

TECHNISCHE UNIVERSITÄT DRESDEN  
BEREICH INGENIEURWISSENSCHAFTEN  
FAKULTÄT INFORMATIK  
INSTITUT FÜR ANGEWANDTE INFORMATIK  
PROFESSUR PROZESSKOMMUNIKATION

## DISSERTATIONSSCHRIFT

### **Infrastruktur für den Online-Zugriff auf prozesstechnische Apparate ohne dedizierte Kommunikationsanschaltung**

vorgelegt von

**Dipl.-Ing. Stefan Theurich**

geboren am 27. Juli 1983 in Erfurt

zum Erlangen des akademischen Grades

**Doktoringenieur**

(Dr.-Ing.)

eingereicht am

17. Dezember 2015

verteidigt am

21. Juni 2016

betreut und begutachtet durch  
Prof. Dr.-Ing. habil. Martin Wollschlaeger  
(Technische Universität Dresden)

begutachtet durch  
Prof. Dr.-Ing. Christian Diedrich  
(Otto-von-Guericke-Universität Magdeburg)



## **Danksagung**

Hiermit möchte ich Allen danken,  
die mich bei der Anfertigung dieser Arbeit unterstützt haben:

meinen Doktorvater Martin (Prof. Wollschlaeger) für die fachliche Betreuung, die vielen Hinweise auf Querbezüge, die ständige Bereitschaft zur Unterstützung, vor allem mit Kleinigkeiten, die das Arbeiten sehr vereinfachen konnten sowie die Übernahme der Verpflichtungen im Rahmen der Betreuung, nicht zuletzt die Begutachtung;

meiner Steffi für Motivation, Unterstützung und der Sicht der Verfahrenstechnikerin;

meinen Kollegen vom Lehrstuhl Prozesskommunikation: Alexander, Robert, Stefan, Roman, Christian, Antje und Andreas für Anregungen, Motivation und Tipps;

Albrecht und Stefan für die zielgerichtete Arbeit in Ihren Beleg- und Diplomarbeiten;

Leon (Prof. Urbas) für das Übernehmen der Rolle des Fachreferenten, sowie Ihm und Seinen Mitarbeitern vom Lehrstuhl Prozessleittechnik (IFA, TUD): Falk, Markus, Michael, Markus, Jens, Mathias, Annett und Johannes für die kritische zweite Sichtweise in den jährlichen Kolloquien;

Andreas und Prof. Ulrich Epple für die Kooperation bei CAEX-Themen und wertvollen Input;

Prof. Christian Diedrich für die Zweitbegutachtung und Ihm und Seiner Forschungsgruppe für wertvolle Anregungen;

und allen Anderen hier nicht genannten, die einen Beitrag daran hatten, dass ich diese Dissertationsschrift fertigstellen konnte.



Science is about what is;  
engineering is about what can be.

---

*(Neil Armstrong)*



## **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die von mir am heutigen Tage dem Promotionsausschuss der Fakultät Informatik eingereichte Dissertationsschrift zum Thema

*Infrastruktur für den Online-Zugriff auf prozesstechnische Apparate ohne dedizierte Kommunikationsanschaltung*

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Leipzig, den 17. Dezember 2015

Stefan Theurich





## **Zusammenfassung**

Der Betrieb von prozesstechnischen Produktionsanlagen wird stetig von verschiedenen Aufgaben begleitet, zum Beispiel der Steuerung und Optimierung der Produktion und der Aufrechterhaltung der Verfügbarkeit. Alle Gewerke, die sich mit dem Zustand der Anlage und des darin ablaufenden Prozesses beschäftigen, sind auf Daten angewiesen, die in der Anlage erfasst werden. Ein Großteil dieser Daten werden für die automatische Steuerungs-, Regelungs- und Sicherheitstechnik erfasst und darin in Echtzeit verarbeitet. Apparate und anderes Equipment sind zumeist nicht mit für deren Zustandsüberwachung dedizierter Messtechnik ausgestattet. Um Qualitätsmerkmale, Anlagenzustände oder Wartungsbedarfe erkennen zu können, müssen andere in der Anlage vorhandene Daten kombiniert und in Berechnungsmodellen kondensiert werden. Diese Methodik teilt sich in unterschiedliche Schritte auf: Datenakquise, Entwurf von Auswertemodellen, Modellintegration und Auswertung von Ergebnissen mit Ableitung von Aufgaben.

Die vorliegende Arbeit ordnet sich in die Softwareaspekte dieser Methodik ein. Dabei versucht sie, die zentrale Frage „Wie könnte eine Infrastruktur auf Basis von verbreiteten Standardtechnologien aussehen, welche alle Schritte des Engineeringprozesses für freie Apparatemodelle automatisieren kann?“ anhand eines Vorschlags für eine Infrastruktur zu beantworten.

Es wird eine Möglichkeit dargelegt, im Betrieb ohne Änderungen am bestehenden System kontinuierlich Daten für die Weiterverwendung in Apparatemodellen auszulesen. Der Entwurf und die Implementierung von Auswertemodellen wurde mit Hilfe eines entwickelten Werkzeugs unterstützt und dadurch die Struktur der Apparatemodelle vorgegeben, um eine einheitliche Modellintegration zu ermöglichen. Die Durchführung der Modellintegration erfolgte über die automatische Auswertung von Planungsdaten. Eine auf offenen Technologien basierende Ausführungsplattform für die Bewertungsmodelle wurde implementiert. Die Auswertung von Berechnungsergebnissen wurde über die Integration der Modelle in verbreitete, für Feldgeräte vorgesehene Standardwerkzeuge ermöglicht. Diese Infrastruktur ermöglicht es den verschiedenen Gewerken des Anlagenbetreibers, generische Bewertungsmodelle auf die Apparateinstanzen in der Anlage anzuwenden, und mit deren Berechnungsergebnissen ihre Aufgaben einfacher oder besser bearbeiten zu können.

Nach einer Analyse der technischen Rahmenbedingungen wurde ein Konzept zur Modellintegration entwickelt und dessen Automatisierbarkeit diskutiert. Dieses Konzept wurde prototypisch umgesetzt. Es wurden Softwarekomponenten für den Betrieb sowie Softwarewerkzeuge für die Unterstützung sowohl der Erstellung als auch der Integration von Apparatemodellen entwickelt. Anhand dieser wurde Umsetzbarkeit des Konzepts überprüft.



## **Abstract**

Operating process plants goes along with different tasks, e. g. control and optimization of the production and maintaining availability of the plant. There are several subsections of operations who deal with the state of the plant and the processes it runs. They are all dependent on information which is gathered throughout the plant. Most of this data is acquired for the automatic control, regulation, and safety gear and is processed in real-time. Apparatuses and other equipment are usually not equipped with measurement devices which are dedicated to monitor their state. For being able to recognize specific quality attributes, states of the plant, or maintenance needs, the existing measurements have to be combined and condensed by calculations. This methodology can be split into the following steps: data acquisition, design of evaluation models, integration of these models, and assessment of findings including inferring actions.

This thesis addresses software aspects of this methodology. It tries to answer the key question „How to build an infrastructure, which shall be based on common standard technologies, in which all steps to engineer equipment models may be automated?“ by proposing a concrete infrastructure.

A technique has been designed to continuously acquire data for further processing in equipment models without any changes to existing systems. The process of design and implementation of equipment models has been supported by a purpose-built tool. This tool puts out the designed models in a uniform structure to allow uniform model integration. This integration has been automated using the plant's engineering data. An execution platform has been developed based on open technologies. Infrastructure and model structure have been designed to easily integrate calculation results into standard tools for being able to use them in common work environments. It enables the different subsections of operations in a plant to apply generic equipment assessment models on concrete equipment instances. Using the output of the models, they shall be enabled to perform their task in an easier or better manner.

The technical requirements and prerequisites have been analyzed. Using the resulting conclusions, a concept to integrate models has been developed and the options to automate it have been discussed. This concept has been implemented prototypically. This implementation includes a runtime component and two tools to support development of models and their instantiation. It has been used to prove the feasibility of the concept.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Zieldomäne . . . . .	1
1.2	Spannungsfeld . . . . .	3
1.3	Kontext, Ziele und Struktur dieser Arbeit . . . . .	4
<b>2</b>	<b>Relevante Technologien und Stand der Technik</b>	<b>7</b>
2.1	Engineering und Engineeringdaten . . . . .	8
2.1.1	Prozess-Planungsdaten . . . . .	8
2.1.2	Anlagen-Planungsdaten . . . . .	9
2.1.2.1	Allgemeine Ansätze zum Austausch von Engineeringdaten . . . . .	9
2.1.2.2	Computer Aided Engineering eXchange . . . . .	11
2.1.2.3	STEP . . . . .	12
2.1.2.4	PlantXML . . . . .	12
2.1.2.5	Formalisierte Prozessbeschreibung . . . . .	13
2.1.2.6	Anwendung von Engineeringdaten . . . . .	13
2.1.3	Instrumentierungs-Planungsdaten . . . . .	13
2.1.4	Vernetzungs-Planungsdaten . . . . .	15
2.1.4.1	Kommunikationstechnologien . . . . .	15
2.1.4.2	Planung . . . . .	17
2.1.4.3	Engineering der Basiskommunikation . . . . .	17
2.1.5	Parametrierungs-Daten . . . . .	17
2.1.5.1	Allgemeine Modelle . . . . .	18
2.1.5.2	Electronic Device Description Language . . . . .	18
2.1.5.3	Field Device Tool . . . . .	19
2.1.5.4	Field Device Integration . . . . .	20
2.1.5.5	Field Device Tool 2 . . . . .	23
2.2	Betrieb und Betriebsdaten . . . . .	23
2.2.1	Betriebsdaten zu Anlage und Prozess . . . . .	24
2.2.1.1	Zugriffe aus höheren Ebenen und vertikale Integration . . . . .	24
2.2.1.2	OLE for Process Control . . . . .	25
2.2.1.3	OPC Unified Architecture . . . . .	26
2.2.2	Betriebsdaten zu Feldgeräten . . . . .	26
2.2.2.1	Wartung und Verwaltung / Plant Asset Management . . . . .	26
2.2.2.2	Geräte-Selbstüberwachung . . . . .	27
2.3	Bewertungsmodelle und deren Anwendung . . . . .	28
2.3.1	Simulationsmodelle . . . . .	29
2.3.2	Apparatemodelle im Prozessleitsystem . . . . .	30
2.3.3	Apparatemodelle außerhalb des Prozessleitsystems . . . . .	30

2.3.4	Weitere Ansätze . . . . .	31
2.3.4.1	Condition Monitoring . . . . .	31
2.3.4.2	Predictive Maintenance . . . . .	32
2.3.4.3	Digitale Fabrik . . . . .	32
<b>3</b>	<b>Konzept zur Modellintegration</b>	<b>35</b>
3.1	Abstraktion der Domäne . . . . .	35
3.1.1	Verfahrensbilanzraum . . . . .	35
3.1.2	Geräteanforderungen . . . . .	36
3.1.3	Konkrete Geräte . . . . .	36
3.1.4	Apparatemodell . . . . .	37
3.1.5	Gesamtmodell . . . . .	37
3.2	Beteiligte Akteure . . . . .	40
3.3	Werkzeugkette . . . . .	42
3.3.1	Apparatemodelle und FDI Pakete . . . . .	43
3.3.2	Instanziierung . . . . .	45
3.3.3	Betrieb . . . . .	46
3.4	Modell-Anwendungsfälle . . . . .	47
3.5	Verwertungsszenarien . . . . .	48
3.6	Migrationsstrategie . . . . .	48
3.6.1	FDI Projektplan und Verzögerungen . . . . .	49
3.6.2	Ohne voll funktionalen FDI Server . . . . .	49
3.6.3	Mit einem marktreifen FDI Server . . . . .	50
3.7	Diskussion . . . . .	51
3.7.1	Betrachtung der beteiligten Akteure . . . . .	51
3.7.2	Technische Voraussetzungen in der Anlage . . . . .	52
3.7.3	Technologische Voraussetzungen in der Anlage . . . . .	53
3.7.4	Geforderte Dynamik . . . . .	54
3.7.5	Skalierbarkeit . . . . .	55
3.7.6	Verhältnis von Aufwand und Nutzen . . . . .	55
<b>4</b>	<b>Entwurf und Implementierung</b>	<b>57</b>
4.1	Bereitstellung des Informationshaushalts . . . . .	57
4.1.1	Engineeringdaten Anlage . . . . .	57
4.1.1.1	CAEX . . . . .	57
4.1.1.2	PandIX . . . . .	58
4.1.1.3	CAEX-Checker . . . . .	58
4.1.1.4	Dia2CAEX . . . . .	61
4.1.2	FDI Server Emulation . . . . .	62
4.1.2.1	FDI OPC UA Informationsmodell . . . . .	63
4.1.2.2	OPC UA Client . . . . .	63
4.1.2.3	FDI OPC UA Server . . . . .	64
4.1.3	Feldgeräte-Informationsmodell . . . . .	64
4.1.3.1	Informationsmodell . . . . .	64
4.1.3.2	Integration als NodeSet-XML . . . . .	65

4.1.3.3	Anbindung der Daten im dynamischen Informationsmodell . . . . .	66
4.2	Datenmodell für Apparatepakete . . . . .	66
4.2.1	Messstellen . . . . .	70
4.2.2	Parameter . . . . .	71
4.2.3	Berechnungsmodell . . . . .	71
4.2.4	Steuerung des Timings der Leseaufrufe im FDI Apparatepaket . .	74
4.2.5	Menüstruktur für Apparatepakete . . . . .	75
4.3	Apparatepaket-Werkzeug . . . . .	75
4.3.1	Erstellen von Modellhierarchie und Elementen . . . . .	75
4.3.2	Notation der Modelle . . . . .	79
4.3.3	Automatische Eingrenzung der ausführbaren Modelle . . . . .	81
4.3.4	User Interface Description/Plugin . . . . .	82
4.3.5	SystemUnitClass Export . . . . .	83
4.3.6	ProList und PAM-Methodenblatt Export . . . . .	83
4.3.7	EDDL Export . . . . .	84
4.3.8	Package Export . . . . .	85
4.4	Equipment Instance Handler . . . . .	86
4.4.1	Datenmodell für die Konfiguration des Equipment Instance Handlers . . . . .	87
4.4.2	FDI Communication Server . . . . .	88
4.4.3	FDI-Discovery und Client für den wiederholbaren Zugriff . . . . .	91
4.4.4	Datenmodell der zu verlinkenden Geräte . . . . .	92
4.4.5	Möglichkeiten der Steuerung des Timings der Leseaufrufe im Equipment Instance Handler . . . . .	93
4.5	Instanziierungswerkzeug . . . . .	94
4.5.1	CAEX-Suche und Matching . . . . .	95
4.5.2	Node-Suche nach Prozesswert . . . . .	100
4.5.3	Übergabe der Konfiguration an den EIH . . . . .	100
4.5.4	Instanz-Konfiguration im FDI Server . . . . .	102
4.6	Beispiele und Evaluation . . . . .	102
4.6.1	Generierung von Modellen bzw. Paketen . . . . .	103
4.6.2	Instanziierung . . . . .	106
4.6.2.1	Instanzen in der $CO_2$ -Absorptionsanlage . . . . .	106
4.6.2.2	Instanzen im ACPLT Pumpwerk . . . . .	112
4.6.2.3	Auswertung . . . . .	113
4.6.3	Anbindung der Kennzahlen . . . . .	115
4.6.4	Dynamik und Skalierbarkeit . . . . .	115
<b>5</b>	<b>Zusammenfassung der Ergebnisse und Ausblick</b>	<b>123</b>
5.1	Ergebnisse . . . . .	123
5.2	Bewertung der Erfüllung gesteckter Ziele . . . . .	124
5.3	Ansätze für Erweiterung und Fortführung . . . . .	127
5.4	Fazit . . . . .	130
<b>Akronyme</b>		<b>131</b>

<b>Glossar</b>	<b>135</b>
<b>Literaturverzeichnis</b>	<b>145</b>
<b>A Veröffentlichte Vorarbeiten</b>	<b>169</b>
A.1 Prüfbare Konstrukte in CAEX-Dateien . . . . .	169
A.2 Benutzungsoberfläche des Package-Werkzeugs . . . . .	172
<b>B Quelltexte</b>	<b>177</b>
B.1 Dia2CAEX . . . . .	177
B.2 SystemUnitClass für einen Apparat . . . . .	181
B.3 Schema für die Konfigurationsdatei des Equipment Instance Handlers . . .	182
<b>C Weitere Abbildungen</b>	<b>185</b>
C.1 Beispielanlage . . . . .	185



