

# **Ontwikkelen van PCS7 software voor een gesimuleerde fabriek**

Afstudeerverslag

**Sander Hassing**

Periode:	7 november 2011 – 16 maart 2012
Opleiding:	Elektrotechniek
Begeleider HHS:	J.E.J. op den Brouw
Begeleider CoNet:	R. Dingenouts

Deze versie is bestemd voor:  
HBO-kennisbank



## Voorwoord

Afstuderen, het laatste onderdeel van de HBO opleiding. Maar in wat voor richting precies, de Elektrotechniek is zo breed! En waar vind ik een leuk en geschikt stage bedrijf? Het selecteren van HBO Elektrotechniek als mijn studiekeuze was al lastig genoeg.

Dat was wat er door mijn hoofd ging nadat ik uiteindelijk alle voorgaande delen van mijn opleiding met succes had afgerond. Omdat mijn vooropleiding VWO was kon ik een versneld traject volgen waardoor het mogelijk was alle vakken en gewone stages binnen drie jaar af te ronden. Vanwege dat ik dit traject zo wel erg vlug vond verlopen heb ik mijn afstuderen nog een jaar uitgesteld door een jaar een plek in het bestuur van Studentenvereniging NOVA te bezetten. Een jaar dat mij veel levenservaring opleverde en veel leerde over samenwerken met andere personen en bedrijven. En daarna moest ik er toch aan geloven. Tijdens mijn studie had ik toch wel gemerkt dat mijn interesse binnen de Elektrotechniek in een richting lag die wat met programmeren te maken had. Nog steeds niet erg concreet, maar het was een idee. Bij NOVA's afsluitende feest van het schooljaar kwam een sponsor met flyers waarop "Engineer je baan!" stond. Goed, ik had nog niks, en dit zag er wel interessant uit. En zo kwam ik bij CoNet terecht, waar men automatiseringssystemen ontwikkelt met SIMATIC PCS7. Het sloot aan bij mij mijn interesses, niet te ver weg en goed bereikbaar, en er is een gezellige werksfeer. Een prima afstudeerplek dus!

Nu mijn tijd bij CoNet nog maar beperkt is wordt het hoog tijd om het verslag te schrijven. Alles wat ik onderzocht en gedaan heb staat hierin beschreven en ik wens de lezer veel leesplezier met dit document.

Rotterdam, 29 februari 2012,  
Sander Hassing



## Samenvatting

Bij het Scheepvaart en Transport College (STC) in Brielle worden MBO-studenten in een oefenfabriek opgeleid tot Operator. Deze oefenfabriek is een chemische fabriek op kleine schaal.

De oefenfabriek is onderverdeeld in meerdere 'Units' die op verschillende DCS-systemen draaien. Eén van deze onderdelen, Unit 700, wordt aangestuurd door een SIMATIC PCS7 systeem. Het STC maakt ook gebruik van gesimuleerde natuurkundige modellen van de oefenfabriek. Deze modellen worden ontwikkeld door Mobatec. Het model voor Unit 700 wordt tevens aangestuurd door SIMATIC PCS7. Het model communiceert met de PCS7-software via OPC. Deze PCS7-software wordt onderhouden door CoNet Rotterdam, waar de stage plaatsvindt.

Het ontwikkelen en testen van zowel het model, de OPC verbinding en de PCS7-software is het doel van deze afstudeeropdracht. Ook kan er Process Safety geïmplementeerd worden. Hiertoe word de werking van PCS7, OPC, en Mobatec Modeller onderzocht. Hierna worden de aanwezige implementaties van deze programma's in dit project geïnventariseerd, getest en verbeterd.

Uiteindelijk komt hieruit een completere PCS7 aansturing en een volledige OPC communicatie voort. Over het model worden een aantal problemen gerapporteerd.



## Inhoud

<b>Voorwoord.....</b>	<b>2</b>
<b>Samenvatting.....</b>	<b>3</b>
<b>Inhoud .....</b>	<b>4</b>
<b>1. Inleiding .....</b>	<b>6</b>
<b>2. CoNet .....</b>	<b>7</b>
2.1. Geschiedenis.....	7
2.2. Visie .....	8
<b>3. Opdrachtschrijving .....</b>	<b>9</b>
3.1. Probleemstelling.....	9
3.2. Doelstelling .....	10
3.3. Resultaat .....	10
3.4. Aanpak en werkwijze.....	11
3.5. Risico's .....	11
3.6. MoSCoW .....	11
<b>4. Unit 700.....</b>	<b>13</b>
4.1. Inleiding op de oefenfabriek en Unit 700.....	13
4.2. Doseren Grondstoffen .....	14
4.3. Reactie .....	14
4.4. Absorptie .....	14
<b>5. PCS7 .....</b>	<b>15</b>
5.1. De Siemens PCS7 Systeemcursus .....	16
5.2. Inleiding op PCS7 .....	17
5.2.1. Veldbuscommunicatie .....	20
<b>6. De PCS7-software voor Unit 700 .....</b>	<b>22</b>
6.1. PLC-SIM.....	22
6.2. Het OS .....	22
6.3. Testen en aanpassen van de software .....	26
6.3.1. Opstarten .....	27
6.3.2. Interlocks .....	28
6.3.3. Faceplate FlowInstellen .....	28
6.3.4. SFC's.....	30
6.3.5. PID-controllers.....	31
6.3.6. Feedback signalen .....	32
<b>7. OPC .....</b>	<b>33</b>
7.1. Inleiding op OPC .....	33
7.1.1. OPC Standaarden.....	34
7.2. OPC binnen het project .....	36
<b>8. Mobatec Modeller .....</b>	<b>38</b>
8.1. Inleiding op Mobatec Modeller .....	39
8.2. Het model voor Unit 700 .....	41



---

<b>9. Conclusie .....</b>	<b>45</b>
<b>10. Evaluatie .....</b>	<b>47</b>
<b>BIJLAGE I Afkortingen.....</b>	<b>48</b>
<b>BIJLAGE II Distributed Control Systems.....</b>	<b>49</b>
<b>BIJLAGE III Tekeningen Unit 700 .....</b>	<b>50</b>



## 1. Inleiding

Bij het Scheepvaart en Transport College (STC) in Brielle worden MBO-studenten in een oefenfabriek opgeleid tot Operator. Deze oefenfabriek is een chemische fabriek op kleine schaal, waar de studenten in een relatief veilige en gecontroleerde omgeving kunnen leren werken in een fabriek.

De oefenfabriek is onderverdeeld in meerdere 'Units' die op verschillende DCS systemen draaien. Een van deze Units, Unit 700, wordt aangestuurd door een SIMATIC PCS7 systeem. Unit 700 is een chemische reactor die een soda- en een azijnzuuroplossing combineert om koolstofdioxide te vormen. De koolstofdioxide wordt door een kalkwateroplossing geleid, waarbij het eindproduct gips ontstaat, dat uiteindelijk door middel van filters van het kalkwater gescheiden wordt. Het STC maakt ook gebruik van gesimuleerde natuurkundige modellen van de oefenfabriek. Deze modellen worden ontwikkeld door Mobatec en zijn een softwarematige representatie van de daadwerkelijke fysieke Units. Deze modellen worden gebruikt om zonder de hardware van de fysieke Unit toch een installatie te kunnen nabootsen. Het model voor Unit 700 wordt tevens aangestuurd door SIMATIC PCS7. Een reden om met de modellen te werken in plaats van een echte installatie is zodat er voor operators die zich in een vroeg stadium van hun opleiding bevinden toch een veilige oefenomgeving is, wanneer het mogelijk nog niet veilig is om te werken met de echte, fysieke Units.

De PCS7-software voor zowel de fysieke Unit 700 als het gesimuleerde model zijn in beheer bij CoNet Rotterdam. Het model zelf is in beheer bij Mobatec. Omdat dit, zoals reeds vermeld, een relatief veilige en gecontroleerde omgeving is, biedt dit ook voor CoNet een mooie mogelijkheid om dit project vrij te geven aan stagiairs en afstudeerders. Hiermee zijn CoNet en Siemens sponsors van de STC oefenfabriek. Omdat SIMATIC PCS7 zeer complex en uitgebreid is kreeg ik eerst de mogelijkheid om bij Siemens Zoetermeer de PCS7-systeemcursus te volgen. Deze cursus duurde twee weken; in deze periode werd kennis gemaakt met alle onderdelen van SIMATIC PCS7.

Dit verslag zal eerst kort het bedrijf CoNet beschrijven. Hierna volgt de verslaglegging van de daadwerkelijke afstudeerstage. Als eerste wordt de opdrachtomschrijving geformuleerd. Hierna wordt dieper ingegaan op de werking van Unit 700. Dan volgt een hoofdstuk over PCS7 waarin eerst de PCS7 systeemcursus wordt beschreven, en daarna een paragraaf waarin PCS7 globaal wordt beschreven. In het daaropvolgende hoofdstuk wordt dieper ingegaan op de specifieke PCS7-software voor Unit 700. Twee hoofdstukken, gericht op respectievelijk OPC en Mobatec Modeller, volgen daarna. Als laatste komt de conclusie, en een evaluatie betreffende de afstudeerperiode.

In dit document zullen ook vaak afkortingen van vaktermen gebruikt worden. Deze worden bij het eerste gebruik voluit geschreven en tevens uitgelegd. Een lijst van deze afkortingen en verwijzingen naar de desbetreffende uitleg is te vinden in BIJLAGE I. In BIJLAGE II wordt volledig beschreven wat een DCS precies is. BIJLAGE III bevat technische tekeningen van Unit 700. In BIJLAGE IV is een lijst te vinden met websites en rapporten die voor dit verslag gebruikt zijn.

## 2. CoNet

Dit hoofdstuk zal het ontstaan en de ontwikkeling van het bedrijf CoNet beschrijven, en wat de visie van het bedrijf is op de toekomst.

### 2.1. *Geschiedenis*

In 1996 zijn drie mensen het bedrijf CoNet Engineering gestart, in Zaandam. In de omgeving van Zaandam worden verschillende S7/WinCC-projecten uitgevoerd.

In het jaar 2000, na vier jaar gewerkt te hebben met de S7/WinCC-projecten, werd het belangrijk dat CoNet zich ging profileren, ook om helderheid naar de opdrachtgevers te geven. Dit is de reden dat CoNet toen Siemens Partner op het gebied van Factory Automation geworden is. Er kwam nog een belangrijke ontwikkeling in het jaar 2000. De huidige projecten in het beheer van CoNet moesten namelijk ook onderhouden worden. Maar dit laten doen door Project Engineers, die ook voor nieuwe projecten ingezet konden worden, was niet efficiënt. Op dat moment werd de bedrijfstak CoNet Maintenance opgericht, bedoeld voor het onderhoud van de afgeronde projecten.

Ondertussen groeide het bedrijf en verkregen zij een steeds grotere bekendheid onder potentiële klanten in de omgeving. De projecten werden hierdoor steeds groter en complexer. Siemens had hiervoor een betere oplossing dan S7/WinCC. Vanaf die tijd heeft CoNet zich ook gespecialiseerd in SIMATIC PCS7. In 2003 werden zij toen ook Siemens Process Automation Partner.

In 2006 werd de vestiging van CoNet te klein en verhuisden zowel de Engineering als de Maintenance afdeling naar de voormalige Verkade-fabriek, waar vroeger chocolade, brood en beschuit werd gefabriceerd, maar die derhalve tegenwoordig verbouwd is tot kantoorpand.



Fig. 1. De voormalige Verkade-fabriek



In 2007 wordt de bedrijfstak CoNet Migration opgericht. Dit deel van CoNet richt zich vanaf dat moment specifiek op de migratie van verouderde Process Automation systemen naar een modernere oplossing.

In het jaar 2010 werd het duidelijk dat er in de Rotterdamse Botlek nog een mooie kans was voor CoNet om zich uit te breiden. Daarom werd in dat jaar de afdeling CoNet Rotterdam opgericht. Om de traditie van oude fabrieken voort te zetten werd deze afdeling gevestigd in de voormalige koffie-, thee- en tabaksfabriek van Van Nelle.



**Fig. 2. De voormalige Van Nelle-fabriek**

Ook deze fabriek is verbouwd tot kantoorpand en biedt vele ruimtes van verschillende formaten voor bedrijven, waardoor het zeer geschikt is voor een groeiend bedrijf als CoNet Rotterdam.

## **2.2. Visie**

De visie van CoNet voor de toekomst (in de eerstvolgende jaren) is als volgt:

- CoNet is bekend bij alle middelgrote- en grote bedrijven in Nederland als dé Siemens System Integrator waar zij ook grote en internationale (DCS-, MES-, en migratie-) projecten willen onderbrengen.
- Senior automatiseringsspecialisten werken graag bij CoNet.
- CoNet realiseert, met aandacht voor het milieu, een substantiële bijdrage aan het behoud van de Nederlandse industrie.
- CoNet realiseert een bovengemiddelde groei, waarbij flexibiliteit en betrokkenheid gehandhaafd blijven.



### 3. Opdrachtschrijving

Dit hoofdstuk zal de opdracht omschrijven, met daarbij de verwachte resultaten. Daarna volgt er een korte beschrijving van de eventuele risico's.

#### 3.1. Probleemstelling

De originele opdracht van CoNet Engineering was om de chemische reactor skid bij het STC in Brielle volledig operationeel te krijgen. Deze skid wordt aangestuurd door Siemens PCS7 software en bevat een Profibus-netwerk.

Het STC is een MBO-instelling waar operators opgeleid worden. Hier staan negen fysieke "Units" opgesteld, die elk een miniatuurversie van een chemische fabriek vormen. Al deze Units reageren en werken exact als een echte fabriek. Om de studenten zo breed mogelijk op te leiden, worden alle Units aangestuurd door Process Control-systemen van verschillende fabrikanten. Daarnaast bestaat er van iedere Unit een simulatiemodel dat de praktijksituatie tot in detail kan nabootsen.

De skid die met PCS7 bestuurd wordt, bevat een chemische reactor waarmee uit de grondstoffen proceswater, soda en azijnzuur, de producten koolstofdioxide en natriumacetaat worden gefabriceerd. Koolstofdioxide reageert daarna nog met kalkmelk om gips te vormen.

Het STC wil echter op relatief korte termijn (rond de zomer van 2012) alle Units op hetzelfde Process Control systeem laten werken. Dit gebeurt omdat het vele malen eenvoudiger en goedkoper in onderhoud is. De kans is erg klein dat hierbij voor Siemens wordt gekozen, waardoor het fysieke Siemens systeem bij het STC zal verdwijnen. Hierdoor zal Siemens ook niet alle benodigde onderdelen gaan leveren om Unit 700 volledig operationeel te krijgen. Om de studenten toch met verschillende systemen te leren werken, zullen de afzonderlijke simulaties van de Units blijven bestaan. De simulatie van Unit 700 zal dus nog steeds worden aangestuurd door PCS7.

De PCS7 software voor de simulatie is al compleet, en de besturing wordt voor de operators gevisualiseerd in WinCC. Het model van de reactor is gemaakt door Mobatec. Mobatec is een bedrijf dat zich specialiseert in het maken van modellen van chemische, natuurkundige, en biologische processen.

De PCS7 software en het Mobatec model communiceren via OPC. Hierbij zal PCS7 ingericht worden als een OPC server en het Mobatec model als OPC client.

De opdracht is als eerste om de PCS7 software via OPC te koppelen aan het model. Hierna moet getest worden of alle processen die via de software aangestuurd worden precies werken zoals het zou moeten. Wanneer dit niet het geval is, moet de software worden aangepast om een juiste werking te verkrijgen. Ondanks dat Mobatec gespecialiseerd is in dergelijke modellen, is het echter ook mogelijk dat er fouten in het model zitten waardoor het proces niet verloopt zoals verwacht. In dat geval moet het model aangepast worden. Waarna dit volledig werkt zullen er ook Safety maatregelen ingebouwd moeten worden, voor wanneer er onverwachte situaties in het model geïntroduceerd worden. Wederom moet ook hierbij gecontroleerd worden of ook het model hier juist op reageert.



Wanneer hierna nog tijd over is, kan er ook nog aan de fysieke Unit 700 gewerkt worden, met de hardware die op dit moment wel bestaat. Er zijn nog een aantal verbeteringen mogelijk aan het Profibus-netwerk, het bevat nog wat storingen. Via een analyzer moeten deze achterhaald en, waar mogelijk, opgelost worden.

Ook de aansluitingen naar de AS zijn op dit moment nog niet goed. De digitale outputs hangen nog los, en een aantal dingen die wel zijn aangesloten zijn niet helemaal optimaal. Dit kan dus ook nog opgelost en verbeterd worden. Ook kan de PCS7 software wel getest worden op het deel van de opstelling dat wél aanwezig is.

Dit komt echter eventueel pas aan de orde na het werk aan de simulatie, omdat de simulatie wel blijft bestaan, maar de PCS7 AS en het Profibus-netwerk niet.

### ***3.2. Doelstelling***

Op projectmatige wijze een volledig werkende simulatie van Unit 700 maken. De simulatie en de PCS7-software moeten via OPC met elkaar communiceren. Alle processen binnen de software moeten volledig getest worden en waar nodig moet de software of het simulatiemodel aangepast worden.

Wanneer dit klaar is kan nog gekeken worden naar het Profibus netwerk en de I/O van de fysieke Unit 700.

### ***3.3. Resultaat***

De resultaten van het project zullen zijn:

- Een volledig werkende simulatie van Unit 700, dit omvat volledig werkende en geteste PCS7 aansturing, een volledig werkend en getest model, en volledig werkende en geteste OPC communicatie.
- Projectdocumentatie over de werking van de simulatie
- Een volledig werkende en geteste simulatie van Unit 700 inclusief Process Safety
- Documentatie over gebruik van Process Safety

Het resultaat voor het STC in dit project is dat studenten naar behoren kunnen worden opgeleid in het gebruik van een Siemens PCS7 systeem. Voor CoNet is dit een mooi project om de naamsbekendheid in deze regio te vergroten.



### **3.4. Aanpak en werkwijze**

Om dit project tot een goed einde te brengen zal de volgende aanpak worden gehanteerd:

**Stap 1:**

Kennismaken met de gebruikte software. Wat is PCS7 precies, wat doet het en hoe werkt het? Ditzelfde moet gebeuren voor OPC en Mobatec Modeller. Hierbij hoort ook de PCS7 Systeemcursus

**Stap 2:**

Inventarisatie van aanwezige software. Hoe compleet is de huidige PCS7 aansturing, OPC communicatie en het Mobatec model?

**Stap 3:**

In stap 2 gevonden problemen en gebreken aanpassen in de aansturing, communicatie, of het model. Sommige problemen in de aansturing zijn mogelijk het gevolg van problemen in het model of de communicatie.

**Stap 4:**

Testen van de (nieuwe) aansturing.

