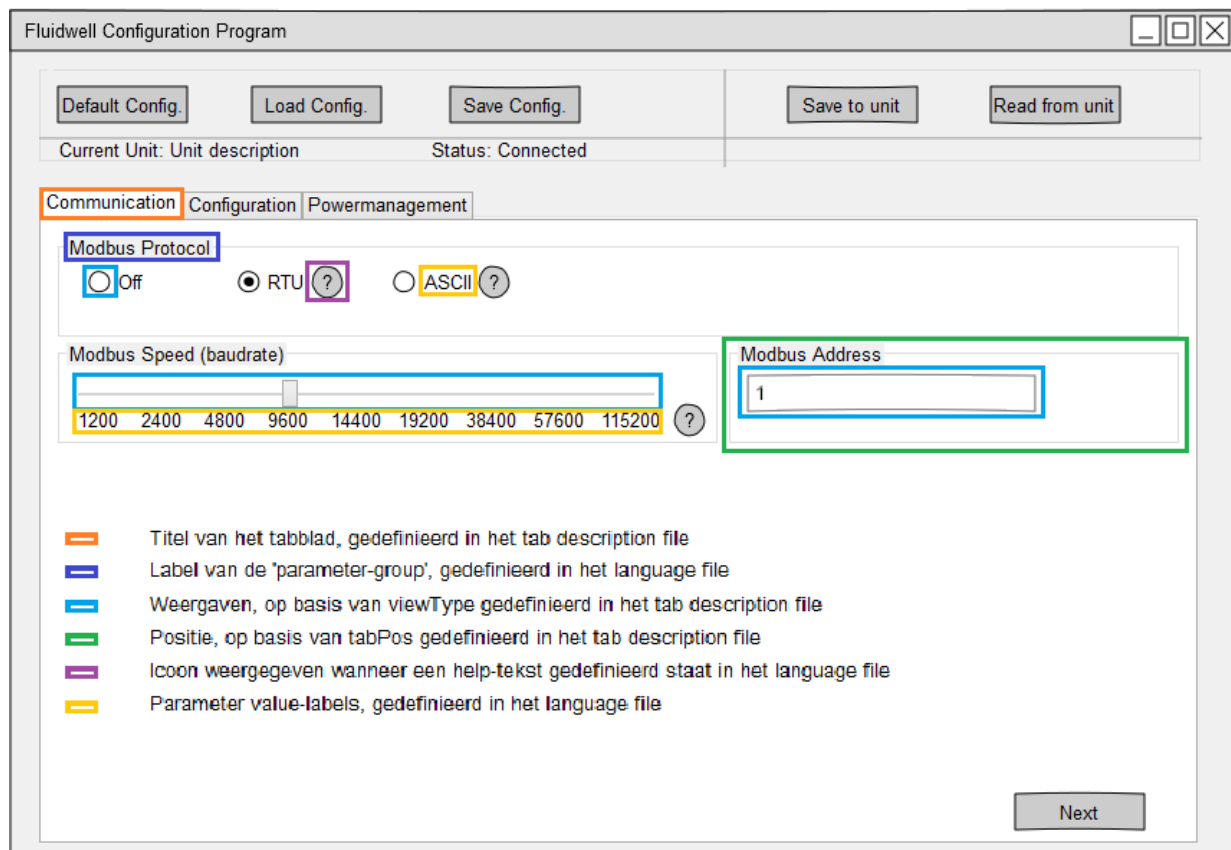
Met behulp van de uit het onderzoek verkregen informatie, is begonnen aan een ontwerp voor de te realiseren software. In het volgende hoofdstuk worden de belangrijkste aspecten uit dit ontwerp toegelicht. Een light-versie van het ontwerpdocument is te vinden in Bijlage II.

6.1 Uitgangspunten ontwerp

Bij het ontwerpen van de configuratietool, is uitgegaan van de User Interface als belangrijkste facet. Er is voor gekozen aanwijzingen e.d. via help-icoontjes rechtstreeks bij de betreffende componenten te plaatsen. Hierdoor is het herkenbaar, en kunnen mensen kiezen of ze de tips wel of niet bekijken. Doordat hiervoor geen apart venster hoeft te worden geopend, is het wel iets toegankelijker dan de traditionele (F1) help functie.

Ook meertaligheid is een aspect waarmee rekening is gehouden, zodat dit zonder verdere aanpassingen toegevoegd kan worden. Fluidwell is een wereldwijde speler, en hoewel Engels de belangrijkste voertaal is, is het zeker ook interessant om klanten de mogelijkheid te bieden de taal te wijzigen.

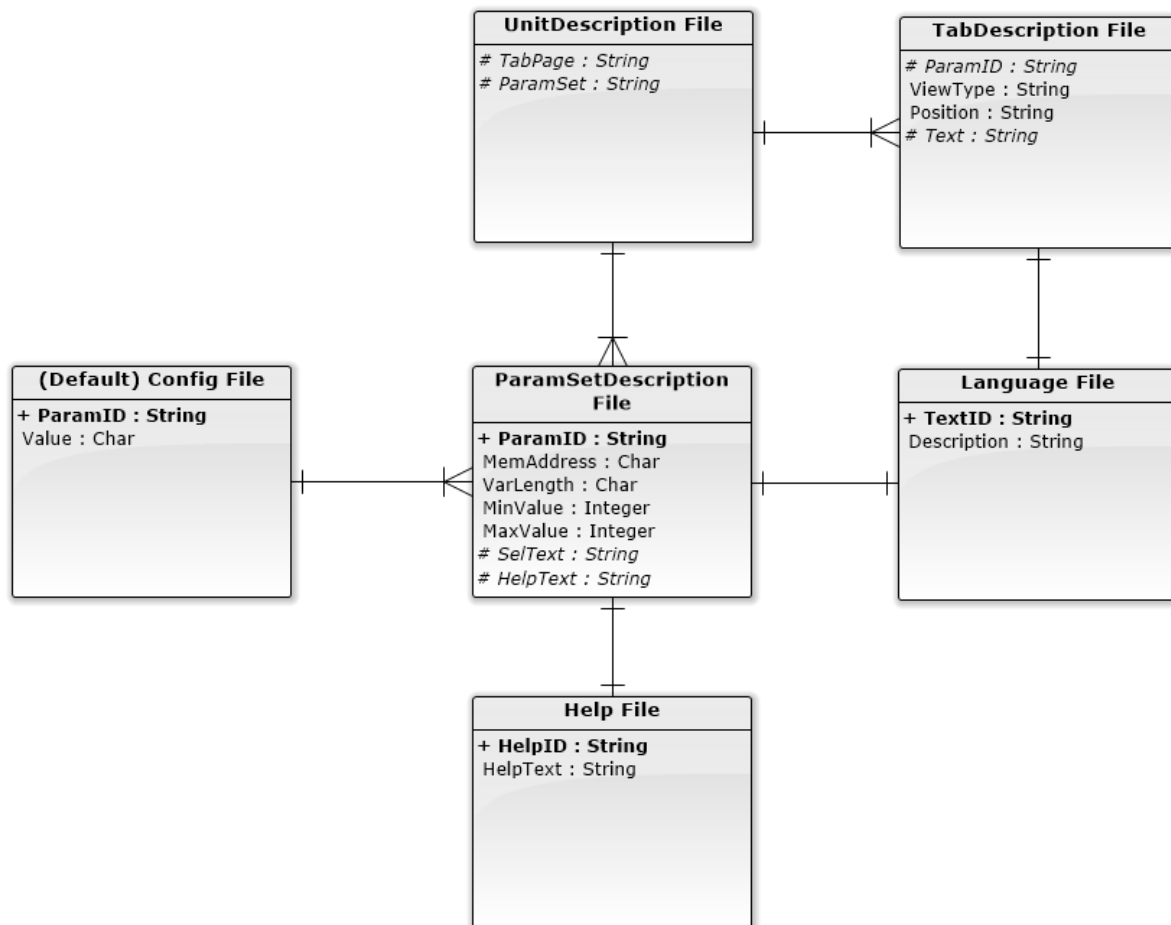
Er is onderzocht welke informatie nodig is, om de te configureren parameters zo goed mogelijk weer te geven. Vervolgens is een datastructuur ontworpen, waarmee dit te realiseren zou moeten zijn. In figuur 8 is weergegeven uit welke componenten de GUI globaal bestaat, en waar de informatie die voor deze componenten benodigd is gevonden kan worden in de filestructuur. Ieder tabblad en iedere groupbox zijn de componenten zoals beschreven in §6.5, Genereren GUI.



Figuur 8; Voorbeeldpagina 'Communication'

6.2 Externe datastructuur

Gebaseerd op de 'oude' configuratiefiles (figuur 6), is een nieuwe filestructuur bedacht, welke deels is weergegeven in figuur 9.



Figuur 9; Relatiediagram configuratiefiles

Naast de in bovenstaande figuur weergegeven filetypes zijn nog een aantal andere files nodig. Allereerst is er een file waarin staat aangegeven welk unit-file voor welke unit-code moet worden ingelezen. Daarnaast is er nog het dependency description file. Als laatste zijn er nog een aantal generieke files, zoals de language file, en de help file. Deze file types worden verder toegelicht.

6.2.1 Unit description file

Dit is het unitspecifieke file. Hierin staan de verwijzingen naar de benodigde tabbladen, met de benodigde parametersets. De tabbladen staan in de volgorde waarin ze moeten worden weergegeven.

```

16 [TABPAGES]
17 TRUNTIMEV2=RUNTIME1,Z1
18 TTOTALV1=TOTAL1,A1
19 TFLOWRATEV1=FLOWRATE1,B1
20 TDISPLAYV2=DISPLAY,D2
21 TFLOWMETERV1=FLOWMETER,J1
22 TANALOGV1=ANALOG,C2
23 TPOWERMANV1=POWERMAN,P1
24 TRELAYSV2=RELAYS,R2
25 TCOMMUNICV1=COMMUNIC,G1
26 TOTHERSV1=OTHERS,O1

```

Figuur 10; Unit description file

De 'code' voor het =-teken, geeft het Tabsheet description-file aan. Vervolgens wordt het id gegeven, aan de hand waarvan de benodigde tekst in het Language description file kan worden opgezocht. Als laatste staat de Parameterset aangegeven. Hiermee kan de juiste set worden ingelezen.

6.2.2 Tabsheet description file

In het Tabsheet description file staat een lijst met parameters (id's), behorende bij de categorie. Hierachter staat het viewType (het type weergave) voor de parameter gedefinieerd.

```

1 TOTUNIT=radio
2 TOTDEC=updown
3 TOTK=intTF
4 TOTKDEC=slider

```

Figuur 11; Tabsheet description file

Er zijn een aantal mogelijke viewType's. Naast de hierboven weergegeven types is het bijvoorbeeld nog mogelijk een 'txtTF', een 'combo' voor een option list, of een 'table' type mee te geven.

6.2.3 Parameterset description file

Ook hier is een lijst met parameters gedefinieerd met id's. De overige informatie in dit file heeft met name betrekking op het opslaan van de parameter in de unit. Zo staat er bijvoorbeeld het memory address en de lengte van de variabele in bytes. Ook is hier de minimale en maximale waarde van een parameter vastgelegd, indien van toepassing. Verder is hier een verwijzing naar de languagefile en eventueel de helpfile.

```

1 [SIGNAL1]
2 SIGNAL=96;2;0;8;SELSIG;Signal;SIGNAL
3 FORMULA=98;2;0;2;SELFFORM;Formula;FORMULA
4 FILTER=99;2;-;-;FILTER;Filter;FILTER
5 METERCUT=100;2;-;-;METERCUT;Cut-off;METERCUT
6 METERLOCALIB=102;2;0;2;SELCAL;Calibrate Low;METERLOCALIB
7 METERHICALIB=103;2;0;2;SELCAL;Calibrate High;METERHICALIB
8 METERLOVAL=104;2;-;-;METERLOVAL;Value Low;METERLOVAL
9 METERHIVAL=106;2;-;-;METERHIVAL;Value High;METERHIVAL
10 SIGNALCOIL=108;2;0;8;SELCOIL;Signal Ext;SIGNALCOIL

```

Figuur 12; Parameterset description file

6.2.4 Dependency description file

Hierin staat aangegeven welke parameters van elkaar afhankelijk zijn. Deze afhankelijkheden gelden met name voor de weergave. De waarde van de variabele FLOWUNIT en FLOWINT worden bijvoorbeeld gebruikt om de eenheid van de flowrate (lBIRATE) weer te geven (bijvoorbeeld L/min).

```
2  lBIRATE=FLOWK+FLOWKDEC
3  lBIRATE=TOTK+TOTKDEC
4  lBIRATE=FLOWUNIT+FLOWINT
5  tFIFLOWVALUE=FLOWRATE+FLOWDEC
6  tFDAC_RANGE=FLOWDEC
```

Figuur 13; Dependency description file

6.2.5 Language en Help file

In het Language file staan alle benodigde stukjes tekst vastgelegd. Dit zijn bijvoorbeeld de labels van parameters of tabbladen, maar ook de teksten die bij een lijst met opties worden weergegeven. Dit file is erg eenvoudig van opzet.

```
29  BAUDRATE=Baud Rate;1200 Baud,2400 Baud,4800 Baud,9600 Baud
30  BUSADDRESS=Bus Address
31  COMMODE=Comm. Mode;ASCII,RTU,OFF
32  COMINDEXUSE=Comm. Index Use;No Features,Auto Increment,Auto Decrement
33  COMINDEX=Comm. Index
34  COMDELAY=Comm. Delay
```

Figuur 14; Language file

Door deze gegevens ook buiten het programma vast te leggen, is de tool ook erg gemakkelijk in verschillende talen aan te bieden. Er hoeft dan slechts een nieuwe versie van de taal-files gemaakt te worden.