# Abbildungsverzeichnis

2.1	Lebenszyklus-Aktivitätsmodell (aus [13]) . . . . .	7
2.2	Interaktionspunkte verschiedener Gewerke im Engineering (aus [23]) . . . . .	8
2.3	Exemplarische Darstellung von Informationsflüssen zwischen CAE-Tools (aus [17]) . . . . .	10
2.4	Struktur eines modernen Prozessleitsystems (aus [88]) . . . . .	16
2.5	Funktionsblöcke und Parameter in Feldgeräten (aus [119]) . . . . .	19
2.6	Field Device Tool-Softwarearchitektur (aus [133]) . . . . .	20
2.7	FDI-Architektur (aus [149]) . . . . .	22
2.8	FDI-Package in einem FDT2-Frame (aus [174]) . . . . .	23
2.9	Generisches Aktivitätsmodell zum Produktionsmanagement (aus der ISA S95 [178]) . . . . .	25
2.10	Verschiedene PAM-Systemanbindungsvarianten (aus [194]) . . . . .	27
2.11	Methoden des Performance Monitoring (aus [209]) . . . . .	28
3.1	Bilanzraummodell . . . . .	36
3.2	Geräteanforderungsmodell . . . . .	36
3.3	Gerätemodell . . . . .	36
3.4	Prozessschritt- oder Apparatemodell . . . . .	37
3.5	Gesamtmodell, Verknüpfungsschritt 1 . . . . .	38
3.6	Gesamtmodell, Verknüpfungsschritt 2 . . . . .	39
3.7	Gesamtmodell, Verknüpfungsschritt 3 . . . . .	40
3.8	Werkzeugkette zur Integration von Apparatemodellen (aus [2]) mit Einordnung in den Anlagenlebenszyklus . . . . .	42
3.9	Generische Darstellung von Apparatemodellen . . . . .	44
3.10	Lesevorgänge bei Aktualisierung von Kennzahlen für das Apparatemodell . . . . .	46
3.11	Architektur und Informationsmodell ohne OPC UA-fähigen FDI Server . . . . .	50
3.12	Architektur und Informationsmodell mit OPC UA-fähigen FDI Server . . . . .	51
4.1	Architektur des CAEX-Checkers . . . . .	59
4.2	Qualitative Speichernutzung der verschiedenen Implementierungen der Referenz- und Integritäts-Checker . . . . .	60
4.3	Einfaches Beispiel einer Cache-Struktur . . . . .	61
4.4	Screenshot von <i>Dia</i> mit einem nach CAEX umwandelbaren Diagramm (aus [4]) . . . . .	62
4.5	Überblick über die in <i>Dia</i> modellierte Gaswaschanlage (aus [4] auf Basis von [274]) . . . . .	63
4.6	Struktur des Geräte-Informationsmodells . . . . .	65
4.7	Klassendiagramm mit den Klassen der <i>Extension</i> zum Einbinden von Messdaten . . . . .	67

4.8	Struktur eines Apparatemodells . . . . .	67
4.9	Datenmodell des Apparatemodells im Werkzeug (aus [2]) . . . . .	76
4.10	<i>UML</i> -Modell der Anbindung des Datenmodells des Apparatemodells in eine beliebige Bearbeitungsumgebung . . . . .	77
4.11	Einbindung des Datenmodells des Apparatemodells in die GUI des AppPaket-Werkzeugs . . . . .	78
4.12	Übersicht über die GUI des Apparatepaket-Werkzeugs (aus [2]) . . . . .	78
4.13	Mögliche Darstellung der Berechenbarkeit im Apparatepaket-Werkzeug (Gut-Fall, auf Basis von [2, Abb. 5]) . . . . .	82
4.14	Mögliche Darstellung der Berechenbarkeit im Apparatepaket-Werkzeug (Schlecht-Fall, auf Basis von [2, Abb. 5]) . . . . .	82
4.15	Werkzeugkette zum Packen von FDI Packages (aus [259]) . . . . .	86
4.16	Komponenten des Equipment Instance Handlers . . . . .	87
4.17	Einbindung eines Netzwerkes über einen FDI Communication Server (aus [159]) . . . . .	89
4.18	Adressraummodell des FDI Communication Servers des Equipment Instance Handlers . . . . .	90
4.19	Struktur des Apparat-Informationsmodells . . . . .	91
4.20	Gerätemodell gemäß IEC 61804-2 (aus [119]) . . . . .	93
4.21	Schritte bei der Verteilung der Instanzdaten für ein Apparatemodell . . . . .	95
4.22	Einbindung des Datenmodells des Apparatemodells in die GUI des Instanziierungswerkzeugs . . . . .	96
4.23	Teile einer aus einem CAE-Tool exportierten Instanzhierarchie (korrigierte Darstellung aus [2]) . . . . .	97
4.24	Verteilung der Konfigurationsartefakte des Equipment Instance Handlers . . . . .	101
4.25	Darstellung des Apparatemodells für einen Wärmeübertrager (aus [2]) . . . . .	103
4.26	Darstellung des Apparatemodells für eine Sorptionskolonne (1) . . . . .	104
4.27	Darstellung des Apparatemodells für eine Sorptionskolonne (2) . . . . .	104
4.28	Darstellung des Apparatemodells für eine Pumpe . . . . .	105
4.29	Darstellung des Apparatemodells für eine Rohrheizung . . . . .	105
4.30	Instanziierung <i>W 1</i> . . . . .	107
4.31	Instanziierung <i>W 5</i> . . . . .	108
4.32	Instanziierung <i>K 3</i> . . . . .	109
4.33	Instanziierung <i>P 1</i> . . . . .	110
4.34	Instanziierung Heizung <i>W 6</i> . . . . .	111
4.35	Instanziierung Heizung <i>W 4</i> . . . . .	112
4.36	Instanziierung Heizung <i>W 8</i> . . . . .	112
4.37	Instanziierung Heizung <i>W 9</i> . . . . .	113
4.38	R&I-Fließbild des ACPLT Pumpwerks (aus [298]) . . . . .	114
4.39	Kommunikationspfade in den verschiedenen Ausprägungen der Infrastruktur . . . . .	117
A.1	[2] Abb. 4: Gesamtansicht Package-Designer-Werkzeug . . . . .	173
A.2	[2] Abb. 5: Darstellung des Apparates und der Modellanforderungen . . . . .	173
A.3	[2] Abb. 6: Einstellungen für den Apparat . . . . .	174
A.4	[2] Abb. 7: Einstellungen für einen Prozessanschluss am Apparat . . . . .	174

A.5	[2] Abb. 8: Einstellungen für eine benötigte Messstelle . . . . .	174
A.6	[2] Abb. 9: Einstellungen für einen benötigten Parameter . . . . .	175
A.7	[2] Abb. 10: Einstellungen für eine zu berechnende Funktion . . . . .	175
C.1	In <i>Dia</i> modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 1 . . .	186
C.2	In <i>Dia</i> modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 2 . . .	187
C.3	In <i>Dia</i> modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 3 . . .	188
C.4	In <i>Dia</i> modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 4 . . .	189
C.5	Vereinfachtes RI-Fließschema der Beispielanlage (aus [274, Bild 4.2, S. 54])	190



# Tabellenverzeichnis

3.1	Technologien und ihre Verfügbarkeit laut [154] und [156] . . . . .	49
4.1	EDD-Parameter im Transducer Block für eine Messstellenanforderung . . .	72
4.2	EDD-Parameter für einen Modell-Parameter . . . . .	73
4.3	EDD-Parameter für eine Funktion eines Berechnungsmodells . . . . .	73
4.4	Instanziierung $W 1$ . . . . .	107
4.5	Instanziierung $W 5$ . . . . .	108
4.6	Instanziierung $K 3$ . . . . .	109
4.7	Instanziierung $P 1$ . . . . .	110
4.8	Instanziierung Heizung $W 6$ . . . . .	111
4.9	Instanziierung Heizung $W 4$ . . . . .	111
4.10	Instanziierung Heizung $W 8$ . . . . .	112
4.11	Instanziierung Heizung $W 9$ . . . . .	113
4.12	Anzahlen von Messgeräteanforderungen in den Apparatemodell-Beispielen	116
4.13	Anzahlen von Mess- und Stellgeräten und Apparaten in den Beispielanlagen sowie deren Verhältnisse . . . . .	119



# Quelltextverzeichnis

4.1	Variable-Element eines NodeSets mit einer Extension zum Zuweisen von Messdaten . . . . .	66
4.2	Gekürztes Beispiel für eine exportierte SystemUnitClass . . . . .	68
4.3	EDD-Definition eines ausblendbaren EDD-Parameters . . . . .	69
4.4	EDD-Code zur Verwendung eines ausblendbaren EDD-Parameters . . . . .	69
4.5	Ausblendbare Berechnung eines Funktionsergebnisses in einer EDD-Methode . . . . .	74
4.6	EDD-Mechanismus zum gleichzeitigen zyklischen Lesen von EDD-Parametern aus Geräten in einer Methode . . . . .	74
4.7	Drei physikalische Größen für die PCE-Kategorie F . . . . .	79
4.8	Beispiele für EDD-Builtins gemäß IEC 61804-5 mit und ohne JScience-Implementierung . . . . .	79
4.9	Bereitstellung eines Wertes für den <i>exp4j</i> -ExpressionBuilder . . . . .	80
4.10	Auswahl aus PA-UnitCodes für <i>JScience</i> . . . . .	81
4.11	Beispiel für ein Untermenü eines Transducers in der EDD . . . . .	83
4.12	Label einer Variablen mit Verweis ins Dictionary . . . . .	84
4.13	Automatisch erzeugte Dictionaryeinträge . . . . .	84
4.14	Einbindung des FDI Communication Servers in den FDI Server . . . . .	88
4.15	Einbindung des ConnectionPoint in der EDD eines Feldgerätes oder Apparatepaketes . . . . .	90
4.16	GSD-Fragment zum Verhindern der Adresszuweisung über den Bus . . . . .	91
4.17	Code zum Suchen nach definierten Rollenklassen mittels <i>XPath</i> . . . . .	99
4.18	Code zum Suchen nach InternalLinks, die eine bestimmte Schnittstelle eines bestimmten InternalElements verbinden, mittels <i>XPath</i> . . . . .	99
B.1	Dia2CAEX XSLT-Template für einen Wärmeübertrager . . . . .	177
B.2	Dia2CAEX Eingabe für einen Wärmeübertrager . . . . .	178
B.3	Dia2CAEX Ausgabe für einen Wärmeübertrager . . . . .	179
B.4	Dia2CAEX Eingabe für eine Messstelle . . . . .	179
B.5	Dia2CAEX Ausgabe für eine Messstelle . . . . .	180
B.6	Vollständiges Beispiel für eine exportierte SystemUnitClass . . . . .	181
B.7	Konfigurationsschema für den Equipment Instance Handler . . . . .	182
B.8	Schema für die NodeSet-Extension des Informationsmodells des FDI Communication Servers im Equipment Instance Handler . . . . .	184





# 1 Einleitung

Eine prozesstechnische Anlage und ihr Prozess sind über die wenigen Schnittstellen der Mess-, Stell- und Regelungstechnik (MSR) mit dem Prozessleitsystem und der digitalen Anlage verbunden. Aufgrund der Größe und der Komplexität der Anlagen sowie der Anschaffungs-, Verwaltungs- und Wartungskosten der MSR-Technik werden diese Schnittstellen auch bewusst schlank gestaltet. Zustandsgrößen und Eigenschaften von Anlage und Prozess werden nicht nur direkt über Messverfahren bestimmt, sondern auch über physikalische und logische Zusammenhänge hergeleitet. Deswegen können Verfahrensoptimierung, Plant Asset Management, Condition Monitoring und weitere wertsteigernde Aufgaben nicht direkt auf alle Zustandsgrößen und Eigenschaften zugreifen, die für eine umfassende Bewertung notwendig wären. Aufgrund dieser schwachen Bindung von physischen Bestandteilen einer Anlage und deren digitaler Repräsentation leitete sich die folgende Forschungsfrage ab:

„Wie könnte eine Infrastruktur auf Basis von verbreiteten Standardtechnologien aussehen, welche alle Schritte des Engineeringprozesses für freie Apparatemodelle automatisieren kann?“

## 1.1 Zieldomäne

Die großindustrielle Automatisierungstechnik teilt sich in zwei Domänen, die Prozessautomatisierung und die Fertigungsautomatisierung. Viele Aspekte sind in beiden Domänen gleich oder ähnlich ausgeprägt. Dies liegt vor Allem an der Harmonisierung seitens der Hersteller, die beide Domänen bedienen, und an der Standardisierung von Technologien.

Dennoch sind für die Umsetzung von Projekten unterschiedliche Anforderungen an die Lösungen maßgeblich. Die Prozessindustrie automatisiert überwiegend vergleichsweise träge Prozesse, die mit einer überschaubaren Menge von Basisstrukturen umgesetzt werden. Die Anlagen sind hierarchisch strukturiert und die Zusammenhänge in ihren Basisstrukturen sind gut modellierbar. Der Fokus liegt vor allem auf Sicherheit (Safety) und Verfügbarkeit. Kommunikationsstrukturen sind auf Robustheit und Verfügbarkeit ausgelegt, die Zykluszeiten sind vergleichbar langsam. Der Ausfall von Teilen einer Anlage hat meist schwerwiegende Auswirkungen auf die Gesamtanlage bzw. die Produktion. Bei An- und Abfahrzeiten von Stunden bis zu Wochen ist eine durchgehende Produktion mit ausschließlich geplanten Unterbrechungen in Abständen von Monaten bis Jahren für die Wirtschaftlichkeit essentiell. Die Anlagen besitzen eine hohe Komplexität und eine große Anzahl von zusammenhängende Teilanlagen und damit eine erhebliche Anzahl von Sensoren und Aktuatoren im Prozessleitsystem (PLS). Die Modularisierung der Automatisierungstechnik ist ein junges Thema in dieser Domäne.

Die Fertigungsautomatisierung hat ein deutlich größeres Spektrum von Anwendungsfällen und Automatisierungskomponenten. Abgesehen von großen, durchgehenden Fertigungsstraßen, wie in der Automobilindustrie, sind die Anlagen weniger komplex und ihre Teile

sind schwächer gekoppelt oder voneinander unabhängig. Die Anlagen, die teilweise auch aus Modulen zusammengesetzt und rekonfiguriert werden können, zeichnen sich durch hohe Taktzeiten und Echtzeitanforderungen mit niedrigen Zykluszeiten aus. Die Vielfalt und Geschwindigkeit der Prozesse macht es schwierig, das in dieser Arbeit beschriebene Konzept in der Fertigungsautomatisierung anzuwenden. Demzufolge bezieht sich die folgende Domänenbeschreibung auf die Prozessautomatisierung. Einige Kernbegriffe sollen im Folgenden definiert werden.

**Prozess** Prozesse in der Prozessautomatisierung sind Verfahren zur Trennung, Umwandlung oder Synthese von Stoffen sowie zur Energiegewinnung oder -speicherung über Stoffumwandlungen. Sie lassen sich über verschiedene Aspekte klassifizieren. In der informationstechnischen Behandlung ist vor allem die Unterscheidung zwischen Chargenprozessen und kontinuierlichen Prozessen relevant (Abschnitt 2.1.1 ab Seite 8).

**Anlage** Prozesstechnische Anlagen werden zum Fahren eines bestimmten Prozesses konzeptioniert, geplant und erbaut. Sie werden mit einer *top-down* Methodik entworfen und sind auch deswegen hierarchisch strukturiert. Neben Teilanlagen, die zum Fahren des Zielprozesses benötigt werden, sind ebenfalls Teilanlagen zur Versorgung mit Energie, Kühlung, Energierückgewinnung oder Bearbeitung von Nebenprodukten vorhanden (Abschnitt 2.1.2 ab Seite 9).

**Apparat** Apparate sind die Teile einer Anlage, in denen Prozessschritte oder betriebswichtige Funktionen ausgeführt werden. Sie werden mit Vorrichtungen zum Stofftransport verbunden (meist Rohrleitungen). Apparate besitzen zumeist keine dedizierte Messtechnik sondern werden im Engineering mit Sensoren und Aktuatoren für die Prozessführung ausgestattet. Meist werden dabei im Anlagenzusammenhang redundante Messgeräte eingespart. Die Auslegung und Eigenschaften eines Apparates hängen sehr stark vom darin ablaufenden Prozess, den beteiligten Stoffen und deren physikalischen Eigenschaften ab. Es gibt schädliche Einflüsse sowohl auf die Apparate als auch auf den Prozess, die betrachtet werden und deren Einflüsse dokumentiert und ausgewertet werden können (Abschnitt 2.1.2 ab Seite 9).

**Feldgerät** Die Feldgeräte sind die physikalische Schnittstelle von Prozess zur Prozessleittechnik. Sensoren quantifizieren vorwiegend physikalische Eigenschaften der in der Anlage befindlichen Stoffe und stellen sie der Prozessführung bereit. Aktuatoren erhalten Stellsignale aus der Prozessführung und beeinflussen den Prozess, um das vorgegebene Maß (Abschnitt 2.1.3 ab Seite 13). Beide sind – teilweise einfache, teilweise sehr komplexe – Systeme die, über Kommunikationsstrukturen an die Prozessführung angebunden sind (Abschnitt 2.1.4 ab Seite 15).

**Prozessführung** Der Prozess in der Anlage wird mittels eines Prozessleitsystems gesteuert. Anlagenfahrer beobachten den Prozess und nehmen gemäß Sicherheitskonzept und Produktionsplanung auf ihn Einfluss. Daneben werden Qualitätskontrolle, Optimierung, Zustandsüberwachung, Wartung und Instandhaltung kontinuierlich durchgeführt (Abschnitt 2.2 ab Seite 23).

**Mehrwertdienste** Mehrwertdienste bieten eine Erweiterung des Funktionsumfangs eines Systems mithilfe der für die Basisfunktion vorgesehenen Ressourcen. In der Prozessindustrie ist die Basisfunktion die Prozessführung und die Sicherheitsfunktionen. Plant Asset Management- oder Condition Monitoring (CM)-Systeme bieten Zusatzfunktionen, die über bereits vorhandene Kommunikationsstrukturen Daten für Mehrwertdienste auslesen (Abschnitt 2.3 ab Seite 28).

## 1.2 Spannungsfeld

Die Zieldomäne besitzt Eigenheiten und stellt nicht-funktionale Anforderungen, die die technologische Weiterentwicklung bestimmen. Diese müssen bei Technologieauswahl und Systementwurf berücksichtigt werden.

In der Prozessindustrie gehen von vielen Edukten und Produkten sowie ihrer Verarbeitung selbst eine Gefahr für Personal, Anlage und die Umwelt aus. Deswegen benötigen die Anlagen Sicherheitszertifizierungen für ihre Betriebserlaubnis. Auch aus diesem Grund ist die Branche gegenüber neuen Technologien konservativ eingestellt. Bekannte, praxiserprobte und betriebsbewährte Technologien werden neueren, kostengünstigeren und vielseitigeren Lösungen vorgezogen, um eine einfachere Risikoeinschätzung vornehmen zu können.

