



Bachelorarbeit

Modulare Beschreibung einer Verfahrenstechnischen Anlage

Verfasser: Patrick Holstein
Betreuer: Prof. Dr.-Ing. Birgit Vogel-Heuser
Jakob Kinghorst
Beginndatum: 25 April 2017
Abgabefrist: 24 Oktober 2017

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Abschlussarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: _____ Unterschrift: _____

Kurzzusammenfassung

Ziel der Bachelorarbeit ist die prototypische modulare Inbetriebnahme einer Laboranlage im Rahmen des Projektes Process Application Composer (ProcAppCom). In diesem Projekt wird erforscht, wie Sudhäuser möglichst modular in Betrieb genommen werden können, um das Engineering durch die Wiederverwendung vom Steuerungscode, der Benutzeroberfläche und des Manufacturing Execution Systems (MES), zu erleichtern. Im Rahmen dieser Bachelorarbeit wird eine Laboranlage, welche die Würzeherstellung in einem Zwei-Behälter-Sudhauses simuliert, in verschiedene Module unterteilt, um anschließend modular programmiert zu werden. Dabei sollen die Modellierungsansätze und die Programmierung mit Steuerungsbausteinen aus ProcAppCom evaluiert und getestet werden.

Abstract

The aim of this Bachelor thesis is the prototypical and modular commissioning of a laboratory facility within the scope of the project Process Application Composer (ProcAppCom). In this project, research will be carried out on how brewhouses can be put into operation as modular as possible, in order to facilitate engineering by reusing the control code, the user interface and the Manufacturing Execution System (MES). Within the framework of this Bachelor thesis, a laboratory system, which simulates the wort production in a two-vessel brewhouse, is subdivided into various modules in order to be subsequently programmed in a modular form. The modeling approaches and programming with control modules from ProcAppCom are to be evaluated and tested.

Inhaltsverzeichnis

Eidesstattliche Erklärung	I
Kurzzusammenfassung	I
Abbildungsverzeichnis	IX
Tabellenverzeichnis	XI
Abkürzungen	XIII
1 Einleitung und Zielsetzung	1
1.1 Ausgangssituation	1
1.2 Problemstellung	1
1.3 Ziel der Bachelorarbeit	2
1.4 Aufbau der Arbeit	3
1.5 Application Composer	4
2 Aktueller Stand der Technik	5
2.1 Was bedeutet Modularisierung	5
2.1.1 Vor- und Nachteile der Modularisierung	6
2.2 Automatisierung von Chargenprozessen	7
2.2.1 Chargenprozesse	7
2.2.2 Prozessmodell	8
2.2.2.1 Prozessabschnitte	9
2.2.2.2 Prozessoperationen	9
2.2.2.3 Prozessschritte	10
2.2.3 Anlagenmodell	10
2.2.3.1 Anlage	11
2.2.3.2 Teilanlage	12
2.2.3.3 Technische Einrichtung	13
2.2.3.4 Einzelsteuereinheit	14

2.2.4	Prozedurmodell	15
2.2.4.1	Prozedur	15
2.2.4.2	Teilprozedur	17
2.2.4.3	Operation	17
2.2.4.4	Funktion	17
2.2.5	Rezepte	18
2.2.6	Betriebsarten	18
2.2.7	Betriebszustände	19
2.3	Schnittstellen	19
3	Aufbau der Laboranlage	21
3.1	Beschreibung der Laboranlage	21
3.1.1	Pumpe und Ventile	21
3.1.2	Tank 101	23
3.1.3	Tank 102	23
3.1.4	Tank 103	23
3.2	Vergleich zu einem industriellen 2-Tank-Sudhauses	25
3.3	Grundlagen Würzeherstellung	26
4	Modularisierung und Programmierung der Laboranlage	29
4.1	Analyse und Dokumentation des Verfahrens	29
4.1.1	Ermittlung der notwendigen Prozessabschnitte	30
4.1.2	Festlegung der Prozessoperationen	30
4.1.3	Festlegung der Prozessschritte	32
4.2	Modularisierung der Laboranlage	33
4.2.1	Unterteilung der Laboranlage in Teilanlagen	33
4.2.2	Unterteilung der Teilanlagen in technische Einrichtungen	36
4.2.3	Unterteilung der technischen Einrichtungen in Einzelsteuereinheiten	38
4.3	Definition der Schnittstellen	40
4.4	Aufbau eines Moduls in Codesys	46
4.5	Aufbau der Process Cell	49
4.6	Anbindung an das Manufacturing Execution System	50
4.7	Erzeugen der fertigen Benutzeroberfläche	51
4.8	Erstellen des fertigen Modulbaums	51
4.9	Hardwareanbindung	53
5	Evaluierung des modularen Aufbaus	55
5.1	Austausch eines Moduls	55
5.2	Wiederverwendung eines Moduls	57
5.3	Test der Laboranlage	59

6	Übertragung auf eine industrielle Anwendung	61
7	Zusammenfassung	63
1	Literatur	65

Abbildungsverzeichnis

2.1	Modulare vs. Integrale Systeme	6
2.2	Prozessmodell mit dem Beispielprozess <i>Herstellen von Polyvinylchlorid</i>	9
2.3	Anlagenmodell	12
2.4	Reaktor Unit mit den technischen Einrichtungen Dosieren, Inertisieren, Entleeren, Überwachung und Temperieren	14
2.5	Prozedursteuerung	16
3.1	R&I-Fließbild der Laboranlage	22
3.2	Formalisierte Prozessbeschreibung der Würzeherstellung	27
4.1	Prozess Hierarchie	31
4.2	Aufbau der <i>Pump Unit</i> , der <i>Tank 101 Unit</i> und der <i>Tank 102 Unit</i>	34
4.3	R&I-Fließbild der verwendeten Komponenten der Laboranlage	35
4.4	Aufteilung der Laboranlage in Teilanlagen	35
4.5	Ablaufdiagramm der technischen Funktionen	37
4.6	Aufteilung des <i>Pump_EM</i> in Einzelsteuereinheiten	39
4.7	UML Diagramm der Schnittstellen	41
4.8	Moduldeklaration des <i>Heater_CM</i>	48
4.9	Moduldeklaration des <i>Heater_EM</i>	48
4.10	Aufbau eines Moduls in CoDeSys	49
4.11	Rezept der <i>Process_Cell</i>	50
4.12	Benutzeroberfläche der Laboranlage	52
4.13	Modulbaum der fertig konfigurierten Anlage	52
5.1	R&I-Fließbild und Schnittstelle des <i>Simple_Pump_EM</i>	56
5.2	R&I-Fließbild und Schnittstelle des <i>Regulated_Pump_EM</i>	56
5.3	Wiederverwendung des <i>Level_Detection_CM</i>	58
5.4	Wiederverwendung des <i>Simple_Valve_CM</i>	58

Tabellenverzeichnis

3.1	Pumpe und Ventile	24
3.2	Sensoren und Aktoren in Tank 101	24
3.3	Sensoren in Tank 102	24
3.4	Sensoren in Tank 103	24

Abkürzungen

Abkürzung	Beschreibung
MES	Manufacturing Execution System
HMI	Human Machine Interface / Benutzeroberfläche
ProcAppCom	Process Application Composer
PLT	Prozessleittechnik

1 Einleitung und Zielsetzung

1.1 Ausgangssituation

Diese Bachelorarbeit wird im Rahmen des Projektes *Process Application Composer* (ProcAppCom) erstellt. Bei dem Projekt arbeiten Anlagenhersteller und Softwareentwickler der Industrie gemeinsam mit dem Lehrstuhl für Automatisierung und Informationssysteme an der Verkürzung des Engineering-Prozesses von Sudhäusern. Bei Sudhäusern handelt es sich um zentrale Teile einer Brauanlage, die für die Produktion von Würze aus Grundzutaten zuständig sind. Das Sudhaus ist in mehreren Versionen verfügbar, wobei das 2-Tank-Sudhaus die einfachste Variante bildet. Es verfügt über eine Pfanne und einen Läuterbottisch [1]. Um eine höhere Anzahl an Suden herzustellen, existieren neben dem 2-Tank-Sudhaus auch noch die 3-Tank- und die 4-Tank-Sudhäuser. Werden mehrere 2-Tank-Sudhäuser miteinander verglichen, so fällt eine große Ähnlichkeit untereinander auf. Aus diesem Grund sollen zur Reduktion des Entwicklungsaufwands der Steuerungscode, das Manufacturing Execution Systeme (MES) und die Benutzeroberfläche (HMI, engl: Human Machine Interface) existierender Sudhäuser in einem neuen Sudhaus wiederverwendet werden. Zur Wiederverwendung der Softwarelösung bestehender Sudhäuser, soll das Entwicklungstool *CODESYS Application Composer* eingesetzt werden. Im Rahmen der Bachelorarbeit soll eine prototypische Laboranlage, welche ein Zwei-Behälter-Sudhaus abbildet, in Betrieb genommen werden. Anhand dieser Laboranlage soll beispielhaft gezeigt werden wie mithilfe des *CODESYS Application Composer* sowohl der Steuerungscode, wie auch das MES und das HMI automatisch erzeugt werden können.

1.2 Problemstellung

Im Rahmen des Projektes ProcAppCom werden 2-Tank-Sudhäuser analysiert. Obwohl zwischen diesen verschiedenen 2-Tank-Sudhäuser eine große Ähnlichkeit besteht, gibt es auch einige Unterschiede, welche aus zusätzlichen, veränderten oder fehlenden Komponenten resultieren. Dieser Umstand erschwert ein einfaches Wiederverwenden existierende Software

durch Kopieren und anschließendes Modifizieren. Die Softwarestruktur ist oft so verwoben, dass Anpassungen in einem Bereich nicht ohne Folgen in anderen Bereichen der Software bleiben, und somit selbst kleinere Modifikationen erhebliche Quellen für Fehler sind, welche durch anschließende Tests eliminiert werden müssen. [2] Aus diesem Grund wird eine Methode benötigt, welche es erlaubt, dass Teile der Softwarelösung aus bereits existierenden Sudhäusern ohne anschließende Modifikation übernommen werden können.

1.3 Ziel der Bachelorarbeit

Die Softwarelösungen existierender Anlagen kann nicht komplett übernommen werden, so dass die Sudhäuser in Modulklassen unterteilt werden müssen. Aus diesem Grund liegt der Schwerpunkt der Inbetriebnahme der Laboranlage in der modularen Gestaltung des Steuerungscode, um anschließend das Konzept auf industrielle Brauanlagen von Industriepartnern übertragen zu können. Dazu wird die Laboranlage physisch in einzelne Module aufgeteilt, welche dann jeweils ihren eigenen Steuerungscode erhalten. Dadurch wird es möglich die einzelnen Module flexibel zu einer Gesamtanlage kombinieren zu können. Als Konzept soll die IEC 61512 (ISA-88), welche im speziellen die modulare Gestaltung einer chargenorientierten Anlage beschreibt, angewendet werden. Es werden zudem folgende Anforderungen gestellt:

- Anforderung 1: Einteilung der Laboranlage in Modulklassen nach ISA88
Um Softwarelösungen aus existierenden Sudhäusern kombinieren zu können, muss die Anlage zunächst in einzelne Module unterteilt werden. Als Standard für diese Modularisierung der Sudhäuser soll die ISA88 verwendet werden. Dabei ist darauf zu achten, dass die gefundenen Modulklassen in Kombination den Brauprozess komplett realisieren können.
- Anforderung 2: Austauschbarkeit und Wiederverwendung von Modulen
Eine Veränderung der Anlage soll vereinfacht werden, indem nur das ausgetauschte Modul ersetzt werden muss, wobei der Rest des Codes gänzlich unberührt bleibt. Außerdem soll es möglich sein einzelne Module ohne großen Aufwand wiederzuverwenden. Sei dies eine mehrfache Verwendung in einem Projekt oder eine Wiederverwendung von einem existierenden Modul in einem neuen Projekt. Um zu zeigen, dass das Ziel der Austauschbarkeit erfüllt worden ist, soll ein einzelnes Modul der Laboranlage ausgetauscht werden, ohne die restlichen verwendeten Module anzupassen. Die Anforderung an die Wiederverwendbarkeit kann bewiesen werden, indem ein Modul mindestens zwei mal verwendet wird.

- Anforderung 3: Einheitliche Schnittstellen zwischen Modulklassen

Damit existierende Modulklassen neu zusammengestellt werden können, wird eine einheitliche Schnittstelle der einzelnen Modulklassen benötigt. Diese erlaubt auch den unkomplizierten Austausch von Modulen. Die Überprüfung der Anforderung kann durch den Austausch von Modulen der Laboranlage gezeigt werden.

- Anforderung 4: Automatische Generierung des Steuerungscode

Der Steuerungscode für ein Sudhaus soll automatisch aus den verwendeten Modulen erzeugt werden. Dadurch soll der Aufwand und die Fehleranfälligkeit, einer manuellen Programmierung reduziert werden. Dazu soll das Sudhaus zunächst aus den vorhandenen Modulen zusammengestellt werden. Aus dieser Konfiguration soll dann mithilfe des *Application Composers* der individuelle Steuerungscode erzeugt werden.

- Anforderung 5: MES- und HMI Anbindung

Zur Reduzierung des Engineeringaufwands, soll zusätzlich die Erstellung der HMI und die MES-Anbindung der Sudhäuser vereinfacht werden. Zu diesem Zweck soll mithilfe des *Application Composers* eine individuelle Benutzeroberfläche aus der gewählten Modulkonfiguration erstellt werden. Zudem soll sich die MES-Anbindung, nach der Steuerungscodeerzeugung, einfach und schnell realisieren lassen.

1.4 Aufbau der Arbeit

Vorerst wird das eingesetzte Entwicklungstool *CODESYS Application Composer* kurz vorgestellt.

Anschließend wird im Kapitel 2 der Begriff der Modularisierung näher betrachtet. Dazu wird die IEC 61512, welche Referenzmodelle für die chargenorientierte Fahrweise definiert, betrachtet. Dabei werden unter anderem die einzelnen Modelle, wie das Prozessmodell, das Anlagenmodell und das Prozedurmodell erklärt. Diese werden in der darauf folgenden Arbeit benötigt um die Laboranlage in die einzelnen Module zu unterteilen.

Darauffolgend wird, im Kapitel 3, der Aufbau der Laboranlage ausführlich erklärt. Hierzu werden die einzelnen Einheiten der Anlage, sowie alle vorhandenen Aktoren und Sensoren, vorgestellt.

Im Kapitel 4 wird die Laboranlage in Module unterteilt und programmiert. Vorweg muss das Verfahren in seine einzelnen Prozessschritte zerlegt werden. Anhand der aus diesem Schritt gewonnen Informationen werden später die einzelnen Module festgelegt. Danach wird für jedes Modul eine eigne Schnittstelle definiert. Als nächstes wird der grundlegende Aufbau eines Moduls in *Codesys*, und im speziellen der der Aufbau der Process Cell, aufgezeigt. Nachdem

die Anbindung an das Manufacturing Execution System erklärt wird, wird erläutert wie das fertige HMI erzeugt wird.

Im Kapitel 5 wird der modulare Steuerungscode bewertet. Dazu wird überprüft ob die beiden Anforderungen der Wiederverwendung und der Austauschbarkeit erfüllt werden.

Schließlich wird im Kapitel 6 gezeigt, wie das vorgestellte Konzept auch auf industrielle Anlagen übertragen werden kann.

1.5 Application Composer

Zur Erstellung des Steuerungscode wird das Programm *CODESYS V3* der Firma *3S-Smart Software Solutions GmbH* verwendet. Um zusätzlich der Forderung der Modularisierung nachzukommen wird das Entwicklungstool *CODESYS Application Composer* eingesetzt.

Mit dem *Application Composer* können komplette Steuerungsapplikationen aus zuvor erstellten Softwaremodulen zusammengestellt und parametrisiert werden. Die Module enthalten neben dem Programmcode typischerweise Komponenten wie Visualisierungselemente, Parametrierung oder Hardwarezuordnungen. Die Konfektionierung, also die Zusammenstellung und Parametrierung, der Module erfolgt im Modulbaum. Wenn neue Elemente in die Modulstruktur eingefügt werden, stehen nur passende Module zur Auswahl. In den Moduleigenschaften werden die Parametrierung, die Hardwareanbindung und die Visualisierungsauswahl der Modulinstanzen definiert [3]. Durch einen integrierten Generator wird der komplette Applikationscode inklusive Visualisierung und Hardwareanbindung automatisch erstellt und unmittelbar auf die Steuerung übersetzt. Außerdem können passende Module sofort von Anbietern mitgeliefert werden, um den Aufwand für den Kunden zu verringern. Die Modularisierung des Steuerungscode und die Bereitstellung kompatibler Module werden durch den *Application Composer* ermöglicht. Somit kann die Steuerungsimplementierung auch durch Personen mit wenig Programmiererfahrung, wie zum Beispiel Technologen, Projektierer oder sogar Kundenberater, vorgenommen werden [4].

2 Aktueller Stand der Technik

2.1 Was bedeutet Modularisierung

Als Module werden eigenständige Baugruppen mit klar abgetrennten Funktionen, welche separat testbar und wiederverwendbar sein sollen, bezeichnet. In den frühen Phasen der Anlagenplanung stellt vor allem die Wiederverwendbarkeit den größten Nutzen dar [5]. Um Module wiederverwenden zu können, sind diese und deren Funktionen dabei austauschbar zu gestalten. Des weiteren sollen Module definierte Schnittstellen zu anderen Modulen besitzen [6]. Für die Wiederverwendbarkeit der Software ist es sowohl entscheidend wie leicht diese zu verstehen ist, als auch welchen Aufwand es bedeutet sie zu verändern und danach neu zu testen. Ohne die Nutzung einer modularen Struktur wird eine Software oft komplett oder teilweise in das neue Projekt kopiert und entsprechend, den neuen Gegebenheiten, modifiziert. Allerdings ist die Softwarestruktur oft so verwoben, dass selbst kleinere Änderungen erhebliche Quellen für Fehler sind. Die Nutzung von Modulen ist vorteilhaft, da sowohl die Änderungseffizienz wie auch die Verständlichkeit gesteigert werden können [2]. Aufgrund vieler positiver Aspekte der modularen Struktur, findet diese einen großen Anklang in der Industrie. Als Beispiel kann hier die Automobilindustrie aufgeführt werden, wo vor allem das Plattformkonzept der Volkswagen AG und das vollständig modulare Fahrzeug SMART bekannt sind. Jedoch existieren auch Fabriken, die einem modulartigen Aufbau folgen. Aktuell werden modulare Konzepte in fast allen Industriesektoren, die komplexe Produkte herstellen, angewendet. Die Systemelemente von komplexen Systemen stehen in komplizierten Beziehungen zueinander. Jedoch ist es möglich, mit Hilfe einer geeigneten Zusammenstellung und Hierarchiebildung der Elemente, relativ autonome Module zu erzeugen. Diese Module zeigen eine starke interne Beziehung auf, wobei die Beziehungen zwischen den einzelnen Subsystemen relativ schwach sind (siehe Abbildung 2.1) [7]. Eine geschickte Modularisierung muss den Nachweis erbringen, dass der Mehraufwand der Funktionskapselung durch eine Verkürzung der Planungsdauer von Anlagen nicht nur kompensiert wird, sondern durch Wiederverwendung sogar wirtschaftliche Vorteile möglich sind [8]. Neben dem Vorteil der Wiederverwendbarkeit können für modulare Architekturen folgende Eigenschaften festgehalten werden [9]:

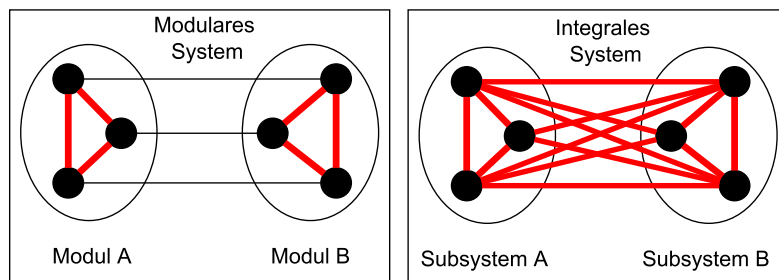


Abbildung 2.1: Modulare vs. Integrale Systeme [9]

- Austauschbarkeit der Module
- Standardisierte Schnittstellen zwischen den Modulen: Die Schnittstellen legen die Inhalte der Interaktion fest.
- Relative Autonomie der Module gegenüber Veränderungen: Es können Veränderungen an den Modulen vorgenommen werden, ohne das Gesamtsystem zu verändern.