**Stap 5:**

Implementatie van Process Safety

### **3.5. Risico's**

Aangezien de opdracht al één maal aangepast moest worden in verband met de situatie binnen het STC, zou het kunnen voorkomen dat hierdoor later ook nog vertraging wordt opgelopen. Andere problemen of risico's zouden misschien het leren werken met nieuwe, onbekende software kunnen zijn, zoals Mobatec Modeller en OPC.

### **3.6. MoSCoW**

MoSCoW staat voor Must, Should, Could, en Won't. Aan de hand hiervan wordt de prioriteit van de te behalen resultaten ingedeeld. De betekenis van Must, Should, Could, en Won't is als volgt:

- **M** - Must: deze eis moet in het eindresultaat terugkomen, zonder deze eis is het product niet bruikbaar.
- **S** - SHOULD: deze eis is zeer gewenst, maar zonder is het product wel bruikbaar.
- **C** - COULD: deze eis mag alleen aan bod komen als er tijd genoeg is.
- **W** - WON'T: deze eis zal in dit project niet aan bod komen maar kan in de toekomst, bij een vervolgproject, interessant zijn.



---

In dit project zijn de te behalen resultaten als volgt verdeeld:

**Must**

- Een volledig werkende en geteste PCS7 aansturing.
- Een volledig werkende en geteste OPC communicatie.

**Should**

- Een volledig werkend en getest Mobatec model.
- Een volledig werkende en geteste Process Safety implementatie in de PCS7 software.

**Could**

- Testrapport van aanwezige apparatuur en Profibus-netwerk op fysieke Unit 700

**Won't**

- Uitbreiding met SIMATIC Batch

De eis bij Won't staat niet in de te behalen resultaten omdat dit niet zal gaan passen binnen het tijdsbestek van deze stage. Wel is het gewenst dat dit in de toekomst wordt toegevoegd.

## 4. Unit 700

In dit hoofdstuk wordt de chemische reactor, Unit 700, meer uitgediept.

### 4.1. Inleiding op de oefenfabriek en Unit 700

Unit 700 is een van de Units in de oefenfabriek van het STC. In de oefenfabriek bevinden zich nog negen andere Units. Deze Units beschrijven elk hun eigen proces maar zijn ook op sommige plekken verbonden met andere Units. Dit is bijvoorbeeld nodig voor de toevoer van grondstoffen of afvoer van afvalstoffen. Sommige Units gebruiken het eindproduct van een andere Unit als grondstof. Unit 700 vormt samen met Unit 300 een uitzondering, beiden voeren namelijk exact hetzelfde proces uit. Het verschil is dat Unit 300 wordt aangestuurd door het Experion PKS systeem van Honeywell. Ook Unit 1000 is een uitzondering, dit is namelijk de centrale controlekamer. Unit 700 wordt aangestuurd door SIMATIC PCS7, met een Profibus veldbussysteem. De besturing door de operator gebeurt met het SCADA programma WinCC, maar er zijn ook een aantal kleppen op de Unit die met de hand bediend moeten worden. Unit 700 begint met het maken van een soda- en een azijnzuuroplossing. Hiervoor wordt gezuiverd proceswater afkomstig uit Unit 800 gebruikt. Deze oplossingen reageren tot natriumacetaat. Het daarbij ontstane koolstofdioxide wordt gemengd met kalkmelk, waaruit dan gips gefilterd kan worden. De hieruit voortkomende afvalstromen worden in Unit 800 weer gezuiverd voor hergebruik. Unit 700 kan eenvoudig opgedeeld worden in het doseren van de grondstoffen, de reactie, en de absorptie. Deze processen worden hieronder uitgediept. Ter verduidelijking zijn er in BIJLAGE III tekeningen te vinden van Unit 700.



Fig. 3. Een foto van de skid met daarin Unit 700





## **4.2. Doseren Grondstoffen**

In dit gedeelte van Unit 700 worden een soda- en een azijnzuuroplossing gevormd.

Voor de sodaoplossing worden eerst twee vaten, V-713 en V-714, gevuld met proceswater, afkomstig uit Unit 800. Een automatisch aangestuurde klep zorgt in combinatie met een flowmeter voor de juiste hoeveelheid, welke in WinCC wordt ingesteld. Hierna wordt soda toegevoegd aan de vaten. Dit moet handmatig gebeuren via een klep boven in het vat. Twee mixers zorgen dan dat de soda en het proceswater goed gemengd worden. Hierna worden vanuit deze twee vaten de oplossingen naar vat V-712 geleid. Ook hier worden zij nogmaals gemixt, en blijven hierna opgeslagen voor gebruik in het Reactie gedeelte.

Voor de azijnzuur oplossing wordt vat V-701 gevuld met proceswater. Vat V-703 bevat het azijnzuur wat door middel van een pomp gedoseerd wordt naar het proceswater. Het is niet nodig om dit te mixen. De oplossing wordt in vat V-701 bewaard totdat het in het Reactie gedeelte wordt gebruikt.

## **4.3. Reactie**

In het Reactie gedeelte reageren de soda- en de azijnzuuroplossing samen tot natriumacetaat. Hierbij komt koolstofdioxide vrij, dat wordt afgevoerd naar het laatste onderdeel, de Absorptie.

De soda- en azijnzuuroplossingen worden door middel van pompen en automatische kleppen in de juiste verhouding naar reactor R-701 geleid. Hier vindt de reactie plaats en wordt het eerste deel van de koolstofdioxide weg geleid. Ondertussen gaat het gevormde natriumacetaat door naar vat V-702, waar het blijft tot het volledig is uitgereageerd. Ook hier wordt koolstofdioxide afgevoerd. Als laatste gaat het natriumacetaat naar de ontgasser: C-701. De druk is hier lager dan in de rest van het proces. Door deze lage druk en door verwarming met behulp van stoom afkomstig uit Unit 600, kan het laatste restje koolstofdioxide van het natriumacetaat gescheiden worden. Het natriumacetaat wordt daarna afgevoerd naar Unit 800 voor zuivering.

## **4.4. Absorptie**

Hier wordt de koolstofdioxide geabsorbeerd in kalkmelk, waarbij gips ontstaat dat via speciale filters gescheiden kan worden van de oplossing.

Allereerst worden in de vaten V-711A en V-711B kalkmelk oplossingen gemaakt. Dit gebeurt met proceswater en handmatig toegevoegde kalk. Met mixers wordt dit goed gemengd. Twee pompen pompen dit dan rond via de absorber C-702, waar de koolstofdioxide uit het Reactie gedeelte wordt geabsorbeerd. Na de absorber komt de oplossing in de filters S-701A en S-701B terecht. Hier wordt het ontstane gips uit de oplossing gefilterd.

## 5. PCS7

Dit hoofdstuk zal ingaan op SIMATIC PCS7. Als eerste wordt de PCS7 systeemcursus kort beschreven. Daarna volgt een globale uiteenzetting over PCS7 en het bijbehorende veldbus-systeem. Dit is het PCS7 deel van stap 1 uit het in paragraaf 3.4 opgestelde stappenplan.

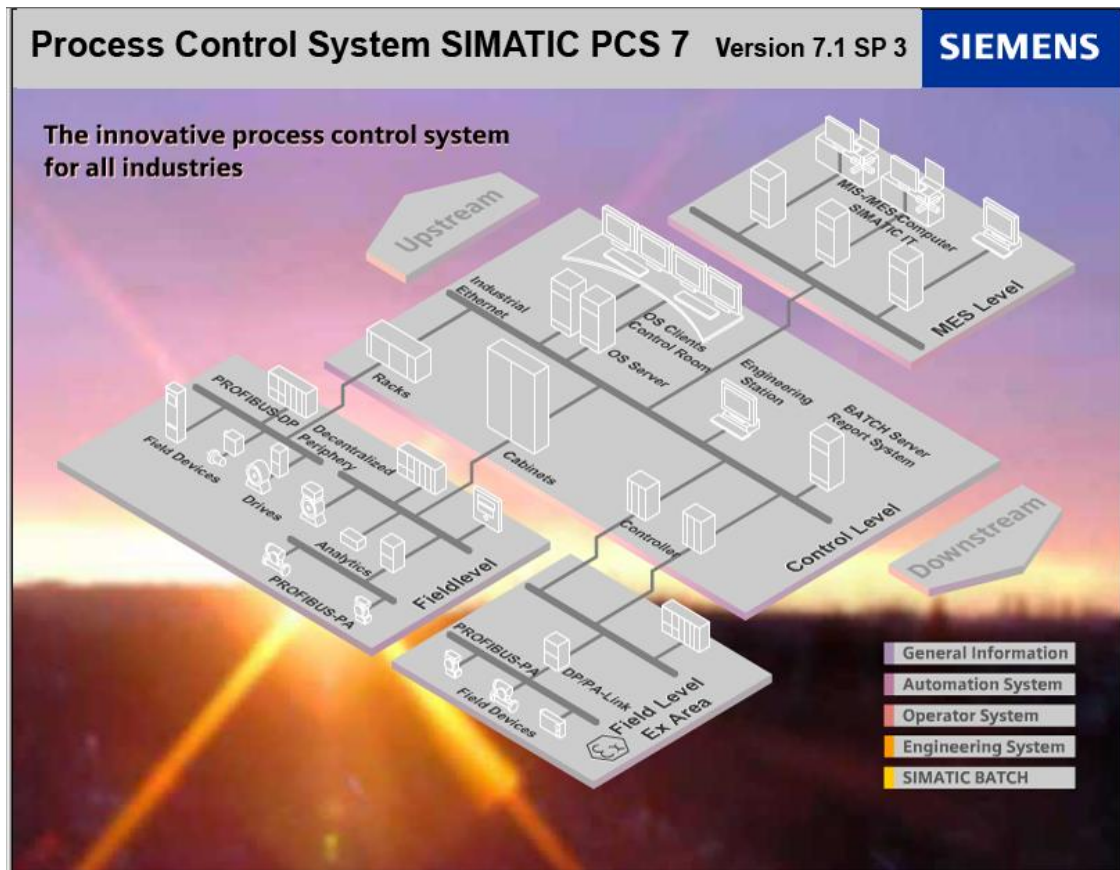


Fig. 4. SIMATIC PCS7





## **5.1. De Siemens PCS7 Systeemcursus**

Voorafgaand aan de stage heb ik de twee weken durende PCS7 Systeemcursus van Siemens gevolgd. Deze werd gegeven bij Siemens Zoetermeer. De planning van de cursus kwam erg goed uit, want deze begon één week voor de stageperiode; de tweede week was de eerste week van de stageperiode. CoNet bood deze cursus aan omdat er met PCS7 gewerkt diende te worden. Zonder deze cursus hiermee starten zou onverantwoord zijn.

In deze training staat het engineeringtraject van een project centraal. Alle gangbare stappen van zowel de hardware- als software engineering worden uitgevoerd met behulp van een uitgebreide realistische processimulatie.

De processimulatie draait samen met de op te bouwen applicatie in de originele SIMATIC PCS7 hardware, inclusief centrale en decentrale I/O. Daarbij wordt een deel van de processignalen op hardware niveau gebracht ter beoefening van alle relevante hardwareaspecten, zoals diagnosefuncties.

Per één of twee deelnemers wordt gebruik gemaakt van een gecombineerd engineering-/operatorstation, gebaseerd op de actuele V7.1 van de SIMATIC PCS7 Engineering Toolset.

De onderwerpen in deze cursus zijn:

- Systeemoverzicht
- SIMATIC PCS 7 Engineering Toolset
- Beheer van projectdata
- Stationsconfiguratie
- Netwerkconfiguratie
- I/O en drijverconcept
- Standaard- en projectbibliotheken
- Basic operations met CFC (APL: uitsluitend gebaseerd op de Advanced Process Library)
- Sequences met SFC
- Operator System met Grafische procesbeelden, bediening, meldconcept en trending
- Gebruikersbouwstenen met SCL of CFC

Aan het einde van de cursus krijgt de student zijn project mee op een USB-stick en de cursus map met daarin alle opdrachten, die later nog gebruikt kan worden als naslagwerk bij het werken met de SIMATIC software. Aansluitend aan de systeemcursus kan ook ter verdieping een 2-daagse cursus met praktijkoefeningen, een 5-daagse cursus AS Engineering of een 5-daagse cursus OS Engineering worden gevolgd. Ook kan de engineer nog een speciaal examen voor Certified SIMATIC PCS7 Engineer afleggen, maar aan te raden is dat de engineer dan eerst al geruime tijd met PCS7 heeft gewerkt.

## 5.2. SIMATIC PCS7

SIMATIC PCS 7 is het distributed control system (DCS) van Siemens. In deze paragraaf zal verder op de werking van dit systeem worden ingegaan. Een verdere uitleg over DCS-systemen is te vinden in BIJLAGE II. De figuur hieronder geeft schematisch het hele SIMATIC PCS 7-systeem weer.

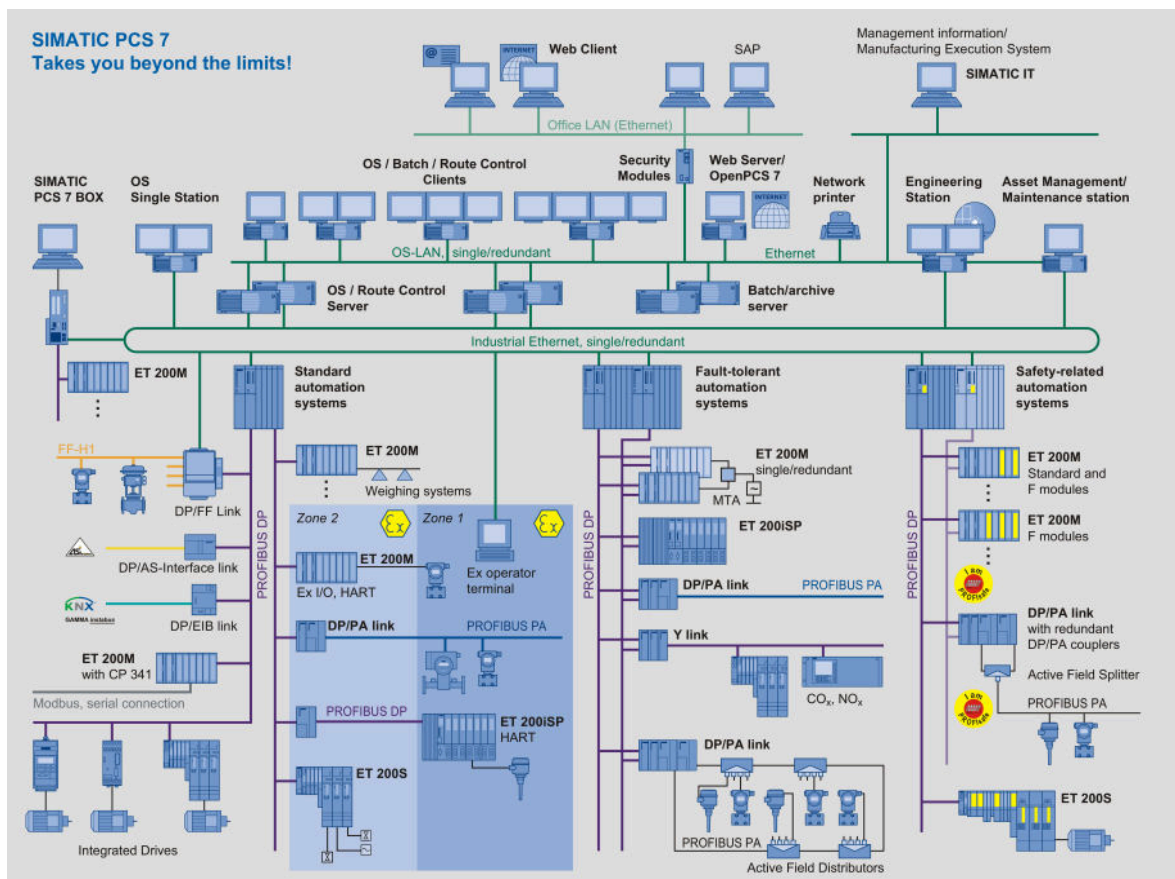


Fig. 5. Schematische weergave van een volledige SIMATIC PCS7-configuratie

SIMATIC PCS 7 onderscheidt zich van andere DCS-systemen doordat het gehele systeem van sensor tot en met MES nu één systeem is. MES staat voor Manufacturing Execution Systems: dit is de automatiseringslaag tussen de proces- en de productiebesturing op de fabrieksvloer en ERP-systemen, Enterprise Resource Planning-systemen, in de kantooromgeving. Dit komt later in deze paragraaf uitgebreider aan bod. Het systeem maakt gebruik van open standaarden op ieder niveau.

De kern van SIMATIC PCS 7 bestaat uit een Automation System (AS) op controllerniveau, een Operator System (OS) op visualisatie- en besturingsniveau en een Engineering Station (ES) voor engineering, beheer en onderhoud.

De AS dient voor de besturing van de controllers. Hiervoor wordt een Siemens S7 PLC gebruikt. Voor de communicatie tussen de PLC en de I/O wordt gebruikt gemaakt van de Profibus.

De OS is de visualisatie van het systeem. Op verschillende OS clients is het systeem te volgen en kan de operator ingrijpen wanneer er fouten in het proces optreden.

De ES is bedoeld om het volledige systeem te beheren. Wanneer er bijvoorbeeld een nieuw proces bij komt kan dit op de ES worden geprogrammeerd. Vanuit de ES is het dan mogelijk zowel de AS als de OS te updaten met de nieuwe installatie of veranderingen. De fabriek kan dan gewoon doordraaien en het nieuwe proces wordt automatisch overal toegevoegd. Dit heeft als voordeel dat er geen fouten kunnen optreden die ontstaan door verschillen tussen het programmeren van de PLC en het programmeren van de visualisatie-server.

De kracht van SIMATIC PCS 7 zit in een gestructureerde opbouw volgens ISA 88 en het automatisch genereren van zowel controller- als visualisatie-objecten. Afhankelijk van de applicatie kan SIMATIC PCS 7 verder uitgebreid worden met Client-Server architectuur, redundantie op ieder niveau, Asset Management, geïntegreerde Batch- & Route Control- en Safety-functionaliteit en SIMATIC PCS7 Web. Deze termen worden hieronder toegelicht.

### **Asset Management**

Asset Management betekent het intelligent beheren en onderhouden van proces automatisering. Het doel is om informatie van de servers, netwerkcomponenten en intelligente field devices te krijgen op het moment dat onderhoud nodig is. Dus ruim voordat het component niet meer kan functioneren binnen het systeem. Informatie over de temperatuur, fansnelheid en harddiskkwaliteit van de servers geeft aan wanneer deze aan onderhoud toe zijn. Informatie van netwerkcomponenten over de kwaliteit van de verbindingen geeft aan wanneer het netwerk te zwaar belast wordt. Informatie van HART devices en Profibus PA componenten zoals vervuiling, onderhoudsinterval en overschrijden van limieten wordt transparant doorgegeven aan de operator. Door de hiërarchische opbouw is het mogelijk vanuit ieder scherm direct door te gaan naar de component die de vervuiling meldt. Vanuit hier kan men tevens een interactie met de onderhoudsdienst starten en is de status van de voortgang te bewaken.