Große Hersteller von Automatisierungslösungen verfolgen das Ziel einer Vollabdeckung ihrer Kunden, die viel beworbene „Lösung aus einer Hand“. Es wird versucht, über geschlossene Systeme, proprietäre Lösungen und Herstellerbindung von Dienstleistern eine Kundenbindung zu erreichen. Die Spanne der Anwendungsfälle ist jedoch nicht klein genug, um die Anlagen vollständig auszurüsten zu können. Um die benötigten Nischenprodukte effizient integrieren zu können, ist eine Standardisierung der Schnittstellen und Integrationstechnologien erforderlich. Diese Standards erscheinen bewusst komplex ausgeführt, um den Zugriff auf die Nischenmärkte steuern zu können. Die entsprechenden Standardisierungsgremien werden ebenfalls von den großen Herstellern dominiert. Es ist nicht unüblich, vorentwickelte Lösungen nachträglich zu standardisieren, was aus Gründen des Investitionsschutzes entweder zu wenig strikten oder zu konkurrierenden Standards führt.

Viele kleine Hersteller füllen das Angebot an Automatisierungskomponenten auf und schließen die Lücken der großen. Sie müssen die Kompatibilität ihrer Produkte jedoch anhand der branchenüblichen Produkte und Integrationslösungen der großen Hersteller sicherstellen, nicht nur anhand der Standards, auf denen sie basieren.

Engineering-Dienstleister und Systemintegratoren setzen die konkreten Automatisierungsprojekte der Anwender um. Zumeist sind auch diese über Zertifizierungen zur Produkt- und Portfolio-Kenntnis an große Hersteller von Automatisierungskomponenten und Integrationslösungen gebunden.

Anwender der Automatisierungstechnik, also Anlagenbetreiber, sind interessiert an funktionierenden und gut dokumentierten Anlagen. Sie sollen anpassbar und erweiterbar sein. Bei der Ausrüstung ist für Innovation und aus Kostengründen ein starker Wettbewerb gewünscht. Deswegen sind Anwender interessiert an Einheitlichkeit und somit auch an Standardisierung. Aus diesem Grund gibt es Interessensgemeinschaften von Anwendern, die gemeinsame Anforderungen an die Hersteller formulieren.

Offene Integrationstechnologien verbreiten sich nur über Anwenderakzeptanz und somit über nutzbringende Anwendungsfälle und Verbreitung in Produkten. Das Wettfeiern unterschiedlicher Lösungen mit gleichem Ziel sorgt für Unsicherheit und schließlich zu ihrem Verschwinden. Nur wenige offene Technologien haben eine kritische Masse erreicht, Beispiele sind OPC (OLE for Process Control) und OPC UA (OPC Unified Architecture) (Abschnitt 2.2.1 ab Seite 24) in der gesamten Automatisierungstechnik und AutomationML (Abschnitt 2.1.2 ab Seite 9) vornehmlich in der Fertigungsautomatisierung.

### 1.3 Kontext, Ziele und Struktur dieser Arbeit

Die vorliegende Ergebnisse wurde im Rahmen meiner wissenschaftlichen Arbeit am Lehrstuhl Prozesskommunikation im Institut für Angewandte Informatik der Fakultät Informatik der Technischen Universität Dresden erstellt, sowie nach deren Ende fertiggestellt und dokumentiert. Sie stützt sich auf veröffentlichte Zwischenergebnisse [1, 2, 3, 4, 5] und eigene Vorarbeiten [6, 7, 8, 9] sowie Ergebnisse betreuter studentischer Arbeiten [10, 11, 12].

#### Ziele

Als eine mögliche Lösung für die aufgeworfene Forschungsfrage wird in dieser Arbeit eine Infrastruktur skizziert, mit der in weitestgehend geschlossenen Systemen über eine geeignete Auswahl von Daten und Schnittstellen nicht von vornherein vorgesehene, aber wertsteigernde Funktionen implementiert und aufwandsarm integriert werden können. Dafür soll es möglich gemacht werden, Modelle zu entwickeln, die den Zustand des Apparates oder des darin ablaufenden Prozesses bewerten können. Die notwendigen Vorgänge sollen nicht in das Produktivsystem oder geschlossene Softwarekomponenten eingreifen und keinen Einfluss auf den sicheren und zuverlässigen Betrieb haben. Dabei sollen standardisierte Datenformate und standardisierte Schnittstellen verwendet werden, um die für die Modelle notwendigen Daten identifizieren und auslesen zu können. Anhand von Beispiel-Modellen und -Anlagendaten sollen Funktion und Skalierbarkeit der Infrastruktur überprüft werden. Eine Integration bzw. Integrierbarkeit der Berechnungsergebnisse in gewerkespezifische Werkzeuge soll die Verwendbarkeit der Ergebnisse nachweisen. Die Inhalte der Berechnungsmodelle sollen nicht Teil dieser Arbeit sein. Im Folgenden werden verschiedene Anforderungen beschrieben, die für die verschiedenen beteiligten Akteure beachtet werden müssen.

**Anlagenbetreiber:** Die Ausführungsplattform für Bewertungsmodelle soll beim Betreiber des Apparates in bestehende (zum Betrieb zwingend notwendige) Infrastruktur integriert werden. Die Modelle sollen mit möglichst wenig Aufwand integriert und in Betrieb genommen werden können. Die durch die Modelle errechneten Kennzahlen sollen bewertbar und weiterverwendbar sein. Der Nutzen der Modelle kann nicht bewertet werden, da deren Inhalt nicht Gegenstand dieser Arbeit ist.

**Apparatemodellhersteller:** Die Struktur des Apparates ist einfach zu notieren. Messwert-Anforderungen an das Modell können einfach verwaltet werden. Berechnungsmodelle können produktunabhängig notiert werden. Die technische Umsetzung ist transparent

für den Hersteller des Modells. Die Modelle sollen mehrfach verwendet werden können.

## **Struktur**

Nach einer kurzen Einführung (Kapitel 1 ab Seite 1) wird in Kapitel 2 ab Seite 7 auf Systeme sowie deren Planung und Betrieb in prozesstechnischen Anlagen mit Fokus auf Kommunikationstechnologien eingegangen. In diesem Themenfeld sind zahlreiche Standards zu beachten, deren Inhalt mit Sicht auf das umzusetzende Konzept betrachtet wurde. Auch sind wissenschaftliche Arbeiten, die ähnliche Ziele verfolgen oder das Konzept unterstützende Ergebnisse vorstellen, darin aufgearbeitet.

In Kapitel 3 ab Seite 35 wird auf Basis der vorangegangenen Analyse die gestellte Forschungsfrage konzeptionell beantwortet. Dabei wird zuerst über die gegebenen Domänenmodelle abstrahiert und eine generische Lösung skizziert. Zunächst wird die getroffene Auswahl der Technologien begründet. Auf dieser Auswahl aufbauend werden verschiedene Anforderungen an die einzelnen Arbeitsschritte identifiziert und die Notwendigkeit von unterstützenden Werkzeugen aufgezeigt. Technische und wirtschaftliche Anwendungsmöglichkeiten für das Konzept werden kurz aufgeführt.

Aufgrund von Verzögerungen bei der Einführung einer für die Verwendung ausgewählten Technologie sind Hilfsmittel notwendig geworden, die ebenfalls beschrieben wurden. Die Migration vom Hilfsmittel zur anvisierten Lösung wird diskutiert. Ebenfalls werden die technischen Abhängigkeiten der konzeptionierten Lösung bewertet.

Kapitel 4 ab Seite 57 beschreibt die Umsetzung von verschiedenen technischen Voraussetzungen und dem Konzept zur Modellintegration. Dabei wird zunächst auf die Lösungen zur Bereitstellung benötigter Daten eingegangen. Dann wird die Struktur der im Betrieb benötigten Daten festgelegt, sowie die Implementierung eines Werkzeugs für deren Erstellung behandelt. Anschließend wird die Softwarekomponente zur Umsetzung der Kommunikation im Betrieb und das Werkzeugs für die Einrichtung der Instanzen beschrieben. Zuletzt wird anhand von Anwendungsbeispielen die vorgestellte Lösung bewertet.

In Kapitel 5 ab Seite 123 werden die Erkenntnisse aus Konzept und Umsetzung zusammengefasst. Es wird auf Aspekte eingegangen, die bei einer umfassenden Umsetzung beachtet werden müssen.

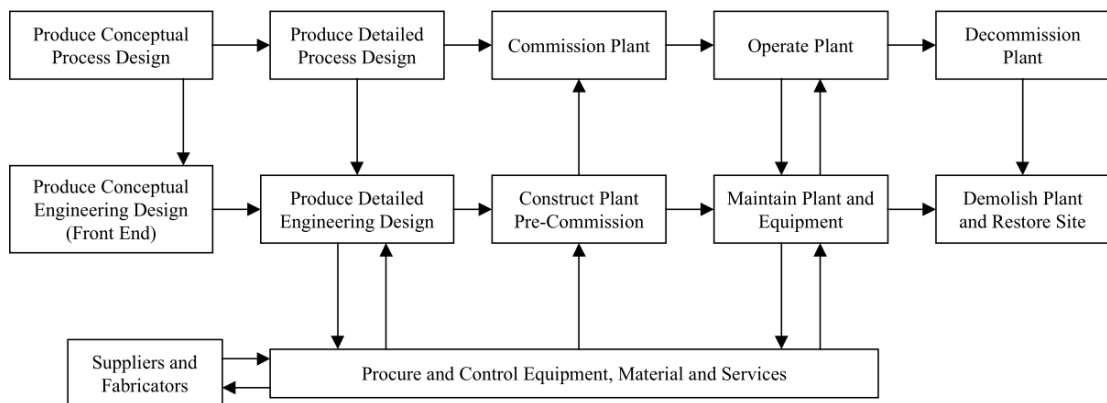


## 2 Relevante Technologien und Stand der Technik

Um das angestrebte Konzept zur Modellintegration umsetzen zu können, werden Informationen aus und über Anlagenstruktur, Automatisierungsstruktur und den verwendeten Automatisierungsgeräten benötigt. Dieses Kapitel soll einen Überblick über die verfügbaren Informationen geben und für die Umsetzung relevanten Informationen näher ausarbeiten. Dabei wird ein besonderes Augenmerk auf die Möglichkeit der automatischen Verarbeitung bzw. Auswertung gelegt.

Neben den Technologien, welche oft in internationalen Standards beschrieben und festgelegt sind, sollen auch wissenschaftliche Arbeiten vorgestellt werden, die eine ähnliche Nutzung der Technologien vorsehen wie diese Arbeit. Auch sollen veröffentlichte Vorarbeiten in diesen Kontext eingeordnet werden.

Die Informationen entstehen in unterschiedlichen Lebenszyklusphasen der Anlage. Abbildung 2.1 zeigt ein Aktivitätsmodell, welches zu den Lebenszyklusphasen korrespondiert. Dabei entstehen die relevante Informationen für Apparatemodelle vor allem in der Detailplanung (Detail Engineering) sowie der Inbetriebnahme (Commissioning) und dem Betrieb (Operation).



**Abbildung 2.1:** Lebenszyklus-Aktivitätsmodell (aus [13, Fig. 1, S. 7]: „Activity model of the oil and gas production facility life-cycle“, vgl. [14, Bild 5, S. 659])

Abschnitt 2.1 behandelt das Vorgehen und die Technologien des Engineerings von prozesstechnischen Anlagen. Die dabei entstehenden Planungsdaten werden auf relevante Inhalte untersucht und Möglichkeiten aufgedeckt, diese zu identifizieren und auf diese zu zugreifen. In Abschnitt 2.2 ab Seite 23 werden die im Betrieb einer Anlage anfallenden Daten kategorisiert und mögliche Zugriffswege dargestellt. Die dabei verwendeten Technologien und Standards werden ebenfalls vorgestellt. Abschnitt 2.3 ab Seite 28 gibt einen

Überblick über bestehende Technologien und Lösungen zur Zustandsüberwachung und Leistungsbewertung von prozesstechnischen Apparaten im Betrieb.

## 2.1 Engineering und Engineeringdaten

Der Engineeringprozess von prozesstechnischen Anlagen ist in Phasen aufgeteilt (vgl. Abbildung 2.1 auf Seite 7). Alle Aspekte einer Anlage (Prozess, Baustruktur, Verrohrung, Instrumentierung, Automatisierung, ...) werden in den Phasen mit unterschiedlichem Detailgrad geplant. In den folgenden Unterabschnitten werden die relevanten Aspekte kurz in die Phasen eingeordnet und auf Engineeringdaten und -technologien untersucht.

In Abschnitt 2.1.1 geht auf den zu automatisierenden Prozess ein, Abschnitt 2.1.2 auf die Anlage. Abschnitt 2.1.3 ab Seite 13 geht auf die Planung der Instrumentierung ein. Abschnitt 2.1.4 ab Seite 15 geht auf die Vernetzung der Automatisierungskomponenten ein und Abschnitt 2.1.5 ab Seite 17 auf deren Parametrisierung.

Das Engineering von Anlagen folgt einem hierarchischen Ansatz, die einer Top-Down-Methodik – vom Großen zum Kleinen und vom Groben zum Detaillierten. Dies zeigt sich unter anderem an der Formalisierung des Entwicklungsprozesses in der VDI/VDE-Richtlinie 3682 [15] und den resultierenden Strukturen [16, Abs. 5]. Dabei wird vermehrt versucht, alle Daten für ein Projekt in zentrale Engineeringdatenbanken zu integrieren, um einen einfacheren Zugriff zu erreichen (vgl. [14, Abs. 8.1.6, S. 672ff] und [17, S. 9, Fig. 1]). Abbildung 2.2 zeigt Interaktionspunkte und die zeitliche Einordnung der Phasen des Engineerings. Die VDI/VDE-Richtlinie 3695 [18] beschreibt Arbeitsprozesse [19], Methoden [20], Hilfsmittel [21] und Organisationsstrukturen [22], um den Engineeringprozess von Automatisierungsprojekten zu vereinfachen und projektübergreifend zu optimieren.

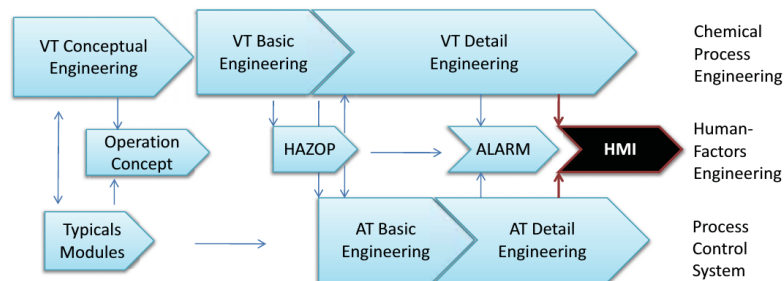


Abbildung 2.2: Interaktionspunkte verschiedener Gewerke im Engineering (aus [23, Fig. 1, S. 1])

Die in den folgenden Abschnitten beschriebenen Ergebnisse der Engineeringphasen ordnen sich in dieses Modell ein und erweitern die Datenbasis. Sie besitzen keinen Anspruch auf Vollständigkeit. So wurde auch bereits auf die Beschreibung von Modellen verzichtet, für die keine Instanzen oder Instanzdaten bzw. Werkzeuge zur Bearbeitung dieser vorhanden waren.

### 2.1.1 Prozess-Planungsdaten

Im verfahrenstechnischen Basic Engineering wird sowohl die Verfahrensauslegung als auch die Festlegung der Grobstruktur der Anlage durchgeführt [14, Abs. 8.1.4.2, S. 659; Abs. 8.1.4.3, S. 659f]. Hierbei kommen Grund- und Verfahrensfliießbilder zum Einsatz.



Diese sind jedoch aufgrund der geringen Detaillierung und der hohen Wahrscheinlichkeit späterer Änderung nicht zur Extraktion von Daten geeignet. Ebenfalls ist die Darstellung von den verschiedenen Bildelementen (Verfahrensschritte, Anlagenteile, Grundoperationen, ...) gleich und somit nicht verwertbar.

Weiterhin werden Prozesssimulationen [24] genutzt, um die Verfahrensauslegung zu unterstützen. Die Ergebnisse bleiben in den Daten enthalten, haben aber eher indirekten Einfluss auf weitere Planungsdaten.

Bei der Planung von modularen Anlagen ändert sich der Planungsprozess [25, Abs. 4.1, S. 17ff], modulare Anlagen liegen jedoch nicht im Fokus dieser Arbeit.

### 2.1.2 Anlagen-Planungsdaten

In der Detailplanung entstehen die R&I-Fließbilder. Dies sind die wichtigsten Dokumente [14, 8.1.4.3.3, S. 661], da sie als Ausgangspunkt für Aufstellungsplanung, Rohrleitungskonstruktion, Prozessleittechnik (PLT)-Engineering, etc. genutzt werden. Sie enthalten alle Strukturinformationen und besitzen eine standardisierte Darstellungsform. Die DIN 19227-1 [26] und ihr Nachfolger, die IEC 62424 [17] beschreiben die Notation von PLT-Stellen bzw. MSR-Anforderungen. Die ISO 10628 [27, 28] beschreibt Symbole für Apparate, Rohrleitungen, etc. In der IEC 81346-2 [29] werden Kennzeichnungssysteme für Anlagenteile festgelegt. Diese Richtlinien geben den teils grafischen Planungsdokumenten eine Semantik. Die graphischen Elemente sind durch eine maschinenlesbare Repräsentation im Computer Aided Engineering (CAE)-System hinterlegt. Somit können die Informationen aus R&I-Fließbildern prinzipiell für die Suche nach zu beobachtenden Apparaten genutzt werden.