2.1.1 Vor- und Nachteile der Modularisierung

Im Folgenden soll ein kurzer Überblick über die Vor- und Nachteile der Modularisierung gegeben werden

- **Vorteile:** Unternehmen, welche das Konzept der Modularisierung anwenden, profitieren von einer erhöhten strategischen Flexibilität, da einzelne Module relativ leicht ausgetauscht werden können. Zudem können sie ihren Entwicklungsprozess beschleunigen, indem sie die dafür notwendigen Innovationen eines Moduls auf ihre Zulieferer verteilen. Mit diesem Ansatz können viele Module gleichzeitig weiterentwickelt werden. Außerdem führt dies zu einem erhöhten Wettbewerb unter den verschiedenen potentiellen Zulieferern und einer möglichen Kostenreduktion. Aus Kundensicht führt die Modularisierung zu einer Komplexitätsreduktion des Produktes und zu einer erhöhten Anzahl an Produktvarianten [9].
Zudem eignet sich ein System, welches modular aufgebaut ist, für modulare Tests. Dies bedeutet, wenn ein Modul mehrmals verwendet wird, ist es nur notwendig eine einzige Einheit zu testen [10].

- **Nachteile:** Trotz den diversen Vorteilen werden mit dem Konzept der Modularisierung auch negative Aspekte hervorgerufen. So ist bei der Entwicklung und der Produktion ein erhöhter Design- und Koordinationsaufwand nötig, da die Module auf einander abgestimmt werden müssen. Außerdem kann die Produktleistung unter der Modularisierung leiden, wenn einzelne Module nicht dem benötigten Qualitätsstandard entsprechen. Schließlich sinkt mit dem Konzept der Modularisierung auch der Schutz vor Imitation, da zum Beispiel Module relativ leicht von Zulieferern bezogen werden können [9].

2.2 Automatisierung von Chargenprozessen

In diesem Abschnitt werden wichtige Begriffe der IEC 61512-1 (ISA-88) über die chargenorientierte Fahrweise erklärt. Die deutsche Übersetzung der internationalen Norm IEC 61512-1 ist die DIN EN 61512-1 [11]. Die vorgestellten Begriffe werden in den nachfolgenden Kapiteln benötigt, da die Modularisierung der Laboranlage in Anlehnung an die Regeln der IEC 61512-1 erfolgt.

Zunächst wird der Begriff Chargenprozess definiert. Anschließend werden das Prozessmodell, das physische Modell und das Prozedurmodell vorgestellt. Diese 3 Modelle werden benötigt um den Prozess der Würzeherstellung, sowie die dafür benötigte Anlage, aufteilen zu können.

2.2.1 Chargenprozesse

Die Chargenprozesse, die in der Norm DIN EN 61512-1 [11] betrachtet werden, führen zur Herstellung von endlichen Mengen an Stoff (Chargen). Dazu werden diese in Teilanlagen eingefüllt, wo sie eine bestimmte Zeit verbleiben, dabei chemisch reagieren oder mechanisch bearbeitet werden und schließlich in nachfolgende Teilanlagen fließen [12].

Laut Urbas [13] zeigt ein Chargenprozesse folgende Eigenschaften auf:

- Modifizierung der Chargen durch Verarbeitungsschritte wie zum Beispiel Heizen, Kühlen, chemische Reaktionsschritte und thermale Trennung.
- Verarbeitungsschritte werden nacheinander entweder in einer universellen Einrichtungen oder in mehreren spezifischen Einrichtungen, mit zwischenschrittlichen Transportstufen, durchgeführt.
- Rezepte beschreiben den festen oder zustandsabhängigen Ablauf der Verarbeitungsschritte.

Ein Chargenprozess ist ein diskontinuierlicher Prozess. Er ist damit weder ein diskreter Prozess, bei dem sich eine bestimmte Produktmenge als Einheit von Arbeitsplatz zu Arbeitsplatz bewegt wird und dabei jedes Teil seine eindeutige Identität behält, noch ein kontinuierlicher Prozess, bei dem die Stoffe in einem stetigen Strom durch die Verarbeitungseinrichtung geführt werden. Dennoch weist der Chargenprozess Merkmale beider Typen auf [11]. Die Automatisierung derartiger Prozesse erfolgt durch diskrete Steuerungen, die häufig mit binären Mess- und Stellgrößen arbeiten und beispielsweise Ventile oder Heizungen steuert. Eine diskrete Steuerung arbeitet, im Gegensatz zu zeitkontinuierlichen Regelungen, nur zu den Zeitpunkten an denen eine Messgröße das Erreichen eines vorgegebenen Grenzwertes signalisiert. Zwischen diesen Zeitpunkten sind die Stellgrößen konstant [12].

2.2.2 Prozessmodell

Unter einem Prozess wird ein Ablauf von chemischen, physikalischen oder biologischen Vorgängen zur Gewinnung, Herstellung, Umwandlung, Transport, Speicherung oder Beseitigung von Stoffen oder Produkten verstanden [14]. Ein Prozess kann in Prozessabschnitte gegliedert werden, welche wiederum in Prozessoperationen unterteilt werden können. Die Prozessoperationen können dann weiter in Prozessschritte zerlegt werden. Die verschiedenen Prozessabschnitte, Prozessoperationen und Prozessschritte können sowohl seriell wie auch parallel ablaufen [11]. Diese hierarchische Aufteilung des Prozesses ist in der Abbildung 2.2 mit der beispielhafte Aufteilung des Prozesses *Herstellen von Polyvinylchlorid* dargestellt.

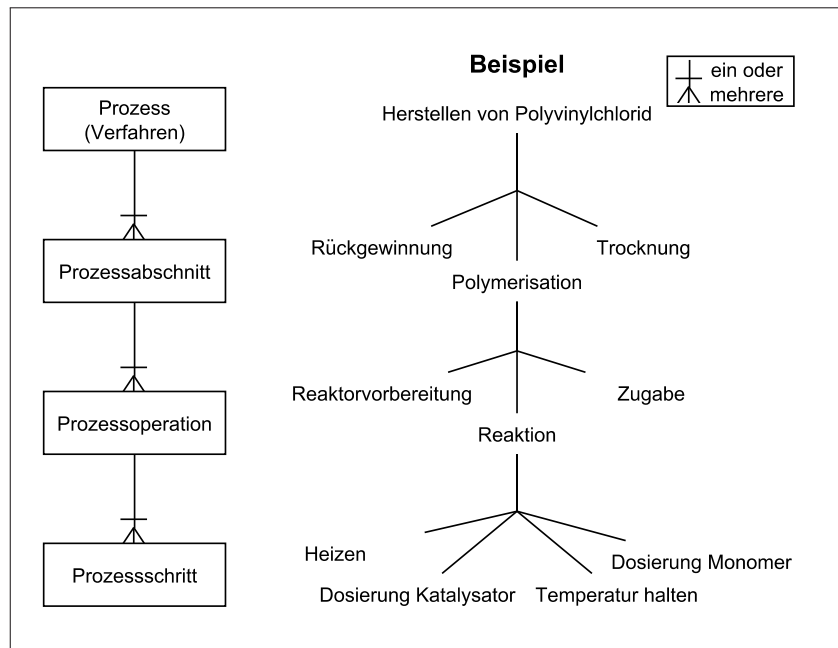


Abbildung 2.2: Prozessmodell mit dem Beispielprozess *Herstellen von Polyvinylchlorid* [15]

2.2.2.1 Prozessabschnitte

Ein Prozess kann in einen oder mehrere Prozessabschnitte untergliedert werden [11]. Diese lassen sich organisatorisch weitgehend autark ausführen und dienen der Verarbeitung von Stoffen [16]. Typische Prozessabschnitte im Polyvinylchlorid-Prozess, wie er in der Abbildung 2.2 dargestellt ist, könnten folgende sein [11]:

- **Polymerisation:** Das Vinylchlorid-Monomer wird zu Polyvinylchlorid polymerisiert.
- **Rückgewinnung:** Rückgewinnung von Vinylchlorid-Monomeren.
- **Trocknung:** Das Polyvinylchlorid Pulver wird getrocknet.

2.2.2.2 Prozessoperationen

Jeder Prozessabschnitt kann in eine oder mehrere Prozessoperationen unterteilt werden [11]. „Prozessoperationen beschreiben größere Verarbeitungsaktivitäten, die gewöhnlich eine chemische bzw. physikalische Umwandlung der verarbeiteten Stoffe bewirken.“[14]

Der Prozessabschnitt *Polymerisation* aus dem Beispiel in Abbildung 2.2 kann in folgende Prozessoperationen unterteilt werden [11]:

- **Reaktorvorbereitung:** Sauerstoff aus dem Reaktor abzuziehen.
- **Zugabe:** Reaktor mit destilliertem Wasser und einem Farbindikator befüllen.
- **Reaktion:** Einfüllen von Vinylchlorid-Monomer und Katalysator. Anschließendes Erhitzen auf 55 °C bis 60 °C.

2.2.2.3 Prozessschritte

Jeder Prozessoperation besteht aus einer geordneten Gruppe von einem oder mehreren Prozessschritten [11]. Ein Prozessschritt beschreibt einen kleineren Verarbeitungsvorgang, welcher zur Ausführung der ihm übergeordneten Prozessoperation erforderlich ist [14]. Sie lassen sich nicht sinnvoll in weitere kleinere autonome Einheiten zerlegen. Ein Prozessschritt ist die unterste Stufe definierbarer Vorgänge innerhalb eines technischen Prozesses [16]. Typische Prozessschritte für die Prozessoperation *Reaktion*, wie sie in Abbildung 2.2 dargestellt ist, könnten sein [11]:

- **Dosierung Katalysator:** Die geforderte Menge Katalysator wird in den Reaktor gefüllt.
- **Dosierung Monomer:** Der Reaktor wird mit der benötigten Menge Vinylchlorid-Monomer befüllt.
- **Heizen:** Der Reaktor wird auf eine Temperatur zwischen 55 °C und 60 °C aufgeheizt.
- **Temperatur halten:** Die Temperatur im Reaktor soll zwischen 55 °C bis 60 °C gehalten werden.

2.2.3 Anlagenmodell

Das Anlagenmodell liefert ein hierarchisches Modell der Dekomposition einer Anlage, wobei die physischen Elemente der Anlage betrachtet werden [13]. Das Anlagenmodell hat sieben Ebenen. Von oben beginnend mit dem Unternehmen, dem Werk und dem Anlagenkomplex.

Die einzelnen Ebenen sind in Abbildung 2.3 dargestellt, auf der auch eine beispielhafte Aufteilung einer Anlage, welche zur Polyesterherstellung dient, gezeigt wird. Da die drei ersten Ebenen oft im Zusammenhang mit unternehmerischen Betrachtungen genutzt werden und auch in der betrachteten Laboranlage nicht existieren, werden sie nicht weiter behandelt [11]. Die im Rahmen dieser Arbeit benötigten Ebenen, Anlage, Teilanlage, technische Einrichtung und Einzelsteuereinheit, werden in den folgenden Abschnitten erläutert.

2.2.3.1 Anlage

Eine Anlage (engl: process cell) ist eine logische Gruppierung von Einheiten, die für die Produktion von einer oder mehreren Chargen erforderlich ist [13]. Sie enthält Teilanlagen, technische Einrichtungen und Einzelsteuereinheiten. Das Vorhandensein der Anlagenebene erlaubt die Disposition auf dieser Ebene und den Entwurf anlagenweiter Steuerungsstrategien [11].

Anlagen können nach der Anzahl der Produkte und nach ihrer physischen Struktur klassifiziert werden. Dabei kann es sich je nach Anzahl der Produkte um eine Einproduktanlage oder um eine Mehrproduktanlage handeln [11]:

- **Einproduktanlage:** Es wird in jeder Charge das gleiche Produkt hergestellt. Jedoch sind Variationen der Prozeduren oder der Parametern möglich um beispielsweise auf veränderte Einsatzstoffe oder auf Änderungen in den Umgebungsbedingungen zu reagieren.
- **Mehrproduktanlage:** Es werden unter Anwendung unterschiedlicher Herstellungs- oder Steuerverfahren verschiedene Produkte hergestellt.

Die Struktur einer Anlage kann je nach physischer Gegebenheiten als Einwege-Struktur, Mehrwege Struktur oder Netzwerk-Struktur klassifiziert werden [11]:

- **Einwege-Struktur:** Eine Charge durchläuft die verschiedenen Teilanlagen sequentiell.
- **Mehrwege-Struktur:** Die Charge kann mehrere parallel angeordneten Einwege-Strukturen, ohne gegenseitigen Produktaustausch, durchlaufen.
- **Netzwerk-Struktur:** Der Weg einer Charge kann sowohl festgelegt als auch veränderlich sein.

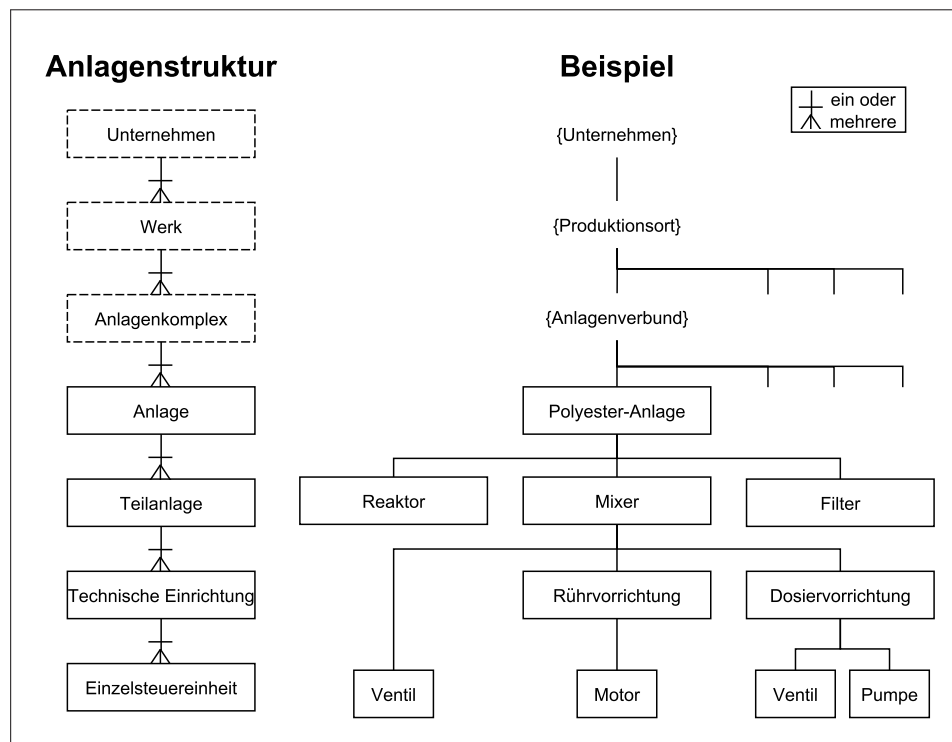


Abbildung 2.3: Anlagenmodell [15]

2.2.3.2 Teilanlage

Aufteilen lässt sich eine Anlage in Teilanlagen (engl. Unit). Eine oder mehrere größere Verarbeitungsaktivitäten, wie zum Beispiel Reaktion, Kristallisation oder Herstellen einer Lösung, können in einer Teilanlage stattfinden. Teilanlagen sollten so definiert sein, dass sie relativ unabhängig voneinander arbeiten [11]. Sie führen während eines Prozesses in der Regel einen einzigen Prozessabschnitt aus [16]. Es wird vorausgesetzt, dass eine Teilanlage nicht mehr als eine Charge zur gleichen Zeit verarbeitet. Jedoch können sie auch nur einen Teil einer Charge enthalten oder verarbeiten [11].

In der Verfahrenstechnik werden unter anderem zwischen folgenden Typen von Teilanlagen unterscheiden [16]:

- **Teilanlagen zur Stoffaufbereitung:** Beispielsweise eine Mühle mit angeschlossener Wiegestation und Verdampfer zur Konzentration von Stoffen. Die ablaufenden Prozessabschnitte bewirken keine chemischen Veränderungen der Edukte.
- **Teilanlagen zur Stoffumwandlung:** Dies können zum Beispiel Reaktionsapparate wie Autoklaven oder Rohrreaktoren, mit ihren zugehörigen Hilfseinrichtungen, sein.

- **Teilanlagen zur Stoffaufarbeitung:** Hierzu gehören unter anderem Destillationskolonnen und Filterapparate oder Verpackungseinrichtungen. Wie bei den Teilanlagen zur Stoffaufbereitung kommt es auch hier zu keiner chemischen Reaktionen.
- **Teilanlagen zur Stoffspeicherung:** Sie bestehen im Wesentlichen aus Pufferbehälter, um andere Teilanlagen voneinander zu entkoppeln. Sie dienen der Aufbewahrung von Stoffen auf, ohne deren Eigenschaften zu ändern.
- **Teilanlagen zum Stofftransport:** Sie übernehmen den Transport zwischen den einzelnen verfahrenstechnischen Apparaten. Dieser findet meistens über Leitungssysteme mit Pumpvorrichtungen, Armaturen und Dosiereinrichtungen statt.

Eine beispielhafte Aufteilung der Anlage zur Herstellung von Polyester, in die drei Teilanlagen Reaktor, Mixer und Filter, wird in Abbildung 2.3 gezeigt.

2.2.3.3 Technische Einrichtung

Laut Krahe [17] stammt der „[...] Begriff Equipment Module [...] aus der Automatisierung und steht für eine funktionale Einheit, die sich aus Armaturen, Sensoren, Apparaten und mechanischen Komponenten zusammensetzt.“

Eine technische Einrichtung (engl. Equipment Module) kann zu einer Teilanlage gehören, aber auch ein eigenständiger Teil innerhalb einer Anlage darstellen. Sie kann aus Einzelsteuereinheiten und untergeordneten technischen Einrichtungen bestehen. Gemäß DIN EN 61512-1 [11] kann eine technische Einrichtung Prozessschritte ausführen und verwirklicht somit die Fähigkeiten der Teilanlage [16]. Zur Definition einer technischen Einrichtung wird die Kenntnis der gewünschten Verarbeitungsaktivitäten und der Fähigkeiten der entsprechenden Einrichtungen benötigt. Die technische Einrichtung schließt alle notwendigen Einrichtungen ein, die zur Ausführung dieser Aktivitäten benötigt werden [11]. Typische technische Einrichtungen in der Verfahrenstechnik sind, Verbindungseinrichtungen, Dosiereinrichtungen, Waagen und Heiz-/Kühleinrichtungen [16]. Eine beispielhafte Aufteilung der Teilanlage Mixer in die zwei technische Einrichtung Rührvorrichtung und Dosiervorrichtung wird in Abbildung 2.3 gezeigt. Zudem ist in Abbildung 2.4 die Aufteilung einer Teilanlage in verschiedene technische Einrichtungen abgebildet.