6.2.6 Extra file met unieke device code

Bij het opstarten, moet het programma ook weten welke unit description file en default configuratie moeten worden geladen voor een bepaalde unit. Hiervoor is een code bedacht, die uniek is voor een combinatie model + firmware. Deze code wordt uit het device uitgelezen, vergeleken met een lijst met codes die het programma kent. Vervolgens wordt aangegeven of de unit bekend is of niet, en wordt vanuit dit file het juiste unit description file en default configuratie aangewezen.

```
1  E110-P;unitE110;defaultE110
2  E110;unitE110;defaultE110
3  E112-P;unitE112;defaultE112
4  E113;unitE113;defaultE113
```

Figuur 15; Opstart file

6.2.7 Default configuratie

Bij het starten van het programma, wordt de GUI geïnitieerd met de waarden uit de default configuratie. Ook is het mogelijk voor de klant een unit hiernaartoe te resetten, door deze configuratie te laden en vervolgens naar de unit te schrijven.


```

29 POWERMAN
30 LCDUPDATE=2
31 BATTMODE=1
32 SIGNAL
33 SIGNAL=1
34 FORMULA=1
35 FILTER=10
36 METERCUT=10

```

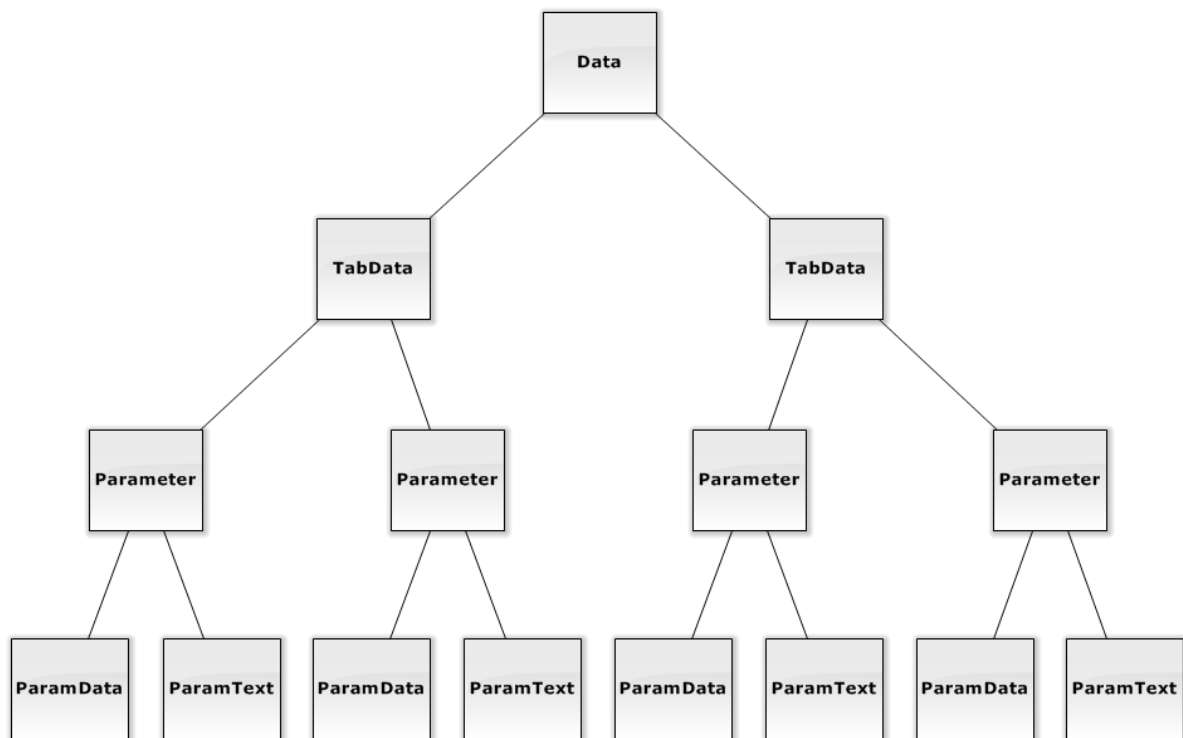
Figuur 16; Configuratie file

Zoals in figuur 16 te zien, staan hierin alle voor een bepaalde unit benodigde parameters gegroepeerd per tabblad (te herkennen aan de ontbrekende '=', bijv. regel 29), opgeslagen als key = value. De configuratie die door de gebruiker opgeslagen wordt, wordt op dezelfde manier opgeslagen als de default configuratie.

6.3 Interne datastructuur

De gebruiker start het programma op en selecteert een unit. Het programma kijkt of er een unit description file voor het geselecteerde type op het systeem aanwezig is. Wanneer dit het geval is, leest het programma de benodigde informatie in.

Vervolgens wordt het Data object gevuld. Er worden TabData objecten aangemaakt, voor de benodigde tabsheets, en vervolgens wordt een Parameter object gecreëerd. Dit bestaat uit een ParamData object, welke alle voor de unit belangrijke informatie (value, memory address) bevat, en een ParamText object, waar alle voor de weergave belangrijke informatie (label, helptekst) is opgeslagen. De hiërarchie van deze objecten is grafisch weergegeven in figuur 17.



Figuur 17; Datastructuur

In bovenstaande figuur is de samenhang van de eerder genoemde objecten duidelijk te zien. Het Data-object zorgt ervoor dat de onderliggende objecten naar behoren gevuld worden. Om de value-

property van het ParamData object te vullen, wordt bij het initialiseren van deze structuur ook direct het file met de default configuratie uitgelezen. Hierdoor is het ook mogelijk de GUI volledig te genereren, zonder dat een apparaat is aangesloten.

6.4 GUI ontwerpen

Er zijn voor alle tabbladen GUI ontwerpen gemaakt, om te zien hoe bepaalde parameters het beste kunnen worden weergegeven, en dus de tabsheet description files beter te kunnen vullen.

In figuur 18 is het ontwerp voor de pagina 'Analog Output' weergegeven.

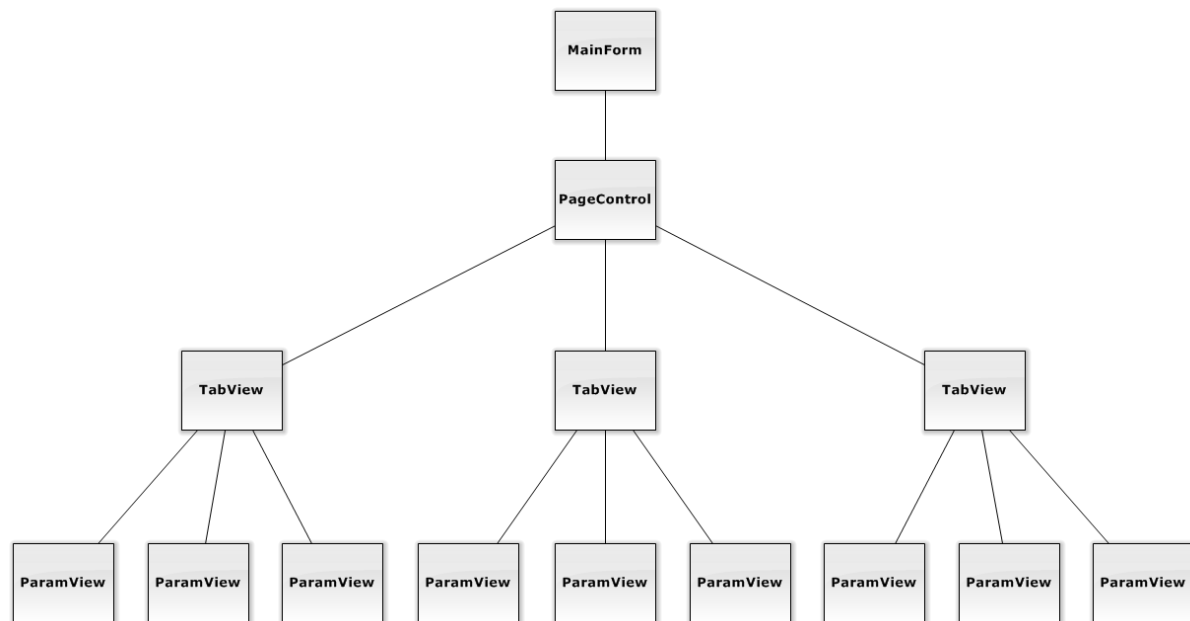
Figuur 18; Ontwerp configuratiepagina 'Analog Output'

6.5 Genereren GUI

In de configuratiefiles kunnen een aantal dingen worden vastgelegd om het genereren van de GUI te vereenvoudigen. In de tabsheet description file wordt per parameter een viewType meegegeven (bijv. label, radio, intinput, etc.). Wanneer dit type niet is toegekend, kan er geen GUI element voor deze parameter gegenereerd worden.

De GUI wordt opgebouwd met een soortgelijke structuur als in figuur 17. Op het Main Form wordt een PageControl geplaatst, een element waaraan Tabsheets kunnen worden gekoppeld.

Er is een klasse TabView, die een uitgebreide implementatie van TTabSheet bevat. Bij het aanroepen van de constructor van deze klasse, wordt een TTabSheet gecreëerd, maar worden ook direct de parameter-componenten aangemaakt en op dit sheet geplaatst. Hier zit indien nodig ook direct het help-icoon in.



Figuur 19; Structuur GUI klassen

Voor het genereren van de GUI is het Data object (§6.3) nodig. Dit object bevat alle benodigde informatie. Als dit object niet gevuld is, is het dus niet mogelijk de TabViews te genereren, en zal dit deel van de GUI leeg blijven.

Om te voorkomen dat het Data object vanwege een communicatiefout (in het geval dat een unit is aangesloten) niet gevuld raakt en hierdoor de GUI niet kan worden gegenereerd, wordt deze met de standaardwaarden van het geselecteerde unittype geïnitieerd. Hierdoor kan het programma altijd goed uitgevoerd worden.

6.6 Modbus driver

Vooralsnog communiceren de displays van Fluidwell alleen via een seriële verbinding. Er zijn daarom twee basis transmissie modes mogelijk: ASCII en Remote Terminal Unit (RTU). De transmissie mode definieert de wijze waarop de Modbus berichten zijn gecodeerd. Met Modbus/ASCII zijn de berichten in een leesbaar ASCII formaat. Het Modbus/RTU formaat gebruikt binaire codering wat het bericht onleesbaar maakt als de communicatie gemonitord wordt. In Bijlage III wordt een uitgebreidere toelichting over het protocol gegeven. [6][7] In tabel 1 worden een aantal verschillen tussen Modbus/ASCII en Modbus/RTU weergegeven.

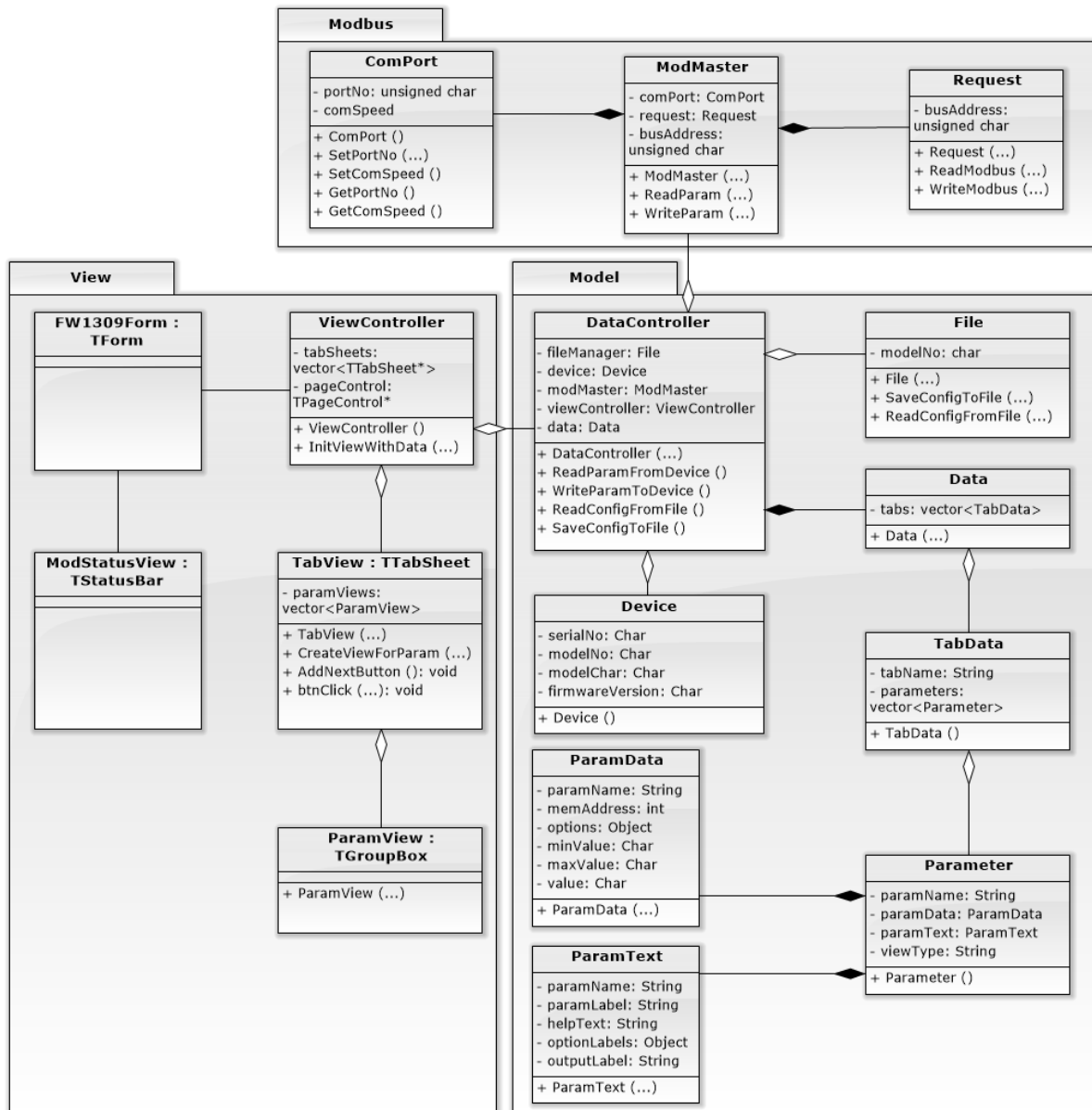
Tabel 1; Verschil Modbus/ASCII en Modbus/RTU

	Modbus/ASCII	Modbus/RTU
Karakters	ASCII 0...9 en A...F	Binair 0...255
Foutcontrole	Longitudinal Redundancy Check (LRC)	Cyclic Redundancy Check (CRC)
Frame start	Karakter ':' (0x3A)	3,5 karakters stilte
Frame eind	Karakters CR/LF (0x0D / 0x0A)	3,5 karakters stilte
Gaten in bericht	1 sec	1,5 karakter
Start bit	1	1
Data bits	7	8

De units van Fluidwell gebruiken niet allemaal dezelfde Modbus modus, en de Modbus driver moet dus beide formaten ondersteunen.

6.7 Overzichtsdiagram klassen

In figuur 20 is een overzichtsversie van het klassendiagram waarmee aan de implementatie is begonnen zichtbaar. De applicatie is grofweg opgedeeld in drie modules: View, Model en Modbus. De Modbus-module is slechts een ontwerp, en wordt nog niet geïmplementeerd.



Figuur 20: Overzichtsdiagram klassen, versie begin realisatie

Wanneer de applicatie gestart wordt, wordt de ViewController geïnitieerd. De ViewController handelt alle zaken die betrekking hebben op de verschillende views af, en verzorgt de communicatie met de Data-module. De DataController houdt een object van Data vast, welke gevuld wordt door de configuratiefiles die de klasse File uitleest. De opbouw van dit diagram komt overeen met de eerder toegelichte hiërarchie.