Die meisten Engineering-Datenbanken liegen in proprietären Formaten vor (vgl. Abbildung 2.3 auf Seite 10). Sie sind dadurch ohne das Engineering-Tool, in dem sie erstellt wurden, nicht lesbar und nicht in maschinenlesbare Formate exportierbar. Die nächsten Abschnitte behandeln unterschiedliche Methoden und Formate, um einen maschinenlesbaren Export und den Transport der Semantik der Daten zu ermöglichen. Es ist durchaus möglich, R&I-Fließbilder in Office Tools anzufertigen [30, 4]. Diese Dateien lassen sich jedoch nicht ohne zusätzlichen Aufwand elektronisch weiter verarbeiten.

#### 2.1.2.1 Allgemeine Ansätze zum Austausch von Engineeringdaten

Es gibt eine Reihe theoretische Betrachtungen welche die Möglichkeiten des Austauschs von Engineeringdaten erörtern. In [31] wird der ein theoretischer Ansatz vorgestellt, der später in Computer Aided Engineering eXchange (CAEX) umgesetzt wurde. Dabei wird die Modellierung von Systemeinheiten beschrieben, die bestimmte Rollen umsetzen kann (Metamodell). Eine Rolle ist ein parametrierbares Fähigkeitsprofil. Somit lassen sich konkrete Anforderungen an Systemeinheiten, die in die Anlage verbaut werden sollen, in einem bekannten Rahmen festhalten (Systemmodell bzw. Anlagenmodell). Bereits hier wird die Notwendigkeit der Standardisierung von Rollen, also einer gewerke- und unternehmensübergreifenden Notation von Engineeringmodellen herausgehoben. Als Anwendung werden aus solchen Modellen Bewertungsmodelle abgeleitet [32, 33] (siehe Abschnitt 2.3 ab Seite 28).

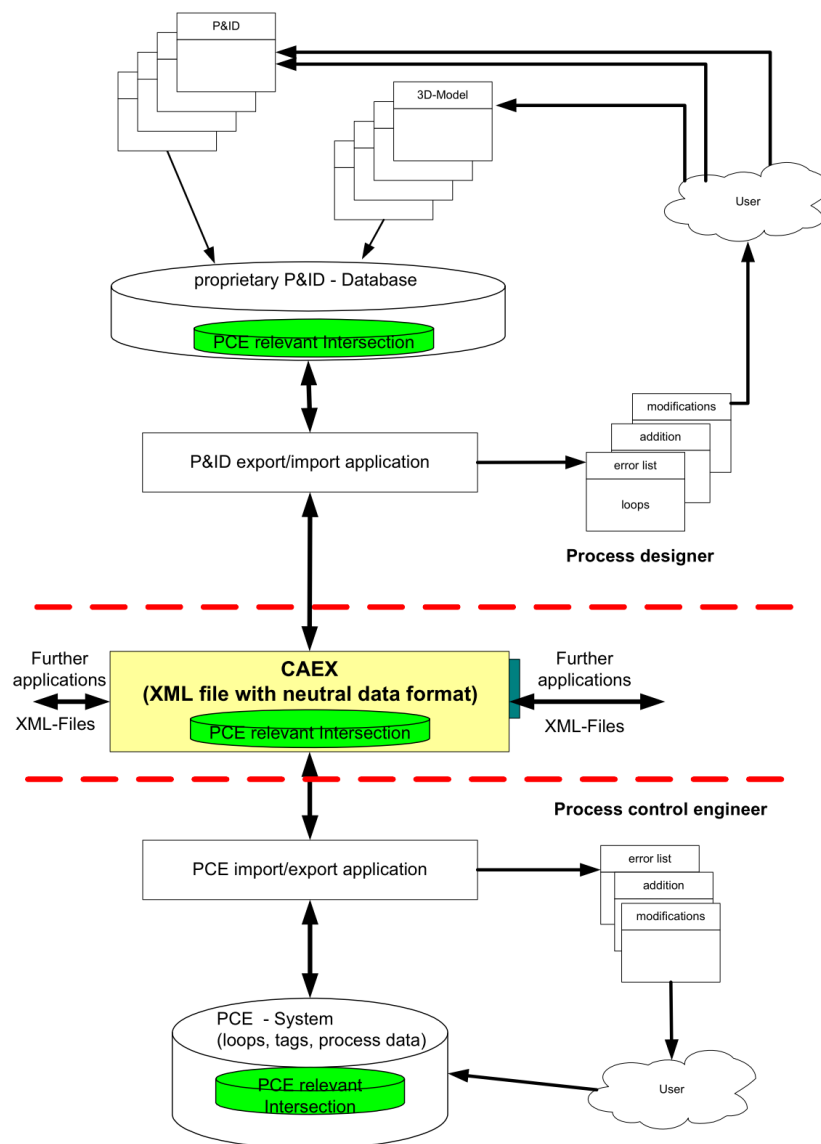


Abbildung 2.3: Exemplarische Darstellung von Informationsflüssen zwischen CAE-Tools (aus [17, Fig. 1, S. 9])

[34] beschreibt den Arbeitsprozess des Engineerings von Anlagen mit einem *Produkt-Prozess-Modell*. Die entwickelten Modelle beschreiben den zwischen Arbeitsprozessen übergebenden Informationshaushalt (Produkt) als Instanz eines Metamodells. Deren Beschreibung wird jedoch nur mit Hilfe von Partialmodellen beschrieben. Das birgt zwar die Gefahr einer Überschneidung von Modellen, erlaubt aber, nur die notwendigen Informationen zu modellieren [34].

In [35] und [36] wird eine Möglichkeit beschrieben, die in [34] ebenfalls angesprochenen Inkonsistenzen von Partialmodellen in einem Konsolidierungsprozess zusammen zu führen. Dabei wird auf semantische Technologien zurückgegriffen. Eine *globale Ontologie* welche das vollständige Anlagenmodellformat eines Betreibers beschreibt, wird auf unterschiedliche

*Quellontologien* abgebildet. Die Abbildung und Umwandlung der Instanzen (Inferenz) wird Anfrage-orientiert durchgeführt. Als Schwächen werden vor allem Rechenaufwand und die temporäre Verfügbarkeit genannt. Jedoch kann zusätzlich die schlechte Wartbarkeit der globalen Ontologie und deren Abbildung erkannt werden (Weltmodellproblem).

### 2.1.2.2 Computer Aided Engineering eXchange

Der CAEX-Standard IEC 62424 [17] wurde 2008 veröffentlicht und stellt mit der Schemaversion 2.15 ein offenes eXtensible Markup Language (XML)-Metamodell für den Austausch von Engineeringdaten zur Verfügung (vgl. Abbildung 2.3). Er erweitert das Modell aus [31] um Schnittstellenklassen und Instanzhierarchien. Die Version 3 [37] führt zusätzlich Attributklassen ein. Das gegebene Metamodell lässt sich nach Belieben instanzieren, was sowohl Stärke (beliebige Verwendbarkeit) als auch Schwäche (Notwendigkeit der Festlegung der Semantik des Modells) darstellt. Wie in [36] kritisiert, gibt es keine standardisierten Schnittstellen- und Rollenklassen, die die Semantik der Modelle transportieren können. Es besteht jedoch die Möglichkeit, offene Bibliotheken zu erstellen und für den Austausch vorzuschreiben. Das Format hat dafür keine Einschränkungen. Auch in [38] wird auf die Notwendigkeit der Standardisierung von Bibliotheken gemäß der standardisierten Darstellungsvorschriften (wie in Abschnitt 2.1.2 ab Seite 9 beschrieben) hingewiesen, um eine eindeutige Interpretation der Daten zu ermöglichen. Das wird in [39, Abs. 3.1-3.2] anhand der Modellebenen genau dargestellt. CAEX als Metamodell-Ebene (M0) wird in Form von Rollen- und Schnittstellenklassen instanziiert, was ein domänenspezifisches Modell (M1) ergibt. Dieses kann wiederum in der Instanzebene (M2) zu einem konkreten Teil der Engineeringdaten für eine konkrete Anlage instanziiert werden.

Aufgrund der Struktur des Metamodells können Domänenmodelle und Instanzen teilweise auf Vollständigkeit und Korrektheit automatisch überprüft werden. Arbeiten hierzu wurden in [3] veröffentlicht und werden in Abschnitt 4.1.1.3 ab Seite 58 weiter ausgeführt.

**Nutzung von CAEX** Wie bereits in den letzten beiden Abschnitten beschrieben, kann CAEX nicht ohne die Verwendung von domänenspezifischen, verbreiteten Bibliotheken effizient eingesetzt werden. In [40] wird ein Engineeringmodell vorgestellt, welches in CAEX abgelegt wird. Die dafür benötigten Bibliotheken sind offen gelegt und wurden in den Arbeiten, die in Abschnitt 4.1.1.3 ab Seite 58 und folgenden vorgestellt werden, verwendet. Mit Hilfe dieses Modells und dieser Bibliotheken können Engineeringdaten aus proprietären Engineeringtools exportiert werden [41].

In [42] wird ein Lebenszyklusmodell für die Standardisierung von Bibliotheken skizziert. Dabei wird die allgemeine Akzeptanz von Partialmodellen als höher eingeschätzt als von einem vollständigen Modell. Durch eine schrittweise, dafür jedoch durchgängige Umsetzung in den Import/Exportfiltern der meisten Engineering-Tool können herstellerneutrale Austauschformate auf Basis eines Metaformats wie CAEX standardisiert werden. Mit [43] wird ein solches Partialmodell für PLT-Stellen definiert, um den „Deadlock“ durch das Warten von Anwendern auf fertige Lösungen und das Warten der Hersteller auf Feedback aus der Anwendung zu durchbrechen.

Eine Anwendung der modellgetriebenen Erstellung von Domänenmodellen wird in [44] vorgestellt. Dabei werden eCl@ss-Gerätemodelle (siehe Abschnitt 2.1.3 ab Seite 13) in Rollenklassenbibliotheken übersetzt. Somit kann eine extern definierte Semantik übertragen

werden. Mit einer derartigen systematischen Umsetzung von Bibliotheken ist die Konformität zur Spezifikation und somit weitestgehend die Fehlerfreiheit leicht zu erreichen. Ebenfalls ist es unproblematisch, die Bibliotheken zugänglich zu machen.

**AutomationML** AutomationML (AML) ist ebenfalls eine Anwendung von CAEX. Es wurde entwickelt, um Anlagenstruktur und -verhalten für die Fabrikautomation abzubilden [45, 46]. Dabei wird CAEX zur Abbildung von Strukturen verwendet. Geometrie und Verhalten (z. B. von Robotern) wird mittels *COLLADA* (einer unabhängig von AutomationML (AML) entwickelten XML-Sprache) abgebildet.

Zudem wird eine Rollenklassenbibliothek [47, 48] entwickelt. Daneben gibt es eine frei verfügbare Programmbibliothek [10] und einen darauf aufsetzenden Editor [49] zum Bearbeiten von .aml-Dateien. AutomationML und diese Bibliotheken wurden in der IEC 62714 standardisiert [50, 51].

### 2.1.2.3 STEP

Zum Austausch von Engineeringdaten wurde mit dem Standard for the Exchange of Produkt Model Data (STEP, ISO 10303) ein Versuch unternommen, ein einheitliches Austauschmodell für CAE-Daten im Allgemeinen und für verschiedene Domänen im speziellen zu schaffen [14, 8.1.7.1, S. 675f]. Unter anderen ist mit dem AP 221 (ISO 10303-221) eine Modellbeschreibung für prozesstechnische Anlagen definiert worden [52, S. 91–95]. Die Modelle werden mit den textbasierten und graphischen EXPRESS-Sprachen (ISO 10303-11) gespeichert. Die Daten besitzen eine entsprechend standardisierte Codierung (ISO 10303-21). In der ISO 13584 [53] wird beschrieben, wie diese Modelle in Bibliotheken organisiert und Systemen, die sie nutzen, zur Verfügung gestellt werden. Dabei werden unterschiedliche Repräsentationen der Modelle verknüpft und organisiert. Auch der Austausch von Modellen über Bibliotheken wird beschrieben.

**ISO 15926** Der Standard ISO 15926 [13, 54] greift die STEP-Norm auf und erweitert sie speziell für die Öl- und Gasindustrie. In Teil 2 wird ein Metamodell definiert und mit domänenspezifischen Beispielen unterlegt. Jedoch gibt es keine mit diesem Standard festgelegten Domänenmodelle. Teil 3 [55] definiert eine Methodik, diese zu erstellen.

**XMpLant** XMpLant ist ein Austauschformat basierend auf einem ISO 15926 zu XML-Mapping. Es wurde unter anderem durch das *Proteus*-Projekt [56, S. 4] voran gebracht und wird von der *Fiatech* [56, S. 3] koordiniert. Ein Beispiel für diese Aktivitäten ist das *Profil* für R&I-Fließbilder. Es besteht aus einer Beschreibung des Informationsmodells [57], einer Schema-Spezifikation [58] und einer Template-Spezifikation [59].

### 2.1.2.4 PlantXML

PlantXML ist ein XML-Austauschformat für Engineeringdaten, welches von und für die *Degussa* (heute: *Evonik*) entwickelt wurde [60]. Es sieht ein XML-Austauschformat für verschiedene Arten von Engineeringdaten vor. Aufgrund der Bindung an firmeneigene Vorgaben ist es weniger universell, dafür aber einfacher handhabbar. Es wurden semantische Technologien eingesetzt, um Daten besser zu konsolidieren [30] (vgl. [35, 36]). Später

wurden auch Standards oder Definitionen wie ISO 15926 oder XMpLant [61, S. 21], [62] verwendet.

#### 2.1.2.5 Formalisierte Prozessbeschreibung

In der VDI/VDE-Richtlinie 3682 [15] wird die übliche Engineeringmethodik derart formalisiert, dass nach der Zergliederung der Prozessschritte in verschiedenen Granularitäten und deren Edukte und Produkte, Energien und Ressourcen in formal prüfbarer Form vorliegen. Für diese Methodik wurde ein Editor entwickelt, der die Symbolik der Richtlinie verwendet und die Modelle in einer XML-Datei ablegt. Dieses formale Vorgehen hat sich in der Praxis jedoch nicht durchsetzen können und wird gelegentlich im wissenschaftlichen Umfeld genutzt [63].

#### 2.1.2.6 Anwendung von Engineeringdaten

In verschiedenen wissenschaftlichen Arbeiten wird versucht, Arbeitsprozesse unter der Verwendung von Engineeringdaten-Exporten zu automatisieren.

[64] beschreibt eine technische Lösung zur Konsistenzsicherung von Engineeringdaten und Simulationsmodellen. In [65] werden Modelle (Bond Graphen) zur Flussweganalyse aus CAEX erzeugt. In [66] wird eine Methode zur automatischen Generierung von PLS-Funktionen aus CAEX beschrieben. [67] generiert Visualisierungsinformationen. [68] bzw. [69] generiert eine OPC UA-Informationsmodell (siehe Abschnitt 2.2.1.3 ab Seite 26) aus CAEX. Ebenfalls gibt es Arbeiten zur Integration von Engineeringdaten in externe Systeme. In [70] und [71] werden Methoden zu Web-basierten Anfragen auf CAE-Daten behandelt. [10] beschreibt die Zuordnung von über die Kommunikationsschnittstelle ermittelten Feldgeräten zu deren Planungsdaten im Betrieb.

Zusammenfassend lässt sich feststellen, dass unter der Verwendung von Engineeringdaten eine bessere Unterstützung von Arbeitsprozessen in allen Lebenszyklusphasen einer Anlage möglich ist.

In [72] führt eine formale Analyse der unterschiedlichen Engineeringmodelle durch und zeigt Möglichkeiten der Nutzung auf. Mit Hilfe von semantischen Technologien wird die Möglichkeit einer Wissensverarbeitung auf Basis der Engineeringmodelle aufgezeigt.

### 2.1.3 Instrumentierungs-Planungsdaten

Das Anlagen-Engineering definiert MSR-Anforderungen, die von der Automatisierungstechnik technisch umgesetzt werden müssen. Diese PLT-Stellen besitzen funktionale und nicht-funktionale Anforderungen, die bei der Auswahl von konkreten Instrumenten beachtet werden müssen. Das sind zuerst die benötigte Funktion, z. B. Messen eines Durchflusses, aber auch verschiedene technische Randbedingungen z. B. der Prozessanschluss, thermische, chemische und mechanische Anforderungen, Kommunikationsanschaltung und viele mehr [72, Abb. 14, S. 22]. Diese sogenannten Merkmale [73, Abs. 3.4.3, S. 39ff], [74] wurden in den PLT-Stellenblättern in der Merkmalleiste eingetragen. Diese wurden zur automatischen Verarbeitung modelliert und standardisiert. Die Merkmalleisten beschreiben demzufolge unterschiedliche Aspekte der beschriebenen Geräte bzw. Gerätschaften [75, S. 2]. Darunter Identifikation, Bestellung, Zertifizierung, Prozess- und Umweltbedingungen, etc. Daraus

können auch Geräteeinstellungen wie zum Beispiel Stellenbezeichnung oder Grenzwerte abgeleitet und vom Gerätehersteller vor Auslieferung gesetzt werden.

Der GMA-Arbeitskreis 5.16 hat auf Basis von CAEX bzw. AML ein standardisiertes Austauschformat für PLT-Stellen definiert [43]. Diese integriert Aspekte des verfahrenstechnischen, PLS-, und PLT-Engineering sowie der Elektroplanung und der Geräteauswahl.

In [76] werden neben anderen Eigenschaften funktionalen Fähigkeiten formalisiert und in einem Informationsmodell abgelegt. Diese Rollenklassifizierung wird dort über Applikationsprofile vorgenommen. [77] gibt einen weiteren Überblick über Merkmalsysteme zum Produktdatenausch.

