

Masterarbeit

MÖGLICHKEITEN VON AUTOMATISIERTEM HARD- WARE-ENGINEERING UNTER ZUHILFENAHME VON MODELLORIENTIERTER ARCHITEKTUR

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von

Ing. Dominik Gruber, BSc

1810322018

betreut und begutachtet von
Dipl.-Ing. Johannes Fritz, BSc

Graz, im Dezember 2019

A handwritten signature in blue ink, appearing to read "Dominik Gruber".

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

A handwritten signature in blue ink, reading "Dominik Gulo", is written above a horizontal dotted line.

Unterschrift

DANKSAGUNG

Ich möchte mich an dieser Stelle bei allen, die mich im Zuge dieser Arbeit moralisch, mit technischem Fachwissen sowie zeitlichem Aufwand unterstützt haben, bedanken. Im speziellen bei meiner Lebensgefährtin Alexandra für das Korrekturlesen, Christoph A. für den technischen Rat und meinem Betreuer seitens der FH CAMPUS 02, Herrn Dipl.-Ing. Johannes Fritz, BSc, für die konstruktiven Ratschläge und Hinweise. Darüber hinaus möchte ich mich bei meinen Kollegen der Firma *metior GmbH*, Christoph L. und Viktor, für die Unterstützung beim durchgeführten Workshop und dem Verifizieren der Workflow-Ergebnisse bedanken. Ein besonderer Dank gilt Stefan für die kompetente Expertise in punkto Programmierung und dem technischen Korrekturlesen der Arbeit. Des Weiteren ein Dank an die Firma *CAE Expert Group GmbH* für die im Zuge dieser Arbeit kostenlose Nutzung des Schaltplangenerators *G8*.

KURZFASSUNG

Steigende Konkurrenz für Dienstleistungsunternehmen im Bereich Elektrotechnik-Planung erfordert kürzere Projekt-Durchlaufzeiten sowie einen hohen Qualitätsstandard der Dokumentation. Darüber hinaus steigt die Anzahl an Engineering-Tools, was dazu führt, dass hoher Ressourcenaufwand sowie plattform-spezifische Fehlerbehebung notwendig ist. Zusätzlich ist die Nachverfolgung von Basisdaten-Änderungen ein wichtiger Umstand, um schnelle Reaktion im Planungsprozess zu garantieren. Das Ziel dieser Masterarbeit ist, Pläne für verschiedene Engineering-Umgebungen unter Zuhilfenahme einer abstrakten Herangehensweise zu generieren, um plattform-spezifische Arbeitsschritte zu minimieren. Die Durchführung basiert auf dem modellgetriebenen Entwicklungsansatz. Dazu wird der aktuelle Planungsprozess untersucht und die darin enthaltenen Dokumente identifiziert. Auf Basis derer wird ein neuer, modellgetriebener Planungsprozess entwickelt, mit einem plattform-unabhängigen Modell als Hauptbestandteil. Die Prozessqualität wird anhand des Erfüllungsgrades der Anforderungen festgelegt. Mit dem ersten Testprojekt, das im Zuge dieser Arbeit erstellt wird, lässt sich das Potenzial zur Reduzierung von Entwicklungsaufwand sowie Projekt-Durchlaufzeiten bereits erkennen. Um diesen modellgetriebenen Planungsprozess effizient anwenden zu können, müssen jedoch Großprojekte damit umgesetzt werden. In der aktuellen Situation ist dieser Prozess nicht anwendbar, da keine Projekte dieser Größenordnung im Unternehmen geplant sind. Jedoch kann der Prozess als Leitfaden für die Planung elektrotechnischer Anlagen dienen.

ABSTRACT

Increasing competition in the electrotechnical service industry requires reduced project lead times and high quality project documentation. Moreover, the number of engineering tools increases, which leads to tool based troubleshooting as well as extensive input of personal resources. Additionally, requirement changes during the planning phase need to be tracked in order to react as quickly as possible. The aim of this thesis is to generate plans for various engineering tools by using a higher level planning approach to reduce platform specific efforts, which can be fulfilled by Model Based Development. Therefore the current planning processes are analysed and the necessary input and output documents are defined. These documents provide the basis for the development of a new Model Driven Planning process, which mainly focuses on a cross-platform model. The process quality is evaluated and it is observed if the level of compliance of requirements is achieved. The first test on the demo-project reflects impressive potential to reduce development effort and project lead times. To apply this Model Driven Planning process in an efficient way, it must be deployed in large scale projects. In the current situation this process is not applicable because of missing large-scale projects, but could be used as a guideline for planning electrotechnical plants.

INHALTSVERZEICHNIS

1	Einleitung.....	5
2	Planung verfahrenstechnischer Anlagen	6
2.1	Planungsphasen der Anlagenplanung.....	6
2.1.1	Grundlagenermittlung (establishment of fundamentals).....	8
2.1.2	Die Vorplanung (Pre-Planning).....	8
2.1.2.1	Verfahrensauswahl und -entwurf.....	9
2.1.2.2	Prozessleittechnik-Konzept	9
2.1.2.3	Beschaffungskonzept	9
2.1.2.4	Konzeptbewertung.....	10
2.1.3	Die Entwurfsplanung (Basic-Engineering).....	10
2.1.3.1	Verfahrensplanung	10
2.1.3.2	3-D-Anlagenentwurfsplanung	12
2.1.3.3	Entwurfsplanung PLT	12
2.1.4	Die Ausführungsplanung (Detail-Engineering)	13
2.1.4.1	Das mechanische Detail-Engineering	13
2.1.4.2	Das Elektrotechnik-, EMSR- und Prozessleittechnik-Detail-Engineering	14
2.1.5	Die Bau-, Montage- und Inbetriebnahmephase.....	17
2.2	Dokumente im Planungsprozess	18
2.2.1	Das Lastenheft.....	19
2.2.2	Das R&I-Fließschema.....	19
2.2.3	Die EMSR-Stellenliste	19
2.2.4	Die elektrische Verbraucherliste	20
2.2.5	Die E-Dokumentation.....	21
2.2.6	Funktionsbeschreibung und Funktionsplan	22
2.2.7	Dokumente der Prozessleittechnik	23
2.2.8	Conclusio der Planungsdokumente	24
2.3	Agiles Engineering und Claim Management	24
2.4	Zusammenfassung	25
3	Modelle und das Systems Engineering.....	27
3.1	Die Hauptmerkmale eines Modells	27
3.2	Modelltypen.....	27
3.2.1	Prinzipien der Modellierung	28
3.3	Das Metamodell	29
3.3.1	Prinzip der Modellierungssprachen	30
3.3.1.1	Domain Specific Language – <i>DSL</i>	30
3.3.1.2	Unified Modeling Language – <i>UML</i>	31
3.3.1.3	System Modeling Language – <i>SysML</i>	31
3.4	Das Systems Engineering	32
3.4.1	Grundprinzipien des systemischen Denkens	33

3.4.1.1	Das Grundprinzip der Strukturierung.....	33
3.4.1.2	Das Grundprinzip vom Abstrakten zum Konkreten	34
3.4.1.3	Das Grundprinzip der Standardisierung	34
3.4.1.4	Das Grundprinzip der Problemlerlegung	34
3.4.2	Vorgehensweisen im Systems Engineering	34
3.4.2.1	Der SIMILAR-Prozess	34
3.4.2.2	Das V-Modell als Makrozyklus	36
3.4.2.3	Erweiterung zum W-Modell	37
3.4.2.4	Münchener Vorgehensmodell.....	38
3.5	Zusammenfassung Modell und Systems Engineering	40
4	Modelgetriebene Architektur – Model Driven Architecture (MDA)	41
4.1	Viewpoints und Schichten/Modelle der MDA.....	41
4.2	Plattformen im MDA-Kontext	42
4.3	Modellierungssprache der MDA	42
4.4	Transformationen zwischen Modellen	42
4.5	Ziel der MDA.....	43
4.6	Zusammenfassung MDA	44
5	Die praktische Umsetzungsidee.....	45
5.1	Unternehmensvorstellung metior GmbH	45
5.2	Workflow-Beschreibung.....	46
5.3	Auswahl der betrachteten Engineering-Plattformen.....	48
5.4	Voraussetzungen/Anforderungen des Workflows	49
6	Basisunterlagen für den Test-Workflow	51
6.1	R&I-Fließschema	51
6.2	Messkreis- und Stellenliste.....	52
6.3	Lastenheftauszüge	53
7	Modellierung des Meta- und Anlagenmodells.....	55
7.1	Erläuterung Metamodell.....	55
7.1.1	Das Metamodellobjekt	55
7.1.2	Die Beziehungstypen.....	55
7.1.3	Ausschnitt aus dem Metamodell.....	56
7.1.4	Auflistung aller verwendeten Metamodellobjekte	57
7.1.5	Erstellvorgang des Metamodells.....	57
7.2	Anlagenmodell basierend auf den Basisunterlagen	58
7.2.1	Funktionaler Aufbau des Anlagenmodells	58
7.2.2	Druckmessstelle H1NDC01CP001	59
7.2.3	Stellstelle H1GHE10AA103	60
7.2.4	Steuerungskomponenten im Anlagenmodell.....	61
7.2.5	Gesamtes Anlagenmodell.....	62
7.2.6	Simultanes Erarbeiten von Typicals und Anlagenmodell	62
8	Erläuterung der Engineering-Plattformen und notwendige Vorbereitungen vor der Generierung.....	63
8.1	<i>Engineering Base</i> der Firma Aucotec	63

8.1.1	Aufbau und Unterteilung der Datenbank bzw. deren Objekte	63
8.1.2	Vorbereitungen in <i>EB</i>	65
8.2	<i>EPLAN Electric P8</i> der Firma <i>EPLAN</i>	66
8.2.1	Strukturierung und Besonderheiten	66
8.2.2	Vorbereitungen in <i>EPLAN</i>	67
8.3	Typicals.....	69
8.3.1	Typicals in <i>EB</i>	69
8.3.1.1	Typical-Stellstelle <i>ARM_P_AufZu</i>	70
8.3.1.2	Typical-Messstelle <i>MSR_TI_Pt100_4-20mA_2L</i>	70
8.3.1.3	Typical-Messstelle <i>MSR_PI_4-20mA_2L</i>	71
8.3.2	Typicals in <i>EPLAN Electric P8</i>	71
8.3.2.1	Typical-Stellstelle <i>ARM_P_AufZu</i>	71
8.3.2.2	Typical-Messstelle <i>MSR_TI_Pt100_4-20mA_2L</i>	72
8.3.2.3	Typical-Messstelle <i>MSR_PI_4-20mA_2L</i>	72
8.4	Erkenntnisse aus den Plattformen und Typicals	72
9	Beschreibung der Generatoren.....	73
9.1	Der <i>Typical Manager</i> in <i>EB</i>	73
9.2	Der Schaltplangenerator <i>G8</i> für <i>EPLAN Electric P8</i>	75
9.2.1	Gesamt-Typicals in Einzel-Makros aufteilen	75
9.2.2	Konfiguration der Typicals in den <i>G8-Stammdaten</i>	78
9.2.3	Die Engineering- und Makro-Liste der Projektdaten.....	80
9.3	Erkenntnisse aus den Generatoren	82
10	Auslesen der Modellinformationen mit <i>Visual Basic for Applications</i> in <i>Microsoft Excel</i>	83
10.1	Das zugrundeliegende Datenformat <i>GraphML</i>	83
10.1.1	Kurzbeschreibung XML	83
10.1.2	Aufbau des <i>GraphML</i> -Formats	83
10.2	<i>Visual Basic for Applications, VBA</i>	84
10.3	<i>Microsoft Excel/VBA</i> -Anwendung zur Transformation vom plattformunabhängigen Modell in die Engineering-Umgebungen	85
10.3.1	Sub-Prozedur <i>GenerateForEB()</i>	86
10.3.1.1	Sub-Prozedur: <i>MSRListe</i>	87
10.3.1.2	Sub-Prozedur: <i>SPSImportListe</i>	87
10.3.1.3	Sub-Prozedur: <i>BaueinheitenListe</i>	88
10.3.1.4	Sub-Prozeduren: <i>KabellImportListeMessstelle / KabellImportListeStellstelle</i>	89
10.3.1.5	Sub-Prozeduren: <i>FeldgeraetImportListeMessstelle / FeldgeraetImportListeStellstelle</i>	89
10.3.1.6	Sub-Prozedur: <i>KlemmleisteImportListeMessstelle / KlemmleisteImportListeStellstelle</i>	90
10.3.2	Sub-Prozedur: <i>GenerateForEPLAN()</i>	90
10.3.3	Funktion: <i>getNodeByString</i>	92
10.4	Conclusio aus dem Lese- bzw. Aufbereitungsvorgang	93
11	Generierung, Ergebnisse und Nachbearbeitung.....	94
11.1	Objekt- und Plangenerierung in <i>EB</i>	94
11.1.1	Import der Objekte aus den <i>Microsoft Excel</i> Tabellen.....	94

11.1.2	Pläne generieren mit dem <i>Typical Manager</i>	97
11.1.3	Ergebnisse und Nachbearbeitung in <i>EB</i>	97
11.2	Plangenerierung in <i>EPLAN</i>	99
11.2.1	Planerstellung mit dem Schaltplangenerator <i>G8</i>	99
11.2.2	Ergebnisse und Nachbearbeitung in <i>EPLAN</i>	100
11.3	Resümee des Generierungsprozesses, der Ergebnisse und Nachbearbeitungsschritte.....	101
12	Verifizieren der Ergebnisse.....	102
13	Zusammenfassung und Ausblick.....	103
	Literaturverzeichnis.....	105
	Abbildungsverzeichnis.....	109
	Quelltextverzeichnis.....	112
	Tabellenverzeichnis.....	113
	Abkürzungsverzeichnis.....	114
	Anhang 1: Workshop Engineering-Prozess.....	115
	Anhang 2: Stellenplan.....	118
	Anhang 3: Vollständige Auflistung aller Metamodellobjekte.....	119
	Anhang 4.1: <i>EB</i> -Typical Stellstelle <i>ARM_P_AufZu</i>	122
	Anhang 4.2: <i>EB</i> -Typical Messstelle <i>MSR_TI_Pt100_4-20mA_2L</i>	123
	Anhang 4.3: <i>EB</i> -Typical Messstelle <i>MSR_PI_4-20mA_2L</i>	124
	Anhang 5.1: <i>EPLAN</i> -Typical Stellstelle <i>ARM_P_AufZu</i>	125
	Anhang 5.2: <i>EPLAN</i> -Typical Messstelle <i>MSR_TI_Pt100_4-20mA_2L</i>	126
	Anhang 5.3: <i>EPLAN</i> -Typical Messstelle <i>MSR_PI_4-20mA_2L</i>	127
	Anhang 6: Einzel-Makros des <i>EPLAN</i> -Stellstellentypicals.....	128
	Anhang 7: Auflistung der Makro-Parameter.....	129
	Anhang 8: Elemente importieren und aktualisieren, Mappings.....	131
	Anhang 9.1: Generierter <i>EB</i> -Stellenplan <i>H1 GHE10 AA103</i>	137
	Anhang 9.2: Generierter <i>EB</i> -Stellenplan <i>H1 NDC01 CP001</i>	138
	Anhang 10: Befüllte Engineering-Liste für den Schaltplangenerator <i>G8</i>	139
	Anhang 11.1: Generierter <i>EPLAN</i> -Stellenplan <i>H1 GHE10 AA103</i>	140
	Anhang 11.2: Generierter <i>EPLAN</i> -Stellenplan <i>H1 NDC01 CP001</i>	141

1 EINLEITUNG

Durch internationalen Wettbewerb im Bereich der elektrotechnischen Anlagenplanung ist die Auseinandersetzung mit automatischer Dokumentationserstellung als Ingenieurbüro unumgänglich. Als Dienstleistungsunternehmen im Bereich Automatisierungstechnik wird besonders auf stetige Qualitätssteigerung der Dokumentation sowie Verkürzung der Projektlaufzeit geachtet. Die Firma *metior GmbH* ist stets bemüht, auf dem aktuellen Stand in punkto automatisiertem Engineering-Prozessen zu sein, die darüber hinaus eine durchgängige Planung mehrerer Fachdisziplinen unterstützen. Durch einen breiten Kundenstock aus der Papier- und Zellstoffindustrie, über Energieversorgungsunternehmen bis hin zur Automotive-Industrie, werden die gängigsten Engineering-Umgebungen angeboten. Diese haben Plangeneratoren bereits integriert oder können über Drittsoftware erweitert werden.

Die ursprünglich aus der Softwareentwicklung stammende, modellgetriebene Architektur versteht sich darauf, dass zur Erstellung von Anwendungsprogrammen unabhängige Modelle verwendet werden, aus denen Programmcode generiert wird. Auf diese Weise ist die Entwicklung unabhängig von Endplattformen.

Diese Arbeit nimmt sich der Fragestellung an, inwieweit der modellgetriebene Entwicklungsansatz für den Planungsprozess einer Elektrotechnischen,- Mess-, Steuer- und Regelungstechnische-Hardware-Planung für Prozessanlagen herangezogen werden kann. Zusätzlich wird eine weitestgehend automatisierte Dokumentationsgenerierung angestrebt, um unabhängig von kundenseitigen Engineering-Umgebungen zu sein. Daraus resultiert einerseits, dass das Hauptaugenmerk auf der technischen Abwicklung von Projekten liegen und andererseits eine Zeit- und Kostenersparnis in der Projektabwicklung generiert werden kann.

Dazu befindet sich im ersten Teil dieser Arbeit eine Analyse der Planungsschritte zur Umsetzung einer Prozessanlage mit den darin notwendigen Dokumenten. Anschließend werden Modelle per se, das Systems-Engineering mit dessen Vorgehensweisen sowie der modellgetriebene Entwicklungsansatz untersucht, um die Basis für die angestrebte praktische Umsetzung zu legen. Im zweiten Teil folgt die Beschreibung des erarbeiteten Workflows, der an einem simulierten Versuchsprojekt erprobt wird. Dafür werden Basisunterlagen aufbereitet, mit denen ein Modell gebildet wird und schlussendlich über eine teilautomatische Transformation Pläne in den zwei Engineering-Plattformen *Engineering Base* und *EPLAN Electric P8* generiert werden.

2 PLANUNG VERFAHRENSTECHNISCHER ANLAGEN

In prozesstechnischen Anlagen werden Produkte durch komplexe chemische, physikalische sowie weitere technische Abläufe hergestellt. Die Planung solcher Anlagen vereint mehrere Disziplinen, Berufsgruppen und Unternehmen, die in den unterschiedlichen Phasen einer solchen Umsetzung tätig sind.

2.1 Planungsphasen der Anlagenplanung

Der allumfassende Ablauf der Entstehung einer Prozessanlage kann grob in vier Teile gegliedert werden (siehe Abbildung 1).

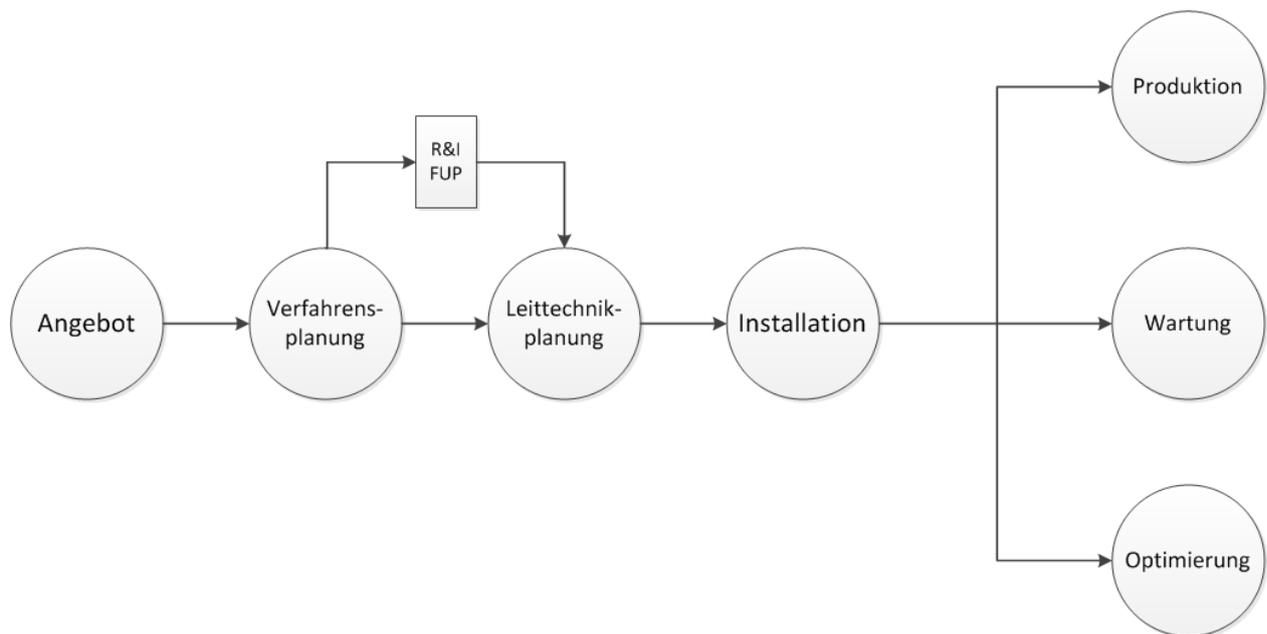


Abbildung 1: Planungsphasen in der Prozessindustrie, Quelle: In Anlehnung an Draht (2009), S.20

Nach dem Angebot wird in der Verfahrensplanung die Anlage technisch spezifiziert. Dabei werden bereits verschiedene Dokumente, Listen und Beschreibungen erstellt. Am aussagekräftigsten sind dabei die Rohrleitungs- und Instrumentierungs-Fließschemata. Darin werden die Abläufe und Prozesse der Anlage, Informationen zur Rohrleitungstechnik sowie Prozessleittechnik (PLT) dargestellt. Als Dokument mit hohem Informationswert ist es für viele Gewerke im Planungsprozess die Grundlage für weitere Planungsschritte. Überdies ist es, mit der Funktionsbeschreibung, die Basis für die leittechnische Umsetzung. Im Anschluss daran kommt es zum Bau und zur Inbetriebnahme der verfahrenstechnischen Anlage, auf Basis der zuvor erstellen Planungsdokumente. ¹

Einen detaillierteren Einblick in die beschriebenen Abläufe einer Anlagenplanung gibt das in Abbildung 2 dargestellte *Phasenmodell der Anlagen-Projektentwicklung*². Darin sind neun Phasen vorgesehen, die vorwiegend sequenziell ablaufen, sich jedoch stets mit einem gewissen Maß überschneiden. Die ersten, vor-

¹ Vgl. Drath/Grimm/Hundt/Keibel/Lips/Lüder/Schleipen/Weidemann (2009), S. 19-22.

² Vgl. Weber (2016)

gelagerten Abschnitte sind im *Konzept- und Entscheidungszeitraum* angesiedelt. Deren Planungsdokumente werden als *vorläufig* bezeichnet, da die durchgeführten Festlegungen und technischen Entscheidungen vor der endgültigen Investitionsentscheidung liegen. Bei der *Grundlagenermittlung* und *Vorplanung* werden grundlegende Aufgabenstellungen sowie die zugehörigen Lösungsansätze definiert. In der *Entwurfsplanung* wird die Basis der technischen Umsetzung erarbeitet. Parallel zu diesen Phasen finden die *Genehmigungsplanung* und *Kostenermittlung* statt. Zu beachten sind dabei unter anderem gesetzliche Bestimmungen wie Umweltauflagen. Zur Ermittlung der Kosten werden aufgrund der getroffenen Entscheidungen Angebote von Lieferanten der einzelnen Disziplinen eingeholt.³

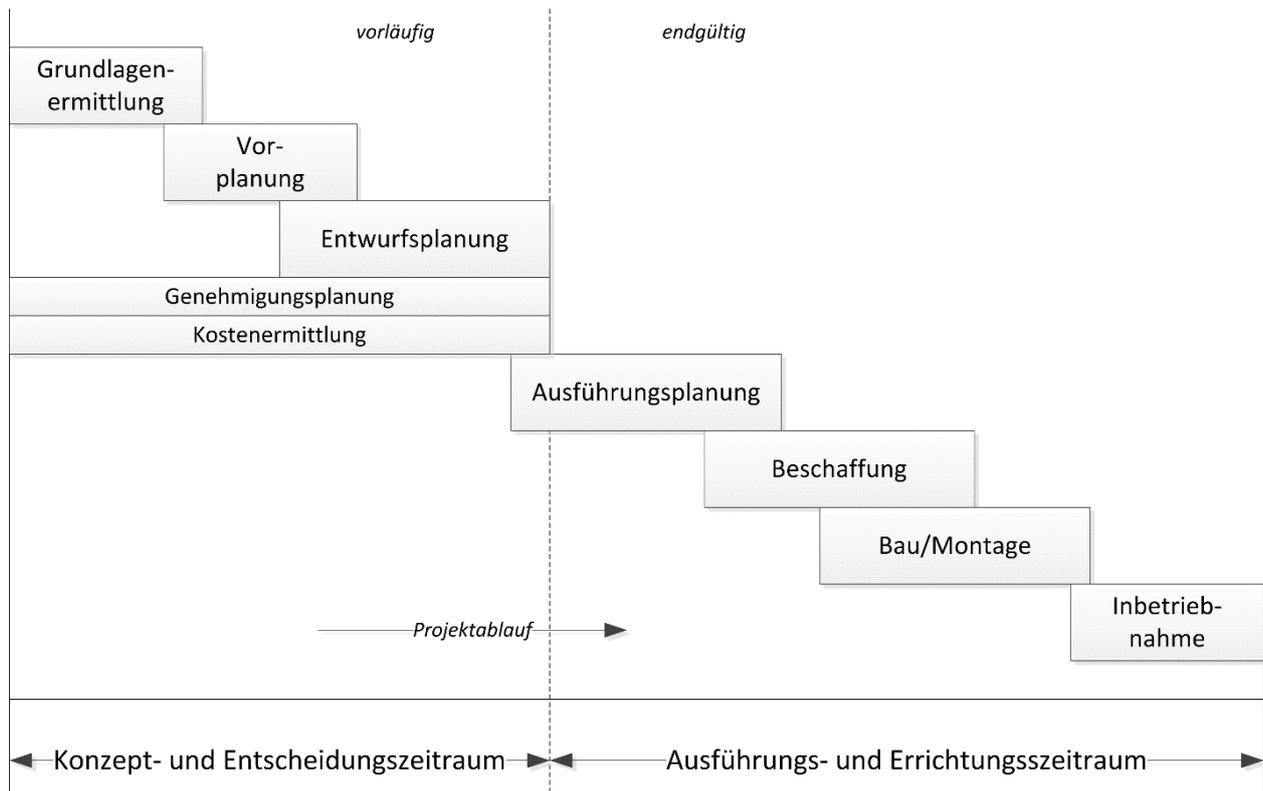


Abbildung 2: Phasenmodell für Realisierung von Anlagen, Quelle: In Anlehnung an Weber (2016), S. 4.

Im zweiten Teil des Ablaufes, dem *Ausführungs- und Errichtungszeitraum*, werden die endgültigen Planungstätigkeiten durchgeführt. Da in diesem Abschnitt bereits eine Freigabe des Budgets und der Investition vorliegt, wird von *endgültigen* Dokumenten oder Listen gesprochen. Wichtige Projektunterlagen, auf die in Punkt 2.2 näher eingegangen wird, werden aus der *Entwurfsplanung* in diese Phase als Basisdokumente für die technischen Umsetzungen übergeben. Die *Ausführungsplanung* wirkt in der finalen Ausarbeitung dieser Dokumente und Festlegungen bereits mit, um technische Expertise einfließen zu lassen. Ein ausgesprochen wichtiger Punkt ist, die Schnittstellen zwischen den Phasen genau zu spezifizieren. Dadurch wird der Übergang zwischen den Disziplinen bzw. Teilphasen effizient und verursacht weniger Informationsverlust.⁴

³ Vgl. Weber (2016), S. 3-5.

⁴ Vgl. Weber (2016), S. 3-5.

Der Ablauf der jeweiligen Phasen ist chronologisch nacheinander, jedoch startet z.B. die erwähnte *Ausführungsplanung* bereits parallel mit der *Entwurfsplanung*. *Beschaffung* und *Bau/Montage* sind sehr eng gekoppelt, da im Zeitraum der Anlagenerrichtung kurzfristig auf entstehende Probleme reagiert wird. Zum Ende des gesamten Planungsablaufes steht die *Inbetriebnahme*, deren Abschluss die Anlagenabnahme durch den Endkunden darstellt.⁵

In den folgenden Unterkapiteln wird auf die einzelnen Teilbereiche des Phasenmodells eingegangen.

2.1.1 Grundlagenermittlung (establishment of fundamentals)

In der ersten Phase werden hauptsächlich grundlegende Entscheidungen getroffen. Die Prüfung auf Durchführbarkeit wird zumeist bereits vor der *Grundlagenermittlung* gemacht. Dazu wird eine *Feasibility study* (Machbarkeitsstudie) erarbeitet, in der die technische und finanzielle Machbarkeit des bevorstehenden Projekts beurteilt wird. Fällt diese positiv aus, wird ein Lastenheft erstellt. Verantwortlich dafür ist das auftraggebende Unternehmen bzw. die auftraggebende Institution. Darin sind technische Standardspezifikationen festgehalten, die für die Errichtung der Anlage einzuhalten sind. Oftmals haben Unternehmen bereits andere Anlagen in Betrieb, bei denen schon ein Ausführungsstandard etabliert ist. Um Durchgängigkeit in allen Betrieben des Unternehmens zu gewährleisten, werden neue Projekte nach diesen Ausführungsvorlagen geplant. Dabei handelt es sich um sogenannte *kundenspezifischen Zielvorgaben*.

Überdies werden im Lastenheft Voraussetzungen wie Energiebereitstellung, Beistelleleistungen, Gesundheitsaspekte sowie Werkstätten, Labore, Sozialeinrichtungen und Umweltaspekte definiert. Bezogen auf die Umweltverträglichkeit werden Emissionswerte (z.B. Feinstaub, Stickoxide oder Geräuschemission), Gefahrenstoffe und sicherheitstechnische Aspekte sowie gesetzliche Ausführungsvorschriften und Nachhaltigkeit festgelegt.⁶

2.1.2 Die Vorplanung (Pre-Planning)

Voraussetzung sowie Hauptunterlage für das *Preliminary-Planning*, oder auch *Vorplanungsphase*, ist das bereits zuvor beschriebene Lastenheft. Darauf aufbauend werden Lösungsansätze für die enthaltenen Aufgabenstellungen erarbeitet. Umgesetzt und visualisiert werden diese zumeist in diversen Konzepten, die bereits eine gewisse technische Ausarbeitungstiefe besitzen, jedoch nicht vollends ausgearbeitet sind.⁷

Zu diesem Zeitpunkt am weitesten gereift ist die Verfahrens- und Prozessplanung. Begründet wird dieser Umstand dadurch, dass andere Disziplinen auf deren Ergebnissen aufbauen, wie z.B. die Sicherheits- und Umweltschutzbetrachtung, die für die Genehmigungsplanung bzw. die Genehmigung der Anlage ausschlaggebend sind. Ausgewählte Entwürfe und Konzepte, die aus der Vorplanung entstehen, sind in den folgenden Unterkapiteln beschrieben.⁸

⁵ Vgl. Weber (2016), S. 5.

⁶ Vgl. Weber (2016), S. 5-6.

⁷ Vgl. Weber (2016), S. 6.

⁸ Vgl. Weber (2016), S. 208-241.

2.1.2.1 Verfahrensauswahl und -entwurf

Als *Verfahren* oder *Prozess* werden alle physikalischen, biologischen und chemischen Vorgänge bezeichnet, die notwendig sind, um aus den Roh- und Hilfsstoffen das gewünschte Produkt herzustellen. Als Grundoperation wird jener Vorgang bezeichnet, der hauptverantwortlich für die stoffliche Umwandlung ist. Nebenverfahren sind technische Standardprozesse, die z.B. zur Bereitstellung von Hilfsstoffen wie Dampf oder Lösungsmitteln verwendet werden. Da das Hauptverfahren zumeist vom Auftraggeber vorgegeben wird, ist es bereits fixiert und im Lastenheft beschrieben. Mit diesem Input wird der Prozess konzipiert, eine grobe Bilanzierung von Stoff- und Energieströmen durchgeführt und ein Grundfließschema erstellt. Es stellt das Verfahren in einfachster grafischer Form dar, indem lediglich Rechtecke in der richtigen Abfolge aneinandergereiht werden. Diese werden z.B. mit *Abgaswäsche* oder *Konzentrat (850m³/h)* näher beschrieben. In der Verfahrensplanung wird, basierend auf dem Grundfließschema, das Verfahrenfließschema entwickelt. Apparate, Behälter und Maschinen sind darin technologisch richtig angeordnet und bereits um Informationen wie Behältervolumen oder Mediumstemperatur erweitert. Ausführliche Ausprägungen beinhalten bereits Messstellen und Regelkreise.⁹

2.1.2.2 Prozessleittechnik-Konzept

Die Prozessleittechnik wird als übergreifende Disziplin bezeichnet und fasst mehrere Gewerke zusammen. Darunter sind die Elektro-, Mess-, Steuer- und Regelungstechnik sowie Nachrichtentechnik und das Leitsystem. In diesen Abteilungen gibt es jeweils vieles zu konzeptionieren. Für die Elektrotechnik ist z.B. festzulegen, ob leistungsstarke Verbraucher direkt an die Mittelspannungsebene oder z.B. schwächere Antriebe direkt bzw. mittels Frequenzumformer betrieben werden. Im Umfeld von Leitsystem und Mess-, Steuer- und Regeltechnik gibt es einige Aspekte zu beachten, die in einem Konzept festgelegt werden können. Darunter fällt z.B. der Aufbau der Leitebenen zwischen Regelgrößenaufnahme und Anzeige. Klassisch wird in drei Ebenen untergliedert, der *Feldleitebene*, *Prozessleitebene* und *Betriebsleitebene*. Die Art der Kommunikation zwischen den Ebenen sowie die Verarbeitung (z.B. via SPS oder direkt ins Prozessleitsystem per Feldbus) sind typische Festlegungen im PLT-Konzept. Die Entscheidung, ob die Kommunikation sowie Programmabarbeitung redundant auszuführen ist, ist gleichermaßen wichtig wie die Festlegung zur Ausführungsvariante von sicherheitsgerichteten Steuerungen (z.B. periphere Spezialsteuerungen, hartverdrahtet oder per *failsafe* Prozessleitsystem-Komponenten). Die Ausführung der Feldgeräte in Bezug auf Messprinzip bzw. -signal ist ebenso wichtig wie deren örtliche Unterbringung in z.B. Rohrleitungen. In dieser Phase soll nicht der exakte Einbauort bestimmt, sondern lediglich allgemeine Bestimmungen getroffen werden (z.B. Mindestabstände nach Aktoren). Bedienphilosophie (z.B. eine Leitwarte oder verteilte Bedienstationen) und Automatisierungsgrad (z.B. händische oder automatische Anfahrsequenzen) sind grundlegende Kriterien, die auf die Gesamtheit der Anlage abzustimmen sind.¹⁰

2.1.2.3 Beschaffungskonzept

Hier wird konzeptioniert, wie die Beschaffung von Lieferungen bzw. Leistungen im Projekt eingeteilt wird. Vorrangig werden dabei das Engineering und die Anlagenteillieferungen betrachtet. Der Auftraggeber hat

⁹ Vgl. Weber (2016), S. 208-217.

¹⁰ Vgl. Weber (2016), S. 225-231.

zu entscheiden, ob die komplette Anlage von einem Generalunternehmer geliefert wird oder die einzelnen Anlagenteile selbst zugekauft werden. Abzuwägen sind die vermeintlichen Mehrkosten bei einem Generalunternehmer gegenüber den zu koordinierenden Schnittstellen bei einer Einzelvergabe. Wird der Auftraggeber bei den Phasen bis zur Investitionsentscheidung von einem Engineering-Unternehmen unterstützt, ist im Beschaffungskonzept festzulegen, ob die gleiche Firma auch das Detail-Engineering durchführt. Ein bestimmter Informationsverlust sowie eine Einarbeitungsphase müssen auch bei ausführlicher Dokumentation in Kauf genommen werden. Ist der Auftraggeber der Generalunternehmer, muss entschieden werden, welche Anlagenteile als sogenannte *Packages* zusammengefasst vergeben werden können. Ziel dieses Konzepts ist es, herauszuarbeiten, wo Kosteneinsparungen möglich sind und wo Konfliktpotenzial in der Abwicklung entstehen kann.¹¹

2.1.2.4 Konzeptbewertung

Die entstehenden Ergebnisse der Vorplanung müssen vor dem Übergang in die nächste Phase bewertet werden. Dabei muss unter anderem darauf geachtet werden, dass Zielvorgaben des Lastenhefts erfüllt sind, dass gesetzliche Auflagen eingehalten werden, dass sich die geplante Ausführung im Rahmen des Kapitals befindet und dass Funktionalität sowie Verfügbarkeit gegeben sind.¹²

2.1.3 Die Entwurfsplanung (Basic-Engineering)

Ziel der *Entwurfsplanung* ist ein verbindliches Design der Anlage und der darin befindlichen Technik sowie die verfahrenstechnische Basis für darauf aufbauende Planungsphasen (*Ausführungsplanung*, siehe Kapitel 2.1.4) zu schaffen. Die Planungstiefe ist, abhängig von der Disziplin, bereits sehr weit gediegen. Grund dafür ist mitunter, dass die im *Basic-Engineering* entstehenden Dokumente für Genehmigungen jeglicher Art herangezogen werden. Ein weiterer Output dieser Phase ist das Pflichtenheft, in dem die Anforderungen sowie Vorgaben an die *Ausführungsplanung* und *Bau* bzw. *Inbetriebnahme* festgelegt sind.¹³

Grundsätzlich wird auch in dieser Phase in denselben Fachdisziplinen gearbeitet. In den folgenden Unterkapiteln werden Ausgewählte näher beschrieben.

2.1.3.1 Verfahrensplanung

Die Verfahrensplanung, oder auch *Basic-Design* genannt, ist im *Basic-Engineering* die Hauptaufgabe und besitzt zu Ende einen sehr hohen Reifegrad. Unterlagen dafür sind unter anderem das Grund- bzw. Verfahrensflißschema, Grobbilanzierungsergebnisse oder PLT-Stellenlisten. Aus diesen Informationen werden unterschiedliche Dokumente bzw. Festlegungen generiert. Eine wichtige Unterlage für die weitere Anlagenumsetzung ist das Rohrleitungs- und Instrumentierungsflißschema, kurz *R&I-Flißschema*. Es ist

¹¹ Vgl. Weber (2016), S. 239-241.

¹² Vgl. Weber (2016), S. 245.

¹³ Vgl. Weber (2016), S. 7-8,414-415.

das zentrale Dokument im Planungsprozess, Änderungen darin bedeuten in den meisten Fällen auch Anpassungen in anderen Planungsschritten. Das R&I-Fließschema stellt z.B. Apparate dar und die Rohrleitungen, die für den Materialzufluss- und abfluss verantwortlich sind, wie in Abbildung 3 ersichtlich.¹⁴

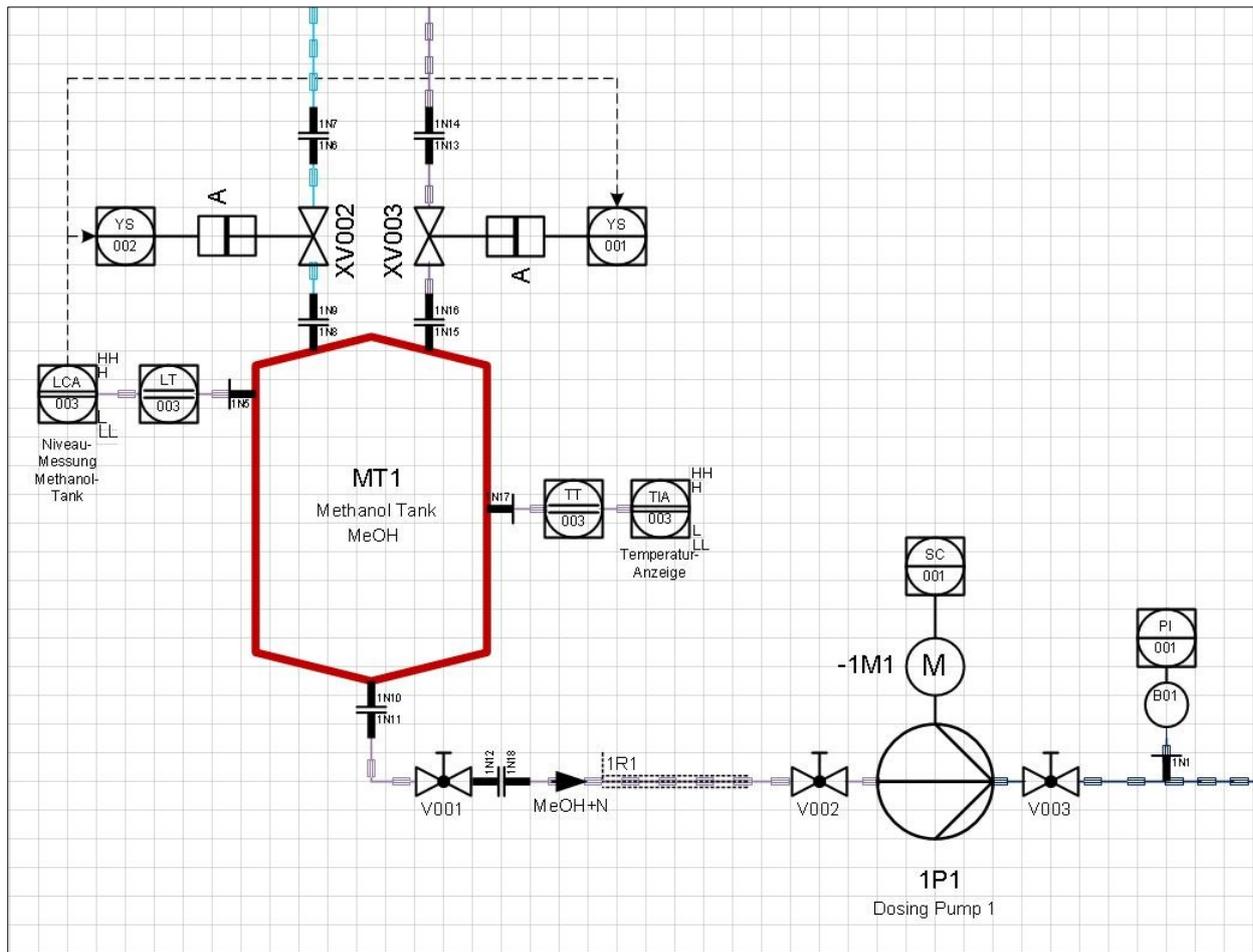


Abbildung 3: Ausschnitt eines typischen R&I-Fließschema, Quelle: Eigene Darstellung

Farblich sowie grafisch unterschiedlich dargestellte Verbindungen weisen auf die verschiedenen Medien hin, die in den Rohrleitungen fließen. Deren Kennzeichnung folgt zumeist einem Standard, in dem z.B. Rohrklasse, Medium und Zählnummer untergebracht sind. Die Symbolik für z.B. Behälter oder Pumpen ist in der *DIN EN ISO 10628* geregelt. Weitere mechanische Einbauten wie Stutzen oder Ventilkörper sind wichtige Bestandteile, um in einem Fließschema den Prozess zu dokumentieren. Es empfiehlt sich, eine Legende im Projekt mitzuführen, die einen Überblick über die verwendeten Symbole, Medien und den Kennzeichnungsstandard gibt. Wie bereits angemerkt, weist das R&I-Fließschema am Ende der *Entwurfplanung* einen sehr hohen Detaillierungsgrad auf. Aus diesem Grund werden elektrische Mess-, Steuer- und Regelungstechnik (EMSR)-Funktionen und -Feldgeräte auch in das Schema eingearbeitet. An diesem Punkt wird die Wichtigkeit des interdisziplinären Arbeitens sichtbar, da die Verfahrenstechnik stark mit der Prozessleittechnik zusammenarbeiten muss.¹⁵

¹⁴ Vgl. Weber (2016), S. 269-273.

¹⁵ Vgl. Weber (2016), S. 273-279.

Eine nähere Beschreibung der EMSR-Stellen und -Feldgeräte erfolgt in Kapitel 2.1.3.3.

Neben dem Fließschema hat die Verfahrensplanung noch andere wichtige Punkte, die erarbeitet werden. Darunter ist die Bilanzierung von Massen, Stoffen und Energien. Dies ist notwendig, um Hauptkomponenten wie Reaktoren oder Wärmeüberträger auszulegen, maximale Drücke oder Temperaturen zu definieren oder die Druckstufen für Rohre, Armaturen und Feldgeräte festzulegen. Die Auslegung dieser Parameter hat direkte Auswirkung auf die Komponentenauswahl anderer Fachdisziplinen. Abgerundet wird die Verfahrensplanung durch eine schriftlich verfasste Anlagen- oder Prozessbeschreibung, die für z.B. sicherheitstechnische Ausarbeitungen eine wichtige Unterstützung zu den R&I-Fließschemas bildet.¹⁶

2.1.3.2 3-D-Anlagenentwurfsplanung

Die 3D-Anlagenplanung setzt auf die Layoutplanung des *Pre-Engineerings* auf. Weitere wichtige Vorlagen sind das Fließschema (die Schnittstellenkoordination ist wichtig, da oft simultanes Engineering an mehreren Fronten notwendig ist), Abmessungen der Hauptausrüstungen oder gesetzliche Rechtsvorschriften. Aufgrund dieser Inputs wird ein dreidimensionales Anlagenmodell erstellt, dessen Detaillierungsgrad nicht so hoch ist wie beim R&I-Fließschema, jedoch bereits weit gereift. Dieses Modell wird anschließend im *Detail-Engineering* verfeinert und für ein weites Feld an Folgeschritten verwendet. Virtuelle Anlagenbegehungen werden vor dem Bau durchgeführt, um mögliche Planungsfehler frühzeitig zu erkennen sowie Kundenwünsche rechtzeitig einfließen zu lassen. Zur besseren Aufbereitung von Montagevorgängen und Inbetriebnahmen werden Ausschnitte des Anlagenmodells bereitgestellt. Dies kann entweder in dreidimensionaler Form oder als 2D-Aufstellungsplan aufbereitet sein.¹⁷

2.1.3.3 Entwurfsplanung PLT

Die Prozessleittechnik spielt für eine der Hauptaufgaben des *Basic-Engineerings*, dem R&I-Fließschema, eine wichtige Rolle. Es müssen die verfahrens- und sicherheitstechnisch relevanten Einrichtungen sowie die Zusammenhänge der Regelkreise im Schema eingetragen werden. Dazu muss das Verfahren in Zusammenarbeit mit der Prozesstechnik betrachtet werden, um die notwendigen Regelungen auszuarbeiten. Aufgrund der ausgewählten Regelungen (z.B. Kaskadenregelung) sind die notwendigen Messstellen sowie Stellstellen festzulegen und in das Schema zu übertragen. Die Summe dieser Stellen wird in Messstellen-, Stellstellen- und E-Verbraucherlisten zusammengefasst. Deren Ausarbeitung beginnt in der PLT-Entwurfsplanung, erreicht jedoch noch nicht den hohen Detaillierungsgrad des Fließschemas. Zu jeder dieser Stellen wird ein Stellenblatt erstellt, in dem Informationen zu den Sensoren und Aktoren enthalten sind, darunter z.B. Prozessdaten (Druck), Stoffdaten (Medium) oder die örtliche Anordnung. In der *Entwurfsplanung* werden nur wenige Informationen eingepflegt, jedoch sollte für jede Stelle ein zugehöriges Blatt angelegt werden, das im *Detail-Engineering* vollständig befüllt wird. Die darin beschriebenen Feldgeräte sind unter anderem Temperatur- oder Drucksensoren wie auch Regelventile oder Pumpen. Ebenso werden Grenz- und Alarmwertlisten angelegt. Darin sind alle Messstellen enthalten mit z.B. Grenztemperaturen, -füllständen oder -drücken. Sicherheitstechnische Aspekte müssen für Grenzwerte und Alarme beachtet werden. Daher

¹⁶ Vgl. Topole (2018), S. 33.

¹⁷ Vgl. Weber (2016), S. 331-335.

soll die Prozessleittechnik bei der sogenannten *Risikobeurteilung* mitwirken, um über die sicherheitstechnischen PLT-Einrichtungen mitentscheiden zu können und z.B. das NOT-AUS-Konzept erstellt werden kann.¹⁸

2.1.4 Die Ausführungsplanung (Detail-Engineering)

Nachdem die ersten Planungsarbeiten im *Basic-Engineering* durchgeführt sind, wird mit dem *Detail-Engineering* begonnen. Wie bereits erwähnt, ist zu diesem Zeitpunkt die verfahrenstechnische Planung sehr weit gereift und weitestgehend abgeschlossen. Ein großer Anteil an der weiteren Entwicklung der Anlage liegt nun bei der mechanischen, elektrotechnischen sowie mess-/regeltechnischen Planung.¹⁹

Häufig ist es der Fall, dass das *Detail-Engineering* nicht mehr in der Hand des Investors liegt, sondern an einen Generalplaner oder mehrere einzelne Unternehmen übergeben wird. Daraus entsteht hohes Fehlerpotenzial, z.B. durch neue Zuständigkeiten und Unklarheiten aufgrund unzureichender Dokumentationsqualität.²⁰

2.1.4.1 Das mechanische Detail-Engineering

Als mechanisches Engineering werden unter diesem Kapitel die Rohrleitungsplanung sowie die Konstruktion von Behältern, Apparaten und Maschinen zusammengefasst. Zusammengeführt werden die jeweiligen Ergebnisse in der finalen 3D-Anlagenplanung.

Verfahrenstechnikerinnen und Verfahrenstechniker haben ihre Hauptaufgaben bereits im *Basic-Engineering* erledigt und arbeiten nun stark mit anderen Gewerken zusammen. Ihre Überlegungen und Planungen sind Grundlage für weitere Umsetzungen, daher ist ein reger Informationsaustausch sehr hilfreich, um die zuvor angesprochenen Qualitätsverluste niedrig zu halten. Ein wichtiger Punkt ist, alle entstandenen Dokumentationen zu aktualisieren und nachzuführen, wenn sich im Zuge des *Detail-Engineerings* Anpassungen ergeben. Hier durchgeführte Planungen sind in dieser Phase z.B. die Feinbilanzierung von Massen- und Stoffströmen oder die Auslegung von Wärme- bzw. Kälte-dämmung von Apparaten, Behältern und Tanks. Diese Auslegung wird dann von einem Konstrukteur oder einer Konstrukteurin technisch detailliert ausgearbeitet. Meist werden Konstruktionen direkt vom herstellenden Unternehmen durchgeführt, daher sind alle Auslegungsdaten für die Ausrüstung entscheidend und die Haftung für deren Richtigkeit festzulegen. Beim Konstruieren selbst ist besonders auf geltende Normen und gesetzliche Vorgaben zu achten (Druckgeräterichtlinie, ATEX-Explosionsschutzrichtlinie). Das Außerachtlassen dieser Vorgaben kann dazu führen, dass im äußersten Fall ganze Anlagen nicht betrieben werden dürfen. Zum Thema der Langlebigkeit und Haltbarkeit von Anlagen ist die mögliche Korrosion von z.B. Behältern zu beachten. Abschließend müssen alle Umsetzungen auf Konformitäten geprüft und gekennzeichnet werden, wie z.B. durch die europäische CE-Kennzeichnung zur Einhaltung von EU-Normen und Richtlinien.²¹

¹⁸ Vgl. Weber (2016), S. 349-370.

¹⁹ Vgl. Topole (2018), S. 49.

²⁰ Vgl. Weber (2016), S. 9-10.

²¹ Vgl. Weber (2016), S. 482-500.

2.1.4.2 Das Elektrotechnik-, EMSR- und Prozessleittechnik-Detail-Engineering

Wie bereits kurz erwähnt, entfalten die drei Disziplinen Elektrotechnik, Mess-, Steuer- und Regelungstechnik und Prozessleittechnik erst im *Detail-Engineering* ihren Detaillierungsgrad. Da sich die vorliegende Arbeit in ihrem Praxisteil mit der Detailplanung dieser Gewerke beschäftigt, wird dieses Kapitel ausführlicher behandelt.

Diese drei Disziplinen haben in der Planung von verfahrenstechnischen Anlagen viele Schnittstellen und verlangen enge Zusammenarbeit der handelnden Personen. Der Zusammenhang wird in Abbildung 4 ersichtlich.

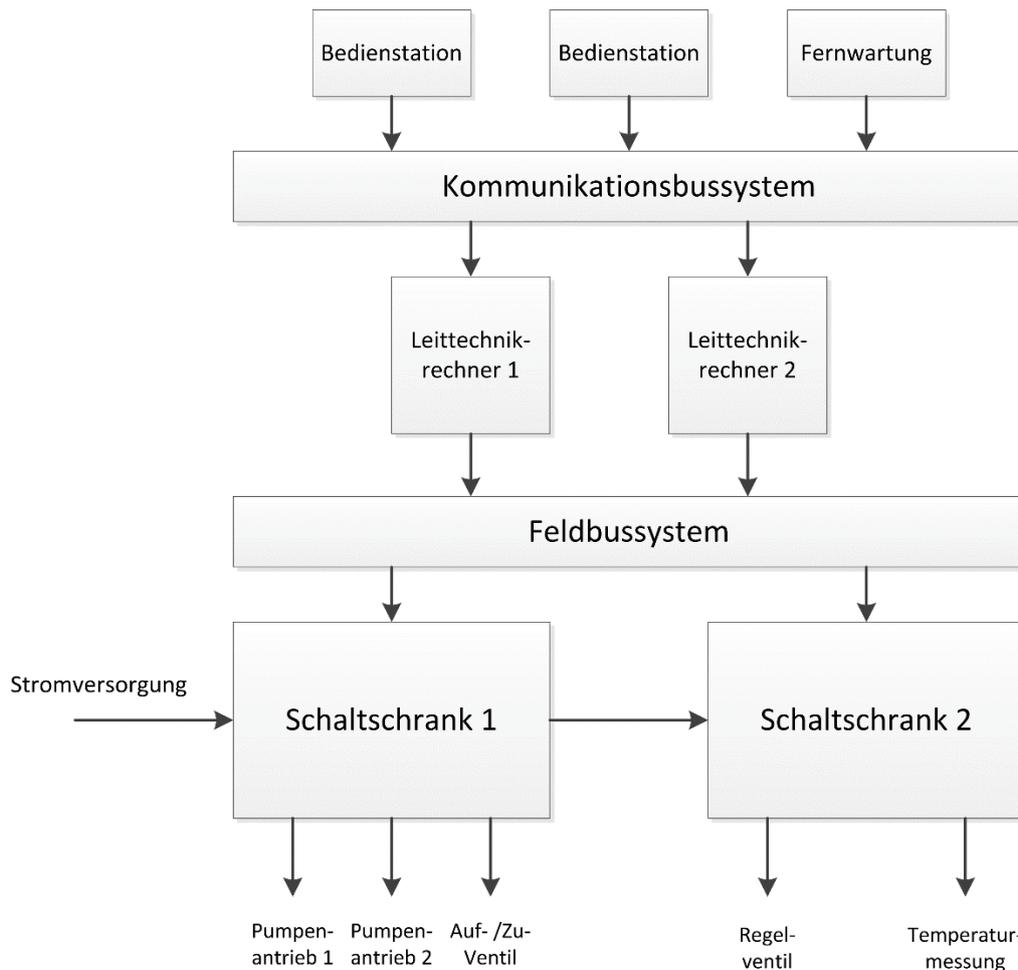


Abbildung 4: Prinzipschaltbild einer Elektrotechnik/EMSR-Anlage, Quelle: In Anlehnung an Helmus (2003), S. 152.

EMSR-Techniker und Technikerinnen wählen in enger Zusammenarbeit mit Verfahrenstechnikern und Technikerinnen die geeigneten Messgeräte und Ventile für den Prozess aus. Verarbeitet werden die gemessenen Signale zumeist in erster Instanz in einem Schaltschrank, der entweder vor Ort oder in einem Schaltraum steht. Diese können reine Mess-, Regel- und Steuertechnikverteiler sein, oder ebenso Elektrotechnikkomponenten enthalten. Die Elektrotechnikplanung hat zur Aufgabe, die Spannungsversorgung dieser Schränke sicherzustellen. Große elektrische Antriebe, wie sie bei Pumpen oder Rührwerken vorhanden sind, müssen mit Spannung versorgt werden. Dazu muss die Elektrotechnikplanung geeignete Absicherungen und Schaltelemente sowie Kabelquerschnitte zur Verfügung stellen. Übergeordnet gesteuert wer-

den sowohl die EMSR-, als auch die Elektrotechnikkomponenten durch Leittechnikrechner. Die gemessenen Signale werden verarbeitet und aufgrund von hinterlegten logischen Verknüpfungen steuert das Programm die Pumpen oder Regelventile. Zur Kontrolle und für Eingriffe durch das Bedienpersonal in den Prozess werden die aktuellen Werte der Anlage auf Bedienstationen in Leitwarten visualisiert. Um dieses Personal zu unterstützen und auf Störungen zu reagieren, ist es üblich, einen Fernzugang zu Wartungszwecken einzurichten.²²

Folgend wird ein typischer Projektablauf des *Detail-Engineerings* aus Sicht eines Engineering-Dienstleistungsunternehmens erläutert, der aufgrund eines Workshops erarbeitet wurde (siehe Anhang 1). Die Teilnehmer setzen sich aus Projektingenieuren und Projektleitern aus Soft- und Hardwareabteilung zusammen.

Zu aller erst wird Kundenkontakt z.B. über Mitarbeiter, öffentliche Ausschreibungen oder Bestandskunden hergestellt. Danach werden zur Angebotserstellung Basisdaten vom potenziellen Kunden übergeben. Darunter z.B. das Pflichtenheft, R&I-Fließschema, Anlagenlayout, Werksstandard und diverse Listen (z.B. Messkreis- oder Verbraucherlisten). Im Idealfall existiert bereits zu diesem Zeitpunkt eine Funktionsbeschreibung der Anlage. Auf Basis dieser Unterlagen wird erstmals eine grobe technische Ausarbeitung durchgeführt, bei der u.a. die benötigten Ein- und Ausgänge einer dezentralen Steuerung ermittelt und Kabellängen und Querschnitte berechnet werden. Dies wird für die Materialkosten im Angebot benötigt. Darüber hinaus wird möglichst genau der Engineeringaufwand (zeitlich/personell) für Hardware- (E-/EMSR-Technik) und Software- (PL-Technik) Engineering kalkuliert. Abschließend werden bereits vor der Angebotslegung Anfragen an Montagefirmen bzw. Schaltschrankbau durchgeführt, um technische Fragen sowie zeitliche Ressourcen abzuklären. Aufgrund dieser ersten Ausarbeitung entsteht das Angebot, das anschließend dem potenziellen Kunden übermittelt wird. Dieser prüft die technische Ausarbeitung und vergleicht es mit Angeboten der Mitbewerberunternehmen. Entspricht es den Vorstellungen, kommt es zu einer Preisverhandlung, bei der kaufmännische und technische Vertreterinnen und Vertreter des Lieferanten anwesend sind und unter anderem die Zahlungsbedingungen (Beträge und Zeitpunkte) festgelegt werden.