7. Realisatie

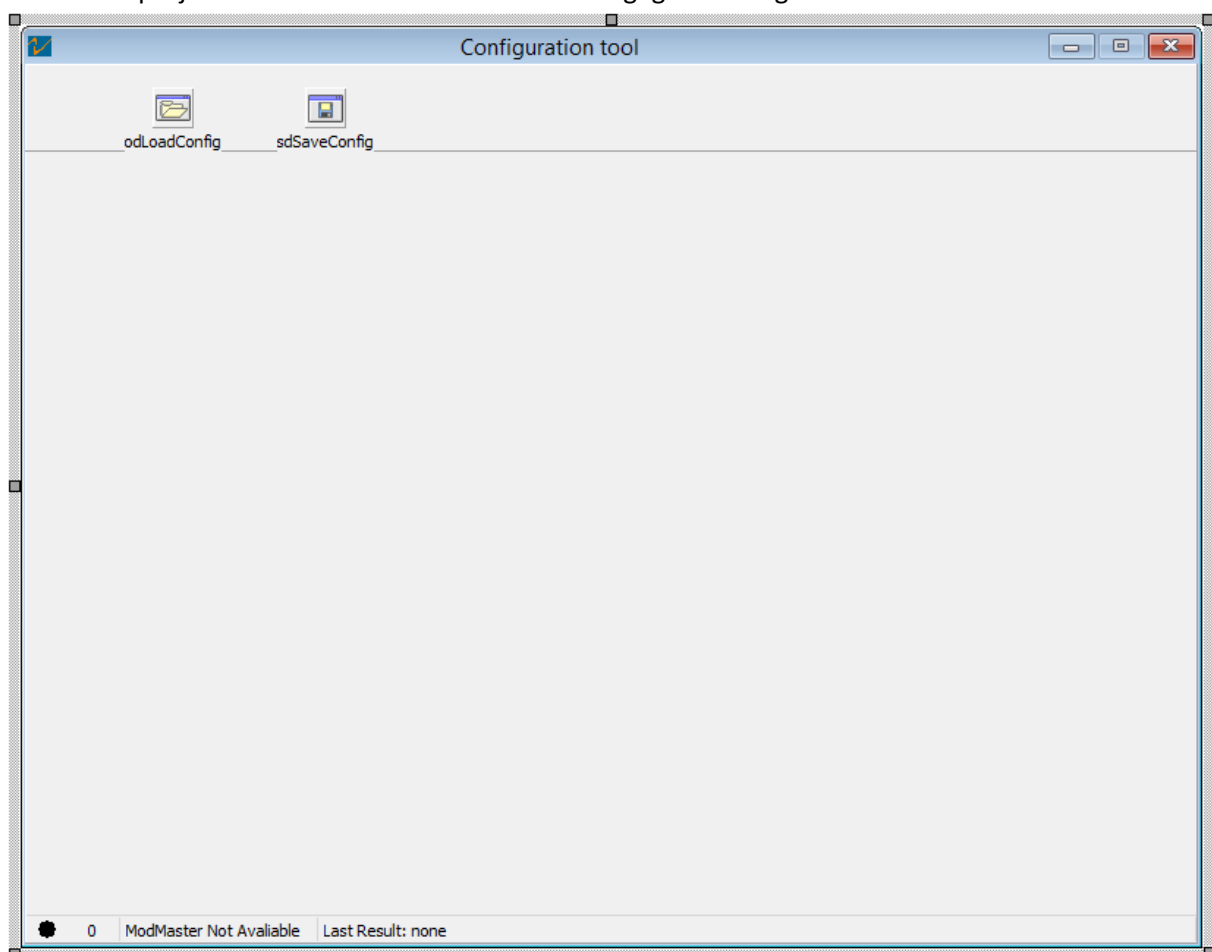
In het volgende hoofdstuk is kort beschreven hoe het ontwerp is geïmplementeerd, en wat de huidige status en vervolgstappen zijn.

7.1 Implementatie ontwerp

Het implementeren van het beschreven ontwerp is vrij complex, vanwege de benodigde flexibiliteit. Het werken met een aantal verschillende configuratiefiles kan problemen opleveren, bijvoorbeeld wanneer een file niet aanwezig is, of wanneer de indeling niet overeenkomt met wat het programma verwacht. Deze mogelijke problemen moeten in de code worden afgevangen. Daarnaast, moet in het geval dat het wél allemaal goed gaat, veel afgehandeld worden met betrekking tot de verwerking van de ingelezen informatie.

Er is begonnen met het genereren van een aantal configuratiefiles, om eenvoudig te kunnen zien of de verschillende onderdelen van de applicatie werken. Vervolgens is begonnen met de implementatie van de FileManager. De FileManager leest de informatie die nodig is om de views te genereren. Na de realisatie van de FileManager, is de Data-structuur geïmplementeerd. Hierbij horen de klassen Data, TabData, Parameter, ParamData en ParamText.

Het hoofdscherm van de applicatie is het default-form, wat gegenereerd wordt bij het aanmaken van een nieuw project in C++Builder. Dit scherm is weergegeven in figuur 21.



Figuur 21; Weergave van de GUI zoals deze er in de UI-builder uit ziet

Op het form zijn slechts een aantal componenten geplaatst. Onderaan is de Modbus statusbalk zichtbaar. Vervolgens is de ViewController gecreëerd, met de bijbehorende View-klassen (TabView en ParamView). Aan de hand van het ViewType dat aan de parameters wordt meegegeven, genereert de ParamView-klasse de juiste weergave. In de View-klassen zijn per component-type (bijvoorbeeld radiobutton, textfield, button) event handlers geschreven. Deze event-handlers zorgen ervoor dat wanneer een waarde door de gebruiker wordt gewijzigd, deze wordt aangepast in de kopie van het Data-object waar de View over beschikt.

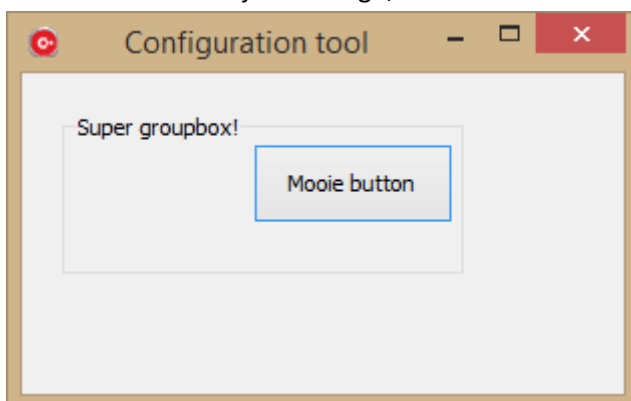
Het is goed mogelijk UI componenten te genereren zonder dat van tevoren vast ligt hoeveel elementen het zijn. Het vergt echter wel iets meer programmeerwerk dan het plaatsen van bijvoorbeeld een button met behulp van een grafische UI-builder. Het stukje code in codefragment 1 genereert de button die te zien is in figuur 22. [1]

```
{
    TGroupBox* gb = new TGroupBox(this);
    gb->Parent = this;
    gb->Height = 80;
    gb->Width = 200;
    gb->Left = 20;
    gb->Top = 20;
    gb->Caption = "Super groupbox!";
    gb->Visible = true;
    gb->Enabled = true;

    TButton* b = new TButton(this);
    b->Parent = gb;
    b->Height = 40;
    b->Width = 100;
    b->Left = 95;
    b->Top = 15;
    b->Caption = "Mooie button";
    b->Visible = true;
    b->Enabled = true;
}
```

Codefragment 1; Genereren GUI-elementen met VCL

Er wordt een GroupBox (gb) gegenereerd. Vervolgens zijn hier allerlei eigenschappen in te stellen (hoogte van gb, breedte, positie, etc.). Hierna wordt TButton (b) definiëerd. Door de Parent-property van b te laten verwijzen naar gb, wordt de button in de groupbox geplaatst (figuur 22).



Figuur 22; Dynamisch gegenereerde GUI (h = 160px; b = 300px)

De button doet nu echter nog niets. Er zal ook een click-event aan gekoppeld moeten worden, en dit moet afgehandeld worden. Normaliter wordt de GUI grafisch ontworpen. Dit houdt in dat een button op het form gesleept kan worden, en hier door bijvoorbeeld te dubbelklikken een click-event aan gekoppeld wordt. In dit geval heeft iedere button, of ieder GUI component, zijn eigen event(-handler). In het geval van een dynamisch gegenereerde GUI is dit niet mogelijk, want op het moment van runnen bestaat het GUI-component nog niet. Ook is niet direct duidelijk hoeveel componenten er gecreëerd zullen worden. Er moet daarom een generieke eventhandler worden geschreven (codefragment 2).

```
{
    TButton* b = new TButton(this);
    b->Parent = this;
    b->Name = "btnNext";
    b->Height = 40;
    b->Width = 100;
    b->Left = 95;
    b->Top = 15;
    b->Caption = "Next";
    b->Visible = true;
    b->Enabled = true;
    b->OnClick = &btnClick; // toekennen actie aan OnClick
}

void btnClick(TObject *Sender)
{
    TButton *btn = dynamic_cast<TButton *>(Sender);
    // welke button is aangeklikt?
    if (btn->Name.SubString(0,7) == "btnNext") {
        // de actie met de button
    }
}
```

Codefragment 2; Voorbeeld afvangen click-event op dynamisch gegenereerde button

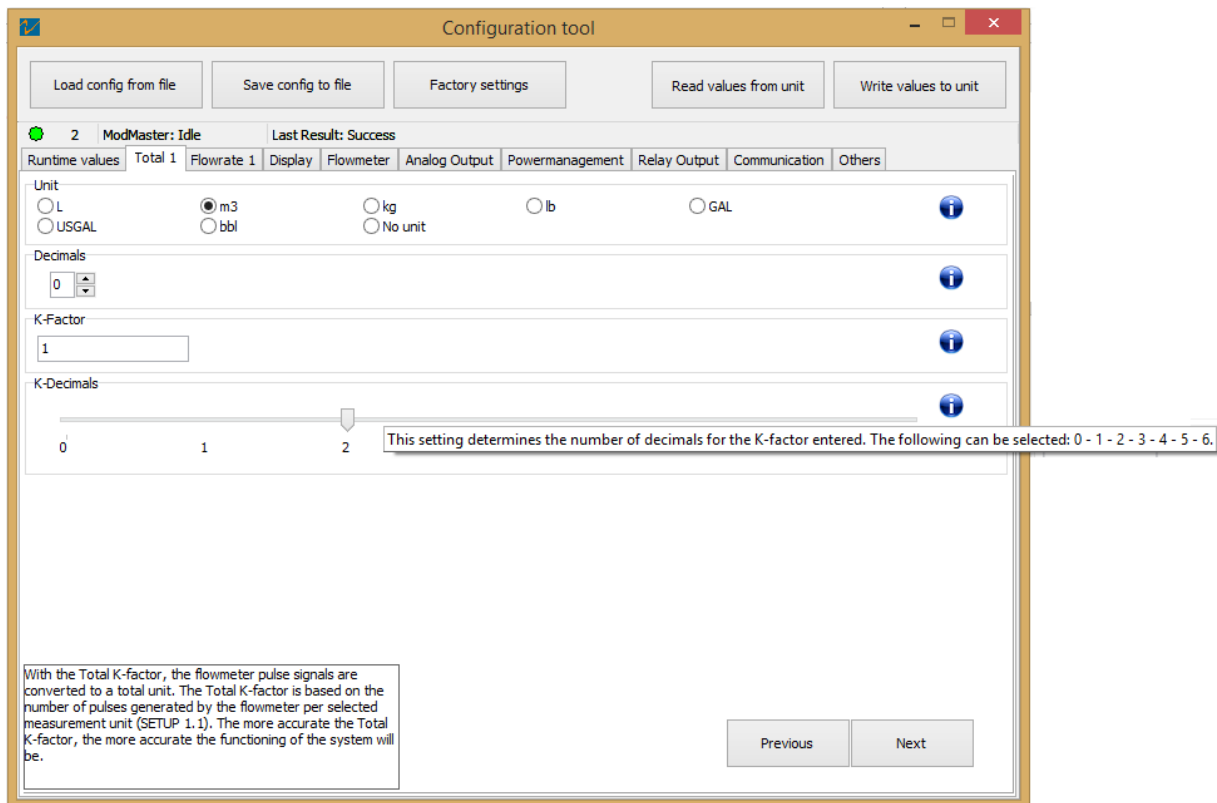
In bovenstaand codefragment is te zien dat in de functie btnClick(TObject *Sender) een button click op een onbekend object wordt afgehandeld. Wanneer op de button geklikt wordt, wordt btnClick() aangeroepen. Door deze functie alleen aan buttons toe te kennen, kan ervan uit worden gegaan dat Sender in dit geval altijd het type TButton is, en kan dit event worden afgehandeld. Het is nodig om voor ieder type component een aparte event handler te schrijven.

Na het implementeren van bovengenoemde onderdelen, is de DataController toegevoegd. Hiermee is de link gelegd tussen de Data-module en de View. Binnen deze klasse zijn ook de functies die nodig zijn voor het opslaan van een configuratie in de unit opgenomen.

Vervolgens is de communicatie met de unit gerealiseerd. Er is tijdens de implementatie gekozen gebruik te maken van de bestaande Modbusdriver, vanwege de tijdsdruk. Eerst is het mogelijk gemaakt de applicatie te vullen met de instellingen zoals ze in de unit zijn, en vervolgens is ook het schrijven van nieuwe waarden mogelijk gemaakt. In figuur 23 is de instelling van de K-factor in de gebruikte test-unit weergegeven. De weergegeven waarde komt overeen met de instellingen in figuur 24.



Figuur 23; Instelling K-factor in test-unit



Figuur 24; Screenshot Configuration tool runtime

In figuur 24 is te zien hoe de help-functie is geïmplementeerd. On mouse-over van het Info-icoon wordt een hint weergegeven. Bij aanklikken van het icoon wordt de help-informatie linksonder getoond. Wanneer de gebruiker geen help-functie wil gebruiken, is deze dus ook niet zichtbaar.

7.2 Status

De functionaliteit die benodigd is voor het opslaan en inladen van configuraties op en van de PC is volledig geïmplementeerd. Ook is de communicatie met de unit (en het uitlezen en schrijven van en naar de unit) operationeel. De applicatie wordt volledig op basis van de configuratiefiles gegenereerd. Hierdoor werkt de tool al met verschillende units (mits de correcte configuratiefiles beschikbaar zijn).

Op dit moment is het product nog niet getest. Ook ontbreken er nog een aantal gewenste functionaliteiten, zoals het door het programma zelf zoeken naar updates voor de filedatabase. De applicatie kan tevens nog niet controleren op afhankelijkheden tussen de parameters, of aan de hand van de gedefinieerde afhankelijkheden bepaalde labels updaten.

Wanneer wordt teruggekeken naar het klassendiagram bij het begin van de realisatie, vallen een aantal zaken op. De klasse 'Device' is bijvoorbeeld komen te vervallen. Er is een klasse LanguageSet toegevoegd, die de controle houdt over alle teksten die binnen de applicatie nodig zijn. Deze is toegevoegd toen de ondersteuning voor meerdere talen in de opdracht is opgenomen. Zie Bijlage II voor het klassendiagram dat overeenkomt met de gerealiseerde applicatie.

De applicatie zal na het eindigen van de afstudeerperiode nog verder worden ontwikkeld, zodat deze kan worden gebruikt door klanten. Voor het zover is zal de applicatie nog getest worden.

8. Conclusie en aanbevelingen

Tijdens de afstudeerperiode is gewerkt aan de realisatie van een dynamische configuratie tool. Fluidwell had een alleskunner voor ogen, een tool waarmee, zonder verdere aanpassingen, alle producten zouden kunnen worden geconfigureerd. Door minder informatie in de code vast te leggen, wordt de tool eenvoudiger onderhoudbaar. Er is dan niet voor iedere aanpassing een programmeur nodig.

Uit het onderzoek toegelicht in hoofdstuk 5, blijkt dat het mogelijk is een dynamische applicatie te ontwikkelen waarmee de producten van Fluidwell kunnen worden geconfigureerd. Er is gebleken dat het volledig dynamisch opbouwen van de applicatie en dus nagenoeg niets in de code vast te leggen, de voorkeur heeft. Dit is te realiseren door het gebruik van een externe database. Uit de onderzochte mogelijkheden komt naar voren dat het mogelijk is een filestructuur te ontwerpen waarmee de applicatie kan worden opgebouwd. Met deze filestructuur is het dan ook mogelijk bestanden te hergebruiken.