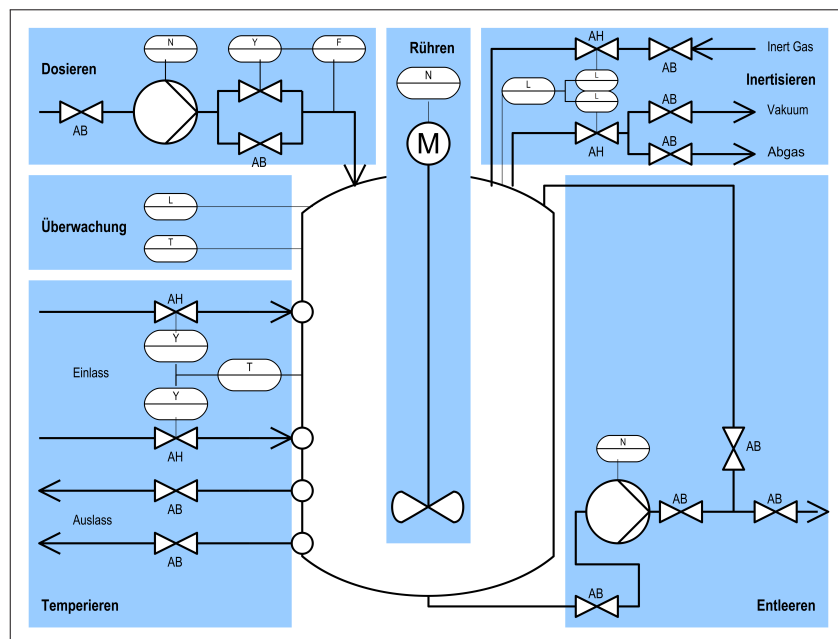


Abbildung 2.4: Reaktor Unit mit den technischen Einrichtungen Dosieren, Inertisieren, Entleeren, Überwachung und Temperieren [13]

2.2.3.4 Einzelsteuereinheit

Eine Einzelsteuereinheit ist typischerweise eine Zusammenfassung mehrere Prozessleittechnik-Stellen (PLT-Stellen) [11]. PLT-Stellen erfassen, verarbeiten und beeinflussen unmittelbar einzelne Prozessgrößen [16]. Sie bestehen stets aus folgende PLT-Stellenelementen [16]:

- eine oder mehrere Signalquellen (z. B. Signale von Sensoren)
- eine Einheit zur Signalverarbeitung (z. B. Regler, Grenzwertbilder)
- eine oder mehrere Signalsenken (z. B. Aktoren)

Einige Beispiele für Einzelsteuereinheiten sind [11]:

- Eine über einen Sollwert gesteuerte Regeleinrichtung, welche aus einem Transmitter, einem Regler und einem Stellventil besteht.
- Eine Einrichtung bestehend aus einem Magnetventil mit Endschaltern, welches über einen Sollwert gesteuert wird.
- Eine Dosiereinrichtung, die mehrere automatische Schaltventile enthält und koordiniert.

2.2.4 Prozedurmodell

Mit den Komponenten des physischen Modells kann dem Produktrezept gefolgt werden, indem zum Beispiel Ventile geöffnet und geschlossen werden. Dies kann unter anderem manuell, über die grafische Benutzeroberfläche des Bedieners, erfolgen. Jedoch würde dies nicht dem Ziel einer automatischen Chargenproduktion entsprechen. Zur Herstellung eines Produktes wird dazu die Prozedur benötigt [18]. Das Prozedurmodell wird erstellt, indem das Prozessmodell in die Ebenen einer Ablaufprozedur übertragen werden. Die Prozedur besteht aus Teilprozeduren, welche jeweils aus Operationen zusammengesetzt sind [19]. Diese bestehen aus einer Folge von Funktionen die eine prozessorientierte Aufgabe durchführen können [20]. Jeder der gerade eben genannten Ebenen des Prozedurmodells können Elemente der Prozedursteuerung zugeordnet sein, die für die automatisierte Umsetzung der betrieblichen Produktionsabläufe sorgen. Abbildung 2.5 zeigt die möglichen Ebenen der Steuerung. Diese korrespondierend mit den entsprechenden Prozedurebenen. Jedoch müssen nicht alle in der Grafik aufgezeigten Ebenen der Steuerungshierarchie zwingend umgesetzt werden. So sind meistens nur die technischen Funktionen, als Steuerungsmodule in den prozessnahen Komponenten der Leittechnik projiziert [19]. Die einzelnen Ebenen des Prozedurmodells werden folgend weiter betrachtet:

2.2.4.1 Prozedur

Die Prozedur bildet die oberste Ebene in der Hierarchie des Prozedurmodells. Durch sie wird die Strategie für die Ausführung einer Verarbeitungsaktion, wie zum Beispiel der Herstellung einer Charge, festgelegt. Die Prozedur kann in Teilprozeduren unterteilt werden [11].

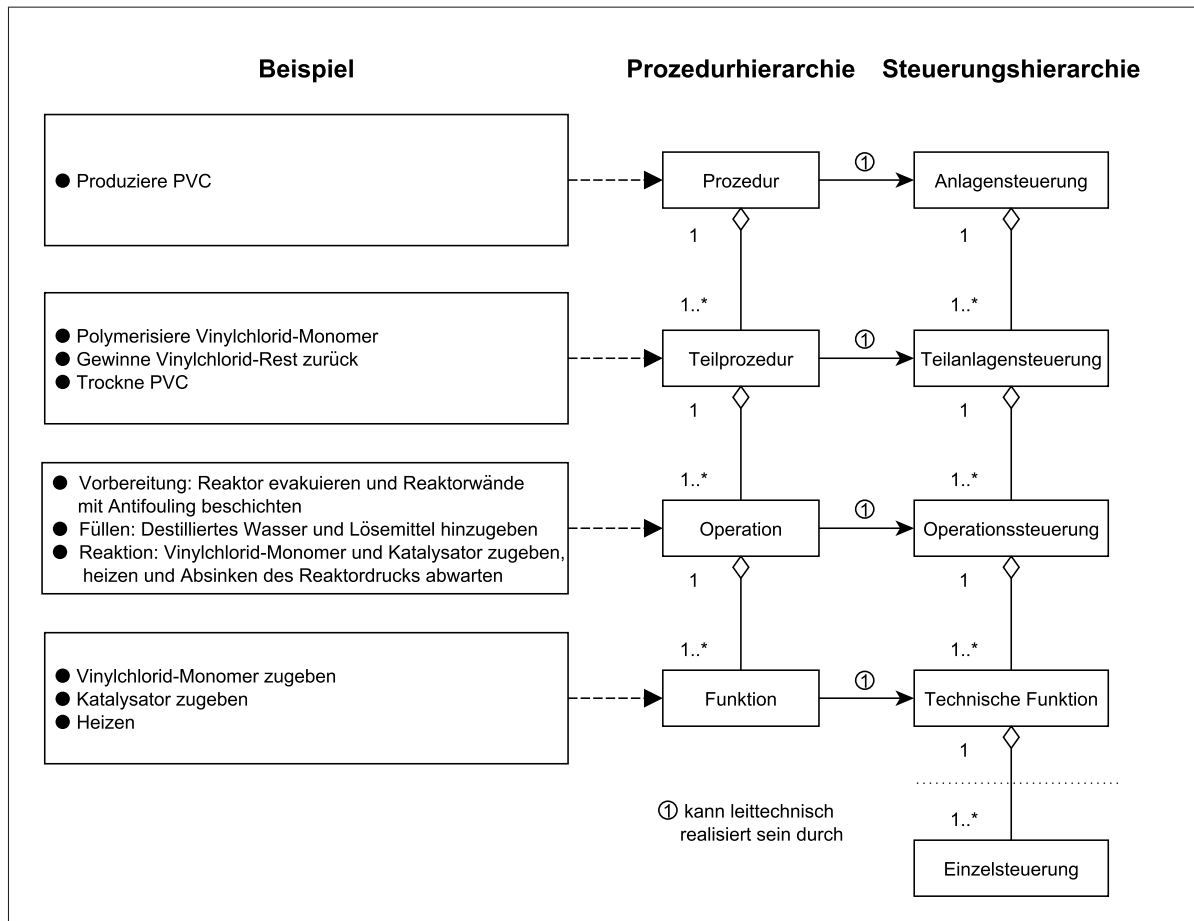


Abbildung 2.5: Prozedursteuerung [13], [19]

2.2.4.2 Teilprozedur

Eine Teilprozedur kann in eine geordnete Menge von Operationen unterteilt werden. Diese bewirken, dass eine zusammenhängende Produktionssequenz in einer Teilanlage stattfindet [11].

2.2.4.3 Operation

Eine Operation legt eine größere Verarbeitungssequenz fest und bewirkt, dass die verarbeiteten Stoffe von einem Zustand in einen anderen überführt werden. Diese Überführung ist gewöhnlich mit einer chemischen oder physikalischen Umwandlung verbunden. Es wird vorausgesetzt, dass zu jedem Zeitpunkt jeweils nur eine einzige Operation pro Teilanlage aktiv ist. In der Regel wird eine Operation vollständig in einer einzelnen Teilanlage ausgeführt und besteht aus einer geordneten Menge von Funktionen [11].

2.2.4.4 Funktion

Das kleinste Element einer Prozedursteuerung, welches in der Lage ist eine prozessorientierte Aufgabe auszuführen, ist die Funktion.

2.2.5 Rezepte

Nachdem die einzelnen Ebenen des Prozedurmodells betrachtet wurden, wird nun das Rezept vorgestellt. Neben der Anweisungen wie eine Produktcharge hergestellt werden kann, liefert die Rezept-Prozedur noch weitere Informationen [18]. In der Praxis handelt es sich dabei um folgende Informationen [11]:

- Rezeptkopf
- Stoff- und Produktionsdaten
- Anforderungen an die Einrichtung
- Rezeptprozedur
- Verfahrensrezeptprozedur

2.2.6 Betriebsarten

Eine Betriebsart legt fest, wie auf Befehle reagiert wird und wie diese wirksam werden [11]. Als Beispiel können folgende Betriebsarten genannt werden [21]:

- 1.) Manuell
- 2.) Automatik

Der Modus *Manuell* wird meistens für Wartungsarbeiten, Reinigung oder ähnliche Zwecke verwendet. Der Bediener kann den Weiterlauf anhalten oder Weiterschaltungen erzwingen [11]. Allerdings liegt die Verantwortung, die Vorschriften für den Betrieb der Geräte zu beachten, in diesem Modus beim Betreiber der Anlage. Der *Automatikbetrieb* wird im Normalbetrieb des Gerätes verwendet [21]. Dabei werden die Weiterschaltungen ohne Unterbrechung ausgeführt, wenn die zugehörigen Bedingungen erfüllt sind. Eine strikte Verwendung der oben genannten Betriebsarten ist nicht zwingen erforderlich. Zudem werden weitere Betriebsarten nicht von der IEC 61512 ausgeschlossen [11]. Im Rahmen dieser Bachelorarbeit wird nur der Betriebszustand Automatik verwendet.

2.2.7 Betriebszustände

Der Betriebszustand gibt den aktuellen Zustand vollständig wieder. Im Falle eines Ventils kann der Betriebszustand *Prozent auf* sein, im Falle eines Prozedurelements kann es *Läuft* oder *Beendet* sein. Als Beispiel für Prozedur-Betriebszustände können folgende genannt werden [11]:

- 1.) *LEERLAUF*: Der START-Befehl wird abgewartet, der einen Übergang zum Zustand *LÄUFT* auslöst.
- 2.) *LÄUFT*: Normale Operation.
- 3.) *BEENDET*: Die normale Operation wird bis zum Ende durchlaufen. Anschließen wird der Befehl RESET erwartet, der einen Übergang zum Zustand *LEERLAUF* auslöst
- 4.) *UNTERBRECHEND*: Der PAUSE-Befehl bewirkt, dass das Prozedurelement an der nächsten festgelegten, sicheren oder stabilen Stelle anhält. Der Zustand geht anschließend automatisch zu *UNTERBROCHEN* über.
- 5.) *UNTERBROCHEN*: Dieser Zustand gilt üblicherweise für kurze Unterbrechungen. Ein Befehl FORTFAHREN veranlasst den Übergang zurück zum *LÄUFT*-Zustand, indem die normale Bearbeitung wieder nach der Unterbrechungsstelle aufgenommen wird.

2.3 Schnittstellen

Mit der Definition der Schnittstellen wird definiert, wie die einzelnen Module von unterschiedlichen Ebenen miteinander kommunizieren. Durch die Verwendung von Schnittstellen kann festgelegt werden, dass Module, die eine identische Schnittstelle benutzen, auch gleich zu verwenden sind, obwohl ihr innerer Aufbau durchaus variieren kann. Dies kann an einem einfachen Beispiel gezeigt werden. Ein einfachwirkender Hydraulikzylinder und ein doppeltwirkender Hydraulikzylinder können die selbe Schnittstelle mit den Methoden „Einfahren“ und „Ausfahren“ besitzen obwohl sie sich in der technischen Realisierung unterscheiden. Dies bietet dem Programmierer den Vorteil, dass er sich nur bei der Erstellung der einzelnen Module mit deren Ansteuerung befassen muss, später hingegen nicht mehr. Zudem wird die Wiederverwendung der einzelnen Module durch die Schnittstelle in Folgeprojekten wesentlich vereinfacht, weil angenommen werden kann, dass die Verwendung in dem neuen Kontext möglich ist, sobald die Schnittstelle übereinstimmt [2]. Bei der Schnittstellendefinition ist zu

berücksichtigen, dass möglichst wenige und möglichst einfache Schnittstellen festgelegt werden. Dies erfordert stets einen Kompromiss zwischen steigender Komplexität und gewünschter Flexibilität [22].

3 Aufbau der Laboranlage

In diesem Kapitel wird die verwendete Laboranlage ausführlich beschrieben. Dazu werden die einzelne Einheiten genauer betrachtet. Um eine bessere Übersicht zu erhalten, zeigen R&I-Fließbilder den genauen Aufbau und aus den Tabellen wird ersichtlich wie die vorhandenen Aktoren und Sensoren im weiteren Verlauf bezeichnet werden und ob diese in der HMI angezeigt und an das MES übergeben werden.

Die Anlage besteht aus drei Behältern, dem *Tank 101*, dem *Tank 102* und einem Druckbehälter, der die Bezeichnung *Tank 103* trägt. In diesem Szenario wird der Druckbehälter, obwohl er angeschlossen ist, nicht verwendet, weil er für die Darstellung der Würzeherstellung nicht benötigt wird. Alle Behälter sind durch Rohre und Ventile miteinander verbunden.

3.1 Beschreibung der Laboranlage

3.1.1 Pumpe und Ventile

Alle Behälter sind so mit dem Leitungsnetz verbunden, dass es möglich ist von jedem einzelnen Tank in jeden anderen Tank zu Pumpen. Zudem besteht beim *Tank 101* und *Tank 102* die Möglichkeit gleichzeitig Wasser am unteren Anschluss des Behälters anzusaugen und dem gleichen Behälter wieder über den oberen Anschluss zuzuführen. Dies wird im späteren Prozess unter anderem dazu verwendet, die Flüssigkeit beim Erhitzen im *Tank 101* umzuwälzen, weil im Gegensatz zu einem realen Sudhaus kein Rührer zur Verfügung steht um eine gleichmäßige Durchmischung zu gewährleisten. Um dies zu ermöglichen besitzt jeder Tank zwei Ventile. Jeweils ein Ventil für den Zufluss und eins für den Abfluss. Dementsprechend besitzt die Anlage insgesamt sechs Ventile und eine Pumpe. Bei der eingesetzte Pumpe kann zwischen zwei verschiedenen Betriebsmodi gewählt werden. Im ersten Modus kann die Pumpe nur in den beiden Zuständen *Ausgeschaltet* und *Volle Leistung* betrieben werden. Im zweiten Modus ist es möglich die Fördermenge der Pumpe anzupassen. Dadurch ist es möglich, durch eine geeignete Regelung, den Durchfluss der Pumpe zu regeln. Von den insgesamt sechs vorhandenen Ventilen, werden fünf Ventile elektrisch und eins mit Druckluft betätigt. Die fünf

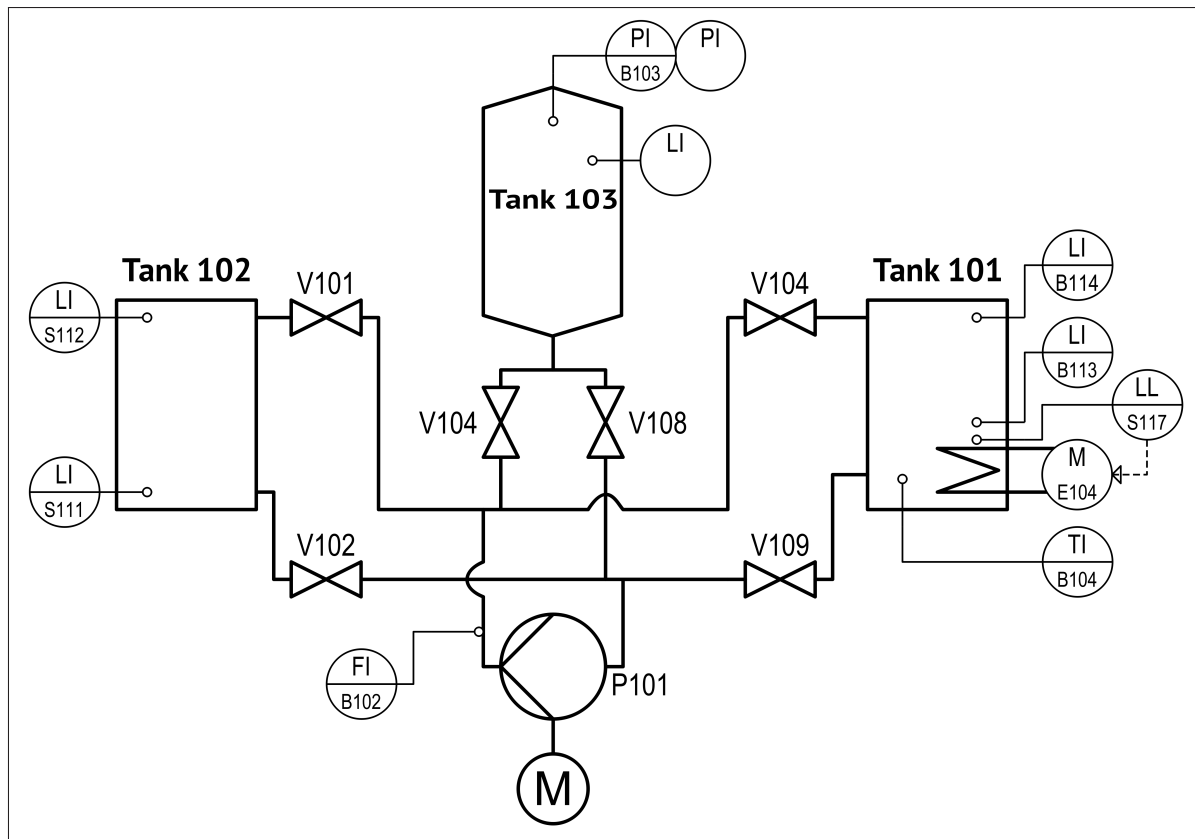


Abbildung 3.1: R&I-Fließbild der Laboranlage

elektromagnetische Ventile besitzen über keine weitere Sensorik. Das bedeutet, dass es nicht möglich ist zu überprüfen ob die Ventile auch tatsächlich den geforderten Befehl ausgeübt haben, oder ob es zu einem Fehler gekommen ist. Im Gegensatz zu den elektromagnetischen Ventilen besitzt das pneumatisch betriebene Ventil über zwei Endlagensensoren. Durch diese ist es möglich zu überprüfen ob sich das Ventil gemäß den Vorgaben entweder vollständig geschlossen oder geöffnet hat. Die Tabelle 3.1 listet alle Aktoren und Sensoren, die im Tank 101 verbaut sind, auf und gibt Auskunft über die verwendete Bezeichnung. Zusätzlich wird angegeben ob der betreffende Sensor Informationen an das MES oder an das HMI weitergibt oder ob der Aktor Informationen von ihnen erhält.

3.1.2 Tank 101

Der Tank 101, der die Rolle der *Maischepfanne* und der *Würzepfanne* übernimmt, muss in der Lage sein Wasser zu erhitzen. Um dieser Aufgabe gerecht zu werden verfügt der Behälter über ein Heizelement und über einen Temperatursensor. Um zu verhindern, dass das Heizelement angeschaltet wird wenn sich nicht genügend Wasser im Behälter befindet ist ein Schwimmer eingebaut, der das Heizelement im Notfall abschaltet. Zudem sind zwei weitere Näherungssensoren auf verschiedenen Höhen verbaut, die es ermöglichen den Füllstand zu identifizieren. In Tabelle 3.2 werden alle Sensoren und Aktoren, die im Tank 101 verbaut sind, aufgelistet.