Kommt es zur Projektvergabe, starten Kunde und Lieferant mit einem sogenannten *Kick-Off-Meeting*, bei dem die finalen Basisunterlagen übergeben und die Projektmitglieder vorgestellt werden (z.B. durch ein Organigramm und persönliche Anwesenheit). Der Lieferant kann zu diesem Zeitpunkt Fragen zu technischen Details stellen, die für die Angebotserstellung noch nicht entscheidend waren, jedoch in der Planungsphase wichtig sind (Funktionsbeschreibung, Layouts, Terminpläne usw.). Lieferantenseitig gibt es davor oder danach eine interne Projektübergabe von Vertrieb zu Projektabwicklung.

Nach diesem Startgespräch wird beim Lieferanten intern die Ressourcenplanung durchgeführt. Dabei werden die Arbeitspakete aufgeteilt (z.B. in Software und Hardware, jeweils mit Unterpaketen) und Infrastruktur wie z.B. IT-Umgebungen, Projektteams oder das Engineering-Tool-Setup geschaffen. Abhängig von der Projektgröße variiert die Anzahl der Teilnehmer beim internen *Kick-Off-Meeting*, in dem die Aufgaben den Projektteams vorgestellt werden.

²² Vgl. Helmus (2003), S. 151-152.

Anschließend startet die Ausarbeitung der Detailplanung pro Arbeitspaket. Zu Beginn werden die Basisunterlagen aufbereitet, um die Durchführung des *Bulk-Engineerings* möglichst effizient gestalten zu können. Dazu werden die Stellen in den Listen typisiert, d.h. es werden z.B. Messungen, die gleiche bzw. ähnliche Geräte erfordern und daher technisch gleich umgesetzt werden können, mit einem Code gekennzeichnet. Gleiches gilt für Antriebe und Verbraucher. Schnittstellen zu Leitsystemen von Package-Anlagen (Anlagen, die unabhängig von anderen Anlagenteilen geliefert werden) müssen definiert und abgeklärt werden. Diese und weitere Informationen fließen in das Leittechnik-Systemkonzept ein, in dem schematisch alle Automatisierungskomponenten dargestellt und verbunden werden. Einerseits verschaffen sich Leittechniker und Leittechnikerinnen auf diese Weise einen guten Überblick über die Gesamtanlage, andererseits kann die Beschaffung von Automatisierungskomponenten mit langen Lieferzeiten gestartet werden. Diese Geräte werden entweder für interne Tests oder als Beistellkomponenten für den Kunden oder die Schrankfertigung benötigt.

Im nächsten Schritt werden die bereits angesprochenen Typicals erstellt. Dazu müssen Geräteunterlagen entweder vom Kunden eingefordert oder selbstständig organisiert werden. Abhängig von der Projektsituation sind zumeist zu Beginn noch nicht alle Feldgeräte ausspezifiziert. In diesem Fall muss abgeklärt werden, ob diese Geräte durch das *Detail-Engineering* auszulegen sind, oder ein Rücklauf zum *Basic-Engineering* gemacht wird. In vielen Fällen haben Kunden ihre eigenen Standards bereits über Jahre aufgebaut und geben bei Neuplanungen die Typicals für Hardware und Software vor. Ist dies nicht der Fall, werden aufgrund der Geräteunterlagen sowie vorgegebenen Verriegelungen die Typicals ausgearbeitet. Ein wichtiger Punkt dabei ist der interdisziplinäre Abgleich zwischen Hardware und Software. Als Beispiel kann ein direkt gestarteter Motor angeführt werden. In der Hardwareplanung wird ein Motorschutz geplant, der via SPS-Digitalausgang angesteuert wird. Als Freigabe ist dafür ein Digitaleingang der SPS vorgesehen, auf den die Schließerkontakte der Motorschutzeinrichtung und eines Sicherheitsschalters geführt werden. Im Softwaretypical ist daher zu beachten, dass z.B. die beiden Schließerkontakte nicht auf separate SPS-Digitaleingänge signalisieren. Abstimmungsversäumnisse in dieser Phase wirken sich stark bei der Inbetriebnahme aus. Ist intern ein Konsens zum Aufbau der Anlagentypicals gefunden, werden diese mit dem Kunden besprochen und freigegeben. Anpassungen durch Kundenwünsche sind in dieser Phase noch komfortabel und mit wenig zeitlichem Aufwand realisierbar.

Anschließend kann mit den aufbereiteten Basislisten und den Typicals die finale Anfrage an die Montage bzw. den Verteilerbau durchgeführt und die Bestellung ausgelöst werden.

Vor dem folgenden Schritt muss sichergestellt werden, dass die Typicals einwandfrei funktionieren und fehlerfrei sind, um Folgedokumente richtig zu erstellen und technische Fehler nicht auf das gesamte Engineering auszubreiten. Beim *Bulk-Engineering* werden die vorbereiteten Typicals für jede Mess-, Stell- und Verbraucherstelle sowie für Signalaustausche und Einzelsignale in Pläne umgewandelt. Dies geschieht gleichermaßen im Soft- und Hardware-Engineering. Meist werden parallel oder im Anschluss die allgemeinen technischen Ausarbeitungen erledigt, wie Spannungsversorgungen für schrankinterne Geräte oder Kommunikationsschnittstellen zu Package-Anlagen. Sind alle Soft- und Hardwarepläne erstellt, werden diese mittels *Vier-Augen-Prinzip* zuerst vom Ersteller oder der Erstellerin und anschließend vom dem oder der Arbeitspaketverantwortlichen geprüft und freigegeben. Erst dann kann eine Übergabe an die Schrankfertigung bzw. Montage erfolgen.

Für die Übergabe stehen verschiedene Dokumente zu Auswahl. Abhängig von der zeitlichen Achse des Projekts kann es auch dazu kommen, dass Verteiler mit Hilfe von Hardwaretypicals und diversen Listen gefertigt werden. Üblicherweise erhält die Fertigung jedoch einen finalen Planungssatz, in dem eventuelle Anpassungen händisch eingetragen werden. Vor der Schrankabnahme beim Hersteller wird ein *FAT (Factory Acceptance Test)* durchgeführt. Dabei wird der Verteiler u.a. auf Vollständigkeit geprüft, offensichtliche Schäden an Geräten gesichtet oder interne Schranksignale zur SPS getestet. Nach Prüfung der Konformitäten werden die Verteiler an die Anlagenbaustelle ausgeliefert. Dort werden diese von der Montage im Zuge des Baus und der Inbetriebnahme aufgestellt und verkabelt. Dazu notwendige Dokumentation wie Anlagenlayouts und diverse Listen werden durch das *Detail-Engineering* zur Verfügung gestellt.

Laufend im *Detail-Engineering* wird kaufmännisches und technisches Controlling durchgeführt. Dabei wird die Einhaltung der abgeschätzten Engineeringstunden und des Zeitplanes gleichermaßen überwacht wie technische Änderungen, die sich in der Projektlaufzeit ergeben. Für Anpassungen an den Basisunterlagen ist ein *Claim-Management* zu empfehlen, bei dem die Aufwände für Änderungen mitgeführt und kommuniziert werden. Im Zuge der Projektabrechnung werden Nachträge und entfallene Punkte gegenübergestellt, um zu einer für beide Seiten vertretbaren Abrechnung zu kommen.

Nach erfolgtem Bau und Inbetriebnahme, auf die näher in Kapitel 2.1.5 eingegangen wird, ist durch das *Detail-Engineering* die finale Anlagendokumentation, auch *As-Built-Dokumentation* genannt, zu erstellen. Dabei werden alle aufgetretenen Änderungen in die Dokumentation eingepflegt und das Betriebshandbuch (Bedienungsanleitung der Leittechnik) fertiggestellt. Software-Sicherungen, CAE-Dateien, Prüfprotokolle, Geräteunterlagen und weiter wichtige Dokumentationen werden in Zusammenarbeit mit allen Gewerken erstellt und abschließend dem Kunden übergeben.

Nach Abnahme der Anlage durch den Kunden wird zumeist ein *Lessons-Learned* abgehalten, in dem über das Projekt resümiert wird und die positiven sowie negativen Aspekte besprochen werden. Für folgende Projekte werden diese Erkenntnisse festgehalten und bei Bedarf kann auf die gewonnene Erfahrung zurückgegriffen werden.

2.1.5 Die Bau-, Montage- und Inbetriebnahmephase

Die Bau- und Montagephase ist gegebenermaßen der letzte Schritt vor Inbetriebnahme und schlussendlich Anlagenübergabe an den Endkunden. Der Baustart kann jedoch bereits zu einem frühen Zeitpunkt, mit dem Erschließen des Anlagengeländes, beginnen. Um alle Bauarbeiten durchführen zu können, wird einiges an Planungsaufwand investiert. Notwendige Baustelleneinrichtungen sind u.a. Materiallager, Vorfertigungsplätze, Energieversorgung, Sozialräume (z.B. WCs, Duschen, Essensbereiche) oder Büroräumlichkeiten. Für den Transport von großen, sperrigen Betriebsmitteln ist der Anfahrtsweg von der Fertigungsstelle zu betrachten. Unter Umständen müssen Straßen verbreitert oder erst angelegt werden. Daher ist die Planung des Anlagenbaus bzw. der Anlagenmontage bereits zu einem frühen Zeitpunkt zu betrachten. Erste Arbeiten auf der Baustelle sind u.a. Erdbewegung, Betonieren der Fundamente und die Installation von Abwasser- sowie Regenwasserkanälen. Anschließend werden die Bau- und Stahlbauarbeiten durchgeführt, bevor die Einbringung und Montage großer Apparate, Maschinen oder Behältern vonstatten geht. Danach werden kleinere Betriebsmittel eingebracht und die bereits vorgefertigten Rohrleitungen montiert.

Arbeiten am Gebäude selbst werden stetig parallel zur Montage vorgebracht. Die EMSR-Montage startet, wenn der Rohrleitungsbau soweit fortgeschritten ist, dass Beschädigungen an den montierten Feldgeräten und Kabeln ausgeschlossen werden können. Schaltschränke, die in Schalträumen aufgestellt werden, können bereits zuvor eingebracht werden, da durch diese Räumlichkeiten zumeist keine Prozess-Rohrleitungen geführt werden. Die Montage gilt grundsätzlich dann als abgeschlossen, wenn Druckproben an Apparaten, Behältern und Rohrleitungen durchgeführt wurden, Kupplungssitz und Drehrichtung von Maschinen und Antrieben geprüft wurden und die, soweit möglich, Leerlaufproben mit Motoren und Maschinen abgeschlossen sind. Die elektrische Prüfung der EMSR-Geräte und aller anderen Verbraucher ist, wie auch bei mechanischen Komponenten, zu dokumentieren und als Nachweis dem Inbetriebnahmeteam sowie dem Anlagenbetreiber zu überreichen.²³

Mit der Inbetriebnahme kann teilweise bereits begonnen werden, obwohl die Montage noch nicht in der ganzen Anlage fertiggestellt ist. Bereits vollständig montierte Anlagenteile werden dabei einfach vorgezogen. Die Inbetriebnahme bedarf Vorbereitungen in terminlicher und technischer Hinsicht, da sich ansonsten zeitliche Überschneidungen ergeben können. Zum Beispiel können Systemtests erst nach dem Funktionstest durchgeführt werden. Aufgaben der Inbetriebsetzung sind, chronologisch aufgelistet, Druckproben (wenn nicht schon nach Montage durchgeführt), Funktions- und Systemtests, Kalt- und Warm-Inbetriebsetzung und abschließend die Reinigung der Anlage. Während der Inbetriebnahme wird meist das Betriebspersonal geschult, indem es operativ am erstmaligen Start und der Prozessoptimierung mitwirkt. An der Anlage selbst werden Funktionstests durchgeführt, wie z.B. Funktionsprüfungen von Auf-/Zu-Armaturen und Regelventilen, Antrieben und der Messtechnik. Dabei steht ein Inbetriebnehmer bzw. eine Inbetriebnehmerin mit der Leitwarte in Kontakt und überprüft die Funktion oder kalibriert z.B. eine Druckmessung. Sind diese Einzeltests abgeschlossen, werden funktionell zusammengehörende Anlagenteile gemeinsam einem Systemtest unterzogen. Als Beispiel kann hier das Befüllen und Entleeren eines Behälters angeführt werden, wobei alle Armaturen und Messungen gemeinsam im Zusammenspiel getestet werden. Bei der kalten Inbetriebsetzung wird die Anlage mit Wasser gefüllt und die Systemsteuerung gestartet. Die Regelungen werden mit diesem Medium vorgetestet, bevor es in der warmen Inbetriebnahme mit dem Originalmedium optimiert und finalisiert wird. Abschließend wird die Anlage gereinigt, bevor sie dem Kunden übergeben wird. Als Nachweis, dass die Betriebsmittel zum Zeitpunkt der Übergabe funktionstüchtig waren, werden Prüfprotokolle erstellt. Darin wird festgehalten, wie getestet wurde und wie sich das Betriebsmittel verhielt.²⁴

2.2 Dokumente im Planungsprozess

Im Laufe des Planungsprozesses entsteht eine große Menge an Dokumenten in den verschiedenen Disziplinen. Im Kontext dieser Arbeit wird lediglich auf die wichtigsten eingegangen, die im Zuge des Basic- und Detail-Engineerings für die Disziplinen EMSR, Elektrotechnik und Prozessleittechnik notwendig sind.

²³ Vgl. Sattler/Kasper (2000), S. 931-933.

²⁴ Vgl. Helmus (2003), S. 199-205.

2.2.1 Das Lastenheft

Zu Beginn steht das *Lastenheft*, in dem der Investor bzw. Auftraggeber seine Vorstellung der zu liefernden Anlage darlegt. Es ist die Aufgabenstellung an alle Lieferanten, wie diese ihre Anlagenteile umzusetzen haben. Dabei gilt, je genauer das Lastenheft spezifiziert ist, umso einfacher ist es für den Auftraggeber, die Qualität der gelieferten Anlage zu beurteilen. Ein schlecht spezifiziertes Lastenheft birgt Potenzial für finanzielle Nachforderungen der Lieferanten, da fehlende Definitionen als nachträgliche Kundenwünsche ausgelegt werden können. Im Laufe des Engineerings zeigt sich, was aus dem Lastenheft wirklich technisch umsetzbar ist und im Einklang mit Normen, Vorschriften, Gesetzen und Standards steht. Diese schlussendliche Umsetzung wird von den Lieferanten in einem *Pflichtenheft* zusammengefasst und mit dem Kunden abgestimmt und freigegeben.²⁵

2.2.2 Das R&I-Fließschema

Das unter Kapitel 2.1.3.1 bereits beschriebene R&I-Fließschema ist das zentrale Dokument der Verfahrenstechnik. Grundlage sind die allgemeiner gestalteten Prozessfließschemata. Der Informationsgehalt eines R&I-Fließschemas ist sehr hoch und für viele nachfolgende Disziplinen die Arbeitsgrundlage, da darin detailliert jeder Anlagenteil mit folgenden Komponenten dargestellt wird:²⁶

- Apparate, Behälter und Maschinen
- Rohrleitungen und Armaturen
 - o Rohrleitungskennzeichnung u.a. mit Medium, Nennweite, Fließrichtung
- EMSR-Einrichtungen mit PLT-Wirklinien
- Dämmungen und Begleitheizungen
- Sicherheitseinrichtungen
- Probenahme-Anschlüsse
- Sicherheitseinrichtungen

2.2.3 Die EMSR-Stellenliste

Wie beschrieben, werden im R&I-Fließschema die EMSR-Stellen dargestellt. Um diese Stellen für die weiterführende Planung kompakt aufzubereiten, werden *EMSR-Stellenlisten* erstellt. Darin sind alle Stellen aufgelistet und mit zusätzlichen Informationen versehen, die meist nicht im R&I-Fließschema ersichtlich sind wie Prozessgrößen oder Alarmwerte. Die eindeutige Kennzeichnung wird über die Stellennummer hergestellt. Als Erleichterung für die Planerinnen und Planer der EMSR-Detailplanung werden alle relevanten Daten zu den Stellen nochmals in der Liste angeführt. Darunter z.B. Bezeichnungen, Einbauorte oder Angaben zu den Feldgeräten (siehe Abbildung 5).²⁷

²⁵ Vgl. Weber (2016), S. 154-156.

²⁶ Vgl. Topole (2018), S. 43-46.

²⁷ Vgl. Bindel/Hofmann (2017), S. 43-44.

KKS-Nummer	Allgemeine Informationen			Auslegungs- und Betriebsbedingungen			
	Bezeichnung	Langtext 1 (Deutsch)	Einbauort/Position	Mediums-Temperatur / Druck (Auslegung)	Messbereit - Anfang (MBA)	Messbereich - Ende (MBE)	Umgebungstemperatur
TKNDB01CT004	Temperaturmessumformer inkl. Einschweißtauchhülse	Temperatur Heizungswasser RL	Trockenkammern 1 NEU RL DN150	Wasser Q+110°C 10 bar	0°C	+120°C	-20 bis +50°C
TKNDB02CT001	Temperaturmessumformer inkl. Einschweißtauchhülse	Temperatur Heizungswasser RL	Trockenkammern 6 RL DN250	Wasser Q+110°C 10 bar	0°C	+120°C	-20 bis +50°C
TKNDB10CT001	Temperaturmessumformer inkl. Einschweißtauchhülse	Temperatur Heizungswasser RL	Trockenkammer 6a RL DN150	Wasser Q+110°C 10 bar	0°C	+120°C	-20 bis +50°C
HZNDA40CP001	Druckmessumformer	Druck Heizungswasser VL	Hallenheizung 23b VL DN150		0 bar	+10 bar	-20 bis +50°C
HZNDA50CP001	Druckmessumformer	Druck Heizungswasser VL	Hallenheizung 23c VL DN150		0 bar	+10 bar	-20 bis +50°C
K1NAE40CP001	Differenzdrucktransmitter	Differenzdruck Dampf/Kondensat	Kondensator NEU DN300		0 bar	+6 bar	-20 bis +50°C

Abbildung 5: Auszug einer Messkreisliste, Quelle: Eigene Darstellung

Aus praktischer Erfahrung merkt der Autor an, dass Stellenlisten auch in *Messkreis-* und *Stellstellenlisten* aufgeteilt werden. Deren Aufbau unterscheidet sich kaum, jedoch kommen spezifische Spalten in jeder der beiden Listen vor. Wie aus den Bezeichnungen der Listen entnommen werden kann, befinden sich in der einen nur Messkreise und in der anderen nur Stellstellen.

2.2.4 Die elektrische Verbraucherliste

Aus den Maschinen, Apparaten und Behältern des R&I-Fließschemas lässt sich eine *Motor-* und *Verbraucherliste* generieren. Zu Beginn enthalten solche Listen jedoch wenig Information, meist weniger als EMSR-Stellenlisten, sondern lediglich die Kennzeichnungsnummer, den Einbauort und einen beschreibenden Langtext. Die Leistung von Pumpen oder Rührwerken wird durch die Verfahrenstechnik definiert und somit ergibt sich schlussendlich die elektrische Leistung eines Antriebes. Weitere Daten, wie z.B. die Spannungsebene, werden durch die Elektrotechnik erarbeitet und in die Liste eingepflegt. Auch elektrische Heizungen oder EMSR-Verteiler werden in die Verbraucherliste aufgenommen. Damit entsteht ein komplettes Bild aller elektrischen Verbraucher einer Anlage und dessen Leistungen. Aufgrund dieser Angaben können schlussendlich die Kabelquerschnitte der Verbraucher berechnet und zugeordnet werden. Zuordnungen von Antrieben, Rohrbegleitheizungen oder Verteilerversorgungen zu z.B. elektrischen Schaltanlagen werden in diesen Listen vorgenommen.²⁸

²⁸ Vgl. Helmus (2003), S. 153.

2.2.5 Die E-Dokumentation

Auf Basis der beschriebenen Listen wird die *Elektro-* und *EMSR-Dokumentation* erstellt. Die übergreifend genannte E-Dokumentation besteht aus mehreren Dokumentationsarten, darunter das Schaltschranklayout, die allgemeinen Stromlaufpläne, die Stellenpläne (Verbraucher und Messungen) und Folgedokumente wie Klemmenpläne, Kabellisten oder Gerätelisten.

Im Schaltschranklayout wird ein zwei- bzw. dreidimensionales Abbild des Schaltschranks erstellt. Darin sind alle notwendigen Geräte wie Klemmen, SPS-Komponenten, Sicherungskomponenten oder Messwandler angeordnet. Abhängig vom Detaillierungsgrad werden Kabelkanäle, die zur Durchführung von Kabeln und Drähten dienen, eingezeichnet. Diese Dokumentenart wird hauptsächlich zur Fertigung der Verteiler und der Inbetriebnahme angefertigt. Im laufenden Betrieb wird sie von der Instandhaltung bei der Fehlersuche verwendet.²⁹

Ein Stromlaufplan ist die Darstellung einer elektrischen Schaltung. Damit werden die Lösung einer technischen Aufgabenstellung und die verwendeten Betriebsmittel mit ihrer Kennzeichnung (auch Betriebsmittelkennzeichen, kurz *BMK* genannt) dargestellt und etwaige Zusatzinformationen zur Schaltung textuell hinzugefügt. Dabei wird sprachlich zwischen Stromlaufplänen und EMSR-Stellenplänen unterschieden. Ersteres trifft auf allgemeine Spannungsversorgungen, Absicherungen, Automatisierungskomponenten und Motorpläne zu. Stellenpläne sind die Darstellung der elektrischen Anbindung von Messgeräten und Stelleinrichtungen an das Prozessleitsystem. Wie in Anhang 2 ersichtlich, wird der Stellenplan in mehrere Ebenen unterteilt, die in der realen Anlage örtlich weit voneinander entfernt sein können. Durch diese Art der Darstellung ist der Stellenplan sehr übersichtlich. Oftmals wird sowohl elektrische als auch pneumatische Hilfsenergie gleichzeitig dargestellt, um einen allesumfassenden Plan zu erhalten. In sehr detaillierten Ausführungen ist auch die logische Verschaltung des Softwareprogramms dargestellt.³⁰

Typischerweise werden Pläne von oben nach unten *gelesen*, jedoch gibt es auch die Möglichkeit, die Anordnung von links nach rechts aufzubauen. In Anhang 2 befindet sich zuoberst die Darstellung eines SPS-Kanals mit BMK und Anschlüssen des Geräts sowie textueller Beschreibung der Stelle. Darunter folgt ein Spannungs/Strom-Umsetzer, der extern auf Klemmen und intern auf die SPS-Peripherie verdrahtet ist. Als Hilfsenergie werden 24 Volt Gleichspannung benötigt. Im oberen Teil des Planes sind die Schaltschrankkomponenten dargestellt. Zusatzinformationen wie Ein-/Ausgangssignal des Umsetzers können textuell dargestellt werden, um die Dokumentation aussagekräftiger zu gestalten. Der untere Teil ist die Feldebene, in der die Messgeräte bzw. die Messumformer platziert sind. Zusätzlich wird das Kabel, das das Feld mit dem Verteiler verbindet, hier dargestellt.

Eine Klemme ist ein Verbindungsstück, das z.B. elektrische Signale eines Messgeräts (mittels Kabelverbindung) mit der SPS-Peripherie verbindet. Für die Fertigung, Inbetriebnahme und Instandhaltung werden

²⁹ Vgl. Bindel/Hofmann (2017), S. 112.

³⁰ Vgl. Weber (2016), S. 550-551.

Klemmenpläne erstellt, in denen die angeschlossenen Geräte jeder Klemme in tabellarischer Form aufgelistet sind.³¹ Sie werden als Folgedokumentation direkt aus dem Planungsprogramm generiert und der E-Dokumentation angefügt.

Kabel sind essenziell in verfahrenstechnischen Anlagen, da Feldgeräte oft sehr weit von den Schalträumen entfernt sind. Um einen übersichtlichen Blick auf alle Kabel zu haben, werden *Kabellisten* erstellt. Diese meist aus dem Planungssystem generierten Listen beinhalten jedes Kabel mit der zugehörigen Kabelkennzeichnung, Quelle und Ziel, Länge, Außendurchmesser, Kabeltype und Leiterquerschnitt. Sie werden besonders zur Beschaffung und bei der Montage der Anlage benötigt.³² Detailliertere Kabellisten werden in der Praxis oft mit Informationen zu Absicherung und ähnlichem erweitert.

Der Inhalt von *Gerätelisten* ist abhängig von Disziplin und Planungsfortschritt. Im elektrischen und mess- und regeltechnischen *Detail-Engineering* sind sie meist für die Beschaffung und zu Kontrollzwecken generiert. Als eindeutiger Identifikator wird wieder die Kennzeichnung herangezogen, die um Daten wie Hersteller, Beschreibung, Typennummer, Bestellnummer, Nennspannung und -strom erweitert wird.³³

2.2.6 Funktionsbeschreibung und Funktionsplan

Als Vorgaben für die Umsetzung der Prozessleittechnik wird eine in der Praxis genannte Funktionsbeschreibung durch die Verfahrenstechnik erstellt. In der Literatur wird diese auch als „Verfahrensbeschreibung“³⁴ oder „Pflichtenhefte für die Leittechniker“³⁵ bezeichnet. Wobei zuletzt genanntes eine für die Programmierung zugeschnittene Formulierung und Informationstiefe besitzt. Darin werden alle Abläufe, die im Prozess vonstattengehen, verbal beschrieben, in dem Ausmaß, dass jeder Steuer- und Regelschritt festgehalten wird.³⁶ Im praktischen Gebrauch werden diese Beschreibungen gerne mit einem Ablaufschema grafisch dargestellt.

Das folgende Beispiel ist ein Auszug aus einer typischen Funktionsbeschreibung:

Die Heizungen starten, sobald der Hauptschalter eingeschaltet ist und bleiben in Betrieb, auch wenn der Hauptantrieb ausgeschaltet ist. Die Einstellung und Regelung der Heizung erfolgt lokal. Die Heizungen sind mit einem Temperaturschutz ausgestattet und geben den Alarm „Übertemperatur ausgelöst“ aus.

Mit diesen Vorlagen beginnt die Prozessleittechnik mit der Programmierung des Leittechnikprogramms. Die einzelnen Teile der Funktionsbeschreibung werden in sogenannte Funktionspläne umgesetzt. Sie zeigen die logischen Abläufe, die zur Bewältigung der Aufgabenstellung notwendig sind.³⁷

³¹ Vgl. Bindel/Hofmann (2017), S. 111.

³² Vgl. Bindel/Hofmann (2017), S. 109-110.

³³ Vgl. Weber (2016), S. 358.

³⁴ Sattler/Kasper (2000), S. 211.

³⁵ Helmus (2003), S. 160.

³⁶ Vgl. Helmus (2003), S. 160-161.

³⁷ Vgl. Weber (2016), S. 362-363.

2.2.7 Dokumente der Prozessleittechnik

In diesem Kapitel wird ein kurzer Überblick über die Unterlagen gegeben, die für die PLT notwendig sind und davon produziert werden.

Die *Signalliste* ist ein Abkömmling der Stellenliste. Diese wird um Informationen zu den Ein- und Ausgangssignalen pro Stelle erweitert.³⁸ D.h. aus einer Stelle werden z.B. drei Signale generiert. Aus einem Auf-/Zu-Ventil entstehen zwei digitale Eingangssignale der Stellungsrückmeldungen und ein digitales Ausgangssignal zur Ansteuerung des Ventils.

Das *Regelschema* wird im *PLT-Basic-Engineering* entworfen. Es wird erstellt, um komplexe Reglerstrukturen und Rechenfunktionen grafisch darzustellen. Diese detaillierte Ergänzung zum R&I-Fließschema wird als eigenes Dokument geführt, da ansonsten die Übersicht am Fließschema verloren geht. Dort wird lediglich eine Black-Box mit einem Eingang und Ausgang dargestellt. Abbildung 6 zeigt ein typisches Regelschema, in diesem Beispiel eine Dampf-Durchflussregelung.³⁹

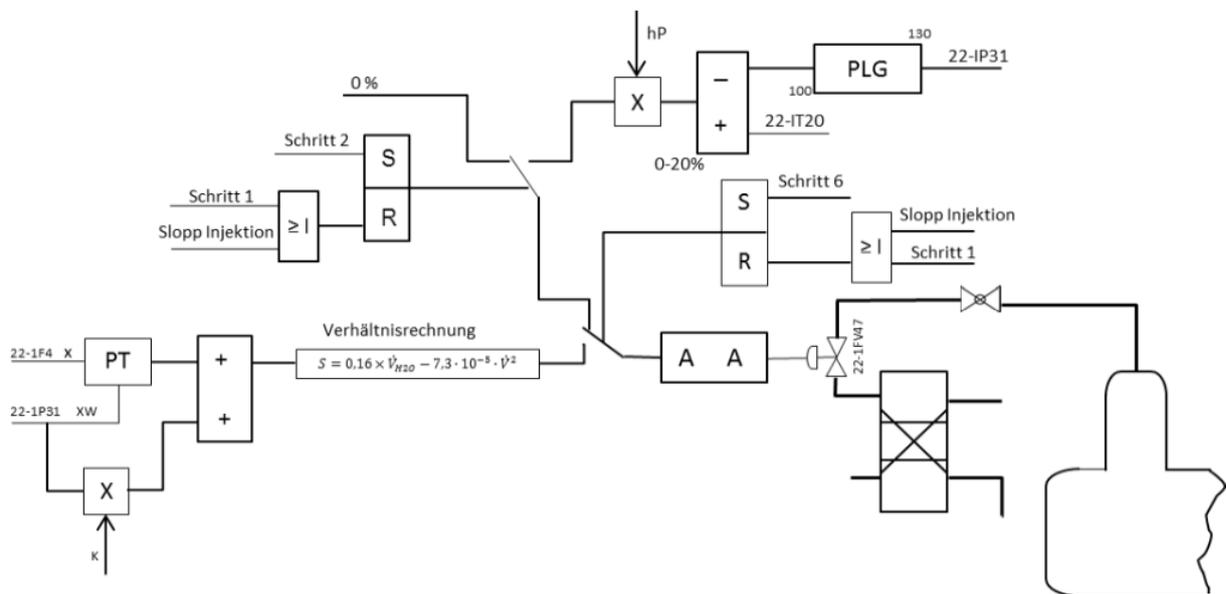


Abbildung 6: Regelschema einer Dampf-Durchflussregelung, Quelle: Weber (2016), S. 360.

Die Bedienung von Anlagen erfolgt heutzutage fast ausschließlich durch Visualisierungen auf einem Bildschirm. In einem *Bedienbild* wird der Prozess der Anlage mit den aktuellen Werten, z.B. Temperaturen, Drücke oder Ventilöffnungsanzeigen, visuell dargestellt. Durch den technischen Fortschritt in punkto Speicher und Rechenleistung ist ein wahrer Hype entstanden, komplette Anlagen genauestens in der Visualisierung darzustellen. Jedoch ist dieser Trend rückläufig und die Gestaltung der Bedienoberfläche für eine effiziente Bedienung und bessere Übersicht tritt in den Vordergrund.⁴⁰

³⁸ Vgl. Bindel/Hofmann (2017), S. 44.

³⁹ Vgl. Weber (2016), S. 360.

⁴⁰ Vgl. Früh/Maier/Schaudel (2015), S. 644.

2.2.8 Conclusio der Planungsdokumente

Es gibt eine Vielzahl an wichtigen Dokumenten und Unterlagen, die im *Basic-* und *Detail-Engineering* entstehen bzw. als Vorlage dienen. Dabei sticht heraus, dass sehr viele davon in einer tabellenorientierten Form vorkommen, darunter Verbraucher-, Stellen-, Messkreis- oder Signallisten. Unterlagen wie das Lasten- bzw. Pflichtenheft oder die Funktionsbeschreibung sind textbasierte Formate, die in Prosa verfasst sind. In der Art der Verarbeitung, durch Mensch oder Maschine, besteht bei diesen beiden Formaten ein großer Unterschied. Da Tabellen strukturiert aufgebaut sind, kann daraus viel Information mit relativ einfachen Mitteln gewonnen werden. Eine Zeile repräsentiert z.B. ein Gerät und die Spalten stehen für zugeordnete Eigenschaften. Da für jedes Gerät die Eigenschaften benannt werden können, können daraus Informationen mit einer strukturierten Vorgehensweise für jedes Gerät erhoben werden. Im Gegensatz dazu muss Fließtext entweder durch einen Menschen oder vergleichsweise aufwendigen Einsatz von komplexen Technologien analysiert und aufbereitet werden.

Alle Listen lassen sich im R&I-Fließschema wiedererkennen, da Komponenten oder Stellen darauf platziert sind. Das macht es zu einem der wichtigsten Dokumente im Planungsprozess. Der Prozessablauf selbst wird zusätzlich im Lastenheft bzw. einer Prozess- oder Funktionsbeschreibung textuell festgehalten.

Die elektrische Dokumentation und die Funktionspläne sind Fertigungsunterlagen, die Eingang in die Fertigung der Anlage finden. Ihr spezifischer Informationsgehalt kann sehr groß sein. Einzeln, um eine Übersicht über das große Ganze zu erhalten, sind diese weniger geeignet. Jedoch ist deren Qualität nicht minder wichtig, da sich Fehler schlussendlich bei der Inbetriebnahme der Anlage massiv auswirken können.

2.3 Agiles Engineering und Claim Management

Die soeben beschriebenen Dokumente sind Ausprägungen der Vorgaben, die durch den Kunden bzw. Anlagenbetreiber an das Engineering weitergegeben werden. Durch Änderungen am Prozess, Lieferprobleme oder aber auch durch gesetzliche Vorgaben kann es zu Anpassungen kommen. Für das Engineering ist es daher entscheidend, in solchen Fällen flexibel zu sein und ohne großen Kostenaufwand reagieren zu können.

Dies kann erreicht werden, indem fixe Strukturen in der Planung vorhanden sind, die jedoch flexibel angepasst werden können, um kurzfristig auf Änderungen zu reagieren. Überdies wird die Kommunikation zwischen den Engineering-Disziplinen sowie den Projektmitarbeiterinnen und -mitarbeitern als sehr wichtig erachtet. Dadurch können Informationen auf kurzem Weg ausgetauscht und frühzeitig in die Planung eingebunden werden. Das agile Engineering beruht dabei auf sogenannten *Iterationen*, *Inkrementen* und *Artefakten*. Ein Produkt oder eine Anlage besteht aus vielen einzelnen Teilsystemen, auch Inkremente genannt. Als Beispiel kann die Verpackungsstation einer Papiermaschine herangezogen werden. Alle diese Teile zusammen ergeben die Gesamtanlage. Diese Inkremente selbst setzen sich aus Artefakten zusammen. Dazu zählen das 3D-Anlagenmodell, die Stromlaufpläne oder das SPS-Programm. Das Engineering aller Teilsysteme wird nacheinander in ähnlich ablaufenden Zyklen, den Iterationen, vollzogen. Am Ende der Planungsphase entsteht so eine Anlage aus mehreren kleinen Teilen, die in ihrer Entstehung stets untereinander abgestimmt werden, um schlussendlich als Gesamtes ihre Funktion erfüllen zu können. Wichtig zu erwähnen ist, dass der Ablauf der Iterationen (darunter z.B. das Erstellen der Stromlaufpläne)

durch Standardisierung zeitlich verkürzt werden kann. Jedoch sollte ein gewisser Freiraum für Anpassungen aufgrund von z.B. Kundenwünschen möglich sein.⁴¹

In Verbindung von Änderungen im Planungsprozess muss stets das Thema *Claims* betrachtet werden. Das englische Wort *claim* wird laut Langenscheidt als Forderung bzw. Anspruch⁴² übersetzt. Diese Forderung erhebt ein Projektpartner, wenn sich Abweichungen zum Vertrag ergeben, wie z.B. Termine, Kosten oder die gewünschten Ergebnisse. Meist entsteht aus technischen Änderungen der Anlage ein terminlicher Verzug und durch mehr Planungsaufwand auch höhere Kosten. Im *Claim-Management* werden Änderungen z.B. der Projektvorgaben überwacht und der Projektpartner damit konfrontiert. Aufbereitet wird dabei das wirtschaftliche und terminliche Ausmaß der Änderung. In vielen Fällen ist dem Gegenüber nicht bewusst, dass die gewünschte Anpassung dementsprechende Folgen nach sich zieht. Die Strategie bei der Durchsetzung von *Claims* kann zwischen einer defensiven (*Claiming* nur bei sicherer Annahme bzw. sofortige Akzeptanz von Fremd-*Claims*) bis zur offensiven (Vertrag nur abschließen mit eigenen Vorteilen bzw. *Claimhöhen* maximieren zum Nachteil des Partners) Haltung reichen. Grundsätzlich bedeutet *Claim-Management* zusätzlichen Einsatz von Ressourcen, jedoch sind komplexe Projekte oftmals nur dadurch wirtschaftlich rentabel durchführbar.⁴³

Änderungen von Engineeringunterlagen gehören dem Planungsalltag an, können jedoch einen großen Aufwand für die Erstellung z.B. des *Detail-Engineerings* bedeuten. Daher gilt es, die beschriebene Flexibilität und Überwachung von Anpassungen im Planungsprozess zu beherrschen.

2.4 Zusammenfassung

Der Planungsprozess von verfahrenstechnischen Anlagen ist sehr komplex und umfangreich. Einige Fachdisziplinen wie Verfahrenstechnik, Rohrleitungstechnik, EMSR- und Elektrotechnik, Prozessleittechnik oder die 3D-Planung sind daran beteiligt. Durch die große Anzahl an Gewerken treten viele Schnittstellen auf, unabhängig ob ein Projekt komplett von einem Generalunternehmer oder durch mehrere Unternehmen abgewickelt wird. Der Prozess kann in zwei Phasen unterteilt werden, der *Konzept-* und der *Ausführungsphase*. Wobei zweiteres durch die detaillierte Planung und den Bau sowie die Inbetriebnahme der Anlage geprägt ist. In der *Konzeptphase* wird der Prozessablauf entworfen, gesetzliche Genehmigungen werden eingeholt, die Projektkosten ermittelt und die Basisdaten für die Ausführung der Anlage im *Basic-Engineering* festgelegt. In der Ausführungsphase werden im *Detail-Engineering* die Pläne zur Komponentenfertigung und der Montage sowie Inbetriebnahme erstellt. Über die gesamte Projektlaufzeit entstehen viele Dokumente, die zur Errichtung der Anlage beitragen. Mitunter das Wichtigste ist das R&I-Fließschema, in dem der Haupt- und die Hilfsprozesse dargestellt sind. Es enthält Rohrleitungen, Apparate, Behälter, Maschinen, EMSR-Stellen und ein weites Feld an Zusatzinformationen. Daraus können für mehrere Planungsdisziplinen Basisdaten generiert werden, die für die Erstellung von z.B. EMSR-Stellenplänen oder Funkti-

⁴¹ Vgl. Klein (2016), S. 84-86.

⁴² Langenscheidt (o.J.), Online-Quelle [15.08.2019]

⁴³ Vgl. PMH (2018), Online-Quelle [17.08.2019]

onsplänen als Basis herangezogen werden. Entstehen über die oft lange Planungszeit einer Anlage Änderungen in den zu Projektstart vereinbarten Vorgaben, sollte möglichst flexibel und mit wenig Aufwand darauf reagiert werden. Um daraus entstehende Kosten erstattet zu bekommen, ist es von Vorteil, ein *Claim-Management* im Projekt zu etablieren.

3 MODELLE UND DAS SYSTEMS ENGINEERING

„Ein Modell ist ein Abbild bzw. ein Vorbild für ein System oder einen Prozess. Ein Modell kann ein begriffliches (z. B. mathematisches, informationstechnisches) oder ein physisches (z. B. maßstäblicher stofflicher Prototyp) Gebilde sein.“⁴⁴

3.1 Die Hauptmerkmale eines Modells

Ein Modell besitzt nach der allgemeinen Modelltheorie nach Stachowiak drei Hauptmerkmale: Das Abbildungs-, das Verkürzungs- und das pragmatische Merkmal. Das Abbildungsmerkmal besagt, dass jedes Modell einem Original nachempfunden ist. Unabhängig, ob dies natürlicher oder technischer Natur ist. Überdies kann auch ein Prozess als Modellvorlage dienen. Im Verkürzungsmerkmal wird beschrieben, dass nicht jedes Attribut des Originals im Modell wiedergegeben sein muss. Es obliegt dem Modellersteller/der Modellerstellerin, zu entscheiden, welche Attribute für das Verständnis des Originals notwendig sind bzw. ob eine exakte Abbildung dessen überhaupt nötig/möglich ist. Oftmals ist es überhaupt nicht möglich, alle Attributwerte bei der Nachbildung eines Originals zu erheben. Wird durch ein Modell eine reale Abbildung generiert, so ergibt sich eine sehr genaue Übereinstimmung. Das letzte Merkmal wird als pragmatisches Merkmal bezeichnet. Es beleuchtet das Modell mit den Fragen: Für wen, wann und wozu. Da Modelle dessen Originale nicht ersetzen, wird hinterfragt, für wen das Modell eingesetzt ist (z.B. Engineeringabteilung oder Betreiber), in welchem Zeitraum ist es gültig (z.B. Planungsphase oder Produktionszeitraum) und wozu wird es gebildet (z.B. Anlagenplanung oder Instandhaltung).⁴⁵

3.2 Modelltypen

Grundsätzlich sind Modelle in zwei Kategorien einteilbar. Zum einen gibt es deskriptive Modelle. Dabei handelt es sich um Abbildungen, die entweder bereits bestehen oder erst zukünftig entstehen, auf dessen Eigenschaften kein Einfluss genommen werden kann. Beispiele dafür sind u.a. Stadtkarten oder Wettervorhersagen. Das Gegenstück dazu sind präskriptive Modelle, bei denen es sich um zu schaffende Modelle handelt, die durch die Modellierung beeinflusst werden können, wie z.B. Konstruktionspläne oder Anforderungsmodelle für Software.⁴⁶

Darüber hinaus lassen sich Modelle aufgrund ihres Zwecks untergliedern:⁴⁷

- Beschreibungsmodell: Damit wird Verhalten und Struktur der Realität deskriptiv erfasst.
- Erklärungsmodell: Erweiterung der Beschreibungsmodelle um Gesetzmäßigkeiten und Zusammenhänge. Dienen zur Erläuterung von Theorien.
- Simulationsmodell: Ermöglichen Prognosen über das Verhalten von Systemen unter gegebenen Rahmenbedingungen. Dazu werden Erklärungsmodelle auf reale Tatbestände angewendet.

⁴⁴ Eigner/Roubanov/Zafirov (Hrsg.) (2014), S. 80.

⁴⁵ Vgl. Stachowiak (1973), S. 131-133.

⁴⁶ Vgl. Glinz (2005), Online-Quelle [31.08.2019]

⁴⁷ Vgl. Kloth (2010), S. 69 f.

- Gestaltungsmodell: Wird auch Problemlösungsmodell genannt, da es vom Erklärungsmodell ausgeht und um Größen erweitert wird, die die Erfüllung von Anforderungen beeinflussen. Dadurch kann eine Aussage über die Ziele des Modells gemacht werden.
- Entscheidungsmodell: Ist ein hypothetisches Modell, das auf die Vorstellungen der Modellanwenderin/ des Modellanwenders eingeht und ein dementsprechendes Ergebnis liefert. Es ist eine andere Form der Gestaltungsmodelle und wird als Entscheidungsgrundlage verwendet.

Bezogen auf Anwendungsgebiete lässt sich eine weitere Unterteilung vornehmen. Dabei handelt es sich um Demonstrationsmodellen, die zur Darstellung von z.B. Prozessen dienen. Das Experimentalmodell dient zur Ermittlung und Prüfung von aufgestellten Thesen, wogegen das theoretische Modell zur Gewinnung dieser Grundlagen herangezogen wird. Ein weiteres ist das operative Modell, dessen Output Entscheidungen und Planungen sind.⁴⁸

Operative Modelle sind über den gesamten Lebenszyklus an ein äußerst genau spezifiziertes und gut strukturiertes Original gebunden. Unabhängig davon, ob dieses bereits existiert oder nicht. Zumeist werden Modelle dieses Typs im technologischen Bereich verwendet.⁴⁹

3.2.1 Prinzipien der Modellierung

Die Modellbildung oder auch Modellierung ist ein Vorgang, bei dem ein Original vereinfacht und in adäquater Form dargestellt wird. Nicht alle Eigenschaften werden in der Abbildung dargestellt, sondern speziell ausgewählte, die zur Darstellung für das gewünschte Modell notwendig sind. Die Vorgehensweise hängt dabei sehr stark von den durchführenden Personen ab. Jedes Modell hat daher definierte Rahmenbedingungen, für die es gültig ist. Unter anderen Umständen besteht keine Garantie, dass die Nachbildung dem Original entspricht. Bei der Durchführung sind drei Ansätze unterscheidbar. Die *Black-Box-Methode*, der *Top-Down* und *Bottom-Up-Ansatz* und die *Variantenbildung*.⁵⁰

Bei der Black-Box-Methode wird das System als *Kiste* betrachtet, d.h. die innere Struktur wird nicht näher untersucht. Durch die Beziehung zwischen Eingabe und Ausgabe wird auf das Verhalten des Systems geschlossen. Somit kann ein komplexes System einfach in ein Gesamtsystem eingebunden werden.⁵¹

Beim Top-Down-Ansatz wird ein System von *oben* als globales Ganzes, mit einem dementsprechend hohen Abstraktionsgrad, betrachtet. Im Laufe des Prozesses wird das Gesamtsystem in einzelne kleinere Systeme untergliedert, die immer konkreter ausgeprägt sind. Der Anzahl der Unterteilungsschritte ist dabei theoretisch keine Grenze gesetzt.⁵²

⁴⁸ Vgl. Kloth (2010), S. 70.

⁴⁹ Vgl. Stachowiak (1973), S. 270 f.

⁵⁰ Vgl. Kloth (2010), S. 70-71.

⁵¹ Vgl. Döppler.Team GmbH (o.J.), Online-Quelle [03.09.2019]

⁵² Vgl. Prof. Dr. Müller-Stewens (o.J.), Online-Quelle [04.09.2019]

Das Gegenteil dazu ist die Bottom-Up-Modellierung, bei der aus konkreten Einzelkomponenten bzw. -lösungen das gesamte Modell zusammengesetzt wird.⁵³

Unter Varianten werden Systeme verstanden, die denselben Output liefern, jedoch andere Funktionalitäten oder Komponenten besitzen. Sie sind in der Modellbildung unerlässlich, da auf diese Weise zwischen den Varianten abgewogen werden kann und die Entscheidung auf die besser geeignete fällt. Es existieren dabei zwei Arten von Varianten. Prinzipvarianten stellen grundsätzlich unterschiedliche Lösungsansätze dar, während sich Detailvarianten eine Ebene darunter auf die genaue Ausarbeitung von Komponenten beziehen.⁵⁴

3.3 Das Metamodell

Ein Metamodell ist dem eigentlichen Modell hierarchisch übergeordnet, es handelt sich dabei um ein Beschreibungsmodell. Dies kommt daher, da die möglichen Bestandteile des Modells im Metamodell beschrieben werden. Überdies werden auch Beziehungen und Abhängigkeiten zwischen den einzelnen Metaobjekten festgelegt. Eigenschaften werden den Objekten zuerst grundsätzlich zugeordnet, um im eigentlichen Modell mit konkreten Werten befüllt werden zu können. Die Typen der Abhängigkeiten bzw. Beziehungen zwischen den Objekten können unterschiedlich sein, darunter z.B. *ist ein*, *gehört zu*, *bietet an* oder *ist Grundlage für*. In Tabelle 1 werden die Unterschiede zwischen den beiden Modellarten gegenübergestellt. Grundsätzlich kann das Metamodell als Bauanleitung eines Modells betrachtet werden. Ein Metamodell kann die Grundlage für mehrere Modelle sein, die als *Instanzen* derer bezeichnet werden.⁵⁵

Metamodell	Modell
Modellgrundlage	Basiert auf Metamodell
Festlegung des prinzipiellen Modellaufbaus	Modell der konkreten Ausprägung
Beschreibt Modellierungssprache	Wird durch Modellierungssprache erstellt
Grundlage für Modellierungstools	Ergebnis der Anwendung von Modellierungstools

Tabelle 1: Unterschiede zwischen Metamodell und Modell, Quelle: In Anlehnung an Brugger (2005), S. 494.

Ein typisches Beispiel für ein Metamodell ist die Sprache. Darin sind Regeln zu Syntax und Semantik festgelegt, d.h., die Strukturierung der Wörter (Grammatik) und welche Bedeutung aus den Wörtern hervorgeht. Das Modell, das daraus instanziiert wird, ist ein Satz wie z.B. *Vor dem Haus steht ein Mensch*. Die konkrete Ausprägung, also das Original dazu, ist das reale Haus mit dem Menschen davor.⁵⁶

In Abbildung 7 werden diese Schichten von Metamodell über Modell zu realem Objekt dargestellt.

⁵³ Vgl. Vertical Media (o.J.), Online-Quelle [04.08.2019]

⁵⁴ Vgl. Kloth (2010), S. 72.

⁵⁵ Vgl. Brugger (2005), S. 493-498.

⁵⁶ Vgl. Gruhn/Pieper/Röttgers (2006), S. 84-85.

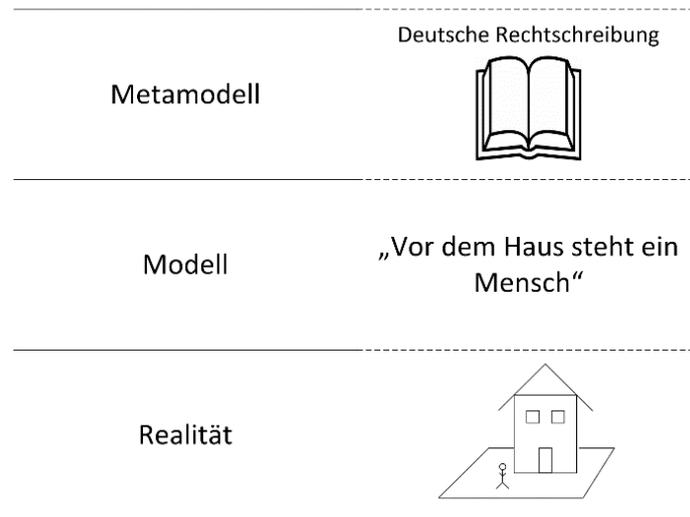


Abbildung 7: Schichten von Metamodell zu Realität, Quelle: In Anlehnung an Gruhn/Pieper/Röttgers (2006), S. 85.

3.3.1 Prinzip der Modellierungssprachen

Auf der gleichen Basis wie soeben beschrieben, funktionieren Modellierungssprachen. Sie beinhalten die Grammatik und das Vokabular, um reale Systeme in Modellen darzustellen. Von der Wahl der Sprache selbst hängt es ab, was durch die Sprache dargestellt bzw. ausgedrückt werden kann. Gewisse bieten Darstellungsmöglichkeiten an, die mit anderen nicht möglich sind. Es gibt verschiedene Sprachen, die dasselbe Ziel haben, sich jedoch im Aufbau unterscheiden. Durch die Verwendung einer bekannten Modellierungssprache wird die Zusammenarbeit von Teammitgliedern untereinander, aber auch mit Computersystemen, vereinfacht. Für Menschen kann das erstellte Modell durch die Sprache leichter verstanden werden und rechnergestützte Systeme haben die Möglichkeit, Automatismen durchzuführen um das Modell weiterzuentwickeln. Durch diese Unterschiede in den diversen Sprachen wird die Wichtigkeit der Sprache im Prozess der Modellbildung sichtbar. Die Auswahl soll daher gut überlegt sein.⁵⁷

Im Folgenden werden ausgewählte Modellierungssprachen näher beschrieben.

3.3.1.1 Domain Specific Language – DSL

Eine Domain Specific Language, DSL, ist eine Sprache, die speziell für eine Domäne, also eine Fachrichtung, entwickelt ist. Diese können sodann als Programmier- oder Modellierungssprache ausgeführt sein. Jeder kann eine DSL selbst entwickeln, es existiert daher nicht *die eine* DSL. Das Hauptaugenmerk bei der Entwicklung einer solchen Sprache liegt darin, die spezifischen Anforderungen des Fachgebiets abzubilden und zu vereinfachen. Je detaillierter die Sprache ist, in der ein System beschreibt ist, umso leichter ist es für einen Experten oder eine Expertin, ein Modell zu erstellen. Im besten Fall ist die Sprache so genau, dass sich z.B. das Softwareprogramm wie die Lösung der Aufgabenstellung lesen lässt. Als Gegenteil dazu existieren *General-Purpose-Languages* wie z.B. die Programmiersprachen C++ oder Java, die nahezu für jede Aufgabenstellung eingesetzt werden können.⁵⁸

⁵⁷ Vgl. Fleischmann/Oppl/Schmidt/Stary (2018), S. 71-72.

⁵⁸ Vgl. Tilkov (2012), Online-Quelle [06.09.2019]

3.3.1.2 Unified Modeling Language – UML

UML ist eine fachgebietsunabhängige Modellierungssprache, die durch die *Object Management Group* in den 90er Jahren der Öffentlichkeit zugänglich gemacht wurde. Es ist eine grafische Sprache, die Notationselemente und deren Anwendung beschreibt. Sie besteht aus Diagrammen mit unterschiedlichen Zwecken. UML selbst bringt keine Anleitung zur Vorgehensweise der Modellierung mit sich, sondern dient lediglich als Grundlage dafür.⁵⁹

UML nutzt für die Modellierung den objektorientierten Ansatz, d.h., es wird angenommen, dass jede reale Komponente eines Originals ein Objekt ist. Diese werden einmalig genauer spezifiziert und im Modell mehrfach instanziiert, d.h., z.B. jeder physikalische Motor einer Anlage ist eine Instanz des Motorobjekts der Modellierungssprache.

Durch die Diagramme der UML lassen sich Systeme übersichtlich darstellen. Vierzehn solcher Diagramme existieren:⁶⁰

- Klassendiagramm
- Kompositionsstrukturdiagramm
- Komponentendiagramm
- Verteilungsdiagramm
- Objektdiagramm
- Paketdiagramm
- Profildiagramm
- Aktivitätsdiagramm
- Use-Case-Diagramm
- Interaktionsübersichtsdiagramm
- Kommunikationsdiagramm
- Sequenzdiagramm
- Zeitverlaufdiagramm
- Zustandsdiagramm

Je nach Bedarf werden diese Diagramme im Modellierungsprozess angewendet.

3.3.1.3 System Modeling Language – SysML

SysML basiert auf der zuvor beschriebenen UML, jedoch werden einige Aspekte daraus ausgeblendet und andere wiederum hinzugefügt. Diese Modellierungssprache ist ebenfalls standardisiert durch die *Object Management Group*, fokussiert sich jedoch auf die Modellierung von technischen Systemen. Dazu werden Teile des UML Metamodells nicht verwendet, die sich sehr spezifisch mit der Softwareentwicklung auseinandersetzen. *SysML* ist ebenso eine grafische Sprache, die auf die Erstellung von Diagrammen zur Visu-

⁵⁹ Vgl. Gruhn/Pieper/Röttgers (2006), S. 73.

⁶⁰ Aufzählung in Anlehnung an Stobitzer (o.J.), Online-Quelle [06.09.2019]

alisierung der Modelle setzt. Der Grundgedanke ist, Systemelemente und deren Zusammenhänge so darzustellen, dass Technikerinnen und Techniker unterschiedlicher Fachdisziplinen das Modell verstehen und damit arbeiten können.⁶¹

3.4 Das Systems Engineering

Wie bereits erwähnt, ist die Planung einer verfahrenstechnischen Anlage eine sehr komplexe Angelegenheit, bei der viele einzelne Fachdisziplinen gefordert sind. Dabei ist sehr oft branchenübergreifendes Wissen im Engineeringprozess gefragt.

Systems Engineering ist ein interdisziplinärer Ansatz für die Planung, Verwendung und Stilllegung von Systemen mithilfe von wissenschaftlichen, technischen und Management-Methoden. Unter einem System wird dabei ein Ensemble von Elementen verstanden, die gemeinsam etwas darstellen, das ein einzelner Teil nicht kann. Weitere wichtige Aspekte sind die Eigenschaften der Elemente und deren Interaktion zueinander sowie zur Umwelt. Es kann sich dabei um ein bereits bestehendes, physisches oder ein konzeptuelles System handeln.⁶²

Es wird versucht, die Komplexität eines Systems zu verringern und transparent darzustellen. Auf diesem Weg werden die Zusammenhänge zwischen Teilsystemen erkennbar. Lösungen können somit einfacher für die Einzelkomponenten generiert werden. Die Vorgehensweise ist, dass zuerst die Anforderungen analysiert werden und ein Anforderungsmodell erstellt wird, dabei handelt es sich um den sogenannten *Problemraum*. Anschließend werden die Funktionen des Systems ausgearbeitet, wodurch das funktionale System modelliert wird. Technische Lösungen zu den Funktionen sind in diesem Schritt jedoch noch nicht gefordert. Abschließend entsteht durch die Auswahl der Einzelkomponenten die Systemarchitektur.⁶³

Systems Engineering besteht grundsätzlich aus drei Hauptaufgaben:⁶⁴

- Der Planung, Organisation, Kontrolle und Leitung der technischen Entwicklung,
- der Definition der technischen Anforderungen aus Stakeholder-Requirements und Erstellen der Systemarchitektur durch Auswahl der Systemkomponenten, inklusive der Zuweisung von Anforderungen zu den Komponenten und
- dem Integrieren der Komponenten in das System und deren Verifikation

Aus dieser Aufzählung wird deutlich, dass diese Engineering-Disziplin nicht direkt mit der Ausarbeitung der technischen Lösungen konfrontiert ist, sondern mit der Organisation des Gesamtsystems und dem Koordinieren von einzelnen Fachdisziplinen, die stellvertretend in Abbildung 8 dargestellt sind.

⁶¹ Vgl. Alt (2012), S. 30-32.

⁶² Vgl. INCOSE (o.J.), Online-Quelle [28.08.2019]

⁶³ Vgl. Winzer (2016), S. 2-4.

⁶⁴ Vgl. Martin (1997), S. 3.

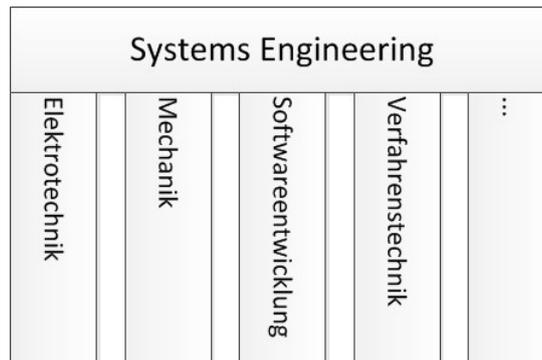


Abbildung 8: Engineering-Disziplinen im Zusammenspiel mit Systems Engineering, Quelle: in Anlehnung an Winzer (2016), S. 6.