### **ISA-88 en Batchprocessen**

Een batchproces is een productiewijze waarbij telkens een afgeronde partij of hoeveelheid van een product wordt gefabriceerd, een batch genaamd. In de procesindustrie worden batchprocessen toegepast om allerlei producten te maken die niet of moeilijk met een continue proces te maken zijn.

Een voorbeeld van een batchproces in het dagelijkse leven:

In de keuken bevinden zich allemaal productiemiddelen zoals gasfornuis, magnetron, mixer, verschillende soorten pannen enz. Allemaal middelen die gebruikt worden om een maaltijd te bereiden. Het recept, de batch, bepaalt welke productiemiddelen en ingrediënten er uiteindelijk gebruikt gaan worden. Een productiemiddel, bijvoorbeeld een koekenpan, kan voor meer dan alleen maar pannenkoeken bakken gebruikt worden.

Om processen te kunnen verbeteren en kennis en kunde met anderen te delen zijn er standaarden ontwikkeld. Eén van deze standaarden is de ANSI/ISA-88. Deze standaard benadert het batch-proces gezien vanuit de MES filosofie en beschrijft de koppeling tussen de PLC- en de MES-laag.

SIMATIC Batch is volledig ISA-88 gebaseerd en geïntegreerd binnen de SIMATIC PCS 7-structuur. Ook op visualisatie- en gebruikersniveau is batch volledig geïntegreerd, waardoor een eenvoudige batch-bediening vanuit de OS mogelijk is. De volledige batchafloop is visueel te volgen en te beïnvloeden. Batchplanning,

receptbeheer en rapportage kunnen desgewenst binnen hetzelfde systeem toegepast worden, maar ook verspreid worden over diverse gebruikers.

### **Route Control**

Route Control is geschikt voor het besturen en beheren van routes. Dit kunnen eenvoudige transportroutes zijn, maar ook complexe flexibele route combinaties. Een route kan geschikt zijn voor vloeistoffen of vaste stoffen. Per deelroute zijn de eigenschappen en voorwaarden aan te geven waar de routebepaling rekening mee moet houden. Dit kan het in batches verwerken van producten zijn, maar ook complexe manifold aansturing of stoffen die nooit na elkaar door een leiding mogen. Deze eigenschappen zijn vooral van belang in de chemische, petrochemische of voedings- en genotsmiddelen-industrie.

Ondanks de complexe routebepalingen, is PCS7 Route Control zeer eenvoudig in gebruik. De Wizard helpt bij het bepalen van de routevoorwaarden en het inrichten van de projecteigenschappen.

### **Redundantie - Beschikbaarheid op ieder niveau**

De beschikbaarheid van de fabriek is een belangrijk uitgangspunt van SIMATIC PCS 7. SIMATIC PCS 7 biedt verhoogde beschikbaarheid op ieder niveau waar het nodig is.

Sensoren zijn twee- of drievoudig uit te voeren en geven diagnoses af binnen SIMATIC PCS 7. De I/O kaarten kunnen twee- of drievoudig uitgevoerd worden voor verhoogde beschikbaarheid. Het drievoudig uitvoeren van I/O kaarten en sensoren heeft als voordeel dat wanneer één kaart of sensor kapot gaat er twee goede signalen over blijven. Daarna is dan beter te bepalen wat de goede waarde is en welke de foute waarde is (bijvoorbeeld stuck at 1).

Eveneens is het mogelijk dat de veldbusbekabeling fysiek als ringstructuur op basis van Profibus DP en PA wordt aangelegd en enkel of redundant wordt uitgevoerd. De controllers hebben in enkele uitvoering een zeer hoge beschikbaarheid en kunnen middels een gepatenteerd synchronisatie-mechanisme redundant worden uitgevoerd. De communicatie naar de servers kan als ringstructuur op basis van Ethernet worden aangelegd en ook zowel enkel als redundant worden uitgevoerd. Ook de servers kunnen gesplitst worden naar functionaliteit en per functie enkel of redundant worden uitgevoerd. Zelfs de Batch server kan dubbel uitgevoerd worden, evenals de koppeling naar de clients en MES laag.

### **MES en ERP systemen**

Zoals eerder vermeld is MES de automatiseringslaag tussen de proces- en de productiebesturing op de fabrieksvloer en ERP-systemen in de kantooromgeving. Het is een samenstelling van technologieën en technieken die zorg dragen voor een efficiëntere manier van produceren. MES regelt niet alleen de informatieoverdracht, het voegt functionaliteit toe aan de transactie verwerkende ERP-systemen en de real-time productiebesturingssystemen. Bijvoorbeeld functies voor gedetailleerde planning, "work in process", kwaliteitsbeheer en product-tracking en -tracing.

ERP-systemen zijn computerprogramma's waarmee de resources zoals geld, voorraden, mensen en machines van een organisatie gepland (beheerd) kunnen worden. Men noemt dit ook wel backbone-systemen omdat deze de ruggengraat vormen van bedrijven. In deze systemen worden de financiële administratie, de voorraadadministratie, de inkoop- en verkoopadministratie, de personeelsadministratie etc. gevoerd. Dit is afhankelijk van het type organisatie. De meeste IT-activiteiten zijn gekoppeld aan een dergelijk systeem. Dit kunnen heel uitgebreide systemen zijn met duizenden gebruikers per bedrijf.



### **SIMATIC PCS7 Web**

Het proces is te bedienen via Internet Explorer en is eenvoudig, comfortabel en snel. PCS7 biedt de mogelijkheid flexibel om te gaan met de procesbediening door middel van een standaard PC met Internet Explorer. Hierdoor is het systeem niet gebonden aan een vaste Bedienplaats.

Een operator kan eenvoudig de assistentie van een procestechnoloog vragen, die met een standaard kantoor-PC via intranet met het proces kan meekijken en advies kan geven. Een operator in het buitenland kan zijn ervaringen delen met een operator in Nederland. Verschillende fabriekslocaties kunnen functioneren als stand-by Bedienlocatie om de beschikbaarheid te verhogen.

Om de veiligheid te blijven garanderen voorziet SIMATIC PCS 7 Web in een geavanceerd gebruikersbeheer, met vergrendeling van onderdelen, met alleen kijken of ook bedienen. Ook industriële VPN (Virtual Private Network)-koppelingen behoren tot de mogelijkheden.

### **5.2.1. Veldbuscommunicatie**

Profibus is wereldwijd een van de bekendste veldbussystemen en is geïntegreerd binnen SIMATIC PCS 7. Profibus is een open veldbus-systeem waarvan de openheid en interoperabiliteit gegarandeerd wordt door de Profibus-vereniging met meer dan 1200 leden. Naast Profibus ondersteunt SIMATIC PCS 7 diverse andere veldbus-technieken zoals Modbus, As-interface, Foundation Fieldbus en andere bekende veldbus-systemen binnen de procesindustrie. Via Profibus is het mogelijk PDM (Proces Device Manager) en het HART-protocol te gebruiken.

#### **Profibus DP en PA**

Profibus DP (Decentralized Peripherals) wordt wereldwijd binnen vrijwel iedere procesindustrie toegepast als remote I/O systeem. Door middel van Profibus DP zijn lange afstanden te overbruggen en veel signalen snel uit te wisselen. Dit in combinatie met verhoogde beschikbaarheid Atex zone 1/21-toelating en Failsafe-certificering maakt het systeem zeer populair. Profibus DP maakt transparante integratie van HART-signalen mogelijk waardoor intelligente sensoren gekoppeld kunnen worden in clusters.

Het koppelen van sensoren rechtstreeks aan de veldbus wordt mogelijk met Profibus PA (Process Automation). Hierbij wordt de instrumentatie gevoed via de bus en is als lijn- of ringstructuur te bouwen. Het aansluiten van de instrumentatie kan middels droplines of junctionboxes. Profibus PA is geschikt voor toepassing in Atex zone 1/21 en is gecertificeerd voor toepassing met Failsafe-instrumentatie. Vanwege de open standaard kan ieder PA-instrument aangesloten worden op ieder PA-systeem. Als het instrument tevens "profile 3" compatible is, kan het 1:1 met een ander fabricaat uitgewisseld worden zonder configuratie aanpassingen.

#### **Proces Device Manager**

PDM is ontworpen voor het in bedrijf nemen en onderhoud van de instrumentatie. Door middel van PDM krijgt het netwerk transparant toegang tot alle instrumentatie. Vanaf ethernet, via Profibus DP, PA en HART is het mogelijk toegang te krijgen tot alle informatie in de instrumentatie. PDM is geschikt voor alle merken instrumentatie op de markt en is zeer geschikt voor frequentie-omvormers en Motor Control Centers. PDM



is rechtstreeks gekoppeld met Asset Management zodat de informatie snel en gestructureerd beschikbaar is voor operator en onderhoudsdienst.

### **HART Protocol**

Het HART Protocol (Highway Addressable Remote Transducer Protocol) is een veldbus protocol voor de automatiseringsindustrie. Het protocol dankt zijn succes aan de communicatie over de 4-20 mA analoge bedrading die bij oudere systemen gebruikt wordt. Het HART Protocol wordt gebruikt voor intelligente fieldbus-instrumentatie. De instrumentatie kan met HART worden geprogrammeerd en uitgelezen. Daarnaast is het mogelijk informatie te krijgen over de toestand van de instrumentatie en om problemen op te lossen. Dit maakt het HART-protocol één van de meest populaire protocollen in de industrie.



## 6. De PCS7-software voor Unit 700

Dit hoofdstuk zal de PCS7-software voor Unit 700 meer uitdiepen. Zoals beschreven in paragraaf 3.4 is eerst de aanwezige software geïnventariseerd. Daarna zijn er tests uitgevoerd en aanpassingen gemaakt, te lezen in paragraaf 6.3.

Op de simulatie PC werkt de PCS7 software samen met het model in Mobatec Modeller. Waar in een reguliere situatie de communicatie tussen het OS en het AS verloopt via Industrial Ethernet, wordt de communicatie in dit geval geregeld via OPC. Over OPC wordt in hoofdstuk 7 meer verteld.

### 6.1. PLC-SIM

Bij gebrek aan een fysieke AS draait het PCS7-programma in dit geval op PLC-SIM, een simpele, gesimuleerde versie van een AS. Een nadeel van PLC-SIM is dat bij iedere herstart voor de zekerheid het programma volledig opnieuw ingeladen moet worden. PLC-SIM heeft wel een "Save" functie om zijn inhoud op te slaan, maar deze werkt helaas niet altijd, waardoor een bestand zonder inhoud wordt opgeslagen. Dit is een bekend probleem van PLC-SIM waar vooralsnog geen oplossing voor is. Het opnieuw downloaden van de software kost echter weinig tijd waardoor dit probleem geen grote gevolgen heeft.

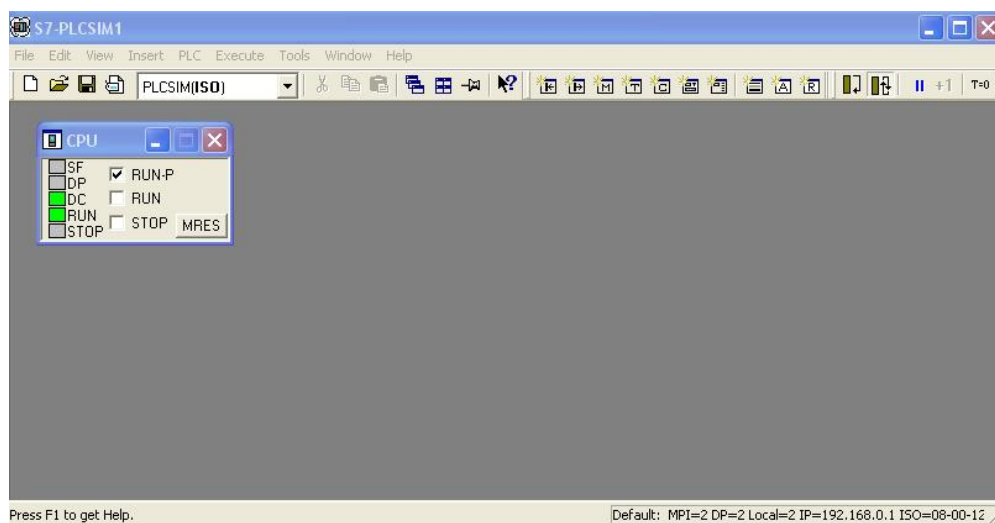
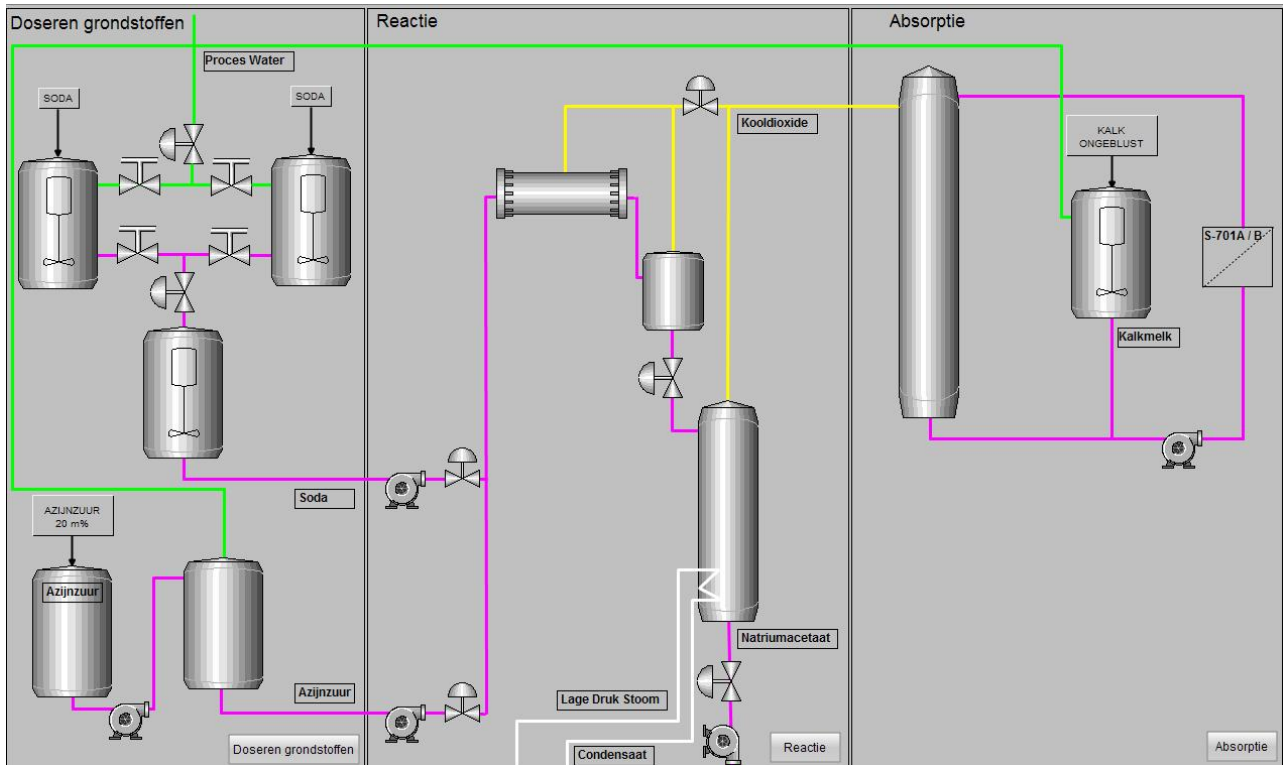


Fig. 6. PLC-SIM

### 6.2. Het OS

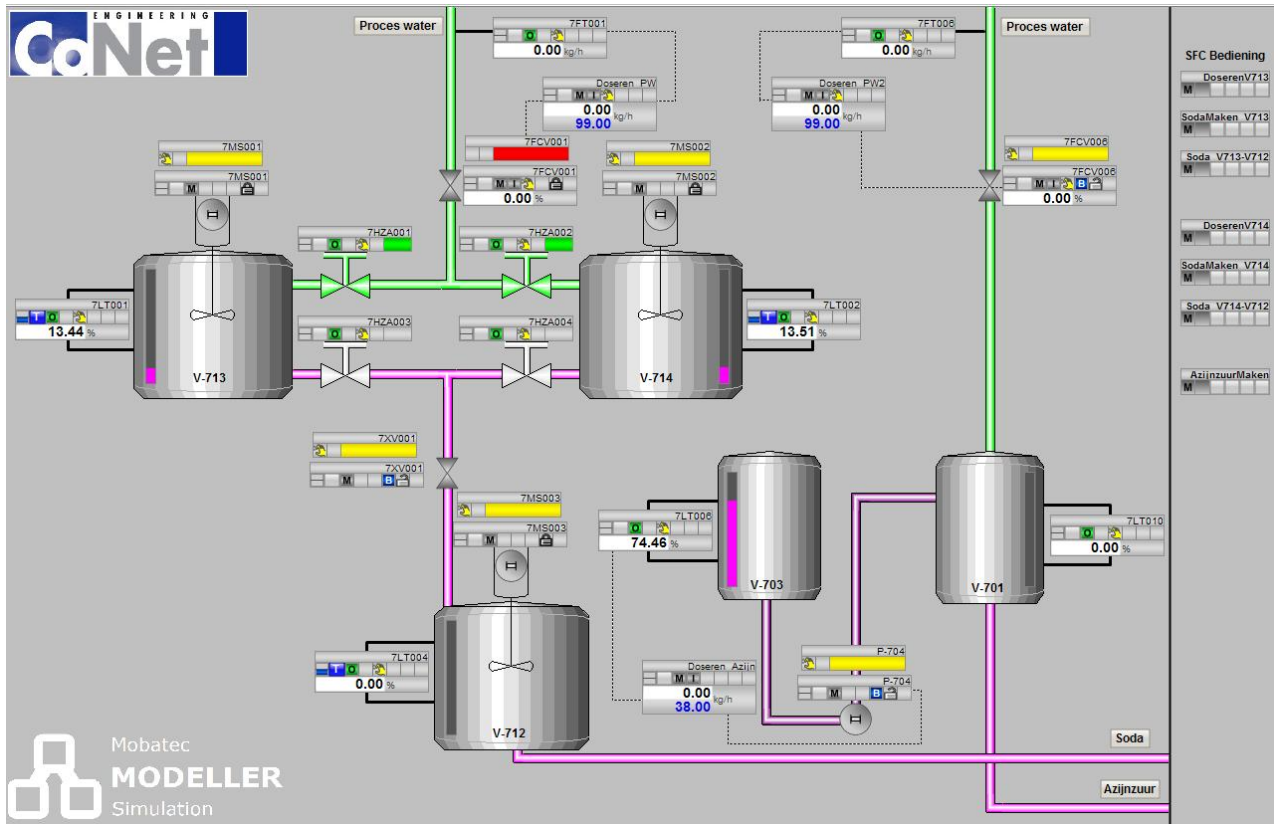
De visualisatie van de processen en aansturing voor het OS gebeurt in het programma WinCC. Een groot deel hiervan wordt automatisch gegenereerd bij de compilatie van het OS. Deze visualisatie is ook verdeeld in de onderdelen Doser en Grondstoffen, Reactie en Absorptie. Een overzicht van de visualisatie is op de volgende bladzijde te zien.