3.1.3 Tank 102

Der Tank 102 übernimmt im Brauprozess die Rolle des *Läuterbottichs* und des *Whirlpools*. Dazu besitzt er lediglich zwei unterschiedliche Schwimmer, die es ermöglichen den Füllstand zu erkennen. Die Tabelle 3.3 zeigt alle Sensoren, die im Tank 102 verbaut sind.

3.1.4 Tank 103

Wie schon oben angesprochen wird der Tank 103 in diesem Szenario nicht benutzt. Der Behälter besitzt einen Drucksensor der ein digitales Signal ausgibt, sowie eine analoge Druckanzeige an der der aktuelle Druck gleich an der Anlage abgelesen werden kann. Anders als die beiden anderen Behälter kann der Drucktank nur von unten befüllt und entleert werden. Die Tabelle 3.4 listet alle Sensoren, die im Tank 103 verbaut sind, auf.

Tabelle 3.1: Pumpe und Ventile

Funktion	Bezeichnung	MES	HMI
Pumpe	P101	Nein	Nein
Durchflusssensor	B113	Ja	Ja
El. Ventil	B105	Nein	Nein
El. Ventil	V101	Nein	Nein
El. Ventil	V104	Nein	Nein
El. Ventil	V108	Nein	Nein
El. Ventil	V109	Nein	Nein
Pneumat. Ventil	V102	Nein	Nein
Endlagensensor	S115	Nein	Nein
Endlagensensor	S116	Nein	Nein

Tabelle 3.2: Sensoren und Aktoren in Tank 101

Funktion	Bezeichnung	MES	HMI
Füllstandsensor	B113	Ja	Ja
Füllstandsensor	B114	Ja	Ja
Temperatursensor	B104	Ja	Ja
Heizelement	E104	Nein	Nein

Tabelle 3.3: Sensoren in Tank 102

Funktion	Bezeichnung	MES	HMI
Füllstandsensor	S111	Nein	Nein
Füllstandsensor	S112	Nein	Nein

Tabelle 3.4: Sensoren in Tank 103

Funktion	Bezeichnung	MES	HMI
Drucksensor digital	B103	Nein	Nein
Drucksensor analog	/	Nein	Nein

3.2 Vergleich zu einem industriellen 2-Tank-Sudhauses

Da diese Laboranlage den Prozess in einem 2-Tank-Sudhaus darstellen soll wird in diesem Unterkapitel auf die Gemeinsamkeiten, aber auch die Unterschiede, zwischen der Laboranlage und einer industriellen Anlage hingewiesen.

Gemeinsamkeiten:

- Genau wie bei einem realen 2-Tank Sudhaus, besitzt auch die Laboranlage 2 verschiedene Behälter.
- Wie bei der realen Anlage sind diese zwei Behälter mittels eines Rohrleitungssystem miteinander verbunden, welches es erlaubt die Flüssigkeit von einem Tank in den jeweils anderen Tank zu pumpen.
- Es besteht die Möglichkeit die Flüssigkeit in einem der Behälter der Laboranlage zu erhitzen.
- Die Füllstandsensoren und der Temperatursensor ermöglichen es den Füllstand in den beiden Behältern, sowie die Temperatur in Tank 101 auszulesen.

Unterschiede :

- Bei der Laboranlage existieren nur die drei Einheiten *Pumpe und Ventile, Tank 101* und *Tank 102*. Bei der realen Anlage gibt es darüber hinaus auch noch die Einheit *Wasserversorgung*, mit der es unter anderem möglich ist die Maischepfanne mit Wasser zu befüllen.
- Der Tank 101 der Laboranlage wird mittels eines elektrischen Heizelementes geheizt, während in Brauanlagen Heißdampf dafür eingesetzt wird.
- Es gibt in den beiden Behältern der Laboranlage keine Rührwerke. Aus diesem Grund muss die Flüssigkeit, im Gegensatz zu realen Anlagen, durch Umwälzen durchmischt werden.

3.3 Grundlagen Würzeherstellung

In diesem Unterkapitel wird der Vorgang der Würzeherstellung kurz erklärt, damit die folgenden Schritte verständlich werden. Die Würzeherstellung lässt sich grob in fünf Teilvorgänge gliedern. In dieser Arbeit werden lediglich die ersten vier Vorgänge *Maischen*, *Läutern*, *Würzen* und *Kühlen und Klären* betrachtet. Der letzte Vorgang, die *Gärung*, kann leider nicht mit der Laboranlage abgebildet werden und wird somit außen vor gelassen. In der Abbildung 3.2 wird anhand einer formalisierten Prozessbeschreibung die Würzeherstellung gezeigt.

Die verschiedenen Vorgänge werden kurz erläutert [23]:

- Zum **Maischen** wird das geschrotete Malz in der Maischepfanne mit dem Brauwasser vermischt. Während des Maischens bauen malzeigene Enzyme die Stärke und das Protein des Malzes ab und machen große Anteile wasserlöslich. Da die Enzyme bei unterschiedlichen Temperaturen optimal arbeiten, wird die Temperatur im Maischepfanne bis ca. 75 °C hochgefahren.
- Der nächste Schritt auf dem Weg zum Bier ist der Läuterbottich. Beim **Läutern** wird das Extrakt von den unlöslichen Bestandteilen (Treber) getrennt. Dazu wird die Maische im Läuterbottich stehengelassen, bis sich der Treberkuchen auf dem eingelegten Senkboden absetzt. Anschließend kann die Flüssigkeit abgepumpt werden.
- Danach wird die extraktreiche Würze in eine Würzepfanne abgelassen, wo sie anschließend mit Hopfen gemischt wird. In der Würzepfanne wird die Würze, während des **Würzekochens**, auf die gewünschte Konzentration eingedampft. Bei diesem Vorgang wird die Würze pasteurisiert und der gewünschte Stammwürzegehalt eingestellt. Dieser bestimmt den Alkoholgehalt des fertigen Bieres.
- Die beim Würzekochen gebildeten Trubstoffe werden beim anschließenden Schritt, der **Klärung** im Whirlpool, mit Filtern abgeschieden. Danach wird die Würze mit Plattenwärmetauschern möglichst schnell abgekühlt. Bevor die abgekühlte Würze in die Gärbehälter abgepumpt werden kann wird abschließend noch die Hefe hinzugefügt

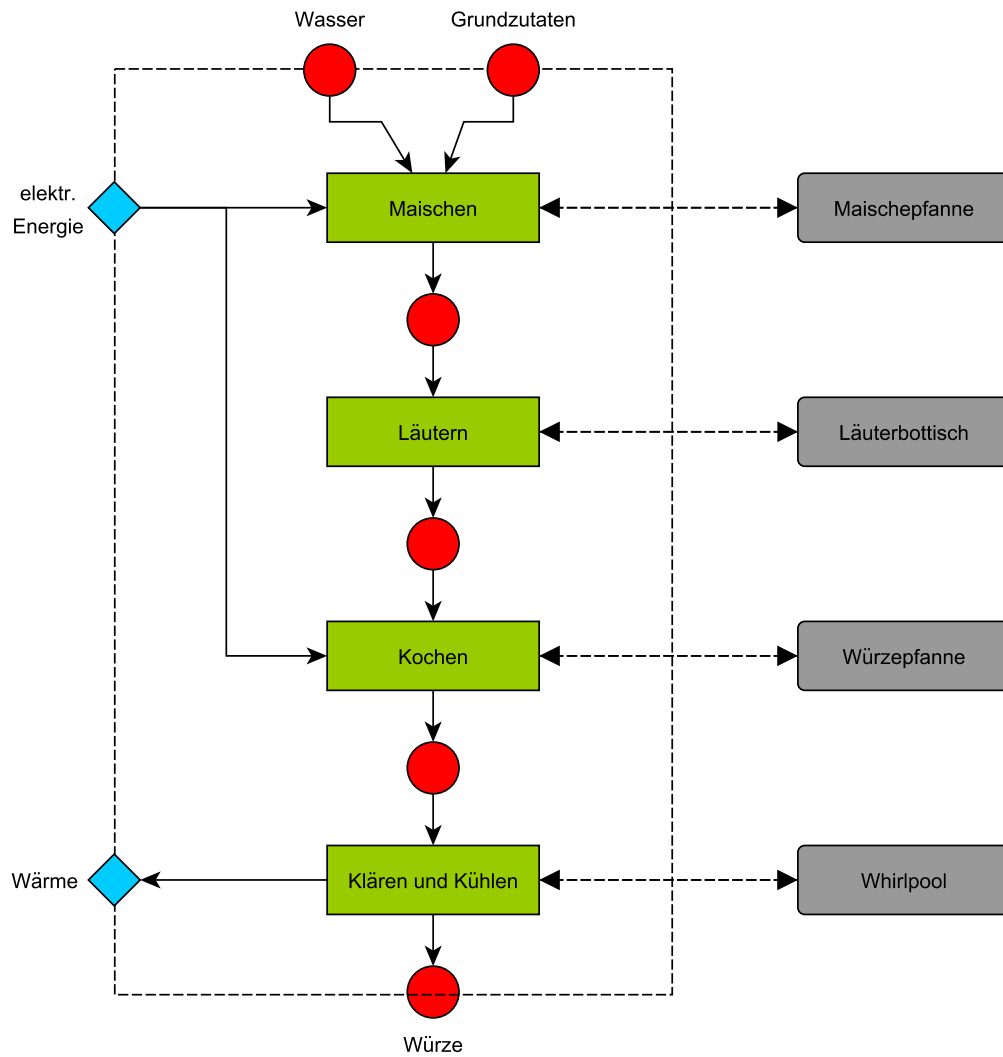


Abbildung 3.2: Formalisierte Prozessbeschreibung der Würzeherstellung

4 Modularisierung und Programmierung der Laboranlage

In diesem Kapitel wird erläutert wie der Steuerungscode für die Laboranlage erstellt wird. Dazu wird das Verfahren der Würzeherstellung zunächst in seine einzelnen Prozessschritte zerlegt. Anhand der aus diesem Schritt gewonnen Informationen werden schließlich die einzelnen Module festgelegt. Danach kann für jedes Modul eine eigne Schnittstelle definiert werden. Als nächstes wird der grundlegende Aufbau eines Moduls in *Codesys*, und im speziellen der der Aufbau der Process Cell aufgezeigt. Nachdem die Anbindung an das MES erklärt wurde, wird aufgezeigt wie die fertige HMI erzeugt wird.

4.1 Analyse und Dokumentation des Verfahrens

In einem ersten Schritt wird das Verfahren der Würzeherstellung analysiert. Dabei wird ein Top-Down Ansatz verfolgt, der den betrachteten Prozess immer weiter in kleinere Einheiten zerlegt, bis schließlich die einzelnen Prozessschritte festgelegt sind. Dazu kann in der folgenden Reihenfolge vorgegangen werden:

- 1.) Als erstes soll der Prozess in seine einzelnen Prozessabschnitte zerlegt werden [10]. Zusätzlich sollen die Übergangsbedingungen und Abläufe (parallel, sequentiell) der einzelnen Prozessabschnitte festgelegt werden [19].
- 2.) Daraufgehend sollen die einzelnen Prozessoperationen der Prozessabschnitte bestimmt werden. Auch hier müssen die Ablauffolgen und die Übergangsbedingungen definiert werden [19].
- 3.) Anschließend können die einzelnen Prozessoperationen in ihre Prozessschritte aufgeteilt werden [10]. Zusätzlich sollen die benötigten Einrichtungen festgelegt werden [19].

Das Ergebnis dieser Analyse ist das Verfahrensrezept [19].

Diese Analyse wird nun am Prozess der Würzeherstellung angewendet. Dazu wird der Prozess in dem folgenden Unterkapitel weiter detailliert. Die fertige Aufteilung wird in der Abbildung 4.1 graphisch dargestellt.

4.1.1 Ermittlung der notwendigen Prozessabschnitte

Der Prozess *Würze herstellen* kann in vier Prozessabschnitte unterteilt werden. Diese lauten:

- 1.) *Maischen*
- 2.) *Läutern*
- 3.) *Kochen*
- 4.) *Klären und Kühlen*

Dabei ist zu beachten, dass alle Prozessabschnitte sequentiell ablaufen. Das bedeutet sobald der Prozessabschnitt *Maischen* abgeschlossen ist, startet der Abschnitt *Läutern*. Ist dieser wiederum abgeschlossen, läuft der Prozessabschnitt *Kochen* und anschließend der Prozessabschnitt *Klären und Kühlen* an. Da der Prozess in zwei Tanks stattfindet, laufen mehrere Prozessabschnitte im gleichen Tank ab. Aus diesem Grund ist die Maischepfanne gleich der Würzpfanne und der Läuterbottisch gleich dem Whirlpool, in dem der Prozessabschnitt *Klären und Kühlen* stattfindet.

4.1.2 Festlegung der Prozessoperationen

Die einzelnen Prozessabschnitte müssen noch weiter detailliert und damit in verschiedene Prozessoperationen unterteilt werden:

- Der Prozessabschnitt **Maischen** wird in die zwei Operationen *Befüllen* und *Erhitzen* unterteilt. Beim Befüllen werden die nötigen Rohstoffe zum Maischen in den Maischepfanne eingefüllt. Jedoch ist diese Prozessoperation an der Laboranlage nicht darstellbar, da die nötige Wasserversorgung und Einfüllmöglichkeit für das Malzschrot nicht vorhanden sind. Bei der Prozessoperation *Erhitzen* wird die Flüssigkeit auf eine vorgegebene

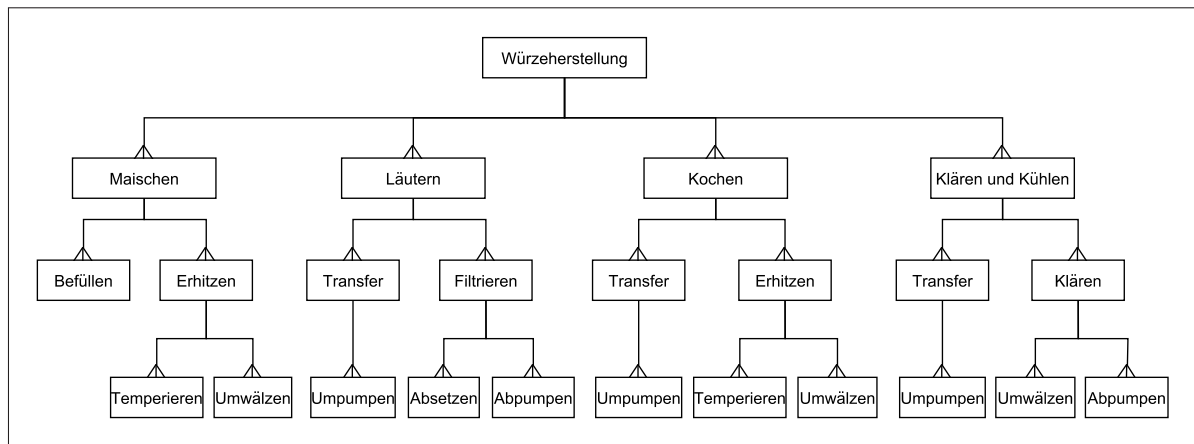


Abbildung 4.1: Prozess Hierarchie

Temperatur erhitzt und nachfolgend für eine gegebene Zeit auf dieser Temperatur gehalten.

- Der Prozessabschnitt **Läutern** kann in die zwei Operationen *Transfer* und *Filtrieren* gegliedert werden. Beim Transfer wird die Maische von der Maischepfanne in den Läuterbottisch umgepumpt, wo während der Operation Filtrieren die festen und flüssigen Bestandteile voneinander getrennt werden.
- Der Prozessabschnitt **Kochen** beinhaltet die zwei Operationen *Transfer* und *Erhitzen*. Beim Transfer wird die Flüssigkeit aus dem Läuterbottisch wieder zurück in die Würzepfanne, welche identisch mit der Maischepfanne ist, gefüllt. Bei der anschließenden Operation Erhitzen wird die Flüssigkeit, genau wie beim Maischen, auf eine vorgegebene Temperatur erhitzt und dann für eine bestimmte Zeit bei der Temperatur gehalten.
- Der Prozessabschnitt **Klären und Kühlen** kann in die drei Prozessoperationen *Transfer*, *Klären* und *Kühlen* unterteilt werden. Auch hier wird beim Transfer die Flüssigkeit zunächst umgepumpt. Nur dieses Mal von der Würzepfanne in den Whirlpool. Im Anschluss an die Operation Transfer wird die Flüssigkeit, während der Prozessoperation Klären, von den vorhandenen Trubstoffen befreit. Normalerweise müsste im Anschluss noch die Prozessoperationen Kühlen ablaufen, jedoch ist auch diese nicht mit der Laboranlage darstellbar.

4.1.3 Festlegung der Prozessschritte

Nun müssen die einzelnen Prozessoperationen, die vorher definiert wurden, noch weiter detailliert werden. Dazu wird jede Operation in mindestens einen Prozessschritt unterteilt:

- Die Prozessoperation **Erhitzen**, die in den Prozessabschnitten *Maischen* und *Kochen* vorkommt, beinhaltet die zwei parallel ablaufenden Prozessschritte *Temperieren* und *Umwälzen*. Beim *Temperieren* wird die Flüssigkeit auf eine vorgegebene Solltemperatur erhitzt und solange auf dieser gehalten bis der Schritt beendet wird. Dazu wird eine Heizung, um die Flüssigkeit zu erwärmen, und ein Temperatursensor, der die aktuelle Temperatur im Behälter misst, benötigt. Da die Laboranlage über keinen Rührer verfügt um die Flüssigkeit beim Erhitzen zu durchmischen, wird dies durch den Schritt *Umwälzen* ersetzt. Dabei wird die Flüssigkeit an der Unterseite des Tanks durch die Pumpe angesaugt und gleich wieder am oberen Einlass des gleichen Tanks eingeleitet. Dadurch bleibt die Flüssigkeit in ständiger Bewegung und die Wärme wird homogen verteilt. Für diesen Schritt wird eine Pumpe mit zwei Ventilen, welche an dem betreffenden Tank angeschlossen sind, benötigt.
- Die Prozessoperation **Transfer**, die ein Teil der Prozessabschnitte *Läutern*, *Kochen* und *Klären und Kühlen* ist, beinhaltet nur den Prozessschritt *Umpumpen*. Dabei wird die Flüssigkeit in einen anderen Behälter geleitet. Hierzu wird eine Pumpe, sowie für jeden Tank jeweils zwei Ventile benötigt. Die Ventile sind so angeordnet, dass es möglich ist von jedem Tank in den jeweils anderen zu Pumpen.
- Die Prozessoperation **Filtrieren** besteht aus den sequentiell ablaufenden Prozessschritten *Absetzen* und *Abpumpen*. Zuerst wird während dem Prozessschritt *Absetzen* nur abgewartet bis sich die festen Bestandteile der Maische auf einem Sieb unten am Boden des Behälters absetzen. Dadurch wird es möglich den flüssigen Teil der Maische im nachfolgenden Prozessschritt *Abpumpen* abzutrennen.
- Die Prozessoperation **Klären** besteht aus den sequentiell ablaufenden Prozessschritten *Umwälzen* und *Abpumpen*. Beim realen Brauprozess würden sich die beim Kochen gebildete Trubstoffe durch eine Rotation absetzen und die klare Würze könnte abgepumpt werden. Da im Whirlpool auch kein Rührer zu Verfügung steht, wird dies auch hier wieder durch den Schritt *Umwälzen* simuliert. Im Anschluss kann die Würze dann abgepumpt werden.