Na het uitvoeren van het onderzoek, is een applicatie ontworpen. Er is bij het ontwerpen rekening gehouden met het zo onderhoudsvrij mogelijk maken van de tool. Dit komt tegemoet aan de wens van Fluidwell, om zonder een applicatie update nieuwe producten te kunnen toevoegen.

Er is op basis van dit ontwerp een configuratie tool gerealiseerd, die voldoet aan de eisen van Fluidwell wat betreft veelzijdigheid en onderhoudbaarheid. De tool zelf is in principe onderhoudsvrij, door het gebruik van een database van configuratiefiles. Bij het uitbrengen van een nieuw type unit, moeten slechts een aantal files geupdate worden, maar dit heeft geen invloed op het functioneren van de applicatie.

Verder zijn er nog een aantal openstaande punten. Het is tijdens de looptijd van het project niet gelukt, de afhankelijkheden tussen de verschillende in te stellen parameters te controleren. Ook is het nog niet mogelijk de applicatie te gebruiken om logdata uit een unit weer te geven. Het was niet mogelijk dit volledig te ontwikkelen, omdat de implementatie hiervan in de firmware van de units nog niet is afgerond. De manier waarop de logdata in de units beschikbaar zou worden, werd pas laat in het project vastgelegd.

De Modbus driver is niet geupdate. Er is tijdens het ontwerp wel rekening gehouden met de nieuwe driver (welke ook direct in het ontwerp meegenomen is). Omdat echter besloten werd in eerste instantie de oude driver te gebruiken, is dit ontwerp niet volledig. Er wordt aanbevolen dit alsnog af te ronden en de Modbus driver te updaten. Vervolgens kan de configuratietool dan met de nieuwe Modbus-driver werken.

Aanbevolen wordt een tool te ontwikkelen voor het genereren van de benodigde configuratiefiles, wanneer een nieuw type unit op de markt wordt gebracht. Zo zou wanneer nodig iedereen bij Fluidwell een nieuw setje files moeten kunnen genereren.

Evaluatie

Hieronder evalueer ik, in mijn rol als afstudeerder, kort hoe ik de afgelopen periode ervaren heb. Ik heb eerder een voltijd opleiding gevolgd, en heb wat verschillen gemerkt m.b.t. het afstuderen in deeltijd.

Ik heb tijdens de afstudeerperiode (zoals het hoort) heel veel geleerd. Behalve dat de te ontwikkelen applicatie geschreven moest worden in een programmeertaal waarmee ik nog niet eerder gewerkt had (C++), en die ook best een steile leercurve heeft, was het ook de eerste keer dat ik een grotere applicatie in mijn eentje heb ontworpen, en dus de gedurende de opleiding opgedane kennis heb kunnen toepassen.

Ik ben blij dat ik de uitdaging ben aangegaan, de applicatie in C++ te schrijven. Ik heb het gevoel dat ik een beter begrip heb gekregen van hoe het er 'onder de motorkap' aan toe gaat. Je zit met deze taal minder 'vast' dan met bijvoorbeeld Java, maar dit houdt ook in dat er veel zaken zijn die je zelf in de gaten moet houden, waar dit bij andere talen vaak niet het geval is. Dit betekent wel, dat met name vroeg in de realisatiefase, het programmeren niet altijd even snel ging.

Ook tijdens de ontwerpfase heb ik wat vertraging opgelopen. Deels kwam dat door het gebrek aan ervaring, en deels door de complexiteit van de te ontwikkelen applicatie. Tijdens de implementatie van mijn ontwerp, heb ik veel 'last' gehad van voortschrijdend inzicht. Hoewel het ontwerp in essentie hetzelfde is gebleven, zijn er toch een aantal wijzigingen geweest, omdat bleek dat wat ik had bedacht, bijvoorbeeld toch niet zo handig was (of soms achteraf, juist wel heel handig bleek).

Wel was het, vaker wel dan niet, moeilijk te combineren met mijn dagelijkse werkzaamheden. Mijn normale werkzaamheden bestaan voornamelijk uit ICT dienstverlening (systeembeheer voor een aantal klanten uit het MKB). Dit houdt in, dat mijn werk zeer onvoorspelbaar is, waardoor ik vaak mijn geplande 'projectdagen' heb moeten laten schieten. Hierdoor is af en toe de planning wat in de war geraakt. Na de eerste maanden zijn maatregelen genomen, om ervoor te zorgen dat ik meer tijd aan de opdracht zou kunnen besteden. Er is naar de klanten toe gecommuniceerd dat verzoeken zo min mogelijk bij mij moesten worden neergelegd. Deze maatregel heeft geholpen, met als gevolg dat ik de laatste maanden gelukkig wel ca. 4 dagen per week aan mijn afstuderen heb kunnen besteden.

Het was, vooral de eerste maanden, dus erg moeilijk de benodigde tijd aan de afstudeeropdracht te besteden. Een hierdoor noodgedwongen zeer sterk ontwikkelde 'nieuwe' competentie is Time Management. Het is duidelijk moeilijker een afstudeeropdracht in deeltijd tot een goed einde te brengen (vooral wanneer je nog op je oude vertrouwde werkplek zit), dan een voltijd opdracht waar je bij een onbekend bedrijf begint, met als enige doel het in een aantal maanden afronden van een gegeven opdracht (wanneer ik terugkijk op mijn eerste afstudeerperiode).

Al met al ben ik blij met wat ik bereikt en geleerd heb, al zijn er natuurlijk veel dingen die anders of beter hadden gekund.

Literatuur

- [1] Stroustrup, B. (2013). *The C++ Programming Language, 4th edition*. New Jersey, USA: Pearson Education Inc.
- [2] Lambooi, P. (2011). Korte samenvatting TSP voor Afgestudeerden met een Ontwikkelopdracht. Geraadpleegd op 11 februari 2015, van <https://portal.fhict.nl/IS/afstuderen/Lesmateriaal/Korte%20samenvatting%20TSP%20voor%20Afstudeerders%20met%20een%20Ontwikkelopdracht.doc>
- [3] van Buul, L. (2015). H104 Organisatiestructuur v15.2. Veghel, NL: Fluidwell Kwaliteitssysteem.
- [4] Fluidwell Corporate Website. About Fluidwell – General. Geraadpleegd op 9 april 2015, van <http://www.fluidwell.com/fluidwell-group/about-us/general.html>
- [5] Borland Software Corporation (2002). *Borland® C++Builder™ 6 for Windows®, Developer's Guide*. Scotts Valley, CA, USA: Borland Software Corporation.
- [6] (2012, 26 april). MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. Geraadpleegd op 24 januari, 2015, van http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
- [7] (2006, 20 december). MODBUS over Serial Line Specification and Implementation Guide V1.02. Geraadpleegd op 25 januari, 2015, van http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf

Bijlage I – Project Initiatie Document

Documenthistorie

Revisies

Versie	Status	Datum	Wijzigingen
0.1	Concept	17 december 2014	Geen wijzigingen
0.2	Concept	22 januari 2015	Wijzigingen in de Projectdoelstellingen en Planning
1.0	Definitief	28 januari 2015	Wijzigingen in Managementsamenvatting en Organisatiestructuur

Tabel 2; Overzicht revisies Project Initiatie Document

Distributie

Dit document is verstuurd aan:

Versie	Datum verzending	Naam	Functie
0.1	17 december 2014	J. Roelofsen	Afstudeerbegeleider Fontys
0.1	17 december 2014	I. Meij	Bedrijfsbegeleider
0.2	23 januari 2015	J. Roelofsen	Afstudeerbegeleider Fontys
0.2	23 januari 2015	I. Meij	Bedrijfsbegeleider
1.0	28 januari 2015	J. Roelofsen	Afstudeerbegeleider Fontys
1.0	28 januari 2015	I. Meij	Bedrijfsbegeleider

Tabel 3; Verzendlijst van dit document

Goedkeuring

Dit document behoeft de volgende goedkeuringen:

Versie	Datum goedkeuring	Naam	Functie	Paraaf
1.0		J. Roelofsen	Afstudeerbegeleider Fontys	
1.0		I. Meij	Bedrijfsbegeleider	

Tabel 4; Benodigde goedkeuringen document

Managementsamenvatting

Doel van dit document

Het doel van dit document is om vast te stellen hoe de afstudeeropdracht, het schrijven van een configuratieprogramma voor het configureren van de verschillende apparatuur die Fluidwell B.V. produceert, zal plaatsvinden.

Dit document zal een overzicht geven van de doorlooptijd en kosten van het project, en de betrokken personen/bedrijven.

Aanleiding

Fluidwell B.V. heeft de afgelopen 20 jaar veel verschillende typen displays ontwikkeld (voor de weergave van data verkregen door bijv. flow sensoren). Een nadeel dat aan veel van deze typen kleeft, is dat het apparaat met slechts 3 knoppen bediend moet worden. Hiermee moet dus ook de initiële configuratie worden uitgevoerd. Dit is erg veel werk, zeker wanneer het om meerdere apparaten gaat. De nieuwste product-serie die Fluidwell B.V. op de markt brengt, is de E-Serie. Binnen deze markt zijn al concurrenten actief, die een bedreiging kunnen vormen, omdat zij een configuratieprogramma bij hun apparaat leveren. Daarom is er vanuit Fluidwell B.V. de wens ontstaan een configurator beschikbaar te stellen, welke de klant kan gebruiken om zijn apparaat te configureren en data uit te lezen. Om ook de positie binnen de andere markten te versterken, is het idee dat het programma geschikt is om ieder type display te configureren.

Globale aanpak

De afstudeeropdracht bestaat uit een aantal deelopdrachten. Deze staan hieronder vermeld:

- Vaststellen eisen en requirements Fluidwell B.V.
- Analyseren hoe de benodigde veelzijdigheid van het programma gerealiseerd kan worden
- Ontwikkelen van de belangrijkste onderdelen van het programma

Er zal eerst een programma ontwikkeld worden waarmee log data uit het apparaat kan worden uitgelezen, en geëxporteerd kan worden naar .csv. Dit zal op termijn ook worden geïntegreerd met het configuratieprogramma, hier wordt bij het ontwerp rekening mee gehouden.

Dan zal er een configuratieprogramma ontwikkeld worden, om aan de wens van de klanten tegemoet te komen. Om dit te kunnen realiseren, wordt eerst met de opdrachtgever een lijst met requirements opgesteld. Na het prioriteren van deze lijst wordt een User Requirements Specification (URS) en een System Requirements Specification (SRS) opgesteld, waarin de verschillende Use Cases worden uitgewerkt. Aan de hand van deze Use Cases wordt een globaal Testplan voor de applicatie opgesteld. Dan wordt het applicatie-ontwerp verder uitgewerkt, waarna aan de realisatie kan worden begonnen.

Globale kosten en doorlooptijd

De afstudeeropdracht zal een doorlooptijd hebben van 75 werkdagen, in te vullen binnen een periode van 7 maanden. Het project vangt op 1 december aan, en zal eind juni worden afgerond. In deze maanden zal een vergoeding van 500 euro betaald worden aan Stoop ICT. De totale kosten zullen dan ca. 3500 euro bedragen.

Inhoudsopgave

Documenthistorie.....	1
Revisies	1
Distributie	1
Goedkeuring	1
Managementsamenvatting	2
Doel van dit document	2
Aanleiding	2
Globale aanpak	2
Globale kosten en doorlooptijd	2
1 Inleiding	4
1.1 Doel van dit document	4
2 Achtergrond.....	5
2.1 Context	5
2.2 Aanleiding.....	5
2.3 Huidige situatie.....	5
3 Projectdefinitie	6
3.1 Projectdoelstellingen.....	6
3.2 Gekozen oplossing of aanpak	6
3.3 Producten c.q. eindresultaat	7
3.4 Budget	7
4 Projectorganisatiestructuur	8
4.1 Opdrachtgever.....	8
4.2 Projectmanager	8
4.3 Projectlid	8
4.4 Projectborging	8
4.5 Project Support.....	9
5 Productdecompositiestructuur	9
6 Planning	10

1 Inleiding

1.1 Doel van dit document

Dit document is opgesteld om alle relevante basisinformatie en uitgangspunten van het project vast te leggen om het op de juiste wijze te kunnen uitvoeren. Het heeft tot doel het project te definiëren, als basis te dienen voor het management en de beoordeling van het succes van het project mogelijk te maken.

In dit Project Initiatie Document (PID) komen de volgende aspecten van het project aan bod:

- Wat wil men bereiken met het project?
- Waarom is het belangrijk deze doelstellingen te bereiken?
- Welke betrokkenen zijn er in dit project en wat zijn de rollen en verantwoordelijkheden?
- Hoe en wanneer worden de maatregelen uit dit PID gerealiseerd?

Het document wordt gebruikt:

- Om er zeker van te zijn dat het project een gezonde basis heeft voordat de opdrachtgever gevraagd wordt zich aan het project te verbinden
- Om te dienen als basisdocument op grond waarvan de opdrachtgever en de projectmanager de voortgang en wijzigingen kunnen toetsen en bewaken.

2 Achtergrond

2.1 Context

Dit project richt zich op het onderzoek naar de mogelijkheid één configuratietool te ontwikkelen voor alle Fluidwell units die via het Modbus protocol kunnen communiceren. Deze tool moet het mogelijk maken deze verschillende units zonder problemen te configureren.

2.2 Aanleiding

Fluidwell B.V. heeft de afgelopen 20 jaar veel verschillende typen displays ontwikkeld (voor de weergave van data verkregen door bijv. flow sensoren). Een nadeel dat aan veel van deze typen kleef, is dat het apparaat met slechts drie knoppen bediend moet worden. Hiermee moet dus ook de initiële configuratie worden uitgevoerd. Dit is erg veel werk, zeker wanneer het om meerdere apparaten gaat.

De producten die Fluidwell levert zijn onderverdeeld in 6 productseries, welke zijn verdeeld over 8 functiegroepen. In totaal gaat het om ca. 75 producten, welke in verschillende versies kunnen worden geleverd. Ook zijn er verschillende firmware-versies in omloop, waardoor er ook tussen apparaten van hetzelfde type verschil kan bestaan in wat de benodigde parameters zijn.

De verschillende functiegroepen hebben echter, zoals de naam al zegt, verschillende functies. Dit houdt in, dat parameters die bij het ene apparaat moeten worden ingesteld, in een ander apparaat waarschijnlijk niet eens bestaan.

Fluidwell wil de markt op met een nieuwe serie displays (de E-serie). In deze markt zijn al concurrenten actief. Een aantal van deze leveranciers, hebben om het configureren van hun apparaat te vergemakkelijken, een configuratietool geschreven. Omdat het handmatig/fysiek configureren van de E-serie (vooral bij grotere aantallen), erg veel tijd zou kosten, lijkt het Fluidwell een goed idee hierbij ook een configuratietool te kunnen leveren. Om ook de positie binnen de andere markten te versterken, en aan de wensen van de klanten tegemoet te komen¹, is het idee dat het programma geschikt is om ieder type meetapparaat te configureren.

2.3 Huidige situatie

Het configureren van de units gebeurt met de drie knoppen die op het apparaat aanwezig zijn. Ook is het niet mogelijk snel meerdere units van dezelfde configuratie te voorzien, maar moeten alle units apart geconfigureerd worden.