**Fig. 7. Overzicht van de WinCC visualisatie**

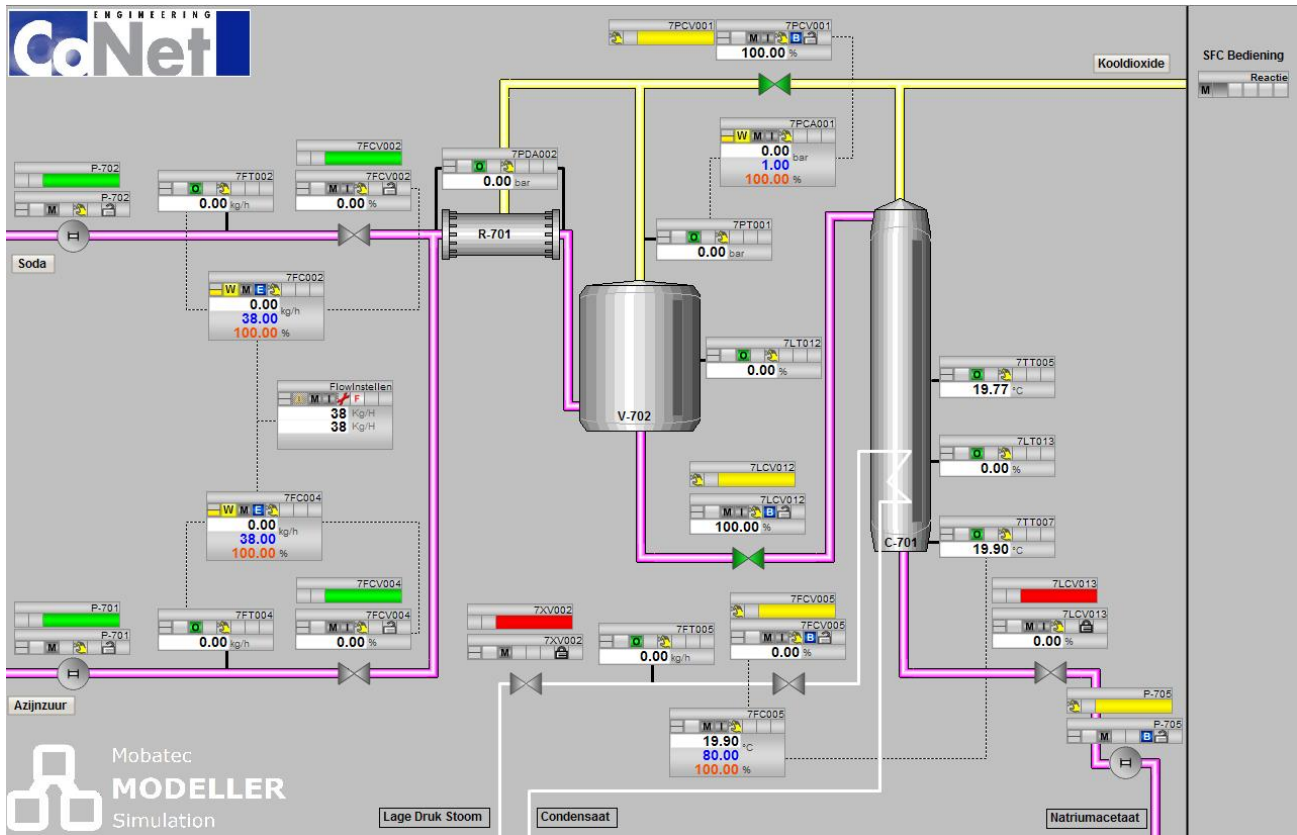
Dit overzicht bevat ook knoppen om door te klikken naar de verschillende onderdelen: dit voor een meer gedetailleerde weergave. Deze detailweergaves zijn op de hierop volgende pagina's te zien. In het overzicht zelf is geen aansturing mogelijk. Dit gebeurt in de detailweergaves. Ook is er in deze detailweergaves veel meer informatie te zien van de verschillende gesimuleerde sensoren.



**Fig. 8. De visualisatie van het 'Doseran Grondstoffen' onderdeel**

Dit is het eerste deel van de Unit 700, waar de soda- en azijnzuuroplossingen worden gemaakt. Leidingen die proceswater vervoeren zijn hier met groen aangegeven. Paarse leidingen zijn leidingen met een willekeurige vloeistof die onderdeel is van het proces. Het proceswater komt bovenin het scherm binnen. Hier bevindt zich dan ook de besturing en controle van de doseer-blocks en de hierdoor automatisch aangestuurde kleppen. Deze kunnen ook door de operator zelf in WinCC worden aangestuurd, in plaats van door de doseer-blocks. Tussen de vaten V-713 en V-714 zijn vier kleppen te zien. Dit zijn weergaves van handafsluiters met digitale monitoren. Deze kleppen kunnen niet bestuurd worden via WinCC. WinCC geeft alleen hun status weer. Er bevinden zich meer van deze kleppen in het model, maar in verband met de beperkte ruimte op het scherm worden deze verder niet weergegeven in WinCC. De vaten bevatten staafindicatoren waarop de inhoud van het vat wordt weergegeven. Ook is er bij de kleppen, pompen en mixers een weergave van de interlocks te zien. Interlocks worden verder besproken in paragraaf 6.3.2.

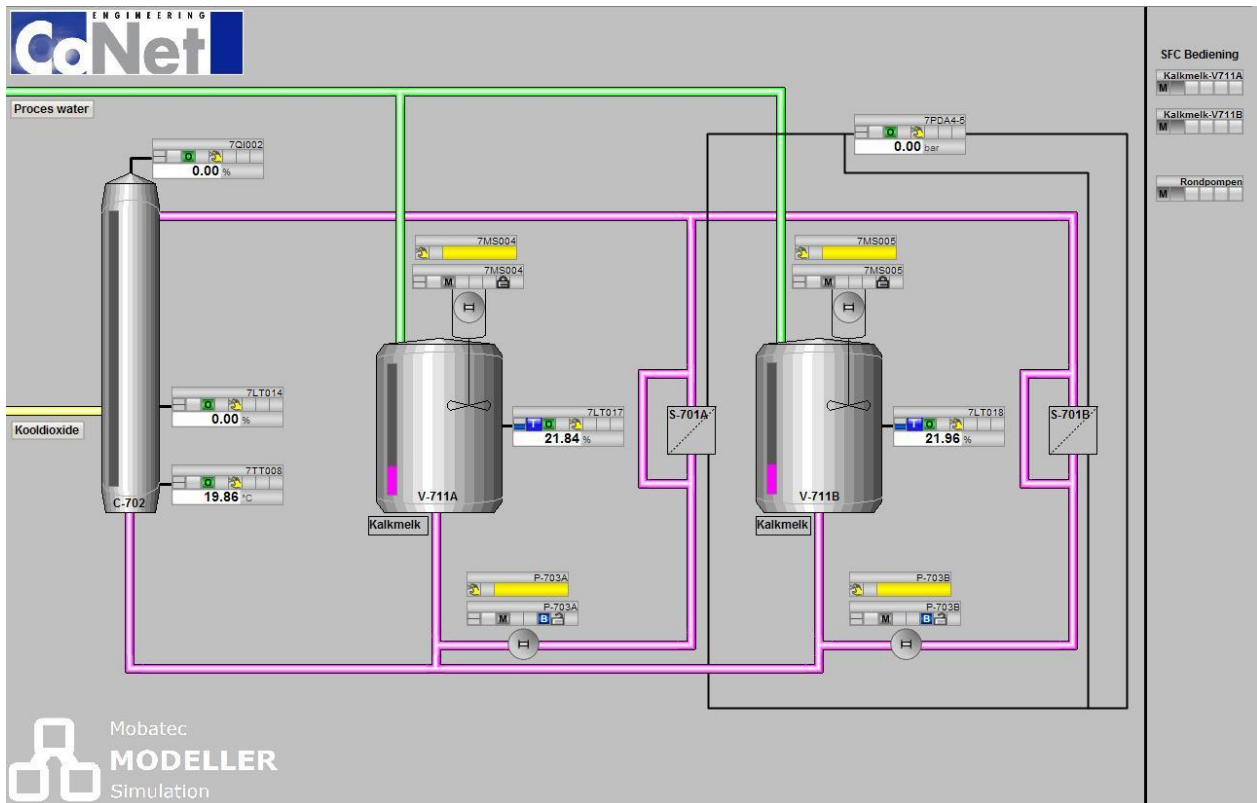
Aan de rechterkant is de bediening voor de verschillende SFC's te zien. Een SFC bestuurt grotendeels automatisch een bepaald deel proces en stuurt kleppen open en pompen aan. Alle pompen en kleppen kunnen echter ook door de operator zelf bediend worden in WinCC. De mogelijkheden en precieze werking van een SFC worden later in paragraaf 6.3.4 verder uitgelegd. Dit scherm bevat SFC's voor het doseren van proceswater naar de vaten V-713 en V-714, Het maken van een soda-oplossing in deze vaten, en het transporteren van deze oplossing naar vat V-712. Ook is er een SFC die zorgt voor de productie van de azijnzuuroplossing.



**Fig. 9. De visualisatie van het 'Reactie' onderdeel**

Dit is de visualisatie van het Reactie gedeelte. Dit deel draagt zorg voor de reactie tussen de soda- en azijnzuuroplossingen tot koolstofdioxide en natriumacetaat. Het natriumacetaat wordt rechtsonderin via een klep en een pomp afgevoerd voor zuivering in Unit 800. De koolstofdioxide gaat linksboven in verder naar het laatste onderdeel. Links komen de soda- en azijnzuuroplossingen binnen vanuit het voorgaande gedeelte. Met geel wordt de leiding met koolstofdioxide aangegeven. Aan de linkerkant bevinden zich als eerste de pompen die de oplossingen richting de reactor pompen. Daarna komt de bediening en controle voor de flow. Hierover wordt later meer verteld in de paragrafen over FlowInstellen en PID-controllers (respectievelijk 6.3.3 en 6.3.5). Dan volgen de verschillende vaten waar de reactie plaatsvindt en de koolstofdioxide afgevoerd wordt. Onder in het midden is de bediening en controle voor de stoomverwarming van vat C-701 te vinden. Boven in zijn sensoren, een klep en een PID-controller die de druk handhaven te vinden.

Het Reactie gedeelte heeft één SFC die dit gehele deelproces bestuurt.



**Fig. 10. De visualisatie van het 'Absorptie' onderdeel**

Het laatste deel van de visualisatie en het proces is de Absorptie. Rechts komt de koolstofdioxide binnen in de absorber C-702. In het midden zijn de vaten V711A en B te zien waar kalkmelk wordt gemaakt met handmatig toegevoegde kalk en proceswater. Deze kalkmelk wordt via de twee pompen rondgepompt langs de absorber en de filters S-701A en B. Ook is er een indicator voor de druk in de filters.

Het Absorptie gedeelte bevat drie SFC's. De eerste twee zorgen voor het produceren van de kalkmelk. De derde SFC zorgt voor het rondpompen van de kalkmelk.

### **6.3. Testen en aanpassen van de software**

De programmering voor de aansturing is al aanwezig. Deze is echter volledig ongetest. Ook is deze aansturing oorspronkelijk ontworpen voor de fysieke opstelling van Unit 700. Hoe het model op het programma reageert is dus ook nog volledig onbekend. Voor het testen van de software is als volgt te werk gegaan. Allereerst was het van belang dat het model draaide, het OS draaide en dat hiertussen op de juiste manier gecommuniceerd werd. Om dit voor elkaar te krijgen moet alles in de juiste volgorde gestart worden. Dit staat beschreven in paragraaf 6.3.1.

Nadat de volledige omgeving naar behoren draaide kon begonnen worden met het echte testen. Allereerst zijn er toen wat kleine visuele aanpassingen verricht aan het OS. Sommige leidingen liepen niet helemaal mooi, en de SFC bediening aan de rechterkant van de schermen was rommelig en ongeordend. Dit waren

de eerste aanpassingen. De SFC's van het 'Doseren Grondstoffen' zijn netjes gegroepeerd per deelproces en de bedieningen staan nu in elk scherm op precies dezelfde plek zodat deze niet een stukje verspringen bij de overgang naar een ander scherm.

Hierna is er getest of alle bedieningen in het OS naar behoren werkten. Wordt alles juist aangestuurd? Is alles duidelijk voor de operator en worden er niet bepaalde aspecten onnodig moeilijk uitgevoerd? Hierbij viel als eerste op dat er vanuit WinCC niet goed gewerkt kon worden met de interlocks. Hoe dit is verbeterd is te lezen in paragraaf 6.3.2. Eén van de belangrijke onderdelen van het reactieproces is ook de verhouding waarmee de soda- en azijnzuuroplossingen gemengd worden. Dit wordt bereikt met een zelf geprogrammeerd block. De faceplate hiervoor is momenteel echter niet veel meer dan een dummy en doet vrijwel niks. Om alles goed te testen moet deze eerst werkend gemaakt worden. Dit is beschreven in paragraaf 6.3.3.

Hierna kon er begonnen worden met het testen van het proces. Hiervoor zijn alle SFC's in de juiste volgorde doorlopen. Hierbij bleek ook snel dat op plekken verbetering nodig was. Sommige aspecten waren niet duidelijk voor de operator, en bepaalde apparatuur die aangestuurd zou moeten worden werd niet aangestuurd. Ook trad er een deadlock op. Hier wordt uitgebreid op ingegaan in paragraaf 6.3.4. Ook kwam hierbij aan het licht dat er voor de regeling van het mengen van de soda- en azijnzuuroplossingen meer nodig was dan de eerder vermelde faceplate. Hier wordt dieper op ingegaan in paragraaf 6.3.5. Als laatste waren er pompen die wel op de juiste manier en op het goede moment aangestuurd werden door de SFC, maar kort daarna in een Error toestand kwamen. Dit is beschreven in paragraaf 6.3.6.

### 6.3.1. Opstarten

Aan het begin van het project waren er wat problemen om de OS en het model goed met elkaar te laten communiceren. Dit had te maken met de juiste opstartprocedure. Om te zorgen dat alles soepel verloopt moet eerst PLC-SIM gestart worden. Hierna moet de SIMATIC Manager geopend worden, en de software in PLC-SIM worden ingeladen. Nu moet het OS geactiveerd worden. Het is belangrijk dat dit gebeurt via de optie "Open Object", en niet met "Start OS Simulation". Met de laatste optie wordt een kopie van het OS gestart, waar bij eventuele aanpassingen niet direct zichtbaar zijn. Wanneer "Open object" wordt gebruikt opent WinCC explorer, vanuit dit programma is het mogelijk om het OS op te starten. Nu wordt Mobatec Modeller gestart. Over het programma OutsideOperator dat hierbij automatisch wordt opgestart, en de verschillende onderdelen van Mobatec Modeller, is meer informatie te vinden in hoofdstuk 8. Als laatste moet OPC scout worden gestart. Dan kan gebrowseed worden naar de variabele "simulatie", welke op "True" gezet wordt. Dit dient om het programma te laten weten dat alle signalen via een simulatie binnenkomen en niet van fysieke sensoren. Waarom deze opstartprocedure belangrijk is komt verder aan bod in het hoofdstuk over OPC: hoofdstuk 7. Na het opstarten zullen er een aantal alarmen en errors weergegeven worden in WinCC. Deze zijn gevolg van het feit dat het model bij het starten van de visualisatie nog niet online en verbonden was. Wanneer dit uiteindelijk wel is gebeurd is ook te zien dat de alarmen en errors voorbij zijn, en kunnen deze ge'acknowledge'd en verwijderd worden.



### 6.3.2. Interlocks

Als eerste waren er de interlocks. Onderdelen van het programma die iets aansturen in de fabriek (in dit geval het model), zijn vaak voorzien van een interlock. Dit kan bijvoorbeeld het geval zijn bij een klep, pomp, of mixer, maar ook bij andere apparatuur. Een interlock zorgt er voor dat er eerst aan bepaalde condities moet worden voldaan voordat het desbetreffende apparaat kan worden aangestuurd. Dit kan zeer nuttig zijn om onveilige situaties te voorkomen, of om te zorgen dat een bepaald deel van het proces niet kan starten tot een ander deel voorbij is. De interlocks zijn ook zichtbaar in de WinCC-visualisatie. Aan de kleuren is dan snel te zien welke status deze heeft. Rood betekent dat de interlock actief is en de bijbehorende apparatuur dus gelocked is, groen betekent dat de interlock inactief is. Bij deze simulatie komt ook vaak geel voor. Dit betekent dat de status van de interlock volledig wordt bepaald aan de hand van gesimuleerde waardes. Hierdoor is dan helaas niet direct duidelijk te zien of deze actief of inactief is.

Een nuttige functie van de interlocks is dat wordt weergegeven wat de toestand is van een bepaalde conditie, en een korte beschrijving wat deze conditie precies inhoudt. Dit kan bijvoorbeeld zijn dat er eerst een bepaalde andere klep open of dicht moet zijn voordat de huidige klep wordt aangestuurd, of dat de inhoud van een vat een bepaald niveau moet hebben voordat een mixer kan starten. Deze beschrijvingen moeten echter voor iedere interlock met de hand worden ingevuld in de blocks op het ES. Voorheen is dit niet gebeurd. Hierdoor kan de operator dus niet zien welke problemen er precies verholpen moeten worden voordat het apparaat start. Daarom heb ik voor alle interlocks de condities ingevuld. De ruimte om deze te omschrijven is echter wel zeer beperkt waardoor sommige omschrijvingen helaas nog steeds verwarrend of onduidelijk kunnen zijn. Dit is ook te zien in de afbeelding hieronder. Er wordt echter wel een indicatie gegeven van waar het probleem zich bevindt, waar dat voorheen niet het geval was.