4.2 Modularisierung der Laboranlage

Bei der Analyse des Verfahrens wurden teilweise schon Anforderungen an die verwendete Anlage gestellt. Im nächsten Schritt wird die Anlage gemäß dieser Anforderungen in Teilanlagen zerlegt. Anschließend können die einzelnen Teilanlagen in technische Einrichtungen aufgeteilt werden, welche wiederum in Einzelsteuereinheiten unterteilt werden können. Die fertige Unterteilung wird in Abbildung 4.2 übersichtlich dargestellt

4.2.1 Unterteilung der Laboranlage in Teilanlagen

Um die Anlage in die verschiedenen Teilanlagen unterteilen zu können, kann das eben definierte Verfahrensrezept zu Rate gezogen werden. Die darin festgelegten Prozessabschnitte können als Leitlinien zur Grenzziehung der einzelnen Teilanlagen verwendet werden. Ein typischer Leitgedanke könnte sein: Welche parallelen Abläufe können in der vorhandenen Anlagenstruktur realisiert werden [19]?

Folgend wird die Laboranlage in sinnvolle Teilanlagen zerlegt. Neben dem oben genannten Leitsatz soll zusätzlich darauf geachtet werden, dass maximal eine Charge pro Teilanlage verarbeitet werden kann. Bei der Bezeichnung einer Teilanlage wird die Konvention getroffen, dass zusätzlich die Bezeichnung „Unit“ verwendet wird.

Abbildung 4.3 zeigt die verwendeten Komponenten der Laboranlage, die im Anschluss in drei Teilanlagen zerlegt wird:

- Die erste Teilanlage bildet die *Tank 101 Unit*. Auf ihr werden die beiden Prozessabschnitte *Maischen* und *Kochen* teilweise realisiert.
- Als zweite Teilanlage wird der *Tank 102 Unit* definiert. Auf ihr werden die beiden Prozessabschnitte *Läutern* und *Klären und Kochen* teilweise ablaufen.
- Diese beiden Teilanlagen werden durch die dritte Teilanlage, die *Pump Unit*, miteinander verbunden. Sie übernimmt den Transport der Flüssigkeit zwischen den beiden Tanks.

Abbildung 4.4 zeigt die Aufteilung der Laboranlage anhand eines R&I-Fließbildes. Jede Teilanlage wird einzeln dargestellt, wobei die Verbindungen zwischen den Teilanlagen mit Übergangspfeilen gekennzeichnet werden. Die Bezeichnung der Übergangspfeile ist so gewählt, dass sofort ersichtlich ist welche Verbindung dargestellt wird. So kennzeichnet zum Beispiel *UP_UT2_1*, dass die erste Verbindung von der *Pump Unit* zur *Tank 102 Unit* führt.

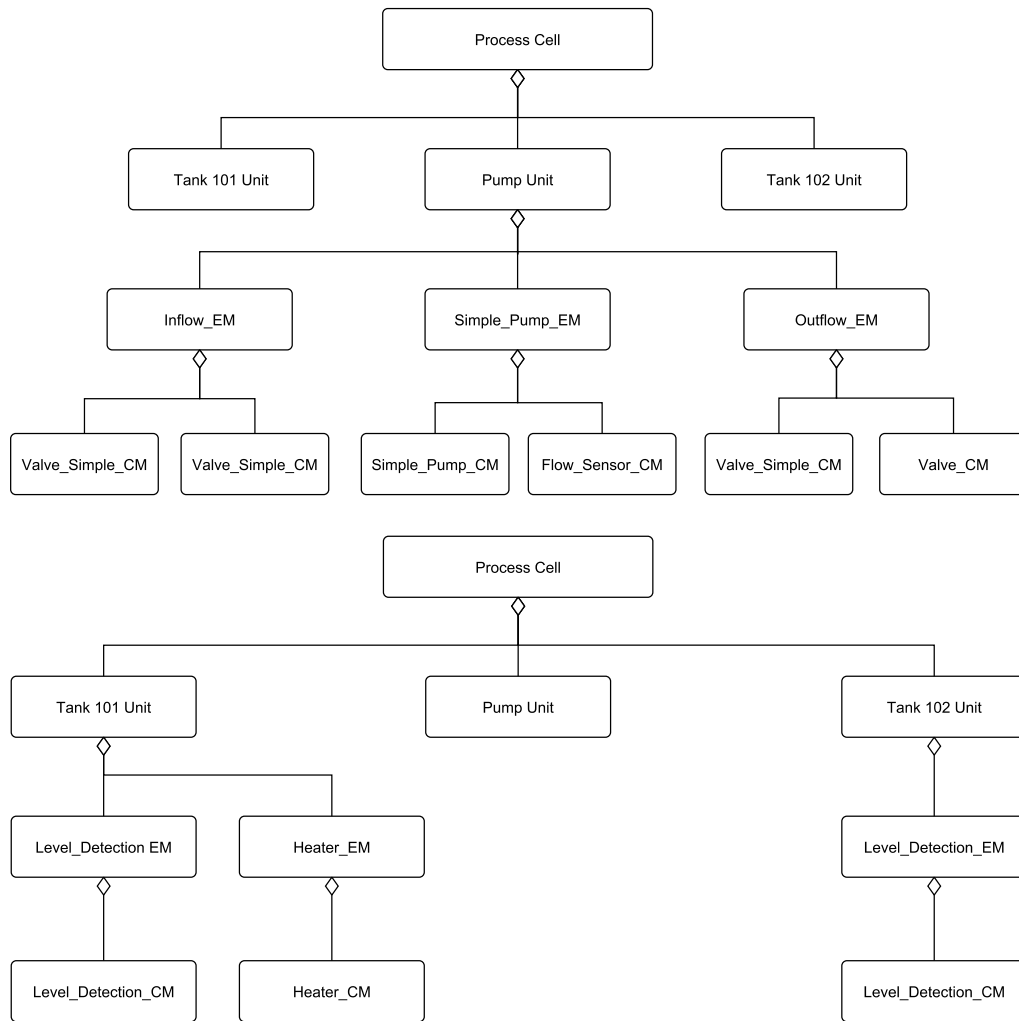


Abbildung 4.2: Aufbau der *Pump Unit*, der *Tank 101 Unit* und der *Tank 102 Unit*

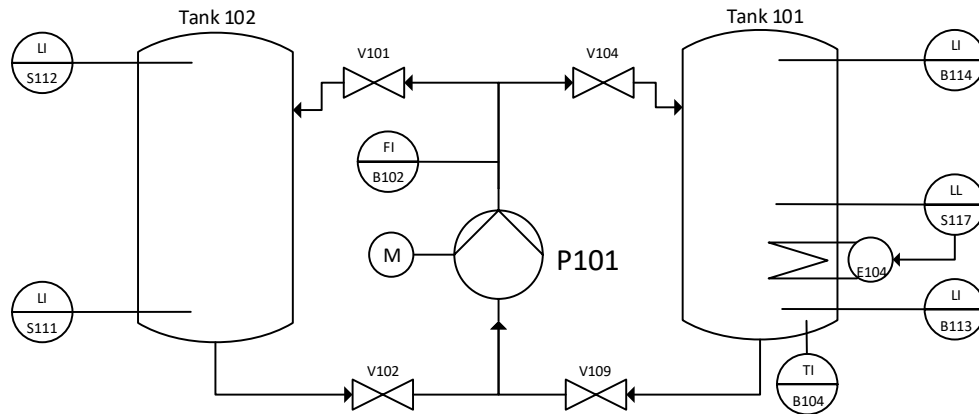


Abbildung 4.3: R&I-Fließbild der verwendeten Komponenten der Laboranlage

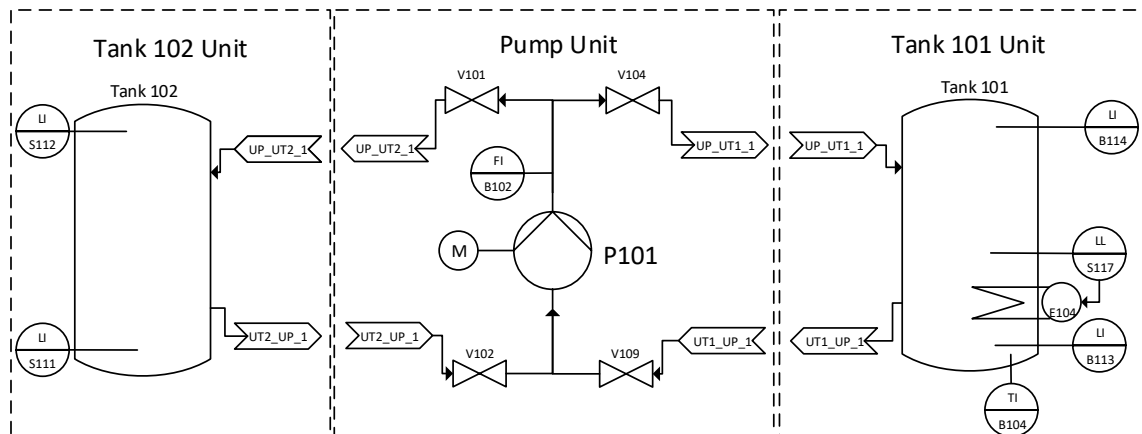


Abbildung 4.4: Aufteilung der Laboranlage in Teilanlagen

4.2.2 Unterteilung der Teilanlagen in technische Einrichtungen

Nachdem die Laboranlage in die drei Teilanlagen *Pump Unit*, *Tank 101 Unit* und *Tank 102 Unit* aufgeteilt wurde, müssen diese noch weiter in technische Einrichtungen unterteilt werden. Bei der Benennung der einzelnen technischen Einrichtungen wird das Kürzel *EM* (engl: Equipment Module) verwendet.

Die *Pump Unit* hat die Aufgabe die Flüssigkeit zwischen den einzelnen Behältern hin und her zu pumpen. Um dieser Aufgabe gerecht zu werden, werden folgende drei technischen Einrichtungen definiert, welche auch in Abbildung 4.5 (linkes R&I-Fließbild) dargestellt sind:

- Die technische Einrichtung *Inflow_EM* steuert die beiden Ventile, die für den Zufluss in die beiden Behälter verantwortlich sind. In der Ausgangssituation sollen beide Ventile geschlossen sein. Wird jedoch verlangt, dass in einen der beiden Behälter gepumpt werden soll, muss sich das entsprechende Ventil öffnen. Dazu kriegt die technische Einrichtung mit zwei Methoden vorgegeben welches Ventil geöffnet oder geschlossen werden muss.
- Die technische Einrichtung mit der Benennung *Outflow_EM* ist ähnlich zur technischen Einrichtung *Inflow_EM*. Nur werden hier anstelle der beiden Zuflussventile, die zwei Abflussventile gesteuert.
- Zusätzlich zu den beiden technischen Einrichtungen, die die Ventile steuern wird auch noch eine technische Einrichtung für die Pumpe benötigt. Es ist möglich die eingebaute Pumpe in zwei Betriebsmodi zu betreiben. Im ersten Modus kann die Pumpe entweder nur vollständig Ein- oder Ausschalten werden. Im zweiten Modus ist es jedoch auch möglich den Durchfluss der Pumpe zu regeln. Je nach gewähltem Betriebsmodus werden unterschiedliche Anforderungen an die technische Einrichtung gestellt. Aus diesem Grund werden zwei technische Einrichtungen festgelegt, welche dann jeweils einen Betriebsmodus abdecken:
 - Einfache Pumpe, die nur Ein- und Ausgeschaltet werden kann:
Für diesen Fall wird eine technische Einrichtungen mit der Bezeichnung *Simple_Pump_EM* definiert. Der technischen Einrichtungen wird der Soll-Durchfluss, von der übergeordneten Teilanlage übergeben. Da dieses Modul den Durchfluss jedoch nicht regeln kann wird die Pumpe, bei einem Soll-Durchfluss größer als Null, der Befehl zum Einschalten übergeben. Zudem soll der aktuelle Durchfluss,

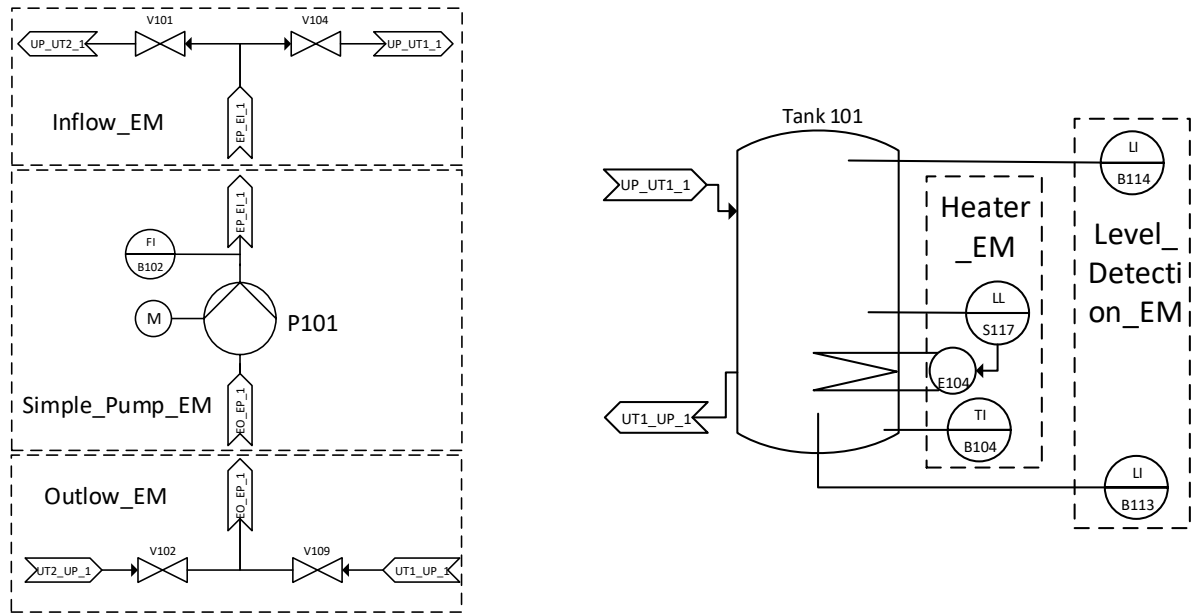


Abbildung 4.5: Ablaufdiagramm der technischen Funktionen

der an der Pumpe gemessen wird, an die übergeordnete Teilanlage übergeben werden. Um diese Aufgaben zu erfüllen besitzt das *Simple_Pump_EM* neben der Pumpe noch über einen Durchflusssensor.

– Geregelte Pumpe:

Für diesen Betriebsmodus wird eine technische Einrichtungen mit der Bezeichnung *Regulated_Pump_EM* festgelegt. Dem Modul wird der Soll-Durchfluss, von der übergeordneten Teilanlage, vorgegeben. Mithilfe des aktuellen Durchflusses an der Pumpe kann die benötigte Auslastung der Pumpe, mittels einer Regelung, errechnet werden. Das *Regulated_Pump_EM* soll zusätzlich den aktuellen Durchfluss an die übergeordnete Teilanlage übergeben. Dazu wird, wie bei dem *Simple_Pump_EM*, eine Pumpe und ein Durchflusssensor benötigt.

Die *Tank 101 Unit* soll zwei Aufgaben übernehmen, für die jeweils eine technische Einrichtung definiert wird. Die Aufteilung der *Tank 101 Unit* wird zusätzlich in Abbildung 4.5 (rechtes R&I-Fließbild) dargestellt:

- Zum einen soll die im Tank 101 befindliche Flüssigkeit auf die vorgegebene Solltemperatur erhitzt und anschließend für eine gegebene Zeit auf dieser Temperatur gehalten

werden. Dazu verfügt die Laboranlage über ein elektrisches Heizelement, was nur über die beiden Zustände An und Aus verfügt, sowie über ein Temperatursensor. Um diese Aufgabe zu erfüllen wird eine technische Einrichtung mit der Bezeichnung *Heater_EM* festgelegt. Jedoch übernimmt die technische Einrichtung keine Steuerung der Laboranlage, sondern lagert diese an die dazugehörigen Einzelsteuereinheit, welche im nächsten Kapitel vorgestellt wird, aus.

- Die technische Einrichtung mit der Bezeichnung *Level_Detection_EM* soll den aktuellen Füllstand im Tank 101 ausgeben. Dazu werden zwei Füllstandsensoren benötigt. Das *Level_Detection_EM* übernimmt, ähnlich wie beim *Heater_EM*, auch keine Steuerungsfunktion. Diese wird in der dazugehörige Einzelsteuereinheit übernommen.

Zur Aufgabe der *Tank 102 Unit* gehört lediglich die Füllstanderkennung. Die Aufgabenstellung sowie die Hardware ist ähnlich wie in der *Tank 101 Unit*. Dies erlaubt eine Wiederverwendung des *Level_Detection_EM* aus der *Tank 101 Unit*.

4.2.3 Unterteilung der technischen Einrichtungen in Einzelsteuereinheiten

Nachdem nun alle Teilanlagen in einzelne technische Einrichtungen zerlegt wurden, können diese noch weiter in Einzelsteuereinheiten eingeteilt werden. Dabei wird der Bezeichnung der verschiedenen Einzelsteuereinheiten das Kürzel *CM* (engl: Control Module) verwendet.

- Die in der *Pump Unit* verwendete technische Einrichtung *Inflow_EM* besteht aus zwei identischen Einzelsteuereinheiten mit der Bezeichnung *Valve_Simple_CM*. Dabei steuert eine Einzelsteuereinheit jeweils eines der beiden verwendeten Ventile. Soll ein Ventil geöffnet oder geschlossen werden, wird die entsprechende Einzelsteuereinheit über zwei Methoden angesteuert. Bei den beiden verwendeten Ventilen handelt es sich um einfache Ventile ohne Endlagensensoren. Deswegen kann leider nicht überprüft werden ob das entsprechende Ventil den gewünschten Befehl ausgeführt hat.
- Ähnlich wie beim *Inflow_EM*, verfügt das *Outflow_EM* über zwei Einzelsteuereinheiten. Eine Einzelsteuereinheit trägt die Bezeichnung *Valve_Simple_CM* und ist identisch zu den beiden Einzelsteuereinheiten im *Inflow_EM*. Auch hier steuert das *Valve_Simple_CM* das einfache Ventil ohne Endlagensensoren. Zusätzlich zu dem einfachen Ventil, verfügt das *Outflow_EM* über ein Ventil, welches mit zwei Endlagensensoren ausgestattet ist. Dies erlaubt eine Überprüfung ob sich das Ventil auch tatsächlich geöffnet oder geschlossen hat, oder ob ein Fehler vorliegt. Aus dem Grund wird für dieses Ventil eine Einzelsteuereinheit mit dem Namen *Valve_CM* festgelegt.

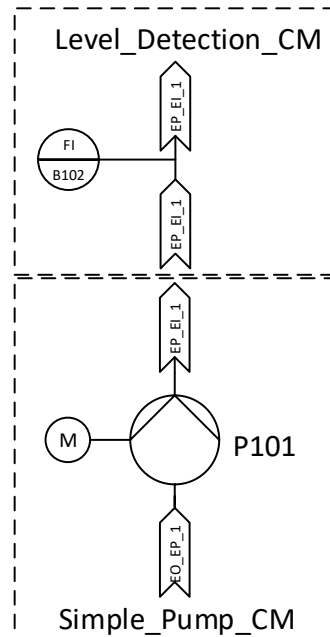


Abbildung 4.6: Aufteilung des *Pump_EM* in Einzelsteuereinheiten

- Das *Simple_Pump_EM* wird in zwei Einzelsteuereinheiten eingeteilt. Diese lauten *Flow_Sensor_CM* und *Simple_Pump_CM* (Siehe Abbildung 4.6). Das *Flow_Sensor_CM* wandelt den vom Durchflusssensor ausgegebenen Wert in die Einheit l/min (Liter pro Minute) um, welcher dann an das *Simple_Pump_EM* übergeben wird. Das *Simple_Pump_CM* steuert die verwendete Pumpe. Je nach Befehl der technischen Einrichtung soll die Einzelsteuereinheit die Pumpe entweder ein- oder ausschalten.
- Ähnlich wie bei dem *Simple_Pump_EM*, wird auch das *Regulated_Pump_EM* in zwei Einzelsteuereinheiten unterteilt. Eine Einzelsteuereinheit bildet das *Flow_Sensor_CM*, welche identisch zu dem im *Simple_Pump_EM* verwendeten Modul ist. Das *Flow_Sensor_CM* liefert der übergeordneten technischen Einrichtung den aktuellen Durchfluss an der Pumpe. Mithilfe dieses Wertes, kann das *Regulated_Pump_EM* die aktuelle Auslastung der Pumpe errechnen. Dieser errechnete Wert wird an die Einzelsteuereinheit mit der Bezeichnung *Regulated_Pump_CM* übergeben, welche die Pumpe mit der gewünschten Auslastung ansteuert.