¹ Bron: Klanttevredenheidsonderzoek 2013

3 Projectdefinitie

3.1 Projectdoelstellingen

Het is voor klanten onnodig moeilijk de meetapparatuur te configureren, en de data hieruit uit te lezen.

Er moet een programma komen waarmee de klant:

- log data uit kan lezen
- log data kan exporteren naar .csv

Er moet een configuratieprogramma komen waarmee de klant:

- simpel de gewenste configuratie kan instellen
- de ingestelde configuratie kan opslaan
- een bestaande configuratie via een config-file kan uploaden
- log data uit kan lezen en exporteren.

Doelstellingen:

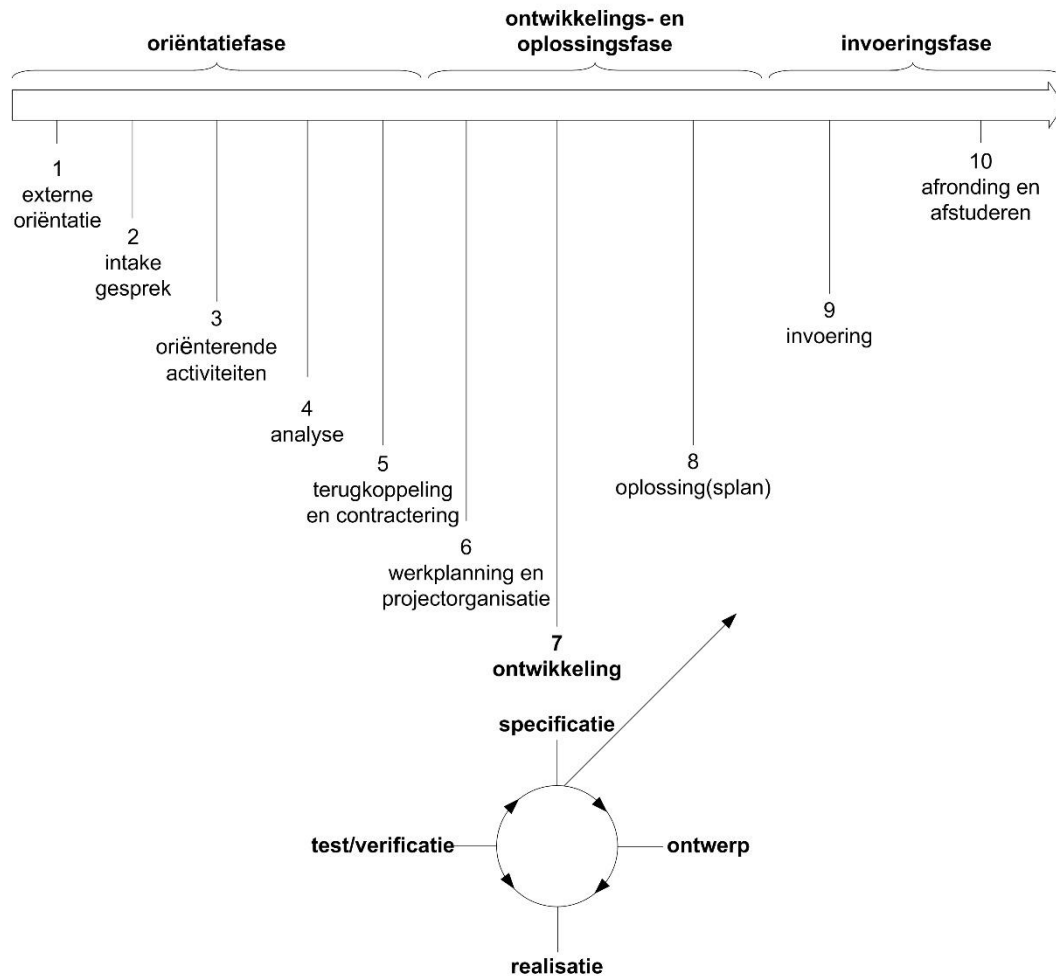
- De GUI/benodigde parameters van het configuratieprogramma worden dynamisch gegenereerd, waardoor het mogelijk is veel verschillende apparaten met één programma te updaten
- Het programma kan zichzelf updaten, en zelf de benodigde config files van het internet plukken
- Er wordt een nieuw format/indeling ontwikkeld voor de config-files

3.2 Gekozen oplossing of aanpak

Er zal een configuratieprogramma komen, welke zowel de mogelijkheid biedt de apparatuur te configureren, als de log data uit te lezen. Omdat op deze wijze de omvang van de opdracht waarschijnlijk te groot wordt om binnen de afstudeerperiode uit te voeren, is besloten het project in een aantal fases op te delen. In fase 1 zal het log data programma worden opgeleverd, in fase 2 het dynamisch genererende configuratieprogramma, geschikt voor (bijna) alle displays, en in fase 3 (na de afstudeerperiode) zullen de resultaten van fase 1 en 2 met elkaar worden geïntegreerd.

Het project wordt aangepakt via het Tien Stappen Plan, met een kleine aanpassing. Omdat het hier een ontwikkelopdracht betreft, zal Stap 7 – Onderzoek, vervangen worden door Stap 7 – Ontwikkeling. Hier zal de realisatie van het project plaatsvinden. Deze aanpassing is nodig, omdat het Tien Stappen Plan is ontworpen voor adviesopdrachten. Een grafische weergave van dit aangepaste Tien Stappen Plan is te zien in Figuur 1.

Tien Stappen Plan^{*)} Afstudeerstage voor (Technisch) Informatici



^{*)} vrij naar "Competent Afstuderen en Stagelopen", Kempen en De Keizer, Noordhoff 2006

Figuur 25; Grafische weergave van het Tien Stappen Plan (voor Software Ontwikkelaars)

3.3 Producten c.q. eindresultaat

Deelproducten (Documentatie):

- PID
- Plan van aanpak
- Software specificatie
- Testplan

Opgeleverd wordt (Eindresultaat):

- Een applicatie waarmee log data kan worden uitgelezen
- De dynamische hoofdapplicatie waarmee configuratiewijzigingen in de meeste door Fluidwell B.V. geleverde displays gemaakt kunnen worden

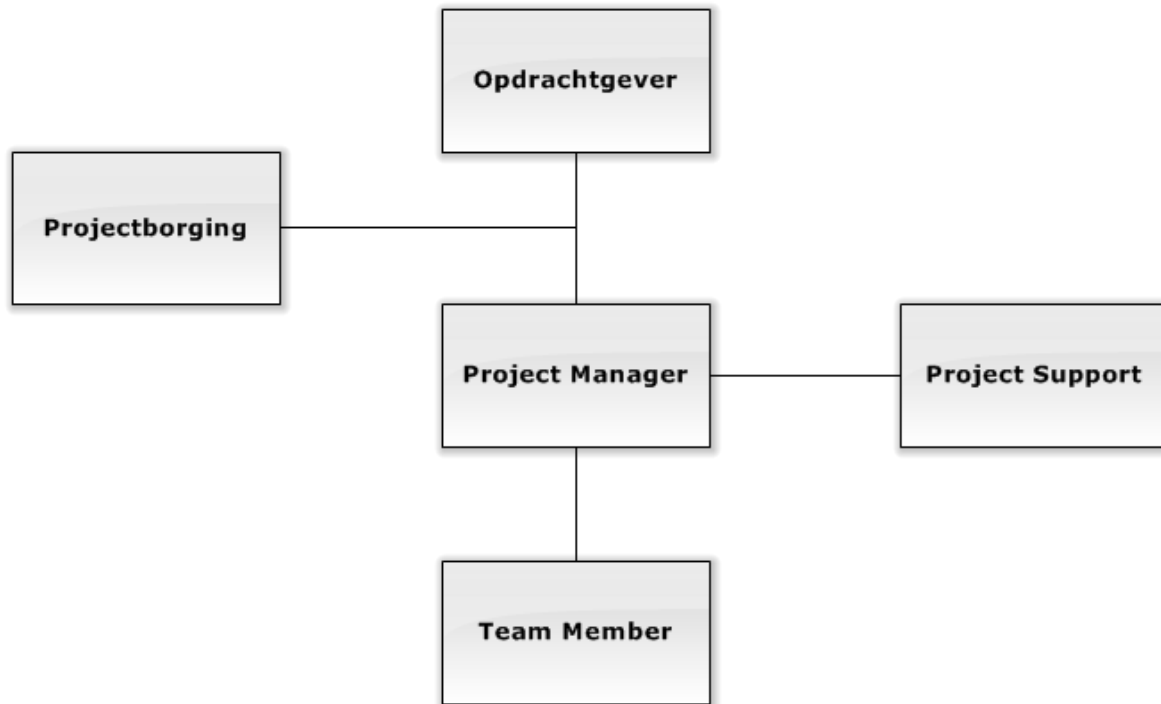
3.4 Budget

Beschikbaar budget	Beschikbare resources	Gewenste opleverdatum	Doorlooptijd
600 uur	N. van der Molen	30 juni 2015	7 maanden

Tabel 5; Beschikbaar budget voor het project

4 Projectorganisatiestructuur

In onderstaande figuur (figuur 2) is een overzicht gegeven van de projectorganisatiestructuur zoals deze geldt voor dit afstudeerproject. Vervolgens wordt per element beschreven wat de taken en verantwoordelijkheden zijn en door wie deze taken worden ingevuld.



Figuur 26; Projectorganisatiestructuur

4.1 Opdrachtgever

Fluidwell B.V., vertegenwoordigd door dhr. E. Rath.

E-mail: e.rath@fluidwell.com

Verantwoordelijkheid: De aansluiting van het project bij de bedrijfsdoelstellingen

Taken: Goed-of afkeuren van (tussentijds) resultaat

4.2 Projectmanager

Manager Techniek, dhr. I. Meij.

E-mail: i.meij@fluidwell.com

Verantwoordelijkheid: Het succesvol realiseren van het project

Taken: Bewaken van de voortgang en kwaliteit

4.3 Projectlid

Afstudeerder, mw. N. van der Molen.

E-mail: n.vandermolen@student.fontys.nl

Verantwoordelijkheid: Producten conform de planning opleveren

Taken: Uitvoering van het project, conform de wensen van de opdrachtgever

4.4 Projectborging

Wisselwerking tussen Project Manager en Opdrachtgever, in dit project is hier geen aparte persoon/groep voor aangewezen.

Verantwoordelijkheid en taken: Bewaken van de business case, controleren van de gebruikersbehoeften.

4.5 Project Support

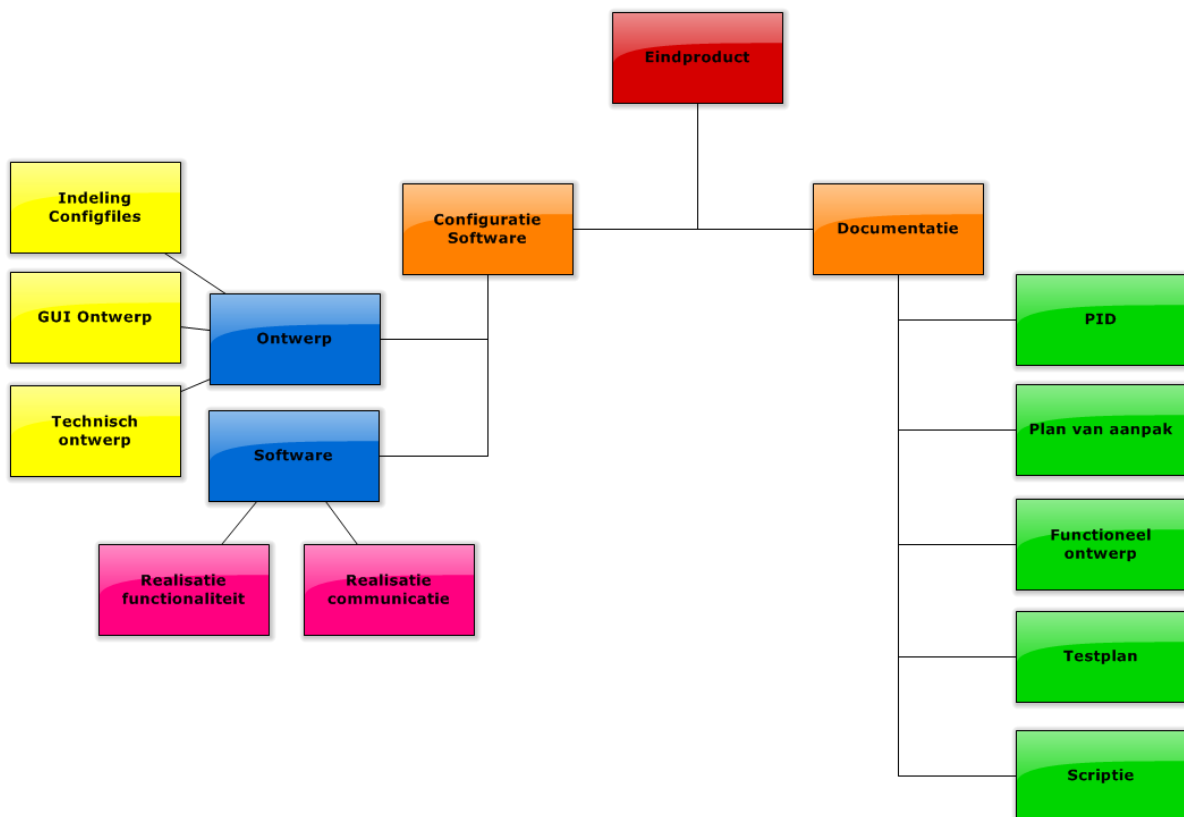
Fontys Hogescholen, dhr. J. Roelofsen.

E-mail: j.roelofsen@fontys.nl

Verantwoordelijkheid en taken: Bewaken van de voortgang, bewaken van de kwaliteit

5 Productdecompositiestructuur

In onderstaande figuur (figuur 2) is de Productdecompositiestructuur te zien.



Figuur 27; Productdecompositiestructuur

De onderzoekscomponent is in de bovenstaande figuur niet specifiek opgenomen. In de eerste weken van het project zal vooral worden onderzocht of het dynamische aspect dat de opdrachtgever voor ogen heeft, ook te realiseren is. Het resultaat hiervan zal in de ontwerpdocumenten terug te zien zijn, en in de scriptie worden behandeld.

6 Planning

Een overzicht van de opleveringsplanning. In de tabel staat aangegeven wanneer welk product als concept, en in de definitieve versie moet worden opgeleverd.

Product	Concept	Definitief
PID	21-01-2015	28-01-2015
Plan van aanpak	22-01-2015	29-01-2015
GUI ontwerp	16-02-2015	09-03-2015
Functioneel ontwerp	25-03-2015	05-04-2015
Indeling Configfiles	29-03-2015	05-04-2015
Technisch ontwerp	12-04-2015	16-04-2015
Testplan	14-04-2015	21-04-2015
Software	29-05-2015	16-06-2015
Afstudeerverslag	29-05-2015	16-06-2015

Bijlage II – Ontwerpdocument

Inhoud

1. Inleiding	2
1.1 Achtergrond.....	2
1.2 Ontwerp doelen	2
2. Architectuur.....	3
2.1 Introductie	3
2.2 Data	4
2.3 Communicatie	6
2.4 Modules.....	6
2.4.1 Module Data.....	6
2.4.2 Module View	7
3. Operation	8
3.1 Gebruikers	8
3.2 Scenarios / use cases.....	9
UC1: Software installeren/de-installeren.....	9
UC2: Apparaat (laten) detecteren	10
UC3: ‘Verbindingsinstellingen’ wijzigen	10
UC4: Configuratiebestand inladen	10
UC5: Configuratie instellingen aanpassen.....	11
UC6: Configuratie opslaan.....	11
3.3 Installatie	12
3.4 Licensing	12
3.5 Upgrades	12
3.6 Uninstall.....	12
4. Bijlage	13
4.1 Sequentiediagram	13
4.2 Activiteitendiagram	13

1. Inleiding

1.1 Achtergrond

Het beschreven systeem is een configuratieprogramma voor de configuratie van de displays van Fluidwell. Het zal in staat zijn alle verschillende typen te configureren, mits het apparaat via het Modbus protocol kan communiceren.