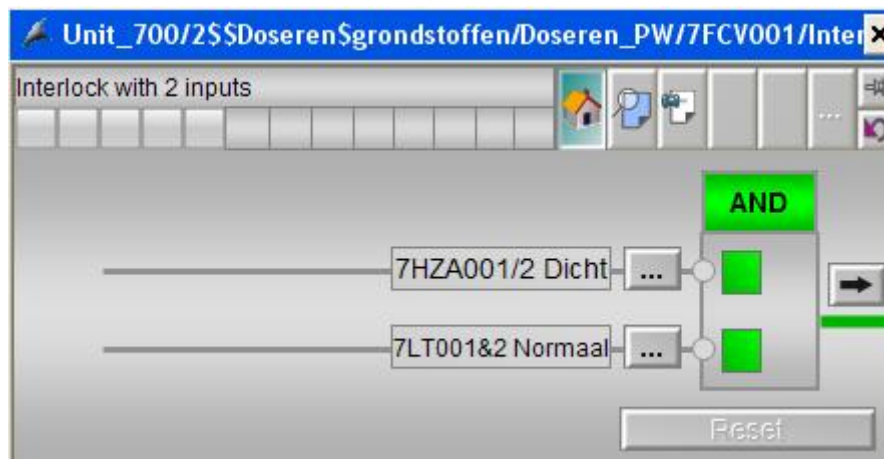
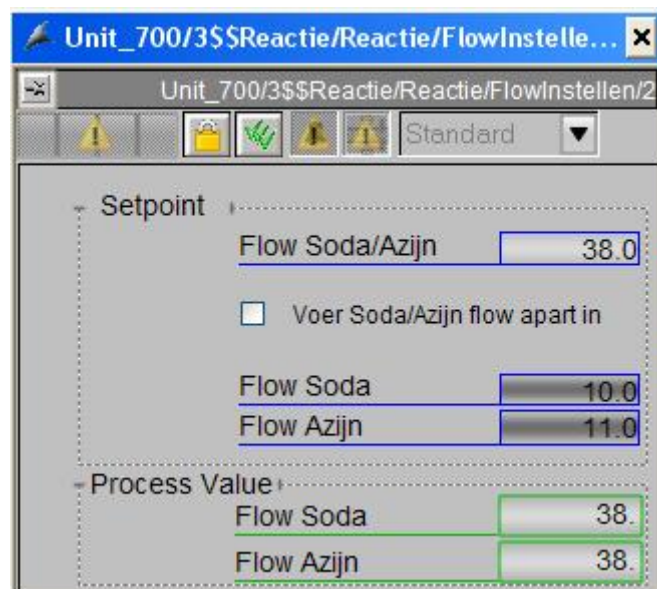


Fig. 11. Voorbeeld van een interlock in de visualisatie

### 6.3.3. Faceplate FlowInstellen

Veel 'faceplates', de vensters in het OS waarmee de aansturing plaats vindt, worden automatisch gegenereerd bij het compileren van het OS. Dit naar aanleiding van de gebruikte bouwstenen (blocks) in het programma. Aangezien blocks echter ook zelf gemaakt en geprogrammeerd kunnen worden, moeten

ook sommige faceplates zelf gemaakt worden in de WinCC Graphics Designer. Dit is ook het geval bij de faceplate voor het zelf ontworpen block 'FlowInstellen' in dit programma. Met dit block kunnen de gewenste hoeveelheden soda- en azijnzuuroplossing worden ingesteld, zodat deze in de juiste verhouding bij de reactor komen. De faceplate hiervoor bevatte een aantal errors waardoor de visualisatie niet juist was, en ook werden hierdoor de variabelen in het block niet goed aangepast.



**Fig. 12. Faceplate voor FlowInstellen**

De faceplate is te zien in Figuur 12. Het idee is dat in het blauwe veld 'Flow Soda/Azijn' een waarde ingevoerd kan worden waarmee de soda- en azijnzuuroplossingen een gelijke flow krijgen. Door echter het vakje eronder aan te vinken zou deze waarde voor beide oplossingen apart ingevuld kunnen worden. In de groene velden kan de huidige waarde van de flow worden afgelezen. De problemen in de faceplate waren dat alleen het eerste invoerveld werkte. Bij het aanvinken werd dat veld niet 'uitgegrijsd' maar wel inactief. De aparte invoervelden voor de soda- en azijnflow werden dan wel actief, maar iets hier invoeren had geen effect. De invoervelden zijn dus gerepareerd zodat ingevoerde waardes ook daadwerkelijk worden doorgevoerd in het programma. Ook is het aankruishokje werkend gemaakt waardoor op de juiste momenten de juiste invoervelden actief zijn, en dit ook duidelijk te zien is.

De faceplate voor FlowInstellen is hierbij een mix geworden van "klassieke"- en APL-onderdelen. De APL (Advanced Process Library) is een uitbreiding op PCS7 die zeer veel nieuwe functionaliteit biedt ten opzichte van de standaard libraries. Hierdoor kunnen veel standaard blokken en faceplates volledig vervangen worden. Er is echter nog geen functionaliteit om APL faceplates voor zelfgemaakte blocks te laten genereren. Het is uiteraard mogelijk om dit volledig zelf te programmeren, maar de APL is zeer complex en verschilt op veel punten van de standaard libraries. Het was daarom niet nuttig om de Faceplate voor FlowInstellen volledig in de APL te maken. Sommige oplossingen waren uiteindelijk eenvoudiger met de klassieke onderdelen. Wel zijn er stukjes van de APL gebruikt om de faceplate een moderner uiterlijk te geven.



### 6.3.4. SFC's

Hierna is begonnen met het testen van de SFC's (Sequential Function Charts). Met een SFC kan een volledig proces beschreven worden in een ladderdiagram. Het diagram bevat dan alle stappen die moeten worden doorlopen om het einde van het proces te bereiken, inclusief de aansturingen van de juiste kleppen, motoren, of andere apparatuur, en aan welke voorwaarden moet worden voldaan voordat de volgende stap begint. Het proces kan dan volledig worden doorlopen door de SFC te starten. Soms zijn er acties van een operator in de fabriek nodig om bijvoorbeeld handmatige kleppen open te zetten, of grondstoffen aan vaten toe te voegen.

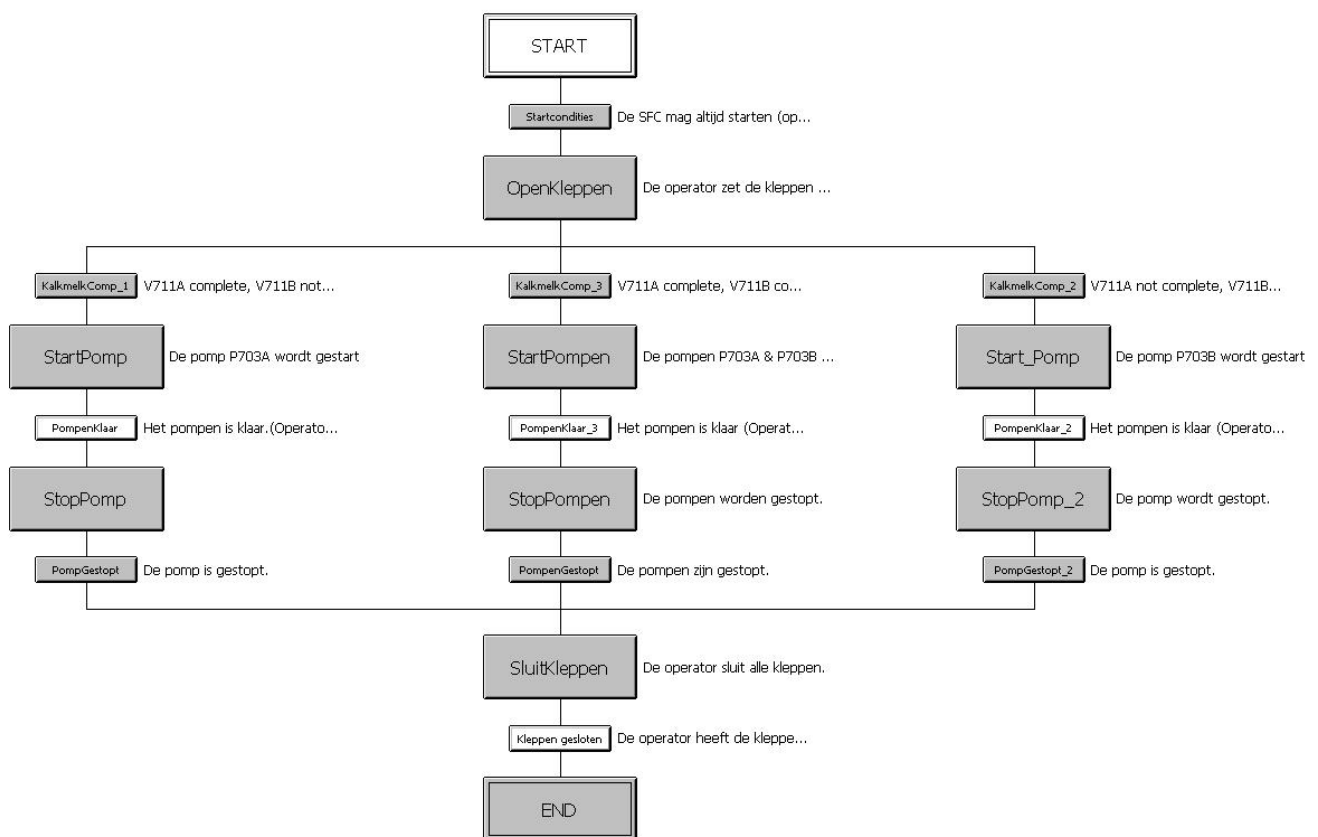


Fig. 13. Voorbeeld van een SFC

In Figuur 13 is een voorbeeld van een SFC te zien. Bij een vertakking met een enkele streep hoeft maar een van de takken te worden doorlopen voordat het gedeelte na de vertakking begint. Een vertakking met een dubbele streep betekent dat alle vertakkingen doorlopen moeten worden. Ook is het mogelijk om loops te maken. Deze laatste twee functies zijn helaas niet zichtbaar in de figuur. Bij de grotere blokken worden acties uitgevoerd. Dit kan bijvoorbeeld betekenen dat er een klep wordt geopend of een pomp wordt aangestuurd. De kleinere blokken beschrijven bepaalde condities. Wanneer hieraan wordt voldaan zal de SFC verder gaan naar de volgende stap. Een wit blok betekent dat hier geen acties of condities beschreven zijn. Het kan wel voorkomen dat hier een actie van de operator nodig is om het proces te vervolgen.

Bij het testen van de SFC's bleek dat deze niet volledig compleet waren. Bij meerdere SFC's werden bepaalde kleppen niet automatisch open gestuurd, waardoor de operator eerst moest constateren dat het



proces gestaakt was en waarom dat gebeurde. Daarna moesten de bijbehorende problemen nog opgespoord en handmatig opgelost worden. Dat werkt natuurlijk zeer inefficiënt en doet het nut van een SFC teniet. Daarom zijn, in de gevallen waar dit geen onveilige situaties creëert, deze kleppen toegevoegd aan de aansturing. Ook de voorwaarden dat deze kleppen in de automatische stand moeten staan zijn opgenomen in de startcondities van de desbetreffende SFC's. Wanneer een apparaat niet in de automatische stand staat kan deze niet worden aangestuurd door een SFC, maar alleen door de operator.

Een ander probleem was de koppeling tussen SFC's van het deel Doseren Grondstoffen en het Reactie gedeelte. Dit probleem ontstond omdat de SFC's vanuit Doseren Grondstoffen op een bepaald moment wachtten totdat de SFC van het Reactie gedeelte volledig doorlopen is. De SFC van Reactie deed echter hetzelfde; wachten totdat de SFC's van het Doseren Grondstoffen gedeelte volledig doorlopen waren. Dit resulteerde in een deadlock waar de aansturingen oneindig op elkaar stonden te wachten omdat geen van beiden volledig doorlopen kon worden. De SFC's van Doseren Grondstoffen hadden echter geen verdere acties meer geprogrammeerd staan na het wachten op de SFC van Reactie: Het wachten gebeurde in de laatste stap voor 'END'. Effectief gezien was het bijbehorende proces dus al afgerond. De deadlock is daarom opgelost door deze voorwaarde uit de SFC's van Doseren Grondstoffen te verwijderen.

Ook waren de startcondities van de SFC's niet duidelijk. De startconditie is vaak een lijst van apparatuur die in de automatische stand gezet moet worden alvorens de SFC start, zodat de SFC alle benodigde apparatuur kan aansturen. Bij veel condities stond er niet meer dan "ActuatorenAuto = True". Het was hierdoor niet duidelijk welke actuatoren het betrof, waardoor het voor de operator niet duidelijk is welke actuatoren er precies in de automatische stand moeten worden gezet. Dit kon alleen gecontroleerd worden door op de ES in het programma te kijken welke signalen er ware aangesloten. Ook dit is aangepast zodat in het OS juist wordt weergegeven welke actuatoren in de automatische stand moeten staan voordat de SFC kan starten. In de beschrijvingen staat nu niet meer "ActuatorenAuto = True" als conditie, maar bijvoorbeeld "P701\_Auto = True", zodat duidelijk is om welke apparatuur het gaat.

### 6.3.5. PID-controllers

Het zelf geprogrammeerde block FlowInstellen is gekoppeld aan twee PID-controllers die zorgen dat twee analoge kleppen op de juiste stand open staan om de flow van de soda- en de azijnzuuroplossing juist te verdelen. PID staat voor Proportional/Integral/Derivative. Dit zijn de drie in te stellen variabelen waarmee een PID controller de error op het ingestelde setpoint zo laag mogelijk probeert te houden. Wanneer deze variabelen juist ingesteld zijn zal in deze situatie de analoge klep zo worden open gestuurd dat er precies genoeg vloeistof doorheen stroomt. Deze gewenste waarde is het setpoint en kan intern worden ingesteld via WinCC, of extern via een ander block. In dit geval is het de bedoeling dat de flow bepaald wordt door het block FlowInstellen, en dat de PID-controllers het setpoint hieruit overnemen. Deze PID-controllers stonden echter ingesteld op een intern setpoint. Hierdoor werd de werking van het block FlowInstellen teniet gedaan. De PID-controllers zijn nu daarom standaard ingesteld om het externe setpoint te gebruiken. Zodat de flow eenvoudig en efficiënt in te stellen is via het block FlowInstellen.

### 6.3.6. Feedback signalen

Ook waren er twee pompen in het Absorptie-deel. Deze pompen, P-703A en P-703B, zorgen voor het rondpompen van de kalkmelk tussen de absorber en de filters waar het gips wordt uitgefilterd. Wanneer deze pompen werden aangestuurd gaven zij echter een "Feedback Error".

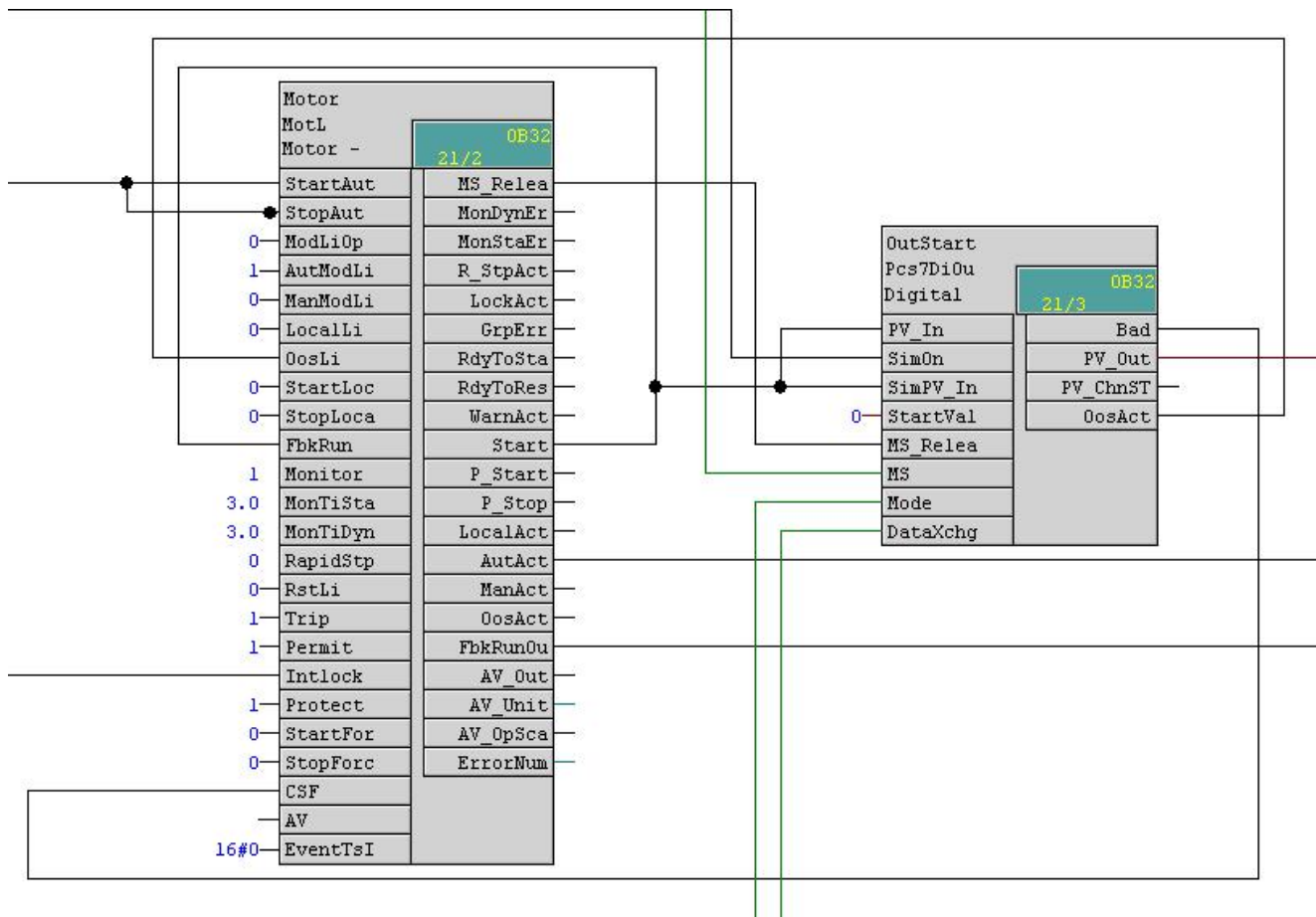


Fig. 14. Aansturing van een pomp

Dit kwam omdat het feedback signaal onjuist was aangesloten, waardoor de feedback van de pomp niet overeen kwam met de werkelijke situatie, zodat deze in een Error-toestand belandde. Normaal komt het feedback signaal terug vanuit de hardware, maar in het huidige model van Mobatec ontbreekt deze. Mogelijke heeft de werkelijke pomp ook geen feedback signaal. Om toch gebruik te kunnen maken van het feedback te signaal en errors te voorkomen, is deze nu gekoppeld aan de 'Start' uitgang.

## 7. OPC

Dit hoofdstuk zal zich richten op OPC, waarmee de communicatie tussen het OS en het model van Mobatec mogelijk wordt gemaakt. De eerste paragraaf is een globale omschrijving van OPC. Daarna volgt het deel dat zich specifiek op dit project richt.