- Wie schon oben angesprochen übernimmt das *Heater_EM* keine Steuerungsfunktion, sondern es lagert diese an Einzelsteuereinheit aus. Diese trägt die Bezeichnung *Heater_CM*. Die technische Einrichtung übergibt der Einzelsteuereinheit die gewünschte Solltemperatur des Wassers im Tank 101 sowie die Zeitspanne, für die die Temperatur gehalten werden soll. Mithilfe des vom Temperatursensor gemessenen Wertes, kann das *Heater_CM* das Heizelement je nach Bedarf ein- und ausschalten.
- Das *Level_Detection_EM* übernimmt, wie beim *Heater_EM*, auch keine Steuerungsfunktion. Die Steuerung wird in der Einzelsteuereinheit mit der Bezeichnung *Level_Detection_CM* ausgeführt. Zur Füllstanderkennung werden zwei Füllstandssensoren benötigt, die an zwei verschiedenen Positionen am Tank angebracht sind. Dabei ist es nebensächlich wie die Sensoren den Füllstand ermitteln. Es macht also keinen Unterschied ob diese mit einem Schwimmer, wie im Tank 102, oder per Näherungssensor, wie im Tank 101, arbeiten. Mittels der beiden Sensorwerte kann ermittelt werden ob der betrachtete Behälter Voll oder Leer ist. Dies ist wichtig um zu verhindern, dass der Tank überläuft oder die Pumpe weiter läuft obwohl sich keine Flüssigkeit mehr im Behälter befindet.

4.3 Definition der Schnittstellen

Nachdem die einzelnen Teilanlagen, technischen Einrichtungen und Einzelsteuereinheiten definiert wurden, müssen die Schnittstellen der einzelnen Module festgelegt werden. Diese werden benötigt, da das Modul der nächst höheren Ebene üblicherweise mit denen ihm unterlagerten Modulen mit Hilfe von Methodenaufrufen und Parameterübergaben kommuniziert [2]. Damit eine Schnittstelle sofort an ihrem Namen erkennbar ist, beginnt dessen Bezeichnung immer mit einem vorangestellten *I*. In der Abbildung 4.7 werden die gleich erläuterten Schnittstellen der einzelnen Module graphisch dargestellt.

Zunächst werden die drei Schnittstellen *IUnit*, *IEquipment* und *IControl* definiert. Die Schnittstelle der einzelnen Teilanlagen, technischen Einrichtungen und Einzelsteuereinheiten erben dann entweder von *IUnit*, *IEquipment* oder *IControl*. Dadurch wird es möglich die Anlagenebene des jeweiligen Moduls gleich an der Schnittstelle zu erkennen. Jedes Modul gibt seinen aktuellen Zustand an das ihm überlagerte Modul weiter. Dazu besitzt jede Schnittstelle die Variable *State*.

- **ISimple_Pump_CM**

Das *Simple_Pump_CM* wird über die Methode *Pump(iWorkload)* gesteuert. Die Variable *iWorkload* gibt dabei an ob die Pumpe an- oder ausgeschaltet werden soll. Als Rückgabewert wird die Variable *State* ausgegeben, welche wie folgt definiert ist:

- 0 Fehler
- 1 Pumpe ist aus
- 2 Pumpe ist an

- **IRegulated_Pump_CM**

Das *Regulated_Pump_CM* wird über die Methode *Pump(iWorkload)* gesteuert. Dabei wird mit der Variable *iWorkload* eingestellt mit wie viel Prozent der Maximalleistung die Pumpe betrieben werden soll. Als Rückgabewert wird die Variable *State* ausgegeben, welche wie folgt definiert ist:

- 0 Fehler
- 1 Pumpe ist aus
- 2 Pumpe ist an

- **IFlow_Sensor_CM**

Das *Flow_Sensor_CM* liefert zwei Rückgabewerte. Zum einen wird dem ihm überlagerten *Pump_EM* der aktuelle Durchfluss an der Pumpe, über die Variable *Current_Flow*, übergeben. Dazu wird der aktuelle Sensorwert ausgelesen und in die Einheit l/min (Liter pro Minute) umgerechnet. Zum anderen wird wie bei jedem Modul auch der aktuelle Zustand über die Variable *State* ausgegeben, welche wie folgt definiert ist:

- 0 Fehler
- 1 Sensor läuft einwandfrei

- **IPump_EM**

Dem *Pump_EM* wird über die Methode *Pump(iPump_Output)* aufgerufen. Dazu wird ihm über die Variable *iPump_Output* der gewünschte Durchfluss, in der Einheit l/min, mitgeteilt. Als Parameterübergabe gibt das Modul *Pump_EM* den aktuellen Durchfluss

Current_Flow an der Pumpe und den aktuellen Zustand *State* des Moduls an das überlagerte Modul weiter. Dabei ist *State* wie folgt festgelegt:

- 0 Fehler
- 1 Pumpe ist aus
- 2 Pumpe ist an

- **IValve_CM**

Die Schnittstelle *IValve_CM* wird von den beiden Einzelsteuereinheiten *Valve_CM* und *Valve_simple_CM* verwendet. Diese werden von der übergeordneten technischen Einrichtung über die beiden Methodenaufrufe *Open()* und *Close()* gesteuert. Als Rückgabewert wird auch hier nur der aktuelle Zustand über die Variable *State* geliefert, wobei zwischen folgenden fünf Zuständen unterschieden werden:

- 0 Fehler
- 1 Geschlossen
- 2 Am Schließen
- 3 Offen
- 4 Am Öffnen

Obwohl beide Einzelsteuereinheiten die selbe Schnittstelle benutzen, werden beide vorhandenen Zustände *Am Schließen* und *Am Öffnen* vom *Valve_simple_CM* nicht berücksichtigt, da die benötigte Sensorik zur Überprüfung fehlt. Somit springt beispielsweise der Zustand des Moduls beim öffnen gleich von *Geschlossen* auf *Offen* ohne den dazwischenliegenden Zustand *Am Öffnen* einzunehmen.

- **IInflow_EM**

Die technische Einrichtung *Inflow_EM* wird von der *Pump_Unit* durch die beiden Methoden *OpenValve(iTarget_Tank)* und *CloseValve(iTarget_Tank)* gesteuert. Durch die Variable *iTarget_Tank* wird dem Modul übermittelt, welches Ventil zu öffnen oder zu schließen ist.

Dabei gilt für *iTarget_Tank*:

- 1 Zieltank ist Tank 101
- 2 Zieltank ist Tank 102

Um dem übergeordneten Modul mitzuteilen ob die geforderte Operation ausgeführt wurde verfügt die Schnittstelle über die Variable *State*, welche folgende Zustände annehmen kann:

- 0 Fehler
- 1 In Bearbeitung: Das gewünschte Ventil wird gerade geöffnet oder geschlossen.
- 2 Abgeschlossen: Die gewünschte Operation wurde ausgeführt.

- **IOutflow_EM**

Das *Outflow_EM* besitzt fast die gleiche Schnittstelle wie das *Inflow_EM*, mit dem einzigen Unterschied, dass die Variable *iStart_Tank* anstelle von *iTarget_Tank* übergeben wird.

- **IPump_Unit**

Die *Pump_Unit* steuert die drei technischen Einrichtungen *Pump_EM*, *Inflow_EM* und *Outflow_EM*. Dazu wird sie von der *Process_Cell* über die zwei Methoden *StartPumping(iStart_Tank, iTarget_Tank, iPump_Output)* und *StopPumping()* befehligt. Die Variable *iPump_Output* ist dabei der vorgegebene Wert des Durchflusses in der Einheit l/min. Die Teilanlage wertet die Zustände der untergeordneten technischen Einrichtungen aus und fasst diese in ihrem eigenen Zustand zusammen. Dieser wird mit der Variable *State* übergeben und kann folgende Zustände annehmen:

- 0 Fehler
- 1 in Bearbeitung
- 2 Abgeschlossen

- **ILevel_Detection_CM**

Das *Level_Detection_CM* soll den Füllstand ermitteln. Da dazu keine Methode benötigt wird, werden ausschließlich die beiden Variablen *iFluid_Level* und *State* übergeben.

Dazu kann die Variable *iFluid_Level*, wie schon vorher beschrieben, folgende Werte annehmen:

- 1 : Leer;
- 2 : Füllstand zwischen voll und leer;
- 3 : Voll;

Die Variable *State* unterscheidet lediglich zwischen den beiden Zuständen:

- 0 : Fehler
- 1 : Fehlerfrei

ILevel_Detection_EM besitzt die gleichen Schnittstellenelemente wie *ILevel_Detection_CM*, weil *Level_Detection_EM* keine weitere Funktionalität übernimmt und nur die Parameter und Methodenaufrufe weiterleitet.

• **IHeater_CM**

Das *Heater_CM* wird über die Methode *Heat(iSet_Temp, tTime)* gesteuert. Die Variable *iSet_Temp* gibt dabei die Solltemperatur und *tTime* die Zeit, für die die Temperatur gehalten werden soll an. Ein Rückgabewert bildet die Variable *Current_Temperature* welche die aktuelle Isttemperatur im Tank weitergibt. Zusätzlich wird auch wieder die Variable *State* mit folgenden Zuständen weitergeleitet:

- 0 Fehler: Es ist ein Fehler aufgetreten.
- 1 Idle: Die Einzelsteuereinheit ist bereit.
- 2 Heizen: Das *Heater_CM* heizt das Wasser.
- 3 Temperatur halten: Das Wasser wurde auf die vorgegebene Temperatur erhitzt und wird nun für die Zeit *tTime* auf dieser gehalten.
- 4 Abgeschlossen: Der Heizvorgang ist abgeschlossen und es kann wieder in den Zustand Idle gesprungen werden.

Da die überlagerte technische Einrichtung *Heater_EM* keine weitere Funktion ausführt, leitet diese lediglich die Befehle von der *Tank101_Unit* weiter.

- **ITank101_Unit**

Die *Tank101_Unit* besitzt die Methode *Heat(iSet_Temp,tTime)*. Als Rückgabewert liefert die Teilanlage den aktuellen Füllstand *iFluid_Level*, die aktuelle Temperatur *Current_Temperature* im Tank, sowie den aktuellen Zustand *State*. Der Zustand von *Tank101_Unit* ist , im Fall wo *Level_Detection_CM* fehlerfrei läuft, identisch mit den Zustand vom *Heater_EM*.

- **ITank102_Unit**

Die *Tank102_Unit* besitzt die identische Schnittstelle wie *Level_Detection_CM*. Auch hier werden lediglich die Variablen an die *Process Cell* weitergeleitet

4.4 Aufbau eines Moduls in Codesys

Die einzelnen Module, sowie die dazugehörigen Schnittstellen, können nun in *Codesys* implementiert werden. Dabei wird ein Bottum-Up Ansatz verfolgt bei dem zuerst die Einzelsteuereinheiten erstellt werden, welche dann beim anschließenden Erstellen der verschiedenen technischen Einrichtungen mit eingebunden werden. Nach dem gleichen Schema wird dann auch der Code für die einzelnen Teilanlagen und später auch die Anlage geschrieben. Jedes Modul, was mithilfe des *Application Composer* erstellt wird, kann aus folgenden vier Teilen bestehen:

- *Schnittstelle*: Hier werden die im Kapitel 4.3, für jedes einzelne Modul, definierten Schnittstellen implementiert. Dazu muss die jeweilige Schnittstelle entweder von *IUnit*, *IEquipment* oder *IControl* erben, je nachdem zu welcher Anlagenebene das betreffende Modul gehört. Ist die Schnittstelle angelegt, können die einzelnen Methodenaufrufe und Parameterübergaben festgelegt werden. Bei den Methodenaufrufen muss neben der Bezeichnung definiert werden, welche Variablen übergeben werden. Für die Parameterübergabe wird festgelegt, ob es sich um eine *get*- oder *set*- Variable handelt. Das heißt ob ein Parameter vom Modul ausgelesen, oder ob ihm ein Parameter vorgegeben wird.
- *Moduldeklaration*: In der Moduldeklaration eines Moduls wird unter anderem festgelegt welche untergeordneten Module diesem angehängt werden. Dies wird zum Beispiel am Listing 4.9 ab Zeile 10 gezeigt, wo die technische Einrichtung *Heater_CM* eingehängt

wird. Zusätzlich können die verschiedenen Input- und Output Variablen festgelegt und die Visualisierung eingebunden werden (Siehe Listing 4.8 Zeile Zeile 24). In Listing 4.8 ab Zeile 10 wird gezeigt wie die Input Variable *Temp_Sensor* und die Output Variable *Heater_on* festgelegt werden. Diese können dann nachdem der Steuerungscode erstellt wurde mit der entsprechenden Sensoren und Aktoren der Anlage verknüpft werden. Damit die Moduldeklaration mit dem entsprechenden Programmierobjekt verknüpft wird, muss dieses gleich in der ersten Zeile über den Befehl *IMPLEMENTED_BY* eingebunden werden. In der ersten Zeile von Listing 4.8 wird dies anhand des Moduls *Heater_CM*, welches mit der gleichnamigen Moduldeklaration verknüpft wird, gezeigt.

- *Programmierobjekt*: (engl:Program Organization Unit (POU)) Im Programmierobjekt wird mittels Methoden der eigentliche Steuerungscode implementiert. Diese Methoden werden sollen die Aufgabe des jeweiligen Moduls ausführen. Zudem wird festgelegt welche Variablen über die Schnittstelle ausgegeben werden, wie zum Beispiel der aktuelle Zustand.
- *Visualisierung*: Damit die Benutzeroberfläche automatisch aus der Modulkonfiguration erstellt werden kann, ist es nötig für jedes Modul, insofern dies nötig ist, eine eigene Visualisierung zu erstellen. Die Visualisierung muss dabei alle benötigten Anzeigeelemente für das dazugehörige Modul enthalten. Diese können dann gleich mit den richtigen Variablen im Programmierobjekt des jeweiligen Moduls verknüpft werden. Somit wird es möglich die Anzeigeelemente nach dem Erstellen des Steuerungscode durch einfaches hin- und herziehen zu einer kompletten Benutzeroberfläche zusammenzufügen.

Der eben beschriebene Modulaufbau kann anhand vom Modul *Heater_CM* in der Abbildung 4.10 nochmals verdeutlicht werden. Zu der Schnittstelle *IHeater_CM* gehören die zwei Methodenaufrufe *Heat* und *Reset_Heater* und die zwei Parameterübergaben *Current_Temperature* und *State*. Zusätzlich zu den eben genannten Methoden befindet sich im Programmierobjekt auch noch die Methode *CallPrioMediumStart*, welche nicht in der Schnittstelle definiert ist. Diese Methode muss nämlich nicht von außen aufgerufen werden, sondern wird automatisch jeden Zyklus aufgerufen. Dies hat den Vorteil, dass die Sensorwerte auch ohne Aufruf immer ausgelesen und weitergegeben werden können.

```
1  MODULE Heater_CM IMPLEMENTED_BY Heater_CM
2
3  SEC std.MetaData
4      Desc      := TL.Heater_CM_Desc ;
5      Category  := 'Misc' ;
6      Icon_16   := IP.Heater_CM_Icon_16 ;
7      Icon_32   := IP.Heater_CM_Icon_32 ;
8  END_SEC
9
10 SEC IO
11     SEC Input : Temp_Sensor
12         Variable := wactual_Temp;
13         Name := TL.Temp_Sensor_Name ;
14         Desc := TL.Temp_Sensor_Desc ;
15     END_SEC
16
17     SEC Output : Heater_On
18         Variable := bHeater_On;
19         Name := TL.Heater_On_Name ;
20         Desc := TL.Heater_On_Desc ;
21     END_SEC
22 END_SEC
23
24 SEC std.Visu
25     Page := [ Visu_Heater_CM ];
26     Embedded := [ '%PAGE_VISU%' ];
27 END_SEC
```

Abbildung 4.8: Moduldeklaration des Heater_CM

```
1  MODULE Heater_EM IMPLEMENTED_BY Heater_EM
2
3  SEC std.MetaData
4      Desc      := TL.Heater_EM_Desc ;
5      Category  := 'Misc' ;
6      Icon_16   := IP.Heater_EM_Icon_16 ;
7      Icon_32   := IP.Heater_EM_Icon_32 ;
8  END_SEC
9
10 SEC Slots
11     SEC Slot : pHeater_CM
12         Variable := pHeater_CM;
13         Role := TL.pHeater_CM_Role ;
14         Type := SUBMODULE;
15     END_SEC
16 END_SEC
```

Abbildung 4.9: Moduldeklaration des Heater_EM

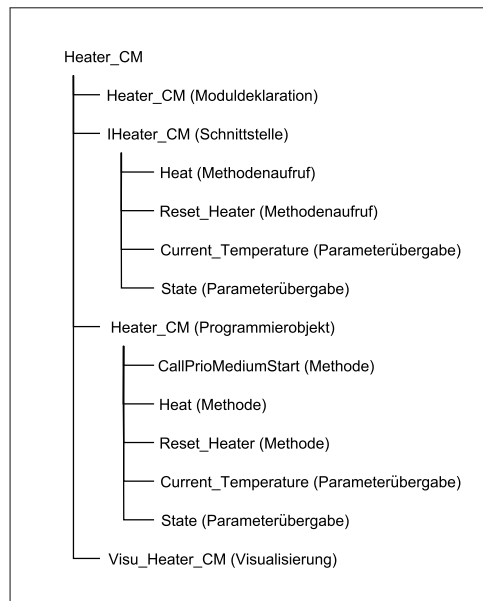


Abbildung 4.10: Aufbau eines Moduls in CoDeSys am Beispiel des Heater_CM

4.5 Aufbau der Process Cell

Das Modul der *Process_Cell* (Anlage) bildet, im Fall der Laboranlage, die oberste Ebene im Anlagenmodell. Sie muss mit keiner übergeordneten Instanz kommunizieren. Aus diesem Grund wird auch keine Schnittstelle benötigt. Als zusätzliche Besonderheit besitzt das Modul, das für den Prozessablauf notwendige Rezept. Um den gespeicherten Prozess zu verwirklichen gibt das Rezept Befehle an Teilanlagenebene weiter. Damit die einzelnen Prozessabschnitte ablaufen können, werden die von den Teilanlagen übermittelten Daten, wie zum Beispiel die Füllstände, benötigt. Wird das Rezept aufgerufen, kann der Bediener mittels einer Eingabe in der Benutzeroberfläche entscheiden, ob die Anlage Initialisiert (Abbildung 4.11 rechter Zweig) oder ob der Prozess der Würzeherstellung gestartet wird (Abbildung 4.11 linker Zweig). Dabei ist zu beachten, dass wenn eine der beiden Aktionen gestartet wird, diese auch bis zum Ende ausgeführt werden muss, und es nicht möglich ist diese zu unterbrechen um die jeweils andere Aktion zu beginnen. Bei der Initialisierung wird das Wasser aus Tank 102 in den Tank 101 gepumpt, weil es bei der Laboranlage nicht möglich ist die Maischepfanne mittels der Wasserversorgung zu befüllen. Ist die Anlage initialisiert, kann mit der eigentlichen Würzeherstellung begonnen werden. Das Rezept besitzt eine ähnliche Unterteilung wie der betrachtete Prozess in Kapitel 4.1. So wurden die definierten Prozessabschnitte in identischer Form im Rezept übernommen. Die vier Abschnitte *Maischen*, *Läutern*, *Kochen* und *Klären und Kühlen* bilden so den linken Ast der Abbildung 4.11. Wenn es sinnvoll ist werden die

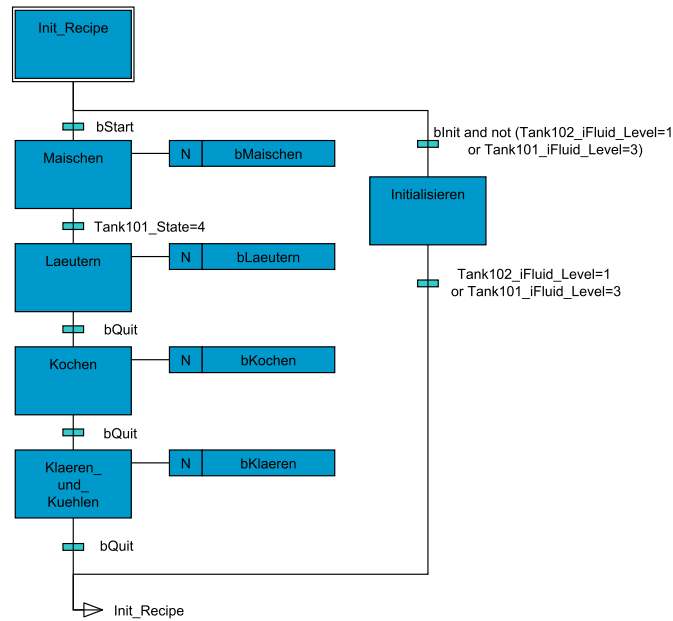


Abbildung 4.11: Rezept der Process_Cell

einzelnen Abschnitte, wie in dem erstellten Prozessmodell festgelegt, auch im Rezept weiter unterteilt. So werden der Prozessabschnitt *Kochen* in die beiden Operationen *Transfer* und *Erhitzen* und der Prozessabschnitt *Klären und Kühlen* in die zwei Operationen *Transfer* und *Klären* aufgeteilt.