3.4.1 Grundprinzipien des systemischen Denkens

Beim systemischen Denken werden, wie bereits angeschnitten, komplexe Aufgabenstellungen in Einzelsysteme unterteilt. Daraus kann ein Systemabbild, oder auch Modell, erstellt werden, das von allen Fachdisziplinen verstanden und für deren technische Ausarbeitungen genutzt wird. In der Literatur finden sich einige solcher Grundprinzipien:⁶⁵

- Grundprinzip des Denkens in Systemen
- Grundprinzip vom Ganzen zum Detail
- Grundprinzip der wiederkehrenden Reflexion
- Grundprinzip der Strukturierung
- Grundprinzip vom Abstrakten zum Konkreten
- Grundprinzip der minimalen Modelle
- Grundprinzip der Verständlichkeit
- Grundprinzip der Anwendung mehrerer Sichten
- Grundprinzip der Neutralität
- Grundprinzip der Mehrfachverwendbarkeit
- Grundprinzip der Standardisierung
- Grundprinzip der Informationskapselung
- Grundprinzip des diskursiven Vorgehens
- Grundprinzip des Denkens in Alternativen
- Grundprinzip des Modalitätenwechsels
- Grundprinzip der Problemzerlegung
- Grundprinzip der Minimierung von Schnittstellen

Im Folgenden wird auf vier ausgewählte Prinzipien kurz eingegangen.

3.4.1.1 Das Grundprinzip der Strukturierung

Dabei werden komplexe Gesamtsysteme in Hierarchien, Gruppen oder Module eingeteilt. Damit entsteht eine bessere Übersicht über das System. Durch Hierarchisierung werden Stufen eingebracht, die eine

⁶⁵ Aufzählung nach Winzer (2016), S. 20.

Rangordnung darstellen. Module und Gruppen fassen verknüpfte Bereiche zusammen, wie z.B. Sicherheitsmodul, Steuerungsmodul, Antriebsmodul oder Pressenmodul. Die Anzahl und Größe der Module ist vom Team und vom Lösungsweg abhängig.⁶⁶

3.4.1.2 Das Grundprinzip vom Abstrakten zum Konkreten

Dieses Prinzip findet meist Anwendung bei Neuentwicklungen. Dabei wird das Abstrakte, z.B. eine Papier- und Zellstofffabrik, untersucht und die Grundfunktionen werden ermittelt. In diesem Beispiel ist das u.a. die Zellstoff-, die Papier- und die Dampfproduktion sowie die Wärmeauskopplung und weiteres. Anschließend wird untersucht, welche Komponenten oder Teilsysteme notwendig sind, um die Funktionen zu erfüllen und wie diese in der Fabrik kombiniert werden müssen.⁶⁷

3.4.1.3 Das Grundprinzip der Standardisierung

Ziel dieses Prinzips ist es, die Teilsysteme so zu gestalten, dass sie unter anderen Einsatzbedingungen auch verwendet werden können, also mehrfach einsetzbar sind. Z.B. der elektrische Antrieb einer Pumpe kann in der Papier- sowie in der Zellstoffproduktion eingesetzt werden oder aber auch in einem anderen Kontext, wie bei der Wärmeverteilung in einem Kraftwerk. Es können in diesem Sinne auch Dokumente und Vorgänge standardisiert werden, wie z.B. Rechnungslegung in der Auftragsabrechnung eines Unternehmens.⁶⁸

3.4.1.4 Das Grundprinzip der Problemzerlegung

Bei diesem durchaus häufig verwendeten Prinzip werden umfangreiche, komplexe Problemstellungen, wie z.B. die Planung einer kompletten Papier- und Zellstofffabrik, in handierbare kleinere Probleme zerlegt. Voraussetzung ist, dass die Lösung des Teilproblems auch zur Lösung des Hauptproblems beiträgt, wovon nicht notwendigerweise auszugehen ist.⁶⁹

3.4.2 Vorgehensweisen im Systems Engineering

Es haben sich bereits viele Konzepte zur Durchführung des Systems Engineering entwickelt. Dabei gibt es jene, die einen universelleren Zugang haben, um fachdisziplinübergreifend zu sein. Andere Ansätze wurden entwickelt, die auf die spezifischen Anforderungen der anwendenden Fachrichtung ausgerichtet sind. Im Folgenden werden übergreifende sowie zugeschnittene Konzepte vorgestellt.

3.4.2.1 Der SIMILAR-Prozess

Dieser Prozess wurde erstmals 1998 von Bahill und Giessing erwähnt. Durch die Untersuchung bereits vorhandener Prozesse im Systems Engineering kamen sie zum Schluss, dass einige Schritte immer wieder in den Vorgehensweisen vorkommen.⁷⁰

⁶⁶ Vgl. Winzer (2016), S. 17.

⁶⁷ Vgl. Winzer (2016), S. 17.

⁶⁸ Vgl. Winzer (2016), S. 18.

⁶⁹ Vgl. Winzer (2016), S. 19.

⁷⁰ Vgl. Bahill/Giessing (1998), S. 516.

Die Bezeichnung *SIMILAR* stammt jedoch nicht vom englischen *gleich, ähnlich* ab, sondern ist ein Akronym aus folgenden Schrittbezeichnungen, die in Abbildung 9 grafisch dargestellt sind:

- **S**tate the problem
- **I**nvestigate alternatives
- **M**odel the system
- **I**ntegrate
- **L**aunch the system
- **A**ssess performance
- **R**e-evaluate

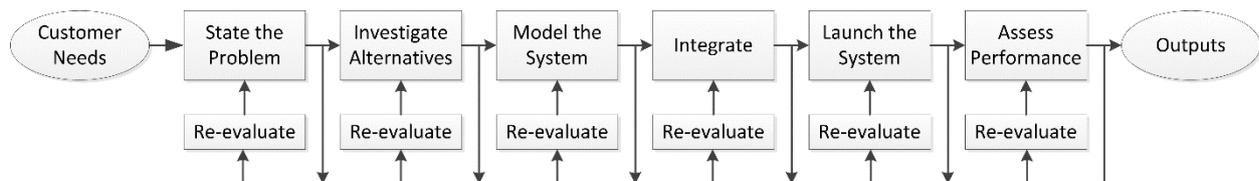


Abbildung 9: Grafische Darstellung des SIMILAR-Prozesses, Quelle: In Anlehnung an Bahill/Gissing (1998), S. 518.

State the Problem: Zu Beginn muss die Aufgabenstellung (z.B. Planung einer Anlage oder Lösung eines aufgetretenen Problems), die es zu lösen gilt, beschrieben werden. Die Beschreibung bezieht sich jedoch auf das übergeordnete Thema. Inhaltlich geht es darum, was getan werden muss, und nicht, wie. Wichtig ist es, Stakeholder in diesen Prozess einzubinden.

Investigate Alternatives: In diesem Schritt werden unterschiedliche Designvarianten auf Basis von z.B. Leistungsdaten und Kosten untersucht. Da selten eine Alternative alle gewünschten Vergleichsparameter bestens erfüllt, gilt: je mehr Parameter herangezogen werden, umso besser wird das ausgewählte Design den Anforderungen entsprechen. Sind neue Daten zur Aufgabenstellung oder zu den Designvarianten vorhanden, sollte dieser Schritt nochmals durchlaufen werden.

Model the System: Oft werden mehrere alternative Modelle entwickelt, wobei schlussendlich nur das bevorzugte Design über den vollständigen Zyklus weiterverfolgt wird. Die Ausführung der Modelle zeigt, ob Anforderungen erfüllt und Engpässe oder kritische Aufgaben vorhanden sind.

Integrate: Modelle können große Ausmaße annehmen, die im Regelfall durch Subsysteme verringert werden. Bei einem sehr komplexen System ist es ratsam, mehrere kleinere Systeme zu etablieren, die die Komplexität übersichtlicher machen. Dadurch entstehen jedoch viele Schnittstellen. In diesem Schritt werden ebendiese definiert, damit ein möglichst reibungsfreies Zusammenspiel möglich ist.

Launch the System: Darunter versteht man, dass das System *Output* generiert, d.h., z.B. ein Produkt wird hergestellt, Bau- sowie Montageunterlagen werden erzeugt oder ein Businessplan wird in Aktivitäten umgesetzt. In dieser Phase soll das Modell vollziehen, wozu es erdacht wurde.

Assess Performance: Es gibt verschiedene Parameter, die bei Modelloutputs geprüft werden können. Darunter technische Messdaten, die Güte bezogen auf andere Designvarianten oder Metriken bei Managementprozessen. Je mehr messbare Daten vorhanden sind, umso besser kann die Erfüllung der Anforderungen nachgewiesen werden.

Re-evaluate: Bei der Evaluierung werden Outputs genau untersucht, um daraus Schlüsse zu ziehen, welche Änderungen an den Inputs gemacht werden können, um diese Outputs wiederum zu verbessern. Als Vergleich kann eine Regelschleife eines Reglers herangezogen werden. Es zählt zu den wichtigsten dieser sieben Schritte, da dadurch stete Verbesserung in das System einfließt. Die Ergebnisse aller Schritte sollten mehrmals über die Prozesslaufzeit evaluiert werden.⁷¹

3.4.2.2 Das V-Modell als Makrozyklus

Die VDI 2206 beschreibt mehrere Vorgehensmodelle zur Entwicklung von mechatronischen Komponenten. Eines davon ist das V-Modell, das ursprünglich aus der Softwareentwicklung stammt und angepasst wurde. Es beinhaltet einige Prozessschritte, die nacheinander durchlaufen werden, wobei in der Praxis die zeitliche Abfolge von der vorgeschlagenen abweichen kann. Abbildung 10 zeigt den Aufbau der einzelnen Schritte, die nacheinander bzw. zeitgleich durchlaufen werden.⁷²

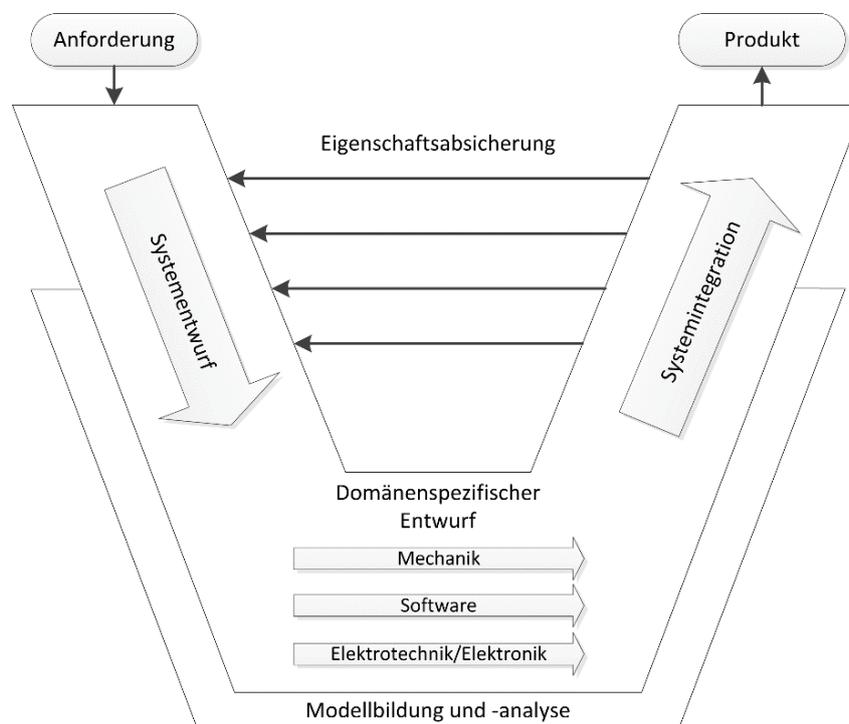


Abbildung 10: Darstellung des V-Modells, Quelle: In Anlehnung an VDI 2206 (2004), S. 29.

Zu Beginn stehen die *Anforderungen*, deren präzise Darlegung die Grundlage der Aufgabenstellung festlegt. Sie dienen als Maßstab für das entstehende Produkt. Das Ziel des *Systementwurfs* ist, ein domänenübergreifendes Lösungskonzept zu erstellen. Dieses beinhaltet physikalische und logische Wirkungsweisen des Produkts. Zur Erleichterung der Abarbeitung wird das Gesamtsystem in Teilsysteme zerlegt, denen geeignete Lösungsweisen zugeordnet werden. Die Erfüllung der Einzelfunktionen muss im Gesamtkontext geprüft werden. Beim *domänenspezifischen Entwurf* werden die gemeinsam erarbeiteten Ansätze separat ausgearbeitet. In diesem speziellen Vorgehensmodell handelt es sich um die Fachabteilungen Maschinenbau, Elektrotechnik/Elektronik und Informationstechnik. Anschließend werden bei der *Systemintegration*

⁷¹ Vgl. Bahill/Gissing (1998), S. 516-517.

⁷² Vgl. VDI 2206 (2004), S. 26.

die spezifischen Ergebnisse zusammengeführt und deren Zusammenwirken geprüft. Anhand des zu Beginn dargelegten Lösungskonzepts werden die Ergebnisse überprüft. Es handelt sich dabei um die *Eigenchaftsabsicherung*, womit sicherzustellen ist, dass das Ergebnis mit dem Entwurf übereinstimmt. Über den gesamten Prozess werden die Systemeigenschaften durch rechnergestützte Systeme untersucht und abgebildet. Im V-Modell wird dieser Vorgang als *Modellbildung und -analyse* bezeichnet. Am Ende entsteht das *Produkt*, das das Ergebnis eines Makrozyklus darstellt. Ein aufwendiges mechatronisches Erzeugnis durchläuft diesen Zyklus einige Male, daher wird dieser Prozess als Makrozyklus bezeichnet.⁷³

3.4.2.3 Erweiterung zum W-Modell

Das W-Modell ist eine Erweiterung des V-Modells, da es bei der Integrationsphase oftmals zu Problemen kommt. Die Einzelsysteme funktionieren unter geschlossenen Bedingungen sehr gut, jedoch entstehen nach der Zusammenführung Ungereimtheiten mit anderen Komponenten. Dem liegt zugrunde, dass in der Entwicklungsphase die Abstimmung von Schnittstellen und der Kommunikation zwischen den Fachdisziplinen zu wenig Aufmerksamkeit geschenkt wird. In der Integrationsphase müssen daher Rückläufe gestartet und Entwicklungen erneut durchgeführt werden, was einen zeitlichen Mehraufwand bedeutet. Der Unterschied zwischen den beiden Modellen liegt in jener Phase, in der die domänenspezifischen Entwürfe erarbeitet werden. Die Entwicklung dieser Komponenten findet, im Gegensatz zum V-Modell, in einem gemeinsamen Modell aller Disziplinen statt, mittig in Abbildung 11 dargestellt. Daraus ergibt sich ein dauerhafter Abgleich der Schnittstellen, was zu einer geringeren Fehlerhäufigkeit aufgrund von falscher Datenbasis führt. Entscheidend dafür ist ein Datenmanagementsystem, das die Datensätze der Fachrichtungen analysiert und die Kompatibilität einzelner Subsysteme überprüft. Überdies kann ein solches Managementsystem zu jedem Zeitpunkt die aktuelle Entwicklung mit den Anforderungen vergleichen und auf geänderte Anforderungen reagieren.⁷⁴

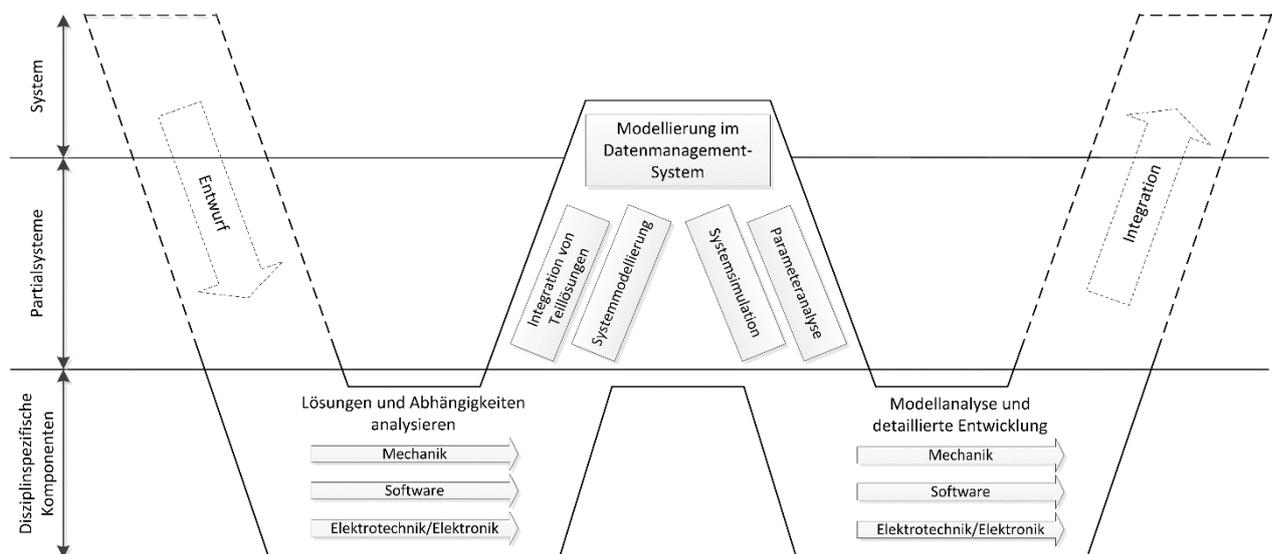


Abbildung 11: Die Erweiterung des V-Modells ergibt das W-Modell, Quelle: In Anlehnung an Anderl/Nattermann/Rollmann (o.J.), Online-Quelle [31.08.2019]

⁷³ Vgl. VDI 2206 (2004), S. 29-31.

⁷⁴ Vgl. Anderl/Nattermann/Rollmann (o.J.), Online-Quelle [31.08.2019]

3.4.2.4 Münchner Vorgehensmodell

Dieses Vorgehensmodell wurde durch Analyse bestehender Modelle entwickelt und soll drei Zwecke erfüllen: In der Planung von Entwicklungsprozessen soll es als Unterstützung dienen. Darüber hinaus soll es in Problemlösungsprozessen als Orientierungshilfe eingesetzt werden und die Vorgehensweise analysieren und reflektieren.

Die Durchführung wird von drei Hauptschritten begleitet:

- Das Ziel bzw. Problem klären
- Lösungsalternativen finden
- Entscheidungen hervorbringen

Während sich der Großteil der anderen Modelle auf lineare Abläufe mit dem Hinweis auf Iterationen und Modellanpassungen fokussiert, ist diesem Vorgehensmodell eine netzwerkartige Vorgehensweise zugrunde gelegt. Wie in Abbildung 12 nachvollzogen werden kann, können die einzelnen Arbeitsschritte individuell verbunden werden. Somit wird mit Hilfe dieses Netzwerks der reale Problemlösungsprozess mit seinem sprunghaften Verlauf besser dargestellt als in anderen Modellen. Da die Übergänge zwischen den Schritten meist nicht klar getrennt werden können, wird dies im Münchner Vorgehensmodell durch Überschneidung der dargestellten Kreise aufgezeigt. Die drei zuvor beschriebenen Hauptschritte werden in sieben Elemente unterteilt, die als Kreise dargestellt sind. Die Verbindungen dazwischen zeigen die angesprochenen Möglichkeiten der individuellen Prozessgestaltung.⁷⁵

Im ersten Element, *Ziel planen*, wird die betrachtete Situation analysiert und daraus durchzuführende Maßnahmen abgeleitet. Als Analysefaktoren werden u.a. Kunde und Wettbewerb, aber auch Ressourcenverfügbarkeit oder die Terminsituation einbezogen. Abhängig, ob es sich beim Ziel des Modells um z.B. die strategische Produkt- bzw. Prozessplanung oder ein gelungenes Meeting im operativen Prozess handelt, werden diese Faktoren unterschiedlich eingesetzt.⁷⁶

Beim nächsten Element, *Ziel analysieren*, wird der gewünschte Zielzustand geklärt und beschrieben. Dazu werden Anforderungen an das Ziel formuliert, damit am Ende ein gerechtes Produkt entstehen kann. Überdies sind die Beziehungen zwischen den Anforderungen zu berücksichtigen, um ein umfassendes Bild der Zielvorstellung zu erlangen.⁷⁷

Das Element *Problem strukturieren* dient zu Wegfindung zwischen Problem und Ziel. Hierbei werden Handlungsschwerpunkte ermittelt, die bei der Lösungssuche unterstützen. Dazu wird das System in Teilsysteme untergliedert, um die Komplexität aus dem Gesamten zu nehmen und in mehrere weniger komplexe Teile zu verlagern. Ziel ist es, ein Problemmodell zu etablieren, aus dem Lösungsansätze herausgearbeitet werden können.⁷⁸

⁷⁵ Vgl. Lindemann (2009), S. 46-48.

⁷⁶ Vgl. Lindemann (2009), S. 48-49.

⁷⁷ Vgl. Lindemann (2009), S. 48-49.

⁷⁸ Vgl. Lindemann (2009), S. 48-49.

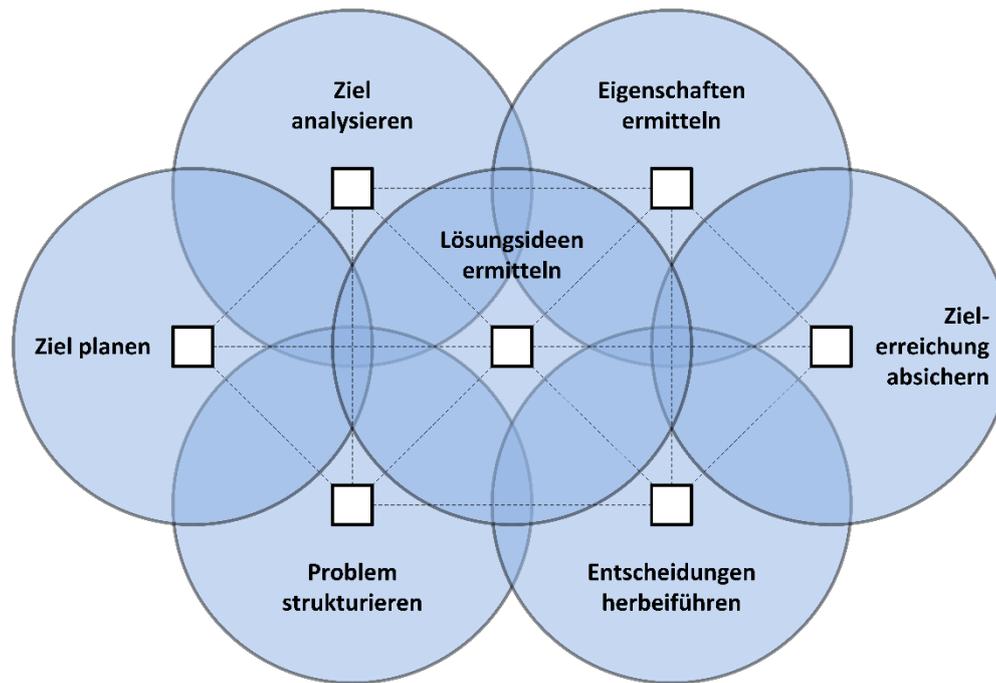


Abbildung 12: Grafische Darstellung des Münchner Vorgehensmodell, Quelle: In Anlehnung an Lindemann (2009), S. 47.

Beim Element *Lösungsideen ermitteln* werden konkrete Lösungen gesucht, darunter fallen bereits angewandte sowie neu entwickelte. Alternativen sind dabei essentiell, um verschiedene Möglichkeiten der Umsetzung aufzuzeigen. Die Anzahl an Lösungsmöglichkeiten gehört übersichtlich dargestellt. Abschließend müssen die Einzellösungen wieder in ein Gesamtkonzept vereint werden.⁷⁹

Unter *Eigenschaften ermitteln* wird die Analyse der vorhandenen Systeme verstanden, bezogen auf deren Eigenschaften. Dies wird auch als *Eigenschaftenanalyse* bezeichnet. Diese Eigenschaften werden aus den Lösungsansätzen herausgearbeitet und dienen schlussendlich zur Bewertung der Zielerreichung und zum Vergleich mit den festgelegten Anforderungen. Solche Eigenschaften sind z.B. die Realisierbarkeit bzw. Geräuschpegel, Leistung oder Energieverbrauch am Ende der Laufzeit.⁸⁰

Im Element *Entscheidungen herbeiführen* werden die erarbeiteten Lösungsideen verglichen und bewertet. Anschließend werden Entscheidungen getroffen, welche davon weiterverfolgt werden. Abhängig von der Wichtigkeit der Entscheidung hat diese mehr oder weniger Einfluss auf das Gesamtsystem, jedoch dürfen kleinere Entscheidungen nicht unterschätzt werden. Die Anzahl an solchen Festlegungen ist sehr groß und nimmt daher, ganzheitlich betrachtet, sehr viel Zeit und Aufwand in Anspruch.⁸¹

Obwohl das Element *Zielerreichung absichern* im Netzwerk am Ende angesiedelt ist, muss es bereits zu Beginn des Prozesses beachtet werden. Um Fehler in Produkten oder Planungen bereits frühzeitig zu

⁷⁹ Vgl. Lindemann (2009), S. 49-50.

⁸⁰ Vgl. Lindemann (2009), S. 49-50.

⁸¹ Vgl. Lindemann (2009), S. 49-50.

unterbinden, müssen die möglichen Risiken für Entscheidungen ermittelt und bewertet werden. Danach sind Maßnahmen einzuleiten, die diese Gefahrenstellen minimieren können.⁸²

Mit diesem Vorgehensmodell ist es möglich, gesamte Entwicklungsprozesse zu betrachten, oder aber auch nur Details daraus. Essentiell sind dabei die Probleme, die untersucht werden, um die richtigen Lösungsalternativen zu erhalten. Durch den Aufbau als Netzwerk ist der Prozess sehr flexibel und kann individuell an die Anwendung angepasst werden.⁸³

3.5 Zusammenfassung Modell und Systems Engineering

Ein Modell ist die vereinfachte Darstellung eines realen Vorbilds. Es kann sich dabei um ein bestehendes oder ein noch zu erstellendes Original handeln. Grundsätzlich können dabei zwei Modelltypen unterschieden werden, das deskriptive und das präskriptive Modell, wobei ersteres das Abbild eines Originals ist, das nicht beeinflussbar ist. Beim anderen handelt es sich um Modelle, die erst geschaffen werden und deren Eigenschaften noch festzulegen sind. Den Prozess, um zu einem Modell zu gelangen, nennt sich Modellierung. Die Ansätze, darunter Bottom-Up oder der Black-Box-Ansatz, unterscheiden sich in ihrer Herangehensweise, jedoch ist das Ziel jeweils dasselbe, nämlich ein Modell zu bilden. In Metamodellen wird beschrieben, welche Regeln bei der Modellbildung zu beachten sind und welche Zusammenhänge zwischen den Modellobjekten bestehen. Bei Modellierungssprachen werden solche Metamodelle genutzt, um die Syntax und Semantik festzulegen. Vorgestellte Sprachen zur Modellierung sind UML oder SysML, beide durch die *Object Management Group* standardisiert. Ist keine vorhandene Sprache für einen bestimmten Anwendungsfall geeignet, werden oftmals spezielle Sprachen selbst entwickelt. Dies wird als domänen-spezifische Sprache, auch DSL, verstanden. Modelle und Modellierung sind die Grundlagen, die im Systems Engineering dringend benötigt werden. Dieser interdisziplinäre Ansatz ist per se keine Fachdisziplin, die Komponenten entwickelt. Sie beschäftigt sich mit der Planung und Organisation der übergeordneten technischen Entwicklung, definiert mit Stakeholdern die Anforderungen und integriert die Ergebnisse der Disziplinen in ein gemeinsames System. Nicht minder wichtig ist dabei die Verifikation des Produkts, so dass sichergestellt wird, dass die Ergebnisse den Anforderungen entsprechen. Bei diesem Ansatz ist es wichtig, in Systemen zu denken. Dabei gibt es mehrere Prinzipien, wie vorgegangen werden kann. Darunter das Denken in Systemen, das Prinzip der Strukturierung oder das Prinzip der Standardisierung. Diese Prinzipien sind entscheidend für die Modellierung im Systems Engineering. Für den Verlauf und die Vorgehensweise gibt es einige sogenannte *Vorgehensmodelle*: Den SIMILAR-Prozess, das V-Modell und dessen Erweiterung oder das Münchner Vorgehensmodell. Als Gemeinsamkeit aller steht das Modellieren und Evaluieren im Mittelpunkt. Unterschiede bestehen in der Vorgehensweise selbst, da einige aufeinanderfolgende Prozessschritte vorsehen, während andere in der Prozessgestaltung variabel sind.

⁸² Vgl. Lindemann (2009), S. 49-50.

⁸³ Vgl. Lindemann (2009), S. 49-50.

4 MODELGETRIEBENE ARCHITEKTUR – MODEL DRIVEN ARCHITECTURE (MDA)

Model Driven Architecture, auch *MDA* genannt, ist ein standardisierter Ansatz für Softwaredesign, -entwicklung und -implementierung, der durch die *Object Management Group* herausgegeben wurde. Es ist ein Leitfaden zur Strukturierung von Softwareanforderungen in Form eines Modells. Dabei werden die funktionale und technische Ebene voneinander getrennt. Es ergeben sich zu Beginn plattformunabhängige Modelle des Systems, die durch mehrere Schichten genauer spezifiziert werden, um daraus schlussendlich Programmcode zu generieren. Die Beschreibung der Systeme wird durch UML und weitere Standards der *OMG* realisiert.⁸⁴

4.1 Viewpoints und Schichten/Modelle der MDA

Wie zuvor angedeutet, durchläuft die modellgetriebene Softwareentwicklung mehrere Schichten. Diese können auch als Abstraktionsschichten bezeichnet werden, da jede einen unterschiedlichen Detaillierungsgrad und sogenannten *Viewpoint* besitzt. Die MDA sieht drei dieser *Viewpoints* vor, den *computation independent*, *plattform independent* und *plattform specific*. Die erste Sichtweise ist unabhängig von jeglichem rechnergestützten System. Es fokussiert lediglich auf den Inhalt und die Anforderungen des Systems. Die plattformunabhängige Sichtweise beschreibt die Struktur und Funktionalität des Systems. Die dritte Sichtweise erweitert die plattformunabhängige mit Spezifika zu den ausführenden Plattformen.⁸⁵

Die *OMG* gibt drei Modelle in der Architektur der MDA an. Das *Computation Independent Model (CIM)*, das *Platform Independent Model (PIM)* und das *Platform Specific Model (PSM)*. Wie beschrieben, werden diese auch als Schichten bezeichnet.

Das *CIM* ist ein Modell des gesamten Systems ohne technische Realisierungsdetails und unabhängig von der Implementierung. Es ist im Fachvokabular der Projektdomäne verfasst und beinhaltet die Anforderungen sowie Schnittstellen des Systems und dessen Umwelt. Im plattformunabhängigen Modell wird die Struktur des Systems festgelegt und die Funktionen, die zur Durchführung der Systemaufgaben notwendig sind. In der darauffolgenden Schicht, dem *PSM*, werden dem *PIM* plattformspezifische Informationen beigegeben. Wenn dieses System genügend Details besitzt, kann daraus schließlich Programmcode generiert werden.⁸⁶

Zusätzlich kommt das *Platform Model (PM)* zum Einsatz, das technische Umsetzungskonzepte und Vorgehensweisen einer speziellen Plattform beinhaltet. Darüber hinaus sind typische Elemente enthalten, die die Verwendung der Plattform beschreiben. Diese werden bei der Erstellung des *PSM* herangezogen.⁸⁷

⁸⁴ Vgl. Object Management Group (o.J.), Online-Quelle [07.09.2019]

⁸⁵ Vgl. Truyen (2006), Online-Quelle [07.09.2019], S. 4-5.

⁸⁶ Vgl. Gruhn/Pieper/Röttgers (2006), S. 27.

⁸⁷ Vgl. Den Haan (2008), Online-Quelle [07.09.2019]

Vor allem im Bereich des PSM besteht die Möglichkeit, dass mehrere Modelle nebeneinander existieren. Das kommt dann vor, wenn Code in unterschiedlichen Programmiersprachen umgesetzt werden soll. Beispiele für parallele PSMs sind *J2EE* und *.NET*, aus denen Code in *Java* und *C#* entstehen kann.⁸⁸

4.2 Plattformen im MDA-Kontext

Die Plattform ist jene Umgebung, in dem das System schlussendlich realisiert wird. Bei der Umsetzung in eine technologische Plattform wird das System darin implementiert und ausgeführt. Daraus entsteht folglich die Anwendung, die die im Modell beschriebenen Funktionen umsetzt und ausführt. Beispiele für typische Softwareplattformen sind *Microsoft .NET*, *J2EE*, *AJAX* oder *Eclipse RCP*. Diese Anwendungen variieren zwischen Plattformen zur Programmerstellung, für interaktive Web-Applikationen und Entwicklungsplattformen.⁸⁹

4.3 Modellierungssprache der MDA

Die in Kapitel 3.3.1 beschriebenen Modellierungssprachen finden sich verständlicherweise auch im MDA-Ansatz wieder. Bekannte und fachlich anerkannte Sprachen sind hilfreich bei der Zusammenarbeit von mehreren Personen, da so ein gewisses Verständnis des Modells vorausgesetzt werden kann. Bei der standardisierten, modellgetriebenen Architektur wird UML verwendet. Dies ist eine semi-formale Modellierungssprache, d.h., damit erstellte Modelle können auf semantische Richtigkeit geprüft werden. Daher kann in weiterer Folge davon ausgegangen werden, dass die Überführung in ein anderes Modell aufgrund der sprachlichen Richtigkeit des Modells möglich ist.⁹⁰

4.4 Transformationen zwischen Modellen

In Abbildung 13 wird gezeigt, dass zwischen den einzelnen Modellen Transformationen stattfinden. Diese können als Verfeinerung des Modells betrachtet werden. Der Übergang von PIM zu PSM fällt unter die *Modell-zu-Modell-Transformation*. Dabei wird ein Modell in ein anderes überführt und um Informationen bzw. technische Detailierung erweitert. Dem steht die *Modell-zu-Text-Transformation* gegenüber, bei der das Modell in *Text*, also Programmcode, umgewandelt wird. Bei diesem Vorgang kann eine Verfeinerung des Systems geschehen, dies ist jedoch nicht zwingend notwendig. Beide Varianten können manuell oder teil- bzw. vollautomatisiert umgesetzt werden.⁹¹

⁸⁸ Vgl. Kempa/Mann (2005), Online-Quelle [08.09.2019]

⁸⁹ Vgl. Object Management Group (2014), Online-Quelle [08.09.2019], S. 9-10.

⁹⁰ Vgl. Object Management Group (2014), Online-Quelle [08.09.2019], S. 6.

⁹¹ Vgl. Gruhn/Pieper/Röttgers (2006), S. 149-150.

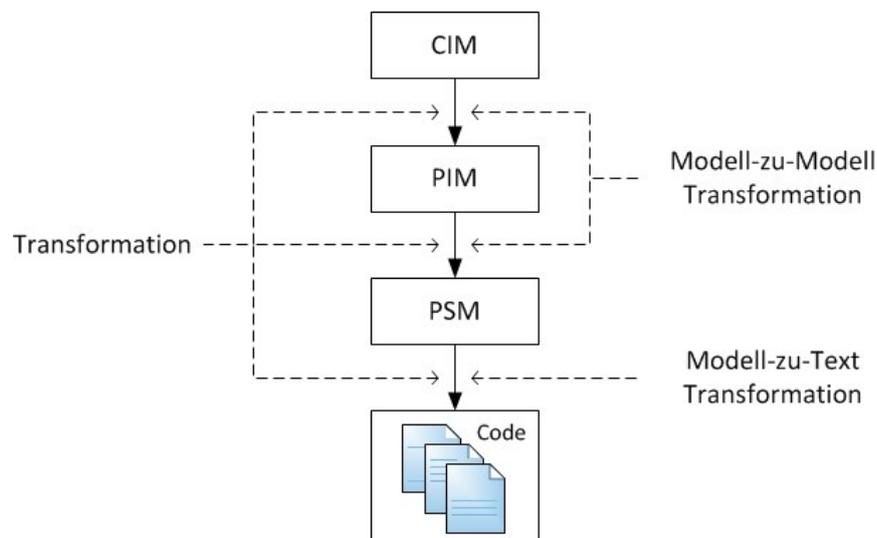


Abbildung 13: Aufbau der MDA-Modelle und Transformationen, Quelle: Eigene Darstellung

Zu klären gilt nun, wie eine Transformation per se funktioniert. Dazu können zwei Modelle betrachtet werden, die hierarchisch untereinander angeordnet sind. Der Inhalt und Aufbau des oberen sowie unteren ist bekannt. Das Mapping oder auch Transformationsregeln genannt, beschreibt, wie Elemente des Quellmodells auf Elementen des Zielmodells abgebildet werden. Im Mapping sind diese Regeln festgelegt.⁹²

4.5 Ziel der MDA

Grundgedanke der modellgetriebenen Architektur ist es, plattformunabhängige Modelle zu erstellen. Diese sollen weiter spezifiziert werden, damit plattformspezifische Modelle daraus entstehen. Womit wiederum Code generiert werden soll, der von der ausgewählten Plattform abhängt. Der große Vorteil dieses Ansatzes entsteht dadurch, dass das unabhängige Modell für die Entstehung des plattformabhängigen Codes nicht angepasst werden muss. Das bedeutet, dass eine agile Entwicklung möglich ist und Änderungen im Grundkonzept nicht mehrfach händisch in den Plattformen nachgezogen werden müssen.⁹³

Weitere Vorteile der MDA sind, dass durch das PIM stets ein Modell zur Verfügung steht, das disziplinenübergreifend als Kommunikationsbasis gelten kann. Auch wenn andere Fachabteilungen der UML nicht mächtig sind, ergibt sich durch die grafische Darstellung doch eine gute Gesprächsbasis. Ein weiterer Faktor ist, dass die Abhängigkeit, an eine Zielplattform gebunden zu sein, verringert wird. Dazu muss nicht das komplette System verworfen, sondern nur um ein PSM erweitert werden. Ein weiterer Vorteil besteht darin, dass bei Entwicklungen mit MDA relativ schnell Ergebnisse generiert werden. Voraussetzung dafür sind ein funktionierender Generator sowie ausgereifte Transformationsregeln. Dem stehen natürlich auch Nachteile gegenüber, wie z.B., dass MDA-Projekte vollständig von der Qualität des zugrundeliegenden Modells abhängig sind. Ist das Modell (zumeist das PIM) gut ausspezifiziert und getestet, steht dem Projekterfolg wenig im Weg. Bei schlechter Qualitätssicherung und nachlässiger Datenpflege besteht die Gefahr, dass der Code schlussendlich nicht das erfüllt, zu dem er eigentlich vorgesehen wurde. Durch die Möglichkeit

⁹² Vgl. Gruhn/Pieper/Röttgers (2006), S. 155.

⁹³ Vgl. Andresen (2003), S. 77.

der automatischen Codegenerierung entsteht ein Problem, das sich bei der Fehlersuche und bei Programmverbesserungen negativ auswirken kann. Da die Entwicklerinnen und Entwickler sich nicht mit der Erstellung des Codes befassen müssen, wird dieser als etwas Fremdes angesehen. Kommt es zu Problemen darin, erschwert sich die Fehlersuche oft, da auch die Vorlagen zur Erstellung des Codes meist aus fremder Feder stammen. Daher ist es sinnvoll, die Teammitglieder auch bei der Erstellung der Vorlagen einzubinden.⁹⁴

4.6 Zusammenfassung MDA

Der modellgetriebene Softwareentwicklungsansatz beinhaltet einige komfortable Eigenschaften. Darunter die Trennung von technischer und funktionaler Ausarbeitung. Jede Disziplin kann sich daher auf ihre fachliche Kompetenz konzentrieren. Durch die Schichten (CIM, PIM, PSM), die im Ablauf eines MDA-Projekts durchlaufen werden, können Systeme in einer guten Art und Weise abstrahiert werden. Mit Hilfe von Transformationen zwischen den Modellen werden diese manuell oder automatisiert ineinander übergeführt. Finales Ziel ist jedoch, den Programmcode für Anwendungen, so weit wie möglich, aus dem PSM zu generieren. Je besser dieser Schritt funktioniert, umso variabler können die plattformunabhängigen Modelle gestaltet werden.

Zu beachten sind die Qualitätssicherung der Modelle und dass Entwicklerinnen und Entwickler nicht den Bezug zum generierten Code verlieren. Anderenfalls wird Fehlersuche und Problemlösung zu einer schweren Aufgabe.

⁹⁴ Vgl. Breu/Matzner/Nickl (2005), S. 43-46.

5 DIE PRAKTISCHE UMSETZUNGSIDEE

Im ersten Teil dieser Arbeit wurden zu Beginn die Themen beleuchtet, die sich mit den Vorgängen bzw. Arbeitsweisen sowie auch den Unterlagen im Planungsprozess einer Anlagenplanung beschäftigen. Dabei konnte herausgearbeitet werden, dass einige Dokumente entscheidender für die ausführende Planung sind als andere. Darüber hinaus müssen Änderungen an den darin enthaltenen Basisdaten möglichst genau erfasst und gekennzeichnet werden. Im Anschluss daran wurden Modelle per se und systemische Engineering-Vorgehensweisen untersucht. Im Zuge dessen wurden unterschiedliche Modelltypen und Modellierungspraktiken beschrieben. Basis dafür sind Modellierungssprachen, die unterschiedlich ausgeprägt sind und daher für verschiedene Anwendungen zum Einsatz kommen. Im Systems Engineering vereinen sich mehrere Fachbereiche, die gemeinsam an der Ausarbeitung eines Produkts arbeiten. Der Grundgedanke dabei ist, dass alle Disziplinen an einem gemeinsamen Modell arbeiten, um die Anzahl an Schnittstellen zu minimieren. Dazu werden diverse Prinzipien angewendet, die sich in bereits veröffentlichten und erprobten Vorgehensweisen wiederfinden. Diesem Ansatz verwandt ist MDA, die modellgetriebene Architektur. Vorrangig verwendet in der Entwicklung von Software, baut es auf Modelle in Schichten auf, die ineinander übergeführt werden. Finales Ziel ist die automatische Generierung von Programmcode in unterschiedlichen Umgebungen, abhängig von z.B. Kundenanforderungen.

In dem nun folgenden Teil wird eine praktische Umsetzung angestrebt, in der die Idee der modellgetriebenen Architektur aufgegriffen wird. Anwendung soll diese in der Planung der Elektro- und EMSR-technischen Planung finden, genauer noch im Bereich der Detailplanung. Die zugrundeliegende Idee dazu stammt daher, dass Dienstleistungsunternehmen in diesem Bereich stark an die Engineeringplattformen gebunden sind, die durch Kunden vorgegeben werden. Aus diesem Grund müssen Mitarbeiterinnen und Mitarbeiter gut im Umgang mit mehreren Engineering-Tools geschult sein. Bei mehreren Kunden mit unterschiedlichen Plattformen entstehen somit hoher Schulungsaufwand sowie Problemstellungen mit den Umgebungen. Die dafür benötigte Zeit muss im äußersten Fall von der technischen Abwicklung abgezogen werden, die jedoch im Vordergrund stehen soll. Überdies entstehen, im speziellen bei der händischen Vervielfältigung von Plänen, Fehler wie z.B. Tippfehler oder es werden falsche Vorlagen herangezogen. Dienstleistungsunternehmen werden zu großen Teilen über die Richtigkeit und Qualität der gelieferten Dokumentation beurteilt. Zu bedenken ist, dass ein Schreibfehler andere Auswirkungen hat als ein technischer Fehler. Trotzdem wird durch eine solche kleinere Abweichung die Gesamtqualität unnötigerweise verringert.

In dieser Umsetzung soll auf Basis von üblichen Projektunterlagen für die EMSR-Detailplanung ein teilautomatisierter Workflow entstehen, mit dem aus einem unabhängigen Modell Pläne in zwei unterschiedlichen Planungsumgebungen entstehen können. Um Beweggründe zur Auswahl des gewählten Workflows zu untermauern, wird folgend die Firma *metior GmbH*, der Dienstgeber des Autors, vorgestellt.

5.1 Unternehmensvorstellung metior GmbH

Die *metior Industrieanlagen Planungs- und Beratungs-GmbH* beschäftigt sich hauptsächlich mit der Planung von elektrotechnischen und leittechnischen Anlagenteilen in Produktionsanlagen. Dabei werden umfassende Projekte von der Automatisierungstechnik-Basisplanung über die anschließende Detailplanung und schlussendlich der Anlageninbetriebnahme abgewickelt. Durch über 30 erfahrene Mitarbeiter und Mitarbeiterinnen werden Konzepte für Budgetentscheidungen gleichermaßen ausgearbeitet wie Software für

die prozessleittechnische Steuerung von Anlagen. Durch sehr starke Kundenorientierung werden die Engineering-Tools der Prozessindustrie wie *COMOS*, *EPLAN Electric P8*, *Engineering Base*, *ELCAD* oder *AutoCAD* als Planungsumgebung angeboten. Ebenso im Bereich SPS/PLS werden abhängig vom Kundenwunsch u.a. Systeme der Hersteller *Siemens*, *Emerson*, *Beckhoff*, *Bernecker & Rainer* oder *CopaData* eingesetzt. Abhängig von der Aufgabenstellung sind die Mitarbeiter und Mitarbeiterinnen beim Kunden als Consultants tätig oder wickeln Projekte als Gesamtpakete ab.



Abbildung 14: Firmenlogo metior GmbH inklusive Slogan, Quelle: metior GmbH (o.J.), Online-Quelle [29.10.2019]

Als Dienstleistungsunternehmen profitiert metior davon, Projekte innerhalb kurzer Durchlaufzeiten, mit möglichst minimalem Zeitaufwand und hoher Qualität, abzuwickeln. Daher wird stets stark an Verbesserungen von Arbeitsweisen sowie Nutzen von Synergien der Entwicklungsumgebungen gearbeitet.

5.2 Workflow-Beschreibung

In diesem Unterkapitel wird der Ablauf des ausgearbeiteten Workflows grundsätzlich beschrieben, der in Abbildung 15 in Form eines Vorgehensmodells dargestellt ist. Die verwendeten Begriffe, Programme, Dateiformate und Vorgänge werden anschließend tiefergehend erläutert.

Den Startpunkt zu diesem Workflow bilden die Basisunterlagen wie Messkreis-, Stellenstellen- und Verbraucherlisten, aber auch ein Pflichtenheft sowie der Kundenstandard. Diese Daten werden analysiert und dementsprechend aufbereitet, damit sie für die Folgeschritte als Basis herangezogen werden können. Mit den Informationen zur Anlage wird im nächsten Schritt das Metamodell erstellt. Wenn der Workflow das erste Mal durchlaufen wird, ist dieser Schritt aufwendiger, da auf kein bestehendes Metamodell zurückgegriffen werden kann. Bei Folgeprojekten können Teile daraus wiederverwendet werden, somit verringert sich dieser Aufwand. Im Anschluss können mehrere Aktivitäten gleichzeitig ablaufen. Auf Basis des Metamodells und den Basisunterlagen wird das konkrete Anlagenmodell modelliert. Zeitgleich können die Typicals in den Endplattformen erstellt werden, die ebenso auf den Informationen aus den Startunterlagen basieren. Der Grund für die Auswahl, die Vorgehensweisen und Unterschiede der beiden Engineering-Tools *Engineering Base* und *EPLAN Electric P8* wird in Kapitel 8 näher erläutert. Für *EPLAN Electric P8* wird ein Plan-generator eines Drittanbieters, *CAE Expert Group GmbH*, verwendet. Darin müssen diverse Vorbereitungen durchgeführt werden, um später die Pläne in *EPLAN Electric P8* generieren zu können.

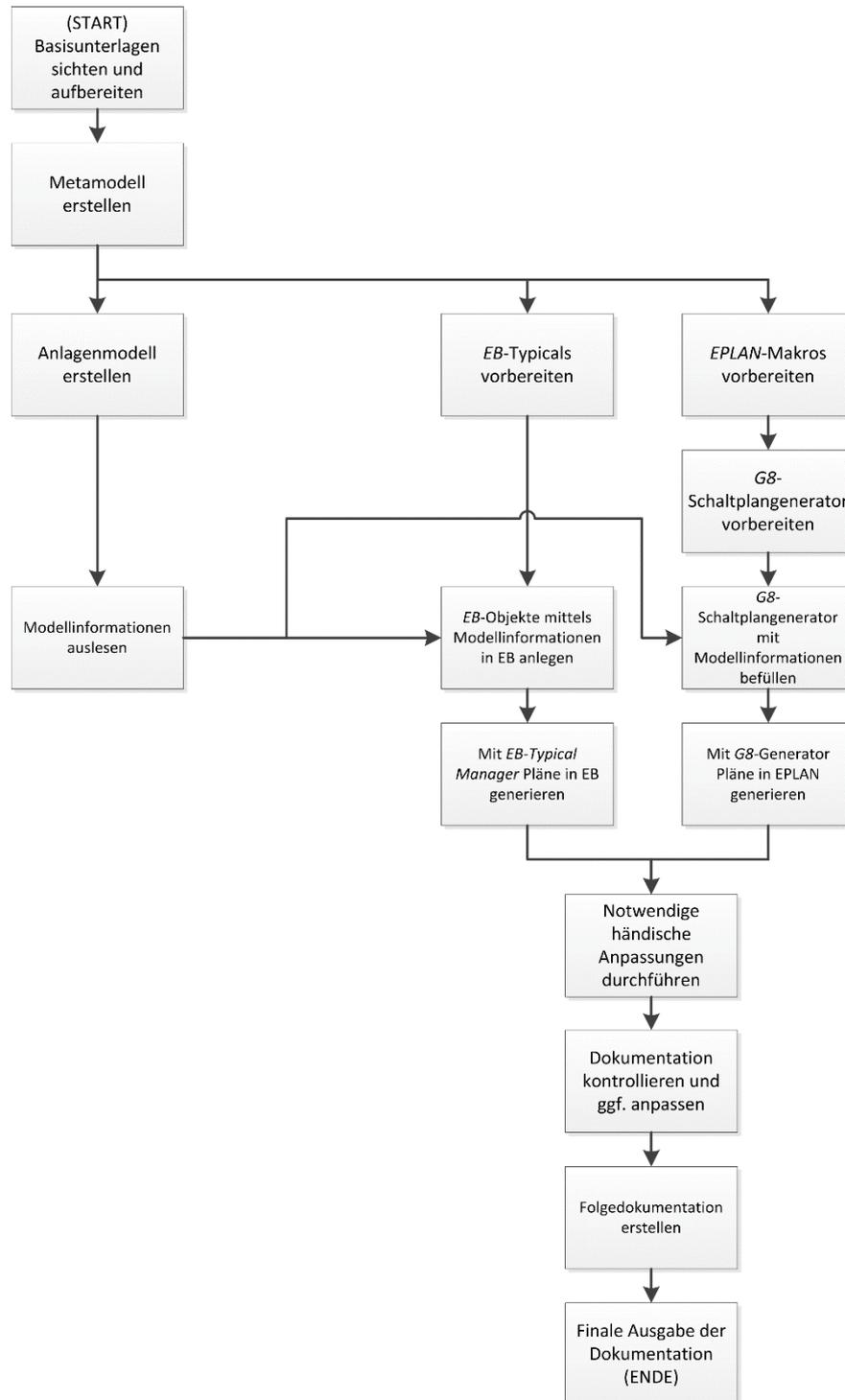


Abbildung 15: Vorgehensmodell des ausgearbeiteten Workflows, Quelle: Eigene Darstellung

Bevor es jedoch zum Erstellen der Pläne in den Planungsumgebungen kommt, werden die Informationen aus dem Anlagenmodell ausgelesen. Dies passiert per *Microsoft Excel VBA*, da einerseits *Engineering Base* Schnittstellen besitzt, um Objekte mittels tabellenorientierten Dateien einzulesen, andererseits basiert der *EPLAN Electric P8 Generator* ebenso auf *Microsoft Excel*. Sodann werden die ausgelesenen Informationen in die gewünschte Vorlage gebracht, die Datenbankobjekte in *Engineering Base* angelegt bzw. die *EPLAN Electric P8 Generator*vorlage befüllt. Mit dieser Basis können die Pläne in den Programmen generiert werden. Dazu wird in *Engineering Base* der Standard-Assistent *Typical Manager* verwendet und in

EPLAN Electric P8 der bereits erwähnte Generator mit der Bezeichnung *G8*. In beiden Tools sind danach händische Nacharbeiten, z.B. das Erstellen von nicht typisierten Plänen, notwendig. Trotz der automatischen Generierung der Dokumentation wird diese anschließend vom Techniker bzw. der Technikerin überprüft und im Anlassfall Fehler ausgebessert. Hat der Plansatz die Qualitätskontrolle durchlaufen, werden die Folgedokumente, z.B. Klemmenpläne oder Kabellisten, erstellt. Abschließend wird die Dokumentation für die geplante Verwendung weiterverarbeitet, zumeist in Form einer PDF-Datei.

5.3 Auswahl der betrachteten Engineering-Plattformen

Wie bereits in Abschnitt 5.1 erwähnt, werden durch die Firma *metior GmbH* die gängigen Planungsprogramme der Prozessindustrie bedient. Für diese Arbeit fällt die Wahl auf die beiden Programme *Engineering Base* (kurz *EB*) der Firma *Aucotec AG* und *EPLAN Electric P8* der Firma *EPLAN*. Diese Entscheidung basiert auf den folgenden Gründen:

- *Comos* ist datenbankbasierend, jedoch im Aufbau der Datenbank meist recht komplex.
- *Comos* wird stark durch die Unternehmen an die individuellen Anforderungen angepasst. Daher gestaltet sich der Zugang zu den kundenspezifischen Strukturierungen bei jedem Projekt anders.
- Bei *metior* intern gibt es lediglich eine Standard-*Comos*-Umgebung zur Schulung von Mitarbeitern und Mitarbeiterinnen.
- Die Administration von *Comos*-Systemen ist aufwendig und wird durch eigens geschulte Administratorinnen und Administratoren betreut.
- *AutoCAD* ist als Elektro-Planungstool für *metior* ein Nischenprodukt und bietet die typischen Funktionalitäten nicht in gewünschter Weise adäquat an.
- *ELCAD* wurde als Planungsprogramm bereits durch einen Nachfolger (*Engineering Base*) ersetzt. Einige *metior*-Kunden haben es jedoch noch in Verwendung oder bereiten den Umstieg vor.
- *Engineering Base* ist datenbankbasierend.
- *Engineering Base* bietet eine große Anzahl an implementierten Assistenten, die bei solchen Workflows hilfreich sind.
- *Metior* hat bereits Erfahrungen gesammelt im Umgang mit den Generator-Mechanismen in *Engineering Base*.
- *EPLAN Electric P8* ist ein sehr stark verbreitetes Planungswerkzeug für die Elektrodokumentation von Anlagen.
- Für *EPLAN Electric P8* gibt es eine große Community und Anbieter für Zusatzsoftware, z.B. der *G8*-Schaltplangenerator.
- *EPLAN Electric P8* sowie *Engineering Base* sind in der Gestaltung und dem Aufbau der Strukturen sehr flexibel einsetzbar.
- *Metior* entscheidet sich bei freier Tool-Auswahl entweder für *Engineering Base* oder *EPLAN Electric P8*, abhängig vom Anwendungsfall.

Basierend auf diesen Faktoren ist die erwähnte Entscheidung auf die beiden Programme gefallen. Eine Ausweitung auf weitere Umgebungen ist theoretisch möglich, wird in dieser Arbeit jedoch nicht tiefergehend betrachtet.

5.4 Voraussetzungen/Anforderungen des Workflows

In einem Workshop mit erfahrenen Projektmitarbeitern und Projektleitern wurden die folgenden Voraussetzungen und Anforderungen an einen solchen Workflow ermittelt.

Teilnehmer: Hr. Gruber Dominik (Junior-Lead-Engineer / metior GmbH), Hr. Tatzler Viktor (Lead-Engineer / metior GmbH)

Einstieg: Beschreibung des Ablaufes (vom Erhalt der Basisunterlagen bis zu den erstellten Plänen in der Engineering-Umgebung)

In Tabelle 2 sind die Voraussetzungen angeführt, die für eine erfolgreiche Umsetzung des Workflows notwendig sind. Daraus entsteht ein Nutzen in der Projektabwicklung, der ebenfalls in der Tabelle ersichtlich ist.

Voraussetzungen	Beschreibung	Erwarteter Nutzen
Aufbereitung der Basisunterlagen	Genaue Aufbereitung der Basisdaten als Basis für die Anlagenmodell-Erstellung	Geringere Nachbearbeitung; schnelleres Modellieren
Projektumfang	Nutzen stellt sich nur bei Großprojekten (>2000 Stellen) ein	Dokumentationsqualität; Engineering-Geschwindigkeit; Zeit- bzw. Kostenfaktor
Gleichartigkeit der Typicals	Nur bei geringer Anzahl an unterschiedlichen Planvarianten sinnvoll, da ansonsten zu viel Aufwand in Typicalerstellung	Wiederverwendbarkeit der Modelle

Tabelle 2: Ermittelte Voraussetzungen für die Verwendung des Workflows, Quelle: Eigene Darstellung

Anforderung	Beschreibung	Erwarteter Nutzen
Zeitfaktor verringern	Aufgrund der teilautomatischen Abläufe muss der Vorgang schneller als eine konventionelle Planung sein	Kosteneinsparung; kürzere Durchlaufzeiten
Fehlerhäufigkeit verringern	Durch automatisches Generieren entfallen <i>manuelle</i> Fehler	Qualitätssteigerung der Dokumentation
Schnellere Reaktion auf kurzfristige Änderungen	Wenn Änderungen an den Basisdaten entstehen (erweitert, verringert, verändert) soll schnell darauf reagiert werden können	Geringere Abhängigkeit von Änderungen
Weniger Zeitaufwand mit Engineering-Umgebung	Die Einschulung auf Engineering-Tools muss nicht bei jedem Mitarbeiter oder Mitarbeiterin im großen Ausmaß ausfallen	Höhere Konzentration auf das Engineering selbst; Geringerer Schulungsaufwand;
Unabhängigkeit von Engineering-Umgebung des Kunden	Der Fokus im Engineering liegt auf dem Modell, die Plattform spielt eine untergeordnete Rolle in der Ausarbeitung	Wiederverwendung des technischen Know-How in unterschiedlichen Plattformen
Planungsqualität unabhängig von Engagement des MA	Ungenauere Durchführung des Bulk-Engineerings hat keinen Einfluss auf die Qualität der Dokumentation	Qualitätssteigerung
Einsparen des Bulk-Engineering	Durch Automatismen entfällt die manuelle Massenaarbeit	Zeit-/ Kosteneinsparung

Tabelle 3: Ermittelte Anforderungen an den Workflow, Quelle: Eigene Darstellung

Tabelle 3 beinhaltet die ermittelten Anforderungen an den Workflow. Bei Erfüllung dieser tritt der erwartete Nutzen ein, der in Spalte drei beschrieben ist.

6 BASISUNTERLAGEN FÜR DEN TEST-WORKFLOW

Um den Workflow mit möglichst praxisnahen Projektbedingungen zu entwickeln, wurden typische Basisunterlagen einer EMSR-Detailplanung herangezogen. Diese Dokumente haben ihre Vorlage in realen Projekten, der Inhalt wurde für diesen Ansatz stark reduziert.