### 7.1. Inleiding op OPC

OPC is een verzameling van een serie standaarden. In 1994 begon de OPC Foundation met de specificatie van de originele OPC standaard. Deze was in 1996 compleet. De OPC Foundation is daarna in stand gelaten om deze te onderhouden. Deze standaard beschreef manieren om control systemen van verschillende fabrikanten real-time met elkaar te laten communiceren. Sindsdien zijn er meerdere standaarden aan OPC toegevoegd. Vanaf 2006 noemt de OPC Foundation het een “serie van standaard specificaties”. Het ging toen om zeven volledig gespecificeerde standaarden en twee opkomende standaarden. Tegenwoordig is er nog een tiende standaard in opkomst, maar deze verschilt sterk van de andere negen, dit wordt verder uitgelegd in paragraaf 7.1.1. De allereerste standaard is nu bekend als OPC Data Access (DA).

Origineel stond de afkorting OPC voor Object Linking and Embedding (OLE) for Process Control. Tegenwoordig wordt OPC echter niet alleen maar meer gebruikt voor de proces industrie, maar bijvoorbeeld ook binnen de discrete productie industrie, waar op maat gemaakte producten worden gefabriceerd, in plaats van processen die altijd hetzelfde zijn. Omdat OPC daardoor tegenwoordig ook buiten de Process Control industrie bekend staat heeft de OPC Foundation besloten dat OPC niet langer een afkorting is, maar dat simpelweg “OPC” de naam is.

OPC is ontworpen om een gemeenschappelijke brug te zijn tussen software applicaties op een Windows platform en process control hardware. De standaarden definiëren een consistente methode om data uit te lezen van apparatuur in een fabriek. Deze methode blijft altijd hetzelfde, ongeacht het type data. De herkomst van deze data is ook niet belangrijk. Een OPC Server voor een bepaald stuk hardware biedt op dezelfde manier toegang tot zijn data voor een OPC Client als iedere andere OPC server voor datzelfde, maar ook ieder ander stuk hardware. Het doel was om de hoeveelheid werk, tussen hardware-fabrikanten met hun bijbehorende software-partners aan de ene kant, en SCADA- en HMI-fabrikanten aan de andere kant, die vereist is om hun toepassingen met elkaar te laten communiceren, te minimaliseren. Zo hoeft de hardware fabrikant alleen een OPC Server te verzorgen voor toegang tot de data op zijn apparaat, waar een SCADA fabrikant alleen een algemene OPC Client hoeft te programmeren om daar bij te kunnen. Sommige OPC standaarden zijn al flink uitgebreid of verbeterd sinds hun ontstaan. OPC DA zit bijvoorbeeld al op versie 3. Deze is aan de Serverkant echter backwards compatible waardoor een Client met OPC DA versie 1 ook gebruik kan maken van servers met versie 3. Andersom kan dit echter niet.

De specificaties voor de verschillende OPC standaarden zijn niet volledig open. Een deel is volledig gepubliceerd, terwijl de rest alleen beschikbaar is voor leden van de OPC Foundation. OPC is dus geen eigendom van een enkel bedrijf en elk willekeurig bedrijf kan en mag een OPC server ontwikkelen. Wanneer een bedrijf echter geen lid is van de OPC Foundation kan het voor komen dat hun Server of Client



niet gebruik maakt van de nieuwste specificaties, omdat deze op dat moment alleen nog voor leden beschikbaar zijn.

### **7.1.1. OPC Standaarden**

Hieronder worden de 10 huidige OPC standaarden beschreven.

#### **OPC Data Access (OPC DA)**

OPC DA is de eerst ontwikkelde standaard van de OPC Foundation, en daarmee het doel waarvoor de OPC Foundation oorspronkelijk was opgericht.

OPC DA is een standaard voor het communiceren van real-time data tussen data-acquisitie apparatuur zoals een PLC aan de ene kant en visualisatie- en besturingsinterfaces zoals HMI (Human-Machine Interface) apparatuur aan de andere kant. Deze specificatie richt zich op de continue communicatie van data.

De data voor OPC DA moet over drie attributen beschikken. Dit zijn ten eerste een waarde, ten tweede informatie over de kwaliteit van deze waarde, en als laatste een tijdstempel van de verkregen waarde.

#### **OPC Alarm and Events (OPC AE)**

Dit zijn specificaties waarmee informatie over alarmen of de huidige toestand van een apparaat op een bepaald moment opgevraagd kunnen worden, in tegenstelling tot de continue datastroom in de OPC DA specificatie.

#### **OPC Batch**

Specificaties om OPC geschikt te maken voor Batch processen.

#### **OPC Data eXchange (OPC DX)**

Specificaties om OPC Servers binnen een industrieel netwerk met elkaar te laten communiceren. Hierin worden ook de mogelijkheden tot configuratie, probleemoplossing, controle, en management communicatie, allen op afstand, beschreven.

#### **OPC Historical Data Acces**

Standaard voor toegang tot vooraf opgeslagen data.

#### **OPC Security**

Standaard voor het beperken van OPC Client toegang tot OPC Servers.

#### **OPC XML-DA**

Breidt OPC DA uit met de mogelijkheid om data in een XML formaat te communiceren.

#### **OPC Complex Data**

Specificaties voor het communiceren van complexere data types.

#### **OPC Commands**

Specificaties voor het besturen van apparaten en systemen via OPC.



### **OPC Unified Architecture (OPC UA)**

Deze specificatie verschilt sterk van de rest omdat het doel ervan is om de basis van OPC volledig te veranderen. OPC zou dan moeten draaien op een SOA (service-orientend architecture) protocol dat cross-platform is, in tegenstelling tot het Microsoft exclusieve COM/DCOM protocol waar het nu op gebaseerd is. Deze standaard zou alle bovenstaande standaarden moeten omvatten en biedt naast de cross-platform mogelijkheden ook verbeteringen op het gebied van beveiliging.

De OPC UA specificatie zou de volgende 13 onderdelen moeten gaan bevatten:

1. Concepts
2. Security Model
3. Address Space Model
4. Services
5. Information Model
6. Mappings
7. Profiles
8. Data Access
9. Alarms and Conditions
10. Programs
11. Historical Access
12. Discovery
13. Aggregates

Het zal dus ook meer gaan omvatten dan alleen de originele 9 OPC standaarden. Het werk aan OPC UA is begonnen in 2006, en in februari 2010 waren de specificaties voor het eerste onderdeel, Concepts, en het tweede onderdeel, Security Model klaar. In juli 2010 volgde de specificatie voor Address Space Model, en in oktober 2011 kwamen de specificaties voor Services, Information Model, Mappings, Profiles en Data Access uit. Hierna zijn er nog geen nieuwe specificaties gepubliceerd.



## 7.2. OPC binnen het project

In dit afstudeerproject verzorgt OPC de communicatie tussen het model van Mobatec en de visualisatie op het OS: WinCC. Omdat OPC voor alle fabrikanten geschikt is, gebruiken de modellen voor de andere Units van de oefenfabriek dit ook om te communiceren met hun desbetreffende SCADA applicatie. In dit geval draait er een OPC Server aan de WinCC kant, en een Client aan de kant van het model. Dit is niet zoals OPC oorspronkelijk bedoeld was, waar de server aan de kant van de hardware (nu dus het model) draait. Mobatec Modeller kan echter dienst doen zowel als OPC Server, als OPC Client. Deze configuratie is te zien in het onderstaande diagram.

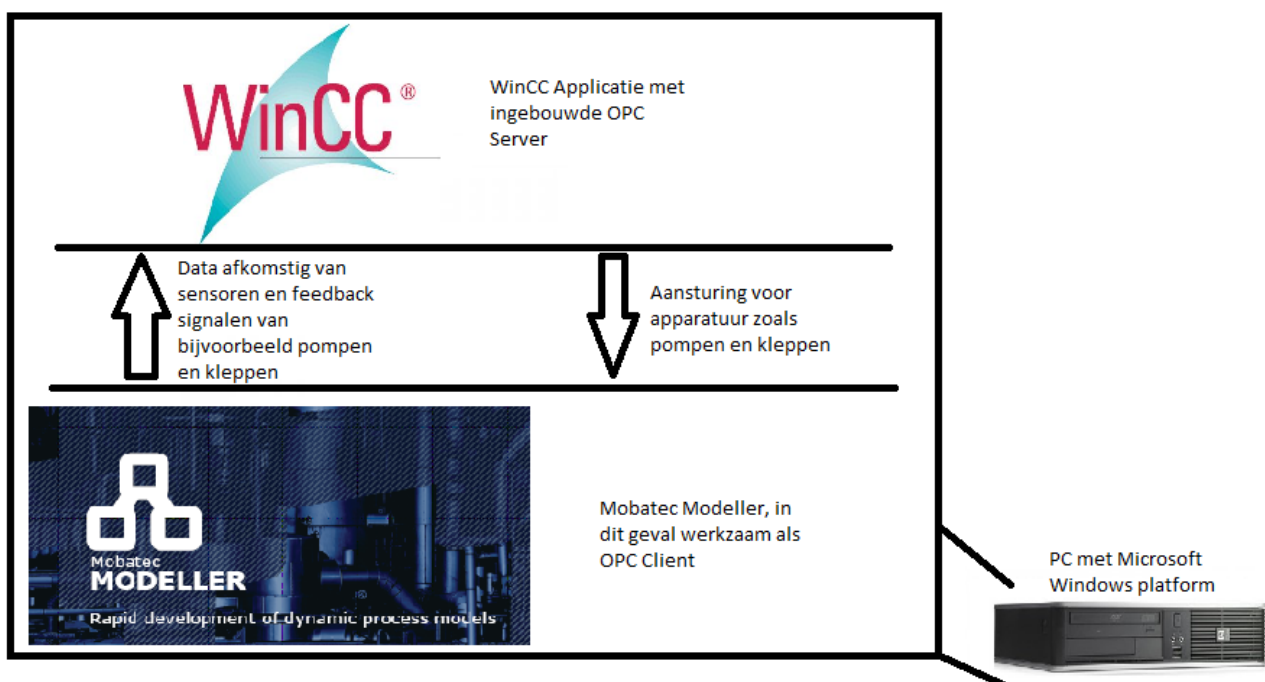


Fig. 15. OPC Configuratie

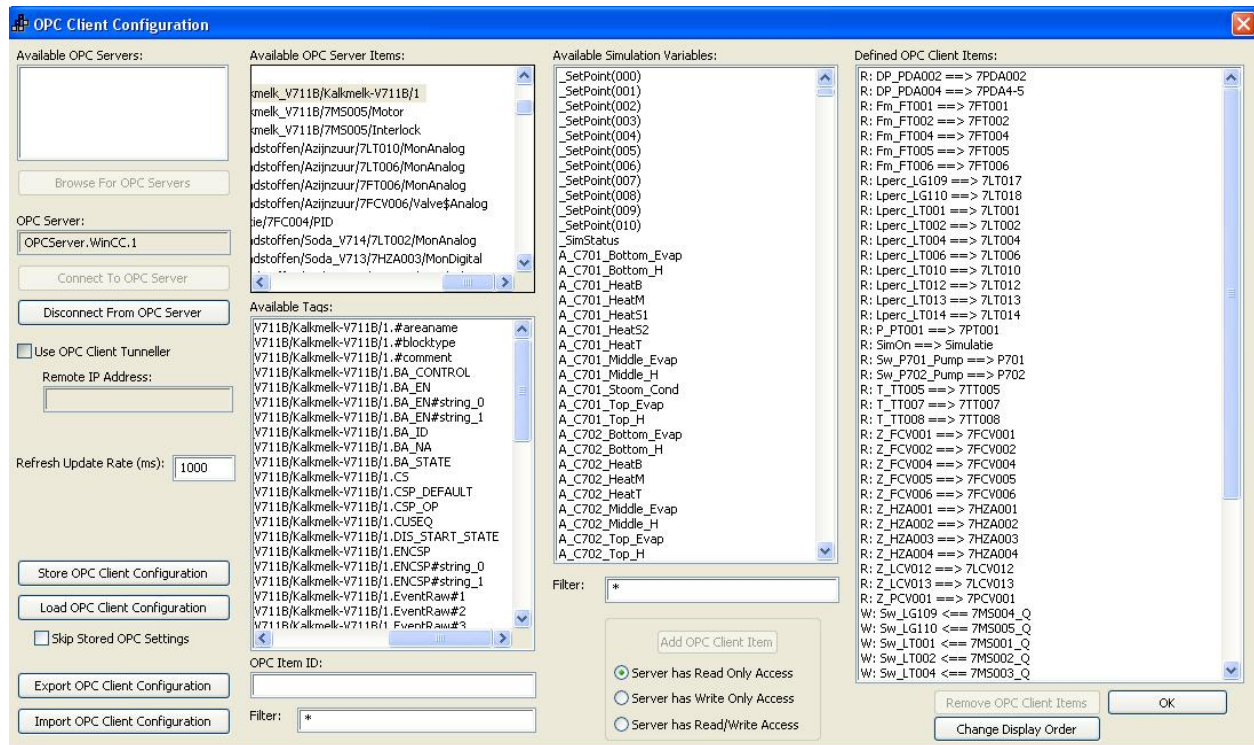
De opstartprocedure zoals beschreven in Paragraaf 6.3.1 is dus van belang omdat de server een onderdeel is van WinCC. De server moet eerst draaien, zodat de Client van Mobatec deze kan vinden en er verbinding mee kan maken.

De OPC Server van WinCC draait met de nieuwste specificatie van OPC DA, versie 3.00. Door de backwards compatibility functionaliteit zal de server dus met iedere andere OPC Client kunnen verbinden.

WinCC geeft via OPC alle mogelijke variabelen door aan Mobatec Modeller. In Mobatec worden deze vervolgens gekoppeld aan signalen of actuatoren uit het model. Al het verkeer dat plaatsvindt tussen de OPC Server en de OPC Client kan bekeken worden met het programma OPC Scout. Via OPC Scout kunnen ook waarden aan signalen worden gegeven, bijvoorbeeld voor situaties wanneer er een signaal uit de hardware of het model ontbreekt.



De eerder genoemde koppeling tussen Mobatec Modeller en de OPC Server was bij de eerste start nog maar zeer summier. Er waren slechts vier signalen door gekoppeld naar de juiste variabelen op de OPC server. Hieraan was wel duidelijk te zien hoe dit voor alle overige leesbare signalen en actuatoren moest gebeuren. Deze procedure is als volgt.



**Fig. 16. OPC Client configuratie**

Links is te zien welke OPC Server verbonden is met de Client. Onder 'Available OPC Server Items' zijn alle groepen met tags te vinden die de OPC server aanbiedt. Daaronder, bij 'Available Tags', zijn deze tags zelf te vinden na het kiezen van de gewenste groep. Dit zijn zowel inputs waar de Client data naar kan versturen, als outputs waar de Client data uit kan lezen. Bij 'Available Simulation Variables' staan alle virtuele sensoren en actuatoren die het model heeft. Na uit de lijst met Tags een Tag gekozen te hebben en uit de lijst met Simulation Variable een variabele, kan geklikt worden op 'Add OPC Client Item', waarna deze combinatie bij 'Defined OPC Client Items' terecht komt.

Uiteindelijk zijn alle leesbare signalen en schrijfbaar actuatoren op deze manier juist gekoppeld.



## 8. Mobatec Modeller

Dit hoofdstuk richt zich op Mobatec Modeller. Zoals bij het voorgaande hoofdstuk richt de eerste paragraaf zich op het programma in het algemeen, waarna bij de tweede paragraaf wordt ingegaan op deze specifieke toepassing van Mobatec Modeller.

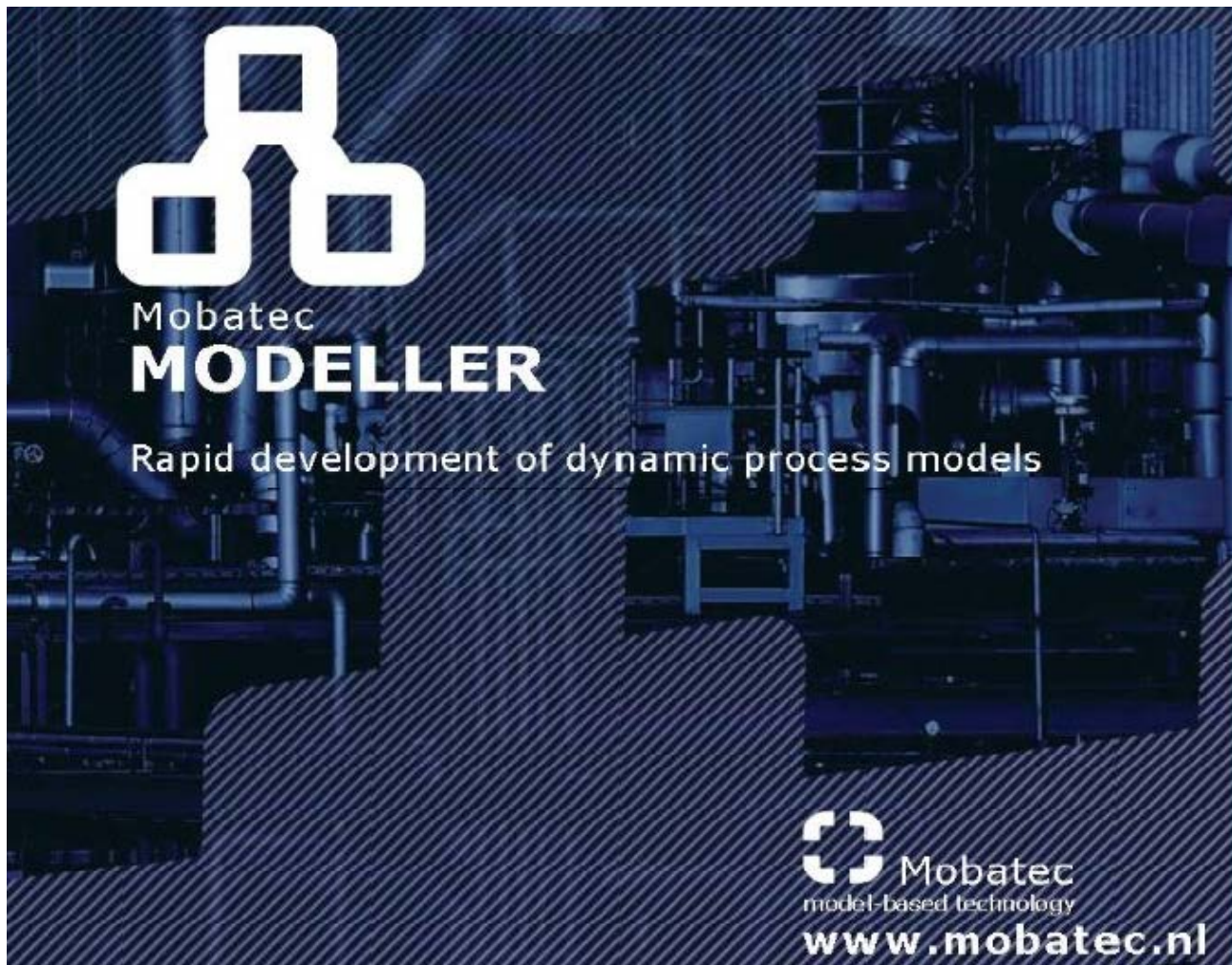


Fig. 17. Mobatec Modeller



## 8.1. *Inleiding op Mobatec Modeller*

Mobatec Modeller is software voor het bouwen van zowel dynamische als steady-state proces modellen van chemische, natuurkundige, en biologische processen. Deze kunnen elke gewenste omvang hebben, van bijvoorbeeld één chemische reactie tot een volledige chemische fabriek. Deze modellen kunnen zeer breed ingezet worden, bijvoorbeeld voor onderzoek, voorspellingen over het gedrag van een proces, of als simulator om operators op te leiden.