**PROLIST und eCl@ss** Die elektronische Verarbeitung verbessert die Datenkonsistenz im Engineering und eine mögliche automatische Auswahl bzw. eine Einschränkung der Optionen von Feldgeräten. Die NE 100 [78] beschreibt die Anforderungen der Geräteanwender (Anlagenplaner und Betreiber) zu dieser Integrationsaufgabe. In der Version von 2010 [73, Abs. 3.4.3, S. 39] bezieht sich auf die beiden Initiativen und deren (Quasi-)Standards eCl@ss und die später darin integrierte PROLIST, die aus diesen Anforderungen abgeleitet wurden (vgl. [75, Abs. 5, S. 4ff]).

Für prozesstechnische Anlagen relevante Merkmalleisten wurden sowohl in der PROLIST- als auch in einer Fachgruppe der eCl@ss-Initiative entwickelt und synchronisiert [75]. Ausgegangen sind beide Initiativen vom Bedarf, die Fehlerquellen bei der manuellen Übertragung der Merkmalsdaten zu beseitigen [79, Abs. 3, S. 4ff].

„PROLIST enthält in der Version 3.2 die Spezifikation von 110 Gerätetypen wie z. B. für Füllstandsensoren, Messumformer oder Absperrventile, die in weiteren Teilen der IEC 61987 [80] bzw. damit verwandter Normen semantisch standardisiert sind bzw. aktuell eingebracht werden. Jeder Gerätetyp wird durch die Administrative Merkmalleiste (AML), Betriebsmerkmalleiste (BML), Gerätemerkmalleiste (GML) sowie die kommerzielle Merkmalleiste (KML) beschrieben. Jede Merkmalleiste besteht aus Merkmalen, die durch Blöcke in beliebiger Hierarchietiefe zusammengefasst werden können.“ ([72, S. 116]) Die verschiedenen Aspekte teilen sich auf mehrere Teile der Merkmalleiste auf. Die eCl@ss-Merkmalleisten besitzen nach [78] nur technische Eigenschaften. Diese sind jedoch mit der Harmonisierung ab 2011 [81, Bild 6, S. 7] ebenfalls eingebracht worden. Ebenfalls besitzt die eCl@ss-Bibliothek eine wesentlich größere Breite in der Anwendung.

Die Merkmalleisten sind von den unterschiedlichen Gremien in unterschiedliche Standardisierungsprozesse eingebracht worden. Die NE 100 [78] wurde erst als DIN PAS 1040 [82] spezifiziert, dann auf Basis der PROLIST-Definitionen auf das Niveau eines internationalen Standards gehoben [81, Bild 7, S. 8], [75, S. 4], [83, S. 3–5]. Die IEC 61987 [80, 84, 85] wird mit der nächsten Überarbeitung auf den synchronisierten Inhalten der eCl@ss fußen.

In der Anwendung konnte [86] zeigen, dass die im Engineeringprozess erstellten Werte für Merkmale einer PLT-Stelle ohne Medienbrüche an einen FDI Server (siehe Abschnitt 2.1.5.4 ab Seite 20) an das umsetzende Gerät übertragen werden konnte. Ein weiter reichender Ansatz [44] zeigt, wie bereits in Abschnitt 2.1.2 ab Seite 9 angerissen, wie die standardisierten Merkmale aus den eCl@ss-Merkmalleisten als Semantikdefinitionen für Bibliotheken für Engineeringdaten genutzt werden können.

**Geräteprofile** Ein weiterer Aspekt der Vereinfachung des Engineerings im Bereich der Instrumentierung ist der einheitliche Zugriff auf Geräte. Dies ist mit Feldgeräteprofilen möglich. Diese werden in Abschnitt 2.1.5 ab Seite 17 behandelt.

#### 2.1.4 Vernetzungs-Planungsdaten

Ein für die Abdeckung dieser Arbeit relevanter Aspekt ist die informationstechnisch durchgängige Vernetzung von intelligenten Feldgeräten. Die Vernetzungsplanung ist Teil der Anlagenplanung und basiert sowohl auf den vorhandenen geplanten Kommunikationsbeziehungen zwischen PLS und Feldgeräten als auch auf der Aufstellungsplanung und den vorgesehenen Kabeltrassen. Bei der Auswahl der Kommunikationstechnologien sind sowohl Umweltbedingungen (Schutzart, Ex-Sicherheit, ...) als auch maximale Kabel- oder Buslängen zu beachten. Genauso sind Technologie- oder Produkt-spezifische Einschränkungen wie die Anzahl von Busteilnehmern oder maximale Übertragungsgeschwindigkeiten relevant (vgl. [87]).

Die Gesamtarchitektur ist eine hierarchische Netztopologie bestehend aus Feldgeräten, Prozessnahen Komponenten (PNK) und Anzeige-Bedien-Komponenten (ABK) (siehe Abbildung 2.4 auf Seite 16; vgl. [14, Abs. 4.1.2.2, S. 208]). Die Vernetzungsplanung liegt wie alle anderen Planungsdaten in der CAE-Datenbank und lässt sich mit den in Abschnitt 2.1.2 ab Seite 9 beschriebenen Technologien und Methoden zugreifen. Zusätzlich kann bei entsprechender Durchgängigkeit der Kommunikationsinfrastruktur die Topologie im Betrieb analysiert werden (vgl. Abschnitt 2.1.5 ab Seite 17 und Abschnitt 2.2.2 ab Seite 26).

##### 2.1.4.1 Kommunikationstechnologien

In der Prozessindustrie sind im Feld zwei unterschiedliche Klassen von Kommunikationstechnologien vorherrschend. Das ist zum einen die klassische Zweidrahtleitung mit 4-20 mA Analogsignal oder 0/10 V bzw. 0/24 V Digitalsignal [14, S. 277ff], [89, S. 2], die vor allem in älteren Anlagen anzutreffen ist. Erweitert werden kann eine solche Installation mit dem HART-Digitalprotokoll [14, S. 282ff], welches auf die Stromsignale aufmoduliert wird. Zum anderen sind verschiedene Feldbusse anzutreffen [14, S. 297ff]. Außerdem existiert eine Mischform, die sich beider Technologien bedient: die Anschaltung mittels Remote IO im Feld [90].

**Klassische Feldkommunikation und HART** Die sternförmige Verdrahtung mit Zweidrahtleitungen besitzt einen vergleichsweise hohen Verdrahtungsaufwand und einen dafür geringen Funktionsumfang [91, S. 11]. Zudem muss die Parametrierung des Messbereichs am Gerät erfolgen. Dies lässt sich unter der Verwendung von HART-fähigen Feldgeräten vermeiden, jedoch ist die HART-Kommunikation nur auf einem Kanal eines Multiplexers gleichzeitig möglich und zudem sehr langsam. Ein regelmäßiges Auslesen über Feldgeräte scheint somit unpraktikabel.

**Feldbusse** Feldbusse bieten neben einem verringerten Verkabelungsaufwand auch die Möglichkeit, Feldgeräte zentral zu parametrieren, und verschiedene Zusatzfunktionen (z. B. Selbstüberwachung) ohne Mehraufwand einzurichten [91, S. 12]. Feldbusse prägen im *ISO-*

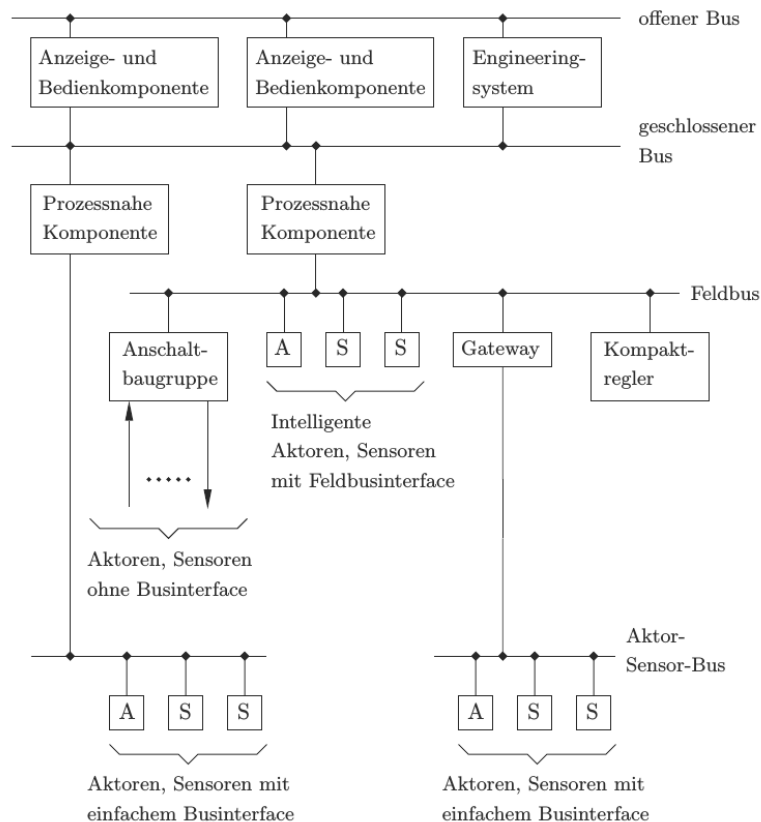


Abbildung 2.4: Struktur eines modernen Prozessleitsystems (aus [88, Bild 1.2])

*OSI-Schichtmodell* üblicherweise nur die Schichten 1, 2 und 7 aus. Die anderen Schichten sind aufgrund der Netzstruktur nicht notwendig [92].

Die meisten Feldbusse wurden nach ihrer Spezifikation auch international genormt. Jedoch ist durch die Vielfalt an Technologien eine unübersichtliche Menge von Standards entstanden. [93] und [94] bietet eine gute Übersicht. Man trennt bei der Standardisierung in Protokollfamilien, [95] welche die Übertragungsprotokolle spezifizieren und Profildfamilien, welche für die einzelnen Protokolle ein oder mehrere physikalische Übertragungsschichten definieren. Die für diese Arbeit wesentlichen Vertreter sind (PROFIBUS) Process Automation (PA) und Foundation Fieldbus (FF). Der azyklische Zugriff auf die Daten ist mit diesen Technologien problemlos möglich. HART ist in diesen Normen spezifiziert.

Die Interessengemeinschaft Automatisierungstechnik der Prozessindustrie (NAMUR) hat mit [89, 96, 87] Anforderungen und Erfahrungswert zu Feldbussen und Feldbusverschaltungen formuliert. Weitere Feldbusse z. B. *I/O-Link* [97, 98] und Controller Area Network (CAN) [99] mit darauf aufsetzenden Protokollen wie z. B. *CANopen* [100, 92] sind in der Fertigungstechnik angesiedelt und für diese Arbeit weniger relevant.

**Remote IO** Remote IOs sind modulare Feldgeräte, welche verschiedene Eingabe-/Ausgabebaugruppen gruppieren und über eine Feldbusschnittstelle ansprechbar machen [90]. So können klassische Feldgeräte angeschlossen werden und ggf. sogar über integrierte HART-Modems angesprochen werden. Deren Primärmesswerte werden von der Remote IO digital



über den Feldbus übertragen. Ein regelmäßiges azyklisches Auslesen dieser Werte über den Feldbus ist möglich. Ebenfalls ist ein Zugriff über HART denkbar, wenn die Remote IO gleichzeitige Zugriffe auf die unterschiedlichen Kanäle unterstützt.

### 2.1.4.2 Planung

Das Engineering der Vernetzung betrifft unterschiedliche Gewerke des Engineerings. Die Planung von Schaltschränken muss in der Bauplanung berücksichtigt werden; Feldgeräte werden nach Verfahrens- und Umwelanforderung sowie Kommunikationsinfrastruktur ausgewählt. Die Lage von Kabeltrassen aus der Elektroplanung wird benötigt und fügt dieser eventuell zusätzliche Kabelwege hinzu. Die Randbedingungen haben Rückwirkung auf die Auswahl von Technologien. Die Vernetzungsplanung lässt sich unter Verwendung und Berücksichtigung der Vorarbeiten (teil-)automatisieren [101]. Die Planung der logischen Verschaltung der Feldgeräte, die Zuweisung der Adressen und die Parametrierung ist meist in den Engineering-Tools der Hersteller der Automatisierungsgeräte bzw. im PLS vorgesehen. Ein Import der relevanten Planungsdaten vereinfacht diese Aufgabe. Entsprechende Dokumente sind demzufolge wahrscheinlich verfügbar und können ebenfalls für andere Zwecke verwendet werden. Ebenfalls ist diese Struktur später im Parametrier-Tool hinterlegt (siehe Abschnitt 2.1.5.4 ab Seite 20).

### 2.1.4.3 Engineering der Basiskommunikation

Vor der Montage oder der Inbetriebnahme der Feldgeräte wird, wenn nicht schon vom Hersteller vorgenommen, im PLS die eine Adresse und die Messstellenbezeichnung gesetzt [87]. Die Geräte werden dann installiert, worauf die Kommunikation mit dem PLS eingerichtet werden kann. Hierfür ist eine Geräte- und Feldbustechnologie-spezifische Gerätebeschreibung verfügbar. In [102] wurde eine Klassifikation der dieser Gerätebeschreibungen durchgeführt. Für die Basiskommunikation ist nur eine einfache Beschreibung (z. B. General Station Description (GSD) bei PROFIBUS (Process Field Bus) oder Common File Format (CFF) bei Foundation Fieldbus) erforderlich oder nur bestimmte Teile (z. B. im Electronic Data Sheet (EDS) für *CANopen*).

Weiterhin gibt es verschiedene Mechanismen zur Identifikation von Geräten über den Feldbus wie zum Beispiel Identification&Maintenance (I&M)-Funktionen [103, 104], [105, Abs. 5.8.2, S. 185]. Zusätzlich werden in neuen Forschungsgebieten Ansätze behandelt, die die Kommunikation dynamisch und zwischen den Akteuren (*peer2peer*) aushandeln (Selbstkonfiguration) [106, 107, 108, 109]. Diese werden jedoch aufgrund von Sicherheitsbedenken und Zertifizierungsanforderungen in der Prozessindustrie derzeit nicht in Betracht gezogen.

### 2.1.5 Parametrierungs-Daten

Der aufwändigste Teil der Inbetriebnahme eines komplexen Feldgerätes ist die Parametrierung der Funktionen wie Alarmer, Kalibrierung, Plant Asset Management (PAM)-relevante Datensätze und andere messstellenspezifische Einstellungen. Diese Daten können aus den Planungsdaten entnommen werden und sind u. U. schon mit der Bestellung an den Gerätehersteller übertragen worden und bereits individuell voreingestellt [87]. Nichtsdestotrotz muss diese Einstellung vorgenommen werden. Es gibt hierfür unterschiedliche Technologien

zur Feldgeräteintegration [110], die von [111] als Middleware klassifiziert werden. Die NAMUR definiert mit der NE 105 [96] Anforderungen an diese Integrationstechnologien, darunter Vermeidung von Software- und Updateabhängigkeiten und Kompatibilität unterschiedlicher Versionen. In [102] wurden verschiedene deskriptive Methoden für diese Aufgaben klassifiziert und unterschiedliche Gerätebeschreibungssprachen analysiert.

### 2.1.5.1 Allgemeine Modelle

Durch die geringe Vielfalt von Gerätetypen in der Prozessindustrie und die Anforderung der gleichartigen Bedienung von Feldgeräten mit der gleichen Funktion wurden Profile gefordert. Die Profile Guideline IEC 62390 [112] beschreibt Vorgehensmodelle und konkrete Kommunikationsstrukturen; die ISO 15745-1 beschreibt Metamodelle, unter anderem zur Beschreibung von Geräteeigenschaften (als Teil des *Application Interoperability Profile*). Diese Vorgaben sollen Feldbusorganisationen dienen, um gleichartig strukturierte Profile zu erzeugen. Die Profile sollen Interoperabilität und eventuell transparente Austauschbarkeit von Geräten auf Ebene der Kommunikationsschnittstelle ermöglichen.

Feldbusorganisationen haben auf dieser Basis eigene Profilrichtlinien und Modelle (z. B. Spezialisierungen der ISO 15745 [113, 114] oder Vorschriften für Profile wie die PROFINET IO (PN IO)-Profile-Guideline [115] oder die I&M-Guideline [104]) erstellt. Weiterhin wurden Profile definiert wie für PA-Geräte [116] mit den zugehörigen Amendments [117, 118, 103]. Diese gehorchen auch anderen Standards wie beim PA-Profil das umgesetzte Funktionsblockmodell der IEC 61804-2 [119] (vgl. [105, Abb. 5–7, S. 140]). Auch zum Common Industrial Protocol (CIP) [120] wurden Geräteprofile definiert, welche Austauschbarkeit von Geräten mit dem gleichen Profil ermöglichen sollen [120, Part 6]. In [121] wird eine Methode und eine Werkzeugkette vorgestellt, mit der man aus einem formal beschriebenen Geräteprofil die Konfiguration für profilkonforme, virtuelle Feldgeräte erzeugt werden kann.

Auch mit anderen Methoden wurde versucht, einheitlichen Zugriff auf heterogene Gerätebestände zu ermöglichen. So wurde in [122] ein einheitliches Informationsmodell definiert und die Feldbus- und Gerätespezifika darauf abgebildet.

Eine einheitliche, durchgängige und verbreitete Zugriffsmethode auf Feldgeräte vereinfacht nicht nur Inbetriebnahme sowie Betrieb und Wartung der Geräte sondern ermöglicht ebenfalls wie für diese Arbeit benötigt eine aufgabenspezifische, automatische Suche und Identifikation von Geräten und Daten für systemfremde Zusatzkomponenten. Diese Methoden stützen sich meist auf Gerätebeschreibungen in unterschiedlicher Ausprägung.