6.1 R&I-Fließschema

Als Grundlage dient ein frei erdachtes Fließschema, das in Abbildung 16 dargestellt ist. Es beinhaltet zwei unterschiedliche Rohrleitungen, in denen Ventile und Messungen verbaut sind. Darunter zwei händisch betätigte Armaturen sowie ein mit einem pneumatischen Zylinder betriebenes Ventil. Die Messungen werden mittels Messgeräten (hier nicht dargestellt) in den Rohrleitungen durchgeführt.

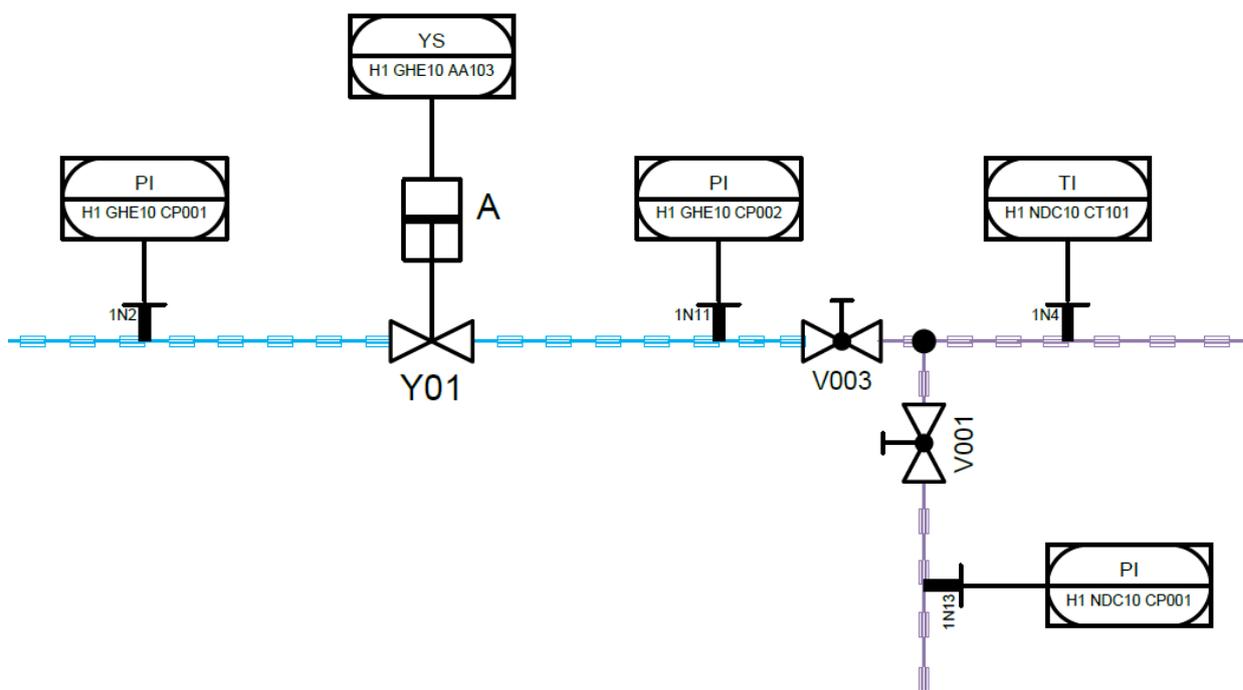


Abbildung 16: R&I-Fließschema-Ausschnitt der betrachteten Kreise, Quelle: Eigene Darstellung

An der Form der EMSR-Stellensymbole ist zu erkennen, dass die Signale aus einer Prozessleitwarte bedient und beobachtet und mit einem Prozessleitsystem verarbeitet werden. Erkennbar an dem horizontalen Strich in der Mitte sowie an dem abgerundeten Rechteck, das von einem normalen Rechteck eingefasst ist. Im oberen Teil des Symbols befindet sich die Kategorie (Druck- bzw. Temperaturanaloganzeige und Auf/Zu-Ventil) und im unteren Teil die eindeutige Bezeichnung der Stelle. Da in diesem Beispiel die Bezeichnung relativ lang ausfällt, wurde auf die Langform der Symbole zurückgegriffen.⁹⁵

⁹⁵ Vgl. Bindel/Hofmann (2016), S. 23-25 u. 38-41.

6.2 Messkreis- und Stellstellenliste

Diese beiden Listen resultieren aus dem zuvor beschriebenen R&I-Fließschema. Alle darin enthaltenen Mess- und Stellstellen werden, zumeist durch Mechanismen der Erstellungsumgebungen, automatisch aus dem Schema in die Liste geschrieben. Die Listen sind zweigeteilt dargestellt, da die Anzahl der Spalten eine durchgehende, dem Original getreue Darstellung erschwert.

In der Messkreisliste befinden sich, wie in Abbildung 17 ersichtlich, vier Messungen, drei Druck- und eine Temperaturmessung. Jede davon nimmt eine Zeile in Anspruch, die mit der eindeutigen Kennzeichnung der Messstelle beginnt. Danach folgt ein beschreibender Langtext, für welche Funktion die Messung verwendet wird. Anschließend sind einige Informationen zum Messgerät, z.B. der Gerätetyp, die Messgröße, Spannungs- bzw. Stromart angeführt. Signaltyp und -pegel sind für die Planung der Hard- und Software entscheidende Informationen, ebenso wie der Messbereich. Zur Ermittlung von spezifischen Gerätedaten empfiehlt es sich, genaue Geräteinformationen bzw. -bezeichnungen wie Hersteller und Typ in der Liste zu führen.

A	B	C	D	E	F	G	H	I	J
1 KKZ	Beschreibung	Geräte Definition	Messgröße	Funktion	IP-Code	SpgArt	Stromart	Signaltyp	Signalpegel
2 H1NDC10CT101	Temperaturmessung Rücklauf FW-HL	Temp.Kopftransmitter mit Pt100	Temperatur	TI	IP54	VDC	Gleichstrom	Analog	4...20 mA
3 H1NDC10CP001	Druckmessung Bypass FW-HL	Drucktransmitter	Druck	PI	IP54	VDC	Gleichstrom	Analog	4...20 mA
4 H1GHE10CP001	Druckmessung vor Hauptventil FW-RL	Drucktransmitter	Druck	PI	IP54	VDC	Gleichstrom	Analog	4...20 mA
5 H1GHE10CP002	Druckmessung nach Hauptventil FW-RL	Drucktransmitter	Druck	PI	IP54	VDC	Gleichstrom	Analog	4...20 mA

A	K	L	M	N	O	P	Q	R	S
1 KKZ	Verarbeitung mit	MessB von	DarMessb bis	MessB Einheit	Lieferant	Hersteller	Typ	mt01-HWTyp	mt02-Schrankzuordnung
2 H1NDC10CT101	PLS	0	150	°C	ABB	ABB	TSP111-Y0.W1(70).M2.U4.S1.D6	MSR_TL_Pt100_4-20mA_2L	+CKC01GH001 (H1)
3 H1NDC10CP001	PLS	0	16	bar	ABB	ABB	261GS-UKPN21/L1/I1/H3/M1/CB	MSR_PL_4-20mA_2L	+CKC01GH001 (H1)
4 H1GHE10CP001	PLS	0	16	bar	ABB	ABB	261GS-UKPN21/L1/I1/H3/M1/CB	MSR_PL_4-20mA_2L	+CKC01GH001 (H1)
5 H1GHE10CP002	PLS	0	16	bar	ABB	ABB	261GS-UKPN21/L1/I1/H3/M1/CB	MSR_PL_4-20mA_2L	+CKC01GH001 (H1)

Abbildung 17: Messungen der Messkreisliste untereinander dargestellt für bessere Übersicht, Quelle: Eigene Darstellung

Nach den Spalten der originalen Liste folgen die Spalten, die durch die EMSR-Planung hinzugefügt werden, dies wird als *Aufbereitung* der Informationen bezeichnet. In diesem Fall wird jeder Messstelle ein Hardware-Typical (nähere Beschreibung siehe 8.3) zugeordnet. Zusätzlich wird adaptiert, in welchen Schaltschrank die Signale an die dezentrale Peripherie geführt werden.

Die Stellstellenliste, dargestellt in Abbildung 18, ist grundsätzlich gleich aufgebaut wie die Messstellenliste. Beginnend mit der Kennzeichnung und der Beschreibung, werden in diesem Fall spezifischere Gerätedaten wie die elektrische Leistung angeführt. Am Ende werden wiederum das HW-Typical und die Zuordnung zum Schaltschrank im Zuge der Aufbereitung ergänzt.

A	B	C	D	E	F	G	H	I
1 KKZ	Beschreibung	Funktion	Spannung	SpgArt	Netzart	IP-Code	FailSafe	Nennleistung [kW]
2 H1GDH10AA103	Hauptventil Rücklauf Fernwärmeleitung	YS	24	VDC	USV	IP00	Nein	0,01

A	J	K	L	M	N	O	P
1 KKZ	Leistungsfaktor	DN	Bemerkung	Hersteller	Fabrikat	mt01-HWTyp	mt02-Schrankzuordnung
2 H1GDH10AA103	1	200		metso	Auf/Zu Ventil mit Stellungsrückmeldung	ARM_P_AufZu	+CKC01GH001 (H1)

Abbildung 18: Ventil der Stellstellenliste untereinander dargestellt für bessere Übersicht, Quelle: Eigene Darstellung

Nicht jede Information aus den Listen ist für die mess-, steuer- und regeltechnische Umsetzung einer Anlage notwendig. Um diese Inhalte möglichst gering zu halten, bedarf es Abstimmung zwischen den beteiligten Gewerken, welche Werte für die Planungen benötigt werden.

6.3 Lastenheftauszüge

Im Lastenheft wird vom Auftraggeber beschrieben, wie er sich die Umsetzung seiner Anlage vorstellt bzw. wie bereits bestehende ausgeführt und dokumentiert sind. Folgend werden einige Auszüge, die als Grundlage für diese Arbeit herangezogen werden, näher erläutert.

Die Bezeichnung der gesamten Anlage wird unter Verwendung des *Kraftwerk-Kennzeichensystems*, kurz *KKS* vorgenommen. Eine Vorgabe dieser Art kann wie in Abbildung 19 gestaltet sein.

2.2.2.2 Grundlegende Vorgaben

2.2.2.2.1 Nummerierungssystem

Die gesamte Anlage ist nach dem KKS (Kraftwerks-Kennzeichnungs-System) zu nummerieren bzw. zu kennzeichnen.

Bezeichnet werden alle in der Anlage installierten maschinen-, elektrotechnischen- sowie bau- und leittechnischen Einrichtungen, alle Messstellen (auch örtliche Messstellen) und zugehörigen Einrichtungen, weiters alle Nebeneinrichtungen und sämtliche Gebäudeeinrichtungen.

Abbildung 19: Auszug der Vorgaben bezüglich Kennzeichnungssystem, Quelle: Eigene Darstellung

Das KKS, ist ein standardisierter Kennzeichnungsschlüssel, der durch die Vereinigung *der Großkesselbesitzer* standardisiert ist. Damit wird alles, von der Anlage bis zum Gerät, abhängig von Aufgabe oder Ort in der Kraftwerksanlage gekennzeichnet. Es ist dabei nicht starr in seiner Auslegung, sondern kann mit anderen Systemen kombiniert und benutzerspezifisch angepasst werden.⁹⁶

Ein anderer Auszug aus einem Lastenheft bezieht sich auf den Aufbau der Stromlaufpläne. In Abbildung 20 ist besonders jener Punkt herauszuheben, der besagt, dass die Pläne unterbrechungsfrei dargestellt werden. Besonders bei looporientierter Darstellung, also eine Stelle pro Stellenplan, ist die Durchgängigkeit sinnvoll.

3.2.4 Aufbau Schaltpläne:

- Strukturierte Darstellung der Mess- und Verbraucherkreise
- Nennströme und Leistungen aller Antrieb, Nennströme von Sicherungen, Einstellwerte von Auslösern, Wächtern, Zeitrelais, Seriennummer bei Getriebemotoren
- Die einzelnen Schaltkreise sind so zu dokumentieren, dass die unterbrechungsfreie Darstellung vom jeweiligen Automaten bis hin zum Geber im Feld inklusive aller dazwischen befindlicher Geräte, Klemmen, Klemmenkästen, Kabel, Adern leicht und unmissverständlich, ohne weitere Listen oder Unterlagen, zu erkennen sind.
- Die Darstellung von mehrpoligen Kontakten erfolgt am selben Blatt mittels Querverweise am Kontaktspiegel.
- Aus den Stromlaufplänen ist ein eindeutiger Querverweis über die Betriebsmittelkennung zu den Stücklisten herzustellen
- Alle Leistungs- und Steuerkabelverbindungen sind am Plan mit der entsprechenden Kennzeichnung einzuzeichnen.
- Kennzeichnung der elektrischen Betriebsmittel
- Es sind Gegenklemmenbezeichnungen (Zielzeichen) einzutragen und an den Schnittstellen muss der gegenseitige Hinweis zum nächsten Plan vorhanden sein auch wenn dieser Bestandteil einer Fremdlieferung ist. Die Schnittstellenklärung sowie das Einholen aller Informationen obliegt den AN. Dieser ist auch für die Eintragung (spätestens bei der IBN) verantwortlich.

Abbildung 20: Vorgabe zum Aufbau der Stromlaufpläne, Quelle: Eigene Darstellung

⁹⁶ Vgl. VGB Power Tech e.V. (2019), Online-Quelle [03.11.2019]

Gründe dafür sind, dass die eindeutige Zuordnung durch die Kreisnummer eine einseitige Darstellung ermöglicht. Um diese Art der Darstellung zu erreichen, werden gewisse Geräte sowie Funktionen auf zwei Planseiten doppelt dargestellt, darunter z.B. der SPS-IO-Kanal. Ein großer Vorteil sind die verbesserten Bedingungen bei der Fehlersuche. Existiert nur ein Plan, auf dem die komplette Verdrahtung und Funktion dargestellt ist, verringert sich der zeitliche Aufwand beim Verstehen der Schaltung. Ein Nachteil ergibt sich bei der Fertigung der Schaltschränke, da eine größere Anzahl an Blättern entsteht, die betrachtet werden müssen.

Eine weitere Vorgabe, die durch ein Lastenheft gegeben sein kann, ist in Abbildung 21 ersichtlich. Sie beinhaltet die zu verwendenden Farben, in der die Drähte der internen Schrankverdrahtung ausgeführt werden sollen.

<p>6.3.11 Ausführungsmerkmale Aderfarben:</p> <p>ROT Steuerspannung 220V AC - L ROT-WEISS Steuerspannung 220V AC - N ORANGE Fremdspannung DUNKELBLAU + 24V DC BLAU-WEISS - 24V DC WEISS kleiner 24 V DC (z.B. Kaltleiter) Messleitungen von Klemmleiste zu SPS oder anderen Geräten 2 polig mit Schirm Aderfarbe Braun-Weis</p>
--

Abbildung 21: Mögliche Ausführungsrichtlinie bezüglich der schrankinternen Aderfarben, Quelle: Eigene Darstellung

In der Dokumentation kann dies auf zwei Varianten realisiert werden. Entweder ist jede elektrische Verbindung mittels grafischem Symbol gekennzeichnet oder der Farben-Standard ist einmalig in Form eines Informations-Deckblatts festgehalten.

Diese drei Auszüge aus Lastenheften stehen an dieser Stelle stellvertretend für eine Vielzahl an möglichen Vorgaben. Deren Inhalte können organisatorischen oder technischen Hintergrund besitzen und in verschiedene Themenbereiche eingreifen. Der Umfang solcher Lastenhefte kann bei wenigen bis zu hunderten Seiten liegen. Grundsätzlich gilt, dass ein genaues Studium aller relevanten Punkte zu Beginn der Planungsphase zu empfehlen ist, da es sich definitiv in alle Bereiche der eigenen Umsetzung auswirkt.

7 MODELLIERUNG DES META- UND ANLAGENMODELLS

Im nächsten Schritt wird die Basis zur Modellierung der Anlage gelegt. Im Metamodell sind die Objekte, Funktionen und deren Zusammenwirken beschrieben, damit daraus ein Modell unter bestimmten Regeln erstellt werden kann (siehe 3.3). Als Modellierungssprache wird UML verwendet, da es eine grafische Sprache ist und die notwendigen Funktionalitäten für diesen Anwendungsfall mit sich bringt (siehe 3.3.1.2). Folgend werden repräsentativ ausgewählte Teile des Metamodells näher beschrieben und anschließend alle Objekte tabellarisch aufgelistet. Darauf folgt die Erläuterung des daraus entstandenen Anlagenmodells.

7.1 Erläuterung Metamodell

Das Metamodell ist als Strukturdiagramm dargestellt, in dem alle benötigten Objekte und deren Verknüpfungen bzw. Wechselwirkungen dargestellt sind.

7.1.1 Das Metamodellobjekt

Der Aufbau dieser UML-Objekte ist grundsätzlich immer gleich, die Ausprägung der Detaillierung unterscheidet sich jedoch abhängig von den notwendigen Informationen. Das in Abbildung 22 dargestellte Metaobjekt beinhalten zwei Abschnitte. Im oberen Abschnitt befindet sich die Bezeichnung des Metaobjekts, in diesem Fall *Verbindungselement*. Es handelt sich dabei um den *Stereotyp* des eigentlichen Modellobjekts.



Abbildung 22: UML-Objekt *Verbindungselement*, Quelle: Eigene Darstellung

Darunter werden die Attribute aufgelistet, die dem Metaobjekt zugeordnet sind. Im Metamodell sind diesen noch keine Werte zugeordnet, jedoch sind zur besseren Verständlichkeit mögliche Inhalte voreingetragen. Darüber hinaus ist es empfehlenswert, diese Beispielwerte bereits hier einzupflegen, da es bei der Erstellung des Anlagenmodells als Formvorlage verwendet werden kann.

7.1.2 Die Beziehungstypen

Um die beschriebenen Objekte miteinander zu verknüpfen, werden in UML verschiedene Beziehungstypen verwendet. Im speziellen Anwendungsfall sind drei unterschiedliche Verbindungsvarianten im Einsatz.

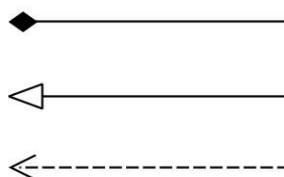


Abbildung 23: Verwendete Verbindungstypen im Metamodell, Quelle: Eigene Darstellung

Die in Abbildung 23 dargestellten Verbindungstypen haben folgende Bedeutung (von oben nach unten):

- Komposition: Kann als *Teil-von* Beziehung beschrieben werden, d.h. das untergeordnete Objekt (rechts der Linie) kann nicht ohne das übergeordnete (links der Linie) bestehen.
- Vererbung: Das untergeordnete Objekt (rechts der Linie) wird von dem übergeordneten (links der Linie) abgeleitet, d.h. dass z.B. Attribute des oberen auch im unteren vorhanden sind.
- Benutzt-Beziehung: Damit wird dargestellt, dass ein Objekt ein anderes zur Erfüllung einer Funktionalität benötigt.

7.1.3 Ausschnitt aus dem Metamodell

Da das erstellte Metamodell umfangreich in seiner grafischen Ausdehnung ist, wird anschließend lediglich ein Teil daraus näher beschrieben und die Funktionalität erklärt.

Abbildung 24 zeigt jenen Ausschnitt des Metamodells, der die funktionelle Hierarchie der Anlage wieder spiegelt. Am oberen Ende steht die *Anlage*, darunter zugeordnet die *Teilanlage* sowie die *Stelle*. Durch die Kompositionsbeziehung wird definiert, dass *Teilanlage* und *Stelle* nur dann existieren können, wenn die darüber angeordneten Objekte ebenfalls vorhanden sind. Grund dafür ist der vorgegebene KKS-Nummerierungsstandard, der in dieser Form aufgebaut ist.

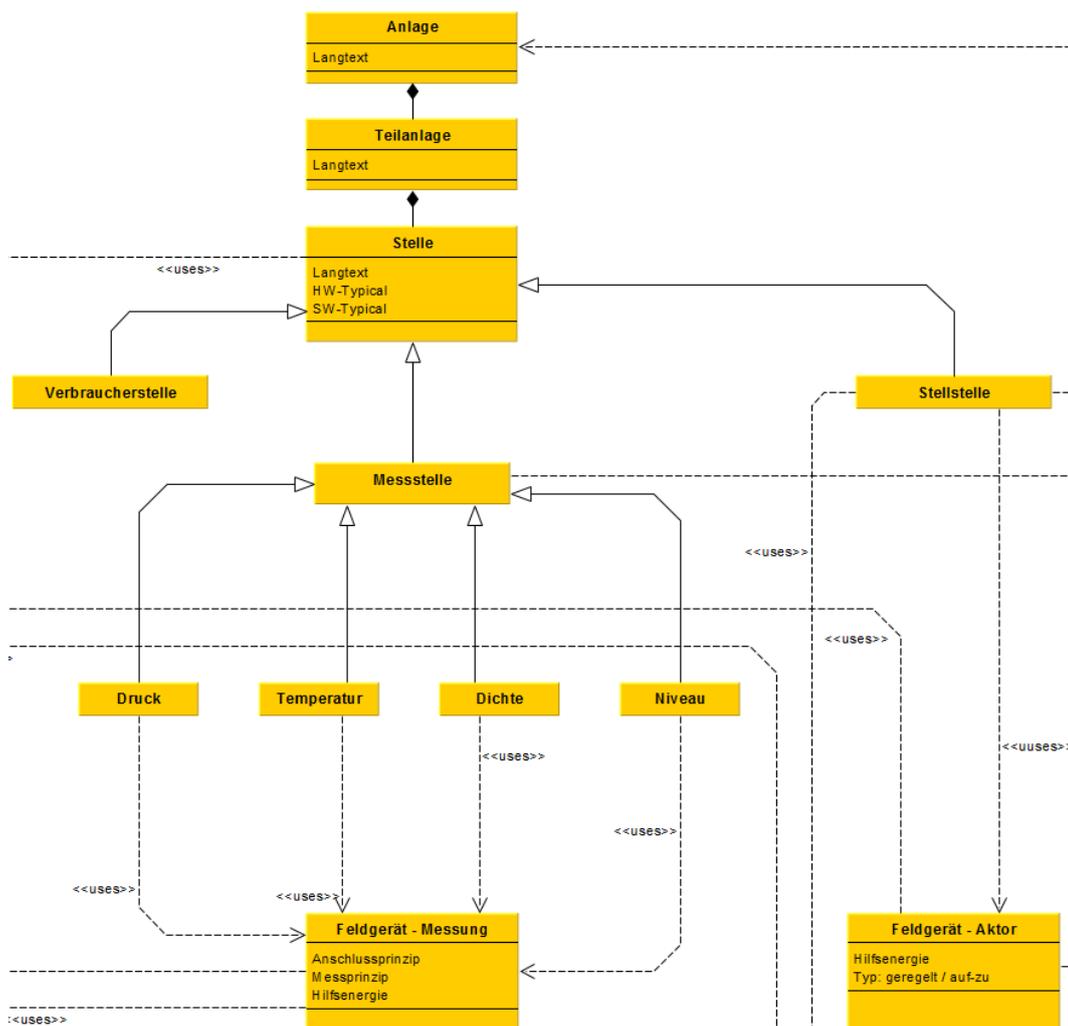


Abbildung 24: Ausschnitt des Metamodells zum Workflow, Quelle: Eigene Darstellung

Von der *Stelle* können drei unterschiedliche Varianten, *Verbraucherstelle*, *Messstelle* und *Stellstelle*, abgeleitet werden. Diese erben die Attribute (*Langtext*, *HW-Typical* und *SW-Typical*) der *Stelle* und stehen in den abgeleiteten Metaobjekten ebenso zur Verfügung. Die *Messstelle* vererbt diese Attribute noch weiter an die darunterliegenden Objekte *Druck*, *Temperatur*, *Dichte* und *Niveau*. Die *Stellstelle* besitzt keine weitere Untergliederung mehr und bezieht sich in weiterer Folge direkt auf physikalische Objekte. Aufgrund des großen Aufwandes, ein umfassendes Metamodell für eine Prozessanlage zu gestalten, wurde u.a. die *Verbraucherstelle* im Zuge dieser Arbeit nicht weitergehend untersucht bzw. modelliert. Die funktionelle Hierarchie einer Anlage kann mit den bereits beschriebenen Elementen abgebildet werden.

In weiterer Folge werden die physikalischen Objekte, in Abbildung 24 *Feldgerät-Messung* und *Feldgerät-Aktor*, mit dem funktionalen Aufbau in Verbindung gebracht. Bei den beiden genannten Objekten wird nicht jener Weg gewählt, ein übergeordnetes Objekt *Feldgerät* zu erstellen und die beiden davon abzuleiten. Grund dafür ist, dass die Gleichartigkeit der Objekte auf ein Attribut reduziert ist, das erst zu einem späteren Zeitpunkt nachgepflegt wurde. Die vier Messstellen-Varianten verwenden (ersichtlich durch die strichlierte *uses*-Verbindung) das Objekt *Feldgerät-Messung* mit den Attributen *Anschlussprinzip*, *Messprinzip* sowie *Hilfsenergie*. Gleichermaßen verwendet die *Stellstelle* das *Feldgerät-Aktor* mit den zugeordneten Attributen *Hilfsenergie* und *Typ*.

Im vollständigen Metamodell sind der physikalische Aufbau der Steuerungsebene, Verbindungselemente wie Kabel oder Luftschläuche, die elektrische sowie pneumatische Energieversorgung, Magnetventile zur Ansteuerung von Zylindern sowie Klemmleisten und die dazugehörigen Klemmen abgebildet. Die *benutzt*-Verbindung wird in zwei unterschiedlichen Varianten verwendet, zum einen die bereits beschriebene *uses*-Beziehung und zum anderen ist eine *belongs to*-Beziehung vorhanden. Diese wird für die Zuordnung der physikalischen Objekte zu z.B. einem Schaltschrank verwendet. Befindet sich das Gerät nicht im Verteiler, wird es einem anderen Objekt des Typs *Örtlichkeit* zugeordnet.

7.1.4 Auflistung aller verwendeten Metamodellobjekte

Um den Lesefluss dieser Arbeit nicht zu beeinträchtigen, ist die vollständige Aufzählung aller verwendeten Metamodellobjekte im Anhang beigefügt. Darin befindet sich jedes Objekt mit den Informationen wovon es abgeleitet ist, welche Attribute es besitzt und welche Kompositions- bzw. Verwendungsbeziehungen bestehen.

7.1.5 Erstellvorgang des Metamodells

Der Vorgang bei der Erstellung des Modells ist gemischt zwischen dem *Top-Down*- und *Bottom-Up-Ansatz* (siehe 3.2.1). Zu Beginn wird auf Basis von Erfahrungswerten die notwendige funktionelle Strukturierung in das Metamodell eingebracht und von der *Anlage* bis zur *Messstelle* aufgebaut. Die gegensätzliche Vorgangsweise wird bei der Modellierung vom *Signal* bis hin zur *Steuerung* durchlaufen. Hierbei besteht zu Beginn das *Signal*, um über die *IO-Karte* bis hin zur *Steuerung* zu gelangen. Wichtige Anhaltspunkte resultieren aus den Vorgaben der Basisunterlagen, z.B. die Gliederung aufgrund des KKS-Bezeichnungsstandards.

Der Vorgang wird sehr iterativ gelebt, d.h., es werden Anlagenmodelle erstellt und nachträglich das Meta-modell angepasst. Ein Beispiel dafür ist, dass die *Örtlichkeit* erst nach einigen Testläufen implementiert wurde, da die Endsysteme eine örtliche Zuordnung der physikalischen Geräte benötigen.

7.2 Anlagenmodell basierend auf den Basisunterlagen

Nach der Fertigstellung des Metamodells kann mit der Modellierung des Anlagenmodells begonnen werden. In diesem Unterkapitel werden der funktionelle Anlagenaufbau, jeweils eine Mess- bzw. Stellstelle sowie die Steuerungskomponenten näher beschrieben. Eine Gesamtansicht des Anlagenmodells ist aufgrund der Ausdehnung nicht möglich.

7.2.1 Funktionaler Aufbau des Anlagenmodells

Auf Basis der Mess- bzw. Stellenlisten, die den Basisunterlagen beiliegen (siehe 6.2), wird der funktionale Aufbau der Anlage modelliert. Der Spalte *A* der Listen wird die *KKS-Nummer* entnommen, die laut KKS-Standard in drei Teile untergliedert wird; der *Gesamtanlage*, der *Funktion* und dem *Aggregat*.⁹⁷ Die Nummer wird, auf das Metamodell umgelegt, folgendermaßen aufgeteilt: Zeichen 1-2 → *Anlage*; Zeichen 3-7 → *Teilanlage*; Zeichen 8-12 → *Stelle*

In Abbildung 25 wird die funktionale Struktur des Modells dargestellt, ohne weiterführende Verbindungen zu physikalischen Objekten. Die höchste Ebene ist die *Anlage H1*, der zwei *Teilanlagen*, *GHE10* sowie *NDC01*, zugeordnet sind. Darunter sind zwei bzw. drei Stellen angeordnet, dessen Attribute mit Werten aus den Listen befüllt sind, dem *Langtext* und dem *HW-Typical*.

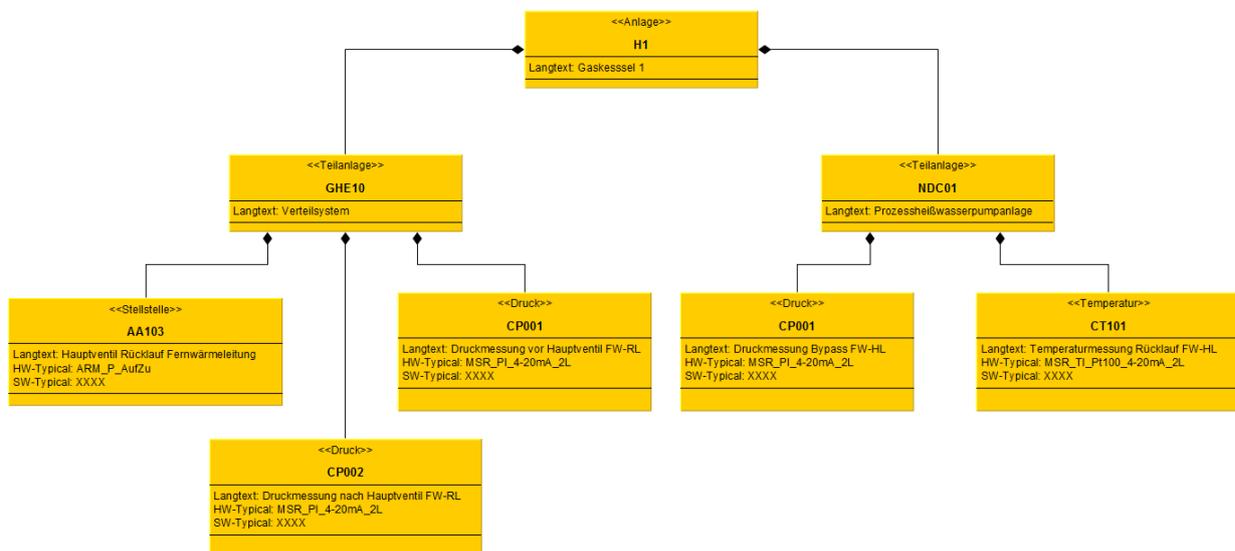


Abbildung 25: Modellausschnitt, der ausschließlich die funktionale Struktur der Anlage zeigt (nicht vollständig verbunden), Quelle: Eigene Darstellung

Das Attribut *SW-Typical* ist nicht befüllt, da diese Umsetzung aktuell lediglich für die Hardwareplanung vorgesehen ist. Für eine erweiterte Verwendung in Richtung Programmierung ist dieses Attribut bereits

⁹⁷ Vgl. VGB Power Tech e.V. 1 (2019), Online-Quelle [4.11.2019]

vorgesehen, findet jedoch noch keine Anwendung im Workflow. Zwischen allen dargestellten Objekten besteht eine Kompositionsbeziehung.

7.2.2 Druckmessstelle H1NDC01CP001

In Abbildung 26 wird o.a. Messstelle durch mehrere Modellobjekte dargestellt. Zentral befindet sich die *Druckstelle* selbst, wie sie bereits unter 7.2.1 in der Übersicht zu sehen ist. Dieses Objekt verwendet ein *Feldgerät-Messung* mit dem Betriebsmittelkennzeichen *-PT01*, eine *Klemmleiste* *-X05* mit den *Klemmen* *1* und *2*, ein *Kabel* *-W01* sowie ein *Analogsignal* *00*, auf dessen weitere Beziehungen unter 7.2.4 eingegangen wird.

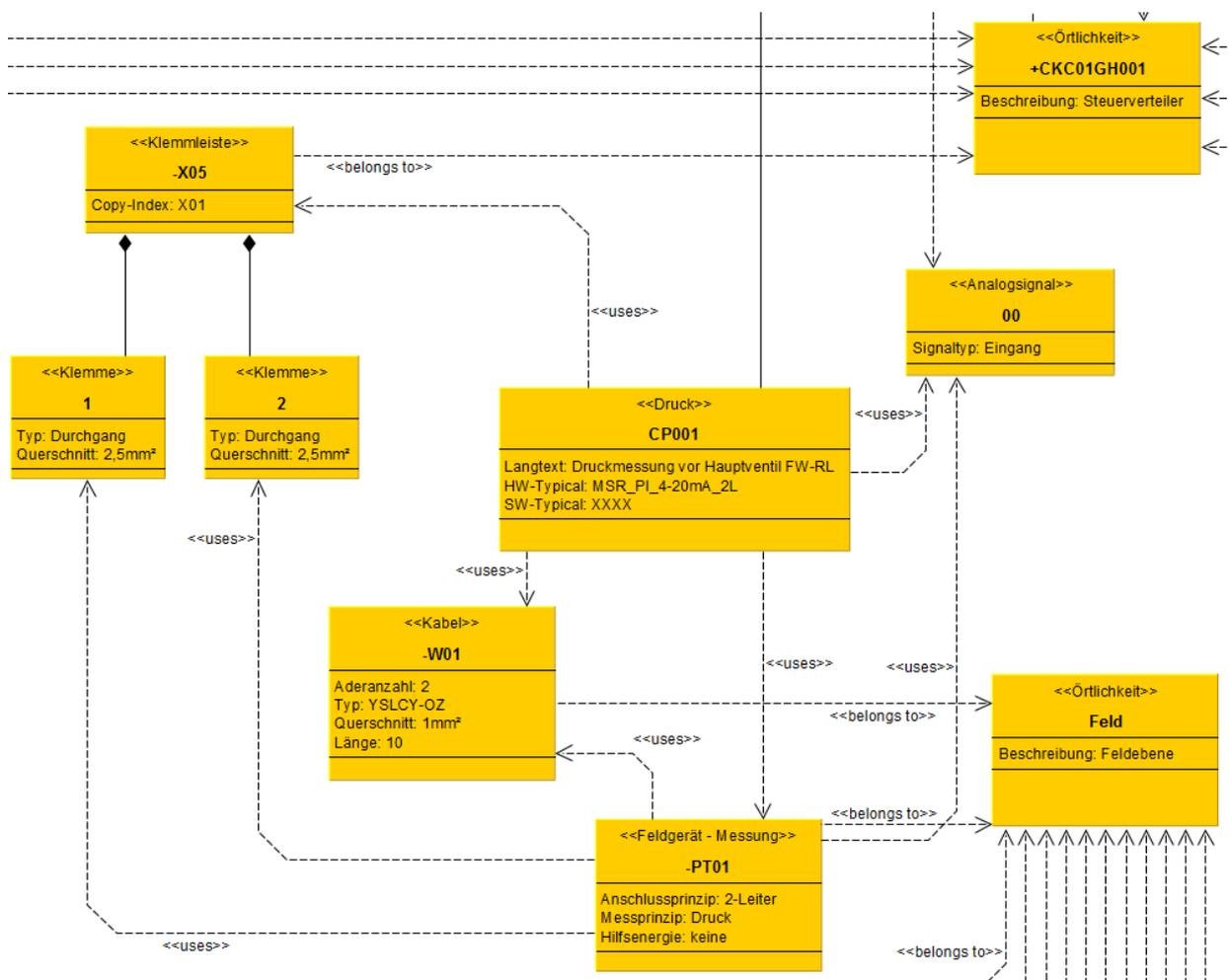


Abbildung 26: Darstellung der Druckmessstelle H1NDC01CP001 inkl. Verbindungen, Quelle: Eigene Darstellung

Die logische Fließrichtung des Messkreises ist von *Feldgerät-Messung* über ein *Kabel* zu den beiden *Klemmen*. Der analoge Eingang, dem das *Analogsignal* zugeordnet ist, ist ebenso mit den Klemmen verbunden. Das *Kabel*, aus dessen Attribute die Kabeltype *YSLCY-OZ 2x1mm²* ausgelesen werden kann, und das *Feldgerät-Messung* sind örtlich dem *Feld* zugeordnet, da sich diese vorort in der Anlage befinden. Jede *Klemmleiste* aller analogen Messstellen ist mit *-X05* bezeichnet und startet mit Klemmennummer *1*. Die eindeutige Unterscheidung ist jeweils über die Zuordnung zur *Stelle* selbst gegeben, dies wird als stellenorientierte Klemmleistenbezeichnung bezeichnet, eine typische Vorgabe, die aus dem Lastenheft stammt

und die Modellierung und Planung beeinflusst. Die *Klemmleiste* ist der *Örtlichkeit* +CKC01GH001 zugeordnet und beinhaltet das Attribut *Copy Index*, das speziell für die anschließende Umsetzung in einer Engineering-Umgebung notwendig ist. Signale mehrerer Messstellen werden gemeinsam im Schaltschrank +CKC01GH001 eingelesen. Dieser ist funktionell der *Anlage H1* zugeordnet.

Die weiteren Druckmessstellen sind exakt gleich aufgebaut, lediglich die Bezeichnung des *Analogsignals* unterscheidet sich, da pro Analogeingang nur eine *Messstelle* eingelesen werden kann. Die *Örtlichkeiten* +CKC01GH001 und *Feld* sind in der Anlage ebenso nur einmal vorhanden. Geräte, die mit anderen *Stellen* assoziiert sind, werden auch auf diese *Örtlichkeiten* verbunden.

Die in der Messstellenliste vorhandene *Temperaturmessstelle* ist ähnlich aufgebaut, unterscheidet sich ausschließlich darin, dass eine *Messstelle – Temperatur* verwendet wird und das Attribut *Messprinzip* im *Feldgerät-Messung PT100* ist. Der technische Aufbau der Messung unterscheidet sich nicht.

7.2.3 Stellstelle H1GHE10AA103

Die einzige Stellstelle, die aus den Basisunterlagen hervorgeht, ist im Modell, wie in Abbildung 27 dargestellt, realisiert. Hauptelement ist hier die *Stellstelle AA103*, die jeweils einen Wert für den *Langtext* und das *HW-Typical* besitzt. Die Fließrichtung der Stelle startet in diesem Fall am *Digitalsignal 00* mit dem *Signaltyp*-Attribut *Ausgang*, d.h. es handelt sich um einen Digitalausgang. Dieser ist über die *Klemmleiste -X10*, mit den *Klemmen 1* und *2*, mit dem *Magnetventil -MV01* verbunden. Beide Objekte sind örtlich +CKC01GH001 zugeordnet. Der *Feldgerät-Aktor, -Y01*, ist über einen *Luftschlauch* mit der Schlauchtype *1x6/8 PA-Schlauch* an das *Magnetventil* angeschlossen.

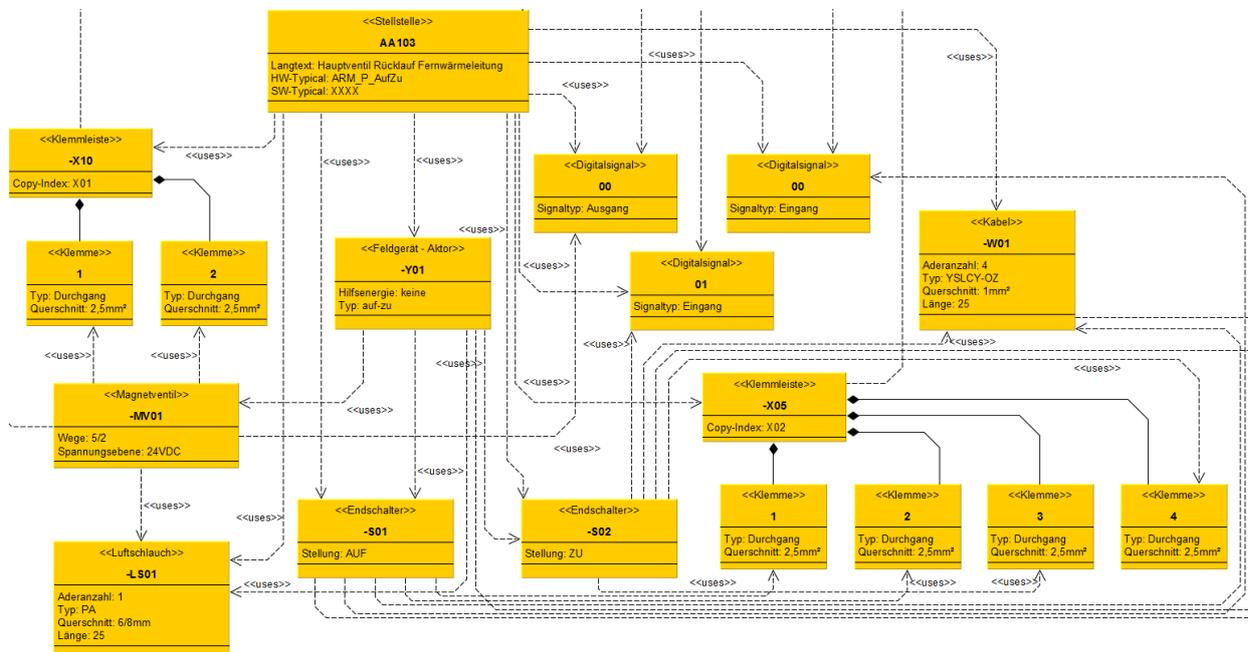


Abbildung 27: Darstellung der Stellstelle H1GHE10AA103 inkl. Verbindungen, Quelle: Eigene Darstellung

Auf dem *Feldgerät-Aktor* sind zwei *Endschalter* aufgebaut, die die Endpositionen *AUF* und *ZU* detektieren. *-S01* ist über die *Klemmen 1* und *2* der *Klemmleiste -X05* mit dem *Digitalsignal 00*, Typ *Eingang*, verbunden. Der zweite *Endschalter -S02* ist mit *Klemme 3* und *4* derselben *Klemmleiste* verdrahtet. Ziel dieser Verbindung ist ebenso das *Digitalsignal, 01*, im Schaltschrank +CKC01GH001. Als *Verbindungselement* wird in

diesem Fall das *Kabel -W01* mit der Kabeltype *YSLCY-OZ 4x1mm²* verwendet. Alle verwendeten Objekte, mit Ausnahme der *Klemme*, sind funktionell der *Stellstelle* zugeordnet und somit eindeutig im Anlagenmodell identifizierbar. *Feldgerät-Aktor*, *Kabel*, *Luftschlauch* und *Endschalter* sind örtlich dem *Feld* zugeteilt, während *Klemmleisten* und *Magnetventil* eine direkte Verbindung mit dem Schaltschrank aufweisen. Durch die *uses*-Beziehungen zwischen *Endschalter* mit *Digitalsignal*, *Endschalter* mit *Kabel* und *Endschalter* mit *Klemmen* wird modelliert, wie die Verbindung bis in die Steuerung zustande kommt. Das Attribut *Copy Index* ist in den beiden vorhandenen *Klemmleisten* mit unterschiedlichen Werten befüllt, um im Generierungsprozess der Stellenpläne eine eindeutige Zuordnung treffen zu können.

7.2.4 Steuerungskomponenten im Anlagenmodell

Ein weiterer Teil der Anlage, der im Modell abgebildet wird, sind die Steuerungskomponenten bzw. jener notwendige Anteil, um die vorhandenen Stellen zu verarbeiten. Hierarchisch betrachtet wird mit einer *Steuerung* gestartet, dargestellt in Abbildung 28. In diesem Fall eine *SPS*, die über *PN*, also das Profinet-Protokoll, kommuniziert. Diese verwendet eine *dezentrale Peripherie*, welche die Gesamtheit der danach folgenden Komponenten repräsentiert. Daher ist diese der *Örtlichkeit +CKC01GH001* über eine *belongs to*-Beziehung zugeordnet. Daraus lässt sich schließen, dass die Objekte, die darunter angeordnet sind, ebenso dieser *Örtlichkeit* angehören.

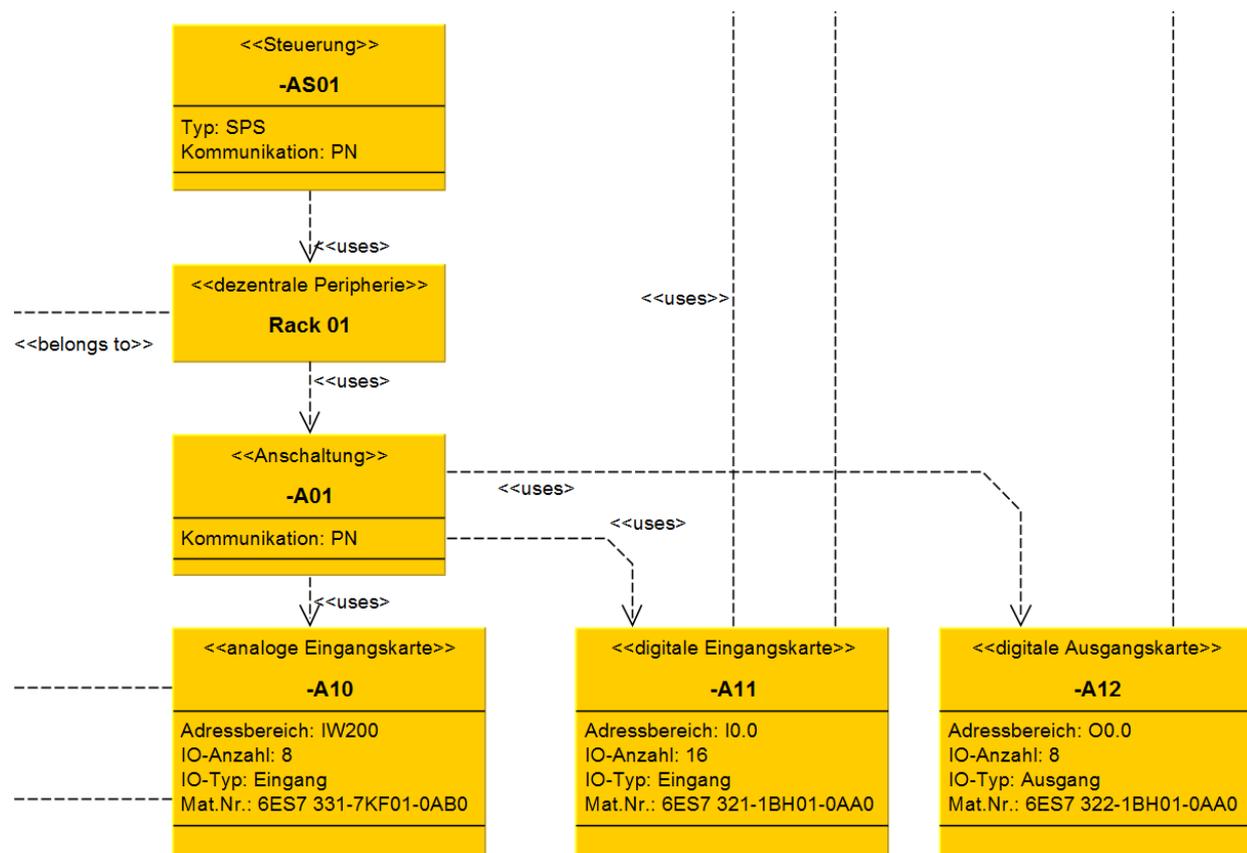


Abbildung 28: Darstellung der verwendeten Steuerungskomponenten inkl. Verbindungen, Quelle: Eigene Darstellung

Die *Anschaltung -A01* dient zur Kommunikation zwischen den IO-Baugruppen und der *Steuerung*. Um dies zu gewährleisten, muss auch die *Anschaltung* als *Kommunikation PN* unterstützen. Darauf folgend werden diverse IO-Karten platziert. Für den vorhandenen Anlagenausschnitt werden digitale und analoge Eingänge

sowie digitale Ausgänge verwendet. Daher werden die Objekte mit den Stereotypen *analoge Eingangskarte*, *digitale Eingangskarte* und *digitale Ausgangskarte* eingesetzt. Die darin verwendeten Attribute *Adressbereich*, *IO-Anzahl*, *IO-Typ* und *Mat.Nr.* sind jeweils befüllt. In *Adressbereich* wird die Startadresse der jeweiligen *IO-Karte* eingetragen. In Verbindung mit dem zweiten Attribut *IO-Anzahl* wird die IO-Kanaladresse für jeden einzelnen Ein- bzw. Ausgang festgelegt. Bei analogen IO-Karten wird der Startwert jeweils um zwei Byte erhöht, digitale Karten werden mit Bytes adressiert, die in 8 Bits (0-7) aufgeteilt sind. Der *IO-Typ* kann für einen Prüfmechanismus herangezogen werden, der die verbundenen Signale durchsucht und kontrolliert ob diese denselben *Signaltyp* aufweisen. Die *Mat.Nr.* findet ihre Anwendung in den Endplattformen, um die richtigen Geräte aus den Materialkatalogen zu verwenden.

Schlussendlich sind die drei benötigten *IO-Karten* mit den ihnen zugehörigen *Signalen* verbunden. Diese werden als die *Kanäle* der Baugruppen bezeichnet. Die Kanalbezeichnung startet bei *00* und wird pro Kanal um die Schrittweite eins erhöht.

7.2.5 Gesamtes Anlagenmodell

Wie bereits zuvor angedeutet, ist eine gesamte Darstellung des grafischen Modells in dieser Arbeit nicht möglich. Die größentechnische Ausdehnung aller Objekte gemeinsam lässt keine sinnvolle Gesamtdarstellung zu, aus der Informationen ersichtlich sind. Daher folgt nun eine kurze Beschreibung des Gesamtmodells:

Das obere Ende der Anlage umfasst, wie unter 7.2.1 beschrieben, die Aufteilung der Anlage in ihre funktionellen Teilanlagen. Von diesen geht es weiter zu insgesamt fünf einzelnen Stellen, die untereinander keine direkten Beziehungen haben. Die erste dieser Stellen ist die unter 7.2.3 beschriebene Stellstelle, gefolgt von drei Druckmessstellen, beschrieben unter 7.2.2, und einer Temperaturmessstelle, die sich von den Druckmessstellen nur in wenigen Details unterscheidet. Die Steuerungskomponenten sind gemeinsam angeordnet, wie unter 7.2.4 ersichtlich. Lediglich die einzelnen *Signale*, welche die Kanäle der Baugruppen darstellen, sind zur besseren Übersicht bei den jeweiligen Stellen dargestellt. Alle physikalisch vorhandenen Geräte haben entweder zur *Örtlichkeit +CKC01GH001* oder *Feld* eine *belongs to*-Beziehung, abhängig von der realen, örtlichen Anordnung in der Anlage selbst bzw. im Schaltschrank.

7.2.6 Simultanes Erarbeiten von Typicals und Anlagenmodell

In den Kapiteln, in denen soeben der Aufbau der Stellen im Anlagenmodell erklärt wurde, ist das HW-Typical bereits sehr stark eingeflossen. Tatsächlich werden die HW-Typicals und das Anlagenmodell parallel erstellt, da Erkenntnisse bidirektional aus dem einen in das andere eingeflossen sind. Ersichtlich ist dieses Vorgehen in Abbildung 15, dem Vorgehensmodell zu diesem Workflow.

8 ERLÄUTERUNG DER ENGINEERING-PLATTFORMEN UND NOTWENDIGE VORBEREITUNGEN VOR DER GENERIERUNG

Wie bereits in Kapitel 5.3 dargelegt, sind zwei Engineering-Plattformen ausgewählt, an denen der erarbeitete Workflow im Zuge dieser Arbeit getestet wird. Dabei handelt es sich um *Engineering Base*, ein Produkt der Firma *Aucotec*, und um *EPLAN Electric P8* der Firma *EPLAN*. Diese beiden Tools unterscheiden sich in einigen Punkten, aus diesem Grund hat diese Wahl einen repräsentativen Charakter für weitere Programme mit ähnlichen Eigenschaften. In diesem Kapitel werden die beiden Programme beschrieben und notwendige Vorbereitungsmaßnahmen erläutert.

8.1 *Engineering Base* der Firma Aucotec

Engineering Base, in der Kurzform auch *EB*, ist ein datenbankorientiertes Planungsprogramm der Firma *Aucotec*. Eingesetzt wird es zur Planung von elektrischen-, mess-, steuer- und regeltechnischen sowie verfahrenstechnischen oder mechanischen Anlagen. Es ist als Mehrschicht-System aufgebaut, d.h. es existiert eine *SQL-Datenbank* als oberste Schicht, die darin enthaltenen Objekte werden durch einen sogenannten *Application-Server* manipuliert. Die unterste Schicht bilden ein oder mehrere *Clients*, auf denen Benutzerinnen und Benutzer in einer grafischen Benutzeroberfläche, im sogenannten *Engineering Base Explorer*, Aktionen durchführen können. Diese Befehle werden an den *Application-Server* gesendet, der entweder Informationen aus der Datenbank ausliest und an den Explorer zur Anzeige zurückgibt oder eine Veränderung an den Daten durchführt. Als Programm für die Erstellung von Plänen wird *Microsoft Visio* eingesetzt, das auf die Datenbankobjekte zugreift.⁹⁸

8.1.1 Aufbau und Unterteilung der Datenbank bzw. deren Objekte

In einer *EB*-Datenbank existieren mehrere Datenbankbereiche mit unterschiedlichen Inhalten, die in Abbildung 29 aufgelistet sind. Die in der praktischen Anwendung am öftesten verwendeten Bereiche sind die *Projekte*, *Gerätstammdaten*, *Typdefinitionen*, *Schablonen* und *Attribute*.

	Projekte	Ordner mit Projekten
	Gerätstammdaten	Ordner mit kommerziellen und technischen Daten von Geräten
	Wörterbücher	Ordner mit Wörterbüchern für mehrsprachige Projekte
	Typdefinitionen	Ordner mit Typdefinitionen
	Projektvorlagen	Ordner mit Vorlagen für Projekte
	Add-In-Vorlagen	Ordner mit Vorlagen für Add-ins
	Assistenten	Ordner mit VBA-Assistenten
	Hinweise	Ordner mit Warnungen und Fehlermeldungen
	Schablonen	Ordner mit Schablonen
	Attribute	Ordner mit Attributen
	Vorlagen	Ordner mit Vorlagen für Blätter, Arbeitsblätter und Mappings
	Benutzergruppen	Ordner mit Benutzern und Gruppen für die Administrierung der Zugriffssteuerung
	Papierkorb	

Abbildung 29: Auflistung der *EB*-Datenbankbereiche, Quelle: Eigene Darstellung

Bei den *Typdefinitionen* handelt es sich um Definitionen zu den Objekttypen und deren Aufbau, bezogen auf die darin verwendeten Attribute. Unterteilt werden diese noch weiter in z.B. *Gerätetypen* wie *Automatisierungsgerät*, *Klemmleiste* oder *Leitungs-/Motorschutz*; *Funktionsstypen* wie *Anlage* oder *Messstelle* oder

⁹⁸ Vgl. Aucotec AG (o.J.), Online-Quelle [06.11.2019]

Kabeltypen wie Mehradriges Kabel. Diese Typen sind durch *Aucotec* vorgegeben und können nicht erweitert werden, jedoch können die Attribute, die dem Typ zur Verfügung stehen, angepasst werden. Diese Attribute werden im Bereich *Attribute* verwaltet und sind beliebig adaptierbar. Mittels sogenannten *Formelattributen* können z.B. Berechnungen von Werten aus anderen Attributen durchgeführt werden. Im Bereich *Gerätstammdaten* werden Echtgeräte gepflegt. Sie stehen für reale Artikel, die von diversen Herstellern angeboten werden und beinhalten Originalwerte wie z.B. Abmessungen, elektrische Leistung oder Kabelquerschnitt. Artikel können in den Projekten mehreren Datenbankobjekten zugewiesen werden und übergeben diesen ihre Werte sowie Objektstruktur. Die angesprochenen Projekte befinden sich in dem Datenbankbereich *Projekte*. Ihr Aufbau ist wie in Abbildung 30 dargestellt. Die drei wichtigsten Bereiche darin sind *Betriebsmittel*, *Funktionen* und *Dokumente*.

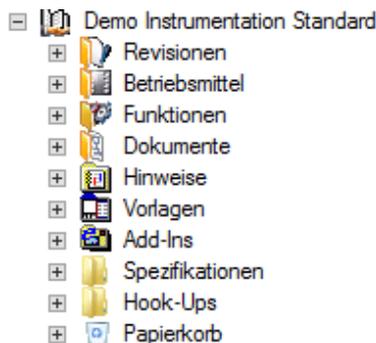


Abbildung 30: Gliederung eines *Engineering Base* Projekts im Objekt-Tree des *EB-Explorers*, Quelle: Eigene Darstellung

Unter *Betriebsmittel* sind die physikalischen Objekte einer Anlage angelegt, z.B. ein Temperaturtransmitter, der in einer Rohrleitung montiert ist, oder die CPU einer SPS, die in einem Schaltschrank verbaut ist. Darüber hinaus werden darin sogenannte *Baueinheiten* als hierarchische Strukturierungseinheit verwendet, die zumeist die elektrischen Verteiler der Anlage darstellen. Unter *Funktionen* entsteht der funktionelle Aufbau der Anlage. In diesem Ordner befinden sich u.a. Objekte vom Typ *Anlage*, *Messstelle* oder *Mechanische Stelle*, die untereinander hierarchisch gliederbar sind. Zwischen *Betriebsmittel* und *Funktionen* besteht die Möglichkeit, Beziehungen herzustellen. Ist ein bestimmtes Betriebsmittel, z.B. der angesprochene Temperaturtransmitter, genau einer logischen Funktion einer Anlage zugeordnet, kann dieses Objekt einer Funktion, z.B. einer Messstelle, zugeordnet werden. Bezeichnet wird das in *EB* als *zugeordnete Funktion*. Im Ordner *Dokumente* sind alle Pläne, Reports und anderen Dokumente zu finden. Diesen kann sowohl eine *Funktion* als auch ein *Betriebsmittel* zugeordnet sein. Auf den Plänen befinden sich *Shapes*, die grafische Ausprägungen der Objekte darstellen. Die *Mastershapes*, sozusagen die Vorlage für die mehrfach verwendeten Shapes in den Plänen, werden unter dem Datenbankbereich *Schablonen* geführt und gepflegt. Unterschiedliche grafische Varianten können für die Objekte angelegt werden.

Vorteil einer objektorientierten Planungsumgebung ist, dass ein physikalisches Objekt einer Anlage im gesamten Projekt nur einmal besteht. Aus diesem Grund werden die Daten an nur einer Stelle angepasst und sind durchgängig in allen Plänen und Folgedokumenten stets aktuell. Werden Pumpendaten durch die Verfahrenstechnik ausgelegt und erstellt, setzt die Elektrotechnik direkt an der angelegten Pumpe an und arbeitet mit durchgängig aktuellen Daten an der Auslegung des elektrischen Antriebes.