Mobatec Modeller is ontworpen om een model ontwikkelaar effectief te assisteren in het bouwen en onderhouden van consistente modellen. Het gebruik van Mobatec Modeller verkleint significant de benodigde tijd en moeite om een nauwkeurig model te ontwerpen.

De software wordt tegenwoordig al gebruikt voor het bouwen van zeer grote dynamische proces modellen voor complete fabrieken, waar meer dan 50.000 vergelijkingen gelijktijdig opgelost moeten worden. Om modellen van een dergelijke omvang werkbaar te maken biedt Mobatec Modeller een uitgebreide functionaliteit om tot automatische berekeningen en genereren van gegevens, en een uitgebreide zoekfunctie om specifieke onderdelen terug te vinden, hoe diep deze ook in het model zitten.

Het oplossen van proces engineering problemen zonder de hulp van geavanceerde software is tegenwoordig bijna onmogelijk. Proces simulatie, proces ontwerp, controller-ontwerp, controller-tests, data-acquisitie en het selecteren van de juiste apparatuur zijn maar enkele voorbeelden van problemen uit een chemische fabriek die opgelost moeten worden met dergelijke software.

Tegenwoordig is er genoeg software op de markt om dynamische proces modellen te ontwikkelen. Het lijkt echter vaak alsof er een expert nodig is om dergelijke modellen te bouwen en aan te passen. Vaak is het ook niet eenvoudig voor gebruikers die het model niet zelf ontwikkeld hebben om het model volledig te doorgronden. Mobatec Modeller is software voor het interactief definiëren en transparant aanpassen van proces modellen. Het kan ook gezien worden als een schil om bestaande dynamische modelleer-pakketten heen. Het is gericht op het effectief assisteren in de ontwikkeling van dergelijke proces-modellen en het faciliteren van een hiërarchisch model via een gebruiksvriendelijke interface.

Mobatec maakt gebruik van simpele basis bouwstenen om proces-modellen te bouwen. Veel andere modelleer-pakketten gebruiken hogere modellen van volledige onderdelen die minder breed toepasbaar zijn. In Mobatec Modeller bestaan dergelijke modellen van volledige onderdelen ook uit de basis bouwstenen. Mobatec Modeller genereert symbolische modellen in de vorm van differentiaalvergelijkingen volledig gebaseerd op natuurkundige wetten. Door een proces op een dergelijke manier te reduceren tot de meest simplistische basiscomponenten wordt het model zeer gedetailleerd en nauwkeurig. Een model bestaat zo uit de 'Physical Topology', waarin de omgeving en de apparatuur beschreven wordt, de 'Species Topology' waarin wordt beschreven welke scheikundige en/of biologische elementen zich in het model bevinden en hoe deze met elkaar reageren. En als laatste de 'Equation Topology' waarin natuurkundige wetten worden beschreven en hoe deze de eerste twee topologieën beïnvloeden.

Een probleem dat vaak de gebruiksvriendelijkheid in de weg staat is dat dergelijke software niet ontworpen wordt door de mensen die het uiteindelijk gaan gebruiken. Een programmeur bevindt zich vaak niet in het zelfde vakgebied als iemand uit de chemische industrie, waardoor het niet eenvoudig is om zich in te leven en het programma te laten werken zoals de eindgebruiker dat vaak ziet. In een poging dit probleem op te



lossen is Mobatec Modeller echter ontworpen door programmeurs die oorspronkelijk uit de industriële wereld afkomstig zijn. Immers, de beste manier om te weten wat de gebruiker wil, is om zelf ook een gebruiker te zijn.

Veel andere modellerings-software is ook veel meer gericht op het oplossen van de problemen en vergelijkingen. Het wiskundige deel van het model is dan helemaal in orde, maar het bouwen van het model zelf is dan vaak onhandig geïmplementeerd. Mobatec Modeller richt zich dan ook meer op het grafisch ontwikkelen van modellen en het automatisch generen van belangrijke basis onderdelen.

Dit zijn de manieren waarop Mobatec probeert haar software te onderscheiden en te verbeteren ten op zichte van andere modellerings-software.



## 8.2. *Het model voor Unit 700*

Voor de simulatie van Unit 700 wordt dus ook een model van Mobatec Modeller gebruikt. Zoals uitgelegd in een eerder hoofdstuk communiceert dit model met het OS via OPC. In dit geval is Mobatec Modeller de Client, wat niet gebruikelijk is in OPC. Mobatec bevat daarom ook de mogelijkheid voor een OPC server, maar dat wordt in dit geval niet gebruikt.

### OutsideOperator

Zoals ook eerder ter sprake kwam heeft Unit 700 ook een aantal handafsluiters; handmatige kleppen die op het apparaat zelf bediend moeten worden en niet aangestuurd kunnen worden vanuit het OS. In het geval van de simulatie is dit opgelost met het programma OutsideOperator van ExplainMedia.

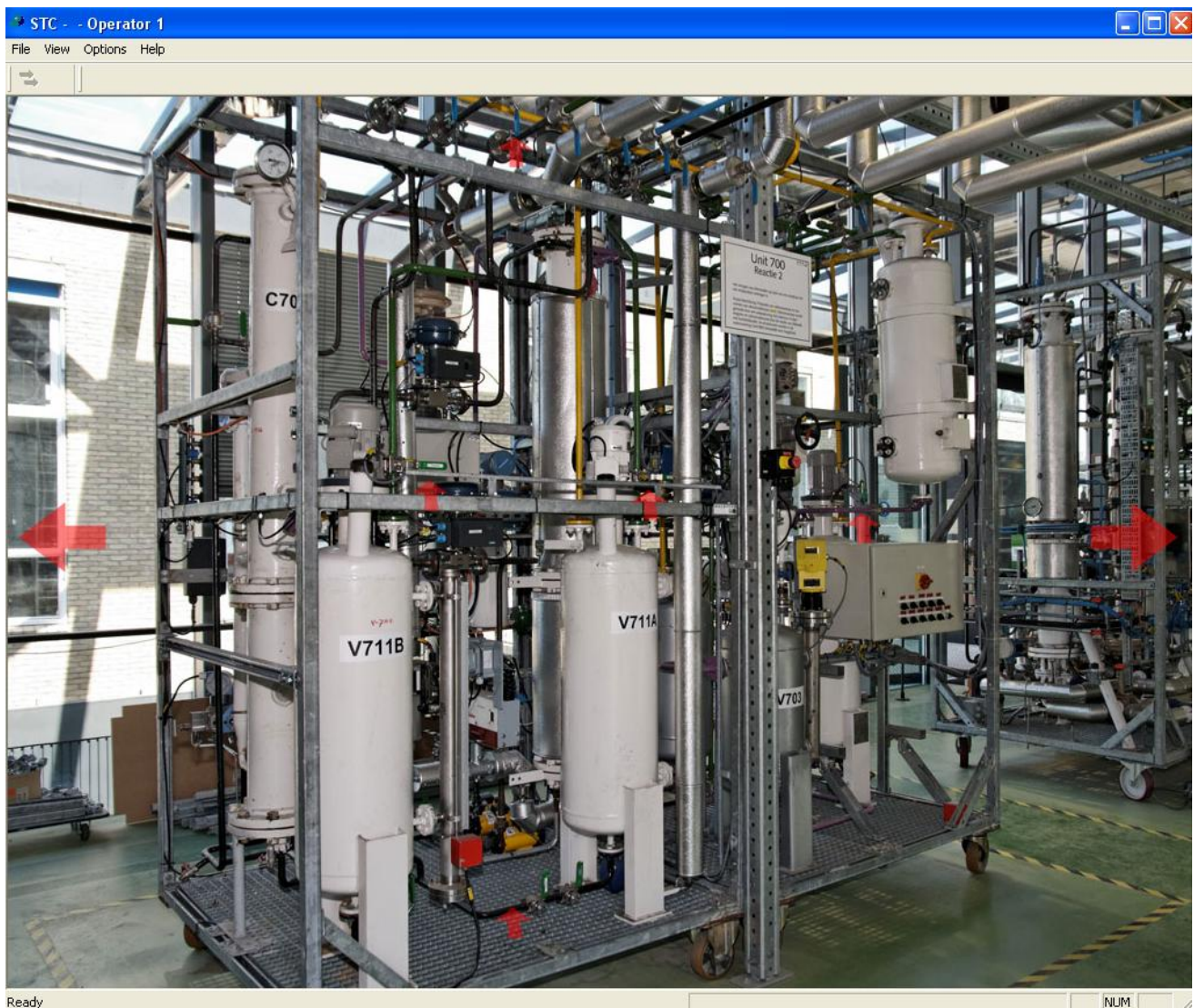


Fig. 18. OutsideOperator

OutsideOperator bevat zes aanzichten van de skid van Unit 700. Op Figuur 18 is één van deze aanzichten te zien. Met de rode pijlen die zich links en rechts aan de zijkant bevinden, deze zijn ook zichtbaar in Figuur 18, kunnen de andere aanzichten bereikt worden, waardoor men als het ware volledig om de skid heen kan lopen. Met de andere rode pijlen in het aanzicht kunnen dan de diverse handafsluiters worden bereikt. Wanneer deze bediend worden reageert het model hierop. Hierna stuurt het model via OPC de informatie naar het OS.

Een andere methode voor het bedienen van deze handafsluiters is via knoppen die zich in het model van Unit 700 bevinden. Wanneer het model echter in de simulatie stand wordt gezet en daadwerkelijk de berekeningen uitvoert, wordt het modelvenster verstoppt en zijn deze knoppen niet meer toegankelijk. Meer informatie over het verstoppen van het modelvenster en een probleem dat hierbij optreedt, volgt later.

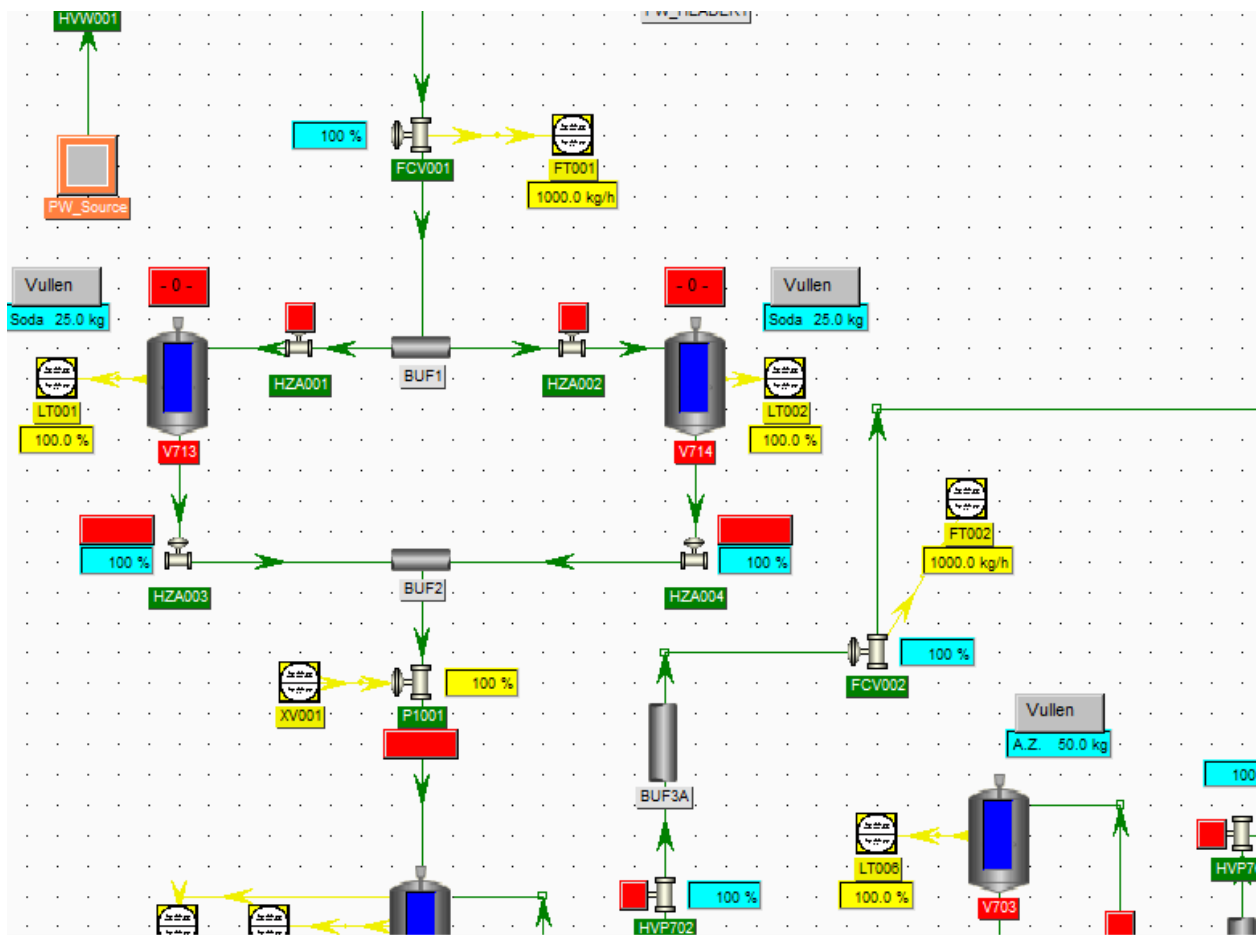


Fig. 19. Een deel van het model van Unit 700

Hierboven is een deel van het model te zien. In het model zijn naast alle sensoren en apparatuur die in het OS zichtbaar zijn ook de handafsluiters te zien. Ook kan door te klikken op bijvoorbeeld een vat meer informatie verkregen worden over de 'werking' van het vat, zoals bijvoorbeeld invloeden van de omgeving op de temperatuur of de druk in het vat.





In het model kunnen de (automatische) kleppen en pompen ook bediend worden door op de rode knoppen te klikken (deze zijn groen als de kleppen open staan, of wanneer een pomp aan staat). Dit is echter niet de bedoeling en door de eenzijdige communicatiestructuur van OPC kunnen hierdoor (feedback-) errors ontstaan op het OS of in de programmering. Dit is waarschijnlijk de reden dat het modelvenster verstopt wordt wanneer de simulatie wordt gestart en de berekeningen uitgevoerd worden. Het venster kan dan weer teruggehaald worden door een wachtwoord in te voeren.

Dit verstopten brengt zoals eerder vermeld echter een probleem met zich mee. Zoals te zien in Figuur 19 bevat het model ook knoppen om te 'vullen'. Dit is bijvoorbeeld om soda aan de vaten toe te voegen, wat bij de fysieke Unit 700 gebeurt via een klep bovenin het vat. Logisch gezien zou dit bij deze simulatie, net als de handafsluiters, via OutsideOperator moeten gebeuren. OutsideOperator blijkt echter niet de functionaliteit hiervoor te bevatten, waardoor dit moet gebeuren in het model. Dit kan echter niet wanneer het model verstopt is. In het OS bevindt zich echter ook de mogelijkheid om het 'vullen' te laten bevestigen door de operator, ook wanneer dit nog niet daadwerkelijk gebeurd is. Het OS heeft verder ook geen manier om dit te controleren. Hierdoor kan effectief gezien het vullen overgeslagen worden, maar dit is natuurlijk geen juiste simulatie van de werkelijkheid. Vooralsnog zijn er helaas echter geen plannen om OutsideOperator hier toe op aan te passen en moet het vullen, indien gewenst, via het model gebeuren.

Om het model volledig juist te laten werken moeten een aantal zaken aan de fysieke apparatuur uitgemeten worden. Deze apparatuur is echter niet compleet. Sommige sensoren of actuatoren ontbreken, terwijl andere niet goed aangesloten zijn. Omdat de oefenfabriek zich in een overgang bevindt naar één proces control systeem, worden deze problemen pas opgelost wanneer dat systeem gekozen is. De benodigde metingen voor het model kunnen daardoor helaas niet worden uitgevoerd.

Een aantal werkingen van het model zijn hierdoor niet correct en belemmeren helaas ook het testen van het PCS7 programma. Sommige dingen had ik zelf willen aanpassen, maar de versie van Mobatec Modeller op de PC van het STC had geen licentie voor de modelleer omgeving, alleen voor de simulatie. CoNet heeft echter veel interesse in het gebruik en de toepassing van Mobatec Modeller in toekomstige projecten. Daarom is er later tijdens mijn stage ook een vertegenwoordiger van Mobatec bij CoNet langsgekomen om uit te leggen wat de mogelijkheden hiervoor precies zijn. Uiteindelijk is er een evaluatie-licentie verkregen om het volledige pakket van Mobatec te kunnen gebruiken tot augustus 2012, zodat CoNet het pakket kan testen en uitproberen, en om zo te beoordelen of het gebruik ervan echt interessant is voor toekomstige projecten.

Deze evaluatie-licentie kwam echter in een dergelijk laat stadium van de stage dat ik, na het leren kennen van de modelleer-omgeving, niet meer de gelegenheid had echt belangrijke aanpassingen te verrichten in het model. Het leren werken met de modelleer-omgeving was naar mijn mening minder eenvoudig dan men van Mobatec het tracht te doen voor komen. Op de volgende pagina volgen een aantal problemen die ik tegen ben gekomen. Deze had ik nog willen aanpassen in het model.



Als eerste is de toevoer van proceswater nogal onrealistisch. In een reële situatie zou het water normaal naar binnen stromen, en door middel van de flow-meter en het doseer-block in de software nauwkeurig gedoseerd kunnen worden. In het model komt het water echter zodra de klep open gaat met honderden (virtuele) liters binnen waardoor doseren niet mogelijk is. De niveaumeters op de vaten zijn ook niet juist afgesteld: Ook al komen er in één keer honderden (virtuele) liters binnen, toch neemt het niveau in de vaten slechts met enkele procenten toe, terwijl deze vaten in werkelijkheid maar 50 liter kunnen bevatten.

Een ander probleem doet zich voor in het azijnzuur deel. Het azijnzuur wordt vanuit een vat gepompt naar een opslagvat waar het gemengd wordt met water. Omdat hier echter geen flow-meter is, ook niet in de echte hardware, wordt de hoeveelheid gedoseerde azijnzuur 'gemeten' aan de hand van de gradient, een variabele in de niveaumeter van het vat die het verschil in vloeistofniveau per cyclus door het programma aangeeft. In theorie zou dit werken, echter het model zorgt ervoor dat het in dit geval niet werkt. Het model simuleert namelijk een 'realistische' schommeling van het vloeistofniveau in rust. Hierdoor is de gradien- waarde ook constant onderhevig aan verandering, waardoor de gedoseerde hoeveelheid onjuist berekend wordt.