Het systeem draait binnen Microsoft Windows (XP, Vista, 7, 8, 10).

In het verleden heeft Fluidwell een ander configuratieprogramma meegeleverd, deze was echter alleen geschikt voor een specifiek type apparaat. De kracht van dit programma, zal de veelzijdigheid zijn.

1.2 Ontwerp doelen

In de onderstaande tabel (tabel 1) is een overzicht gegeven van de vastgestelde functionele requirements, op basis waarvan de applicatie ontwikkeld zal worden.

Tabel 6; Overzicht functionele requirements

Functionele requirements		M	S	C	W
F01	Het systeem moet (minimaal) werken op Windows XP, Vista, 7 en 8.	X			
F02	Het systeem moet een aangesloten apparaat kunnen detecteren.		X		
F03	Het systeem moet via Modbus met het apparaat communiceren.	X			
F04	Het systeem moet de software-versie uit het apparaat kunnen lezen.	X			
F05	Het systeem moet het apparaat-type uit het apparaat kunnen lezen.	X			
F06	Het systeem moet de benodigde instellingen uit een .udf file kunnen lezen.	X			
F07	Het systeem moet verschillende typen apparaten kunnen configureren.	X			
F08	Het systeem moet de configuratie in het apparaat kunnen wijzigen.	X			
F09	Het systeem moet de configuratie uit het apparaat kunnen lezen.	X			
F10	Het systeem moet een (gewijzigde) configuratie op de pc kunnen opslaan.	X			
F11	Het systeem moet een bestaande configuratie vanaf de pc kunnen openen.	X			
F12	Het systeem moet online controleren op updates.			X	
F13	Het systeem moet logdata kunnen lezen uit het apparaat.	X			
F14	Het systeem moet de uitgelezen logdata weer kunnen geven.		X		
F15	Het systeem moet logdata kunnen exporteren (als .csv).	X			
F16	Het systeem moet met een installatiewizard geïnstalleerd kunnen worden.			X	
F17	Het systeem beschikt over een configuratie-wizard.				X
F18	Het systeem kan per PC maar 1 keer gestart zijn.		X		
F19	Het systeem moet de GUI dynamisch genereren	X			

2. Architectuur

2.1 Introductie

De applicatie die ontwikkeld gaat worden is stand-alone. Er zal altijd maar 1 instantie van de applicatie gelijktijdig worden uitgevoerd.

Het systeem draait binnen Microsoft Windows (XP, Vista, 7, 8, 10).

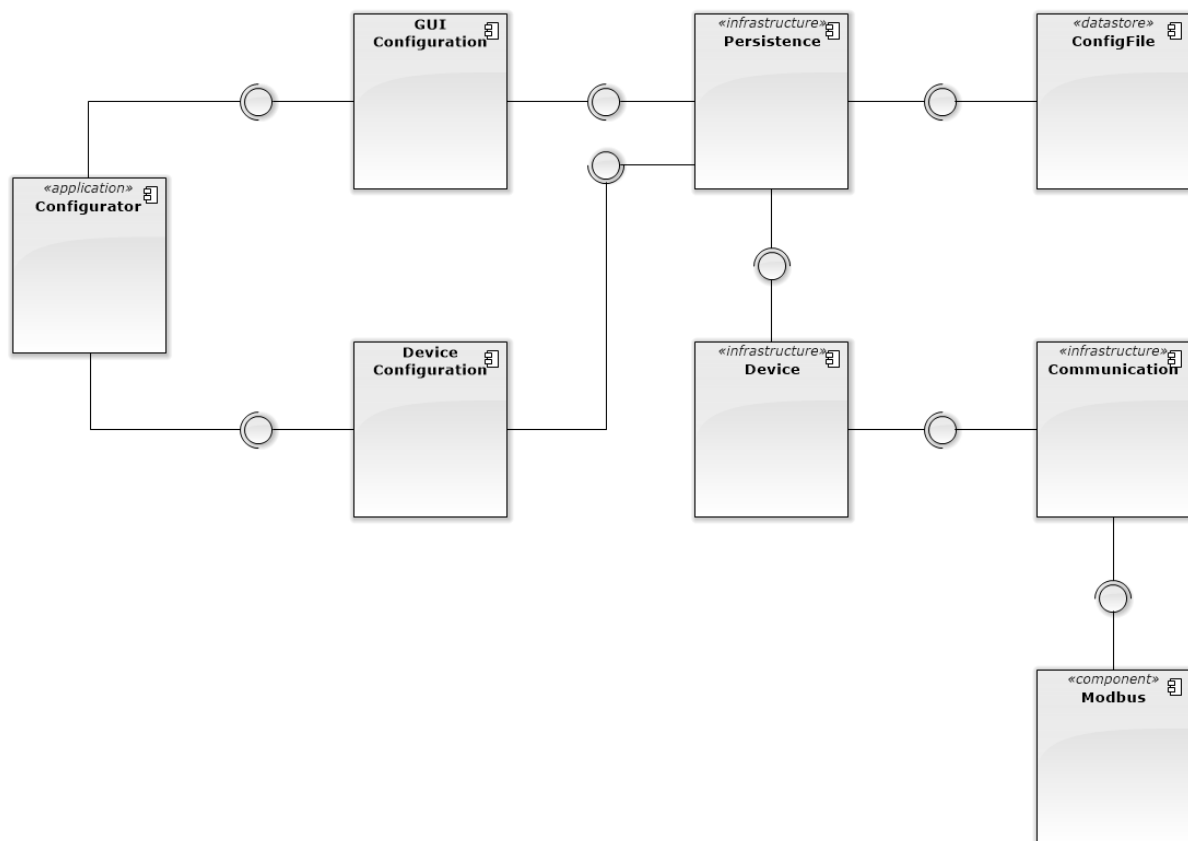
Major inputs en outputs

Om het programma te kunnen starten zijn een aantal configuratie files nodig. Het programma weet namelijk zelf niet hoe het eruit moet zien. Afhankelijk van de geselecteerde unit, wordt een bepaalde set files ingelezen. Op basis van deze files wordt dan de User Interface gegenereerd.

Major outputs zijn een configuratie opgeslagen op de PC, voor later gebruik; en de output naar het te configureren device.

Componenten

De componenten van het programma draaien allemaal op dezelfde PC. Communicatie met het apparaat verloopt via een seriële verbinding, volgens het Modbus protocol.

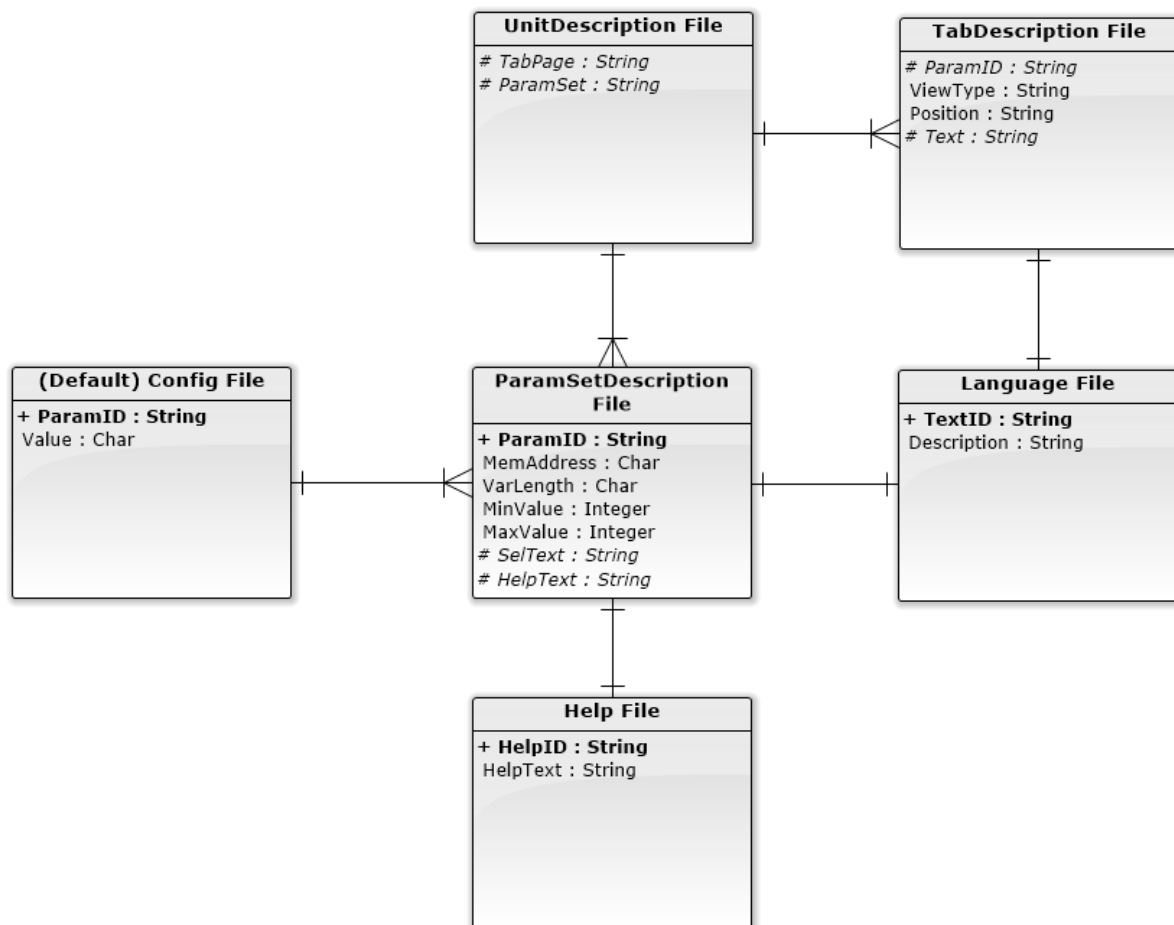


Figuur 28; Componentendiagram

2.2 Data

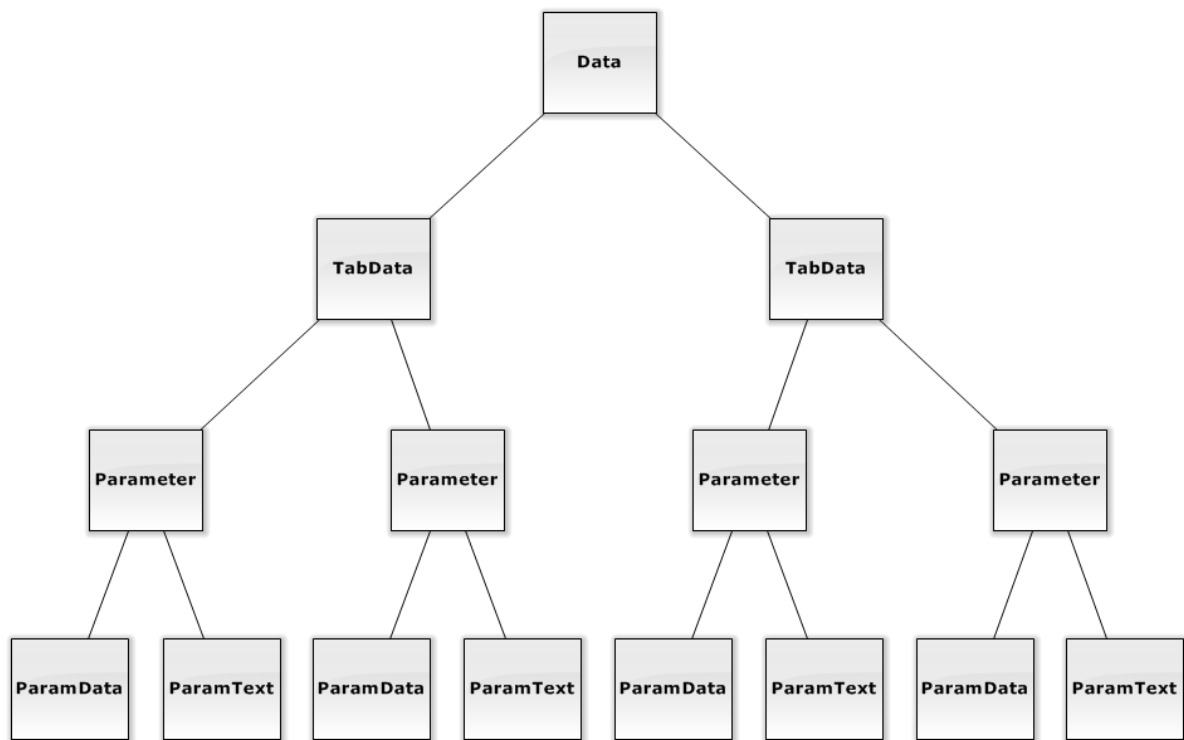
Het systeem wordt opgebouwd op basis van de informatie in een aantal configuratiefiles. Er is gekozen dit niet in een echte database te plaatsen, om het programma zo lichtgewicht mogelijk te houden. Door het systeem opgeslagen configuraties komen in een soortgelijk file terecht. Zonder de eerder genoemde configuratiefiles, kan het systeem niet overweg met een opgeslagen configuratie.

In figuur 2 is een grafisch overzicht gegeven van de verschillende configuratiefiles die nodig zijn om een unit te configureren.

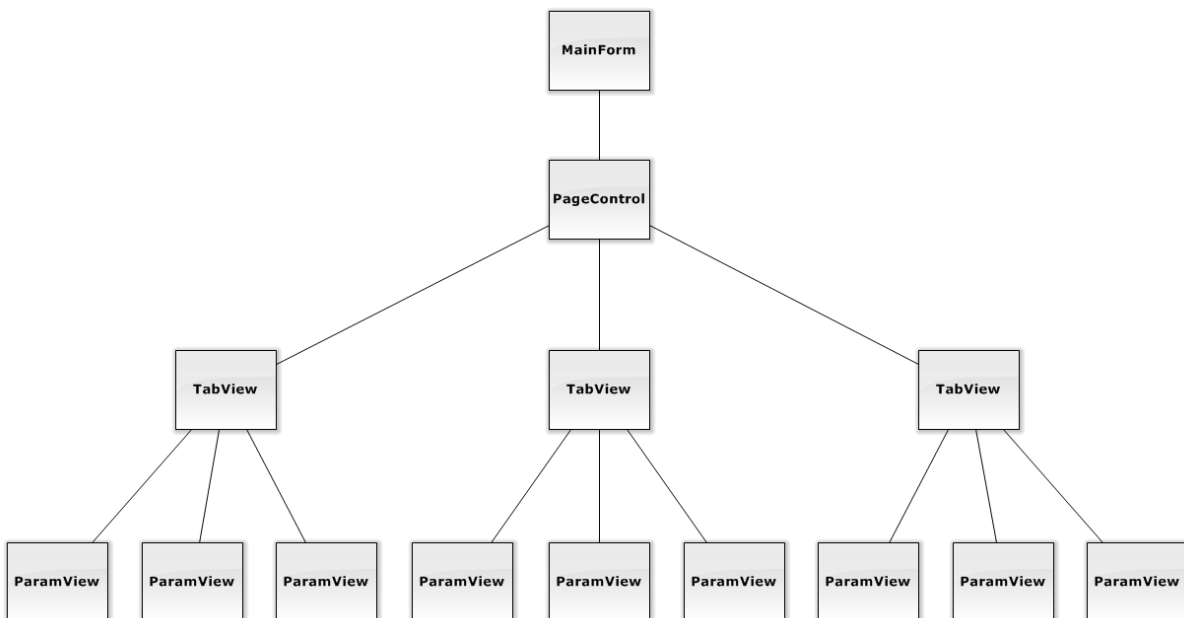


Figuur 29; Structuur configuratiefiles/database

Wanneer deze files door de applicatie worden ingelezen, worden ze geplaatst in een datastructuur zoals weergegeven in figuur 3.