8.1.2 Vorbereitungen in EB

Bevor in EB Pläne aus dem Modell heraus generiert werden können, sind einige Schritte auszuführen um, die Planungsplattform vorzubereiten. Es wird angenommen, dass alle notwendigen Mastershapes bereits in der Datenbank vorhanden sind. Darüber hinaus sind die Objekttypen mit den notwendigen Attributen versehen. Wäre dies nicht der Fall, müssten abhängig von den notwendigen Objekten die relevanten Mastershapes erstellt bzw. Attribute zugeordnet werden.

Um in EB Pläne automatisiert vervielfältigen zu können, muss der Assistent *Typical Manager* (nähere Beschreibung siehe 9.1) verwendet werden. Dieser setzt voraus, dass dem Zielprojekt ein Typicalprojekt zugewiesen ist. In diesem Projekt werden die Typicals für die automatische Generierung erstellt. Das Typicalprojekt unterscheidet sich vom Aufbau nicht von einem normalen Projekt, die angelegten Pläne und Objekte müssen allerdings bestimmten Regeln folgen.

Am wichtigsten ist, dass Baueinheiten, darunter fallen z.B. Schaltschränke, in sogenannte *Typical-Baueinheiten* aufgeteilt werden müssen. Der *Typical Manager* baut darauf auf, dass die Geräte in diesen *Typical Baueinheiten* im Typicalprojekt angelegt sind. Grundsätzlich handelt sich es dabei um ein Standard-Objekt vom Typ *Allgemeine Baueinheit*. Entscheidend ist die genaue Bezeichnung mit *Typical Baueinheit 1* bis *Typical Baueinheit 10*. Diese zehn in Betracht kommenden Baueinheiten ermöglichen es, in einem Typical theoretisch zehn unterschiedliche Einbauorte zu verwenden. Selbiges ist mit sogenannten *Aufstellungsorten* möglich. Sie sind zur räumlichen Unterteilung einer Anlage, z.B. in Schalträume oder Hallen, in Verwendung. Zu beachten ist, dass die Strukturierung der Dokumente im Typicalprojekt der späteren Ablage im Zielprojekt entspricht. Aus diesem Grund sollte die Ablagestruktur bereits beim Erstellen der Typicals bedacht werden. Diese kann durch normale Ordner oder sogenannte *Zeichnungen*, die prinzipiell wie Ordner funktionieren, jedoch mehrere Attribute besitzen, erreicht werden.

Im Gegensatz dazu ist die Strukturierung der Stellen, z.B. eine *Messstelle*, *Verbraucherstelle* oder *Stellstelle*, nicht wie im Zielprojekt aufzubauen. Diese können direkt unter dem Ordner *Funktionen* abgelegt werden. Zu beachten ist jedoch die Bezeichnung der Stelle, da diese als Typicalbezeichnung im Zielprojekt den Stellen zugeordnet wird. Daher ist auf eine adäquate und eindeutige Kennzeichnung zu achten.

Werden bereits im Typicalprojekt Echtgeräte aus den Gerätestammdaten verwendet, die anschließend auch im Zielprojekt verwendet werden sollen, muss in den Einstellungen beider Projekte der gleiche *Katalog* eingestellt werden.

Selbstverständlich ist es notwendig, die Typicals selbst im Typicalprojekt anzulegen. Die dazu notwendigen Vorgänge werden unter Abschnitt 8.3.1 erläutert.

Das Zielprojekt kann mit den Standard-Methoden in EB angelegt werden. Damit es für den Generierungsprozess bereit ist, sind einige wenige Einstellungen vorzunehmen. Werden Echtgeräte für die Planung der Anlage verwendet, muss derselbe Materialkatalog eingestellt sein wie im Typicalprojekt. Um den *Typical Manager* verwenden zu können, wird im Zielprojekt das Typicalprojekt hinterlegt, zu finden in den Projekteinstellungen in der Kategorie *Typicals*.

8.2 *EPLAN Electric P8* der Firma *EPLAN*

EPLAN Electric P8, anschließend lediglich als *EPLAN* bezeichnet, ist eine Elektrotechnik-Planungsumgebung, die in vielen unterschiedlichen Branchen in Verwendung ist. Das Programm baut teilweise auf Datenbanken auf, jedoch nicht in allen Bereichen. Unter anderem sind z.B. die Geräte und Funktionen in Dateien abgespeichert und verweisen aufeinander, die Artikelverwaltung hingegen ist als Datenbank ausgeführt. Der infrastrukturelle Aufwand zur Installation eines *EPLAN*-Systems ist dementsprechend geringer. Es kann ohne weiteres auf einem einzelnen Rechner installiert werden.

8.2.1 Strukturierung und Besonderheiten

Ein großer Vorteil von *EPLAN* ist, dass zügig mit der elektrotechnischen Planung begonnen werden kann. Aufwendige Vorbereitungsschritte sind die Ausnahme, außer es müssen komplexe Strukturen abgebildet werden. *EPLAN* verwendet zur Strukturierung von Anlagen- oder Maschinendokumentationen die in Abbildung 31 dargestellten Einheiten. Diese können zusätzlich durch einen Punkt unterteilt werden, z.B. =H1.NDC01.

Struktur	
Funktionale Zuordnung	==
Anlage	=
Aufstellungsort	++
Einbauort	+
Dokumentenart	&
Benutzerdefinierte Struktur	#
Anlagennummer	

Abbildung 31: Strukturierungseinheiten in *EPLAN*, Quelle: Eigene Darstellung

Wenn ein Projekt erstellt ist, können dessen Strukturen an die realen Gegebenheiten angepasst werden. Diese Einstellung muss nicht für das gesamte Projekt gleich sein, sondern kann auf einzelne Bereiche individuell angepasst werden. Darunter z.B. die Seiten-, Betriebsmittel-, Klemmleisten- oder Kabelstrukturierung.⁹⁹

Durch sogenannte *Navigatoren* wird in *EPLAN* auf Geräte, Klemmleisten oder Seiten zugegriffen. Die drei zumeist verwendeten sind in Abbildung 32 dargestellt. Diese ermöglichen die Bearbeitung von z.B. Geräten auch übergreifend über mehrere Darstellungen. Da *EPLAN* kein objektorientiertes Programm ist, sind Änderungen an einem Gerät, das als Symbol auf einem Plan platziert ist, nicht automatisch in allen Darstellungen direkt angepasst. Um dies zu erreichen, muss die richtige Bearbeitungsweise gewählt werden. Platzierungen kann es auf einer großen Auswahl verschiedener Planseiten geben. *EPLAN* bietet insgesamt mehr als zehn unterschiedliche Seitentypen an, darunter z.B. *Schaltplan allpolig*, *Schaltschrankaufbau* oder *Übersicht*. Hinzu kommen diverse auswertende Typen wie *Klemmenplan* oder *Kabelliste*.

⁹⁹ Vgl. Gischel (2016), S. 69.

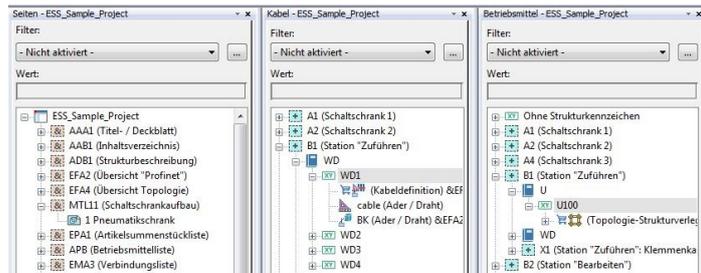


Abbildung 32: Ansicht einiger Navigatoren in EPLAN, Quelle: Eigene Darstellung

Der grafische Editor ist direkt in EPLAN implementiert und zeichnet sich durch eine intuitive Bedienung aus. Symbole, die darin platziert werden, sind zum überwiegenden Teil bereits in den Standard-Bibliotheken vorhanden und müssen nicht selbst angelegt oder verändert werden. Für spezielle Geräte, z.B. eine SPS-IO-Karte, werden sogenannte SPS-Kästen eingesetzt. Damit wird der Symbolbau durch die Anwenderin oder den Anwender nahezu ausgeschlossen. Darüber hinaus bietet EPLAN eine sehr umfangreiche und stets aktuelle Online-Artikeldatenbank an, das EPLAN Data Portal. Darin werden Informationen und sogenannte Makros, eine Komposition von Informationen, Anschlüssen oder Grafik, zu Artikeln bereitgestellt.

EPLAN bietet einige Möglichkeiten für Datenimport sowie -export. Alle Gerätetypen, egal ob allgemeine Betriebsmittel oder SPS-Kästen, können durch intelligente Routinen in z.B. ein XSLX-Format ausgegeben werden. Auf diese Weise können unter anderem IO-Adressen von IO-Karten oder elektrische Nennströme von Motoren in einem externen Programm wie z.B. Microsoft Excel auf eine schnellere und intelligente Art befüllt und wieder nach EPLAN importiert werden.

8.2.2 Vorbereitungen in EPLAN

Um in EPLAN die angesprochenen Makros erstellen zu können, wird ein Makroprojekt benötigt. Es ist grundsätzlich ein normales Projekt, bietet jedoch die Möglichkeit, solche Makros zu generieren. Für die Umsetzung in dieser Arbeit werden die EPLAN-Typicals als Makros vorbereitet. Unter 8.3.2 folgt die Beschreibung der Typicals selbst, nun werden die notwendigen Schritte erläutert, die vor dem Erstellen notwendig sind.

Zu Beginn wird ein neues Projekt erstellt, das auf der EPLAN Standard-Projektvorlage IEC_tp1001.ept basiert. Unter den Projekteinstellungen wird die Art des Projekts als Makroprojekt ausgewählt. Werden die Makros ausschließlich für ein spezielles Projekt erstellt oder ist die Struktur der Projekte immer gleich, empfiehlt es sich, die Strukturen des Makroprojekts an die des Zielprojekts anzupassen. Dies ist grundsätzlich nicht notwendig, erleichtert jedoch die Erstellung der Makros. Um ein solches zu generieren, sind einige wenige Vorgänge zu beachten. Zu Beginn wird eine Seite vom Typ Schaltplan allpolig angelegt, da in diesem Fall ausschließlich Schaltplanmakros erstellt werden. Anschließend wird die Schaltung, die zum Makro werden soll, auf dem Plan erstellt. Anschließend muss ein sogenannter Makrokasten auf der Seite eingefügt werden. Lediglich Symbole und Verbindungen innerhalb dieses Kastens werden zu einem gewünschten Makro zusammengestellt. Durch Öffnen der Eigenschaften des Makrokastens wird der in Abbildung 33 dargestellte Dialog geöffnet. Darin befindet sich die Bezeichnung sowie Version des Makros und eine Beschreibung z.B. der Schaltung, die damit abgebildet wird. Ist der Inhalt in gewünschter Weise

erstellt, wird im nächsten Schritt das Makro erstellt. Dieser Vorgang wird über das Kontextmenü des Makrokastens gestartet. Es erscheint erneut ein Dialog, in dem der Speicherort des Makros ausgewählt wird. Nach Bestätigung mit *OK* schreibt *EPLAN* das Makro als *.ema*-Datei in den gewünschten Ordner. Anschließend kann das Makro mehrfach in verschiedenen Projekten eingefügt werden.

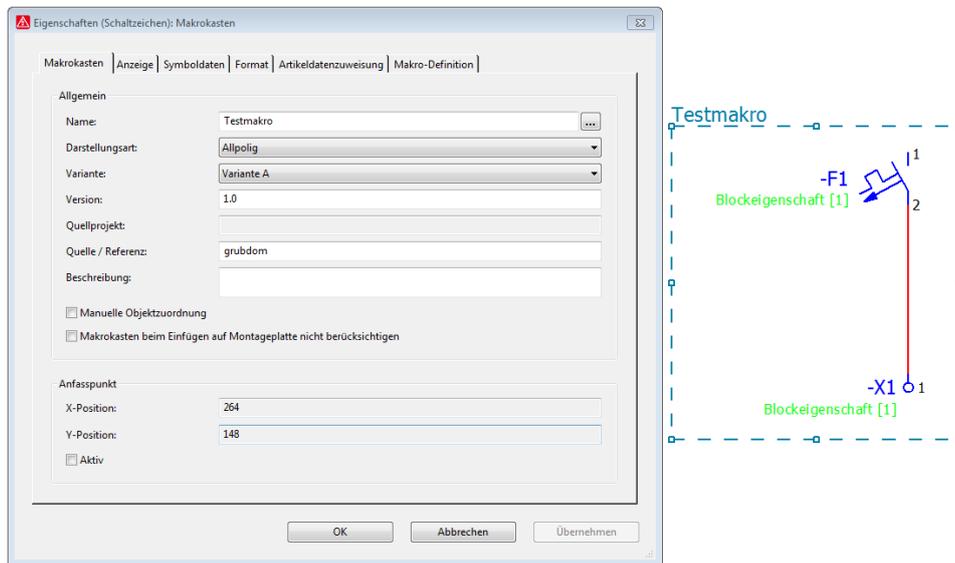


Abbildung 33: *EPLAN*-Makrokasten-Dialog inkl. Schaltung, Quelle: Eigene Darstellung

Gründe für diese Arbeitsweise sind, dass so eine hohe Wiederverwendung von einmal erdachten Planungsteilen möglich ist und in unterschiedlichen Projekten die Dokumentationen ähnlich bis gleich aussehen. Dies bringt wiederum einen Vorteil bei der Inbetriebnahme, da sich Techniker und Technikerinnen umgehend in den Plänen zurechtfinden.

Das Arbeitsprojekt wird auf Basis des gleichen Vorlageprojekts erstellt wie das Makroprojekt. Da die *Art des Projekts* standardmäßig auf *Schaltplanprojekt* eingestellt ist, ist diesbezüglich keine Änderung vorzunehmen. Die Strukturierung des Projekts wird hingegen spezifisch angepasst, wie in Abbildung 34 ersichtlich.

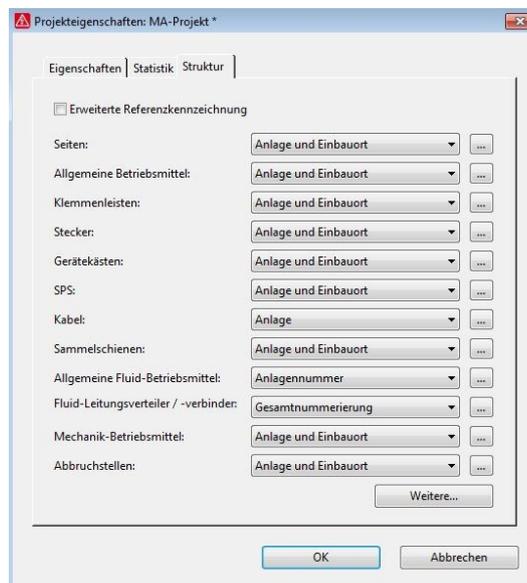


Abbildung 34: Strukturierung des Arbeitsprojekts (Ansicht im Eigenschaftsdialog), Quelle: Eigene Darstellung

Dazu werden bei nahezu allen relevanten Typen die beiden Strukturierungseinheiten *Anlage* und *Einbauort* ausgewählt. Sie sind das Äquivalent zu den Modellobjekten *Anlage*, *Teilanlage* und *Stelle* sowie Örtlichkeit. Der Typ *Kabel* wird nur mit *Anlage* strukturiert, da ansonsten in der Kabelbezeichnung ein + für den Einbauort verwendet wird. Die Kabel im erstellten Modell besitzen keinen Einbauort in der Form, wie er von *EPLAN* verwendet wird. Die Typen *Allgemeine Fluid-Betriebsmittel*, *Fluid-Leitungsverteiler/ -verbinder* und *Mechanik-Betriebsmittel* werden für diese Umsetzung nicht verwendet und sind daher wie im Vorlageprojekt eingestellt.

8.3 Typicals

In der EMSR-Hardware-Planung wird unter einem Typical ein Planungsdokument, z.B. einen Stellenplan, verstanden, der für mehrere Stellen eingesetzt werden kann, da diese technisch gleich umzusetzen sind. Aus den beiden Basislisten geht hervor, dass insgesamt drei verschiedene Typicals vorliegen. Zwei davon, *MSR_TI_Pt100_4-20mA_2L* und *MSR_PI_4-20mA_2L*, sind nahezu gleich und unterscheiden sich lediglich in der Type des Messgeräts. Das Typical der Stellstelle ist umfangreicher als jene der Messstellen, da darin Ansteuerung und Rückmeldungen sowie zusätzliche Geräte vorhanden sind.

Der technische Aufbau der beiden Messstellen besteht aus einem Messgerät, einmal ein Druckaufnehmer inkl. Transmitter und einmal ein PT100-Element inkl. Transmitter, die über ein Kabel in den Schaltschrank verbunden sind. Darin ist pro Messstelle eine eigene Klemmleiste vorgesehen, an der extern die Adern des Kabels und intern die IO-Karten über Drähte angeschlossen sind.

Wie bereits angesprochen, ist der Aufbau des Stellstellen-Typicals aufwendiger. Dazu wird im Schaltschrank eine digitale Ausgangskarte über eine Klemmleiste mit der Spule eines Magnetventils verbunden, das sich auf einer Ventilinsel befindet. Von dort wird es mit der benötigten Druckluft versorgt. Der pneumatische Zylinder, der die Armatur betätigt, befindet sich in der Anlage und wird mit dem Magnetventil über einen Pneumatikschlauch verbunden. Auf dem Zylinder sind zwei induktive Schalter angebracht, welche die Auf- und Zu-Stellung rückmelden. Beide gemeinsam sind über ein Kabel auf eine Klemmleiste verbunden. Im System werden die Rückmeldungen durch die digitale Eingangskarte erkannt, die mit den Klemmen verbunden ist.

Diese Typicals sind in den beiden Plattformen, basierend auf den darin geltenden Richtlinien und Mechanismen, erstellt. Folgend werden diese näher beschrieben.

8.3.1 Typicals in EB

Die Typicals werden, wie bereits beschrieben, in einem Typicalprojekt erstellt. Darin sind alle Geräte angelegt, die auf den Plänen platziert sind und anschließend im Zielprojekt angelegt werden. Da die Objekte der drei Typicals in zwei Örtlichkeiten, nämlich Schaltschrank und Feldebene, angeordnet sind, werden unter den Betriebsmitteln zwei *Baueinheiten* angelegt. Diese müssen die Bezeichnung *Typical Baueinheit x* tragen, damit sie vom *Typical Manager* erkannt werden. In diesem Fall wird die *Typical Baueinheit 1* als Feldebene und die *Typical Baueinheit 7* als Schaltschrankebene verwendet. Es werden nicht zwei aufeinanderfolgende Baueinheiten eingesetzt, da bei anderen Typicals eventuell mehrere Ebenen zwischen Feld und Schaltschrank vorhanden sind. Unter den Funktionen werden zwei Messstellen und eine Stellstelle ange-

legt. Der Funktionsname ist im Zielprojekt die Bezeichnung des Typicals. In *EB* sind die grafischen Stellenpläne im Hochformat dargestellt und in zwei Teilbereiche unterteilt. Im oberen Teil befindet sich die Schaltschrankebene und im unteren die Feldebene. Mit sogenannten *Baueinheitenrahmen* werden die Baueinheiten am Plan dargestellt. Alle darin befindlichen Objekte sind der jeweiligen Baueinheit zugeordnet.

8.3.1.1 Typical-Stellstelle *ARM_P_AufZu*

Zu Zwecken der Übersichtlichkeit wird der Stellenplan dieses Typicals unter Anhang 4.1 dargestellt. Es werden nun die Geräte, die darin platziert sind, erläutert. In Abbildung 35 sind alle verwendeten Geräte aufgelistet. In der *Typical Baueinheit 1*, ist die Armatur, die nicht im Modell abgebildet wird, der pneumatische Antrieb und die beiden darauf befindlichen Sensoren angelegt.

10 Datensätze	Zugeordnete Funktion	Teil von	Name	Typ	Typical Copy Merge Index
√ Filter	= ARM_P_AufZu	*	*	*	*
Oder	= ARM_P_AufZu GS01	*	*	*	*
Oder	= ARM_P_AufZu GS02	*	*	*	*
Oder	= ARM_P_AufZu YG01	*	*	*	*
1	ARM_P_AufZu	Typical Baueinheit 1	-SE0001	Sensor, Abstand Länge, Stellung	B01
2	ARM_P_AufZu	Typical Baueinheit 1	-SE0002	Sensor, Abstand Länge, Stellung	B02
3	ARM_P_AufZu	Typical Baueinheit 1	KA	Armatur, allgemein (Prozess / Fluid)	KA01
4	ARM_P_AufZu	Typical Baueinheit 1	MA	Antriebe (nicht elektrisch) (Prozess / Fluid)	MA01
5	ARM_P_AufZu	Typical Baueinheit 7	-MV01	Elektrisch angetriebenes mechanisches Gerät	Y01
6	ARM_P_AufZu	Typical Baueinheit 7	-X0x1	Klemmleiste	X01
7	ARM_P_AufZu	Typical Baueinheit 7	-X0x2	Klemmleiste	X02
8	ARM_P_AufZu GS01	Typical Baueinheit 7	DI01	Eingang	SIG02
9	ARM_P_AufZu GS02	Typical Baueinheit 7	DI02	Eingang	SIG03
10	ARM_P_AufZu YG01	Typical Baueinheit 7	DO01	Ausgang	SIG01
2 Datensätze	Zugeordnete Funktion	Teil von	Name	Typical Copy Merge Index	
√ Filter	= ARM_P_AufZu	*	*	*	
1	ARM_P_AufZu	Typical Baueinheit 1	-W20	WS01	
2	ARM_P_AufZu	Typical Baueinheit 1	-W99	WP01	

Abbildung 35: Geräte/Kabel des Stellstellentypicals in *EB*, Quelle: Eigene Darstellung

In der *Typical Baueinheit 7* sind die Schrankgeräte angelegt. Darunter fallen die beiden Klemmleisten mit zwei bzw. vier Klemmen, das Magnetventil (*Elektrisch angetriebenes mechanisches Gerät*) und die Ein- bzw. Ausgänge der IO-Karte. In *EB* sind IO-Karten so aufgebaut, dass die Karte selbst als ein *Automatisierungsgerät* angelegt ist und darunter, entsprechend der IO-Anzahl, Objekte vom Typ *Eingang* oder *Ausgang*. Die drei angelegten IOs sind im Typical noch keiner Karte zugeordnet, da im Zielprojekt ein Nachfolgeschritt nach dem Vervielfältigen notwendig ist. Die beiden eingesetzten Kabel werden mit einem eigenen Arbeitsblatt abgefragt und sind der Feldebene zugeordnet. Das Attribut *Typical Copy Merge Index* wird dazu verwendet, um Geräte, die bereits im Zielprojekt angelegt sind, mit den Objekten aus dem Typical zu vereinen. Ist z.B. ein Kabel bereits angelegt, wird durch das Typical kein neues Kabel angelegt, wenn die Bezeichnung und dieser Index übereinstimmen.

8.3.1.2 Typical-Messstelle *MSR_TI_Pt100_4-20mA_2L*

Der Stellenplan dieses Typicals ist unter Anhang 4.2 zu finden, die drei verwendeten Geräte sind in Abbildung 36 dargestellt. Durch den relativ einfachen technischen Aufbau dieses Typicals sind dementsprechend wenige Geräte im Einsatz, nämlich ein Temperatursensor und ein Kabel im Feld sowie die Klemmleiste mit zwei Klemmen und ein Eingangssignal im Schaltschrank. Der *Typical Copy Merge Index* wird hier für die Klemmleiste und das Kabel verwendet.

3 Datensätze	Zugeordnete Funktion	Teil von	Name	Typ	Typical Copy Merge Index
√ Filter	= MSR_TI_Pt100_4-20mA_2L	*	*	*	*
Oder	= MSR_TI_Pt100_4-20mA_2L	*	*	*	*
1	MSR_TI_Pt100_4-20mA_2L	Typical Baueinheit 1	-BT0001	Sensor, Temperatur	B01
2	MSR_TI_Pt100_4-20mA_2L	Typical Baueinheit 7	-X0x	Klemmleiste	X01
3	MSR_TI_Pt100_4-20mA_2L ...	Typical Baueinheit 7	AI01	Eingang	SIG01
1 Datensätze	Zugeordnete Funktion	Teil von	Name	Typical Copy Merge Index	
√ Filter	= MSR_TI_Pt100_4-20mA_2L	*	*	*	
1	MSR_TI_Pt100_4-20mA_2L	Typical Baueinheit 1	-W20	WS01	

Abbildung 36: Geräte/Kabel des Temperaturmessstellentypicals in EB, Quelle: Eigene Darstellung

8.3.1.3 Typical-Messstelle MSR_PI_4-20mA_2L

Wie bereits erwähnt, sind die beiden Messstellen-Typicals sehr ähnlich aufgebaut. Der Transmitter, der in Abbildung 37 aufgelistet ist, ist in diesem Fall vom Typ *Sensor, Druck* und besitzt ein anderes Betriebsmittelkennzeichen. Der grafische Stellenplan ist wiederum unter Anhang 4.3 als vollständiger Plan zu finden.

3 Datensätze	Zugeordnete Funktion	Teil von	Name	Typ	Typical Copy Merge Index
√ Filter	= MSR_PI_4-20mA_2L	*	*	*	*
Oder	= MSR_PI_4-20mA_2L XQ01	*	*	*	*
1	MSR_PI_4-20mA_2L	Typical Baueinheit 1	-BD0001	Sensor, Druck	B01
2	MSR_PI_4-20mA_2L	Typical Baueinheit 7	-X0x	Klemmleiste	X01
3	MSR_PI_4-20mA_2L XQ01	Typical Baueinheit 7	AI01	Eingang	SIG01
1 Datensätze	Zugeordnete Funktion	Teil von	Name	Typical Copy Merge Index	
√ Filter	= MSR_PI_4-20mA_2L	*	*	*	
1	MSR_PI_4-20mA_2L	Typical Baueinheit 1	-W20	WS01	

Abbildung 37: Geräte/Kabel des Druckmessstellentypicals in EB, Quelle: Eigene Darstellung

8.3.2 Typicals in EPLAN Electric P8

In *EPLAN* werden die Typicals in einem Makroprojekt als Makros vorbereitet. Bezüglich der technischen Realisierung sind sie natürlich den *EB*-Typicals identisch, jedoch gibt es grundsätzliche Unterschiede im Aufbau aufgrund von *EPLAN*-Mechanismen. Da die *Örtlichkeit Feld* in *EPLAN* nicht verwendet wird, entfällt die Abgrenzung des Planes im unteren Bereich. Der obere Teil wird trotzdem mit einem *Strukturkasten* eingegrenzt. Dieser bekommt einen allgemeinen *Einbauort* zugewiesen, da das Typical allgemeingültig sein soll. Ansonsten befinden sich darin wieder jene Geräte, die dem Schaltschrank zugeordnet sind. In *EPLAN* ist der Stellenplan im Querformat angelegt. Dies ist lediglich historisch bedingt, da im Unternehmen stets diese Darstellungsform gewählt wurde und für *EPLAN* nur solche Vorlagen vorhanden sind, im Gegensatz dazu gibt es in *EB* Standard-Hochformat-Vorlagen. Die Strukturierung der Stellenpläne wird über die *EPLAN*-Seite selbst durchgeführt, indem bei der Seiteneigenschaft *vollständiger Seitenname* die KKS-Nummer eingetragen wird. Eine Unterteilung in drei Untergruppen ist jedoch nicht möglich, daher wird die Gesamtanlage als ein Teil und die Funktion sowie das Aggregat als der zweite Teil verwendet. Die funktionale Bezeichnung ist somit wie folgendes Beispiel aufgebaut: *E1.MBD10CL001*. Die Strukturierung der *Örtlichkeiten* ist über den Einbauort der *EPLAN-Struktur* mit einem Strukturkasten realisiert.

Alle Typicals sind aufgrund der seitenfüllenden Größe im Anhang 5.1 bis 5.3 zu finden.

8.3.2.1 Typical-Stellstelle ARM_P_AufZu

Für dieses Typical ist es notwendig, zwei Geräte, das Magnetventil und der pneumatische Zylinder mit Ventilkörper, mit einem *Gerätekasten*, Anschlüssen und Grafik zu erstellen. Grund dafür ist, dass diese

Geräte nicht in der gewünschten Form in *EPLAN* als Symbol vorhanden sind. Die Betriebsmittelkennzeichen aller Symbole werden im Typical beliebig gewählt, da dies für die Vervielfältigung in *EPLAN* nicht relevant ist. Die verwendeten Signale der IO-Karten sind einmal vom Typ Digital-Ausgang und zweimal vom Typ Digital-Eingang, die jeweils mit einer eigenen Klemmleiste verbunden sind. Das Kabel und der Schlauch sind feldseitig und ohne die Strukturierungseinheit *Einbauort* dargestellt. Da ein geschirmtes Kabel im Einsatz ist, befindet sich der Schirm inklusive Schirmanschluss am Plan. Die Endschalter sind getrennt vom Zylinder angeordnet, jedoch über die Messkreisnummer diesem eindeutig zugeordnet.

8.3.2.2 Typical-Messstelle *MSR_TI_Pt100_4-20mA_2L*

In diesem Typical ist ein Analog-Eingang im Einsatz, der auf die Klemmleiste verdrahtet ist, die wiederum über ein Kabel am Transmitter angeschlossen ist. Das Feldgerät ist ein PT100-Element mit Kopftransmitter erkennbar durch die textuelle Beschreibung am Symbol und dem Hinweis *I/T* in der Mitte. Die Strukturierung von Funktion und Einbauort sind gleich wie im Stellstellen-Typicals realisiert.

8.3.2.3 Typical-Messstelle *MSR_PI_4-20mA_2L*

Dieses Typical ist dem Temperatur-Typical sehr ähnlich. Der Unterschied ist die Darstellung des Druckmessumformers, dessen Hinweis *I/PT* und Beschreibung der Druckmessbereich ist.

8.4 Erkenntnisse aus den Plattformen und Typicals

Durch den Umstand, dass die beiden Programme unterschiedlich aufgebaut sind, ist die Umsetzung der Typicals dementsprechend anders. Bei *EB* werden alle Geräte in einem Typicalprojekt als Objekte angelegt, die Strukturierung bzw. Benennung der *Örtlichkeiten* für den *Typical Manager* ist einzuhalten. Im Gegensatz dazu ist die funktionelle Strukturierung nicht wie im Zielprojekt aufgebaut, da nur die Stelle selbst existiert. In *EPLAN* dagegen werden die Typicals durch die Pläne selbst repräsentiert. Die Strukturierung entsteht funktionell über die *EPLAN*-Seite und örtlich mit einem *Strukturkasten* am Plan. Prinzipiell ist anzumerken, dass in *EPLAN* die Typicalerstellung weniger komplex ist, jedoch kein Plangenerator im Standard-Umfang vorhanden ist.

Zur Handhabung der Plattformen selbst ist, besonders in *EB*, ein Experte oder eine Expertin im Umgang mit dem Tool notwendig. In *EB* ist zu empfehlen, die Typicals für den *Typical Manager* passend vorzubereiten. Händisches Vervielfältigen der Stellen ist möglich, aber relativ zum Standard-Generator betrachtet, aufwendig. Aus diesem Grund ist der Mehraufwand im Typicalbau gerechtfertigt. Manuelles Kopieren in *EPLAN* ist jedoch wesentlich einfacher und für weniger umfangreiche Projekte eine durchaus gangbare Lösung.

In Verbindung mit dem MDA-Ansatz stellen die Typicals die plattformspezifische Schicht dar. Sie sind notwendig, um das unabhängige Modell in die gewünschten Endplattformen zu transformieren.

9 BESCHREIBUNG DER GENERATOREN

Nachdem unter Kapitel 8 die beiden Planungsplattformen beschrieben wurden, liegt der Fokus dieses Abschnitts auf den Generatoren. Der *Typical Manager* in *EB* wurde bereits mehrfach erwähnt und die Typicals bereits dementsprechend erstellt. Für *EPLAN* kommt der Schaltplangenerator G8 der Firma *CAE Expert Group* zum Einsatz. Im folgenden Kapitel werden die beiden Generatoren beschrieben und soweit vorbereitet, dass diese für das Erstellen des Modells bereit sind.

9.1 Der *Typical Manager* in *EB*

Den Ausgangspunkt des Managers bilden die *Funktionen* in *EB*. Sie enthalten ein Attribut, mit dem das zugehörige Typical ausgewählt wird. Passiert das Zuweisen der Typicals direkt in *EB*, wird dazu die Funktion geöffnet und beim Attribut *Typical* der Typical-Auswahldialog gestartet. Dieser greift auf die Funktionen des verbundenen Typicalprojekts zu und zeigt die Auswahl der möglichen Typicals, inkl. Vorschau, an. Anschließend wird das gewünschte Typical ausgewählt und der Dialog bestätigt. Wie bereits erläutert, besteht ein solches nicht ausschließlich aus Plänen, sondern auch aus Objekten, die in der Datenbank angelegt sind. Dessen hierarchische Ablage im Typicalprojekt ist entscheidend für die spätere Ablage und richtige Funktionsweise des *Typical Managers* im Zielprojekt. Sind alle Typicals zugewiesen, wird der *Typical Manager* auf dem Ordner *Funktionen* gestartet. Es erscheint der in Abbildung 38 gezeigte Dialog, der die vorhandenen Funktionen einliest. In dem dargestellten Beispiel sind noch keine Typicals vervielfältigt, daher steht in Spalte *Aktion* das *Kopieren* zur Auswahl. Ein weiteres Mal wird das Typical angezeigt sowie der Funktionsname und der *Typ* der Funktion, das Typical selbst könnte hier noch geändert werden.

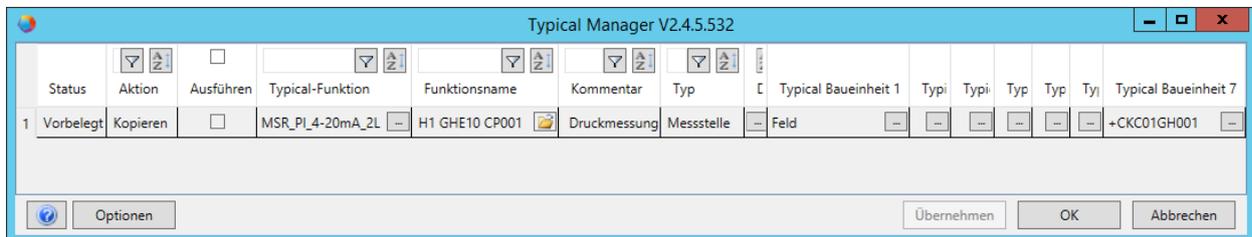


Abbildung 38: *EB*-Assistent *Typical Manager* zum Generieren von Stellenplänen, Quelle: Eigene Darstellung

Im rechten Teil des Dialogs befinden sich die bereits mehrmals angesprochenen *Typical Baueinheiten*. Diese werden entweder hier, im Dialog der Funktion oder über mehrere Funktionen hinweg in einem Arbeitsblatt zugeordnet. Der *Typical Manager* ist darüber hinaus über einige Optionen zusätzlich einstellbar. Abbildung 39 zeigt die beiden Register *Allgemein* und *Verhalten* des Optionen-Dialogs. Unter den allgemeinen Einstellungen ist es möglich, die Funktionstypen einzuschränken, die durch den Manager verarbeitet werden. Darunter wird die Anzahl der angezeigten *Typical Baueinheiten* eingestellt, in diesem Fall sieben, da die erste und siebente von den erstellten Typicals verwendet wird. Am unteren Ende des ersten Dialogs besteht die Möglichkeit, einen weiteren *EB*-Assistenten anzugeben, der, nachdem die Typicals ins Zielprojekt kopiert wurden, weitere Bearbeitungsschritte ausführt. Im Register *Verhalten* sind Einstellungen zur Vorgangsweise des *Typical Managers* untergebracht. Die erste Einstellung der Kategorie *Betriebsmittel* bewirkt, dass wenn Baueinheit 1 im Typical verwendet, jedoch in der Stelle nicht befüllt ist, die Geräte in einer Baueinheiten-Struktur abgelegt werden, die gleich der Funktions-Struktur ist. Durch Einstellung zwei

werden übergeordnete Strukturen von Typical-Geräten mitkopiert, wenn ein Ziel-Gerät und Typical-Gerät *verschmolzen* werden. Dies tritt ein, wenn im Zielprojekt ein Objekt bereits besteht und durch den Manager kein neues Gerät angelegt wird, sondern die bestehenden Informationen im Ziel-Objekt um die des Typical-Objekts ergänzt werden.

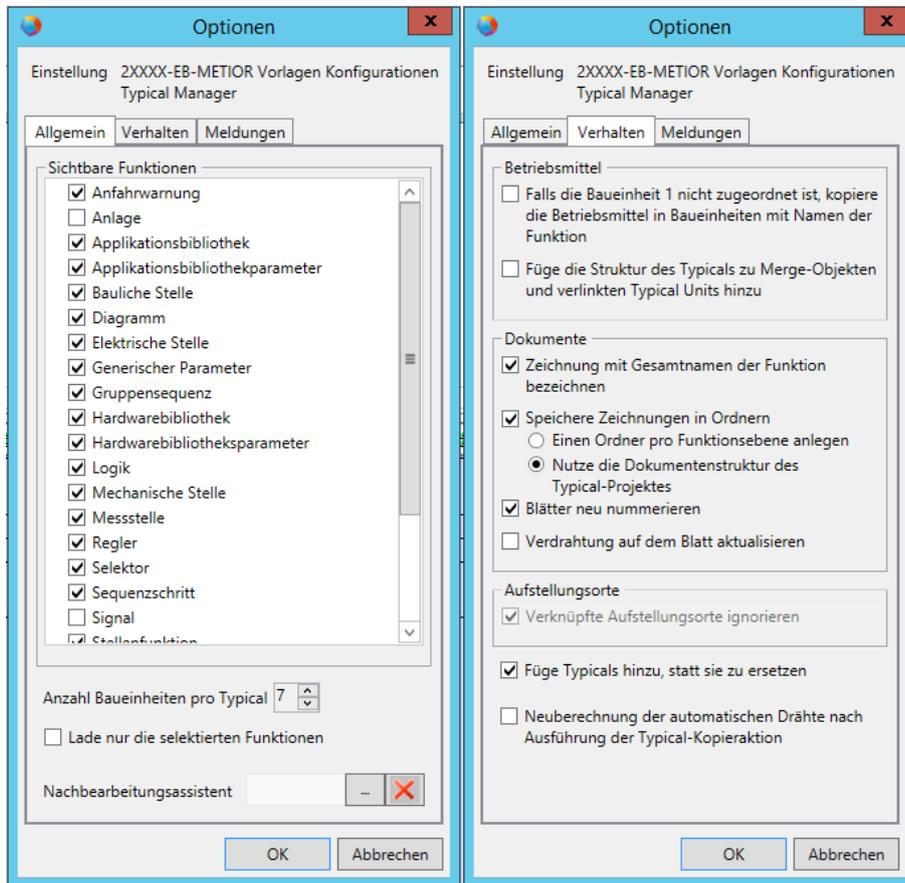


Abbildung 39: EB-Typical Manager Optionen-Dialog (Register *Allgemein* und *Verhalten*), Quelle: Eigene Darstellung

Die zweite Kategorie beschäftigt sich mit den Dokumenten der Typicals. Sind die Pläne in *Zeichnungen* abgelegt, werden diese, mit Hilfe der ersten Einstellung, mit dem Gesamtnamen der Funktion bezeichnet. Die zweite Einstellmöglichkeit bietet dem Benutzer, der Benutzerin an, die *Zeichnungen* in Ordnern zu speichern. Ist dies bereits im Typical-Projekt dementsprechend umgesetzt, empfiehlt es sich, die bestehende Struktur weiter zu nutzen. Ist das nicht der Fall, kann hier ebenso die Struktur der Funktionen als Strukturierungsvorlage verwendet werden. Ist die dritte Einstellung aktiviert, werden die hinzugefügten Blätter neu nummeriert. Bei looporientierter Planung sind die Blätter in den meisten Fällen ohnehin pro Stelle von eins beginnend nummeriert, daher ist diese Option in diesem Fall nicht von Bedeutung. Die letzte Einstellung dieser Kategorie stößt eine Aktualisierung der Verbindungen auf dem Blatt an. Es empfiehlt sich, diese erst dann zu starten, wenn die Planung des Projekts einen hohen Reifegrad besitzt. Dieser Vorgang nimmt bei großen Projekten einiges an Zeit in Anspruch und ist zu diesem Zeitpunkt noch nicht zielführend. Die Funktionalität *Aufstellungsorte* wird in dieser Umsetzung nicht verwendet. Dies muss in den Projekteigenschaften aktiviert werden und ist daher im *Typical Manager* ausgegraut. Abschließend stehen noch zwei allgemeine Optionen zur Auswahl. Durch die erste werden im Manager die Varianten *Hinzufügen* und *Entfernen* unter *Aktionen* aktiviert bzw. deaktiviert. Es kann durch den Manager ein Typical

durch ein anderes ersetzt werden. Soll jedoch ein weiteres Typical zu einem bestehenden hinzugefügt werden, ist diese Option zu aktivieren. Durch die letzte Einstellmöglichkeit werden automatische Drähte, die z.B. zum Erstellen eines Klemmenplans notwendig sind, direkt nach dem Kopiervorgang erzeugt. Dies ist ebenfalls ein längerer Vorgang in großen Projekten, daher gilt auch hier die Empfehlung, diese Aktivität manuell zu einem späteren Zeitpunkt anzustoßen.

Sind die Optionen entsprechend vorgenommen, wird die Plangenerierung im *Typical Manager* gestartet. Dieser kopiert sodann die Objekte und Blätter aus dem Typicalprojekt in das Zielprojekt. Abhängig von der Menge an ausgewählten Funktionen dauert dieser Vorgang unterschiedlich lange. Etwaige Probleme im Generiervorgang werden in einer Meldungsdatei gespeichert. Nach Abschluss des Vorganges erscheint ein Fenster mit den unter Umständen aufgetretenen Meldungen.

Für die Umsetzung in *EB* können die unter 8.3.1 erläuterten Typicals ohne weitere Anpassungen verwendet werden.

9.2 Der Schaltplangenerator *G8* für *EPLAN Electric P8*

Für die automatisierte Planerstellung in *EPLAN* kommt der *G8 Schaltplangenerator* der Firma *CAE Expert Group* zum Einsatz. Die prinzipielle Vorgehensweise dieses Tools ist relativ einfach, da es manuelle Vorgänge im Planungsprozess mit *EPLAN* automatisiert durchführt. Die üblichen Schritte sind Seite anlegen, Makro platzieren und spezifische Informationen pro z.B. Messstelle in die Symbole eintragen. Der *G8* führt diese Schritte direkt aus, sucht die Stelleninformationen aus einer speziellen *Microsoft Excel*-Datei heraus und erstellt ein Fenstermakro. Ein solches Makro kann mehrere komplette *EPLAN*-Seiten inkl. Strukturierungsinformationen sowie Schaltungen enthalten, die in *EPLAN* importiert werden können. Durch den Import in *EPLAN* entsteht auf diese Weise die Dokumentation einer elektrotechnischen Anlage, anschließend besteht natürlich die Möglichkeit, weitere manuelle Anpassungen darin vorzunehmen.¹⁰⁰

Für diesen umfangreichen Import ist jedoch einiges an Vorbereitung notwendig, die folgend genauer beschrieben wird.

9.2.1 Gesamt-Typicals in Einzel-Makros aufteilen

Zunächst werden die bereits erstellten Makros für den Generator angepasst. Der *G8* unterstützt die Möglichkeit, Makros mit verschiedenen Varianten zu erstellen. Grundsätzlich bedeutet das, dass wiederverwendbare Einheiten eines Typicals zusammengefasst werden, daraus einzelne Makros erstellt und anschließend variabel miteinander kombiniert werden können. Der Vorteil dieser Vorgangsweise liegt darin, dass Änderungen an weniger Makros vorgenommen werden müssen und mit geringem Aufwand viele unterschiedliche Planvarianten erstellt werden können.

Dazu werden die unter Abschnitt 8.3.2 erstellten Typicals herangezogen und in sinnvolle Einheiten unterteilt. Aus den beiden Messstellentypicals wird der Kanal der analogen Eingangskarte als eigenes Makro erstellt. Gleichmaßen werden die unterschiedlichen Feldgeräte inklusive Kabel, Schirm und Klemmen in

¹⁰⁰ Vgl. Klein/CAE Expert Group (2018), S. 7.

jeweils ein Makro vereinzelt. In Abbildung 40 sind die Einzelteile inkl. der Makrokästen dargestellt. Die Makrobezeichnung lautet *FRAG_SPS_Bit_AE_2AN.ema* für den SPS-Kanal, *FRAG_ANA_MEss.ema* für die Druckmessung und *FRAG_ANA_MEss_Temp.ema* für die Temperaturmessung.

Damit der Generator die Einzelfragmente richtig zusammensetzen kann, ist in jedem Makro ein Referenzpunkt, *Plug*, bzw. ein Einfügepunkt, *Socket*, vorhanden. Ein *Socket* ist die Andockstelle, an die der *Plug* eines anderen Einzelteils gesetzt wird. Da in *EPLAN* die Verbindungen zwischen Symbolen automatisch entstehen, wenn diese richtig zueinander positioniert sind, ist auf die richtige Ausrichtung dieser beiden Funktionen zueinander zu achten. *Plug* und *Socket* sind freie *EPLAN*-Texte, die einer bestimmten Notation folgen müssen. Diese wird in den Einstellungen des *G8* festgelegt und ist standardmäßig für den Referenzpunkt mit *#<Plug:RP>* belegt. Jedes Makro darf nur maximal einen *Plug* besitzen.¹⁰¹

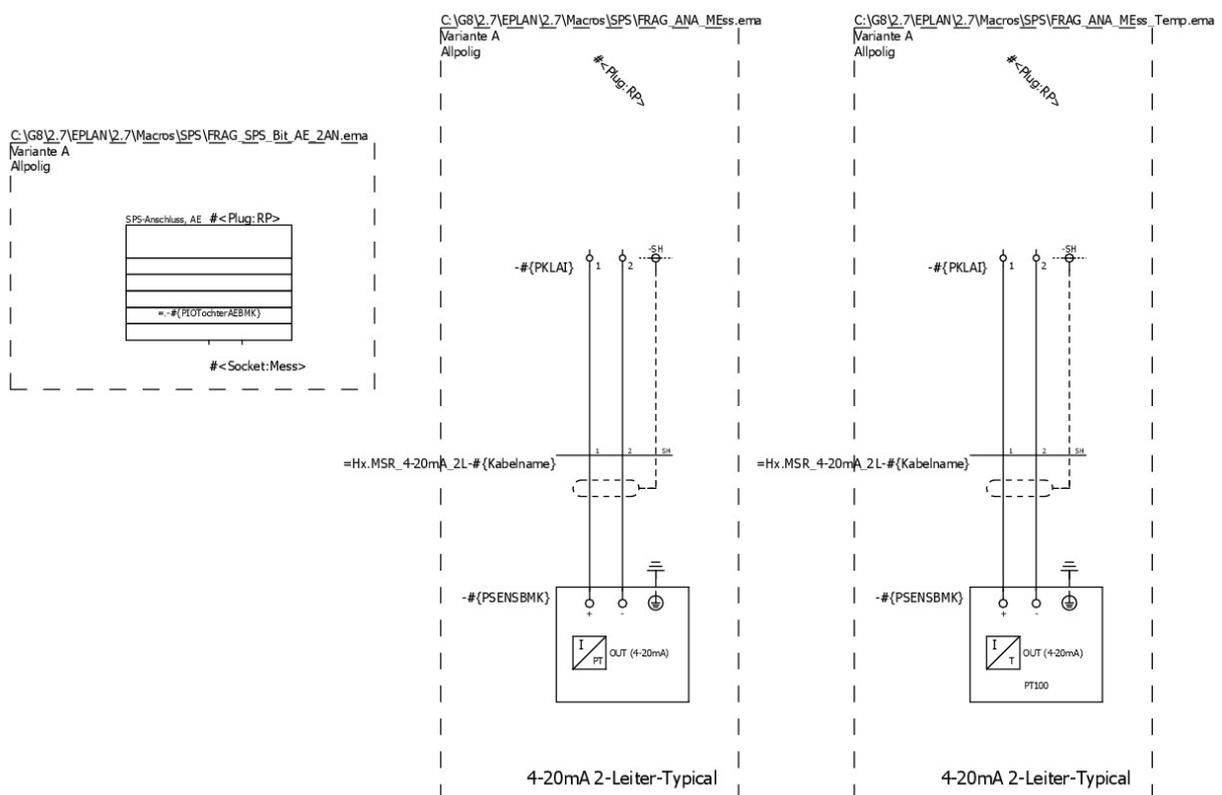


Abbildung 40: Einzel-Makros der Messstellentypicals für den Schaltplangenerator G8, Quelle: Eigene Darstellung

Einfügepunkte sind mehrfach pro Fragment einsetzbar und müssen daher eine eindeutige Kennzeichnung besitzen. Start und Ende der *Sockets* ist ebenso in den Einstellungen festgelegt und ist zu Beginn *#<Socket:*, danach folgt der Parametername, z.B. *Mess*, und endet mit *>*, dieses Beispiel ergibt die vollständige Bezeichnung *#<Socket: Mess>*. Diese Bezeichnung wird später zum Zusammenstellen der Typicals aus den Einzelmakros verwendet.¹⁰²

¹⁰¹ Vgl. Klein/CAE Expert Group (2018), S. 16.

¹⁰² Vgl. Klein/CAE Expert Group (2018), S. 16.

Damit mit den Einzelmakros eine möglichst große Anzahl an Varianten entstehen kann, bietet der G8 an, Werte in Symbolen oder freien Texten durch den Generator zu befüllen. Es handelt sich dabei um die *Parametrierung der Makros*, speziell die *Ersetzungs-Technik*. Nahezu jedes Attribut kann auf diese Weise durch die Vorgaben des Technikers, der Technikerin für alle Messstellen separat befüllt werden. In den Einstellungen ist dazu wieder der eindeutige Bezeichner zu vergeben, der standardmäßig mit #*{* beginnt, gefolgt wird vom Parameternamen und mit einem *}* abschließt.¹⁰³

Später in diesem Abschnitt sind alle verwendeten Parametrierungen übersichtlich pro Typical in einer *Microsoft Excel*-Tabelle aufgelistet.

Die beschriebenen Makros werden durch den G8 auf einer Schaltplanseite platziert. Um die Platzierung auf der Seite festzulegen, wird eine *Basisseite* erstellt. Darauf befinden sich nur *Sockets* und keine *Plugs*. Grundsätzlich handelt es sich hierbei wieder um ein Makro, das vom G8 analysiert und durch die fehlenden *Plugs* als *Basisseite* erkannt wird. Diese Seite wird exakt auf den gleichen Koordinaten im Plan platziert, an denen sie im Makro sitzen und dienen somit als Anhaltspunkt für die Ausrichtung des Typicals.

In Abbildung 41 ist das Makro (*BAS_4SPS.ema*) für die *Basisseite* der Messstellen-Typicals dargestellt. Darin befinden sich nur ein Strukturkasten, der die Örtlichkeit beinhaltet, sowie vier *Sockets*, an denen sich *Plugs* anderer Makros anfügen können.



Abbildung 41: *Basisseite* für das G8-Typical der Messstellen, Quelle: Eigene Darstellung

Das *EPLAN*-Stellstellen-Typical ist auf die gleiche Weise anzupassen, die notwendigen Änderungen werden nun beschrieben.

Das Typical ist in drei Makros aufgeteilt, darunter einmal der digitale Ausgangskanal *FRAG_SPS_Bit_DA_2AN.ema*, zwei digitale Eingangskanäle *FRAG_SPS_Bit_DI1_2AN.ema* und *FRAG_SPS_Bit_DI2_2AN.ema* sowie die elektrische Schaltung inkl. pneumatischer Komponenten *FRAG_DIG_Ventil.ema*. Anhang 6 zeigt die vier Makros, die im Makroprojekt auf einer Seite platziert sind.

¹⁰³ Vgl. Klein/CAE Expert Group (2018), S. 14-15.

Daraus wird ersichtlich, dass die Positionierung der Makros keine Rolle für das spätere Typical spielt, sie ist nur von den *Sockets* der *Basisseite* und den einzelnen Makros abhängig.

In diesen Makros wird wieder mit Makro-Parametern gearbeitet, speziell bei den Betriebsmittelkennzeichen, aber auch bei den Kabel-Artikeln, welche die gesamten Kabel- bzw. Schlauchinformationen beinhalten.

Das *Basisseiten*-Makro des Stellstellen-Typicals ähnelt jenem der Messstellen-Typicals, einzig die Anzahl und Position der *Sockets*, an denen die IO-Kanäle anknüpfen, ist anders. Aus diesem Grund wird es an dieser Stelle nicht grafisch dargestellt, die Makrobezeichnung lautet *BAS_Ventil.ema*.

Abschließend werden die Makros erstellt und als *Fenstermakros* in einem Verzeichnis abgespeichert.

9.2.2 Konfiguration der Typicals in den G8-Stammdaten

Wie zuvor bereits ausgeführt, können die einzelnen Makros beliebig miteinander kombiniert werden, um daraus Typicals zu erhalten. Diese Kombinationen werden in einer *Microsoft-Excel*-Datei mit der Bezeichnung *Stammdaten* zusammengestellt, die als Vorlage bereits im Lieferumfang des G8 enthalten ist. In Abbildung 42 ist der erste Teil dieser *Stammdaten* dargestellt, in dem zuerst allgemeine Angaben und darunter (ab Spalte B7 nach rechts) die einzelnen Bezeichnungen der drei Typical-*Ausprägungen* zu finden sind. In Spalte I befinden sich die zuvor erstellten Makros, die entweder manuell eingetippt oder per Auswahlfenster importiert werden können. Die Makros werden anschließend den jeweiligen Typical-*Ausprägungen* zugewiesen, wobei ein Makro mehrfach verwendet werden kann. Die Zuordnung wird über ein x in der betreffenden Spalte (B für Druckmessung und C für Temperaturmessung) realisiert. So benötigen die beiden Messstellen-Typicals jeweils die *Basisseite*, einen analogen Eingangskanal sowie das passende Feldgerät-Makro (*FRAG_ANA_Mess.ema* für das Druckmessung-Typical und *FRAG_ANA_Mess_Temp.ema* für das Temperaturmessung-Typical).

	A	B	C	D	E	H	I	J
1	Typical Name	Typicals MA						
2	Typical Beschreibung	EPLAN-Typicals						
3	Typical Struktur	analog/digital						
4	Parameter Name	ausführung						
5	Parameter Struktur							
6	Parameter Abhängigkeit							
7	Ausprägung Name	Druck	Temperatur	Ventil			Makros	
8	Ausprägung ID	Druckmessung	Temperaturmessung	Ventil				
9	Ausprägung Plausibilität							
10							Makroname	Makrovariante
11								
12		x	x				SPS\BAS_4SPS.ema	A
13		x	x				SPS\FRAG_SPS_Bit_AE_2AN.ema	A
14		x					SPS\FRAG_ANA_MEss.ema	A
15								
16			x				SPS\FRAG_ANA_MEss_Temp.ema	A
17								
18				x			SPS\BAS_Ventil.ema	A
19				x			SPS\FRAG_SPS_Bit_DA_2AN.ema	A
20				x			SPS\FRAG_SPS_Bit_DI1_2AN.ema	A
21				x			SPS\FRAG_SPS_Bit_DI2_2AN.ema	A
22				x			SPS\FRAG_DIG_Ventil.ema	A

Abbildung 42: G8-Stammdaten mit Darstellung der Kombinationen und den verwendeten Makros, Quelle: Eigene Darstellung

Das Stellstellen-Typical entsteht aus den Makros *BAS_Ventil.ema*, *FRAG_SPS_Bit_DA_2AN.ema*, *FRAG_SPS_Bit_DI1_2AN.ema*, *FRAG_SPS_Bit_DI2_2AN.ema* und *FRAG_DIG_Ventil.ema*, die alle in Spalte D dem Typical *Ventil* zugeordnet sind.

Beschreibung der Generatoren

Anschließend werden die *Stammdaten* mit den Makros synchronisiert, d.h. alle relevanten Informationen für den Schaltplangenerator, wie *Plugs*, *Sockets* oder Parameter, werden aus den Makros ausgelesen und in die *Stammdaten*-Datei geladen.

Abbildung 43 zeigt die in den Makros verwendeten *Plugs* und *Sockets*, die nun zueinander zugeordnet werden. Dazu bekommen jene *Plug-Socket*-Paare, die miteinander verbunden werden sollen, die gleiche Bezeichnung. Der *Plug* des Druck- bzw. Temperaturmakros wird dem *Socket* des IO-Kanal-Makros zugewiesen, in diesem Fall mit der Ziffer 2. Durch die Zuordnung von IO-Kanal-Makro mit der *Basisseite* (über die Ziffer 1 in *Socket SPS_1*) sind Mess- inkl. IO-Kanal-Makro gemeinsam auf der *Basisseite* platziert. Das gleiche Prinzip wird beim Stellenstellen-Typical verwendet, jedoch unter Zuhilfenahme von Buchstaben anstelle von Ziffern, zum Zweck einer besseren Übersichtlichkeit.

	I	J	K	L	M	N	O	P	Q	R
7	Makros			Plug / Socket						
8										
9				Referenzpunkt	Socket	Socket	Socket	Socket	Socket	Socket
10	Makroname	Makrovariante	RP	SPS_1	SPS_2	SPS_3	SPS_4	Mess	Ventil	
11										
12	SPS\BAS_4SPS.ema	A		1						
13	SPS\FRAG_SPS_Bit_AE_2AN.ema	A	1					2		
14	SPS\FRAG_ANA_MEss.ema	A	2							
15										
16	SPS\FRAG_ANA_MEss_Temp.ema	A	2							
17										
18	SPS\BAS_Ventil.ema	A		A	B	C				
19	SPS\FRAG_SPS_Bit_DA_2AN.ema	A	A							D
20	SPS\FRAG_SPS_Bit_DI1_2AN.ema	A	B							
21	SPS\FRAG_SPS_Bit_DI2_2AN.ema	A	C							
22	SPS\FRAG_DIG_Ventil.ema	A	D							

Abbildung 43: G8-Stammdaten Makros mit zugeordneten *Plug-Socket*-Abhängigkeiten, Quelle: Eigene Darstellung

Die Makro-Parameter werden durch das Synchronisieren in die Datei geladen und spaltenweise aufgelistet. Die graue Markierung in der jeweiligen Makro-Zeile weist darauf hin, dass dieser Parameter in dem zugehörigen Makro verwendet wird.

Abbildung 44 ist eine Übersichtsdarstellung aller Parameter, die unter Anhang 7 aufgelistet und kurz beschrieben sind.