4.6 Anbindung an das Manufacturing Execution System

Produktionssysteme müssen unter anderem aufgrund von Produktänderungen und schwankender Bedarfe laufend angepasst werden. In der Praxis führen Änderungen an Produktionsanlagen vor allem zu Anpassungen an der steuernden Software von Maschinen und Anlagen sowie an der Informationstechnik. Zur Kommunikation wird zunächst die IP-Adresse der Steuerung veröffentlicht. Anschließend muss der OPC-Server dem Client bekannt gegeben werden. Im darauffolgenden Schritt müssen alle vom OPC-Server bereitgestellten Variablen ausgewählt und mit dem vorab erstellten Prozessabbild verbunden werden.[24]

Um in *Codesys* die aktuellen Werte über einen OPC-Server an das MES weiterzuleiten, wird die Symbolkonfiguration verwendet. Dort können alle die in der *Process_Cell* bekannten Variablen ausgewählt werden. Muss also ein Wert an das MES übergeben werden, muss dieser bis zur Anlagenebene weitergeleitet werden. So wird zum Beispiel der aktuelle Durchfluss der

Pumpe zuerst vom *Flow_Sensor_CM* an das *Simple_Pump_EM* übergeben, welches ihn weiter an die *Pump_Unit* und dann an die *Process_Cell* leitet. Wird nun ein Modul ausgetauscht, bleibt die MES-Anbindung auch ohne Veränderung bestehen. Es sollen folgende Werte an das MES übermittelt werden:

- Solltemperatur im Tank 101
- Füllstand im Tank 101
- Durchfluss an der Pumpe

4.7 Erzeugen der fertigen Benutzeroberfläche

Der *Application Composers* kann neben dem Steuerungscode auch noch mithilfe des *Visualisierungs Generator* eine Benutzeroberfläche erstellen. Dazu wird zunächst für jedes Modul, insofern es sinnvoll ist, eine eigene Visualisierung erstellt. Hierfür werden alle Werte die für dieses Modul über die Benutzeroberfläche eingestellt werden, sowie alle Werte die dieses Modul anzeigen muss, mit entsprechenden Elementen in der eigenen Visualisierung abgebildet. Zudem werden die einzelnen Anzeigeelemente noch mit den entsprechenden Variablen des Moduls verknüpft. Diese Schritte sind die Voraussetzung um bei der Steuerungscodegenerierung auch gleichzeitig eine individuelle Benutzeroberfläche zu erzeugen. Dazu müssen lediglich die einzelnen Visualisierungen der benutzten Module skaliert und in die richtige Position verschoben werden. Wird nun ein Modul ausgetauscht, wird das HMI automatisch angepasst. In Abbildung 4.12 wird gezeigt wie die fertige Benutzeroberfläche aussieht. Man kann leicht den Unterschied zwischen den einfachen Ventilen und dem Ventil mit Endlagensensoren, welches noch zwei Lampen zur Anzeige des aktuellen Zustand besitzt, erkennen.

4.8 Erstellen des fertigen Modulbaums

Nachdem der Code der einzelnen Module fertig ist, kann mithilfe des *Application Composers* der Steuerungscode für die Laboranlage erzeugt werden. Dazu werden zunächst die benötigten Module in der Modulkonfiguration zusammengefügt. Hierzu wird nach dem Top-Down Prinzip vorgegangen. Das heißt als erstes Modul wird der obersten Ebene, also die *Process_Cell* (oder Anlage), eingefügt. Dann werden die einzelnen Teilanlagen, technischen Einrichtungen und Einzelsteuereinheiten angehängt. Wenn mehrere Module die selbe Schnittstelle verwenden,

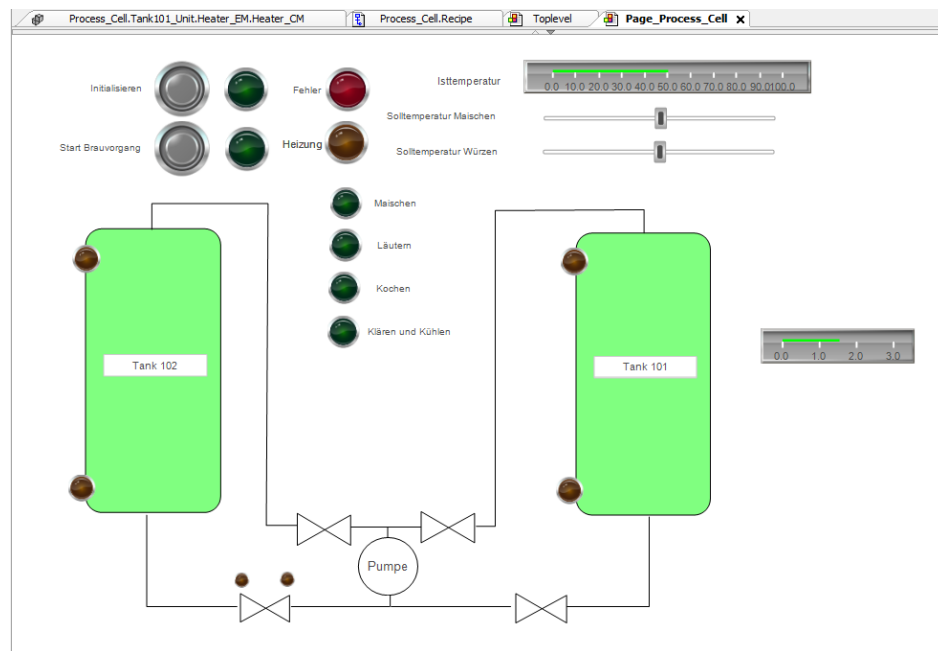


Abbildung 4.12: Benutzeroberfläche der Laboranlage

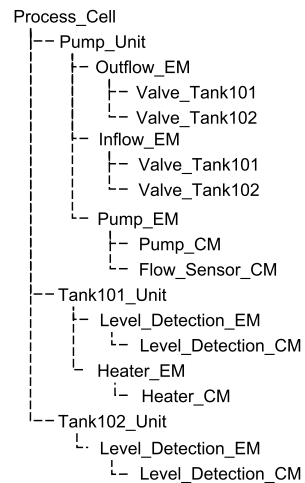


Abbildung 4.13: Modulbaum der fertig konfigurierten Anlage

kann bei der Modulkonfiguration zwischen diesen Modulen ausgewählt werden. Dies ist zum Beispiel der Fall bei der technischen Einrichtung der Pumpe. Hier kann bei der Konfiguration zwischen der einfachen Pumpe und der geregelten Pumpe entschieden werden. Genauso ist es bei den Einzelsteuereinrichtungen der Ventile, bei der auch entweder das einfache Ventil oder das Ventil mit Endlagensensoren ausgewählt werden kann. Ist der individuelle Modulbaum für die jeweilige Anlage erstellt (siehe Abbildung 4.13), kann mithilfe des *Application Composers* der dazugehörige Steuerungscode erstellt werden.

4.9 Hardwareanbindung

Nachdem der Steuerungscode, sowie die Benutzeroberfläche erstellt wurden, müssen die einzelnen Ein- und Ausgänge verknüpft werden. Hierzu bietet sich eine Verknüpfung mit der *Mapping* Funktion von *Codesys* an. Dazu müssen die Hardwareanbindung der Einzelsteuereinheiten lediglich durch einfaches Auswählen mit den entsprechenden Aktoren und Sensoren verbunden werden. Die Verbindung zwischen den Ein- und Ausgängen des Steuerungscode und den Steckplätzen wird graphisch mithilfe einer entsprechenden Linie dargestellt. Zudem werden Eingänge und Ausgänge farblich voneinander getrennt um eine falsche Zuweisung zu vermeiden.

5 Evaluierung des modularen Aufbaus

In diesem Kapitel wird der modulare Steuerungscode bewertet. Zu Beginn der Bachelorarbeit wurde unter anderem die Anforderung der Austauschbarkeit und der Wiederverwendung von Modulen gestellt. Ob diese beiden Anforderungen erfüllt werden wird nun gezeigt.

5.1 Austausch eines Moduls

Eine Veränderung der Anlage soll vereinfacht werden, indem nur das ausgetauschte Modul ersetzt werden muss, wobei der Rest des Codes gänzlich unberührt bleibt. Um zu zeigen, dass das Ziel der Austauschbarkeit erfüllt worden ist, soll das Pumpenmodul der Laboranlage ausgetauscht werden, ohne die restlichen verwendeten Module anpassen zu müssen. Wie schon in den vorherigen Kapiteln beschrieben stehen zwei verschiedene Pumpmodule zu Verfügung. bei der ersten Variante handelt es sich um eine einfache Pumpe. Diese Pumpe verfügt lediglich über die beiden Zustände *Eingeschaltet* und *Ausgeschaltet*. Die zweite Variante bildet eine geregelte Pumpe, welche die Flüssigkeit mit einem vorgegebenen Durchfluss fördern kann.

Damit ein Modul ausgetauscht werden kann müssen zwei Bedingungen erfüllt sein:

- Beide Module müssen über eine identische Softwareschnittstelle zu dem übergeordneten Modul verfügen. Dabei muss das betrachtete Modul nicht zwingend jedes Element der Schnittstelle benutzen. Jedoch muss die Schnittstelle alle vom Modul benötigten Elemente zur Verfügung stellen.
- Neben der gemeinsamen Schnittstelle im Programmcode wird auch eine gemeinsame hardwareseitige Schnittstelle des auszutauschenden Moduls benötigt. Dies bedeutet, dass beide Module die gleichen Verbindungen zu den benachbarten Modulen aufweisen müssen. Dies lässt sich ganz einfach an einem Beispiel verdeutlichen. Ein 3-Wege-Ventil, mit 3 Anschlüssen, kann nicht durch ein einfaches Ventil, mit lediglich zwei Anschlüssen,

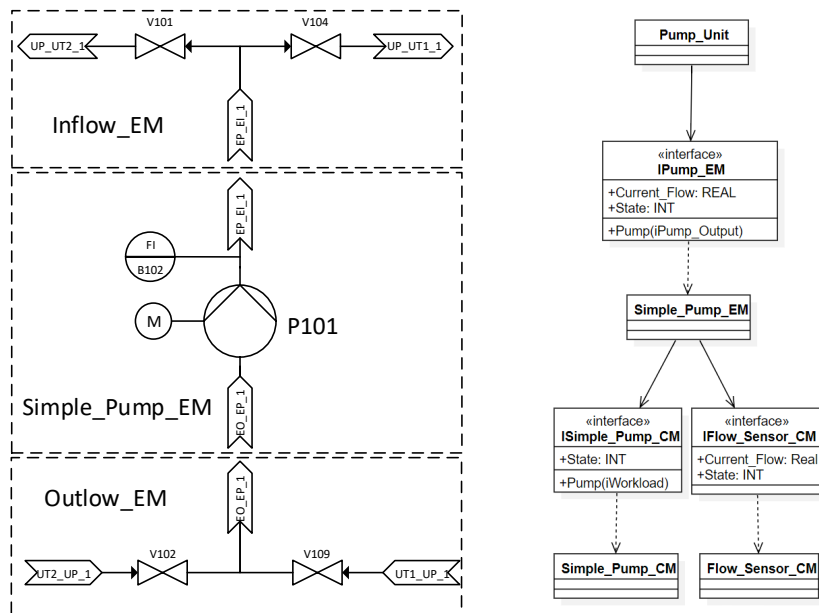


Abbildung 5.1: R&I-Fließbild und Schnittstelle des Simple_Pump_EM

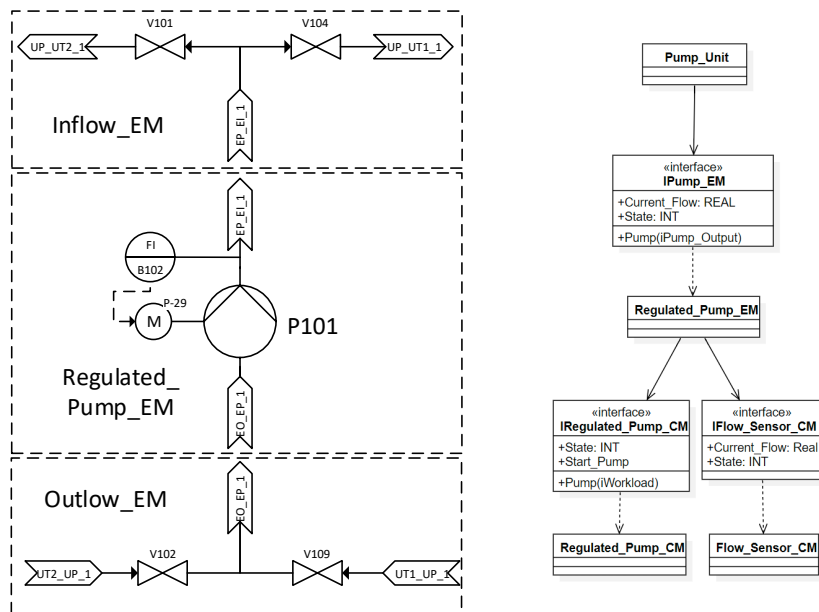


Abbildung 5.2: R&I-Fließbild und Schnittstelle des Regulated_Pump_EM

ersetzt werden, weil sonst eine Verbindung ungenutzt bleiben würde. Das 3-Wege-Ventil kann also nur durch ein Modul ersetzt werden, welches auch über mindestens 3 Anschlüsse verfügt.

Bevor die beiden technischen Einrichtungen *Simple_Pump_EM* und *Regulated_Pump_EM* ausgetauscht werden, wird überprüft ob diese die Voraussetzungen erfüllen. Wie in den Abbildungen 5.1 und 5.2 gezeigt verfügen die beiden technischen Einrichtungen über die gleiche Schnittstelle *IPump_EM*. Damit wäre die erste Bedingung der identischen Software-schnittstelle erfüllt. Beide Module verfügen über jeweils eine physische Verbindung zu den beiden benachbarten technischen Einrichtungen *Inflow_EM* und *Outflow_EM*. In Abbildung 5.1 und 5.2 wird gezeigt, dass die hardwareseitige Schnittstelle der Pumpenmodule zu den Nachbarmodulen, über die beiden Verbindung mit der Bezeichnung *EP_EL1* und *EO_EP1*, identisch ist. Damit erfüllen die beiden technischen Einrichtungen *Simple_Pump_EM* und *Regulated_Pump_EM* auch die zweite Bedingung.

Nachdem die Voraussetzung zum Austausch von *Simple_Pump_EM* und *Regulated_Pump_EM* erfüllt ist, kann im Kapitel 5.3 überprüft werden ob es tatsächlich möglich ist die beiden Module in Codesys gegeneinander auszutauschen.

5.2 Wiederverwendung eines Moduls

Neben dem Vorteil der Austauschbarkeit wurde auch gefordert, dass die einzelnen Module wiederverwendbar sein sollen. Dies kann unter anderem bewiesen werden indem ein Modul mindestens zwei mal in diesem Projekt verwendet wird.

Bevor ein Modul an anderer Stelle wiederverwendet werden kann müssen jedoch zwei Bedingungen erfüllt sein:

- Das Modul muss an seinem neuen Verwendungsort die gleiche Aufgabe übernehmen wie bei der ursprünglichen Verwendung.
- Damit ein Modul ohne Änderung wiederverwendet kann muss am Verwendungsort die gleiche Hardware verfügbar sein, wie am ursprünglichen Verwendungsort.

Am Beispiel der Laboranlage kann gezeigt werden, dass gleich zwei Module wiederverwendet werden können.

Beim ersten Modul handelt es sich um die technische Einrichtung *Level_Detection_EM*. Dieses

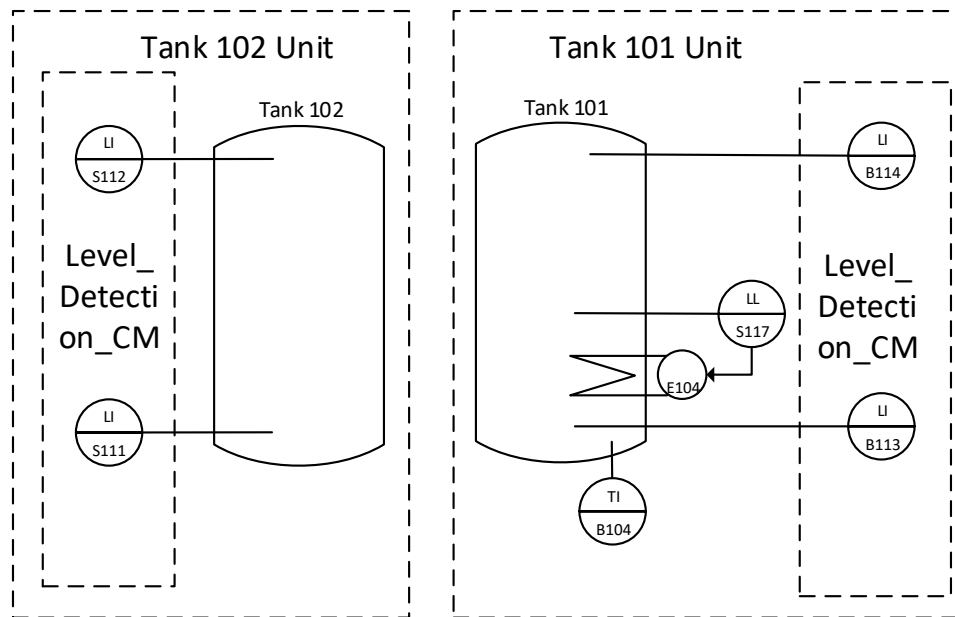


Abbildung 5.3: Wiederverwendung des Level_Detection_CM

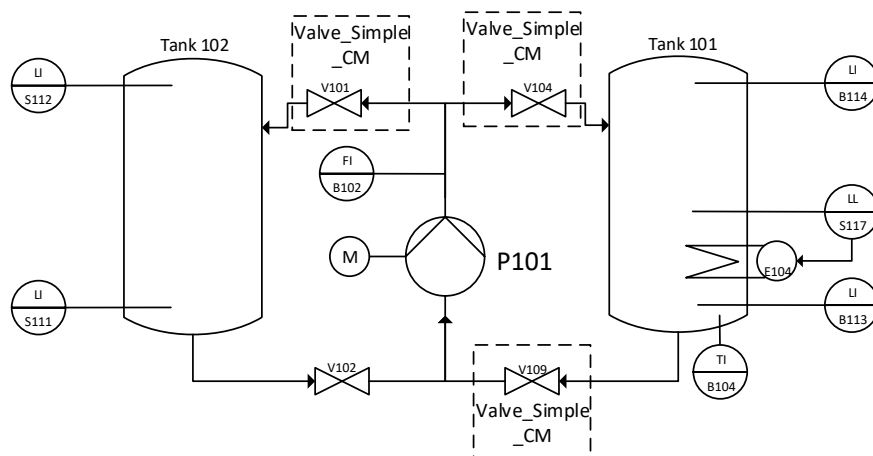


Abbildung 5.4: Wiederverwendung des Simple_Valve_CM

Modul ermittelt mithilfe von zwei Füllstandssensoren den aktuellen Füllstand in einem Behälter. Die technische Einrichtung wird dazu in der *Tank101_Unit*, wie auch in der *Tank102_Unit* eingesetzt. Somit wäre die erste Bedingung, nach der das Modul die gleiche Aufgabe übernehmen muss, erfüllt.