*Figuur 30; Datastructuur applicatie*

Dit Data-object wordt vervolgens gebruikt om de View te genereren. De View wordt opgebouwd volgens de hiërarchie in figuur 4.

*Figuur 31; Datastructuur GUI*

De UI's zullen op de Visual Component Library (VCL) gebaseerd zijn. Dit zijn UI componenten die zeer veel op Windows componenten lijken.

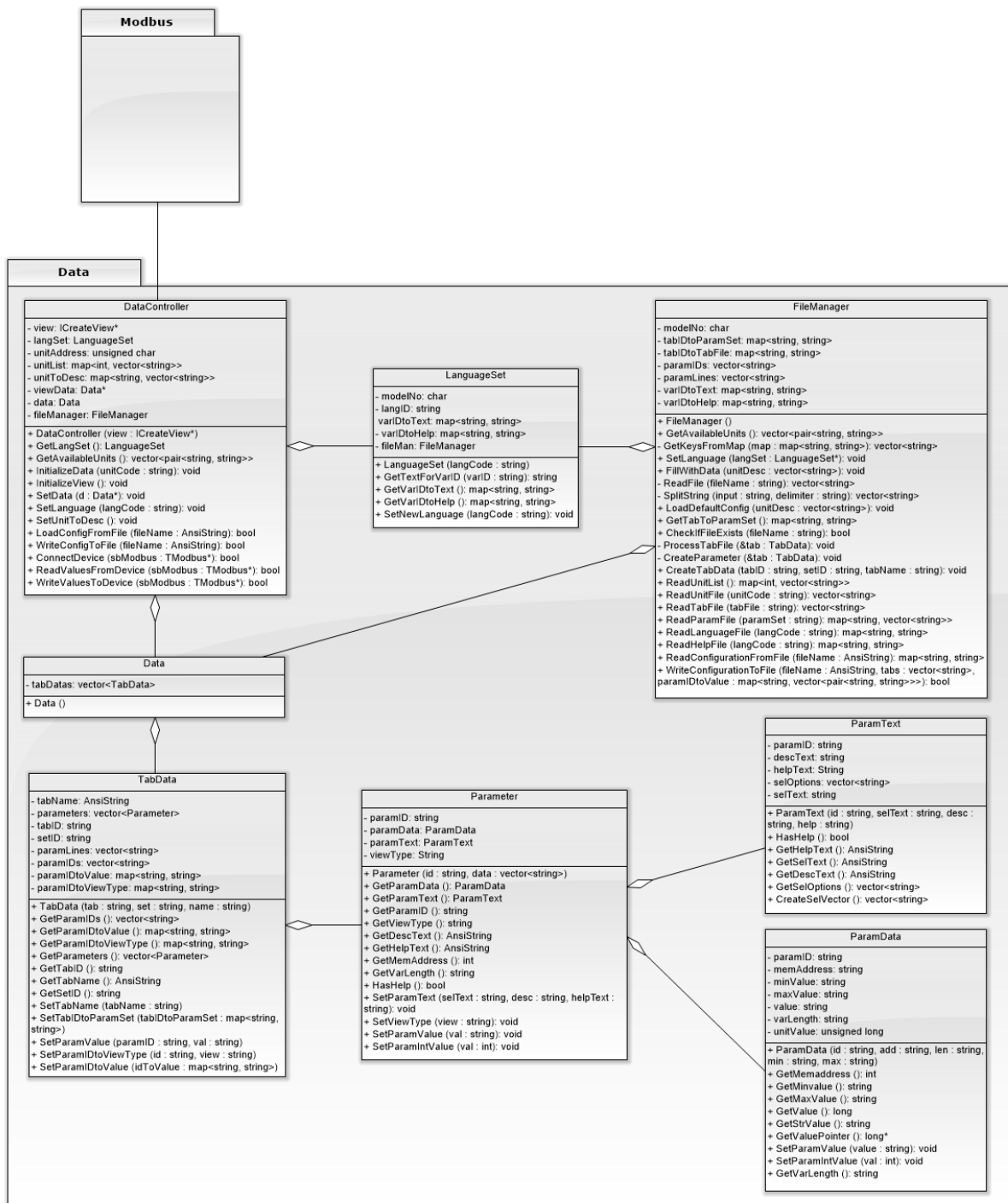
2.3 Communicatie

Het programma communiceert met de unit via het Modbus protocol. Dit protocol kan ingezet worden voor seriële communicatie, en TCP/IP communicatie. In dit geval verloopt de communicatie via een seriële verbinding.

Het programma verbindt met de unit, en stuurt requests (master). Wanneer nodig, geeft de unit (slave) antwoord.

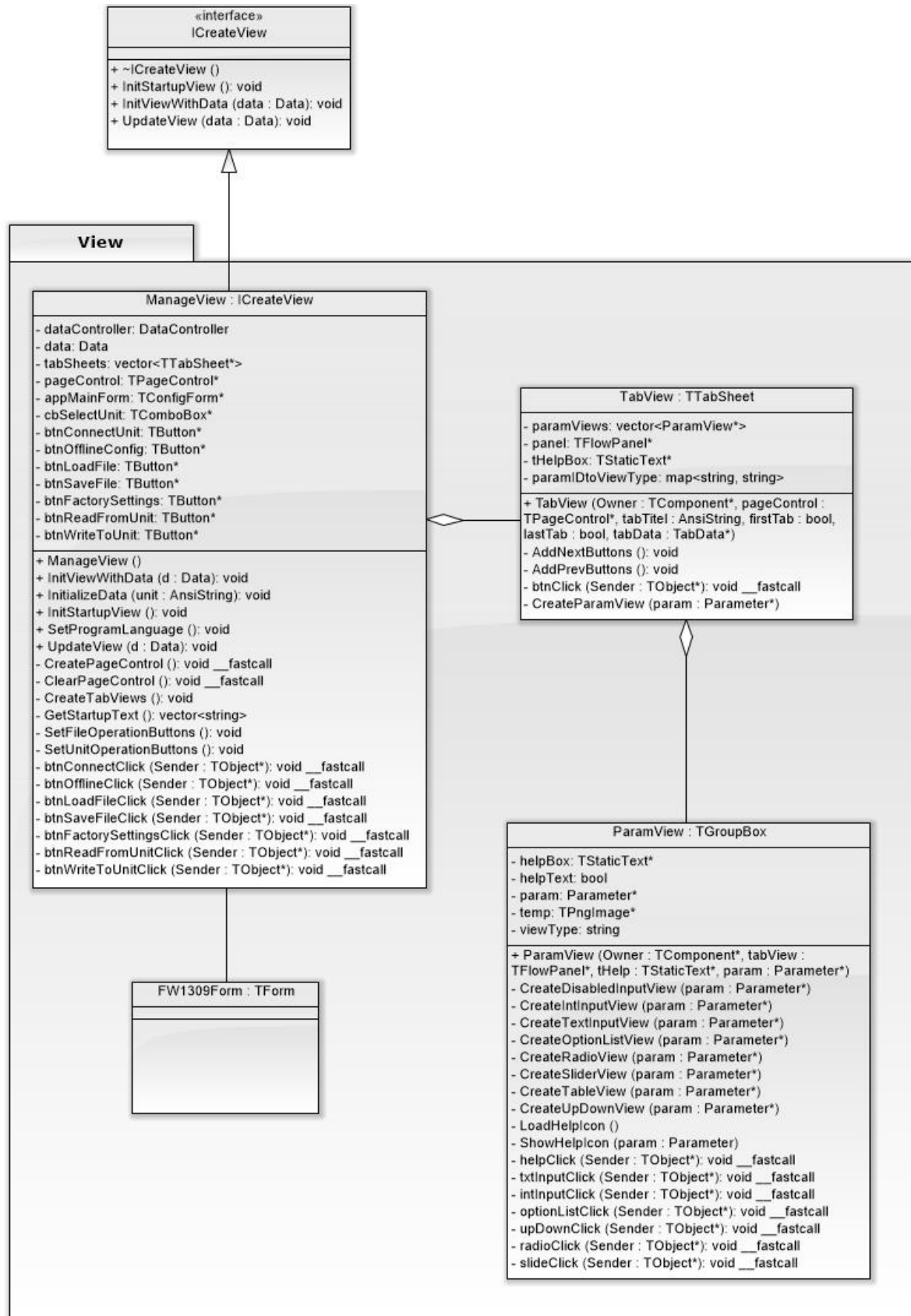
2.4 Modules

2.4.1 Module Data



Figuur 32; Klassendiagram module Data

2.4.2 Module View



Figuur 33; Klassendiagram module View

3. Operation

3.1 Gebruikers

Er is slechts één type gebruiker. Er bestaat geen administrator rol. Op termijn is het mogelijk het programma te beveiligen met een pincode. Gebruikers die niet over deze code beschikken, kunnen dan alleen de instellingen zien, maar niet wijzigen.

De belangrijkste eindgebruikers van het nieuwe systeem zijn:

- De klanten van Fluidwell B.V.

Kennis en achtergrond

De klanten zijn veelal mensen in het bedrijfsleven, waarschijnlijk met een technische achtergrond. Er zal een redelijk groot leeftijdsverschil zitten tussen de verschillende klanten, en kennis en achtergrond zijn moeilijk in kaart te brengen.

De kennis van IT binnen de groep klanten zal variëren van zeer uitgebreid tot nauwelijks, afhankelijk van leeftijd en achtergrond. Een ruime kennis van IT zal echter niet nodig zijn om met deze software overweg te kunnen.

Gebruikersomgeving

De gebruikers hebben meerdere taken waarvan zij een klein deel uit gaan voeren via het systeem. De gebruikers werken daarnaast met de standaard Office applicaties, internet browsers en voor die organisatie specifieke applicaties. Deze zouden niet van invloed moeten zijn.

3.2 Scenarios / use cases

Een aantal scenarios over hoe het systeem werkt. Wat gebeurt er als een gebruiker inlogt, en paar commando's geeft en weer uitlogt. Welke data flow is er, wat veroorzaakt welke actions, enz.

UC1: Software installeren/de-installeren

Naam	UC1: Software installeren/de-installeren
Samenvatting	De gebruiker installeert de software en kan deze uitvoeren (1). De gebruiker de-installeert de software en kan deze niet langer gebruiken (2).
Actoren	Gebruiker
Aannamen	De gebruiker heeft het installatiebestand gedownload.
Beschrijving	<p>(1)</p> <ol style="list-style-type: none"> 1. De gebruiker start het installatiebestand 2. De installatiewizard wordt gestart 3. De gebruiker kiest 'installeren' 4. Het systeem controleert of de software niet reeds geïnstalleerd is 5. De gebruiker doorloopt de installatieprocedure 6. Het systeem laat weten dat de installatie geslaagd is 7. Het programma wordt gestart <p>(2)</p> <ol style="list-style-type: none"> 1. De gebruiker start het installatiebestand 2. De installatiewizard wordt gestart 3. De gebruiker kiest 'verwijderen' 4. Het systeem controleert of de software geïnstalleerd is 5. De gebruiker doorloopt de de-installatieprocedure 6. Het systeem laat weten dat het programma verwijderd is
Uitzonderingen	<p>(1)</p> <ol style="list-style-type: none"> 3. De software is reeds op het systeem geïnstalleerd. De installatiewizard geeft een melding. <p>(2)</p> <ol style="list-style-type: none"> 3. De software is niet op het systeem geïnstalleerd. De installatiewizard geeft een melding.
Resultaat	<p>(1) De software is op de PC geïnstalleerd.</p> <p>(2) De software is van de PC verwijderd.</p>

UC2: Apparaat (laten) detecteren

Naam	UC2: Apparaat (laten) detecteren
Samenvatting	Een aangesloten apparaat wordt door het programma gedetecteerd.
Actoren	Gebruiker
Aannamen	De gebruiker heeft de configuratie software geïnstalleerd. De gebruiker heeft het apparaat aangesloten.
Beschrijving	(1) 1. De gebruiker start het programma. 2. Het programma detecteert het apparaat automatisch. (2) 1. De gebruiker start het programma. 2. Het programma detecteert het apparaat niet automatisch. 3. De gebruiker kiest de optie 'detecteren' 4. De gebruiker geeft het nummer van de COM-poort waaraan het apparaat is aangesloten 5. Het programma kan het apparaat vinden.
Uitzonderingen	(1) 2. Het programma detecteert niet automatisch. Vervolgt bij (2)3. (2) 5. Het programma kan het apparaat nog steeds niet vinden.
Resultaat	Het aangesloten apparaat is gedetecteerd.

UC3: 'Verbindingsinstellingen' wijzigen

Naam	UC3: 'Verbindingsinstellingen' wijzigen
Samenvatting	De gebruiker kan o.a. de COM-poort en snelheid waarmee verbonden moet worden wijzigen.
Actoren	Gebruiker
Aannamen	De gebruiker heeft het apparaat aangesloten. De gebruiker heeft het programma gestart.
Beschrijving	1. De gebruiker selecteert 'Verbindingsinstellingen'. 2. De gebruiker wijzigt de instellingen. 3. Het systeem slaat de instellingen op. 4. Het systeem probeert met de nieuwe instellingen het apparaat te detecteren.
Uitzonderingen	3. Het programma kan de instellingen niet wijzigen. 4. Het systeem kan het apparaat niet detecteren.
Resultaat	De instellingen zijn gewijzigd, en het aangesloten apparaat is gedetecteerd.

UC4: Configuratiebestand inladen

Naam	UC4: Configuratiebestand inladen
Samenvatting	De gebruiker kan een op zijn PC opgeslagen configuratiebestand inladen.
Actoren	Gebruiker
Aannamen	De gebruiker heeft het programma gestart. De gebruiker heeft een configuratiebestand op zijn PC opgeslagen.

Beschrijving	1. De gebruiker selecteert "Een configuratiebestand inladen". 2. Het geselecteerde bestand wordt ingelezen. 3. De ingelezen configuratie wordt weergegeven.
Uitzonderingen	2. Het systeem kan het geselecteerde bestand niet lezen.
Resultaat	De configuratiewaarden zijn in het programma ingevuld.