Ook het verpompen van de soda- en azijnzuuroplossing verloopt niet zonder problemen. Het gebeurt wel, maar de stroom is erg traag en schommelend en in sommige gevallen zou volgens de sensoren de stroom van de azijnzuuroplossing zelfs negatief zijn.

Het laatste (grote) probleem is dat wanneer de kalkmelk-vaten via het model gevuld worden met kalkmelk, in de realiteit gaat dit net als bij de soda met een klep aan de bovenkant, het niveau in het vat daalt in plaats van stijgt. Het toevoegen van nog meer kalkmelk resulteert uiteindelijk in een leeg vat.

## 9. Conclusie

De oefenfabriek van het STC is een zeer interessante locatie die veel mogelijkheden biedt voor studenten. Het is jammer dat dit gemigreerd wordt naar één enkel DCS systeem. Vanuit financieel oogpunt is dit echter zeer begrijpelijk. Om te bepalen of de resultaten naar tevredenheid behaald zijn kan nogmaals gekeken worden naar de MoSCoW analyse.

### Must

#### - Een volledig werkende en geteste PCS7 aansturing.

De PCS7 programmering voor de simulatie is nu compleet, maar kan echter niet volledig naar behoren worden getest, omdat het model niet correct werkt. Helaas wordt dit niet binnen afzienbare tijd opgelost. Pas wanneer de migratie van de oefenfabriek compleet is kunnen deze problemen worden verholpen.

#### - Een volledig werkende en geteste OPC communicatie.

De OPC communicatie is nu volledig en alle virtuele sensoren uit het model sturen hun informatie naar de juiste inputs van het programma. Het programma stuurt ook weer via OPC via haar outputs de juiste actuatoren aan.

### Should

#### - Een volledig werkend en getest Mobatec model.

Het model is getest en werkt. De werking is echter niet naar behoren en de bij het testen gevonden problemen zijn gedocumenteerd. Deze zijn niet verholpen doordat het leertraject van Mobatec Modeller langer dan verwacht was.

#### - Een volledig werkende en geteste Process Safety implementatie in de PCS7-software.

Door tijdsgebrek is hier geen werk aan verricht. De PCS7-software heeft dit echter niet nodig om werkzaam te zijn en een goede, complete aansturing te verzorgen. Het zou alleen extra veiligheid kunnen toevoegen.

### Could

#### - Testrapport van aanwezige apparatuur en Profibus-netwerk op fysieke Unit 700.

Een 'Could' doel gebeurt alleen wanneer er tijd voor vrij is. Hier was geen tijd voor.

### Won't

#### - Uitbreiding met SIMATIC Batch

Dit is nog steeds wenselijk in de toekomst.

De doelen bij 'Must' zijn behaald voor zover dat kon binnen dit project. In verband met het incomplete model kon alleen de aansturing niet naar behoren getest worden.

Het eerste doel bij 'Should' is deels behaald omdat het model wel uitvoerig is getest, en de gevonden problemen in dit document gerapporteerd staan.



Het is wel jammer dat het tweede doel bij 'Should', de Process Safety, uiteindelijk niet aan bod is gekomen in dit project.

Hieruit kan de conclusie getrokken worden dat de gewenste resultaten, behalve Process Safety, waar mogelijk binnen de huidige situatie van de oefenfabriek, zijn behaald.

Het gebruik van Mobatec Modeller in combinatie met OutsideOperator is zeer interessant en een mooie oplossing voor de simulatie. Het zou voor CoNet zeker de moeite waard kunnen zijn om dit pakket grondig uit te proberen. Het vergt echter wel wat meer moeite om te leren dan de mensen van Mobatec de gebruiker proberen te laten geloven.



## 10. Evaluatie

Ik vond dit een zeer interessante en leerzame stage. Als eerste de kennismaking met PCS7 en de cursus van Siemens. Het vak dat ik op de HHS in een keuz- module had gevolgd over PLC's was één van de interessantste vakken van de opleiding naar mijn mening, wat ook een van de redenen is dat ik graag deze stage wilde gaan doen. Achteraf gezien denk ik daar nog steeds zo over en blij te hebben leren werken met het SIMATIC PCS7 pakket.

Het gebruik van OPC was ook nieuw voor mij en een leuke en leerzame ervaring.

Met een pakket als Mobatec Modeller had ik nog nooit gewerkt. Ook niet met andere vergelijkbare simulatie pakketten. Ik vond het erg leuk en interessant om mee te werken, en ook de uiteindelijke toevoeging van de modelleer omgeving was belangwekkend. Op het eerste gezicht leek het echter allemaal wel zeer complex, maar na een tijdje de tutorials doorlopen te hebben werd het al een stuk duidelijker.

Het was erg jammer dat door de migratie van de oefenfabriek mijn stage wat vertraging heeft opgelopen. Eerst was er het feit dat mijn opdracht aangepast moest worden naar het werken met de simulatie. Later volgde het probleem dat het model niet correct werkte, wat niet opgelost kon worden, wederom door de reorganisatie van het STC. Desondanks heb ik het idee dat mijn tijd over het algemeen goed besteed is.

Uiteindelijk kan ik concluderen dat ik de stage met plezier heb doorlopen, wat te danken is aan zowel de kwaliteit van de opdracht, het stagebedrijf CoNet, als de collega's bij CoNet.



## BIJLAGE I Afkortingen

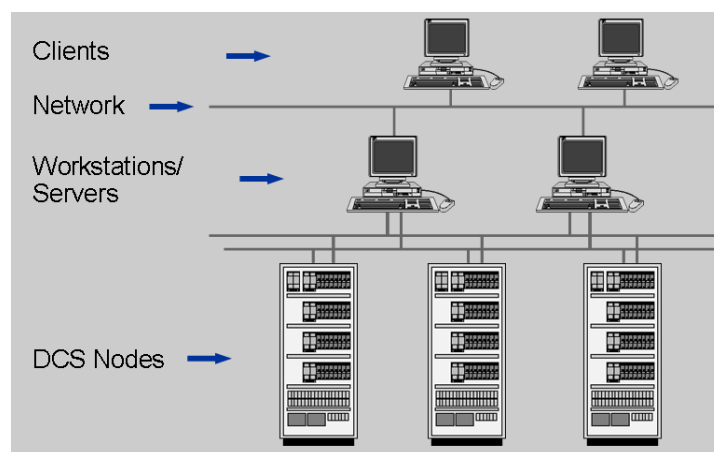
Afkorting	Betekenis	Locatie in document
STC	Scheepvaart en Transport College	Pagina 3
DCS	Distributed Control System	Pagina 6/BIJLAGE II
MES	Manufacturing Execution Systems	Pagina 19
ERP	Enterprise Resource Planning	Pagina 19
AS	Automation System	Pagina 17
ES	Engineering Station	Pagina 18
OS	Operator System	Pagina 17
Profibus DP	Decentralized Peripherals	Pagina 20
Profibus PA	Process Automation	Pagina 20
PDM	Proces Device Manager	Pagina 20
HART	Highway Addressable Remote Transducer	Pagina 21
APL	Advanced Process Library	Pagina 29
SFC	Sequential Function Charts	Pagina 30
PID	Proportional/Integral/Derivative	Pagina 31
OPC	Object Linking and Embedding (OLE) for Process Control	Pagina 33



## BIJLAGE II Distributed Control Systems

In de industriële sectoren wordt een DCS (Distributed Control System) gebruikt om de processen te volgen, te controleren en te beheren. Door de jaren heen zijn de DCS-systemen steeds meer ontwikkeld en zijn koppelingen naar de MES- en ERP-systemen mogelijk gemaakt. De opbouw is redelijk hetzelfde gebleven.

Een DCS bestaat meestal uit één of meerdere visualisatie-clients en een visualisatie-server, DCS controllers (DCS Nodes), de I/O (Input/Output) en één of meerdere workstations om alles te programmeren, het engineerings-systeem. Met het visualisatiegedeelte kunnen de processen in een fabriek worden gevolgd en worden gemanipuleerd. Bij Siemens wordt er dan gesproken van een OS (Operator Station). In de controller lopen de software-routines af. Bij Siemens is dit een SIMATIC PLC, deze wordt bij SIMATIC PCS7 de AS (Automation System) genoemd. Om deze processen te kunnen regelen zijn er I/O kaarten aangesloten. Bij de meeste DCS-installaties is er een systeem om het volledige DCS te programmeren. Deze is bij sommige systemen samengevoegd in de OS. Bij Siemens is dit meestal een workstation de ES (Engineering Station).



### Verschil met PCS

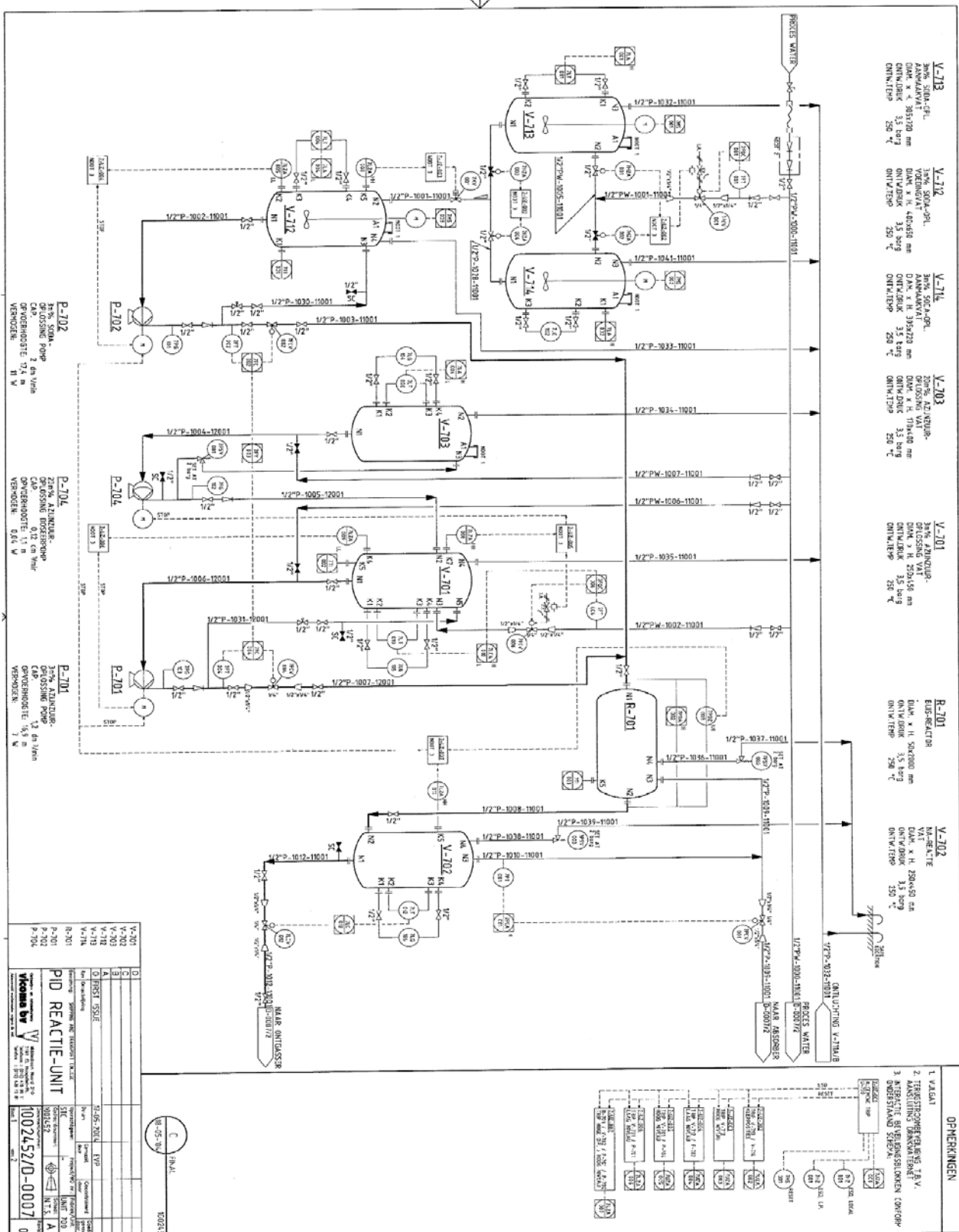
Een Process Control System (PCS) is een automatiseringssysteem dat speciaal is toegerust voor het besturen van industriële processen in de procestechniek. Vroeger stonden dergelijke systemen ook wel bekend onder de naam DCS, omdat de componenten van een dergelijk systeem veelal verschillende functies hebben en door één of ander communicatienetwerk met elkaar verbonden zijn. De controllers waren toen gedistribueerd, die stonden in het veld. Dit werd gedaan om de controller dicht bij de I/O te hebben en zo kabel-fouten tegen te gaan. Wanneer de systeembus uitviel, kon de controller in het veld doorwerken. Vandaar dat zij distributed control systems werden genoemd. Latere systemen kregen er steeds meer functionaliteiten bij zoals een redundante uitvoering van de systeembus (ring of zelfs een redundante ring) en een busstructuur voor het I/O gedeelte. Hierdoor was het niet meer nodig de controllers gedistribueerd te plaatsen. Deze worden steeds vaker in één ruimte centraal geplaatst. Bij moderne systemen wordt dan ook steeds vaker over een PCS gesproken. Dit is omdat het distribueren van de controllers niet meer nodig is. Daarom is het mogelijk om te zeggen dat een PCS een moderne vorm is van een DCS.

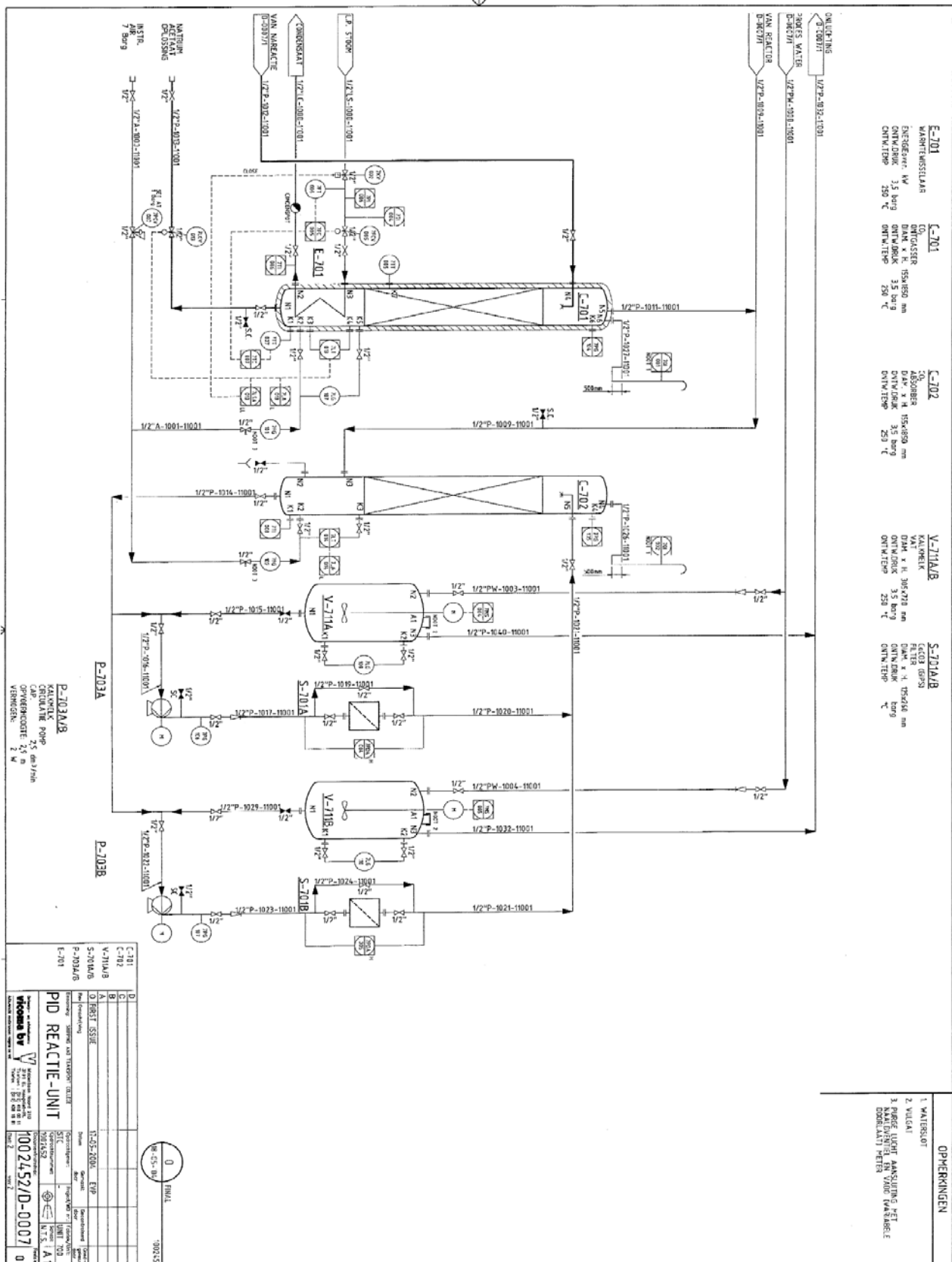
De termen DCS en PCS worden vaak door elkaar gebruikt, meestal met dezelfde betekenis: een systeem om de industriële processen te controleren en te regelen.



**CoNet Engineering B.V.**  
van Nelleweg 1  
3044 BC Rotterdam  
Postbus 13030  
3004 HA Rotterdam  
T: 075-612 16 24  
F: 075-612 16 72

## BIJLAGE III Tekeningen Unit 700







---

## BIJLAGE IV Bronnenlijst

### Rapporten:

- FS\_STC\_v05 -Functionele Specificatie van de fysieke opstelling Unit 700, aanwezig bij CoNet.  
DS\_v02 -Ontwerp (Design) Specificatie van de fysieke opstelling Unit 700, aanwezig bij CoNet.

### Naslagwerken:

PCS7 Systeemcursus cursusmap.

### Websites:

- <http://www.siemens.nl/dcs>  
<http://www.mobatec.nl>  
<http://www.conet.nl>  
[http://en.wikipedia.org/wiki/Distributed\\_Control\\_System](http://en.wikipedia.org/wiki/Distributed_Control_System)  
[http://en.wikipedia.org/wiki/OLE\\_for\\_process\\_control](http://en.wikipedia.org/wiki/OLE_for_process_control)  
[http://en.wikipedia.org/wiki/OPC\\_Foundation](http://en.wikipedia.org/wiki/OPC_Foundation)