Sowohl der Tank 101 als auch der Tank 102 verfügen über jeweils zwei Sensoren, mit denen der aktuelle Füllstand erkannt werden kann (siehe Abbildung 5.3). Obwohl im Tank 102 zwei Schwimmer und in Tank 101 zwei Näherungssensoren verbaut sind, liefern alle ein binäres Signal an die SPS. Dementsprechend ist es egal wie genau der Sensor den Füllstand erkennt, solange das eingehende Signal an der Steuerung gleich ist. Somit wäre auch die zweite Bedingung und damit einhergehend alle Voraussetzungen für eine Wiederverwendung der technische Einrichtung *Level_Detection_EM* erfüllt.

Neben dem Modul zur Füllstanderkennung kann auch die Einzelsteuereinheit *Valve_Simple_CM* wiederverwendet werden. Dieses Modul soll die einfachen Ventile, von denen drei Stück in der Laboranlage verwendet werden (Abbildung 5.4), je nach Befehl Öffnen oder Schließen. Die Voraussetzungen für eine mehrmalige Verwendung der Einzelsteuereinheit *Valve_Simple_CM* werden erfüllt, da diese immer die gleiche Aufgabe erfüllen und die verwendeten Ventile identisch sind.

5.3 Test der Laboranlage

Nachdem überprüft wurde ob es theoretisch möglich ist Module auszutauschen und wiederzuverwenden, muss dies am realen Steuerungscode noch verifiziert werden.

Um zu überprüfen ob es möglich ist die technische Einrichtung *Simple_Pump_EM* auszutauschen muss zuerst der dazugehörige Steuerungscode erzeugt werden. Dazu wird die Laboranlage im Modulbaum aufgebaut, wobei die technische Einrichtung *Simple_Pump_EM* ausgewählt wird. Dies ist nötig um mithilfe des *Application Composers* den dazugehörigen Steuerungscode und das HMI zu erstellen. Nach diesem Schritt kann das HMI noch durch skalieren und verschieben der einzelnen Anzeigeelemente angepasst werden. Bevor die Softwarelösung getestet werden kann muss noch die Verbindung des MES über die Symbolkonfiguration, sowie die Anbindung an die Hardware über die Mapping-Funktion hergestellt werden. Sind all diese Schritte abgeschlossen ist der Steuerungscode fertig und damit einsetzbar. Es kann nun getestet werden ob alle Funktionen den Vorgaben entsprechen. Also ob die Laboranlage dem Prozess der Würzeherstellung folgt und das HMI alle relevanten Daten anzeigt.

Wenn alle Funktionen den Anforderungen entsprechen, kann überprüft werden ob die technischen Einrichtungen *Simple_Pump_EM* durch die *Regulated_Pump_EM* getauscht werden kann. Dazu soll lediglich die *Simple_Pump_EM* aus dem Modulbaum entfernt werden und durch die *Regulated_Pump_EM* ersetzt werden. Anschließend müssen der Steuerungscode und

das HMI neu erzeugt werden. Da die neuen Einzelsteuereinheiten der *Simple_Pump_EM* eine andere Anbindung an die Hardware benötigen, müssen diese noch über die Mapping-Funktion neu verknüpft werden. Der HMI wurde automatisch ein Schieberegler, für die Durchflussvorgabe, hinzugefügt. Dieser muss dann noch skaliert und in die richtige Position verschoben werden. Alle anderen Anzeigeelemente haben ihre Position und Größe nicht verändert und müssen somit nicht nochmal angepasst werden. Auch hat sich auf Seite des MES nichts verändert, da die neue technische Einrichtung, genau wie ihr Vorgänger, den aktuellen Durchfluss an der Pumpe an die *Pump_Unit* übergibt.

Anhand des nun fertigen Steuerungscode kann gezeigt werden, dass es möglich ist ein Modul auszutauschen, weil das Pumpenmodul ausgewechselt wurde ohne den restlichen Codes zu verändern. Zudem wurde mit dem neuen Modul die Möglichkeit hinzugefügt den Durchfluss an der Pumpe zu regeln. Außerdem wurde das HMI entsprechend den neuen Gegebenheiten angepasst und die Anbindung an das MES wurde nicht verändert.

Neben der Anforderung der Austauschbarkeit muss auch noch bewiesen werden, dass es möglich ist ein Modul mehrmals zu verwenden.

Bei der Moduldeklaration der *Tank101_Unit* und der *Tank102_Unit* wurde schon festgelegt, dass die technische Einrichtung *Level_Detection_EM* angehängt werden kann. Nach dem gleichen Schema wurde bei der Moduldeklaration des *Input_EM* und des *Output_EM* vorgegangen, welche die Einzelsteuereinheit *Valve_Simple_CM* verwenden. Bei der Modulbaumerstellung der Laboranlage kann dementsprechend die technische Einrichtung *Level_Detection_EM* doppelt und die Einzelsteuereinheit *Valve_Simple_CM* dreifach ausgewählt werden. Nach der Erzeugung durch den *Application Composers*, steht der fertige Steuerungscode und die dazugehörige HMI zur Verfügung. Anhand der HMI lässt sich leicht erkennen, dass die Anzeigelämpchen des *Level_Detection_EM* doppelt und die Ventilsymbole des *Valve_Simple_CM* dreifach eingefügt wurden. Es zeigt sich, dass die beiden *Level_Detection_EM* und die drei *Valve_Simple_CM*, nach der Hardwareanbindung, im Betrieb völlig unabhängig voneinander betreiben lassen. Auch wurde die Zuordnung der Anzeigeelemente richtig vorgenommen, da die Lampen des *Level_Detection_EM* immer den aktuell richtigen Zustand ihren dazugehörigen Sensors anzeigen und die Ventile des *Valve_Simple_CM* immer dann leuchten wenn das entsprechende Ventil geöffnet ist.

Es kann also gezeigt werden, dass sich einzelne Module ohne großen Aufwand wiederverwendet lassen.

6 Übertragung auf eine industrielle Anwendung

Im vorherigen Kapitel konnte aufgezeigt werden, dass es möglich ist einzelne Module wiederzuverwenden und unkompliziert auszutauschen. Jedoch wurden die einzelnen Module speziell für die Verwendung an der Laboranlage geschrieben. Um das vorgestellte Konzept auch auf industrielle Anlagen übertragen zu können müssen einige Punkte beachtet werden.

Bei der Wiederverwendung eines Moduls kann es passieren, dass dieses nicht genau den Anforderungen am neuen Einsatzort entspricht. Dabei ist es unerheblich ob das Modul aus einem vorherigen Projekt übernommen, oder ob ein Modul in einem Projekt mehrmals verwendet werden soll. Auch wenn nicht das ganze Modul wiederverwendet werden kann, kann es jedoch sein, dass Teile davon erneut verwendet werden können. Aufgrund der Unterteilung der einzelnen Module in Teilmodule, ist es möglich aus diesen ein neues Modul zusammenzustellen. Dazu muss lediglich der Teil des Moduls ausgetauscht werden, welcher nicht den neuen Anforderungen gerecht wird. Wenn zum Beispiel eine technische Einrichtung wiederverwendet werden soll, aber eine dazugehörige Einzelsteuereinheit nicht den neuen Gegebenheiten entspricht, muss nicht die gesamte technische Einrichtung neu programmiert werden. Es genügt wenn die Einzelsteuereinheit neu erstellt wird und in die vorhandene technische Einrichtung eingefügt wird.

Neben dem Problem, dass ein Modul nicht genau den neuen Anforderungen entspricht kann es sein, dass nicht alle Schnittstellenelemente eines vorhandenen Moduls genutzt werden können. Wird zum Beispiel ein Modul in einem neuen Projekt verwendet kann durchaus sein, dass nicht alle, vom Modul zu Verfügung gestellten, Funktionen verwendet werden. Dies wäre der Fall wenn ein Modul hardwareseitig fünf Anschlussmöglichkeiten zu Verfügung stellt, aber nur vier davon genutzt werden. Dann soll es trotzdem möglich sein dieses Modul zu verwenden, unter der Bedingung, dass dieser ungenutzter Anschluss nicht verwendet wird und somit dauerhaft verschlossen bleibt.

Auch kann es sein dass ein Modul softwareseitig einige Funktionen über Methoden zu Verfügung stellt, welche nicht benötigt werden. In dem Fall werden die nicht benötigten Funktionen eben nicht durch das übergeordnete Modul aufgerufen.

Es kann also festgehalten werden, dass nicht jede vom Modul zu Verfügung gestellte Funktion auch zwingend am neuen Einsatzort verwendet werden muss. Jedoch muss darauf geachtet werden, dass das wiederverwendete Modul mindestens alle geforderten Funktionen mit sich bringt.

7 Zusammenfassung

In diesem Kapitel wird noch einmal zusammenfassen festgehalten ob das Ziel der Bachelorarbeit erreicht wurde. Dazu wird überprüft ob die gestellten Anforderungen erfüllt wurden.

- Anforderung 1: Einteilung der Laboranlage in Modulklassen nach ISA88

Zur Einteilung der Anlage wurde zunächst der Prozess der Würzeherstellung in seine einzelnen Prozessschritte gegliedert. Mit den gewonnenen Informationen wurde die Anlage in drei Teilanlagen zerlegt. Anschließend konnten die einzelnen Teilanlagen in technische Einrichtungen aufgeteilt werden, welche wiederum in Einzelsteuereinheiten unterteilt wurden. Somit wurde die Anlage, gemäß der Anforderung, in die drei Ebenen, Teilanlagen, technischen Einrichtungen und Einzelsteuereinheiten, nach ISA88 eingeteilt.

- Anforderung 2: Austauschbarkeit und Wiederverwendung von Modulen

Am Beispiel der Pumpe kann gezeigt werden, dass es möglich ist die technische Einrichtung *Simple_Pump_EM* durch die technische Einrichtung *Regulated_Pump_EM* auszutauschen, ohne die restlichen verwendeten Module anpassen zu müssen. Dazu muss lediglich das Modul in der Modulkonfiguration ersetzt werden und die Anbindung der Ein- und Ausgänge erneuert werden. Es ist somit möglich, mithilfe des *CODESYS Application Composer*, ein Modul zu tauschen ohne den Programmcode zu verändern.

Zudem kann an zwei Beispielen gezeigt werden, dass es möglich ist verschiedene Module mehrfach zu verwenden. So wird die technische Einrichtung *Level_Detection_EM* zur Füllstanderkennung sowohl in der *Tank 101 Unit* als auch in der *Tank 102 Unit* verwendet. Des weiteren wird die Einzelsteuereinheit *Valve_simple_CM* des einfachen Ventils gleich drei mal wiederverwendet.

- Anforderung 3: Einheitliche Schnittstellen zwischen Modulklassen

Für jede Anlagenebene wurde eine Schnittstelle erstellt, von der die jeweiligen Modulschnittstellen erben können. Dadurch wird sofort ersichtlich zu welcher Ebene das

betrachtete Modul gehört. Zusätzlich erhalten die Modulschnittstellen die benötigten Methodenaufrufe und Parameterübergaben um mit dem übergeordneten Modul kommunizieren zu können. Können an einer Stelle mehrere Module eingefügt werden, besitzen diese die selbe Schnittstelle. So besitzen die beiden technischen Einrichtungen *Simple_Pump_CM* und *Regulated_Pump_CM* die gleiche Schnittstelle *IPump_EM*. Diese gemeinsame Schnittstelle erlaubt bei der Modulkonfiguration die Auswahl zwischen diesen beiden Modulen.

- Anforderung 4: Automatische Generierung des Steuerungscode

Nachdem alle Module implementiert wurden, kann der Steuerungscode mithilfe des *CODESYS Application Composer* automatisch erstellt werden. Dazu müssen zunächst die einzelnen Module im Modulbaum zu einer Anlage zusammengefügt werden. Anschließend kann ganz ohne Programmierung der Steuerungscode erzeugt werden.

- Anforderung 5: MES- und HMI Anbindung

Es kann gezeigt werden, dass die Benutzeroberfläche automatisch aus der Modulkonfiguration erzeugt wird. Dies ist möglich da einige Module über eine eigene Visualisierung verfügen, die mittels des *Visualisierungs-Generator* zu einer Benutzeroberfläche zusammengesetzt werden. Zudem wird die Benutzeroberfläche beim Modulaustausch mit angepasst, was beim Tausch der Pumpe gezeigt werden konnte.

Auch wird die MES Anbindung, nachdem sie einmal erstellt wurde, immer mit angepasst. Zum Beweis kann auch hier wieder der Austausch der Pumpe hergenommen werden. Wird die technische Einrichtung ersetzt, bleibt die Übertragung des aktuellen Durchflusses der Pumpe an das MES erhalten.

1 Literatur

- [1] K.-U. Heyse, Hrsg., *Handbuch der Brauerei-Praxis*, 3., verb. und erw. Aufl. Nürnberg: Carl Getränke-Fachverl., 1995, ISBN: 3418007368.
- [2] B. Vogel-Heuser und A. Wannagat, *Modulares Engineering und Wiederverwendung mit CoDeSys V3: Für Automatisierungslösungen mit objektorientiertem Ansatz*. München: Oldenbourg-Industrieverl., 2009, ISBN: 978-3-8356-3105-2.
- [3] *CODESYS Application Composer: Konfektionieren von IEC 61131-3 Applikationssoftware für Automatisierungssysteme*, 29.11.2016. Adresse: <https://de.codesys.com/produkte/codesys-engineering/application-composer.html>.
- [4] B. Vogel-Heuser, *Engineering von der Anforderung bis zum Betrieb*, Ser. Embedded Systems. 1, Tagungen und Berichte. 2013, Bd. 3, ISBN: 9783862193783.
- [5] C. Bramsiepe und G. Schembecker, “Die 50 %-Idee: Modularisierung im Planungsprozess”, *Chemie Ingenieur Technik*, Jg. 84, Nr. 5, S. 581–587, 2012, ISSN: 0009286X. DOI: 10.1002/cite.201100250.
- [6] M. Obermeier, *Untersuchung von Programmieransätzen im Maschinen- und Anlagenbau bezüglich ihrer Gebrauchstauglichkeit sowie ihrer Einflüsse auf die Softwaremodularisierung: Vollst. zugl.: München, Techn. Univ., Diss., 2015*, 1. Aufl., Ser. Technische Universität München 2, Forschung. Göttingen: Sierke, 2015, Bd. 13, ISBN: 978-3-86844-735-4.
- [7] O. Eitelwein und J. Weber, *Unternehmenserfolg durch Modularisierung von Produkten, Prozessen und Supply Chains: WHU Benchmarking-Studie Modularisierung*. Norderstedt: Books on Demand, 2008, ISBN: 3842322194.
- [8] L. Urbas, F. Doherr, A. Krause und M. Obst, “Modularisierung und Prozessführung”, *Chemie Ingenieur Technik*, Jg. 84, Nr. 5, S. 615–623, 2012, ISSN: 0009286X. DOI: 10.1002/cite.201200034.
- [9] O. Eitelwein, S. Malz und J. Weber, “Erfolg durch Modularisierung”, *Controlling & Management*, Jg. 56, Nr. 2, S. 79–84, 2012. DOI: 10.1365/s12176-012-0389-3.

- [10] D. James, “Batch process automation - applying the principles of batch automation and selecting a suitable control system and supplier is only part of the story. managing a batch automation project requires considerations over and above those needed for the automation of a continuous process plant”, *Manufacturing Engineer*, Jg. 85, Nr. 6, S. 36–41, 2006, ISSN: 0956-9944.
- [11] DIN Deutsches Institut für Normung e.V., *Chargenorientierte Fahrweise - Teil 1: Modelle und Terminologie (IEC 61512-1:1997)*, 1999-05-01.
- [12] J. Lunze, *Automatisierungstechnik: Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*, 2., überarb. Aufl. München: Oldenbourg, 2008, ISBN: 978-3-486-58061-7. Adresse: <http://dx.doi.org/10.1524/9783486595116>.
- [13] L. Urbas, *Process control systems engineering*. München: Oldenbourg-Industrieverl., 2012, ISBN: 3-8356-3198-5.
- [14] G. Yang und H.-O. Günther, *Produktionsplanung in komplexen Wertschöpfungsnetzwerken: Ein integrierter hierarchischer Ansatz in der chemischen Industrie: Zugl.: Berlin, Techn. Univ., Diss., 2004*, 1. Aufl., Ser. Gabler Edition Wissenschaft Produktion und Logistik. Wiesbaden: Dt. Univ.-Verl., 2005, ISBN: 9783824482917.
- [15] K. F. Früh und W. Ahrens, Hrsg., *Handbuch der Prozessautomatisierung: Prozessleittechnik für verfahrenstechnische Anlagen*, 3., völlig überarb. und erw. Aufl. München: Oldenbourg, 2004, ISBN: 3-486-27048-6.
- [16] R. Langmann, Hrsg., *Taschenbuch der Automatisierung*, 3., neu bearbeitete Auflage. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2017, ISBN: 978-3-446-44664-9. Adresse: <http://www.hanser-fachbuch.de/9783446446649>.
- [17] Dr.-Ing. Martin Krahe, “Vom Equipment Module zur Produktionsanlage: Effiziente Abwicklung von Grossprojekten im Biotech-Sektor”, *CHEManager*, Jg. 21, Nr. 18/2012, S. 9, 2012.
- [18] D. James, “Batch of the day [batch processing]”, *Computing and Control Engineering*, Jg. 17, Nr. 4, S. 30–35, 2006, ISSN: 0956-3385. DOI: 10.1049/cce:20060406.
- [19] K. F. Früh, D. Schaudel, U. Maier und R. Bleich, Hrsg., *Handbuch der Prozessautomatisierung: Prozessleittechnik für verfahrenstechnische Anlagen*, 5. komplett überarb. Aufl. München: DIV Dt. Industrieverl., 2015, ISBN: 9783835671195.
- [20] K. Akesson und M. Fabian, “Implementing supervisory control for chemical batch processes”, in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*, IEEE, 1999, S. 1272–1277, ISBN: 0-7803-5446-X. DOI: 10.1109/CCA.1999.801155.
- [21] S. Ibrahimkadić und S. Kreso, “Development of a batch control system”, in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, S. 947–951.

- [22] M. Roth, S. Scholz, K. Gövert, D. Kasperek, C. Lozano, H. Mund und U. Lindemann, “Standardisierungskonzept für Kleinserien im Maschinen- und Anlagenbau”, in *Tag des Systems Engineering*, M. Maurer, S.-O. Schulze und J. Abulawi, Hrsg., München: Hanser, 2015, S. 361–370, ISBN: 978-3-446-44357-0.
- [23] “Alkoholische Getränke”, in *Lebensmittelverfahrenstechnik*, H. P. Schuchmann und H. Schuchmann, Hrsg., Weinheim: Wiley-VCH, 2005, S. 95–103, ISBN: 9783527623549. DOI: 10.1002/9783527623549.ch10. Adresse: <http://onlinelibrary.wiley.com/doi/10.1002/9783527623549.ch10/summary>.
- [24] O. Sauer und M. Ebel, “Plug-and-work von Produktionsanlagen und übergeordneter Software”, in *GI Jahrestagung (2)*, 2007, S. 331–338.