UC5: Configuratie instellingen aanpassen

Naam	UC5: Configuratie instellingen aanpassen
Samenvatting	De gebruiker kan met behulp van de applicatie de configuratie instellingen aanpassen.
Actoren	Gebruiker
Aannamen	De gebruiker heeft het programma gestart. De gebruiker heeft een unit aangesloten.
Beschrijving	1. De gebruiker selecteert de gewenste unit. 2. De user interface wordt gegenereerd. 3. De gebruiker past de weergegeven instellingen aan.
Uitzonderingen	3. De gebruiker probeert de instellingen aan te passen op een manier die door het programma niet wordt toegestaan; een min-waarde hoger dan een max-waarde, of tekst input in een integer veld.
Resultaat	De configuratie is aangepast.

UC6: Configuratie opslaan

Naam	UC6: Configuratie opslaan
Samenvatting	De gebruiker kan met behulp van de applicatie een (aangepaste) configuratie opslaan.
Actoren	Gebruiker
Aannamen	De gebruiker heeft het programma gestart. De gebruiker heeft een configuratie geladen/aangepast.
Beschrijving	(1) 1. De gebruiker selecteert "De configuratie opslaan". 2. De gebruiker selecteert een opslaglocatie. 3. De gebruiker kiest 'Opslaan'. 4. De configuratie wordt opgeslagen. (2) 1. De gebruiker selecteert "Configuratie in apparaat opslaan". 2. De waarden worden naar de unit weggeschreven.
Uitzonderingen	(1) 4. De locatie is niet toegankelijk/het file kan niet gecreeerd worden. (2) 2. Het systeem kan niet met de unit verbinden. 2. De gegevens voor communicatie met het apparaat zijn niet correct.
Resultaat	De configuratie is opgeslagen.

3.3 Installatie

Het programma draait als stand-alone executable. Het hoeft dus niet geïnstalleerd te worden. Hierdoor kan iedereen het programma gebruiken. Het programma zal snel starten, de gebruiker krijgt geen 'installatie-GUI' te zien.

De gebruiker kan het programma van de website van Fluidwell downloaden. Er is geen licentiecode nodig.

3.4 Licensing

Er is geen licentie nodig om het programma te kunnen gebruiken.

3.5 Upgrades

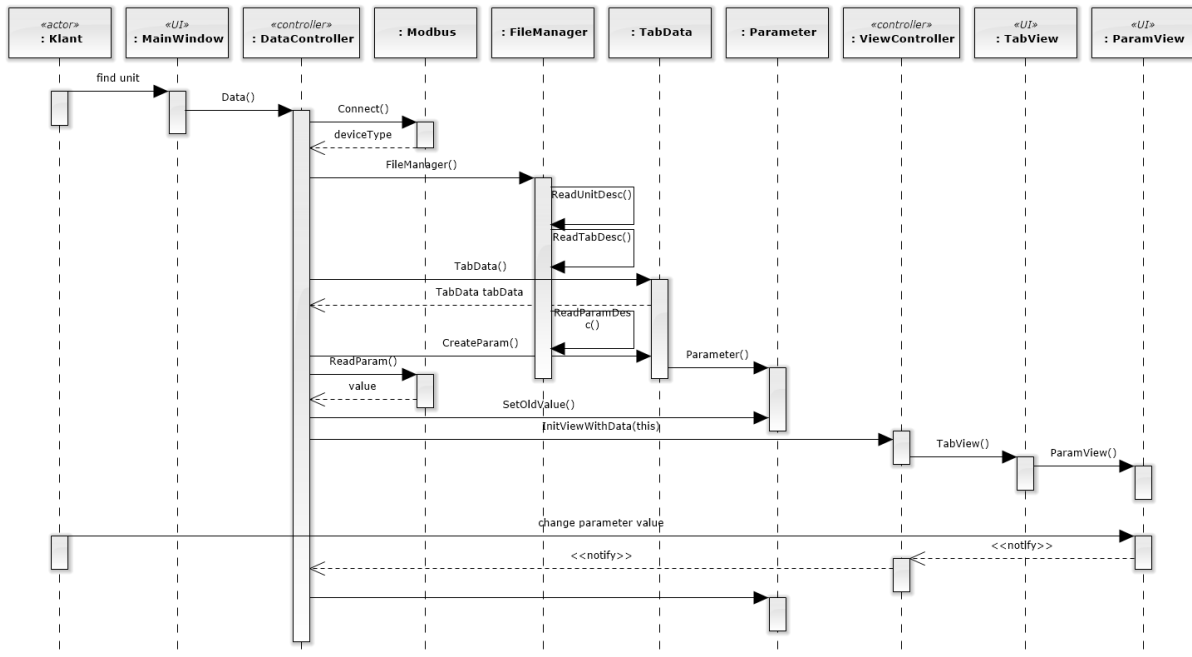
Upgrades worden via de website van Fluidwell gedistribueerd. Over het algemeen zal dit gaan over updates aan de configuratiefiles. Het programma kan deze updates zelf zoeken (mits de PC met het internet verbonden is). Alle gebruikerstypes kunnen de updates downloaden. Omdat geen verdere installatie nodig is, worden hier geen problemen verwacht.

3.6 De-installatie

Omdat het programma niet geïnstalleerd hoeft te worden, hoeft het ook niet gedeïnstalleerd te worden. Om het programma te verwijderen kunnen simpelweg de files worden verwijderd. Wel zal het op termijn mogelijk worden de configuratiefiles te laten verwijderen, als een soort de-installatie.

4. Bijlage

4.1 Sequentiediagram

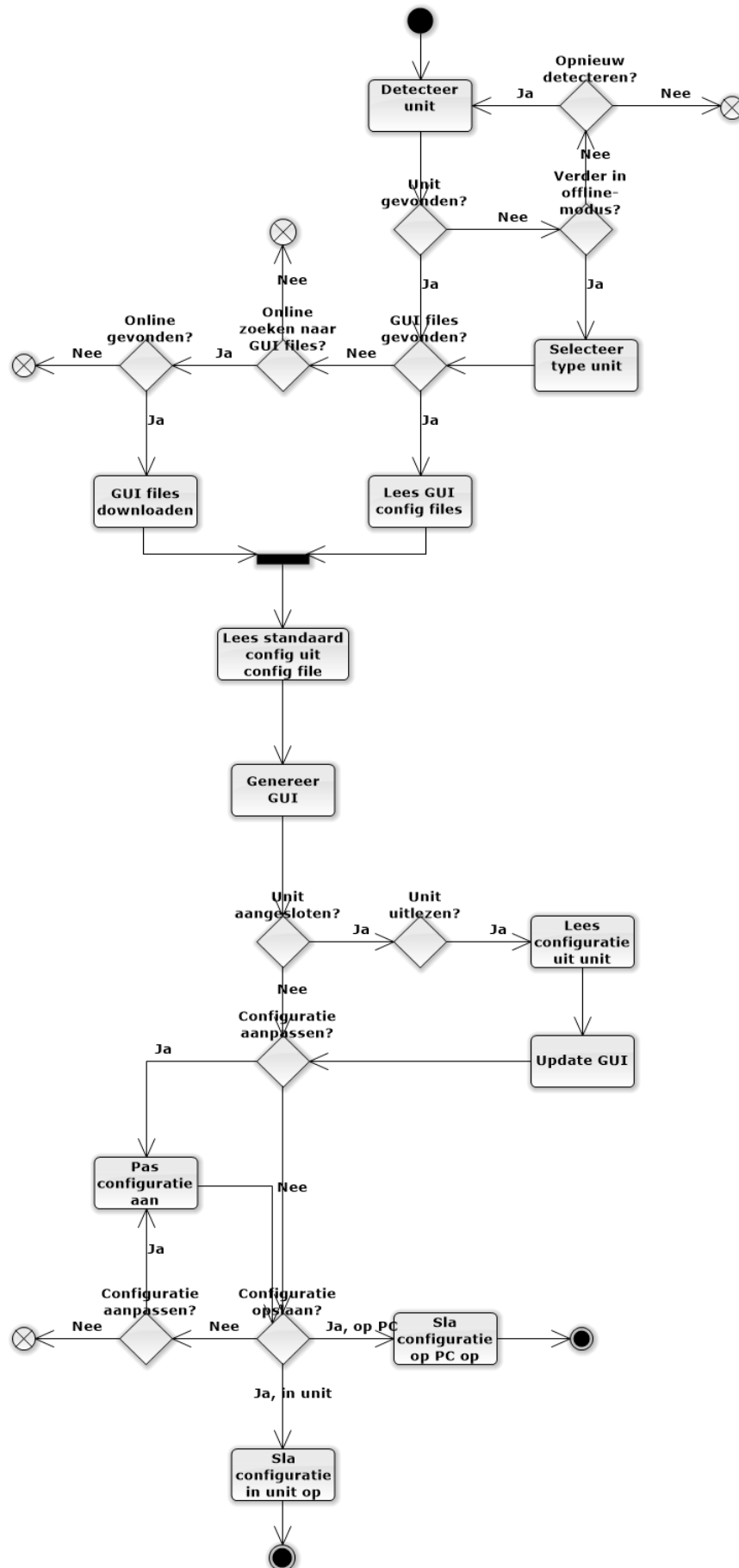


Figuur 34; Opstarten programma, zoals in eerste ontwerp

In bovenstaande figuur is een sequentiediagram te zien van de stappen die door het programma worden doorlopen wanneer het programma wordt opgestart en een unit wordt gedetecteerd.

4.2 Activiteitendiagram

In de figuur op de volgende pagina is een activiteitendiagram te zien waarin op hoog niveau een overzicht wordt gegeven van alle activiteiten die met het programma kunnen worden uitgevoerd. Hiermee is aan het begin van het ontwerp traject vastgelegd op welke momenten het programma welke informatie moet aanbieden/verzamelen.



Figuur 35; Activiteitendiagram

Bijlage III – Modbus

De Modbus communicatie is opgebouwd rond berichten, waarvan het formaat onafhankelijk is van het type fysieke interface dat wordt gebruikt (via een seriële verbinding dezelfde berichten als via een ethernet kabel). In tabel 3 is een overzicht gegeven van de opbouw van zo'n bericht.

Tabel 7; Opbouw Modbus bericht

Veld	Omschrijving
Apparaat adres	Adres van de ontvanger
Functie code	Code die het bericht type definieert
Data	Data blok met extra informatie
Foutcontrole	Numerieke waarde voor foutdetectie

Een conversatie via Modbus wordt altijd gestart door een master. Deze master verzendt een bericht en een slave onderneemt actie en beantwoordt indien nodig het verzoek.

Vooralsnog communiceren de displays van Fluidwell alleen via een seriële verbinding. Er zijn daarom twee basis transmissie modes mogelijk: ASCII en Remote Terminal Unit (RTU). [6]

De transmissie mode definieert de wijze waarop de Modbus berichten zijn gecodeerd. Met Modbus/ASCII zijn de berichten leesbaar, ASCII formaat. Het Modbus/RTU formaat gebruikt binaire codering, waardoor het bericht onleesbaar wordt. Ook verkleint het Modbus/RTU formaat de afmetingen van elk bericht, waardoor de communicatie sneller verloopt. Een apparaat kan maar voor één van deze modes geconfigureerd zijn. De units van Fluidwell gebruiken niet allemaal dezelfde Modbus modus, en de Modbus driver moet dus beide formaten ondersteunen.

Modbus/ASCII

Wanneer Modbus/ASCII gebruikt wordt, worden de berichten gecodeerd in hexadecimale waarden die gerepresenteerd worden met leesbare ASCII karakters (0...9 en A...F). Voor elk informatiebyte zijn twee communicatiebytes noodzakelijk, omdat elk communicatiebyte slechts 4 bits kan definiëren in het hexadecimale systeem. De dubbele punt (':') wordt gebruikt om het begin van een bericht te markeren en elk bericht wordt beëindigd met een CR/LF combinatie.

Modbus/RTU

Met Modbus/RTU wordt de data uitgewisseld in binair formaat, waarbij elk informatiebyte is gecodeerd in één communicatiebyte. Modbus/RTU maakt gebruik van stilteperioden tijdens de communicatie om het begin en einde van een frame aan te geven. Elk bericht moet worden vooraf gegaan door een stilteperiode van minstens 3,5 karakters. Als een ontvanger binnen een frame een stilte 'hoort' van tenminste 1,5 karakters, dan neemt het aan dat er een nieuw bericht komt en wordt de ontvangstbuffer geleegd.

Modbus adressering

De eerste informatie in elk Modbus bericht is het adres van de ontvanger. Deze parameter bevat één byte informatie. Toegestane adressen liggen in de range 0...247. De waarden 1...247 worden toegewezen aan de slaves. Een slave beantwoordt altijd een Modbus adres. In zo'n antwoord bericht

wordt hetzelfde adres gebruikt als in het verzoek van de master. Op die manier kan de master zien dat de slave heeft geantwoord.

Intern in een Modbus apparaat zijn de holding registers, ingangen en uitgangen genummerd van 1 tot en met 10000. De adressen die in een Modbus bericht worden gebruikt om waarden te zetten of uit te lezen, zijn echter genummerd van 0 tot en met 9999. Voor de communicatie met holding registers (waar de configuratie van de unit staat opgeslagen) moet nog een andere offset worden afgetrokken van het apparaat adres om het juiste adres te verkrijgen dat in de Modbus bericht structuur moet worden geplaatst.

Tabel 8; Modbus adressen

Apparaat adres	Modbus adres	Omschrijving
1...10000*	adres – 1	Coils (uitgangen)
10001...20000*	adres – 10001	Ingangen
40001...50000*	adres – 40001	Holding registers

* Maximale waarde is apparaat afhankelijk

Modbus functie codes

De tweede parameter in elk Modbus bericht is de functiecode. Dit definieert het bericht type en het type actie die van de slave verwacht wordt. De parameter bevat één byte aan informatie. Toegestane functies liggen in het bereik 1...255. Niet alle Modbus apparaten herkennen dezelfde serie van functiecodes. Normaliter, wanneer een Modbus slave een verzoek beantwoordt gebruikt deze dezelfde functiecode als in het verzoek. Echter, wanneer een fout is gedetecteerd, wordt het hoogste bit van de functiecode aangezet. Op die manier kan de master succesvolle en mislukte antwoorden eenvoudig onderscheiden.

Tabel 9; Modbus functie codes

Code	Omschrijving
01	Lees coil status
02	Lees ingang status
03	Lees holding registers
04	Lees input registers
05	Forceer enkele coil
06	Preset enkel register
07	Lees uitzonderingsstatus
15	Forceer meerdere coils
16	Preset meerdere registers
17	Rapporteer slave ID

Interne waarden in een Modbus apparaat worden opgeslagen in holding registers. Deze registers zijn twee bytes groot en kunnen voor velerlei doeleinden gebruikt worden. Sommige registers bevatten configuratie parameters terwijl anderen worden gebruikt om meetwaarden terug te zenden naar een host. Registers in een Modbus compatibel apparaat beginnen bij 40001. Ze worden geadresseerd in Modbus berichten met adressen die beginnen bij 0. Modbus functie 03 wordt gebruikt om de waarde

van één of meerdere holding registers op te vragen. Slechts één slave apparaat kan worden geadresseerd in een verzoekbericht. [7]

Tabel 10; Opbouw bericht Modbus functie 03

Byte	Waarde	Omschrijving
1	1...247	Slave adres
2	3	Functie code
3	0...255	Startadres, hoge byte
4	0...255	Startadres, lage byte
5	0...255	Aantal registers, hoge byte
6	0...255	Aantal registers, lage byte
7(...8)	LRC/CRC	Foutcontrole waarde