	I	J	S	T	U	V	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
7	Makros		Parameter																			
8																						
9	Makroname	Makrovariante	Parameter Kabel-Artikel	Parameter Kabelname	Parameter Schlauch-Artikel	Parameter Schlauchname	Parameter Einbauort	Parameter PLOTschneidKana	Parameter PLOTschneidAEBM	Parameter PSENSEB	Parameter PKLN	Parameter PLOTschneidKana	Parameter PLOTschneidOEB	Parameter PLOTschneidKana	Parameter PLOTschneidOEB	Parameter PLOTschneidKana	Parameter PLOTschneidOEB	Parameter PKLDO	Parameter PKLDI	Parameter PKLEBK	Parameter PENLEB	Parameter PENLEB
12	SPS\BAS_4SPS.ema	A																				
13	SPS\FRAG_SPS_Bit_AE_2AN.ema	A																				
14	SPS\FRAG_ANA_MEss.ema	A																				
16	SPS\FRAG_ANA_MEss_Temp.ema	A																				
18	SPS\BAS_Ventil.ema	A																				
19	SPS\FRAG_SPS_Bit_DA_2AN.ema	A																				
20	SPS\FRAG_SPS_Bit_DI1_2AN.ema	A																				
21	SPS\FRAG_SPS_Bit_DI2_2AN.ema	A																				
22	SPS\FRAG_DIG_Ventil.ema	A																				

Abbildung 44: G8-Stammdaten-Übersichtsdarstellung der Makro-Parameter, Quelle: Eigene Darstellung

Durch die stellenorientierte Darstellung der Pläne werden die Seiten pro Stelle von eins startend nummeriert. Bei den verwendeten Typicals ist jeweils nur ein Plan vorhanden, daher kann bereits in den Stammdaten die Seitennummer 1 festgelegt werden.

Um die Seitennummern zuzuweisen, wird diese Information den Basismakros vergeben, da durch sie die Seiten im *Seitenmakro* erstellt werden. Wie in Abbildung 45 zu sehen ist, wird die Seitenbeschreibung nicht

in den Stammdaten festgelegt, da diese für die einzelnen Stellenpläne aus der Stellenbezeichnung gebildet wird.

	I	J	AN	AO	AP
7	Makros			Seite	
8					
9					
10	Makroname	Makrovariante		SeiteName	Seitenbeschreibung
11					
12	SPS\BAS_4SPS.ema	A		1	
13	SPS\FRAG_SPS_Bit_AE_2AN.ema	A			
14	SPS\FRAG_ANA_MEss.ema	A			
15					
16	SPS\FRAG_ANA_MEss_Temp.ema	A			
17					
18	SPS\BAS_Ventil.ema	A		1	
19	SPS\FRAG_SPS_Bit_DA_2AN.ema	A			
20	SPS\FRAG_SPS_Bit_DI1_2AN.ema	A			
21	SPS\FRAG_SPS_Bit_DI2_2AN.ema	A			
22	SPS\FRAG_DIG_Ventil.ema	A			

Abbildung 45: G8-Stammdaten-Seitenkennzeichnung, Quelle: Eigene Darstellung

9.2.3 Die Engineering- und Makro-Liste der Projektdaten

Diese Schritte bzw. Festlegungen in den Stammdaten sind notwendig, um die Basis für den eigentlichen Generierungsprozess zu legen. Der *G8-Schaltplangenerator* sieht dazu eine weitere *Microsoft-Excel*-Datei vor, worin die spezifischen Daten der geplanten Umsetzung eingepflegt sind. Die Datei wird unter der Bezeichnung *Projektdaten* geführt und ist der Startpunkt, an dem der Generator die Seitenmakros für das *EPLAN*-Projekt erstellt. Der *G8* ist in der Lage, eine große Vielfalt an Plänen für *EPLAN* zu generieren, daher befinden sich in der Projektdaten-Beispieldatei einige Tabellen mit Vorlagen für die Schaltplangenerierung. Für die vorliegende Arbeit werden zwei davon verwendet, die *Engineering_FKT_Manuell* (folgend als *Engineering-Liste* bezeichnet), die Projektdaten enthält, und die *Makro-Liste*, in der sich die Generierungsdaten befinden.

Abbildung 46 zeigt die leere Engineering-Liste mit den Spalten, die mit den notwendigen Informationen gefüllt werden. Diese Liste ist grundsätzlich frei von Formatierungsvorschriften, einzig in Zeile 10 sind die exakten Bezeichnungen eingetragen, die auch in der Makro-Liste existieren und die Basis für eine spätere Generierung sind. Die Anordnung der Spalten selbst ist frei gewählt, jedoch an die Vorlage-Datei angelehnt. Die Liste wird entweder händisch oder mit Hilfe von diversen *Microsoft-Excel*-Mechanismen befüllt. Eine Zeile in der Engineering-Liste steht in diesem Fall für eine Stelle der Anlage, respektive für einen Stellenplan in *EPLAN*. In Spalte C befindet sich die Seitenbeschreibung des Planes, unter dem Strukturkennzeichen in Spalte K wird die *KKS*-Bezeichnung der Stelle eingetragen und unter *Typical* in Spalte F steht die *Typical*-Bezeichnung. Diese wird aus den zuvor beschriebenen Stammdaten mittels Auswahlfenster ausgewählt, oder händisch eingetragen. Danach folgen die Parameter der Makros, die durch den Generator befüllt werden sollen. Zur besseren Lesbarkeit sind nicht alle davon in der Darstellung sichtbar, in der im Anhang befindlichen Tabelle selbst sind jedoch alle Parameter vorhanden.

	C	E	F	G	K	L	P	Q	S
7	Seiten		Typical		Strukturkennzeichen		Parameter		
8	Seitenbeschreibung		Typical	Typicalname	Strukturkennzeichen	Parameter	Sensor BMK	Einbauort	Analogtochter Kanal
9	Seitenbeschreibung		TYPICAL_ECAD		=	PSENSBMK	Einbauort	PIOTochterAEKanal	
10	Seitenbeschreibung		TYPICAL_ECAD		=	PSENSBMK	Einbauort	PIOTochterAEKanal	
11									
12									
13									
14									

Abbildung 46: G8-Projektdatei, Aufteilung der Engineering-Liste in Seiten-, Typical-, Strukturkennzeichen- und Parameter-Sektionen, Quelle: Eigene Darstellung

Mit Hilfe von G8-Funktionen in *Microsoft-Excel* werden die Daten aus der Engineering-Liste in die Makro-Liste übernommen. Es wird vom Autor empfohlen, diese Tabelle auf jene Spalten zu reduzieren, die in Abbildung 47 dargestellt sind. Durch das *Erzeugen* der Liste aus der Engineering-Liste werden die Daten für den Plangenerator geladen. Die Quelle, aus der diese kopiert werden, hängt von der Auswahl in Zeile 8 ab. Mit *copyFromSource* werden die Stellen-Informationen aus der Engineering-Liste übernommen und durch *copyFromClass* aus den Stammdaten mitkopiert. Der *SeiteName* ist hier als Beispiel geeignet, da dieser durch die stellenorientierte Planerstellung bereits in den Stammdaten mit 1 fixiert wurde.

Der G8 durchläuft während des Erstellens der Makro-Liste die einzelnen Zeilen der Engineering-Liste. Findet dieser den Typicalnamen, listet die G8-Funktion die einzelnen Makros des Typicals auf und kopiert die Beschreibung sowie das Strukturkennzeichen in die Makro-Liste.

	A	G	I	O	P	Q	R	S
1	Makroliste							
7								
8	copyFromSource		copyFromClass	copyFromSource	copyFromClass		copyFromClass	
9								
10	Querverweise	Seitendaten		SeiteName	Seitenbeschreibung	Makros	Makroname	Makrovariante
11								
12								

Abbildung 47: G8-Projektdatei, Makro-Liste vor dem ersten Erzeugen aus der Engineering-Liste, Quelle: Eigene Darstellung

Anschließend wird diese Liste durch eine G8-Funktionalität synchronisiert, dabei lädt der Generator alle *Plugs*, *Sockets* und *Parameter* und legt diese als Spalten an.

Abbildung 48 zeigt die geladenen Inhalte der Makros, aufgeteilt in die Kategorien Plug/Socket und Parameter. Das Kopierverhalten jeder Spalte ist festzulegen, wobei die Informationen zu den Einfüge- und Referenzpunkten aus den Stammdaten kopiert werden, da diese dort bereits festgelegt wurden. Alle Parameter werden aus der Engineering-Liste geladen.

	T	U	V	W	X	Y	Z	AA	AB	AC
	copyFromClass		copyFromSource							
	Referenzpunkt	Socket		Parameter						
Plug / Socket		RP	SPS_1	SPS_2	SPS_3	SPS_4	Mess	Ventil	Parameter	Einbauort

Abbildung 48: G8-Projektdatei, synchronisierte Inhalte aus den Makros, Quelle: Eigene Darstellung

Im vorletzten Schritt muss die Anwenderin/ der Anwender erneut die Makro-Liste (gleiche Vorgehensweise wie zuvor) aus der Engineering-Liste erzeugen. Da nun bereits die Spalten aller *Plugs*, *Sockets* und *Parameter* in der Makro-Liste vorhanden sind, lädt der Generator die Werte aus den unterschiedlichen Listen direkt in die Makro-Liste, die somit bereit zur Planerstellung ist. Im weiteren Verlauf dieser Ausarbeitung ist eine vollständig befüllte Liste enthalten.

Der finale Schritt ist, dass der *G8* ein *EPLAN*-Seitenmakro generiert, in dem die zusammengestellten Makros mit allen Informationen gespeichert werden, die anschließend ins *EPLAN* importiert werden.

9.3 Erkenntnisse aus den Generatoren

Nachdem beide Generatoren intensiv untersucht wurden, stellt sich heraus, dass beide in ihren Umgebungen geeignet sind, um Pläne bzw. Objekte zu vervielfältigen. Bedingt durch die Unterschiede der Planungsplattformen ist der *Typical Manager* in *EB* in der Lage, Datenbankobjekte anzulegen und diese z.B. mit bereits bestehenden Funktionen zu verbinden. Der *G8* kann eine große Anzahl an möglichen Planvarianten in *EPLAN* mit kleinen Makro-Einheiten erstellen, die einfach gewartet und miteinander kombiniert werden können. Während der *Typical Manager* beim Erstellen von Stellenplänen jeglicher Art zum Einsatz kommt, ist der *G8* darüber hinaus auch für allgemeine Stromlaufpläne einsetzbar. Grundsätzlich kann festgehalten werden, dass der *Typical Manager* eine genaue Ablage der Objekte verlangt und die Vorbereitungsarbeiten beim Erstellen der *Typicals* liegen. Die Generierung selbst funktioniert über den Assistenten einfach. Beim *G8* sind mehrere Schritte notwendig, um von den Einzel-Makros zu einem fertigen Plan zu gelangen. Zu Beginn werden die Makros mit den *Plugs* und *Sockets* sowie den Parametern erstellt, die in den Stammdaten zu *Typicals* kombiniert werden. Danach wird die Engineering-Liste aufbereitet, aus der schlussendlich eine Makro-Liste erzeugt wird, welche die Grundlage für das *EPLAN*-Seitenmakro ist. Dieser Vorgang wird vom *G8* mit einigen *Microsoft-Excel*-Funktionen unterstützt und schließt mit der Generierung der Pläne aus der Liste ab.

Im Zuge der Betrachtung wird festgestellt, dass sich beide Generatoren für den geplanten Workflow eignen.

10 AUSLESEN DER MODELLINFORMATIONEN MIT *VISUAL BASIC FOR APPLICATIONS IN MICROSOFT EXCEL*

Die Transformation des Anlagenmodells, der plattformunabhängigen Schicht, zu den Endplattformen benötigt einen Zwischenschritt, in dem die Informationen aus dem Modell ausgelesen werden. Um dem modellorientierten Ansatz gerecht zu werden, wird ein weitestgehend automatisierter Übergang zwischen dem Modell und der Engineering-Umgebung angestrebt. In diesem Kapitel werden die notwendigen Schritte zur Beschickung der Generatoren beschrieben.

10.1 Das zugrundeliegende Datenformat *GraphML*

Das Metamodell sowie das Anlagemodell sind mit dem Visualisierungsprogramm *yEd* der Firma *yWorks* erstellt. Damit können unterschiedliche Arten von Diagrammen entstehen, darunter auch UML-Diagramme wie bei den erstellten Modellen. Diese Umgebung bietet einige Dateiformate, aus denen ausgewählt werden kann, darunter auch *GraphML*. Es ist ein umfangreiches Dateiformat für Graphen, das von mehreren Plattformen unterstützt wird und ist somit zum Austausch des Modells geeignet. *GraphML* verwendet keine eigens entwickelte Syntax, sondern basiert auf *XML*, der *Extensible Markup Language*.¹⁰⁴

10.1.1 Kurzbeschreibung XML

XML ist eine erweiterbare Auszeichnungssprache, die gleichermaßen für Mensch und Maschine lesbar ist. Durch den einfachen Aufbau ist sie gut für spezielle Anwendungen verschiedener Disziplinen adaptierbar. Da XML oft als Austauschformat genutzt wird, können Programme auf die Inhalte der Dateien zugreifen, diese lesen, verändern oder erweitern. Damit eine Anwendung Daten auslesen kann, sind diese von sogenannten *markups*, also Markierungen, umgeben. Eine Einheit von markup und Daten nennt sich *element*. XML selbst definiert die Bezeichnung der markups nicht, das passiert durch eine Meta-Markup-Sprache. Diese kann selbst entwickelt oder auf eine bereits bestehende Sprache zurückgegriffen werden. Der Vorteil dabei ist, dass die Struktur der Daten und Elemente frei wählbar ist und auf die notwendigen Gegebenheiten angepasst werden kann. Trotzdem sind durch XML einige Regeln festgelegt, wie z.B. die Anordnung der *tags*, die Trennzeichen zwischen den Elementen oder wie Attribute an ein *element* angefügt werden. Durch die Verschachtelung von Elementen werden Zugehörigkeiten und Strukturen in diesem Format erreicht, die auch einfach durch den Menschen innerhalb einer Datei identifiziert werden können.¹⁰⁵

10.1.2 Aufbau des *GraphML*-Formats

Ein *GraphML*-File ist in zwei Teile unterteilt, dem *Header* und dem Graphen. Die erste Zeile bildet immer den Header, der Informationen zur verwendeten XML-Version und der Zeichenkodierung, die im Dokument verwendet wird, enthält. Danach folgt das Wurzelement *<graphml>*, in dem die verwendeten *Name-*

¹⁰⁴ Vgl. *GraphML Team* (2007), Online-Quelle [15.11.2019]

¹⁰⁵ Vgl. *Harold/Means* (2004), S. 3-6.

spaces eingetragen sind. Sie werden verwendet, um die Bezeichnung von Elementen und Attributen eindeutig zuordnen zu können, wenn innerhalb eines XML-Dokuments mehrere solcher Namespaces existieren. Als Beispiel kann ein Element herangezogen werden, das die Bezeichnung `<koerper>` trägt. Es kann in einem medizinischen Kontext als menschlicher Körper oder in einem geometrischen Kontext als Form eines runden Objekts interpretiert werden. Anschließend folgen die Informationen, die den Graphen darstellen. Dazu wird das Element `<graph>` verwendet, das aus beliebig vielen Unterelementen vom Typ `<node>` und `<edge>` bestehen kann. Eine `<node>` stellt dabei einen Knoten dar, der mit einer eindeutigen ID im GraphML-File gekennzeichnet sein muss. Eine `<edge>` ist in diesem Zusammenhang eine Verbindung zwischen zwei Knoten. Jede davon hat wiederum eine eindeutige ID und enthält die Information bezüglich Quelle und Ziel, das über die Attribute `source` und `target` erkennbar ist. Um einem Graphen zusätzliche Informationen mitzugeben, existieren sogenannte *GraphML-Attribute*, die nicht mit den XML-Attributen verwechselt werden dürfen. Sie dienen dazu, dem Graphen strukturierten Inhalt anzufügen und verwenden die *key/data*-Erweiterung von GraphML.¹⁰⁶

In Abbildung 49 sind die zuvor beschriebenen GraphML-Spezifikationen anhand eines einfachen Beispiels dargestellt, in dem die beiden Knoten `n0` und `n1` über die Verbindung `e0` miteinander verbunden sind.



Abbildung 49: Darstellung eines simplen Graphen als GraphML-File (links) und in grafischer Darstellung (rechts), Quelle: Eigene Darstellung

10.2 Visual Basic for Applications, VBA

VBA ist die Erweiterung der Programmiersprache *Visual Basic*, die speziell für den Einsatz mit den Programmen der *Microsoft Office* Reihe, z.B. *Word* oder *Excel*, adaptiert ist. Sie wird in diesen Tools meistens eingesetzt, um manuelle Vorgänge zu verbessern bzw. automatisch abzuarbeiten. Oft ergibt sich erst durch eine VBA-Lösung ein möglicher Workflow, um bestimmte Aufgaben zu bewältigen. Eine solche Anwendung wird als *Makro* bezeichnet und kann z.B. Berechnungen durchführen, Zellen formatieren, Spalten mit Werten

¹⁰⁶ Vgl. Brandes/Eiglsperger/Lerner (o.J.), Online-Quelle [15.11.2019]

befüllen oder Informationen aus externen Quellen lesen. Diese Programmiersprache bietet einige Objekte, Eigenschaften und Methoden an, mit denen auf die Bestandteile der *Microsoft Office*-Programme zugegriffen wird.¹⁰⁷

Die Programmiersprache VBA wird für diese Umsetzung verwendet, da die Plangeneratoren auf *Microsoft Excel* basieren bzw. die Engineering-Tools Standard-Schnittstellen zu diesem Programm anbieten. Es wird in dieser Arbeit nicht näher auf VBA eingegangen, da diese Sprache als Hilfsmittel zum Zweck der Umsetzung eingesetzt wird und der Fokus der Arbeit auf dem Workflow liegt.

10.3 *Microsoft Excel/VBA*-Anwendung zur Transformation vom plattformunabhängigen Modell in die Engineering-Umgebungen

Zur Überführung des plattformunabhängigen Modells in die beiden Engineering-Plattformen steht eine *Microsoft Excel*-Datei zur Verfügung, in der ein *Makro* implementiert ist, welches das Modell ausliest und die Daten für den *EB*-Objektimport bzw. die *G8*-Engineering-Liste aufbereitet. Dazu sind zwei Buttons, dargestellt in Abbildung 50, in einer Tabelle platziert, die entweder die Aufbereitung für *EB* oder *EPLAN* durchführt.

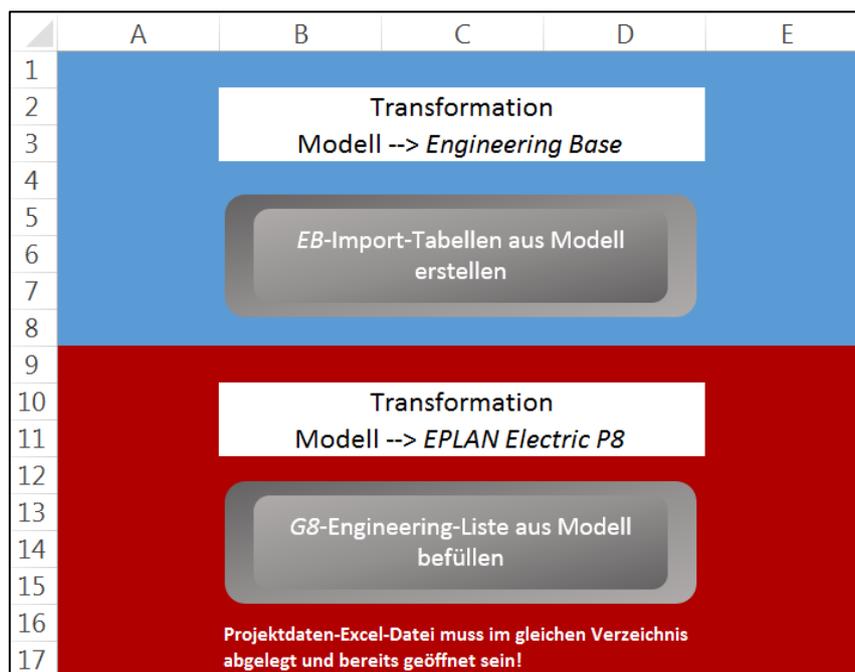


Abbildung 50: Buttons für die Transformationsvorbereitung nach *EB* bzw. *EPLAN*, die zum Start der *Microsoft-Excel*-Makros verwendet werden, Quelle: Eigene Darstellung

Wird einer der beiden Buttons gedrückt, öffnet sich ein *openDialog* mit der Dateityp-Voreinstellung *GRAPHML File (*.graphml)*. Das Modell muss zuvor in diesem Dateiformat abgespeichert werden, damit das *Microsoft Excel*-Makro darauf zugreifen kann. Ist die gewünschte Datei ausgewählt und wird der Dialog mit *OK* bestätigt, startet das jeweilige *Makro* mit dem Auslesen und Aufbereiten der Informationen für die Engineering-Plattformen.

¹⁰⁷ Vgl. Theis (o.J.), Online-Quelle [15.11.2019]

10.3.1 Sub-Prozedur *GenerateForEB()*

Durch den oberen Button wird die *Sub-Prozedur GenerateForEB()* gestartet. Zu Beginn wird der *openDialog* aufgerufen und der erwähnte Dateityp-Filter eingestellt. Nach erfolgter Auswahl und Bestätigung wird der Dateipfad in eine Variable gespeichert. Anschließend wird ein Objekt *xDoc* vom Typ *MSXML2.DOMDocument* erstellt. *MSXML* ist ein Parser, der von *Microsoft Office* Produkten verwendet wird, um Informationen aus einem XML-Format richtig zu interpretieren und in weiterer Folge verarbeiten zu können. Ein *DOMDocument* repräsentiert in diesem Kontext ein gesamtes XML-File. In weiterer Folge lädt der Parser die Datei in das Objekt *xDoc*, in dem der Funktion *Load* der Dateipfad übergeben wird. Danach werden die Namespaces dem Objekt als Eigenschaften übergeben, damit ein komfortableres Auslesen der Knoten möglich ist. Als nächstes wird eine Variable vom Typ *IXMLDOMNodeList* deklariert, worin eine Liste von XML-nodes geladen werden kann. Dies wird gleich darauf durchgeführt, indem die Funktion *SelectNodes* des *xDoc*-Objekts aufgerufen wird. Mithilfe dieser Funktion können alle Knoten einer bestimmten Ebene ausgewählt und, wie in diesem Fall, in eine *nodeList* geschrieben werden. Damit die richtige Ebene der XML-Struktur ausgelesen wird, erhält die Funktion als Übergabeparameter den Pfad zu den *nodes*, wie in Quelltext 1 ersichtlich ist.

```

If .Show = True Then
    XMLFileName = .SelectedItems(1)

    Dim xDoc As Object
    Set xDoc = CreateObject("MSXML2.DOMDocument")
    xDoc.async = False
    xDoc.validateOnParse = False
    xDoc.Load (XMLFileName)

    'Namespaces hinzufügen
    xDoc.setProperty "SelectionNamespaces", _
    |"xmlns:a='http://graphml.graphdrawing.org/xmlns'   xmlns:y='http://www.yworks.com/xml/graphml'"

    'Variablen deklarieren
    Dim nodes As MSXML2.IXMLDOMNodeList

    'NodeList auslesen
    Set nodes = xDoc.SelectNodes("/a:graphml/a:graph/a:node")
    MSRListe nodes, xDoc
    SPSImportListe nodes, xDoc
    BaueinheitenListe nodes, xDoc
    KabelImportListeMessstelle nodes, xDoc
    KabelImportListeStellstelle nodes, xDoc
    FeldgeraetImportListeMessstelle nodes, xDoc
    FeldgeraetImportListeStellstelle nodes, xDoc
    KlemmleisteImportListeMessstelle nodes, xDoc
    KlemmleisteImportListeStellstelle nodes, xDoc

End If

```

Quelltext 1: Auszug aus der Prozedur, die durch den *EB*-Button aufgerufen wird, Quelle: Eigene Darstellung

Anschließend werden die erstellten Sub-Prozeduren aufgerufen, die jeweils als Übergabeparameter die *nodeList* und das *xDoc*-Objekt erhalten. Mithilfe dieser Prozeduren wird jeweils eine eigene Tabelle in der *Microsoft Excel*-Datei erstellt und mit den jeweiligen Informationen befüllt, die anschließend als Objekte ins *EB* importiert werden. Diese werden jeweils in einem eigenen Unterkapitel beschrieben.

10.3.1.1 Sub-Prozedur: *MSRListe*

Diese Tabelle enthält alle Stellen, die sich im Modell befinden und in *EB* vor dem Generierungsprozess angelegt werden müssen. Die Informationen darin sind die KKS-Nummer, der Langtext, das HW-Typical, die beiden benötigten *Typical Baueinheiten* und der Typ der Stelle (Mess- oder Stellstelle).

Nach dem Aufrufen wird geprüft, ob die Tabelle in *Microsoft Excel* bereits vorhanden ist; wenn nicht, wird diese neu angelegt. Danach werden die oben beschriebenen Attribute als Spaltenüberschriften in die erste Zeile eingetragen, bevor in eine *for-each*-Schleife gesprungen wird, in der alle *nodes* der übergebenen *nodeList* durchlaufen werden. Als erstes wird das Attribut *Stereotyp* aus der *node* ausgelesen und abgefragt, ob es sich um eine Stelle vom Typ *Druck*, *Temperatur*, *Dichte*, *Niveau* oder *Stellstelle* handelt. Andere Nodes werden in der *MSRListe* nicht benötigt. Zuvor wird ein Zähler initialisiert, der zur Ausgabe der Informationen in der richtigen Zeile der *MSRListe* notwendig ist. Als erstes wird die Nummer der Stelle selbst ausgegeben, dann durch die *edge* zur Teilanlage navigiert und dessen Nummer ausgegeben und schließlich die Nummer der Anlage selbst. Quelltext 2 zeigt das Navigieren und die Ausgabe als Codeausschnitt. Diese Dreiteilung ist für *EB* notwendig, um die richtige Strukturierung der Funktionen zu erhalten.

```
'Teilanlage suchen
nodeIdTeilanlage = getNodeByString(node, "Teilanlage", xmlDoc)
Set helpNodes = xmlDoc.SelectNodes("/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdTeilanlage & Chr$(34) & "]")
Cells(c, 2) = getNodeName(helpNodes.Item(0))
```

Quelltext 2: Suchvorgang zwischen zwei *nodes*, die durch eine *edge* verbunden sind, Quelle: Eigene Darstellung

Das Navigieren zwischen zwei *nodes* wird über eine eigens erstellte Funktion *getNodeByString* gelöst, die als Übergabeparameter die aktuelle *node*, den Stereotyp der gesuchten *node* und das gesamte XML-Dokument enthält. Als Rückgabewert wird die ID der gesuchten *node* geliefert. Eine nähere Beschreibung dieser Funktion befindet sich unter Kapitel 10.3.3. Mit der ID wird im gesamten XML-File nach dieser *node* gesucht, dessen *name* anschließend ausgegeben wird.

Auf die gleiche Weise wird, vom Feldgerät ausgehend, nach dessen örtlicher Zuordnung gesucht, die ebenso in die Tabelle geschrieben wird (notwendig für *Typical Baueinheit 1*)

Um die Zuordnung der Stelle zum Schaltschrank zu erhalten, wird eine umfangreiche Navigation gestartet, die über fünf Objekte des Modells läuft. Gestartet wird von der Stelle zum verbundenen Signal, weiter zur IO-Karte, der Anschaltung und über das Rack zur verbundenen Örtlichkeit.

Zuletzt wird für den *EB*-Import eine Klassifizierung zwischen *Messstelle* und *Stellstelle* benötigt. Diese Unterscheidung wird mit einem *select-case*-Statement realisiert, in dem aufgrund des Stereotypen der Stelle zwischen diesen beiden Varianten gewählt wird.

10.3.1.2 Sub-Prozedur: *SPSImportListe*

In dieser Tabelle werden die SPS-IO-Komponenten für den *EB*-Import aufgelistet. Zu Beginn wird wieder nach einer bestehenden Tabelle gesucht, bei erfolgreicher Suche wird der Inhalt gelöscht, andernfalls eine neue erstellt. Anschließend werden die Überschriften zu den Attributen *Teil von* (örtliche Zuordnung), *BMK*, *Typ*, *Materialnummer*, *Adressbereich Eingänge*, *Adressbereich Ausgänge* sowie die *EB-Katalognummer* in die Tabelle geschrieben. Danach wird wieder jede *node* durchlaufen und auf die Stereotypen *analoge Eingangskarte*, *analoge Ausgangskarte*, *digitale Eingangskarte* und *digitale Ausgangskarte* abgefragt. Wird

eine solche *node* gefunden, wird das BMK in die dafür vorgesehene Spalte geschrieben und dessen Attribute ausgelesen. Diese sind in der *node* innerhalb einer sogenannten *ChildNode*, also in einer *node*, die einer anderen untergeordnet ist, eingetragen und überdies noch durch einen Zeilenumbruch getrennt. Aus diesem Grund wird das ausgelesene Gesamtattribut in Einzelzeile unterteilt und in weiterer Folge erneut aufgespalten, da jede Zeile aus der Attributsbezeichnung und dessen Wert besteht. In Quelltext 3 sind im linken Teil die Attribute im XML-File zu sehen und rechts der notwendige Code, um diese adäquat für die Tabelle auszulesen. Zuerst wird der gesamte *String* in einzelne Zeilen aufgeteilt, die wiederum nach dem ersten Leerzeichen getrennt werden; der zweite Teil dieser Trennung entspricht dem Attributwert.

<pre><y:AttributeLabel xml:space="preserve">Adressbereich: IW200 IO-Anzahl: 8 IO-Typ: Eingang Mat.Nr.: 6ES7 331-7KF01-0AB0 </y:AttributeLabel></pre>	<pre>'Attribute aus Karte auslesen ioTyp = nodeChild.ChildNodes.Item(4).ChildNodes.Item(0).Text split_vbLf = Split(ioTyp, vbLf) split_space = Split(split_vbLf(2), " ") Cells(c, 3) = split_space(1) If StrComp("Eingang", split_space(1)) = 0 Then split_space = Split(split_vbLf(0), " ") Cells(c, 5) = split_space(1) Cells(c, 6) = "-" ElseIf StrComp("Ausgang", split_space(1)) = 0 Then split_space = Split(split_vbLf(0), " ") Cells(c, 5) = "-" Cells(c, 6) = split_space(1) End If</pre>
--	--

Quelltext 3: Auslesen der Attributwerte, Links: Attribute in XML-Format, Rechts: Code zum Auslesen, Quelle: Eigene Darstellung

Danach wird aufgrund der *Mat.-Nr.* in einem *select-case*-Statement die spezifische *EB*-Katalognummer eruiert, die entscheidend beim Anlegen des Objekts in *EB* ist.

Der letzte Schritt in dieser Sub-Prozedur ist, die örtliche Zuordnung der IO-Karte zum Schaltschrank zu ergründen. Dies wird wieder über die Navigation von IO-Karte zu Anschaltung, zum Rack und schlussendlich zur Örtlichkeit realisiert.

10.3.1.3 Sub-Prozedur: *BaueinheitenListe*

Durch diese relativ kurze Sub-Prozedur werden die im Modell vorhandenen Örtlichkeiten aufgelistet, um als *Baueinheiten* ins *EB* importiert zu werden. Dazu werden, nach dem Suchen bzw. neu Anlegen der Tabelle, die Spaltenüberschriften *zugeordnete Funktion*, *Schrankbezeichnung* und *Beschreibung* eingetragen. Anschließend werden alle *nodes* des XML-Dokuments durchsucht und auf den Stereotyp *Örtlichkeit* abgefragt. Wird eine solche gefunden, schreibt das Makro zuerst die Schrankbezeichnung in die Tabelle, bevor der Wert des Attributs *Beschreibung* ausgelesen wird. Am Ende wird die verbundene Anlage gesucht, wobei in diesem Fall möglich ist, dass keine gefunden wird. Tritt dies ein, wird die entsprechende Zelle ohne Wert beschrieben, wie in Quelltext 4 ersichtlich ist.

```
'Funktionelle Zuordnung Schaltschrank
nodeIdAnlage = getNodeByString(node, "Anlage", xmlDoc)
Set helpNodes = xmlDoc.SelectNodes( _
"/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdAnlage & Chr$(34) & "]"")
If helpNodes.Item(0) Is Nothing Then
    Cells(c, 1) = ""
Else
    Cells(c, 1) = getNodeName(helpNodes.Item(0))
End If
```

Quelltext 4: Navigation zur funktionellen Zuordnung des Schaltschranks inkl. Empty-Abfrage, Quelle: Eigene Darstellung

10.3.1.4 Sub-Prozeduren: *KabelImportListeMessstelle / KabelImportListeStellstelle*

Diese beiden Sub-Prozeduren wurden im ersten Anlauf in einer gemeinsamen Prozedur abgearbeitet, jedoch durch die Importversuche in *EB* wurde klar, dass es zwei solcher Prozeduren benötigt. Der Grund liegt bei *EB*, da Kabel, die einer Funktion zugeordnet sind, einen Importlauf für den Funktionstyp *Messstelle* und einen für den Typ *Stellstelle* benötigen. Die Prozeduren sind nahezu ident und werden in diesem Unterkapitel gemeinsam beschrieben, deren Unterschiede werden beim Auftreten erläutert.

Nachdem wiederum auf Bestehen der jeweiligen Tabellen geprüft wurde, werden die Importattribute *Anlage*, *Teilanlage*, *Messstelle* bzw. *Stellstelle*, *Kabel-BMK*, *Kabeltyp-EB-Katalognummer*, *Länge [m]*, *Typical Copy Merge Index* und *Örtliche Zuordnung* als Spaltenüberschriften vergeben. Danach werden alle *nodes* durchlaufen und auf die Stereotypen *Kabel* und *Luftschlauch* abgefragt. Anschließend wird geprüft, ob die verbundenen Modellobjekte bei der Messstellen-Prozedur vom Stereotyp *Druck*, *Temperatur*, *Dichte* und *Niveau* bzw. bei der Stellstellen-Prozedur vom Stereotyp *Stellstelle* sind. Sollte dies nicht der Fall sein, wird die weitere Bearbeitung dieser *node* abgebrochen. Wird die Abfrage erfolgreich passiert, schreibt das *Makro* sogleich das Kabel-BMK in die Tabelle und liest die Attribute aus dem Kabelobjekt aus. Gleich wie unter 10.3.1.2 beschrieben, werden diese aufgeteilt und anschließend, abhängig vom Stereotypen des Kabels, zur *EB-Kabel-Katalognummer* zusammengesetzt, wie in Quelltext 5 ersichtlich.

```
'Attribute aus Kabel auslesen
split_vbLf = Split(nodeChild.ChildNodes.Item(4).ChildNodes.Item(0).Text, vbLf)
split_space = Split(split_vbLf(3), " ")
Cells(c, 6) = split_space(1)

aderanzahl_array = Split(split_vbLf(0), " ")
aderanzahl = aderanzahl_array(1)
typ_array = Split(split_vbLf(1), " ")
kabeltyp = typ_array(1)
qs_array = Split(split_vbLf(2), " ")
querschnitt = qs_array(1)

If StrComp("Kabel", stereotyp) = 0 Then
    Cells(c, 5) = kabeltyp & " " & aderanzahl & "x" & Left(querschnitt, (Len(querschnitt) - 3))
    Cells(c, 7) = "WS01"
ElseIf StrComp("Luftschlauch", stereotyp) = 0 Then
    Cells(c, 5) = kabeltyp & "-" & aderanzahl & "x" & Left(querschnitt, (Len(querschnitt) - 2))
    Cells(c, 7) = "WP01"
End If
```

Quelltext 5: Auslesen und auftrennen der Kabel- bzw. Schlauchattribute und Verknüpfung zur Artikelnummer, Quelle: Eigene Darstellung

In diesem Zuge werden die Länge und der Typical Copy Merge Index in die Tabelle übertragen. Am Ende dieser Prozeduren wird die funktionelle sowie örtliche Zuordnung der Verbindungselemente gesucht und übertragen.

10.3.1.5 Sub-Prozeduren: *FeldgeraetImportListeMessstelle / FeldgeraetImportListeStellstelle*

Auch für den Import der Feldgeräte müssen zwei eigene Tabellen verwendet werden. Gestartet wird wiederum mit der Suche nach einer bestehenden Tabelle mit der gleichen Bezeichnung, die entweder geleert oder neu angelegt wird. Danach werden alle Attributsbezeichnungen, *Anlage*, *Teilanlage*, *Messstelle* bzw. *Stellstelle*, *Feldgerät-BMK*, *Gerätetyp-EB*, *Typical Copy Merge Index* und *Örtliche Zuordnung*, der Tabelle hinzugefügt. Anschließend durchläuft eine *for-each*-Schleife alle *nodes* und fragt diese auf die Stereotypen *Feldgerät-Messung* bzw. *Feldgerät-Aktor* und *Endschalter* ab. Im nächsten Schritt werden Bezeichnung

und Typical Copy Merge Index ausgelesen, danach die funktionelle sowie örtliche Zuordnung eruiert und zuletzt der *EB*-Gerätetyp in die Tabelle übertragen. Der Typ des Geräts wird über den Stereotypen der verbundenen Stelle mit einem *select-case*-Statement ermittelt, wie in Quelltext 6 ersichtlich ist.

```
'Gerätetyp definieren
stereotypMessst = helpNodesMST.Item(0).ChildNodes(1).ChildNodes(0).ChildNodes(4).Attributes.Item(4).NodeValue
Select Case stereotypMessst
Case "Druck"
    Cells(c, 5) = "Sensor, Druck"
Case "Temperatur"
    Cells(c, 5) = "Sensor, Temperatur"
Case "Dichte"
    Cells(c, 5) = "Sensor, Dichte"
Case "Niveau"
    Cells(c, 5) = "Sensor, Stand"
End Select
```

Quelltext 6: *select-case*-Statement für Messstellen-Stereotypen zur Gerätedefinition, Quelle: Eigene Darstellung

10.3.1.6 Sub-Prozedur: *KlemmleisteImportListeMessstelle / KlemmleisteImportListeStellstelle*

Ebenso wie in den Kapiteln zuvor, wurde auch hier je eine Prozedur für Messstellen sowie Stellstellen erstellt, welche die Klemmleisten im Modell sucht und für den Import vorbereitet. Nachdem die Tabellen neu angelegt oder bestehende bereinigt wurden, wird durch eine *if*-Abfrage jede *node* auf den Stereotyp *Klemmleiste* geprüft. Anschließend wird geprüft, ob es sich um eine Messstelle bzw. eine Stellstelle handelt, danach wird das Attribut *Typical Copy Merge Index* aus der *node* extrahiert und das Klemmleisten-BMK übertragen. Am Ende werden die örtliche und funktionelle Zuordnung der Klemmleisten der Tabelle angefügt.

Mit diesen Sub-Prozeduren ist der Import für *EB* vollständig vorbereitet.

10.3.2 Sub-Prozedur: *GenerateForEPLAN()*

Mit dieser Sub-Prozedur werden die Informationen aus dem Modell ausgelesen und die Engineering-Liste des *G8* befüllt. Im Unterschied zu der zuvor beschriebenen Vorgangsweise für *EB* werden hier nicht mehrere Tabellen erstellt, sondern der Programmablauf startet immer von einer Stelle und navigiert zu den diversen Modellobjekten. Für die aktuelle Umsetzung ist es notwendig, dass die Projektdaten-Datei des Generators und die *Microsoft-Excel-Makro*-Datei im gleichen Verzeichnis abgelegt sind. Darüber hinaus muss die Projektdaten-Datei geöffnet sein.

Zu Beginn der Sub-Prozedur wird in gleicher Weise wie in der *EB*-Sub-Prozedur ein *openDialog* verwendet und das XML-File in ein Objekt des Typs *DOMDocument* geschrieben, woraus eine *nodeList* aller Objekte des Modells erstellt wird. Danach wird eine *for-each*-Schleife über alle *nodes* gestartet, in der zu Beginn die Stereotypen der *nodes* auf *Druck*, *Temperatur*, *Dichte*, *Niveau* und *Stellstelle* abgefragt werden. Nach positivem Ergebnis dieser Abfrage wird zuerst die Stellennummer ermittelt und danach die Attribute der Stelle nach dem *HW-Typical* untersucht, die im *G8* andere Benennungen haben als im Modell. Dieses Mapping wird mit einem *select-case*-Statement realisiert. Im Anschluss wird aus den Attributen die Stellenbeschreibung ausgelesen und über die *edges* zur Teilanlage sowie Anlage navigiert. Da *EPLAN* keine dreiteilige Funktionsstruktur unterstützt, wird die KKS-Nummer zusammengefügt, sodass zuerst die Anlage, anschließend ein *Punkt* und danach die Teilanlage und Stelle geschrieben werden. Im Anschluss

daran wird die örtliche Zuordnung der Stelle eruiert, startend vom Signal über die gesamten IO-Komponenten bis hin zur Örtlichkeit. Bei analogen Messstellen werden in diesem Zuge das BMK und der IO-Kanal in die Liste übertragen, da hier pro Messstelle nur ein solches Signal vorhanden ist. Im Gegensatz dazu sind bei Stellstellen drei Signale vom Typ *DigitalSignal* vorhanden, wobei hier nur eines dieser durchlaufen wird, um zur Örtlichkeit zu gelangen.

Im nächsten Schritt werden die Feldgeräte der Stellen ermittelt, wofür, wie in Quelltext 7 ersichtlich, der Stereotyp der Stelle abgefragt wird und anschließend entweder nach dem Objekttyp *Feldgerät-Messung* oder *Feldgerät-Aktor* gesucht wird.

```
'Feldgerät suchen
If (StrComp("Stellstelle", stereotyp) = 0) Then
    nodeIdFeldgerät = getNodeByString(node, "Feldgerät - Aktor", xDoc)
    Set helpNodes = xDoc.SelectNodes("/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdFeldgerät & Chr$(34) & "]"")
    Cells(c, 16) = getNodeName(helpNodes.Item(0))
Else
    nodeIdFeldgerät = getNodeByString(node, "Feldgerät - Messung", xDoc)
    Set helpNodes = xDoc.SelectNodes("/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdFeldgerät & Chr$(34) & "]"")
    Cells(c, 16) = getNodeName(helpNodes.Item(0))
End If
```

Quelltext 7: Verbundenes Modellobjekt *Feldgerät* auffindig machen, Quelle: Eigene Darstellung

Als nächstes werden die Verbindungselemente, *Kabel* und *Luftschlauch*, der Stellen gesucht. Dazu wird zuerst für beide Stellentypen das verbundene *Kabel* der aktuellen *node* gefunden, um anschließend nach dem *Luftschlauch* der Stellstellen zu suchen. An dieser Stelle wird, ähnlich wie in der *EB-Sub-Prozedur*, die *EPLAN*-Artikelnummer des Verbindungselements aus den Attributen zusammengestellt.

Als vorletzter Teil dieser Sub-Prozedur werden die verbundenen Klemmleisten aus den Modellobjekten ausgelesen. Da die betrachteten analogen Messstellen jeweils nur eine Klemmleiste besitzen, wird in diesem Fall lediglich die verbundene Klemmleiste gesucht und dessen BMK in die Engineering-Liste übernommen. Bei den Stellstellen werden zwei unterschiedliche Klemmleisten verwendet, daher ist hier mehr Aufwand notwendig. Gestartet wird mit einer eigens erstellten Funktion *getAllNodesOfAKindByString*, die als Übergabeparameter die aktuelle *node*, den gesuchten Typ als *String* und das XML-Dokument erhält. Zurückgegeben wird eine Array mit den IDs aller gefundenen *nodes* des bestimmten Typs. Für diesen Fall werden alle verbundenen Klemmleisten benötigt, die anschließend durchlaufen werden. Um herauszuarbeiten, ob es sich bei der aktuellen Klemmleiste um die digitale Eingangs- bzw. Ausgangsklemmleiste handelt, wird über die Klemme das angeschlossene Gerät ermittelt. Handelt es sich um ein Magnetventil, wird die Klemmleistenbezeichnung in jene Spalte der Engineering-Liste geschrieben, die den Parameter der Ausgangsklemmleiste enthält. Zeitgleich wird das BMK des Magnetventils ausgelesen und in die Liste weitergegeben. Ist das Gerät ein Endschalter, wird das Klemmleisten-BMK in die dafür vorgesehene Parameter-Spalte geschrieben.

Der letzte Abschnitt der Sub-Prozedur befasst sich mit dem Auslesen der digitalen Signale und der Endschalter bei der Stellstelle. Dazu werden alle verbundenen *nodes* vom Typ *DigitalSignal* geladen und durchlaufen. Danach werden aus jeder *node* die Attribute ausgelesen und, abhängig ob ein *Eingang* oder *Ausgang* vorliegt, unterschiedliche Schritte gesetzt. Bei einem *Ausgang* wird zuerst der Kanal des Signals übergeben und anschließend zur IO-Karte navigiert und dessen BMK in die Tabelle geschrieben. Da zwei *Eingang-nodes* in der Stellstelle vorliegen, muss hier zusätzlich in einem *select-case*-Statement entschieden werden, ob es sich um das erste oder zweite Signal handelt. Im ersten Durchlauf werden die IO-

Kanalnummer und das IO-Karten-BMK ausgegeben. Zusätzlich wird zum verbundenen Endschalter navigiert und dessen Bezeichnung ausgegeben.

Quelltext 8 lässt erkennen, dass der zweite Durchlauf die gleiche Vorgehensweise enthält, jedoch in andere Zellen in der Engineering-Liste schreibt. Dies ist sichtbar bei den Zuweisungen der *Cells()*, da der zweite Übergabeparameter die Spaltennummer in *Microsoft Excel* darstellt und dieser Wert in den beiden *cases* unterschiedlich ist.

```
If split_space(1) = "Ausgang" Then
    Cells(c, 26) = "CH" & getNodeName(helpNodes.Item(0))
    nodeIdKarte = getNodeByString(helpNodes.Item(0), "digitale Ausgangskarte", xDoc)
    Set helpNodes = xDoc.SelectNodes( _
        "/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdKarte & Chr$(34) & "]" )
    Cells(c, 27) = getNodeName(helpNodes.Item(0))
ElseIf split_space(1) = "Eingang" Then
    inputTimes = inputTimes + 1
    Select Case inputTimes
        Case 1: Cells(c, 30) = "CH" & getNodeName(helpNodes.Item(0))

            nodeIdEdnl = getNodeByString(helpNodes.Item(0), "Endschalter", xDoc)
            Set endlNodes = xDoc.SelectNodes( _
                "/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdEdnl & Chr$(34) & "]" )
            Cells(c, 29) = getNodeName(endlNodes.Item(0))

            nodeIdKarte = getNodeByString(helpNodes.Item(0), "digitale Eingangskarte", xDoc)
            Set helpNodes = xDoc.SelectNodes( _
                "/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdKarte & Chr$(34) & "]" )
            Cells(c, 31) = getNodeName(helpNodes.Item(0))

        Case 2: Cells(c, 33) = "CH" & getNodeName(helpNodes.Item(0))

            nodeIdEdnl = getNodeByString(helpNodes.Item(0), "Endschalter", xDoc)
            Set endlNodes = xDoc.SelectNodes( _
                "/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdEdnl & Chr$(34) & "]" )
            Cells(c, 32) = getNodeName(endlNodes.Item(0))

            nodeIdKarte = getNodeByString(helpNodes.Item(0), "digitale Eingangskarte", xDoc)
            Set helpNodes = xDoc.SelectNodes( _
                "/a:graphml/a:graph/a:node[@id=" & Chr$(34) & nodeIdKarte & Chr$(34) & "]" )
            Cells(c, 34) = getNodeName(helpNodes.Item(0))
    End Select
End If
```

Quelltext 8: Auslesen der Digitalsignale und Endschalter bei Stellstellen für *EPLAN*, Quelle: Eigene Darstellung

Somit sind alle notwendigen Spalten in der Engineering-Liste der Projektdaten-Datei befüllt und der Generierungsprozess kann gestartet werden.

10.3.3 Funktion: *getNodeByString*

Diese Funktion ist entstanden, um das Navigieren zwischen zwei *nodes* zu vereinfachen und wird überdies sehr häufig in den beiden Sub-Prozeduren verwendet. Als Übergabeparameter benötigt sie die aktuelle *node*, von welcher der Navigationsvorgang gestartet werden soll, einen *String*, der den Stereotypen des gesuchten Modellobjekts enthält, und das gesamte XML-File in Form eines *DOMDocument*-Objekts.

Zu Beginn wird eine *nodeList* mit allen *edges* des Modellobjekts erstellt. Danach werden alle *edges* durchlaufen und verglichen, ob die Quelle bzw. das Ziel der *edge* mit der ID der übergebenen *node* übereinstimmt. Trifft eines der beiden zu, wird eine *nodeList* erzeugt, in der ausschließlich die gefundene *node* enthalten ist. Dessen Stereotyp wird abschließend mit dem übergebenen Stereotypen verglichen; stimmen diese überein wird aus der Funktion gesprungen und die ID der gefundenen *node* als Rückgabewert zurückgegeben.

10.4 Conclusio aus dem Lese- bzw. Aufbereitungsvorgang

Bevor mit der Programmierung des *Microsoft Excel*-Makros gestartet wird, muss die *GraphML*-Datei genau analysiert werden, damit die Verbindung zwischen dem Modell und den Informationen in der Datei hergestellt werden kann. Außerdem muss die Arbeitsweise der Generatoren verstanden werden, um die Importdateien entsprechend befüllen und aufbereiten zu können. Um die Programmierung voranzutreiben, wurden einige Testgenerierungen durchgeführt, deren Ergebnisse direkt in das Makro zurückgeflossen sind. Eine Erkenntnis ist beispielsweise, dass beim *EB*-Geräteimport zwei Tabellen und somit zwei Importläufe notwendig sind, um Objekte mit funktioneller Zuordnung zu importieren. Das liegt daran, dass entweder der Funktionstyp *Messstelle* oder *Stellstelle* als *zugeordnete Funktion* ausgewählt werden kann. Diese Vorgangsweise wird durch den Support der Firma *Aucotec* bestätigt. Ein weiterer Importversuch zeigt auf, dass die Anpassung eines *EPLAN*-Makros notwendig gewesen ist, um mit dem Programm die richtigen Parameter beschreiben zu können.

Die Einschätzung des Autors ist jene, dass der Vorgang zum Auslesen und Aufbereiten der Informationen für *EB* aufwendiger ist als für den Schaltplangenerator *G8*. Es müssen dazu mehrere Tabellen erstellt und das gesamte XML-File öfter durchlaufen werden. Dieser Umstand ist bei kleinen Versuchsprojekten wie diesem weniger dramatisch, jedoch bei Großprojekten ausschlaggebend in Hinsicht auf die Laufzeit des *Microsoft Excel*-Makros. Da im *G8* keine Geräte vorab importiert, sondern die Pläne pro Stelle erzeugt werden, ist die Aufbereitung der Engineering-Liste weniger komplex. Trotzdem müssen einige Zusammenhänge zwischen Modellobjekten bestehen, die durchlaufen werden müssen, um alle notwendigen Informationen zu erhalten.

11 GENERIERUNG, ERGEBNISSE UND NACHBEARBEITUNG

Nachdem die Modellinformationen ausgelesen und aufbereitet sind, starten die eigentlichen Generierungen in den beiden Endplattformen. In diesem Kapitel werden diese Vorgänge beschrieben sowie die notwendigen Unterlagen und Einzelschritte erläutert.

11.1 Objekt- und Plangenerierung in EB

Durch den Umstand, dass der *Typical Manager* keine Möglichkeit bietet, Objekte während des Generierungsvorganges mit Informationen zu befüllen, werden die relevanten Objekte vorab angelegt. Anschließend werden die bereits angelegten Objekte, u.a. mit Hilfe des Attributs *Typical Copy Merge Index*, vom *Typical Manager* verwendet, um Pläne zu generieren. Jene Objekte, die noch nicht angelegt, jedoch im *Typical* vorhanden sind, werden durch den *Typical Manager* hinzugefügt. Um diesen Workflow umzusetzen, ist der Generierungsvorgang in zwei Teile aufgeteilt. Zuerst werden die Objekte mit Hilfe der durch das *Microsoft Excel-Makro* erstellten Tabellen angelegt. Dieser Vorgang wird mit Hilfe des *EB-Assistenten Elemente importieren und aktualisieren* durchgeführt. Im zweiten Teil kommt der *Typical Manager* zum Einsatz, um die Pläne anzulegen und die Verbindung zwischen Stellenplan und Objekten herzustellen.

11.1.1 Import der Objekte aus den *Microsoft Excel* Tabellen

Für den Objektimport wird der bereits angesprochene Assistent verwendet, der im Standardumfang von *EB* enthalten ist. Gestartet wird dieser auf dem *Betriebsmittel-Ordner* bzw. *Funktionen-Ordner* im angelegten Zielprojekt.

Abbildung 51 zeigt den Aufbau des Assistenten, in dem zuerst die *XLSX*-Datei und die gewünschte Tabelle ausgewählt werden. Im Bereich *Mapping* kann eine genauere Auswahl der Elementart, z.B. nur Import von *Gerätetypen*, und des Elementtyps, z.B. nur Geräte vom Typ *Motor*, vorgenommen werden. Darunter sind alle Spalten der Tabelle aufgelistet, denen eine bestimmte Funktionalität im Mapping zugeteilt ist. In Abbildung 51 wird der Import der Tabelle *MSR-Liste* vorbereitet, bei dem die ersten zwei Spalten *Anlage* und *Teilanlage* als Funktionsstruktur des Objekts verwendet werden, das eigentlich importiert wird, der *Stelle*. Dazu ist diesen beiden die *Operation Funktion* zugeteilt und als Funktionstyp *Anlage* vergeben. Unter *Ebene* ist der hierarchische Aufbau geregelt, wobei 1 für die höchste Ebene steht. Der Spalte *Stelle* ist das Attribut *Name* zugeordnet, die darüber hinaus als Primärschlüssel verwendet wird. Dieser ist dort zu vergeben, wo eine eindeutige Kennzeichnung aller Importobjekte möglich ist. Der Inhalt der Spalten *Bezeichnung*, *HW-Typical*, *Schrank* und *Feld* werden in die ausgewählten *EB*-Attribute eingetragen. Die beiden *Typical Baueinheiten* werden an der *Stelle* eingetragen und vom *Typical Manager* verwendet. Aufgrund der Einträge der letzten Spalte *Funktionstyp-EB* wird die Stelle entweder als *Messstelle* oder als *Stellstelle* in *EB* angelegt.

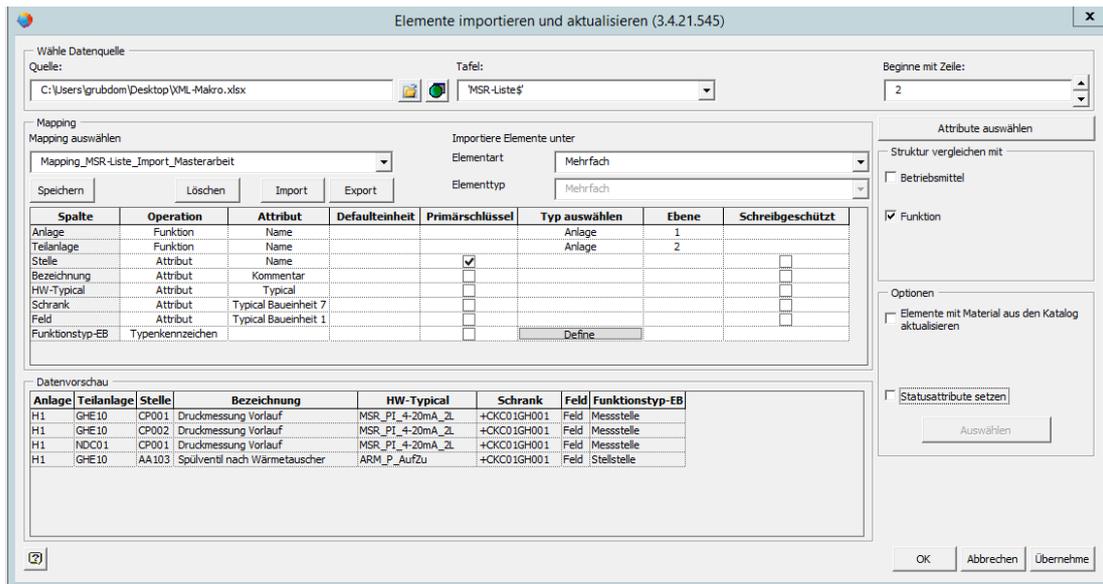


Abbildung 51: EB-Assistent *Elemente importieren und aktualisieren*, Quelle: Eigene Darstellung

Abbildung 52 zeigt das Typen-Mapping für Funktionstypen, in dem die Schlüsselwörter aus der Spalte der Tabelle ausgelesen werden und die Zuordnung zu den EB-Funktionstypen manuell durchgeführt wird.

Im unteren Bereich des Assistenten wird eine Vorschau der Tabellen-Daten angezeigt, die Optionen auf der rechten Seite werden nicht für jedes Mapping verwendet. Der Strukturabgleich wird entsprechend der zu importierenden EB-Typen entweder auf *Betriebsmittel* oder *Funktionen* eingestellt. Speziell beim Import der SPS-Komponenten wird das Aktualisieren der Betriebsmittel mit dem Katalog benötigt, da auf diese Weise die Objektstruktur der IO-Karten selbstständig angelegt wird.

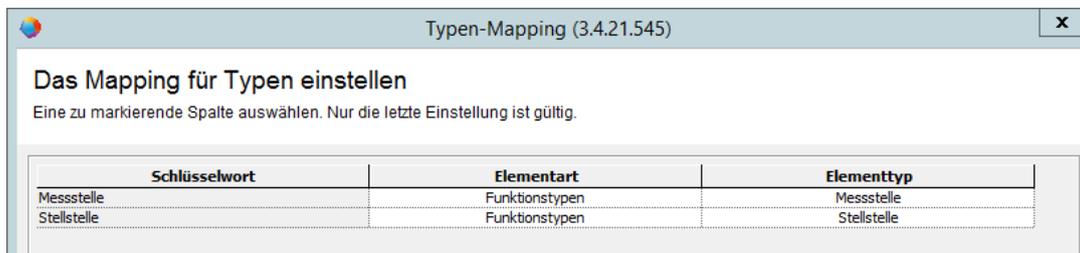


Abbildung 52: Typen-Mapping für Funktionstypen im Assistenten *Elemente importieren und aktualisieren*, Quelle: Eigene Darstellung

Abbildung 53 zeigt das Mapping zum Import von Verbindungselementen, die einer Stellstelle zugeordnet sind. Die ersten drei Spalten in der Mappingtabelle stellen die funktionelle Zuordnung des Kabels bzw. Luftschlauchs dar. Unter *Stelle* kann in diesem Assistenten nur ein spezieller *Typ* ausgewählt werden, in diesem Fall *Stellstelle*. Aus diesem Grund existieren für alle funktionell zugeordneten Geräte zwei Mappings, um den Geräten den jeweiligen Funktionstyp zuzuordnen zu können. Das *Kabel-BMK* dient wieder als Primärschlüssel und bildet die Bezeichnung des angelegten Objekts. Durch die Auswahl der *Elementart* sowie dem *Elementtyp* importiert der Assistent die Elemente der Tabelle als *Mehradriges Kabel* in EB.

Die erstellten Mappings können abgespeichert und wiederverwendet werden. Für jede Tabelle in der erstellten *Microsoft Excel*-Datei ist ein solches Mapping vorhanden.