### 2.1.5.2 Electronic Device Description Language

Die Electronic Device Description Language (EDDL) [123] ist in der IEC 61804-3 [124] spezifiziert und beschreibt das Funktionsblockmodell aus der IEC 61804-2 [119]. Im Rahmen der Field Device Integration (FDI)-Initiative wurde die Sprache zuletzt überarbeitet [125, 126]. Es gibt drei Varianten der EDDL. Die Struktur unterscheidet sich für PA- FF- und HART-Geräte. Ebenfalls sind unterschiedliche Builtin-Funktionen [127] erlaubt. Mit der durch FDI angestrebten Harmonisierung [128] sind die Builtins weitestgehend vereinheitlicht. Zusätzlich gibt es ein Kommunikationsprofil für PROFINET und (PROFIBUS) Dezentrale Peripherie (DP).

Generell beschreibt jede dieser Varianten die Gerätestruktur und die Einstellmöglichkeiten (den Informationshaushalt) über die Feldbusschnittstelle (Kommunikationssicht, vgl. Abbildung 2.5). Ein Parametrierungswerkzeug speichert die Electronic Device Description (EDD), die mit dem Gerät ausgeliefert wird, und legt sie im System ab. Wird auf das Gerät zugegriffen, wird sie geparkt und dem Nutzer eine einheitliche Oberfläche mit den Parametern des Gerätes präsentiert. Im Hintergrund kann das Werkzeug die Einstellungen aus dem Gerät laden oder speichern. Die EDDL erlaubt auch Methoden in einem C-Dialekt zu implementieren, z. B. um Berechnungen durchzuführen oder komplexe Abläufe in der Gerätekommunikation zu beschreiben.

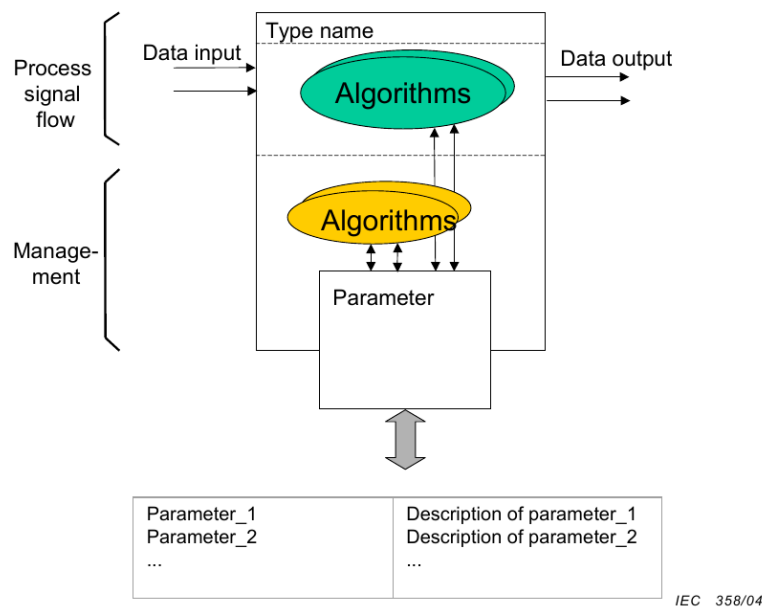


Abbildung 2.5: Funktionsblöcke und Parameter in Feldgeräten (aus [119, Fig. 7, S. 19])

Zur Erstellung und Bearbeitung von EDD-Dateien können Werkzeuge wie die *is EDD Workbench* [129] verwendet werden. Ein verbreitetes EDD-Parametrierwerkzeug ist *SIMATIC PDM* [130]. EDD-Dateien können auch über ihren eigentlichen Zweck hinaus verwendet werden. Zum Beispiel wird sie in [8] zur Abstraktion des Feldgerätezugriffs verwendet; in [131] dient ein EDD-Parser als Informationsmodell-Adapter.

### 2.1.5.3 Field Device Tool

Field Device Tool (FDT) [132] ist ein Softwareframework zur Integration von Feldgeräten. Es ist parallel zu EDDL entstanden und versucht das Problem der einheitlichen Parametrierung mittels einheitlich integrierbaren Softwarekomponenten, den sogenannten Device Type Managern (DTMs), zu lösen. FDT ist in der Normenreihe IEC 62453 standardisiert [133]. Teil 2 [134] beschreibt die Schnittstellen zwischen FDT-Frameapplication und dem DTM (siehe Abbildung 2.6 auf Seite 20). Die 300er-Teile (z. B. [135], [136]) beschreiben die Integration der verschiedenen IEC 61784-Kommunikationsprofile [137]. Der Teil 41 legt das Implementierungskonzept fest und schreibt dabei (für einen Standard ungewöhnlich) eine proprietäre Technologiebasis (*Microsoft Windows, ActiveX, Component*

Object Model (COM)) vor. Um ein einheitliches Bedienkonzept zu ermöglichen, wurde von der PROFIBUS Nutzerorganisation (jetzt PROFIBUS International (PI)) (PNO) ein Styleguide [138] für die Entwicklung von DTM-Oberflächen herausgegeben.

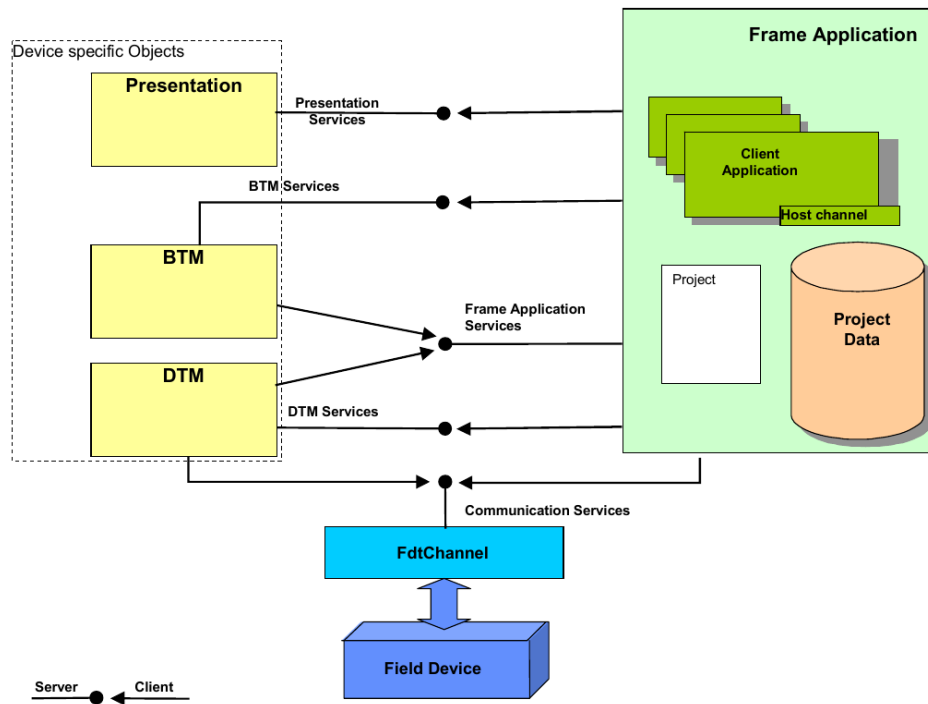


Abbildung 2.6: Field Device Tool-Softwarearchitektur (aus [133, Fig. 3, S. 14])

Bei der Integration eines Gerätes wird der vom Gerätehersteller programmierte DTM in die Rahmenapplikation (z. B. *PactWare* [139], *FieldCare* [140], *Filedmate* [141]) geladen und dargestellt. Über die verschiedenen Schnittstellen der Rahmenapplikation kann der DTM auf persistent abgelegte Daten (z. B. offline Parametersätze) und die Kommunikationsschnittstellen zu den Feldgeräten zugreifen. Das Festlegen von Parametern und ausführen von Gerätefunktionen passiert gemäß der vom Gerätehersteller festgelegten Abläufe.

Obwohl keine direkten Eingriffe ins Informationsmodell der DTMs möglich sind, gibt es Versuche, die FDT-Technologie für weiterführende Informationsgewinnung einzusetzen. [122] versucht Teile der Parameter für übergeordnete Zwecke auszulesen. Wie in [142] beschrieben, nutzen auch PAM-Tools wie die eben genannten [140, 141] FDT für den Zugriff auf Geräte. Bei frühen Konzepten von FDI [131] sollte auch auf FDT für ausprogrammierte Oberflächen zurückgegriffen werden.

#### 2.1.5.4 Field Device Integration

Aus dem Konzept der Trennung von Geräteinformationsmodell und dessen Darstellung [143, 144, 145, 131, 146] ging eine Initiative zur Vereinheitlichung der Integrationstechnologien hervor. Firmen, die die unterschiedlichen Technologien getrieben hatten, gründeten die *FDI-Corporation* und arbeiteten an der Harmonisierung der EDDL [147, F. 22–23] und

an internationalen Standards [148, S. 9] für die Festlegung der Informationsmodelle und Kommunikationsmechanismen (später IEC 62769 [149]).

FDI verspricht eine einheitliche Integration von Geräten auf einer einheitlichen technologischen Basis [150, 147, 151, 152, 148]. Das Informationsmodell ist als OPC UA-Modell (siehe Abschnitt 2.2.1.3 ab Seite 26) notiert [153] und sollte auch über OPC UA zugreifbar sein [149]. Zum Zeitpunkt der letzten Recherche vor der prototypischen Implementierung des in dieser Arbeit beschriebenen Konzepts gab es keine Implementierung außer der von der *FDI-Corporation* in Auftrag gegebene Referenzimplementierung [154]. Eine Unterstützung von OPC UA ist darin nicht vorgesehen. Die Spezifikation enthält die Formulierung, dass der Zugriff mittels OPC UA optional wäre [147, F. 15]. In [152, S. 34] wird diese Funktionalität jedoch beschrieben, als wäre sie zwingend notwendig. Sie wird zumindest von Anwendern wie der NAMUR schon früh gefordert [155, F. 9]. Nach Abschluss der Implementierung wurde die OPC UA-Unterstützung im Referenzhost eingeplant [156]. Ein erstes Produkt mit OPC UA-Implementierung wurde im zweiten Quartal 2015 ebenfalls vorgestellt [157]. Es unterstützt über EDD und FDT auch alle älteren Geräte, für die kein FDI Package mehr zur Verfügung gestellt werden wird.

**Standards** Die FDI-Spezifikationen der FDI-Corporation sind in die internationale Normung bei der IEC eingeflossen. Die sieben Teile behandeln die unterschiedlichen Aspekte und Komponenten von FDI. Zusätzlich gibt es noch Profile für die verschiedenen Kommunikationstechnologien wie PA und FF.

Teil 1 [149] gibt einen Überblick über die Technologie. In Abbildung 2.7 auf Seite 22 sind die Komponenten vorgestellt. Die Kommunikationsinfrastruktur unten in der Mitte der Abbildung entspricht der heutigen Integrationstechnologien. Zusätzlich erlaubt FDI eine dezentrale Zugriffsmethode über FDI Communication Server [158].

Die zentrale Komponente ist der FDI Server [159]. Er hat Zugriff auf alle Feldgeräte über die zwei spezifizierten Zugriffsmethoden. Für jedes Gerät hält er ein Informationsmodell [160] vor, welches mithilfe von Gerätebeschreibungen instanziiert wird. Es ist ein informationstechnischer Stellvertreter für das entsprechende Gerät. Das Informationsmodell basiert auf dem (OPC UA) for Device Integration (OPC UA-DI)-Informationsmodell [161]. Die Gerätebeschreibung folgt der harmonisierten EDDL [126].

Je nach Nutzeraktion und gemäß der Regeln in der Business Logic (BL) von Server und Gerätebeschreibung wird mit dem Gerät kommuniziert und die Daten im Informationsmodell mit denen im Gerät synchronisiert. Ebenfalls können Parametersätze *offline* gespeichert werden.

Der FDI Client [162] stellt dem Nutzer das Informationsmodell mithilfe von per EDDL beschriebenen Standardoberflächen (User Interface Descriptions (UIDs)) oder programmierten, gerätespezifischen Oberflächen (User Interface Plugins (UIPs)) dar. Für die verschiedenen Nutzerrollen werden unterschiedliche Teile des Informationsmodells angezeigt.

Die Gerätebeschreibung mit Device Definition (DD), UID und BL und optionale UIP werden vom Gerätehersteller als FDI Package [163] ausgeliefert. Die FDI Packages werden gemäß der *Open Packaging Conventions* [164] zusammengestellt.

Die IEC 61804-4 [165] legt die Umsetzung der Menüdefinitionen aus der EDD in die einzelnen Elemente der beschriebenen Benutzungsoberflächen fest. Die BL wird in der

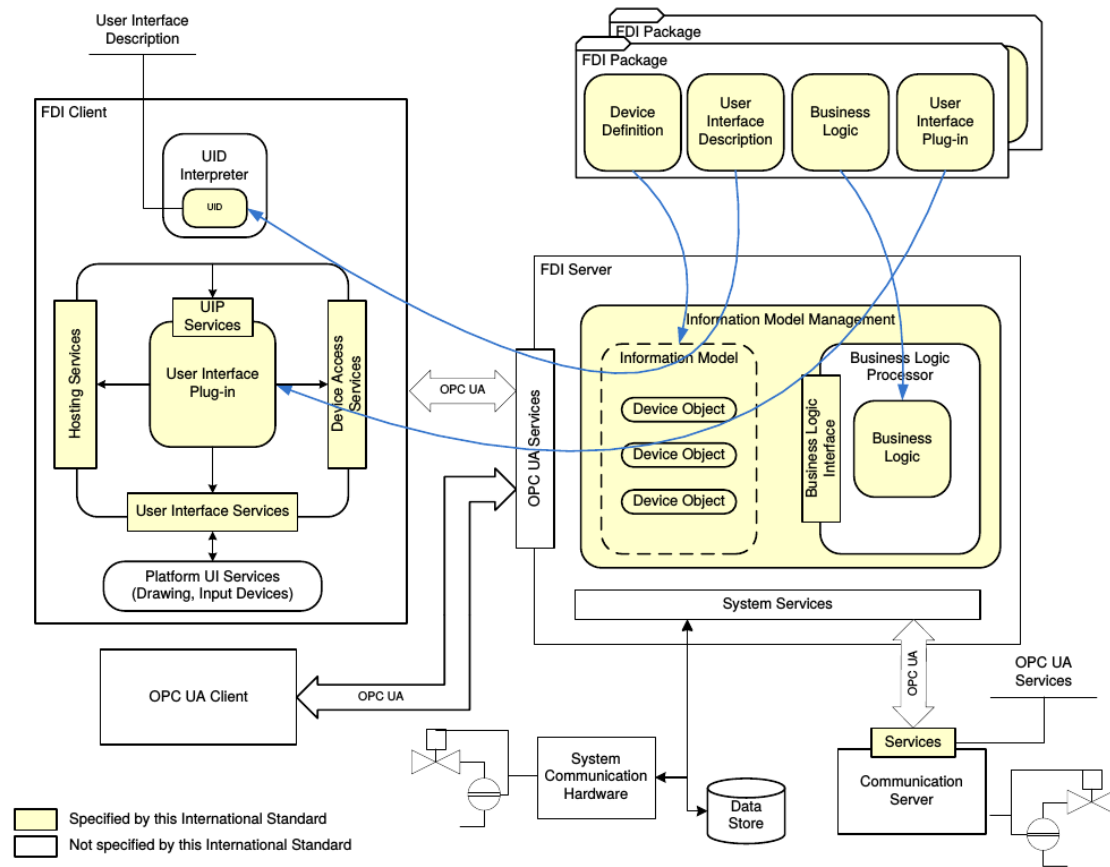


Abbildung 2.7: FDI-Architektur (aus [149, Fig. 1, S. 17])

EDD in Pre/Post-Read/Write-Ereignissen sowie Methoden festgelegt. Darin können die harmonisierten *Builtins* aus der IEC 61804-5 [127] verwendet werden.

Die UIP werden in [166] weiter beschrieben. Wieder wurde wie bei FDT unnötigerweise eine Bindung an proprietäre Technologien/Produkte mit wahrscheinlich zu kurzer Lebenszeit festgeschrieben (.NET-Framework (.NET) und Windows Presentation Foundation (WPF)).

Bei der Erstellung der Benutzungsoberflächen (UIDs und UIPs) soll der FDI Styleguide [164] beachtet werden, um für die Benutzer für unterschiedliche Geräte ein einheitliches *Look& Feel*, eine einheitliche Navigation sowie gleichartig angeordnete Bedienelemente sicherzustellen.

In den Kommunikationsprofilen für FF [167, 168], PROFIBUS [169], PROFINET [170] und HART [171] werden die Feldbus-spezifischen zusätzlichen Gerätebeschreibungen sowie einige zusätzliche obligatorische (*mandatory*) Parameter für die EDD festgelegt.

**Anwendungen** Neben [1] gibt es inzwischen weitere Ansätze um Mithilfe von FDI komplexe Geräte zu parametrieren. So stellen [172] und [173] eine Lösung für die Parametrierung von modularen Niederspannungsschaltgeräten vor. Im beschriebenen Anwendungsfall ist es notwendig, proprietäre Protokolle zu überbrücken. Dazu werden auch Optionen diskutiert,

die direkten Zugriff vom UIP auf das Gerät ermöglichen, mit dem Nachteil, dass das Informationsmodell nicht verwendet wird.

### 2.1.5.5 Field Device Tool 2

Gleichzeitig mit der gemeinsamen Entwicklung von FDI mit der FDI-Corporation hat sich die *FDT Group* entschlossen, einen Nachfolger für FDT zu entwickeln. Hauptgrund war die Abkündigung der Technologien, auf denen FDT basiert [174, S. 1]. Der Struktur der Spezifikation [175] ist zu entnehmen, dass sie bei der IEC standardisiert werden soll. Field Device Tool Version 2 (FDT2) soll FDI Packages nativ integrieren können [174, S. 4], [176, S. 3]. Deswegen ist eine zweistufige Komponentenarchitektur vorgesehen, die Geräteinformationsmodell und Darstellung an separaten Schnittstellen in die Frameapplication integriert (siehe Abbildung 2.8).