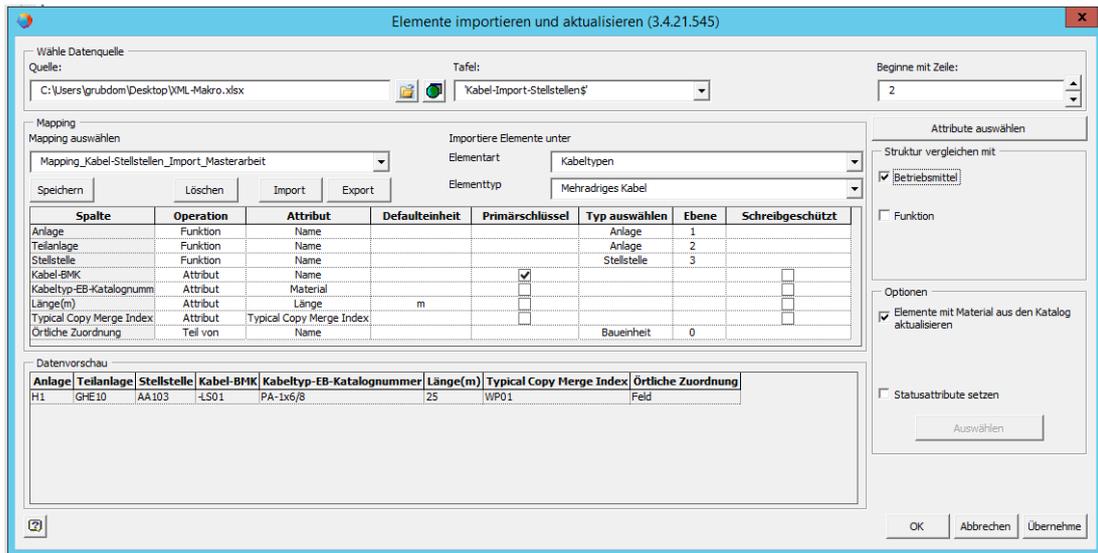


Abbildung 53: Stellschleifenbezogenes Kabelimport-Mapping, Quelle: Eigene Darstellung

Der Import der diversen Elemente muss in einer bestimmten Reihenfolge geschehen, damit die Geräte die richtigen Zuordnungen erhalten können. Aus diesem Grund müssen zuerst die Funktionen und anschließend die Baueinheiten importiert werden. Die Reihenfolge, in der die Geräte angelegt werden, spielt keine Rolle. Es werden dazu die Mappings *SPS*, *Kabel-Messstellen*, *Kabel-Stellschleifen*, *Feldgeräte-Messstellen*, *Feldgeräte-Stellschleifen*, *Klemmleisten-Messstellen* und *Klemmleisten-Stellschleifen* verwendet. Unter Anhang 8 werden alle Mappings als Abbildungen angefügt, diese hier zu beschreiben, würde keinen Mehrwert für die Arbeit bringen.

Sind alle Elemente der Tabellen importiert, entstehen die Objekte, die in Abbildung 54 dargestellt sind.

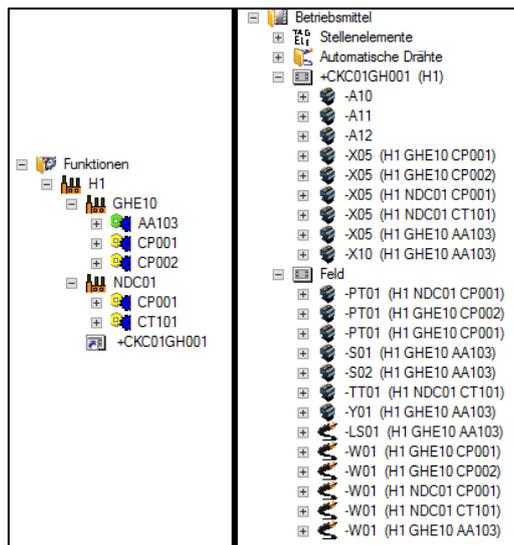


Abbildung 54: Angelegte Funktionen und Geräte durch die ausgelesenen Modellinformationen, Ansicht im EB-Explorer, Quelle: Eigene Darstellung

Die Zuordnung von Betriebsmitteln zu Funktionen wird in *EB* neben der Objektbezeichnung in Klammer angezeigt. Da der Schaltschrank der Anlage *H1* zugeordnet ist, erscheint dieser als *Verweis* im *Funktionen*-Ordner (im linken Teil von Abbildung 54, letztes Symbol).

11.1.2 Pläne generieren mit dem *Typical Manager*

Nachdem alle Funktionen angelegt sind, werden anschließend die Pläne mit Hilfe des *Typical Managers* generiert. Dazu wird dieser wie in Kapitel 9.1 angewendet und eingestellt. Ohne weitere manuelle Eingriffe in *EB* wird der *Typical Manager* auf dem *Funktions*-Ordner gestartet. Es öffnet sich der Dialog aus Abbildung 55, in dem bei allen Stellen die Aktion *Kopieren* ausgewählt und der Kopiervorgang mit Bestätigen des *OK*-Buttons gestartet wird.

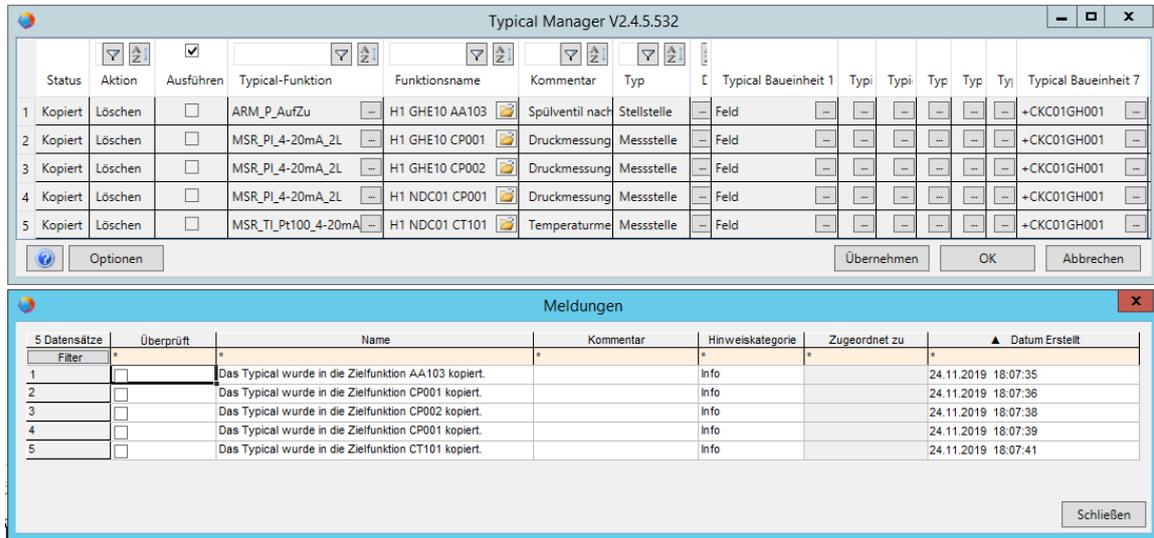


Abbildung 55: *EB*-Assistent *Typical Manager* und darunter Meldungsanzeige nach erfolgreichem Kopiervorgang, Quelle: Eigene Darstellung

Nachdem alle Stellen kopiert sind, erscheint eine Meldungsanzeige, die keine Fehler enthält (siehe Abbildung 55 unten).

11.1.3 Ergebnisse und Nachbearbeitung in *EB*

Somit ist der Generierungsprozess in *EB* abgeschlossen; ein Teil des Ergebnisses, die Objekte der Datenbank, sind in Abbildung 56 in der Tree-Ansicht dargestellt.

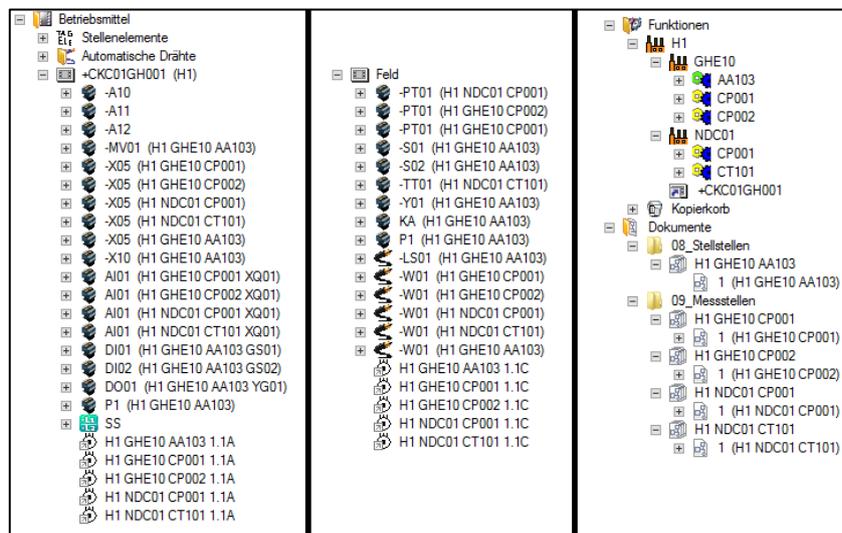


Abbildung 56: Ansicht der *Betriebsmittel*-, *Funktions*- und *Dokumenten*-Ordner nach erfolgreichem Generierungsprozess, Quelle: Eigene Darstellung

Die Pläne sind im *Dokumente*-Ordner mit der gleichen Strukturierung wie im Typicalprojekt angelegt, wobei jedes *Blatt* in einer *Zeichnung* mit der vollständigen funktionellen Kennzeichnung der Stelle angelegt ist. Auch hier zeigt die eingeklammerte Nummer neben der *Blattnummer* auf die *zugeordnete Funktion* des *Blattes* hin. Zwei der erstellten Stellenpläne sind dem Anhang beigelegt.

Es werden durch den *Typical Manager* weitere Geräte angelegt, die nicht im Modell enthalten sind und somit nicht daraus ausgelesen wurden. Darunter z.B. das Objekt *KA*, das die Armatur repräsentiert, die in der Rohrleitung eingebaut ist, oder der Luftanschluss *P1* zur Druckluftversorgung des pneumatischen Antriebes. Die Objekte im linken Teil der Abbildung, bezeichnet mit *AI01*, *DI01* bzw. *02* und *DO01*, sind ebenso durch den *Typical Manager* angelegt. Diese sind vom Gerätetyp *Eingang* bzw. *Ausgang* und müssen in einem Nachbearbeitungsschritt mit den IO-Kanälen der bereits angelegten IO-Komponenten (*-A10*, *-A11* und *-A12*) verbunden werden. Dazu wird ein weiterer *EB-Assistent*, der *Gerätezuordnungsassistent*, verwendet, der aufgrund des Attributs *IO-Typ* die Kanäle der angelegten IO-Karten mit den *freien* Kanälen der Typicals verschmilzt.

Abbildung 57 zeigt den Assistenten, in dem die freien Kanäle links und die Kanäle der IO-Karten rechts dargestellt sind. Aufgrund der Einträge im Attribut *IO-Typ* werden die Kanäle miteinander verschmolzen, d.h. der freie Kanal wird gelöscht und dessen Attributwerte in den anderen geschrieben. Mit dem Button *Auto-Merge* werden die Kanäle links, in denen der Wert *WORD* eingetragen ist, automatisch in die Kanäle rechts übergeführt. Für die restlichen Kanäle werden die anderen Karten angewählt und der Vorgang wiederholt.

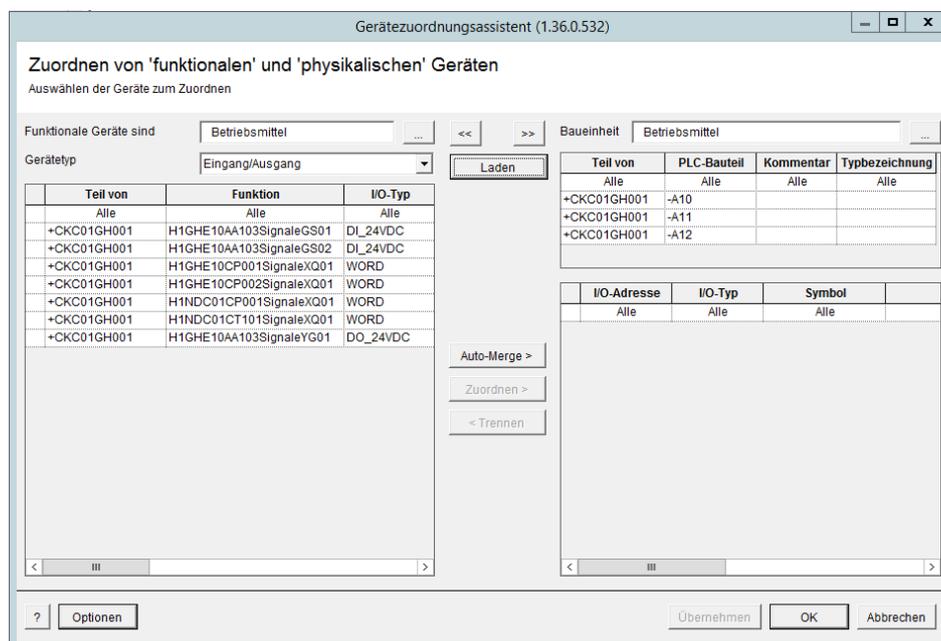


Abbildung 57: *Gerätezuordnungsassistent* mit den geladenen IO-Kanälen, links die freien, rechts die der IO-Karte, Quelle: Eigene Darstellung

An dieser Stelle enden die weiteren Schritte, die in dieser Arbeit für den Generierungsprozess dieses Testmodells vorzunehmen sind. In der Praxis wären nun die Stromlaufpläne der IO-Karten, der Spannungsversorgung, weitere allgemeine Pläne, der Schaltschrankaufbau und die Folgedokumentation wie Klemmenpläne oder Kabellisten zu erstellen.

11.2 Plangenerierung in EPLAN

Für die Plangenerierung in *EPLAN* wird der bereits beschriebene Schaltplangenerator *G8* verwendet, der die Informationen zur Planerstellung aus einer *Microsoft Excel*-Datei bezieht. Durch den Auslese-Vorgang ist diese Datei befüllt und bereit für die notwendigen Schritte zur finalen Generierung.

11.2.1 Planerstellung mit dem Schaltplangenerator G8

Wie bereits unter Kapitel 9.2 genau beschrieben, wird aus der Engineering-Liste die Makro-Liste erzeugt. Abbildung 58 zeigt einen Ausschnitt dieser Engineering-Liste, in der zu Beginn die Seitenbeschreibung und die *G8*-Typicalbezeichnung stehen, anschließend die Strukturkennzeichen der Stellen und danach alle Makroparameter. Die gesamte Engineering-Liste mit allen Spalten ist unter Anhang 10 zu finden.

	C	E	F	G	K	L	P	Q	S	T
7	Seiten		Typical		Strukturkennzeichen		Parameter			
8		Seitenbeschreibung	Typical	Typicalname	Strukturkennzeichen	Parameter	Sensor BMK	Einbauort	Analogtochter Kanal	Analogtochter BMK
9										
10	Seitenbeschreibung	TYPICAL_ECAD	=		PSENSBMK	Einbauort	PIOTochterAEKanal	PIOTochterAEBMK		
11										
12	Druckmessung vor Hauptventil FW-RL	Typicals MA_Druckmessung	H1.GHE10CP001	-PT01	+CKC01GH001	CH00	-A10			
13	Druckmessung nach Hauptventil FW-RL	Typicals MA_Druckmessung	H1.GHE10CP002	-PT01	+CKC01GH001	CH01	-A10			
14	Druckmessung Bypass FW-HL	Typicals MA_Druckmessung	H1.NDC01CP001	-PT01	+CKC01GH001	CH02	-A10			
15	Hauptventil Rücklauf Fernwärmeleitung	Typicals MA_Ventil	H1.GHE10AA103	-Y01	+CKC01GH001					
16	Temperaturmessung Rücklauf FW-HL	Typicals MA_Temperaturmessung	H1.NDC01CT101	-TT01	+CKC01GH001	CH03	-A10			

Abbildung 58: Ausschnitt der befüllten Engineering-Liste des *G8*, Quelle Eigene Darstellung

Nun wird aus der Engineering-Liste die Makro-Liste erstellt, indem der dafür vorgesehene *G8*-Mechanismus gestartet wird. Dieser lädt alle notwendigen Komponenten, wie unter 9.2.3 erläutert, in die Makro-Liste. Ein Ausschnitt des Ergebnisses ist in Abbildung 59 dargestellt.

	A	G	I	O	P	Q	R	S	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1	Makroliste																				
7																					
8		copyFromSource	copyf	copyFromSource		copyFromClass		copyFromClass	copyf	copyf	copyf	copyf	copyf	copyf	copyf	copyFromSource	copyFromSource	copyFromSource	copyFromSource	copyFromSource	copyFromSource
9			Seitenname		Seitenbeschreibung	Makros	Makroname	Makrovariante	Plug / Socket	RP	SPS_1	SPS_2	SPS_3	SPS_4	Mass	Ventil	Parameter	Einbauort	PIOTochterAKanal	PIOTochterAEBMK	PSENSBMK
10																					
11																					
12																					
13	H1.GHE10CP001	1	Druckmessung vor Hauptventil FW-RL		SPS\BAS_4SPS.ema	A					1						+CKC01GH001	CH00	-A10		-PT01
14	H1.GHE10CP001		Druckmessung vor Hauptventil FW-RL		SPS\FRAG_SPS_Bit_AE_2AN.ema	A			1								+CKC01GH001	CH00	-A10		-PT01
15	H1.GHE10CP001		Druckmessung vor Hauptventil FW-RL		SPS\FRAG_ANA_Mess.ema	A			2								+CKC01GH001	CH00	-A10		-PT01
16	H1.GHE10CP002	1	Druckmessung nach Hauptventil FW-RL		SPS\BAS_4SPS.ema	A				1							+CKC01GH001	CH01	-A10		-PT01
17	H1.GHE10CP002		Druckmessung nach Hauptventil FW-RL		SPS\FRAG_SPS_Bit_AE_2AN.ema	A			1								+CKC01GH001	CH01	-A10		-PT01
18	H1.GHE10CP002		Druckmessung nach Hauptventil FW-RL		SPS\FRAG_ANA_Mess.ema	A			2								+CKC01GH001	CH01	-A10		-PT01
19	H1.NDC01CP001	1	Druckmessung Bypass FW-HL		SPS\BAS_4SPS.ema	A				1							+CKC01GH001	CH02	-A10		-PT01
20	H1.NDC01CP001		Druckmessung Bypass FW-HL		SPS\FRAG_SPS_Bit_AE_2AN.ema	A			1								+CKC01GH001	CH02	-A10		-PT01
21	H1.NDC01CP001		Druckmessung Bypass FW-HL		SPS\FRAG_ANA_Mess.ema	A			2								+CKC01GH001	CH02	-A10		-PT01
22	H1.GHE10AA103	1	Hauptventil Rücklauf Fernwärmeleitung		SPS\BAS_Ventil.ema	A				A	B	C					+CKC01GH001				-Y01
23	H1.GHE10AA103		Hauptventil Rücklauf Fernwärmeleitung		SPS\FRAG_SPS_Bit_DA_2AN.ema	A			A								+CKC01GH001				-Y01
24	H1.GHE10AA103		Hauptventil Rücklauf Fernwärmeleitung		SPS\FRAG_SPS_Bit_D1_2AN.ema	A			B								+CKC01GH001				-Y01
25	H1.GHE10AA103		Hauptventil Rücklauf Fernwärmeleitung		SPS\FRAG_SPS_Bit_D2_2AN.ema	A			C								+CKC01GH001				-Y01
26	H1.GHE10AA103		Hauptventil Rücklauf Fernwärmeleitung		SPS\FRAG_DIG_Ventil.ema	A			D								+CKC01GH001				-Y01
27	H1.NDC01CT101	1	Temperaturmessung Rücklauf FW-HL		SPS\BAS_4SPS.ema	A					1						+CKC01GH001	CH03	-A10		-TT01
28	H1.NDC01CT101		Temperaturmessung Rücklauf FW-HL		SPS\FRAG_SPS_Bit_AE_2AN.ema	A			1								+CKC01GH001	CH03	-A10		-TT01
29	H1.NDC01CT101		Temperaturmessung Rücklauf FW-HL		SPS\FRAG_ANA_Mess_Temp.ema	A			2								+CKC01GH001	CH03	-A10		-TT01

Abbildung 59: Makro-Liste des *G8* vor der Plangenerierung des Seitenmakros, Quelle: Eigene Darstellung

Durch das *Synchronisieren* der Liste werden die Werte der einzelnen Makros grau hinterlegt, dadurch wird sichtbar gemacht, welche Werte in das jeweilige Makro übernommen werden. Es fällt auf, dass aus den

fünf Stellen der Engineering-Liste mehrere Zeilen entstanden sind. Das liegt daran, dass pro Typical mehrere Makros verwendet werden, die hier jeweils in einer separaten Zeile aufgelistet sind. Dadurch ist die Zuordnung der Parameter bzw. *Plugs* und *Sockets* möglich.

Somit wird der letzte Schritt im *G8* gestartet, das Erstellen des Seitenmakros aus der Makro-Liste. Im bereits unter 8.2.2 erstellten Projekt existieren zu diesem Zeitpunkt noch keine Pläne, es ist ein leeres Projekt. Darin wird das soeben generierte Seitenmakro eingefügt, das insgesamt fünf Stromlaufpläne enthält. Der Generierungsvorgang an sich ist damit abgeschlossen.

11.2.2 Ergebnisse und Nachbearbeitung in *EPLAN*

Die eingefügten Pläne erhalten durch den Schaltplangenerator die funktionelle Strukturierung, wie in Abbildung 60 ersichtlich ist.

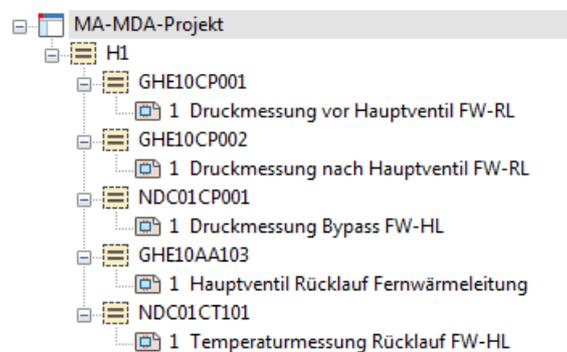


Abbildung 60: Seitenstruktur der eingefügten Pläne des Seitenmakros, erstellt aus dem Schaltplangenerator *G8*, Quelle: Eigene Darstellung

Die Reihenfolge der Pläne ist so, wie sie aus dem Modell ausgelesen und in die Engineering-Liste importiert wurde. Damit diese nach der KKS-Nummer aufsteigend sortiert sind, wird über die *Strukturkennzeichen-Verwaltung* die Sortierung der Anlagenkennzeichen auf *alphanumerisch* eingestellt. Anschließend ist die zweite Strukturebene aufsteigend angeordnet.

Als weiteren Nachbearbeitungsschritt werden die SPS-IO-Komponenten als Pläne angelegt. Dazu werden bereits bestehende Makros verwendet, die als *Übersichtsdarstellung* auf einer *EPLAN*-Seite vom Typ *Übersicht* platziert sind. Es werden drei verschiedene Karten benötigt, wie sie auch im Modell vorhanden sind. Sind diese erstellt, folgt die Zuweisung der IO-Kanäle der Übersichtsdarstellung zu den IO-Kanal-Symbolen auf den Stellenplänen.

Als nächsten Schritt müssen die Kabelartikel in den Symbolen aktualisiert werden. Durch den Import des Artikels als Parameter bekommen die Kabel nicht die Informationen des Kabelartikels aus der Artikelstammdatenbank und müssen daher aktualisiert werden. Dazu werden im *Kabel-Navigator* Filter erstellt, bei denen immer die Kabel mit der gleichen Artikelnummer gemeinsam angezeigt werden. Nachdem alle Kabel gemeinsam markiert und geöffnet sind, wird im Register der Artikel aufgerufen. Die Artikeldatenbank öffnet sich mit dem gewünschten Artikel und wird mit *OK* bestätigt. Danach sind die Kabelinformationen aus der Datenbank in das Symbol geladen und werden am Plan angezeigt.

Die gleichen Stellenpläne wie aus *EB* sind ebenso in Form der *EPLAN*-Seiten unter Anhang 11.1 und 11.2 beigefügt.

11.3 Resümee des Generierungsprozesses, der Ergebnisse und Nachbearbeitungsschritte

Die eigentlichen Generierungsprozesse in den Engineering-Plattformen sind, bezogen auf den Aufwand, nicht sehr umfangreich. Die notwendige Vorbereitung, von Makros bzw. Typicals über den Objektimport bis hin zum richtigen Aufbau der *Microsoft Excel* Tabellen, ist maßgeblich für die Qualität des Workflows.

Die Generierung in *EB* ist relativ einfach und schnell durchführbar, da bereits alles genau vorbereitet sein muss. Die wichtigsten Punkte dabei sind der richtige Aufbau der Typicals und der *Typical Copy Merge Index*, der die bestehenden Objekte mit den neuen zusammenfügt. Das erhaltene Ergebnis entspricht den Erwartungen bezüglich der Vollständigkeit der Pläne sowie Objekte. Die hier betrachteten Nachbearbeitungsschritte sind durch den Assistenten gut und schnell durchführbar.

Die Vorbereitung für den Schaltplangenerator *G8*, der die Pläne für *EPLAN* vervielfältigt, ist verhältnismäßig aufwendig. Sind diese Schritte durchgeführt, ist der Prozess jedoch intuitiv und gut anwendbar. Das erhaltene Ergebnis entspricht ebenfalls den Erwartungen, die Nachbearbeitungsschritte sind umfangreicher verglichen zu *EB*, könnten jedoch durch Erweiterung des Generierungsprozesses verringert werden.

Grundsätzlich kann angemerkt werden, dass die Generierung in beiden Plattformen sehr stark von der Vorbereitung abhängt und in guter Weise funktioniert.

12 VERIFIZIEREN DER ERGEBNISSE

Aufgrund der unter Kapitel 5.4 ausgearbeiteten Anforderungen an den Workflow, wird nun die Verifizierung der Ergebnisse durchgeführt. Dazu wird der gesamte Durchlauf vorab mit den Teilnehmern durchgegangen und die Erfüllung der Anforderungen bewertet. In Tabelle 4 werden diese Anforderungen inklusive Ergebnisse sowie dem Erfüllungsgrad aufgelistet

Teilnehmer: Hr. Gruber Dominik (Junior-Lead-Engineer / metior GmbH), Hr. Tatzler Viktor (Lead-Engineer / metior GmbH)

Anforderung	Ergebnis	Erfüllungsgrad (geschätzt)
Zeitfaktor verringern	Vergleich der Engineering-Stunden: 100 Stunden konventionell 75 Stunden (geschätzt) mit Workflow NEU	25%
Fehlerhäufigkeit verringern	Automatische Generierung selbst fehlerfrei → Fehleranfälligkeit verschiebt sich in Modellierung	60%
Schnellere Reaktion auf kurzfristige Änderungen	Keine Verbesserung, da sich die Vorgänge von der Engineering-Plattform in das Modell verschieben	0%
Weniger Zeitaufwand mit Engineering-Umgebung	Für den End-User (Techniker/Technikerin) wenig bis keine Interaktion im Engineering-Tool	85%
Unabhängigkeit von Engineering-Umgebung des Kunden	Bei vollständig umgesetztem Workflow komplett unabhängig, ansonsten Aufbereitung der Transformation notwendig (Typicalbau nicht betrachtet, da immer notwendig)	80%
Planungsqualität unabhängig von Engagement des Mitarbeiters/ der Mitarbeiterin	Tagesverfassung, Krankheitsbedingt, Persönliche Befindlichkeit → Listen werden trotzdem von MA aufbereitet	50%
Einsparen des Bulk-Engineering	Weitestgehend möglich, jedoch abhängig von Nachbearbeitungsaufwand in Engineering-Umgebungen	70-80%

Tabelle 4: Verifizierungstabelle inklusive geschätztem Erfüllungsgrad, Quelle: Eigene Darstellung

Gesamt ergibt sich ein Erfüllungsgrad von rund 50%, der sich aufgrund von geschätzten Einzelerfüllungen ergibt. Die Erfüllung ist darauf bezogen, dass der Workflow komplett umgesetzt und für den vollumfänglichen Einsatz bereit ist. Diese ca. 50% bedeuten in einem Projekt, dass die Qualität der Dokumentation verdoppelt und das Bulk-Engineering um rund ein Drittel reduziert wird.

13 ZUSAMMENFASSUNG UND AUSBLICK

Die Erkenntnis dieser Arbeit ist, dass der modellgetriebene Entwicklungsansatz durchaus für die EMSR-Hardware-Planung eingesetzt werden kann. Der umgesetzte Umfang, insgesamt fünf Stellen sowie einige SPS-IO-Komponenten, kann auf ein Großprojekt ausgeweitet werden.

Aus dem zu Beginn untersuchten, umfassenden Projektablauf inklusive dessen Unterlagen und Dokumenten wird der ausgearbeitete Workflow für die EMSR-Detailplanung, genauer die Realisierung der Hardware-Dokumentation, verwendet. Basierend auf einem Modell, einer vereinfachten Darstellung der Realität, das die Komponenten einer Prozessanlage enthält, kann die Dokumentation in zwei unterschiedlichen Engineering-Plattformen generiert werden. Dieser Vorgang ist nicht vollständig automatisiert, sondern bedarf Eingaben bzw. manuelle Schritte der Technikerin bzw. des Technikers. Als Basis für die Modellierung existiert ein Metamodell, in dem die verwendbaren Komponenten, wie z.B. Klemmen, Feldgeräte oder IO-Karten, aber auch funktionelle und örtliche Strukturierungseinheiten, mit dessen Attributen angeführt sind. Die Interaktion zwischen diesen Objekten wird ebenso in dem Metamodell beschrieben. Die Modellierung selbst wird auf Basis des R&I-Fließschema, der Stellenlisten sowie der Lastenheftauszüge erstellt. Das Anlagenmodell dieser Ausarbeitung entsteht in einer Visualisierungsumgebung, die als Exportformat das XML basierte, standardisierte Graphen-Format GrapML verwendet. Daraus werden die Modellinformationen über ein eigens dafür erstelltes VBA-Makro ausgelesen und für die Generatoren der beiden Engineering-Plattformen aufbereitet. Diese sind *Engineering Base* und *EPLAN Electric P8* mit dem *EB-Standard-Generatordassistenten Typical Manager* sowie dem *EPLAN-Schaltplangenerator G8* der Firma *CAE Expert Group*. Da beide Systeme mit *Microsoft Excel* interagieren bzw. darauf basieren, sind die Modellinformationen in diesem Format aufbereitet. Die zur Vervielfältigung benötigten Typical-Pläne werden simultan mit dem Modell erarbeitet und dienen als Grundlage für den Generierungsprozess. Notwendige Nachbearbeitungsschritte in den Engineering-Umgebungen runden die Dokumentation der EMSR-Hardware-Detailplanung ab.

Die Ergebnisse daraus lassen darauf schließen, dass durch diesen Workflow ein großer Mehrwert bei der Erstellung der Anlagenplanung generiert werden kann. Dennoch ist eine solche Umsetzung mit großem Aufwand verbunden, da die Programmierung aktuell genau auf dieses Modell zugeschnitten ist. Durch diesen teilautomatisierten Vorgang verlagern sich die Planungsaktivitäten weg von der Engineering-Plattform hin zur technischen Ausarbeitung. Somit werden zeitliche und menschliche Ressourcen besser eingesetzt und kürzere Projektdurchlaufzeiten können entstehen. Die Umsetzung einer Änderungsverfolgung des Modells wurde im Zuge dieser Arbeit nicht implementiert, da das Hauptaugenmerk auf der teilautomatischen Generierung von Stellenplänen liegt. Die aktuelle Ausprägung verlangte bereits vollständig die technische und zeitliche Aufmerksamkeit des Autors, sodass dieses zusätzliche Feature nicht ausgearbeitet wurde.

Nachdem in dieser Arbeit nicht alle Typicals sowie ein umfassendes Modell umgesetzt sind, stellt sich die Frage einer weiteren Vorgangsweise. Da, wie bereits erwähnt, eine allumfassende Umsetzung viel Zeit in Anspruch nimmt, ist eine Weiterführung im Unternehmen vorerst nicht angedacht. Die weiteren Schritte, nämlich die Vervollständigung des Metamodells, Ausbreitung auf weitere Engineering-Plattformen, Umsetzung aller Typical-Varianten, Erweiterung bzw. Anpassung der Programmierung und Vereinfachung der Nachbearbeitungsschritte, bedürfen einiger Großprojekte, die in diesem Ausmaß im Unternehmen selten abgewickelt werden. Andernfalls können die notwendigen Entwicklungskosten nicht kompensiert bzw. Zeit

und daraus resultierende Kosten eingespart werden. Aus diesem Grund wird das Thema von automatisiertem Hardware-Engineering unter Zuhilfenahme von modellorientierter Architektur aktuell ruhend gelegt.

LITERATURVERZEICHNIS

Gedruckte Werke (25)

Alt, Oliver (2012): *Modellbasierte Systementwicklung mit SysML*, Carl Hanser, Dreieich, Deutschland

Andresen, Andreas (2003): *Komponentenbasierte Softwareentwicklung - mit MDA, UML und XML*, Carl Hanser, München/Wien

Bindel, Thomas; Hofmann, Dieter (2016): *R&I-Fließschema - Übergang von DIN 19227 zu DIN EN 62424*, Springer Vieweg, Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Deutschland

Bindel, Thomas; Hofmann, Dieter (2017): *Projektierung von Automatisierungsanlagen - Eine effektive und anschauliche Einführung*, 3. Auflage, Springer Vieweg, Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Deutschland

Breu, Ruth; Matzner, Thomas; Nickl, Friederike (2005): *Software Engineering - Objektorientierte Techniken, Methoden und Prozesse in der Praxis*, Oldenburg Wissenschaftsverlag, Rosenheimer Straße 145, 81671 München, Deutschland

Brugger, Ralph (2005): *IT-Projekte strukturiert realisieren - Situationen analysieren, Lösungen konzipieren - Vorgehen systematisieren, Sachverhalt visualisieren - UML und EPKs nutzen*, 2 Auflage, Vieweg+Teubner, Wiesbaden, Deutschland

Drath, Rainer; Grimm, Björn; Hundt, Lorenz; Keibel, Andreas; Lips, Steffen; Lüder, Arndt; Schleipen, Miriam; Weidemann, Dirk (2009): *Datenaustausch in der Anlagenplanung mit AutomationML*, 1. Auflage, Springer-Verlag, Wallstadter Str. 59, 68526 Ladenburg, Deutschland

Eigner, Martin; Roubanov, Daniil; Zafirov, Radoslav (Hrsg.) (2014): *Modellbasierte Virtuelle Produktentwicklung*, Springer Vieweg, Kaiserslautern, Deutschland

Fleischmann, Alber; Oppl, Stefan; Schmidt, Werner; Stary, Christian (2018): *Ganzheitliche Digitalisierung von Prozessen - Perspektivenwechsel - Design Thinking - Wertegeleitete Interaktion*, Springer Vieweg, Wiesbaden, Deutschland

Früh, K.; Maier, U.; Schaudel, D. (2015): *Handbuch der Prozessautomatisierung - Prozessleittechnik für verfahrenstechnische Anlagen*, 5. Auflage, DIV Deutscher Industrieverlag GmbH, Arnulfstraße 124, 80636 München, Deutschland

Gischel, Bernd (2016): *EPLAN Electric P8 Reference Handbook*, 4 Auflage, Carl Hanser Verlag, Lünen, Deutschland

Gruhn, Volker; Pieper, Daniel; Röttgers, Carsten (2006): *MDA® - Effektives Software-Engineering mit UML 2® und Eclipse TM*, Springer, Heidelberg, Deutschland

Guttman, Michael; Parodi, John (2007): *Real-Life MDA - Solving Business Problems with Model Driven Architecture*, Morgan Kaufmann Publisher, 500 Sansome Street, Suite 400, San Francisco, CA 94111, USA

Literaturverzeichnis

Harold, Elliotte; Means, Scott (2004): *XML in a Nutshell*, 3 Auflage, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA

Helmus, Frank (2003): *Anlagenplanung - Von der Anfrage bis zur Abnahme*, WILEY-VCH Verlag, Osnabrück, Deutschland

Klein, Thorsten (2016): *Agiles Engineering im Maschinen- und Anlagenbau (Dissertation)*, München

Klein, Ulf; CAE Expert Group (2018): *Handbuch G8 Schaltplangenerator*, 2.7.3 Auflage

Kloth, Christoph (2010): *Systemgestaltung im Broadcast Engineering - Prozessorientierte Konzeption integrierter Fernsehproduktionssysteme*, 1 Auflage, Vieweg+Teubner, Illmenau, Deutschland

Lindemann, Udo (2009): *Methodische Entwicklung technischer Produkte - methoden flexibel und situationsgerecht anwenden*, 3 Auflage, Springer, Boltzmannstraße 15, 85748 Garching, Deutschland

Martin, James (1997): *Systems Engineering Guidebook - A Process for Developing Systems and Products*, CRC Press LCC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431

Sattler, Klaus; Kasper, Werner (2000): *Verfahrenstechnische Anlagen - Planung, Bau und Betrieb*, WILEY-VCH Verlag, 69469 Weinheim, Deutschland

Stachowiak, Herbert (1973): *Allgemeine Modelltheorie*, Springer, Berlin -- Paderborn, Deutschland

Topole, Klaus (2018): *Grundlagen der Anlagenplanung - Einstieg in den Anlagenbau mit zahlreichen Praxisbeispielen*, Springer Vieweg, Dortmund, Deutschland

Weber, Klaus (2016): *Engineering verfahrenstechnischer Anlagen, Praxishandbuch mit Checklisten und Beispielen*, 2. Auflage, Springer Vieweg, Dresden, Deutschland

Winzer, Petra (2016): *Generic Systems Engineering - Ein methodischer Ansatz zur Komplexitätsbewertung*, 2 Auflage, Springer Vieweg, Wuppertal, Deutschland

Wissenschaftliche Artikel (1)

Bahill, A.; Gissing, B. (1998): *Re-evaluating Systems Engineering Concepts Using Systems Thinking*, in: IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, 28/1998, IEEE, S. 516 - 527

Online-Quellen (22)

Anderl; Nattermann; Rollmann (o.J.): *plmportal.org*

<https://www.plmportal.org/de/forschung-detail/das-w-modell-systems-engineering-in-der-entwicklung-aktiver-systeme.html> [Stand: 31.08.2019]

Aucotec AG (o.J.): *Hannover Messe*

<https://donar.messe.de/exhibitor/hannovermesse/2018/V836467/engineering-base-essentials-ger-484640.pdf> [Stand: 06.11.2019]

Brandes, Ulrik; Eiglsperger, Markus; Lerner, Jürgen (o.J.): *GraphML Primer*

<http://graphml.graphdrawing.org/primer/graphml-primer.html> [Stand: 15.11.2019]

Literaturverzeichnis

Den Haan, Johan (2008): *theenterprisearchitect.eu*

<http://www.theenterprisearchitect.eu/blog/2008/01/16/mda-model-driven-architecture-basic-concepts/>
[Stand: 07.09.2019]

Döppler.Team GmbH (o.J.): *leanmagazin.de*

<https://www.leanmagazin.de/lexikon/36-black-box-methode.html> [Stand: 03.09.2019]

Glinz, Martin (2005): *Universität Zürich, uzh.ch*

https://files.ifi.uzh.ch/rrerg/amadeus/teaching/courses/inf_II_fs10/inf_II_kapitel_02.pdf [Stand: 31.08.2019]

GraphML Team (2007): *graphml.graphdrawing.org*

<http://graphml.graphdrawing.org/index.html> [Stand: 15.11.2019]

INCOSE (o.J.): *incose.org*

<https://www.incose.org/about-systems-engineering> [Stand: 28.08.2019]

Kempa, Martin; Mann, Zoltán (2005): *gi.de*

<https://gi.de/informatiklexikon/model-driven-architecture/> [Stand: 08.09.2019]

Langenscheidt (o.J.): *langenscheidt.com*

<https://de.langenscheidt.com/englisch-deutsch/claim> [Stand: 15.08.2019]

metior GmbH (o.J.): *metior engineering zeitgeist*

<http://www.metior.at/home/> [Stand: 29.10.2019]

Object Management Group (2014): *omg.org*

<https://www.omg.org/cgi-bin/doc?ormsc/14-06-01> [Stand: 08.09.2019]

Object Management Group (o.J.): *omg.org*

<https://www.omg.org/mda/> [Stand: 07.09.2019]

PMH (2018): *projektmanagementhandbuch.de*

<https://www.projektmanagementhandbuch.de/handbuch/projektrealisierung/claim-management/> [Stand: 17.08.2019]

Prof. Dr. Müller-Stewens, Günter (o.J.): *wirtschaftslexikon.gabler.de*

<https://wirtschaftslexikon.gabler.de/definition/top-down-prinzip-49846> [Stand: 04.09.2019]

Stobitzer, Christian (o.J.): *oop-uml.de*

<http://www.oop-uml.de/uml-uebersicht.php> [Stand: 06.09.2019]

Theis, Thomas (o.J.): *www.it-treff.de*

<https://www.it-treff.de/it-lexikon/vba-programmiersprache> [Stand: 15.11.2019]

Tilkov, Stefan (2012): *innoc.com*

<https://www.innoc.com/de/articles/2012/07/domain-specific-languages/> [Stand: 06.09.2019]

Truyen, Frank (2006): *omg.org*

https://www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf [Stand: 07.09.2019]

Vertical Media (o.J.): *gruenderszene.de*

https://www.gruenderszene.de/lexikon/begriffe/bottom-up-ansatz?interstitial_click [Stand: 04.08.2019]

Literaturverzeichnis

VGB Power Tech e.V. (2019): *vgb.org*

https://www.vgb.org/db_kks.html [Stand: 03.11.2019]

VGB Power Tech e.V. 1 (2019): *vgb.org*

https://www.vgb.org/aufbau_anwendung_kraftwerk_kennzeichensystem.html [Stand: 4.11.2019]

Normen (1)

(VDI-EKV), VDI-Gesellschaft (Hrsg.) (2004): *VDI 2206: VDI 2206 - Entwicklungsmethodik für mechatronische Systeme*

ABBILDUNGSVERZEICHNIS

Abbildung 1: Planungsphasen in der Prozessindustrie, Quelle: In Anlehnung an Draht (2009), S.20	6
Abbildung 2: Phasenmodell für Realisierung von Anlagen, Quelle: In Anlehnung an Weber (2016), S. 4.. 7	
Abbildung 3: Ausschnitt eines typischen R&I-Fließschema, Quelle: Eigene Darstellung	11
Abbildung 4: Prinzipschaltbild einer Elektrotechnik/EMSR-Anlage, Quelle: In Anlehnung an Helmus (2003), S. 152.	14
Abbildung 5: Auszug einer Messkreisliste, Quelle: Eigene Darstellung	20
Abbildung 6: Regelschema einer Dampf-Durchflussregelung, Quelle: Weber (2016), S. 360.	23
Abbildung 7: Schichten von Metamodell zu Realität, Quelle: In Anlehnung an Gruhn/Pieper/Röttgers (2006), S. 85.	30
Abbildung 8: Engineering-Disziplinen im Zusammenspiel mit Systems Engineering, Quelle: in Anlehnung an Winzer (2016), S. 6.	33
Abbildung 9: Grafische Darstellung des SIMILAR-Prozesses, Quelle: In Anlehnung an Bahill/Gissing (1998), S. 518.	35
Abbildung 10: Darstellung des V-Modells, Quelle: In Anlehnung an VDI 2206 (2004), S. 29.	36
Abbildung 11: Die Erweiterung des V-Modells ergibt das W-Modell, Quelle: In Anlehnung an Anderl/Nattermann/Rollmann (o.J.), Online-Quelle [31.08.2019]	37
Abbildung 12: Grafische Darstellung des Münchner Vorgehensmodell, Quelle: In Anlehnung an Lindemann (2009), S. 47.	39
Abbildung 13: Aufbau der MDA-Modelle und Transformationen, Quelle: Eigene Darstellung	43
Abbildung 14: Firmenlogo metior GmbH inklusive Slogan, Quelle: metior GmbH (o.J.), Online-Quelle [29.10.2019].	46
Abbildung 15: Vorgehensmodell des ausgearbeiteten Workflows, Quelle: Eigene Darstellung	47
Abbildung 16: R&I-Fließschema-Ausschnitt der betrachteten Kreise, Quelle: Eigene Darstellung	51
Abbildung 17: Messungen der Messkreisliste untereinander dargestellt für bessere Übersicht, Quelle: Eigene Darstellung	52
Abbildung 18: Ventil der Stellenliste untereinander dargestellt für bessere Übersicht, Quelle: Eigene Darstellung	52
Abbildung 19: Auszug der Vorgaben bezüglich Kennzeichnungssystem, Quelle: Eigene Darstellung	53
Abbildung 20: Vorgabe zum Aufbau der Stromlaufpläne, Quelle: Eigene Darstellung	53
Abbildung 21: Mögliche Ausführungsrichtlinie bezüglich der schrankinternen Aderfarben, Quelle: Eigene Darstellung	54
Abbildung 22: UML-Objekt <i>Verbindungselement</i> , Quelle: Eigene Darstellung	55

Abbildungsverzeichnis

Abbildung 23: Verwendete Verbindungstypen im Metamodell, Quelle: Eigene Darstellung	55
Abbildung 24: Ausschnitt des Metamodells zum Workflow, Quelle: Eigene Darstellung.....	56
Abbildung 25: Modellausschnitt, der ausschließlich die funktionale Struktur der Anlage zeigt (nicht vollständig verbunden), Quelle: Eigene Darstellung	58
Abbildung 26: Darstellung der Druckmessstelle H1NDC01CP001 inkl. Verbindungen, Quelle: Eigene Darstellung	59
Abbildung 27: Darstellung der Stellstelle H1GHE10AA103 inkl. Verbindungen, Quelle: Eigene Darstellung	60
Abbildung 28: Darstellung der verwendeten Steuerungskomponenten inkl. Verbindungen, Quelle: Eigene Darstellung	61
Abbildung 29: Auflistung der <i>EB</i> -Datenbankbereiche, Quelle: Eigene Darstellung	63
Abbildung 30: Gliederung eines <i>Engineering Base</i> Projekts im Objekt-Tree des <i>EB-Explorers</i> , Quelle: Eigene Darstellung	64
Abbildung 31: Strukturierungseinheiten in <i>EPLAN</i> , Quelle: Eigene Darstellung.....	66
Abbildung 32: Ansicht einiger Navigatoren in <i>EPLAN</i> , Quelle: Eigene Darstellung.....	67
Abbildung 33: <i>EPLAN</i> -Makrokasten-Dialog inkl. Schaltung, Quelle: Eigene Darstellung.....	68
Abbildung 34: Strukturierung des Arbeitsprojekts (Ansicht im Eigenschaftsdialog), Quelle: Eigene Darstellung	68
Abbildung 35: Geräte/Kabel des Stellstellentypicals in <i>EB</i> , Quelle: Eigene Darstellung	70
Abbildung 36: Geräte/Kabel des Temperaturmessstellentypicals in <i>EB</i> , Quelle: Eigene Darstellung.....	71
Abbildung 37: Geräte/Kabel des Druckmessstellentypicals in <i>EB</i> , Quelle: Eigene Darstellung	71
Abbildung 38: <i>EB-Assistent Typical Manager</i> zum Generieren von Stellenplänen, Quelle: Eigene Darstellung	73
Abbildung 39: <i>EB-Typical Manager</i> Optionen-Dialog (Register <i>Allgemein</i> und <i>Verhalten</i>), Quelle: Eigene Darstellung	74
Abbildung 40: Einzel-Makros der Messstellentypicals für den Schaltplangenerator <i>G8</i> , Quelle: Eigene Darstellung	76
Abbildung 41: <i>Basisseite</i> für das <i>G8-Typical</i> der Messstellen, Quelle: Eigene Darstellung	77
Abbildung 42: <i>G8-Stammdaten</i> mit Darstellung der Kombinationen und den verwendeten Makros, Quelle: Eigene Darstellung	78
Abbildung 43: <i>G8-Stammdaten</i> Makros mit zugeordneten <i>Plug-Socket</i> -Abhängigkeiten, Quelle: Eigene Darstellung	79
Abbildung 44: <i>G8-Stammdaten</i> -Übersichtsdarstellung der Makro-Parameter, Quelle: Eigene Darstellung	79

Abbildungsverzeichnis

Abbildung 45: G8-Stammdaten-Seitenkennzeichnung, Quelle: Eigene Darstellung	80
Abbildung 46: G8-Projektdatei, Aufteilung der Engineering-Liste in Seiten-, Typical-, Strukturkennzeichen- und Parameter-Sektionen, Quelle: Eigene Darstellung	81
Abbildung 47: G8-Projektdatei, Makro-Liste vor dem ersten <i>Erzeugen</i> aus der Engineering-Liste, Quelle: Eigene Darstellung	81
Abbildung 48: G8-Projektdatei, synchronisierte Inhalte aus den Makros, Quelle: Eigene Darstellung	82
Abbildung 49: Darstellung eines simplen Graphen als GraphML-File (links) und in grafischer Darstellung (rechts), Quelle: Eigene Darstellung	84
Abbildung 50: Buttons für die Transformationsvorbereitung nach <i>EB</i> bzw. <i>EPLAN</i> , die zum Start der <i>Microsoft-Excel</i> -Makros verwendet werden, Quelle: Eigene Darstellung	85
Abbildung 51: <i>EB</i> -Assistent <i>Elemente importieren und aktualisieren</i> , Quelle: Eigene Darstellung	95
Abbildung 52: Typen-Mapping für Funktionstypen im Assistenten <i>Elemente importieren und aktualisieren</i> , Quelle: Eigene Darstellung	95
Abbildung 53: Stellenstellenbezogenes Kabelimport-Mapping, Quelle: Eigene Darstellung	96
Abbildung 54: Angelegte Funktionen und Geräte durch die ausgelesenen Modellinformationen, Ansicht im <i>EB-Explorer</i> , Quelle: Eigene Darstellung	96
Abbildung 55: <i>EB</i> -Assistent <i>Typical Manager</i> und darunter Meldungsanzeige nach erfolgreichem Kopiervorgang, Quelle: Eigene Darstellung	97
Abbildung 56: Ansicht der <i>Betriebsmittel</i> -, <i>Funktions</i> - und <i>Dokumenten</i> -Ordner nach erfolgtem Generierungsprozess, Quelle: Eigene Darstellung	97
Abbildung 57: <i>Gerätezuordnungsassistent</i> mit den geladenen IO-Kanälen, links die freien, rechts die der IO-Karte, Quelle: Eigene Darstellung	98
Abbildung 58: Ausschnitt der befüllten Engineering-Liste des G8, Quelle: Eigene Darstellung	99
Abbildung 59: Makro-Liste des G8 vor der Plangenerierung des Seitenmakros, Quelle: Eigene Darstellung	99
Abbildung 60: Seitenstruktur der eingefügten Pläne des Seitenmakros, erstellt aus dem Schaltplangenerator G8, Quelle: Eigene Darstellung	100

QUELLTEXTVERZEICHNIS

Quelltext 1: Auszug aus der Prozedur, die durch den <i>EB</i> -Button aufgerufen wird, Quelle: Eigene Darstellung	86
Quelltext 2: Suchvorgang zwischen zwei <i>nodes</i> , die durch eine <i>edge</i> verbunden sind, Quelle: Eigene Darstellung	87
Quelltext 3: Auslesen der Attributwerte, Links: Attribute in XML-Format, Rechts: Code zum Auslesen, Quelle: Eigene Darstellung	88
Quelltext 4: Navigation zur funktionellen Zuordnung des Schaltschranks inkl. Empty-Abfrage, Quelle: Eigene Darstellung	88
Quelltext 5: Auslesen und auftrennen der Kabel- bzw. Schlauchattribute und Verknüpfung zur Artikelnummer, Quelle: Eigene Darstellung	89
Quelltext 6: <i>select-case</i> -Statement für Messstellen-Stereotypen zur Gerätedefinition, Quelle: Eigene Darstellung	90
Quelltext 7: Verbundenes Modellobjekt <i>Feldgerät</i> ausfindig machen, Quelle: Eigene Darstellung	91
Quelltext 8: Auslesen der Digitalsignale und Endschalter bei Stellstellen für <i>EPLAN</i> , Quelle: Eigene Darstellung	92

TABELLENVERZEICHNIS

Tabelle 1: Unterschiede zwischen Metamodell und Modell, Quelle: In Anlehnung an Brugger (2005), S. 494.....	29
Tabelle 2: Ermittelte Voraussetzungen für die Verwendung des Workflows, Quelle: Eigene Darstellung	49
Tabelle 3: Ermittelte Anforderungen an den Workflow, Quelle: Eigene Darstellung.....	50
Tabelle 4: Verifizierungstabelle inklusive geschätztem Erfüllungsgrad, Quelle: Eigene Darstellung	102

ABKÜRZUNGSVERZEICHNIS

CAE	<i>Computer Aided Engineering</i>
CIM	<i>Computation Independent Model</i>
DSL	<i>Domain Specific Language</i>
EB	<i>Engineering Base</i>
EMSR	<i>Elektro-, Mess-, Steuer- und Regelungstechnik</i>
FAT	<i>Factory Acceptance Test</i>
KKS	<i>Kraftwerk-Kennzeichensystem</i>
MDA	<i>Model Driven Architecture</i>
PIM	<i>Platform Independent Model</i>
PLS	<i>Prozessleitsystem</i>
PLT	<i>Prozessleittechnik</i>
PM	<i>Platform Model</i>
PSM	<i>Platform Specific Model</i>
R&I-Fließschema	<i>Rohrleitungs- und Instrumentierungs-Fließschema</i>
SPS	<i>Speicher Programmierbare Steuerung</i>
SysML	<i>System Modeling Language</i>
UML	<i>Unified Modeling Language</i>
VBA	<i>Visual Basic for Applications</i>
XML	<i>Extensible Markup Language</i>

ANHANG 1: WORKSHOP ENGINEERING-PROZESS

- Thema: Ablauf einer typischen Planung in der Prozessindustrie
Vom Kundenkontakt bis zur Anlagenabnahme → Fokus auf Hardware/Software Engineering
- Teilnehmer: Ing. Gruber Dominik, BSc; HW-Junior Lead Engineer (6 Jahre einschlägige Berufserfahrung in HW-Engineering)
Ing. Lex Christoph, BSc; SW-Projektleiter (11 Jahre einschlägige Berufserfahrung in HW und SW-Engineering und Projektleitung)
Ing. Tatzler Viktor; HW-Projektleiter (21 Jahre einschlägige Berufserfahrung in HW-Engineering und Projektleitung)
- Datum: 08.07.2019
- Ort: metior GmbH, Graz

Inhalt:

Kontaktaufnahme (Kunde → (Mitarbeiter) → MT)

- Mündlich / schriftlich / Ausschreibung

Übergabe Basisunterlagen

- Pflichtenheft
- R&I, Werkstandard, diverse Listen z.B. Verbraucher-, Instrumentenlisten, Funktionsbeschreibung (im Idealfall)
- Terminplan
- Anlagenlayouts

Grobe technische Ausarbeitung der Basisunterlagen

- Kalkulation der benötigten I/Os, Kabellängen/-Querschnitte, Engineeringstunden (HW/SW), Vorabanfragen bei Montagefirmen/Schrankbau (technisch/zeitlich)

Angebot ausarbeiten

- Für Abgabe beim potenziellen Kunden

Durchsicht durch Kunde mit Rückfragen

Preisverhandlung

Kaufmännische und technische Vertretung

Zahlungsbedingungen (Beträge, Zeitpunkt)

Vergabe (Bestellvorgang durch Kunden)

Auftragsbestätigung (durch Lieferanten)

Kick-Off-Meeting extern

Anhang 1: Workshop Engineering-Prozess

- Übergabe *finale* Basisunterlagen
- Vorstellung Projektmitarbeiter (Ansprechpartner pro Disziplin, Organigramm)
- Klären/Besprechen technischer Details (Details Funktionsbeschreibung, Layouts, Terminplan...)
 - Vertretung durch Techniker

Ressourcenplanung intern

- Arbeitspakete aufteilen (z.B. SW/HW)
- Infrastruktur schaffen (Projektteam, IT, Engineering-Tool-Setup)
- Kick-Off intern (abhängig von Projektgröße variiert Personenanzahl)

Detailplanung pro Arbeitspaket

- Basisunterlagen aufbereiten
 - Listen aufbereiten → Technisch gleiche Messungen/Verbraucher typisieren und standardisieren (Typicals zuteilen)
 - Schnittstellen definieren (zu z.B. Package-Lieferanten)
 - Systemkonzept erarbeiten → Bestellung AT-Komponenten möglich und notwendig (Lieferzeiten)
- Beschaffung von AT-Komponenten
 - Als Beistellung (Lieferant an Kunden)
 - Für interne Vorabtests (Schnittstellentests)
 - Zeitkritisch für z.B. Schrankbau
- Typicals erstellen
 - Kundenvorlagen einbeziehen/einfordern
 - Recherche bzw. anfordern der Geräteunterlagen → Aufgrund von z.B. Messkreisliste Geräte definieren, danach Internetrecherche oder beim Kunden anfordern
 - Technische Ausarbeitung (z.B. Sicherungsgröße (HW) oder Verriegelung(SW))
 - Interdisziplinärer Abgleich (HW-Typical mit SW-Typical vergleichen und abstimmen)
- Freigabe durch Kunden
 - Anpassungen bei Kundenwünschen
 - Ersatzteilliste für Kunden (wenn gefordert)
- Montagedokumentation für finale Anfrage der Montage
 - Schrankbau; Montage im Feld
 - Verhandeln / Bestellung
- Bulk-engineering (SW/HW)
 - Vervielfachen der freigegebenen Typicals (händisch / automatisiert)
- Interner Prüflauf durch AP-Verantwortliche
 - Korrekturen des Engineerings
 - Revisionsmanagement → Schrank-/Montagedokumentation
- Laufend Controlling
 - Kaufmännisch (Ressourcen)
 - Technisch (Projektänderungen → Prozess → Geräte/Visu-Bilder → IO)
 - Änderungen bis zu definiertem Freezing-Point (davor Kundenwünsche) *inkludiert* → danach Claim-Management
 - Dokumentation der Änderungen mitführen
 - Bei vielen Änderungen → möglichen Terminverzug bekanntgeben
- Übergabe an Schrankbau
 - Dokumentation und Komponenten (-listen)
 - Schrankabnahme vor Auslieferung zur Anlage
 - Funktionstest SW/HW (FAT)
 - Fallen beim Schrankbau bzw. FAT Änderungen an müssen diese dokumentiert werden (Rotkorrektur) → Kann vor Schrankübergabe an Kunden in Engineeringsystem eingearbeitet werden
- Übergabe an Montage
 - Dokumentation (Pläne, Layouts 2D/3D, wenn notwendig Listen)

Anhang 1: Workshop Engineering-Prozess

Inbetriebnahme der Anlage

- Unterstützung der Inbetriebnahme bei technischen Rückfragen

Enddokumentation erstellen

- Betriebshandbuch erstellen (SW)
 - Funktionsbeschreibung (Bedienkonzept)
- As-built Dokumentation (HW)
 - Einarbeiten der Rotkorrektur (Anmerkungen Montage, Inbetriebnahme)
- Technische Unterlagen
- Prüfprotokolle
- SW-Sicherung/ECAD-Dateien

Betreuung bis zur Abnahme der Anlage durch Kunden

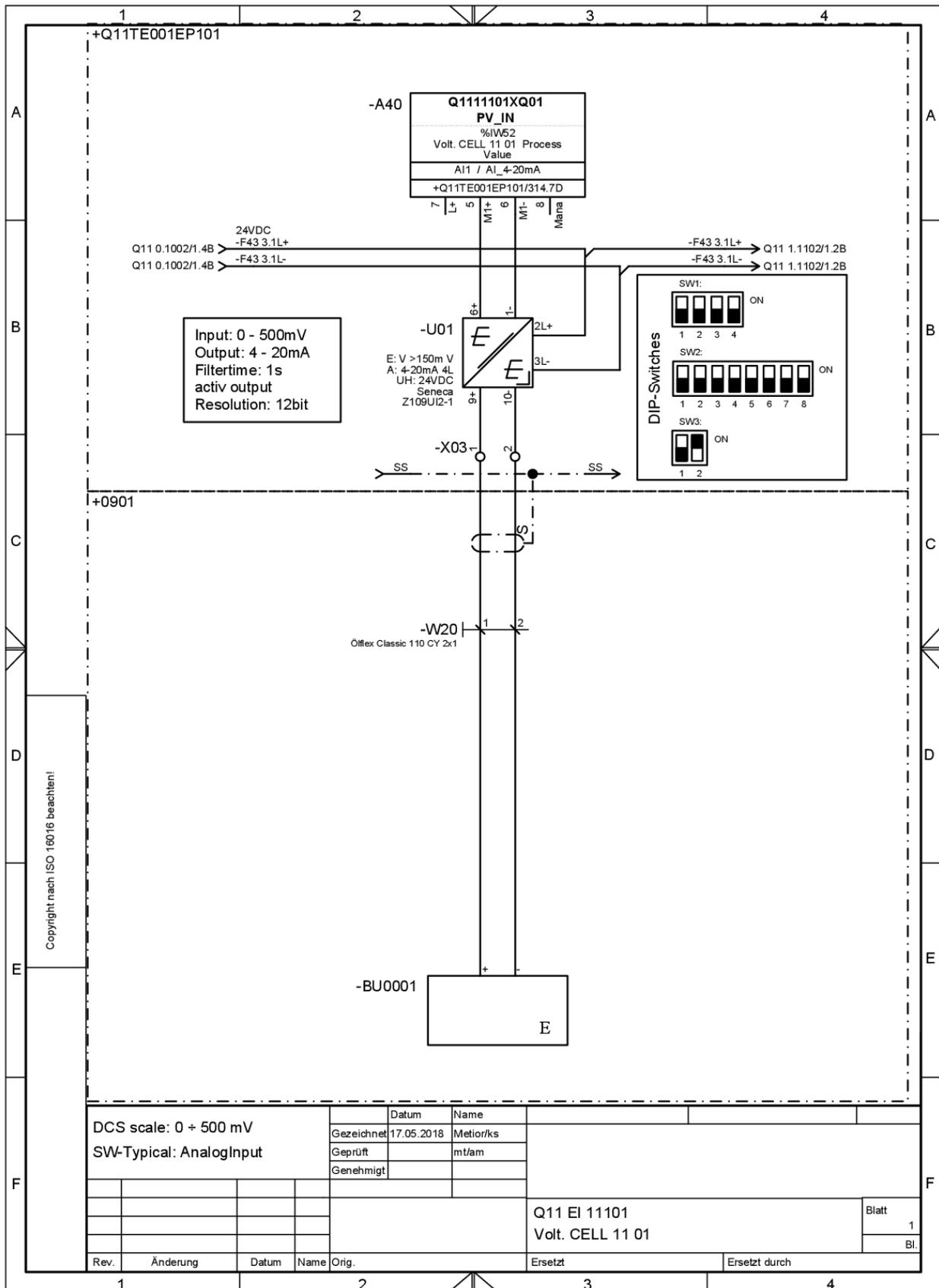
Gewährleistung

Schlussrechnung

Lessons-learned

- Alle Beteiligten (Leistungspositionen)
- Kaufmännisch/technisch

ANHANG 2: STELLENPLAN



ANHANG 3: VOLLSTÄNDIGE AUFLISTUNG ALLER METAMODELLOBJEKTE

Bezeichnung	Abgeleitet von	Attribute	Komposition mit	Verwendungsbeziehung
Anlage	-	Langtext	-	<<belongs to>> Örtlichkeit
Teilanlage	-	Langtext	Anlage	-
Stelle	-	Langtext HW-Typical SW-Typical	Teilanlage	Klemmleiste
Verbraucherstelle	Stelle	(geerbt)	-	-
Messstelle	Stelle	(geerbt)	-	Signal
Stellstelle	Stelle	(geerbt)	-	Signal Feldgerät-Aktor Endschalter
Druck	Messstelle	(geerbt)	-	Feldgerät-Messung
Temperatur	Messstelle	(geerbt)	-	Feldgerät-Messung
Dichte	Messstelle	(geerbt)	-	Feldgerät-Messung
Niveau	Messstelle	(geerbt)	-	Feldgerät-Messung
Feldgerät-Messung	-	Anschlussprinzip Messprinzip Hilfsenergie	-	Signal Verbindungselement Klemme Örtlichkeit
Feldgerät-Aktor	-	Hilfsenergie Typ	-	Signal Verbindungselement Klemme Örtlichkeit Magnetventil Endschalter
Endschalter	-	-	-	Klemme

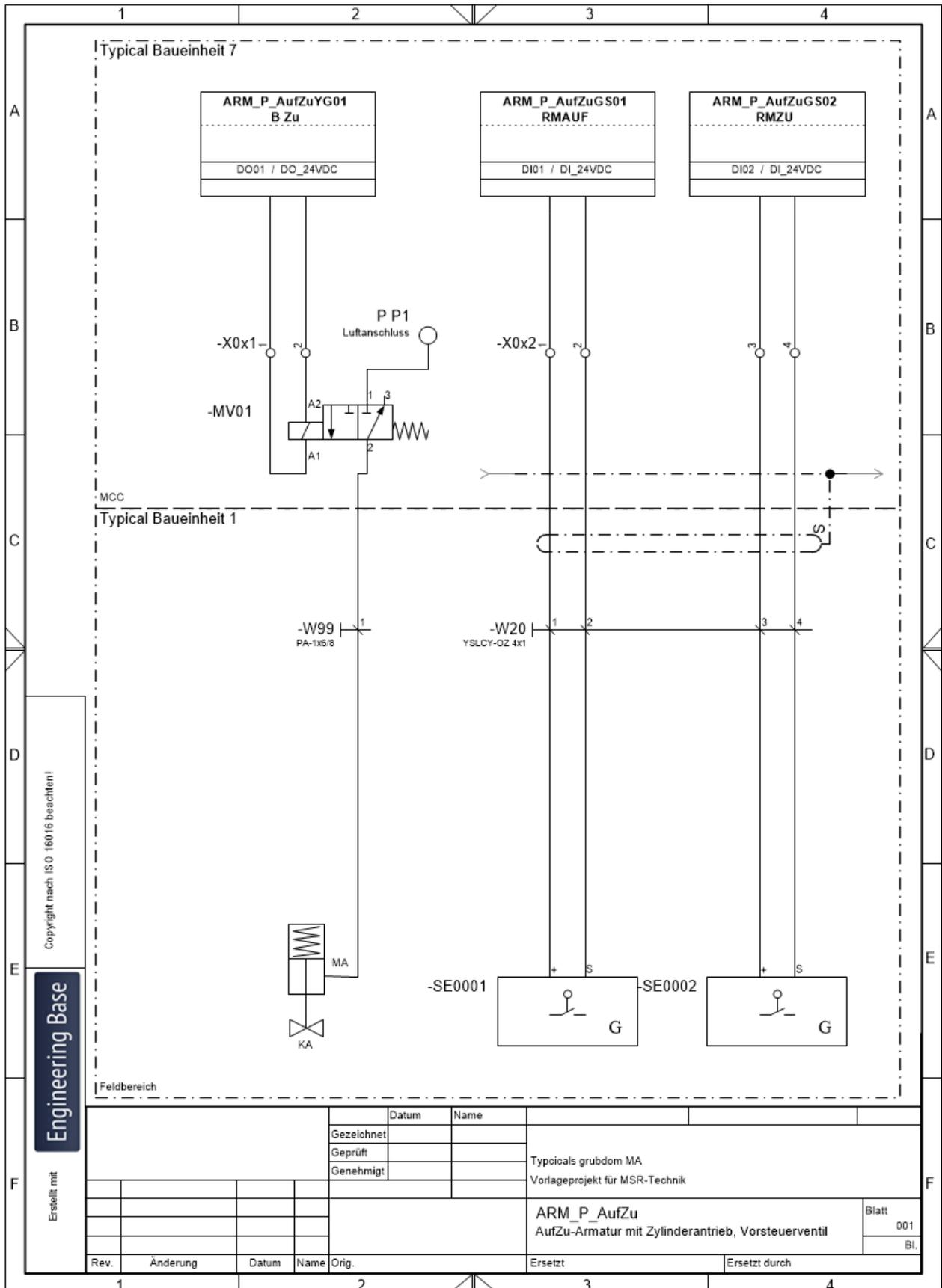
Anhang 3: Vollständige Auflistung aller Metamodellobjekte

				Örtlichkeit
Verbindungselement	-	Aderanzahl Typ Querschnitt Länge	-	-
Kabel	Verbindungselement	(geerbt)	-	-
Luftschlauch	Verbindungselement	(geerbt)	-	-
Energieversorgung	-	Spannungsebene	-	Einspeisung Sicherungseinrichtung Magnetventil Klemmleiste
Einspeisung	-	Trenneinrichtung Polzahl	-	-
Sicherungseinrichtung	-	Charakteristik Polzahl	-	-
Magnetventil	-	Wege Spannungsebene	-	Signal Verbindungselement Klemme
Klemmleiste	-	Copy-Index	-	
Klemme	-	Typ Querschnitt	Klemmleiste	-
Steuerung	-	Typ Kommunikation	-	Dezentrale Peripherie IO-Karte
Dezentrale Peripherie	-	-	-	Anschaltung
Anschaltung	-	Kommunikation	-	IO-Karte
IO-Karte	-	Adressbereich IO-Anzahl IO-Typ	-	
Digitale Eingangskarte	IO-Karte	(geerbt)	-	Signal

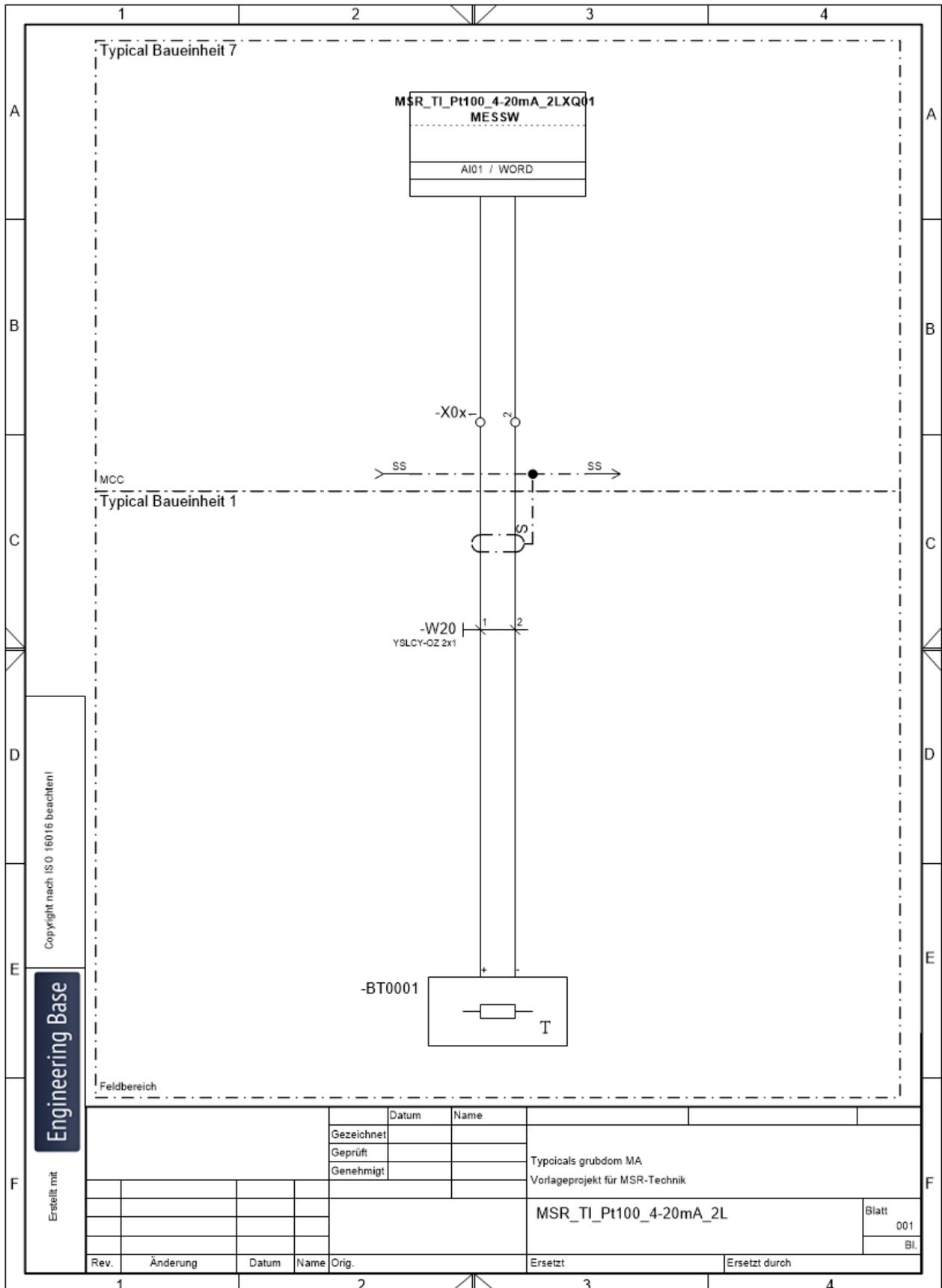
Anhang 3: Vollständige Auflistung aller Metamodellobjekte

Digitale Ausgangskarte	IO-Karte	(geerbt)	-	Signal
Analoge Eingangskarte	IO-Karte	(geerbt)	-	Signal
Analoge Ausgangskarte	IO-Karte	(geerbt)	-	Signal
Signal	-	Signaltyp	-	-
Analogsignal	Signal	(geerbt)	-	-
Digitalsignal	Signal	(geerbt)	-	-
Örtlichkeit	-	Beschreibung	-	<<belongs to>> Endschalter Einspeisung Steuerung Sicherungseinrichtung Feldgerät-Messung Feldgerät-Aktor Dezentrale Peripherie Magnetventil Klemmleiste IO-Karte

ANHANG 4.1: EB-TYPICAL STELLSTELLE ARM_P_AUFZU



ANHANG 4.2: EB-TYPICAL MESSSTELLE MSR_TI_PT100_4-20MA_2L



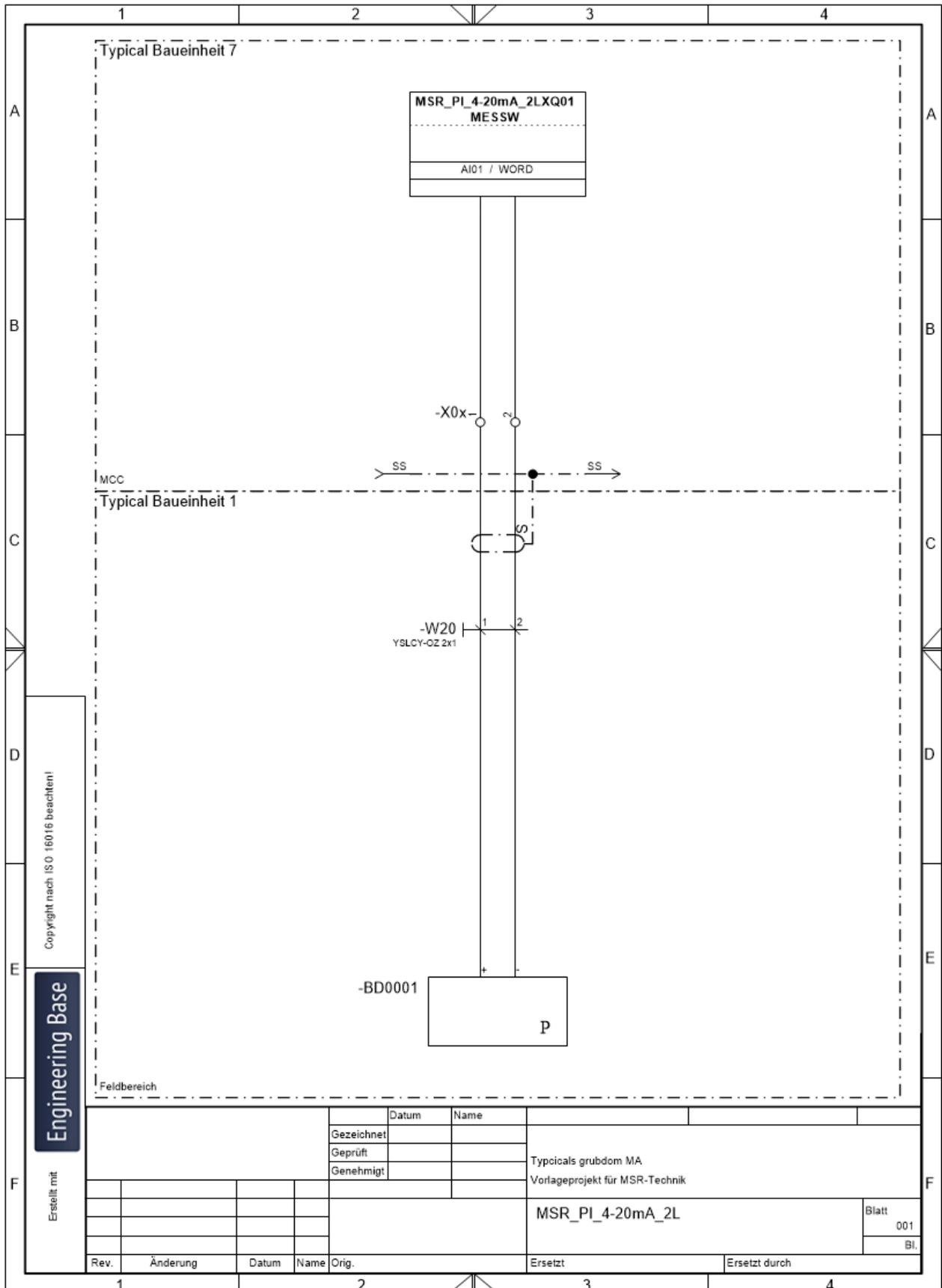
Copyright nach ISO 16016 beachten!

Engineering Base

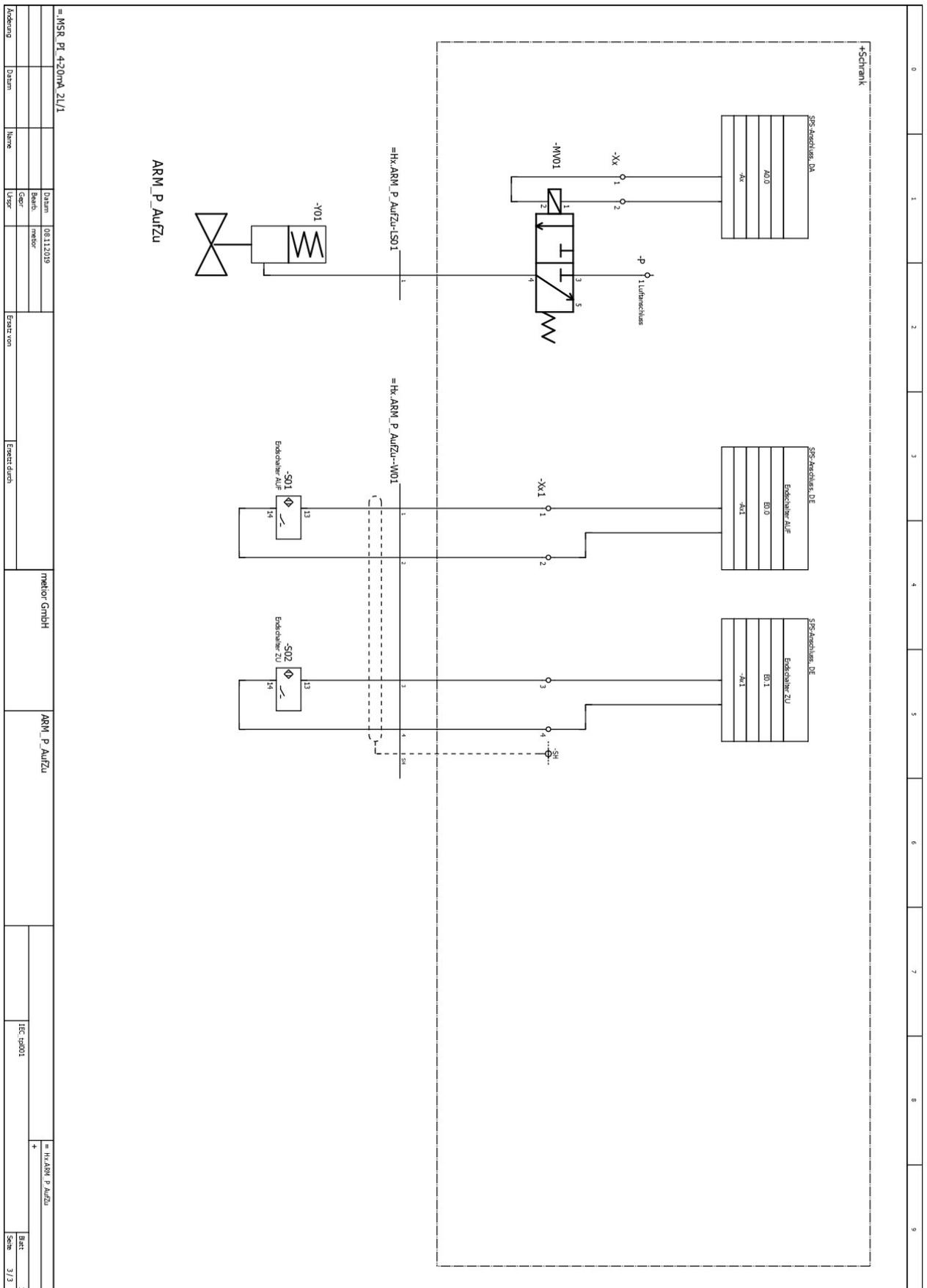
Erstellt mit

		Datum		Name			
		Gezeichnet					
		Geprüft				Typicals grubdom MA	
		Genehmigt				Vorlageprojekt für MSR-Technik	
						MSR_TI_Pt100_4-20mA_2L	
						Blatt 001	
						Bl.	
Rev.	Änderung	Datum	Name	Orig.	Ersetzt	Ersetzt durch	

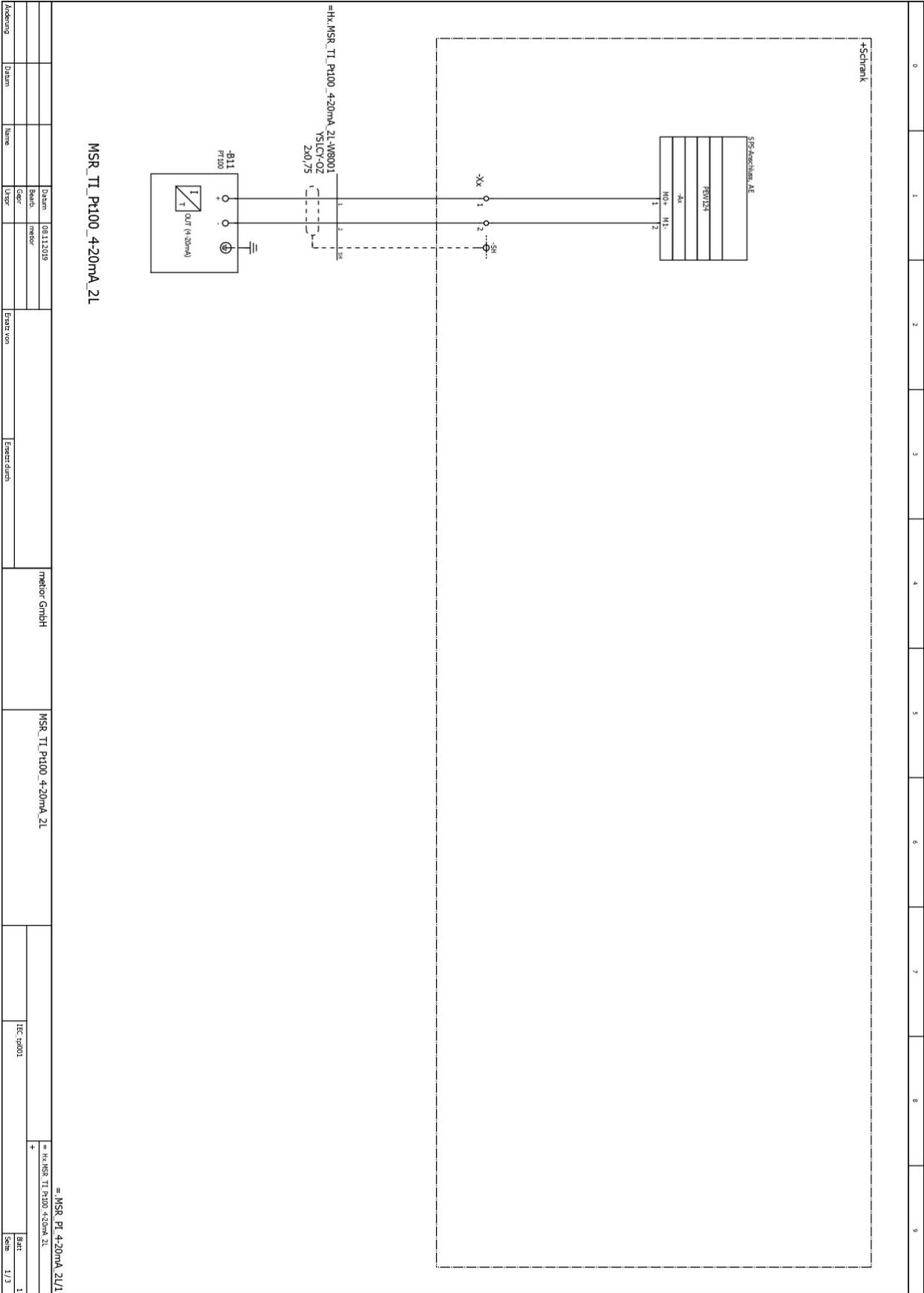
ANHANG 4.3: EB-TYPICAL MESSSTELLE MSR_PI_4-20MA_2L



ANHANG 5.1: EPLAN-TYPICAL STELLSTELLE ARM_P_AUFZU

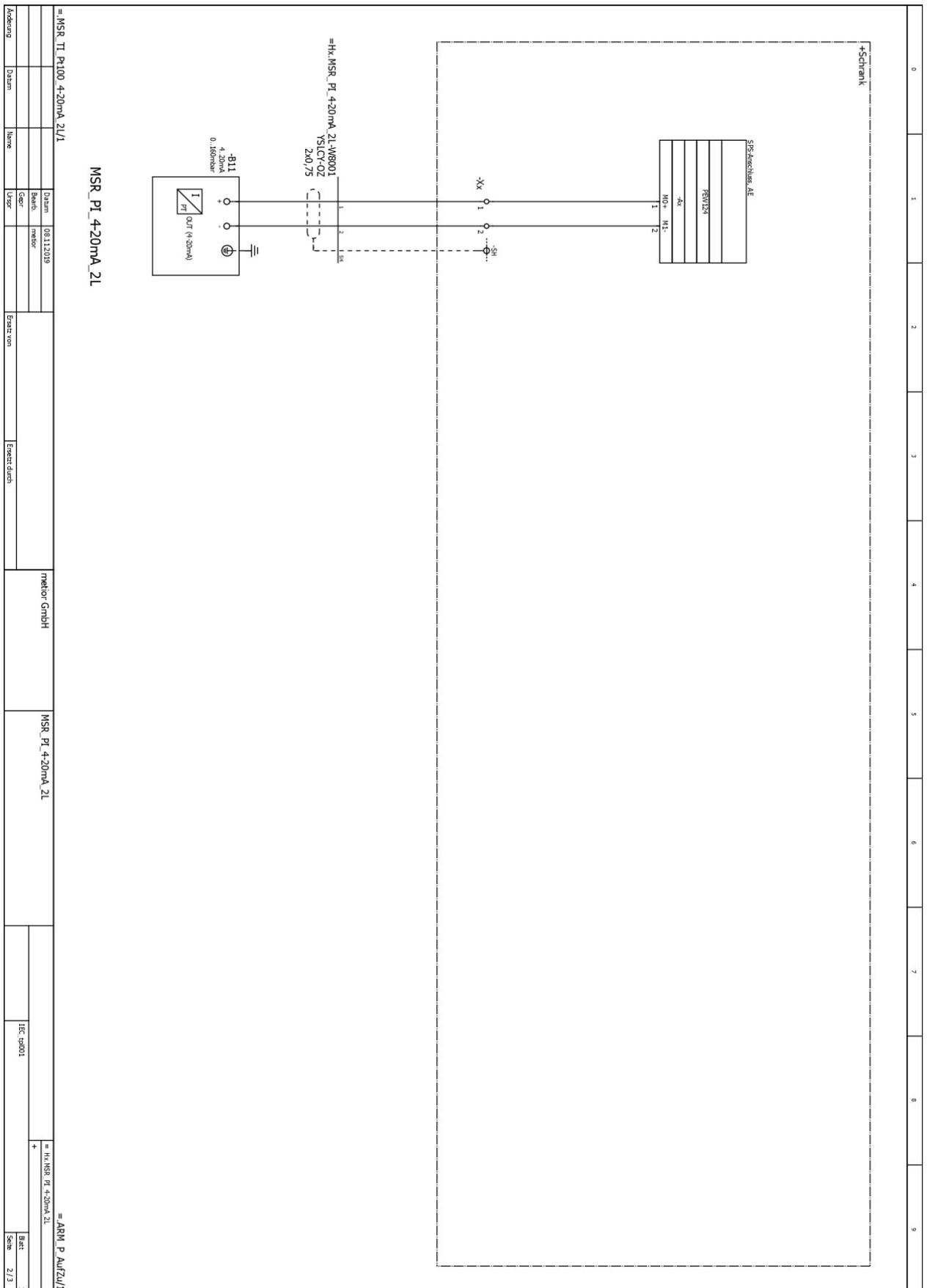


ANHANG 5.2: EPLAN-TYPICAL MESSSTELLE MSR_TI_PT100_4-20MA_2L



Adressierung	Datum	Name	Urspr	Datum	08.11.2019	Bearbte	mebner	Erstellt durch	mebner GmbH	MSR_TI_PT100_4-20mA_2L	IEC: ip001	+	= MSR_TI_PT100_4-20mA_2L	B Blatt	1

ANHANG 5.3: EPLAN-TYPICAL MESSSTELLE MSR_PI_4-20MA_2L



= MSR_TI_P100_4-20mA_2L/1		Datum		08.11.2019		Bearb.		meator	
Adressung		Datum				Gerät			
Name		Urspr				Ersatz von			
		Ersatz durch				meator GmbH		MSR_PI_4-20mA_2L	
								IEC 6001	
								= Hx.MSR.PI.4-20mA.2L	
								Blatt	
								Seite	
								2 / 3	

= ARM_P_AufZu/1

ANHANG 6: EINZEL-MAKROS DES EPLAN-STELLSTELLENTYPICALS

0									
<p>C:\S8\2\7\EPAN\2\Macro\SPS\FRAG_SPS_BIL_DA_ZM.ema Variante A Alpolig</p>									
1									
<p>SPS-Anschlus DA #<PlugRPs> = #F01(ochterDIBNK)</p>									
2									
<p>C:\S8\2\7\EPAN\2\Macro\SPS\FRAG_SPS_BIL_DZL_ZM.ema Variante A Alpolig</p>									
3									
<p>SPS-Anschlus DE #<PlugRPs> = #F01(ochterDIBNK)</p>									
4									
<p>C:\S8\2\7\EPAN\2\Macro\SPS\FRAG_DIG_Ventil.ema Variante A Alpolig</p>									
5									
<p>#<PlugRPs> P P1 Luftanschluss</p>									
6									
<p>=HX_ARM_P_AufZu_#(Schlauchname) =HX_ARM_P_AufZu_#(Kabelname)</p>									
7									
<p>-Y01 = #FEND1B(K) Disconnector Z1 = #FEND2B(K) Disconnector Z2</p>									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									
51									
52									
53									
54									
55									
56									
57									
58									
59									
60									
61									
62									
63									
64									
65									
66									
67									
68									
69									
70									
71									
72									
73									
74									
75									
76									
77									
78									
79									
80									
81									
82									
83									
84									
85									
86									
87									
88									
89									
90									
91									
92									
93									
94									
95									
96									
97									
98									
99									
100									

ANHANG 7: AUFLISTUNG DER MAKRO-PARAMETER

Parameter	Beschreibung	Makro
Kabel-Artikel	Artikelnummer des Kabels	FRAG_ANA_MEss.ema FRAG_ANA_MEss_Temp.ema FRAG_DIG_Vetnil.ema
Kabelname	BMK des Kabels	FRAG_ANA_MEss.ema FRAG_ANA_MEss_Temp.ema FRAG_DIG_Vetnil.ema
Schlauchartikel	Artikelnummer des Schlauchs	FRAG_DIG_Vetnil.ema
Schlauchname	BMK des Schlauchs	FRAG_DIG_Vetnil.ema
Einbauort	Örtlichkeit	BAS_4SPS.ema BAS_Ventil.ema
PIOTochterAEKana	Analogeingangskanal Kanalnummer	FRAG_SPS_Bit_AE_2AN.ema
PIOTochterAEBM	Analogeingangskanal BMK	FRAG_SPS_Bit_AE_2AN.ema
PSENSBM	Analogsensor BMK	FRAG_ANA_MEss.ema FRAG_ANA_MEss_Temp.ema
PKLAI	Analogtypical Klemmleistenbezeichnung	FRAG_ANA_MEss.ema FRAG_ANA_MEss_Temp.ema
PIOTochterDOKanal	Digitalausgangskanal Kanalnummer	FRAG_SPS_Bit_DA_2AN.ema
PIOTochterDOBM	Digitalausgangskanal BMK	FRAG_SPS_Bit_DA_2AN.ema
PIOTochterDI1Kanal	Digitaleingangskanal 1 Kanalnummer	FRAG_SPS_Bit_DI1_2AN.ema
PIOTochterDI1BMK	Digitaleingangskanal 1 BMK	FRAG_SPS_Bit_DI1_2AN.ema
PIOTochterDI2Kanal	Digitaleingangskanal 2 Kanalnummer	FRAG_SPS_Bit_DI2_2AN.ema
PIOTochterDI2BMK	Digitaleingangskanal 2 BMK	FRAG_SPS_Bit_DI2_2AN.ema
PKLDO	Digitalausgangsklemmleiste BMK	FRAG_DIG_Vetnil.ema

Anhang 7: Auflistung der Makro-Parameter

PKLDI	Digitaleingangsklemmleiste BMK	FRAG_DIG_Vetnil.ema
PMVBMK	Magnetventil BMK	FRAG_DIG_Vetnil.ema
PENDL1BMK	Endlagenschalter 1 BMK	FRAG_DIG_Vetnil.ema
PENDL2BM	Endlagenschalter 2 BMK	FRAG_DIG_Vetnil.ema

ANHANG 8: ELEMENTE IMPORTIEREN UND AKTUALISIEREN, MAPPINGS

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: MSR-Liste\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_MSR-Liste_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Mehrfach
 Elementtyp: Mehrfach

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Stelle	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Bezeichnung	Attribut	Kommentar		<input type="checkbox"/>			<input type="checkbox"/>
HW-Typical	Attribut	Typical		<input type="checkbox"/>			<input type="checkbox"/>
Schrank	Attribut	Typical Baueinheit 7		<input type="checkbox"/>			<input type="checkbox"/>
Feld	Attribut	Typical Baueinheit 1		<input type="checkbox"/>			<input type="checkbox"/>
Funktionstyp-EB	Typenkennzeichen			<input type="checkbox"/>			<input type="checkbox"/>

Datenvorschau

Anlage	Teilanlage	Stelle	Bezeichnung	HW-Typical	Schrank	Feld	Funktionstyp-EB
H1	GHE10	CP001	Druckmessung Vorlauf	MSR_PI_4-20mA_2L	+CKC01GH001	Feld	Messstelle
H1	GHE10	CP002	Druckmessung Vorlauf	MSR_PI_4-20mA_2L	+CKC01GH001	Feld	Messstelle
H1	INDC01	CP001	Druckmessung Vorlauf	MSR_PI_4-20mA_2L	+CKC01GH001	Feld	Messstelle
H1	GHE10	AA103	Spülventil nach Wärmetauscher	ARM_P_AufZu	+CKC01GH001	Feld	Stellstelle

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen

OK Abbrechen Übernehme

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: Baueinheiten-Imports\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Baueinheiten_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Baueinheitstypen
 Elementtyp: Baueinheit

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
zugeordnete Funktion	Funktion	Name			Anlage	0	
Schrankbezeichnung	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Beschreibung	Attribut	Kommentar		<input type="checkbox"/>			<input type="checkbox"/>

Datenvorschau

zugeordnete Funktion	Schrankbezeichnung	Beschreibung
Feld		Feldebene

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen

OK Abbrechen Übernehme

Anhang 8: Elemente importieren und aktualisieren, Mappings

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: 'SPS-Import\$'
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_SPS_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Gerätetypen
 Elementtyp: Automatisierungsgerät

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Teil Von	Teil von	Name		<input checked="" type="checkbox"/>	Baueinheit	0	<input type="checkbox"/>
BMK	Attribut	Name		<input type="checkbox"/>			<input type="checkbox"/>
Typ	--			<input type="checkbox"/>			<input type="checkbox"/>
Materialnummer	Attribut	Katalognummer		<input type="checkbox"/>			<input type="checkbox"/>
Adressbereich Eingänge	Attribut	Adressbereich Eingänge		<input type="checkbox"/>			<input type="checkbox"/>
Adressbereich Ausgänge	Attribut	Adressbereich Ausgänge		<input type="checkbox"/>			<input type="checkbox"/>
EB-Katalognummer	Attribut	Material		<input type="checkbox"/>			<input type="checkbox"/>

Datenvorschau

Teil Von	BMK	Typ	Materialnummer	Adressbereich Eingänge	Adressbereich Ausgänge	EB-Katalognummer
+CKC01GH001 -A10	Eingang	6ES7 331-7KF01-0AB0	IW200	-	SIE_331-001	
+CKC01GH001 -A11	Eingang	6ES7 321-1BH01-0AA0	IO.0	-	SIE_321-002	

OK Abbrechen Übernehme

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\MA-MDA-Makro.xlsx
 Tafel: 'Kabel-Import-Messstellen\$'
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Kabel-Messstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Kabeltypen
 Elementtyp: Mehrdriges Kabel

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name		<input type="checkbox"/>	Anlage	1	<input type="checkbox"/>
Teilanlage	Funktion	Name		<input type="checkbox"/>	Anlage	2	<input type="checkbox"/>
Messstelle	Funktion	Name		<input type="checkbox"/>	Messstelle	3	<input type="checkbox"/>
Kabel-BMK	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Kabeltyp-EB-Katalognummer	Attribut	Material		<input type="checkbox"/>			<input type="checkbox"/>
Länge(m)	Attribut	Länge	m	<input type="checkbox"/>			<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name		<input type="checkbox"/>	Baueinheit	1	<input type="checkbox"/>

Datenvorschau

Anlage	Teilanlage	Messstelle	Kabel-BMK	Kabeltyp-EB-Katalognummer	Länge(m)	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	CP001	-W01	YSLCY-OZ 2x1	10	WS01	Feld
H1	GHE10	CP002	-W01	YSLCY-OZ 2x1	10	WS01	Feld
H1	NDC01	CP001	-W01	YSLCY-OZ 2x1	10	WS01	Feld

OK Abbrechen Übernehme

Anhang 8: Elemente importieren und aktualisieren, Mappings

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: Kabel-Import-Stellstellen\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Kabel-Stellstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Kabeltypen
 Elementtyp: Mehrdriges Kabel

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Stellstelle	Funktion	Name			Stellstelle	3	
Kabel-BMK	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Kabeltyp-EB-Katalognumm	Attribut	Material		<input type="checkbox"/>			<input type="checkbox"/>
Länge(m)	Attribut	Länge	m	<input type="checkbox"/>			<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	0	

Datenvorschau

Anlage	Teilanlage	Stellstelle	Kabel-BMK	Kabeltyp-EB-Katalognummer	Länge(m)	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	AA103	-LS01	PA-1x6/8	25	WP01	Feld

OK Abbrechen Übernehme

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: Feldgerät-Import-Messstellen\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Feldgeräte-Messstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Mehrfach
 Elementtyp: Mehrfach

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Messstelle	Funktion	Name			Messstelle	3	
Feldgerät-BMK	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Gerätetyp-EB	Typenkennzeichen			<input type="checkbox"/>	Define		<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	0	

Datenvorschau

Anlage	Teilanlage	Messstelle	Feldgerät-BMK	Gerätetyp-EB	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	CP001	-PT01	Sensor, Druck	801	Feld
H1	GHE10	CP002	-PT01	Sensor, Druck	801	Feld
H1	NDC01	CP001	-PT01	Sensor, Druck	801	Feld

OK Abbrechen Übernehme

Anhang 8: Elemente importieren und aktualisieren, Mappings

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: Feldgerät-Import-Stellstellen\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Feldgeräte-Stellstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Mehrfach
 Elementtyp: Mehrfach

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Stellstelle	Funktion	Name			Stellstelle	3	
Feldgerät-BMK	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Gerätetyp-EB	Typenkennzeichen				Define		
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	0	

Datenvorschau

Anlage	Teilanlage	Stellstelle	Feldgerät-BMK	Gerätetyp-EB	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	AA103	-Y01	Antriebe (nicht elektrisch) (Prozess / Fluid)	MA01	Feld
H1	GHE10	AA103	-S01	Sensor, Abstand Länge, Stellung	B01	Feld

OK Abbrechen Übernehme

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\MA-MDA-Makro.xlsx
 Tafel: Kabel-Import-Messstellen\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Kabel-Messstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Kabeltypen
 Elementtyp: Mehrdriges Kabel

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Messstelle	Funktion	Name			Messstelle	3	
Kabel-BMK	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Kabeltyp-EB-Katalognummer	Attribut	Material		<input type="checkbox"/>			<input type="checkbox"/>
Länge(m)	Attribut	Länge	m	<input type="checkbox"/>			<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	0	

Datenvorschau

Anlage	Teilanlage	Messstelle	Kabel-BMK	Kabeltyp-EB-Katalognummer	Länge(m)	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	CP001	-W01	YSLCY-OZ 2x1	10	WS01	Feld
H1	GHE10	CP002	-W01	YSLCY-OZ 2x1	10	WS01	Feld
H1	NDC01	CP001	-W01	YSLCY-OZ 2x1	10	WS01	Feld

OK Abbrechen Übernehme

Anhang 8: Elemente importieren und aktualisieren, Mappings

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: Kabel-Import-Stellstellen\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Kabel-Stellstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Kabeltypen
 Elementtyp: Mehrdriges Kabel

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Stellstelle	Funktion	Name			Stellstelle	3	
Kabel-BMK	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Kabeltyp-EB-Katalognumm	Attribut	Material		<input type="checkbox"/>			<input type="checkbox"/>
Länge(m)	Attribut	Länge	m	<input type="checkbox"/>			<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	0	

Datenvorschau

Anlage	Teilanlage	Stellstelle	Kabel-BMK	Kabeltyp-EB-Katalognummer	Länge(m)	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	AA103	-LS01	PA-1x6/8	25	WP01	Feld

OK Abbrechen Übernehme

Elemente importieren und aktualisieren (3.4.21.545)

Wähle Datenquelle
 Quelle: C:\Users\grubdom\Desktop\XML-Makro.xlsx
 Tafel: Klemmleiste-Import-Messstellen\$
 Beginne mit Zeile: 2

Mapping
 Mapping auswählen: Mapping_Klemmleisten-Messstellen_Import_Masterarbeit
 Importiere Elemente unter:
 Elementart: Gerätetypen
 Elementtyp: Klemmleiste

Attribute auswählen
 Struktur vergleichen mit:
 Betriebsmittel
 Funktion
 Optionen:
 Elemente mit Material aus den Katalog aktualisieren
 Statusattribute setzen
 Auswählen

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Messstelle	Funktion	Name			Messstelle	3	
Bezeichnung	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	1	

Datenvorschau

Anlage	Teilanlage	Messstelle	Bezeichnung	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	CP001	-X05	X01	+CKC01GH001
H1	GHE10	CP002	-X05	X01	+CKC01GH001
H1	NDC01	CP001	-X05	X01	+CKC01GH001

OK Abbrechen Übernehme

Anhang 8: Elemente importieren und aktualisieren, Mappings

Wähle Datenquelle

Quelle: C:\Users\grubdom\Desktop\WML-Makro.xlsx

Tafel: 'Klemmleiste-Import-Stellstellen'

Beginne mit Zeile: 2

Mapping

Mapping auswählen: Mapping_Klemmleisten-Stellstellen_Import_Masterarbeit

Importiere Elemente unter

Elementart: Gerätetypen

Elementtyp: Klemmleiste

Speichern Löschen Import Export

Spalte	Operation	Attribut	Defaulteinheit	Primärschlüssel	Typ auswählen	Ebene	Schreibgeschützt
Anlage	Funktion	Name			Anlage	1	
Teilanlage	Funktion	Name			Anlage	2	
Stellstelle	Funktion	Name			Stellstelle	3	
Bezeichnung	Attribut	Name		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Typical Copy Merge Index	Attribut	Typical Copy Merge Index		<input checked="" type="checkbox"/>			<input type="checkbox"/>
Örtliche Zuordnung	Teil von	Name			Baueinheit	0	

Datenvorschau

Anlage	Teilanlage	Stellstelle	Bezeichnung	Typical Copy Merge Index	Örtliche Zuordnung
H1	GHE10	AA103	-X10	X01	+CKC01GH001

Attribute auswählen

Struktur vergleichen mit

Betriebsmittel

Funktion

Optionen

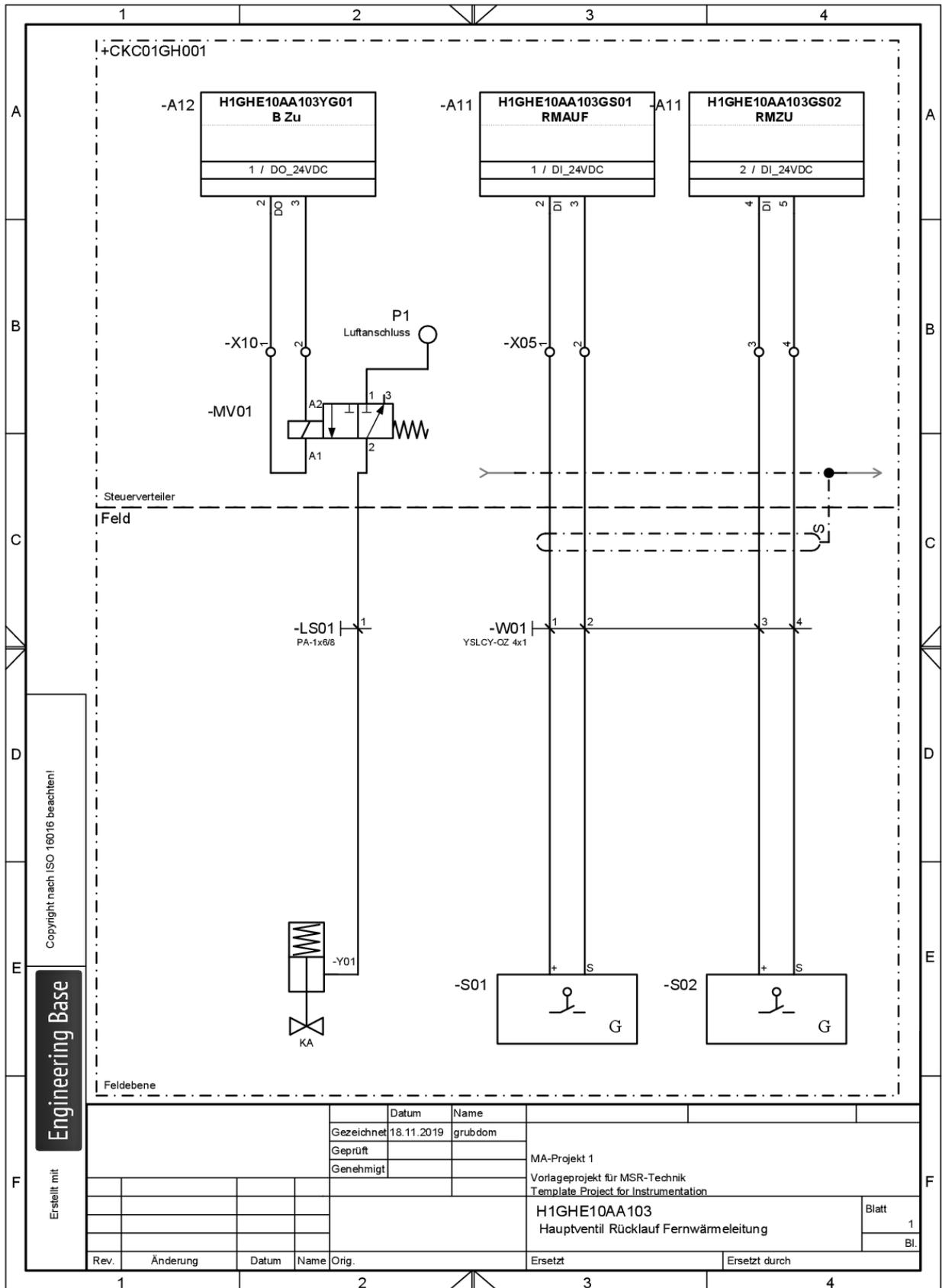
Elemente mit Material aus den Katalog aktualisieren

Statusattribute setzen

Auswählen

OK Abbrechen Übernehme

ANHANG 9.1: GENERIERTER EB-STELLENPLAN H1 GHE10 AA103



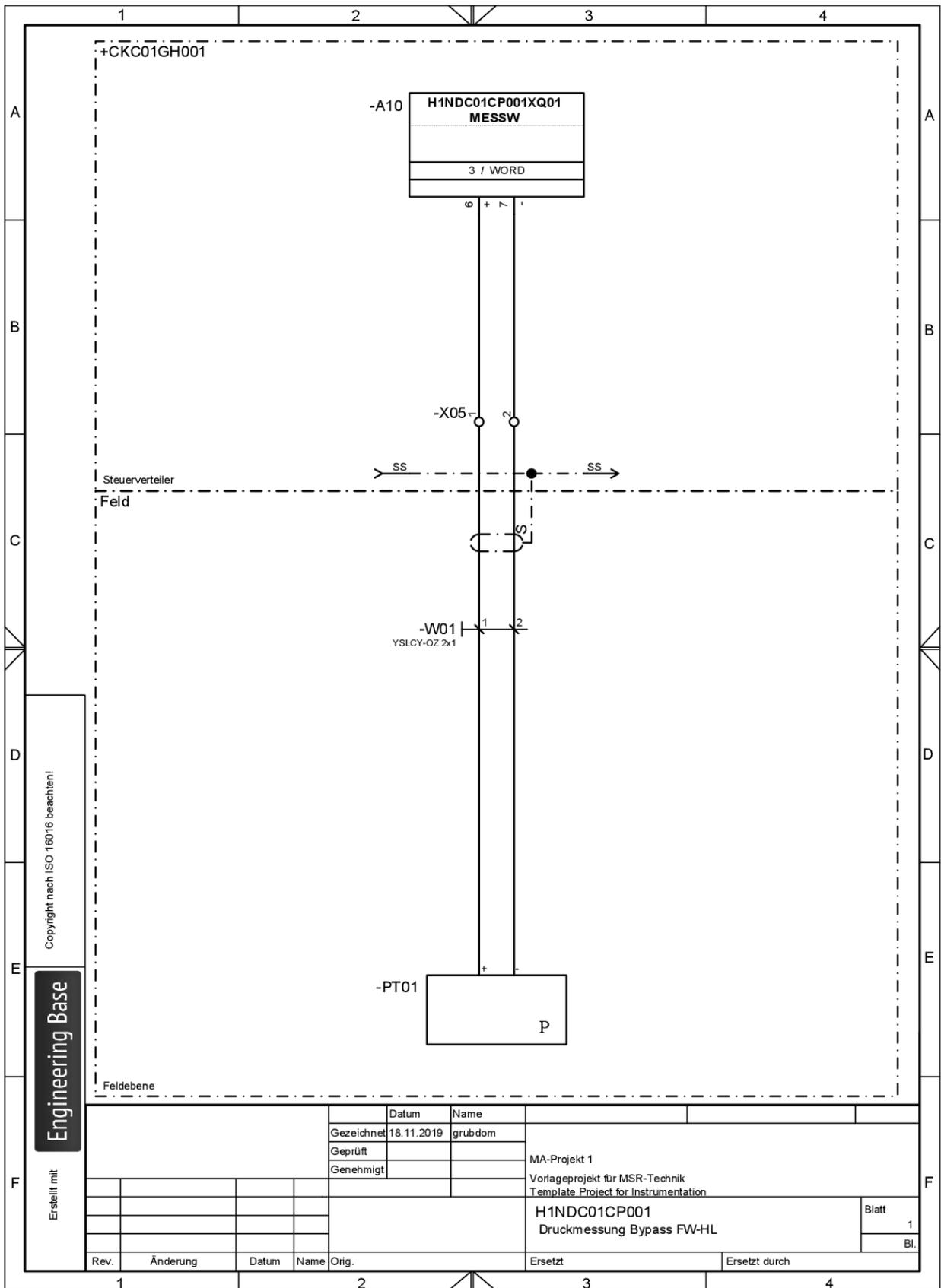
Copyright nach ISO 18016 beachten!

Engineering Base

Erstellt mit

		Datum	Name		
		Gezeichnet	18.11.2019	grubdom	
		Geprüft			
		Genehmigt			
					MA-Projekt 1
					Vorlageprojekt für MSR-Technik
					Template Project for Instrumentation
					H1GHE10AA103
					Hauptventil Rücklauf Fernwärmeleitung
					Blatt 1
					Bl.
Rev.	Änderung	Datum	Name	Orig.	Ersetzt durch
1					

ANHANG 9.2: GENERIERTER EB-STELLENPLAN H1 NDC01 CP001



Copyright nach ISO 18016 beachten!

Engineering Base

Erstellt mit

		Datum	Name		
Gezeichnet		18.11.2019	grubdom		
Geprüft				MA-Projekt 1	
Genehmigt				Vorlageprojekt für MSR-Technik Template Project for Instrumentation	
				H1NDC01CP001	
				Druckmessung Bypass FW-HL	
				Blatt	1
				Bl.	
Rev.	Änderung	Datum	Name	Orig.	Ersetzt durch

ANHANG 10: BEFÜLLTE ENGINEERING-LISTE FÜR DEN SCHALT- PLANGENERATOR G8

	C	E	F	G	K	L	P	Q	S	T	U	V	W	X	Y	Z	AA	AB
7	Seiten		Typical		Strukturkennzeichen		Parameter											
8																		
9	Seitenbeschreibung	Typical	Typicalname	Strukturkennzeichen	Parameter	Sensor BMK	Einbauort	Analogtochter Kanal	Analogtochter BMK	Kabel-Artikel	Kabel BMK	Klemmleiste AI	Schlauchbezeichnung	Schlauch-Artikel	DO-Tochter Kanal	DO-Tochter BMK	Klemmleiste DO	
10	Seitenbeschreibung	TYPICAL_ECAD		=	PSENSBMK Einbauort		PIOTOCHTERKANAL	PIOTOCHTERKANAL	Kabel-Artikel	Kabelname PKLAI	Schlauchname	Schlauch-Artikel	PIOTOCHTERDOKKANAL	PIOTOCHTERDOB	PKLDO			
11	Druckmessung vor Hauptventil FW-RL	Typicals MA Druckmessung	H1.GHEI0CP001		-PT01	+CKCOIGH001	CH00	-A10	MEI.VSICV-OZ 2x1	-W01	-X05							
12	Druckmessung nach Hauptventil FW-RL	Typicals MA Druckmessung	H1.GHEI0CP002		-PT01	+CKCOIGH001	CH01	-A10	MEI.VSICV-OZ 2x1	-W01	-X05							
13	Druckmessung Bypass FW-HL	Typicals MA Druckmessung	H1.ND001CP001		-PT01	+CKCOIGH001	CH02	-A10	MEI.VSICV-OZ 2x1	-W01	-X05							
14	Hauptventil Rücklauf Fernwärmeleitung	Typicals MA Ventil	H1.GHEI0AA103		-V01	+CKCOIGH001			MEI.VSICV-OZ 4x1	-W01	-X05	-LS01	PA-1x6/8	CH00		-A12	-X10	
15	Hauptventil Rücklauf Fernwärmeleitung	Typicals MA Ventil	H1.GHEI0AA103		-V01	+CKCOIGH001	CH03	-A10	MEI.VSICV-OZ 2x1	-W01	-X05							
16	Temperaturmessung Rücklauf FW-HL	Typicals MA Temperaturmessung	H1.ND001CT101		-T01	+CKCOIGH001			MEI.VSICV-OZ 2x1	-W01	-X05							
7	Seiten		Typical		Strukturkennzeichen													
8																		
9	Seitenbeschreibung	Typical	Typicalname	Strukturkennzeichen	Parameter	DI-Tochter 1 Kanal	DI-Tochter 1 BMK	PENDL2BMK	DI-Tochter 2 Kanal	DI-Tochter 2 BMK	Klemmleiste DI	Magnetventil BMK						
10	Seitenbeschreibung	TYPICAL_ECAD		=	PENDL1BMK	PIOTOCHTERDI1KANAL	PIOTOCHTERDI1BMK	PENDL2BMK	PIOTOCHTERDI2KANAL	PIOTOCHTERDI2BMK	PKLDI	PMVBMK						
11	Druckmessung vor Hauptventil FW-RL	Typicals MA Druckmessung	H1.GHEI0CP001															
12	Druckmessung nach Hauptventil FW-RL	Typicals MA Druckmessung	H1.GHEI0CP002															
13	Druckmessung Bypass FW-HL	Typicals MA Druckmessung	H1.ND001CP001															
14	Hauptventil Rücklauf Fernwärmeleitung	Typicals MA Ventil	H1.GHEI0AA103		-S01	CH00	-A11	-S02	CH01	-A11	-X05	-MV01						
15	Hauptventil Rücklauf Fernwärmeleitung	Typicals MA Ventil	H1.GHEI0AA103															
16	Temperaturmessung Rücklauf FW-HL	Typicals MA Temperaturmessung	H1.ND001CT101															

ANHANG 11.2: GENERIERTER EPLAN-STELLENPLAN H1 NDC01 CP001