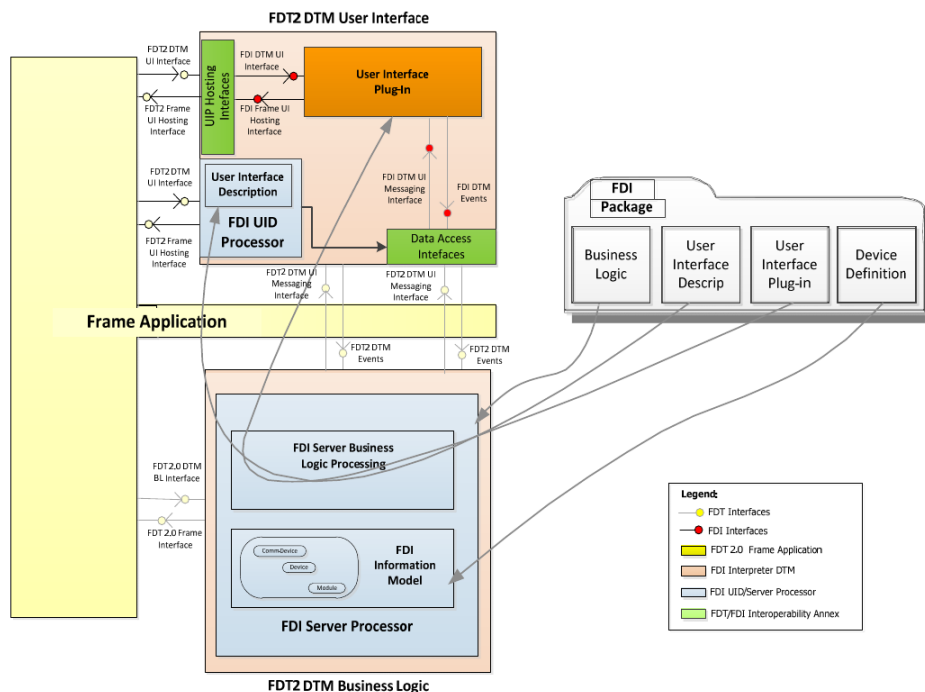


Abbildung 2.8: FDI Package in einem FDT2-Frame (aus [174, S. 4])

## 2.2 Betrieb und Betriebsdaten

Mit Beginn der Inbetriebnahme von Instrumentierung fallen in einer Anlage Betriebsdaten an. Diese werden vornehmlich in Feldgeräten, dem PLS und der Produktionsplanung und -dokumentation generiert.

Prozessdaten besitzen üblicherweise kurze Zykluszeiten und kurze Gültigkeitsdauern. Sie werden nur nach Bedarf und u. U. mit reduzierter Samplingrate archiviert. Für ihren hauptsächlichen Nutzen – Prozessführung, Steuerung und Regelung – sind die Betriebsdaten nur flüchtiger Form notwendig. Produktionsplanungsdaten wie zum Beispiel Batch-

Beschreibungen und Rezepte sind länger gültig und werden üblicherweise vollständig archiviert [14, Abs. 3.2, S. 72ff]. Abschnitt 2.2.1 behandelt Lösungen und deren Schnittstellen, um auf entsprechende Betriebsdaten zugreifen zu können.

Durch ständige Verbesserungen aber auch durch Wartungsarbeiten finden Änderungen an Instrumentierung und deren Einstellungen statt. Entsprechende Arbeiten werden meistens mit PAM-Werkzeugen unterstützt bzw. vorgenommen. Abschnitt 2.2.2 ab Seite 26 behandelt die Möglichkeiten, gerätezentriert auf Betriebsdaten zuzugreifen.

### 2.2.1 Betriebsdaten zu Anlage und Prozess

In Abhängigkeit von der Architektur des Automatisierungs- bzw. Prozessleitsystems werden die Mess- und Stelldaten der Sensor- und Aktuator-Feldgeräte in ein oder mehrere Prozessabbilder abgebildet. Diese können in Prozessnahen Komponenten (PNKn) oder Speicherprogrammierbaren Steuerungen (SPSen) liegen [14, Abs. 4.2.4, S. 218f]. Für die Steuerungsprogramme ist die Übertragung der Daten über verschiedene Kommunikationswege (siehe Abschnitt 2.1.4.1 ab Seite 15 und Abbildung 2.4 auf Seite 16) transparent. Somit sind die Daten im den PNKn uneingeschränkt über ihre Bezeichner verfügbar und Regelkreise können darauf einfach implementiert werden [14, Abs. 3.3, S. 113ff]. Auch PLS-eigene Zusatzsoftware kann an dieser Stelle mit dem entsprechenden Engineeringaufwand und unter dem entsprechenden Ressourcenverbrauch eingebunden werden.

Das im Engineering vorhandene Wissen über Prozess und dessen Randbedingungen geht in die Softwarestruktur ein. Von außerhalb des Systems ist es von dort jedoch nicht mehr effizient abrufbar. Generell ist es unüblich, Komponenten, die keine Steuerungs- oder Sicherheitsaufgaben haben, direkt auf diese Prozessabbilder zugreifen zu lassen. Auch ABKn/Human Machine Interface (HMI)s und Supervisory Control and Data Acquisition (SCADA)-Systeme [14, Abs. 4.2.6, S. 219f] benötigen diesen Zugriff. Sie sind aber als Teil des Steuerungs- und Sicherheitssystems eng gekoppelt angebunden. Dies passiert meist über einen proprietären Systembus [14, Abs. 4.2.5, S. 219]. Anhand der Fließbilder werden Bedienbilder entwickelt. Dies kann auch in einem automatisierten Engineering Prozess erfolgen [23]. Alarme und verwendete Anzeigen werden in der ABK engineeriert und bei Bedarf übertragen und angezeigt. Auch hier ist die Integration der relevanten Prozesswerte Tool-spezifisch und kein generischer Zugriff möglich.

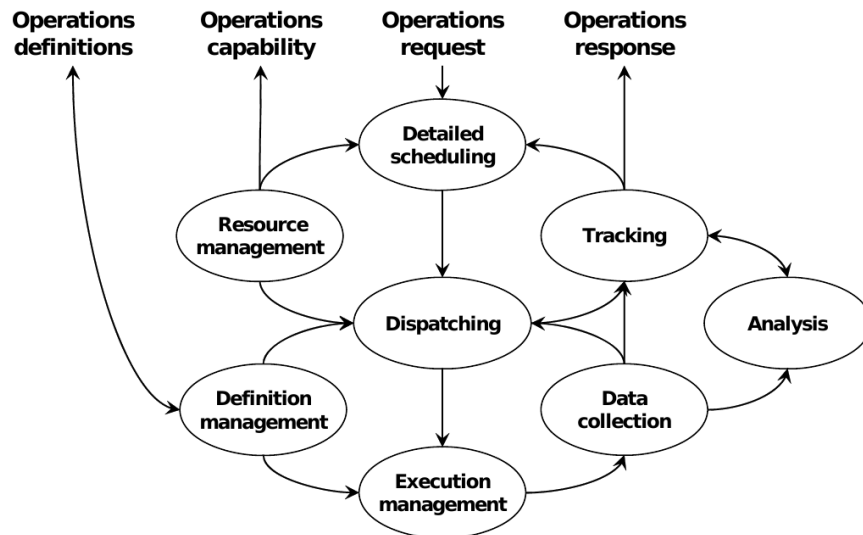
#### 2.2.1.1 Zugriffe aus höheren Ebenen und vertikale Integration

Der Zugriff auf Eigenschaften des Gesamtprozesses einer Anlage außerhalb des Prozessleitsystems ist aus zweierlei Gründen schwierig. Einerseits kann nur über die Feldgeräte zugegriffen werden, die die Daten bereitstellen und zweitens sind die entsprechenden Orts- und Stellenbezeichnungen notwendig, um sie zuzuordnen. Dennoch wird eine solche Lösung unter Umständen bevorzugt, weil sie ohne Rückwirkung auf das PLS integriert werden kann [7, 8]. Der Zugriff auf Parametrier- und Prozessdaten erfolgt zumeist über Zugriffspunkte unterhalb der PNK, z. B. an Feldbussen. Anlagenbezogene Daten müssen dann aus den gerätebezogenen Daten abgeleitet werden [177, 8].

Die ISA S 95 Teil 3 [178] beschreibt ein generisches Aktivitätsmodell, welches die Zusammenhänge von Unternehmenssteuerung und Prozesssteuerung anhand von abstrakten Aktivitäten und deren Abhängigkeiten modelliert (Abbildung 2.9). Dieses Modell lässt sich



auf verschiedene Teilaufgaben spezialisieren, z. B. Wartungs- und Qualitätsmanagement. Die VDI-Richtlinie 5600 beschreibt Systeme (Manufacturing Execution Systems (MES)), die diese Aktivitäten umsetzen. Davon behandelt Blatt 1 [179] Aufgaben und Bedeutung für Unternehmensprozesse, Blatt 2 [180] den Nutzen solcher Systeme und Blatt 3 [181] eine semantisch annotierte Schnittstelle vom Prozess zum MES. Über derartige Schnittstellen sind MES und Enterprise Resource Planning (ERP)-Systeme angebunden.



**Abbildung 2.9:** Generisches Aktivitätsmodell zum Produktionsmanagement (aus der ISA S95 [178, Fig. 6])

Der Zugriff auf die Daten kann mit unterschiedlichen Technologien erfolgen. Zum einen werden Standardtechnologien wie OPC und OPC UA eingesetzt, aber auch verschiedene Webtechnologien sind neben den verbreiteten, proprietären Schnittstellen geeignet. Weitere Beispiele sind in Abschnitt 2.2.2.1 ab Seite 26 und Abschnitt 2.2.2.2 ab Seite 27 beschrieben.

### 2.2.1.2 OLE for Process Control

OPC [182] ist ein Protokoll zum Zugriff auf beliebige Daten in Automatisierungssystemen. Es basiert auf Object Linking and Embedding (OLE) und damit auf COM/DCOM, einer inzwischen nicht mehr unterstützten *Microsoft*-Technologie. OPC kapselt die Daten in Objekte und strukturiert diese als *Items* in *Groups*. OPC-Server können eines von unterschiedlichen *Profilen* unterstützen, z. B. *Data Access* (DA), *Alarms & Events* (A&E), etc.

OPC kann genutzt werden, um auf Prozessdaten zuzugreifen, ohne fest im PLS integriert zu sein z. B. auf lokalen Bedienpanels oder mobilen Geräten. Die über OPC zugreifbaren Objekte müssen explizit eingerichtet werden und bedeuten bei jeder Änderung oder neuen Anforderung Engineeringaufwand.

### 2.2.1.3 OPC Unified Architecture

OPC UA [183], [184] ist der Technologienachfolger von OPC. Die Spezifikation wurde von der *OPC Foundation* erstellt und später als IEC 62541 standardisiert. Es definiert die Übertragungsprotokolle ohne Technologieabhängigkeit. Die Übertragung kann kryptographisch signiert und verschlüsselt werden. Zusätzlich ist ein Authentifizierungsmechanismus implementiert [185]. OPC UA definiert ein Adressraummodell [183, Abs. 2.12, S. 81f], [186] aus Objekten, Variablen, Referenzen und Methoden sowie zugehörigen Typen und Datentypen. Somit ist das Informationsmodell [183, Abs. 2.12, S. 81f], [187] eines Servers semantisch selbst-beschreibend, sofern die Semantik der Typen festgelegt ist. Es erlaubt frei strukturierte Netzwerke von sich untereinander referenzierenden Nodes (full mesh network). Ein Server kann beliebig viele Profile wie *Data Access* [188] oder *Alarms & Events* [189] unterstützen.

Zusätzlich wurden weitere Informationsmodelle entwickelt, zum Beispiel OPC UA-DI [190]/[161] oder *Aggregates* [191]. Die OPC UA-DI-Spezifikation definiert Basisdatentypen und Strukturen für die Geräteintegration bzw. Inbetriebnahme. Es ist in der Lage, den Datenhaushalt von komplexen und modularen intelligenten Feldgeräten zu reflektieren. FDI (siehe Abschnitt 2.1.5.4 ab Seite 20) verwendet dieses Informationsmodell. Die *Aggregates*-Spezifikation beschreibt ein Informationsmodell und Regeln für die Zusammenfassung der Adressraum- und Informationsmodelle unterschiedlicher Server.

### 2.2.2 Betriebsdaten zu Feldgeräten

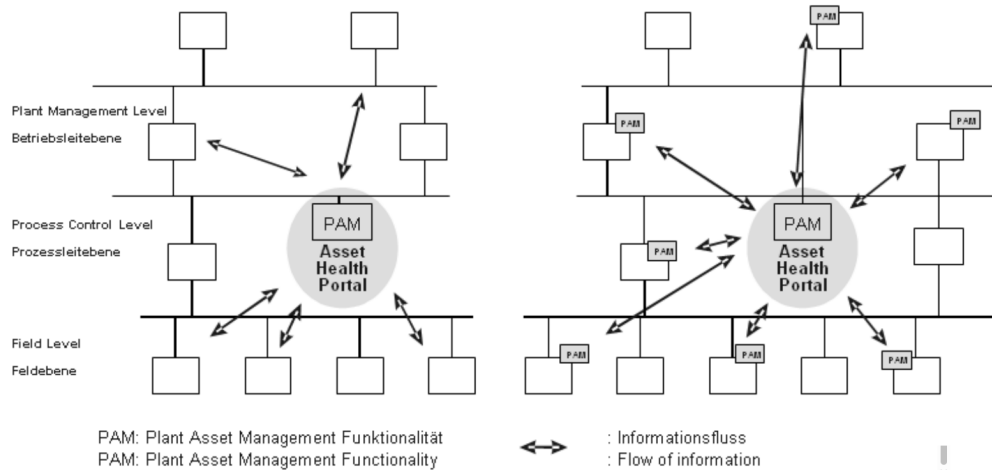
Neben den Prozessdaten, die die Feldgeräte zyklisch mit dem PLS austauschen, sind darin komplexe Konfigurationsdaten enthalten (siehe Abschnitt 2.1.5 ab Seite 17). Diese Daten werden über den Lebenszyklus der Feldgeräte hinweg gepflegt und müssen u. U. betriebsbedingt häufig geändert werden, z. B. in *Batch*-Anlagen. Es wird als immer wichtiger bewertet, solche Änderungen konsistent zu dokumentieren, in bestimmten Anwendungsfällen ist das sogar zwingend vorgeschrieben. Ebenfalls werden Verwaltungsaufgaben (Abschnitt 2.2.2.1) ausgeführt und viele Feldgeräte besitzen auch Selbstüberwachungsfunktionen (Abschnitt 2.2.2.2).

Diese Daten sind weitestgehend hersteller-spezifisch und ihre Semantik wird hinter Schnittstellen oder in verschlüsselten EDDs oder übersetzten DTMs codiert (siehe Abschnitt 2.1.5 ab Seite 17). So können sie nur über die Beschreibungen in den Tools durch einen Bediener interpretiert werden. Es werden wahrscheinlich keine Rohdaten sondern nur Diagnosen, Leistungs- Wartungsdaten zugreifbar sein.

#### 2.2.2.1 Wartung und Verwaltung / Plant Asset Management

Das PAM beschäftigt sich mit der Verwaltung und den nicht direkt mit der Produktion verbundenen Aufgaben zu Gerätschaften aller Art (*Assets* / Produktionsmittel). Die NAMUR, VDI und VDE haben als Ergebnis teils gemeinsamer Arbeiten eine Reihe von Dokumenten zu Zweck, Sinn und Aufgaben des PAM [192, 193, 194] herausgegeben [195]. Unter anderem werden in Abbildung 2.10 eine abstrakte Automatisierungsstruktur mit verschiedenen Systemanbindungsvarianten für PAM-Systeme (zentral / dezentral) illustriert. Die Abbildung demonstriert die Umsetzung der Anforderung, dass PAM-Systeme von den eigentlichen Regelungs-, Prozessführungs- und Sicherheitssystemen

unabhängig sein sollen. Demzufolge arbeiten viele PAM-Systeme nur gerätezentriert, da diese Aufgaben über die Integrationstechnologien (siehe Abschnitt 2.1.5 ab Seite 17) relativ einfach umzusetzen sind. Die ISA 108 beschreibt, wie in [196] und [197] zusammengefasst wird, die *Intelligent Device Management* als den Bereich des PAMs, der sich mit der Verwaltung von Intelligenten Feldgeräten befasst.



**Abbildung 2.10:** Verschiedene PAM-Systemanbindungsvarianten (aus [194, Abb. 4, S. 22])

Geräteübergreifende Aspekte des PAMs werden in Abschnitt 2.3 ab Seite 28 behandelt. In [177] sind verschiedene Kategorien von PAM-Modellen bezüglich ihres Nutzens eingeordnet. Dabei wird der Nutzen um so höher eingeschätzt, je komplexer die Assets und die Modelle sind.

Die VDI/VDE-Richtlinie 2651 Blatt 2 [198] beschreibt Methoden zur Integration von Assetmodellen in den klassischen Engineeringprozess. Dabei es werden PAM-Funktionen mittels einem *Methodenblatt* notiert und mittels der Angaben in einem *Spezifikationsblatt* in die Anlage integriert. Weiterhin werden PAM-Funktionen sowohl für einfache Assets wie Feldgeräte also auch für komplexe Assets wie Teilanlagen beispielhaft beschrieben.

[32] zeigt, wie modellbasiert auf der Grundlage von Merkmalen (siehe Abschnitt 2.1.3 ab Seite 13) einfache Modelle automatisch abgeleitet werden können. In verschiedenen Anwendungen wird der Zugriff auf die Feldgeräte über DTMs [142, 140, 141], EDD-Parser [7, 8] mit entsprechenden Standard-Feldbustreibern oder spezieller Mappingsoftware [199, 200] realisiert. Der Clientzugriff erfolgt über die Oberflächen der DTMs [142, 140, 141] oder Web-Services [7, 8, 199]. Alle Lösungen haben gemein, dass sie mittels azyklischer Kommunikation auf die Feldgeräte zugreifen und nur nach Bedarf kommunizieren. Eine Beeinflussung der Kommunikationsrelationen des PLS sind durch die Feldbusprotokolle ausgeschlossen [9].

### 2.2.2.2 Geräte-Selbstüberwachung

Ein wesentlicher Aspekt der PAM-Aufgaben ist die Zustandsüberwachung von Assets. Unter dem Begriff Condition Monitoring (CM) wird sie auch als eigenständige Disziplin beschrieben. Für Feldgeräte in der Prozessindustrie beschreiben die NE 107 [201] und die VDI/VDE Richtlinie 2650 [202, 203] Anforderungen an die technische Lösung und

die Abbildung von Fehlerzuständen. Die Ergebnisse dieser Zustandsüberwachung werden, sofern sie Prozessalarme sind oder anderweitig prozessrelevant sind, mit den Prozessdaten übertragen. Anderenfalls können feldbuspezifische Diagnosemechanismen verwendet werden. Eine klare Trennung ist dabei in vielen Fällen schwierig.

[204, 205] und [206, 207] zeigen verschiedene Anwendungsbeispiele für die physikalische Ableitung von Kennzahlen für Feldgeräte. Die technische Lösung ist bei ersteren beiden im Gerät verortet und bei zweiten als Zusatzkomponente für das PLS ausgeführt. Bei letzteren Anwendungsbeispielen kann deswegen auf die Verteilung der Auswertefunktionen nicht geschlossen werden, ein gerätezentrierter Zugriff kann trotzdem möglich sein.

### 2.3 Bewertungsmodelle und deren Anwendung

Unter dem Begriff Bewertungsmodell werden im Rahmen dieser Arbeit verschiedene Verfahrens- und Apparatemodelle mit Berechnungsvorschriften zusammengefasst, um aus Betriebsdaten oder Simulationsergebnissen Aussagen über einen Prozess in Anlage bzw. deren Apparaten abzuleiten. Diese können im Betrieb oder offline ausgewertet werden und in unterschiedlichen Lebenszyklusphasen aktiv sein. Die Modelle werden für unterschiedliche Aufgaben erstellt. Die modellierten Sachverhalte wären meist für andere Aufgaben nützlich aber meist sind die Modelle in den bestehenden Werkzeugen technisch nicht für andere Zwecke übertragbar [64].

Das NA 96 [208] nennt die online Auswertung von solchen Modellen *Prozessdiagnose* und beschreibt beispielhaft Modelle, Implementierungsaspekte und Anforderungen an die Integration der bestimmten Kennzahlen. Ebenfalls ist eine Unterstützung der Ursachenbestimmung und eine Ausgabe von Therapieempfehlungen gefordert. [209] beschreibt Anwendungsbeispiele und verschiedene Verfahren zur Auswertung solcher Modelle. Abbildung 2.11 zeigt eine Übersicht über verschiedene Methoden und Assets, auf die sie angewendet werden können.

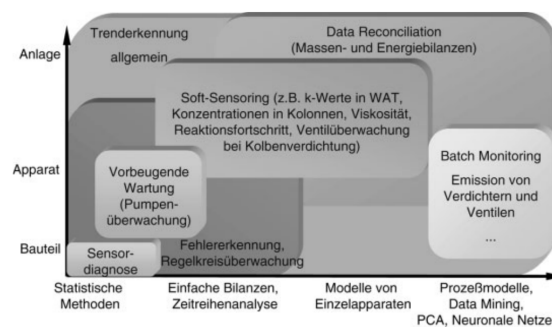


Abbildung 2.11: Methoden des Performance Monitoring (aus [209, Abb. 4, S. 530])

In [210] wird die formale Modellierung der Anlage auf Basis von Strukturinformationen aus dem R&I-Fließbild beschrieben. Auf einer solchen Basis können Modelle generiert werden [33] als auch unbekannte Messgrößen zur Verbesserung der Modellabdeckung aus bekannten Messwerten und Strukturwissen abgeleitet werden [210, 211].

Für den Trend der Modularisierung von Anlagen [25, 212] ergeben sich Möglichkeiten, mit der Modul-Automatisierung auch Prozessmodelle und deren Auswertung in die Module

bzw. *Package Units* zu integrieren. Eine effiziente Modulintegration ergäbe auch eine einfachere Integration des Modells als noch in [177, F. 8 unten] dargelegt.

Abschnitt 2.3.1 beschreibt verschiedene Simulationslösungen zur Untersuchung von Apparaten und Prozessen, in Abschnitt 2.3.2 ab Seite 30 werden Apparatemodelle, die im PLS ausgeführt werden, behandelt. Weitere Integrationslösungen für Apparatemodelle werden in Abschnitt 2.3.3 ab Seite 30 vorgestellt und in Abschnitt 2.3.4 ab Seite 31 werden einige komplexe Ansätze beschrieben.

### 2.3.1 Simulationsmodelle

Prozesssimulationen sind zumeist die numerische Berechnung von Anfangswertproblemen von gewöhnlichen oder partiellen Differenzialgleichungssystemen. Dabei erhält man keine allgemein gültige Aussage sondern nur ein Beispiel für einen konkreten Zeitverlauf. Eine allgemeine Aussage kann nur aus der Struktur des DGL-Systems und einer Reihe von Simulationsläufen mit verschiedenen, gut gewählten Anfangswerten und evtl. Führungsgrößenverläufen.

**Prozesssimulation im Prozeszentwurf** Eine Anwendung der Simulation ist im VT-Engineering angesiedelt. Hier kann auf Basis des Wissens über physikalische und chemische Gesetzmäßigkeiten, Struktur bekannter Verfahrensschritte und Eigenschaften der geplanten Prozesses der Verlauf unter verschiedenen Annahmen bestimmt werden. Hierfür gibt es spezialisierte Softwarewerkzeuge, z. B. *Aspen* [24], *ChemCAD* [213] oder *gPROMS* [214].

**Prozesssimulation in der Inbetriebnahme** Im klassischen Ablauf der Inbetriebnahme [14, Abs. 3.3.5, S. 123ff] können Simulationen eingesetzt werden, um die Prozessmodelle und Reglereinstellungen zu validieren.

[215, Abs. 3.3] gibt einen ausführlichen Überblick über Simulationsmethoden und Simulationsmodelle für den Test der Steuerungssoftware im PLS. Dabei sind insbesondere System- und Hardware-in-the-Loop-Tests interessant, da sie Anlagenmodelle benötigen. [215, Abb. 3-3, S. 29] beschreibt vier hierfür verwendete Klassen von Simulationsmodellen mit ansteigender Komplexität: „E/A-Modelle“ für den Test von Sensoren und Aktuatoren, „Modelle mit Dynamik und adaptivem Parametrierungsgrad“ zum Test von Abläufen, Verriegelungen und Grenzwerten, „Modelle mit Dynamik und hohem Parametrierungsgrad“ für die Anwenderschulung und „Dynamische Modelle mit numerisch korrekter Wiedergabe der physikalischen und chemischen Vorgänge“ für die Verfahrensoptimierung. Die entwickelten Modelle basieren der auf typischen Apparaten, Grundoperationen oder Strukturen, sog. „Funktionsmustern“ [215, Kap. 6, S. 81ff]

**Verwendung von Simulationen im Betrieb** Zur Optimierung von Prozessen können, wie bei der Inbetriebnahme, zur Überprüfung von Änderungen und Regleranpassungen vor der Übernahme ins laufende System die Auswirkung der Änderung simuliert werden. Ebenfalls können in Bewertungssystemen fehlende Größen/Ersatzwerte oder auf theoretischer Basis abgeleitete Vergleichswerte nicht nur durch statische Gleichungen sondern auch durch eine Echtzeitsimulation bestimmt werden [216, 211].

### 2.3.2 Apparatemodelle im Prozessleitsystem

Eine Variante, Apparatemodelle zu integrieren, ist die Implementierung als Funktionsblock in einem PLS. Dies vereinfacht die Erfüllung der Anforderung des NA 96 [208] nach einer kontinuierlichen Auswertung in Echtzeit. Andererseits wird dadurch die mögliche Komplexität von Modellen beschränkt.

Ins PLS integrierbare Modelle sind schon länger bekannt [217, 204, 218, 219]. In den großen PLS bzw. deren PAM-Komponenten z. B. *SIEMENS' PCS 7* bzw. [220, 221, 222, 206, 207], *ABBs 800xA Asset Optimization* [223, 224, 225, 226] und *Yokogawas PRM* [141] sind überall derartige Modelle verfügbar. Gemeinsam ist, dass sie nur für sehr weit verbreitete Asset-Typen wie Wärmeübertrager, Pumpen, Kompressoren, Ventile oder allgemeine Trendanalysen Bausteine vorhanden sind. Dies lässt sich auf die große Anzahl entsprechender Assets in Anlagen und die strukturelle Einfachheit der überwachten Funktion zurückführen.

Eine Integration von PLS-unabhängigen Modellen über IEC 61131-3 [227] Funktionsblöcke über das entsprechende Austauschformat *PLCopenXML* [228, 229] scheint auch möglich. Deren Unterstützung in verbreiteten Tools war jedoch lange problematisch [230], könnte jedoch durch die Verbreitung von AML [45] einfacher werden. Die Berechnungsmodelle würden dabei allerdings offengelegt.

### 2.3.3 Apparatemodelle außerhalb des Prozessleitsystems

Außerhalb des PLS gibt es eine deutlich heterogenere Menge von Konzepten und Lösungen für Apparatemodelle. Als Alternative zur Implementierung im PLS schlägt das NA 96 [208] zwei Varianten zur Implementierung in einem Process Information and Management System (PIMS) vor. Das VDMA-Einheitsblatt 24582 Condition Monitoring [231] bezieht sich auf die Fertigungstechnik und den Anlagen- und Maschinenbau, beschreibt aber eine durchaus in der Prozessautomatisierung anwendbares Konzept zu Verteilung von Bewertungsmodellen. So können in einem verteilten PAM-System übergreifende Kennzahlen ebenfalls dezentral aggregiert werden. Es wird darin auch eine deutliche Trennung von funktionaler (und CM-) Sicht und Automatisierungsgerätehierarchie vorgenommen. Das Einheitsblatt baut auf den Festlegungen in der ISO 13372 [232] auf und lehnt sich an den Vorgaben zu Architektur, Methoden und Strukturierung von CM-System aus der ISO 13374 [233, 234, 235]. Letztere trennen die Bearbeitungskette zur Auswertung von Maschinenzuständen in sechs Schritte: Datenerfassung, Datenmanipulation, Zustandserkennung, Bewertung des Gesundheitszustands, Erstellung einer Prognose und Generierung von Handlungsempfehlungen. Für die einzelnen Schritte werden abstrakte Bearbeitungsblöcke [234] und abstrakte Kommunikationsprimitive zwischen Komponenten untereinander und mit externen Clients definiert [235]. Die ISO 17359 [236] beschreibt das Vorgehen bei der Implementierung solcher CM-Modelle für Maschinen.

In der VDI/VDE-Richtlinie 2651 Blatt 2 [198] werden Formate für Spezifikation und Funktionsbeschreibung für Modelle für Teilanlagen definiert. Die beschriebene Methode soll sich in den klassischen und die zukünftig elektronisch vernetzten Engineeringablauf integrieren. Die konkrete Umsetzung für Modelle und Ausführungsplattformen ist offen gelassen.

In [237] wird eine hierarchische, unterschiedlich verteilbare Struktur von Geräten beschrieben, die übergreifende Funktionen in höheren Ebenen aggregieren kann. [238] beschreibt geräteübergreifende Bedienfunktionen für einfache Subsysteme. Diese sollen über generische DTMs miteinander verknüpft werden und einfache Diagnosemodelle integrieren. Nach [239] können solche Strukturen mittels einer Suche nach bekannten Mustern automatisch aus einem Engineeringmodell extrahiert werden. [32] stellt einen merkmalsbasierten Ansatz für diese Suche vor. Somit können Anwendungsmöglichkeiten für generische Modelle regelbasiert gefunden werden. Eine Lücke ist jeweils noch die Integration der Datenpunkte.

In [240] werden solche Modelle nach ihrer Methodik klassifiziert und einige Anwendungsbeispiele in verschiedenen Domänen dargelegt. Jedoch wird nicht auf Ausführungsplattformen eingegangen. [205] beschreibt ebenfalls einige bekannte Anwendungsfälle und geht auf Werkzeuge und die Integration in bestehende Systeme ein. [241] beschreibt eine CM-Anwendung mit einem sehr komplexen Modell und einem Dienst- und Ontologie-basierten Auswertesystem.

In [8], [242] werden Lösungen vorgestellt, in denen die bestimmten Kennwerte mittels Web-Services angeboten werden. Sie ermöglichen eine Aggregation auf unterschiedlichen Systemebenen. [10] beschreibt dabei die Möglichkeit der Zuordnung von Rollen- und Systemklassen für übergreifende Funktionalitäten.

Die in [243], [244, S. 78–79] beschriebene Auswertesoftware ist eine Zusatzsoftware, die per OPC auf die Prozessdaten zugreift und über Bausteine auch in das HMI des PLS integrierbar ist. Sowohl die Bereitstellung der *OPC-Items* als auch die Integration der Ampel-Symbole in die Benutzungsoberflächen bedeuten zusätzlichen Engineeringaufwand.

[245, 246] beschreiben unterschiedliche Methoden zur (semi-)automatischen Generierung von Simulationsmodellen für Steuerungstests und die virtuelle Inbetriebnahme. Dabei werden Festlegungen aus den Engineeringdaten zur Parametrierung der Modelle verwendet.

### 2.3.4 Weitere Ansätze

Neben den Modellen, die konkreten Apparaten, Anlagenteilen oder Anlagenstrukturmustern zugeordnet werden können, gibt es auch weitere Anwendungsfälle und Methodiken, die ein breiteres Anwendungsfeld besitzen. Vor allem sind verschiedene Lösungen in der Automatisierungstechnik, welche für die Fertigungstechnik entwickelt wurden, auch in der Prozessindustrie anwendbar, wenn auch nicht in allen Bereichen. Es gibt deswegen auch Aspekte, die sich mit verschiedenen Modellen oder Methoden aus den vorherigen Abschnitten überdecken.

#### 2.3.4.1 Condition Monitoring

Condition Monitoring (CM) ist ein in der Literatur unterschiedlich verwendeter Begriff. Er wird sowohl für die Diagnose von Geräten als auch für die Überwachung von komplexen Anlagenteilen verwendet. Teils bezieht man sich ausschließlich auf die Generierung von Zustandsinformationen, teils auf deren Bewertung und Ursachenforschung und an anderer Stelle weitestgehend nur auf Datenakquise, -aggregation und -bereitstellung [69].

Die ISO 13372 [232] und die ISO 17359-1 [247] sind eine anerkannte Referenz für die relevanten Begriffe. Sie sind jedoch mehr von den Bereichen der Fertigungstechnik und

des Maschinenbaus geprägt. Dies ist ebenfalls beim VDMA-Einheitsblatt 24582 [231] und den ISO-Normen 13374 [233, 234, 235] und 17359 [236] der Fall.

Das NA 96 [208] beschreibt ähnliche Überwachungsaufgaben unter dem Namen Prozessdiagnose. Die Informationen, die dabei zu Apparaten und Geräten gewonnen werden, sollen dem PAM zur Verfügung gestellt werden. Das PAM seinerseits führt die Zustandsüberwachung als eine seiner Aufgaben [194, 202] auf (vgl. Abschnitt 2.2.2.1 ab Seite 26) und Abschnitt 2.2.2.2 ab Seite 27).

### 2.3.4.2 Predictive Maintenance

Ein nächster Schritt nach der Zustandsüberwachung ist die zuverlässige Vorhersage von Funktionsreserve und Restlaufzeit. Gemeinsam mit der Wartungsplanung als Nutzer solcher Vorhersagen wird dieses Vorgehen als *Predictive Maintenance* oder *zustandsorientierte Instandhaltung* [248, 249] bezeichnet. Hierbei wird auf Basis von Geräte- und Prozesswissen mithilfe von Prozess- und Gerätedaten die voraussichtliche Qualität der Produkte und Standzeit der Anlagenteile berechnet [250, 251].

### 2.3.4.3 Digitale Fabrik

Die zunehmende Digitalisierung der Produktion hat zu der Anforderung geführt, alle Prozessschritte verfolgen, analysieren und übergreifend verbessern zu können. Der dabei geprägte Begriff *Digitale Fabrik* wird in der VDI-Richtlinie 4499 [252, 253] folgendermaßen definiert:

„Die Digitale Fabrik ist der Oberbegriff für ein umfassendes Netzwerk von digitalen Modellen, Methoden und Werkzeugen – u. a. der Simulation und der dreidimensionalen Visualisierung –, die durch ein durchgängiges Datenmanagement integriert werden.

Ihr Ziel ist die ganzheitliche Planung, Evaluierung und laufende Verbesserung aller wesentlichen Strukturen, Prozesse und Ressourcen der realen Fabrik in Verbindung mit dem Produkt.“

Die IEC 62832 [254] beschreibt Modelle zur Integration und Kombination von verschiedenen Informationen aus verschiedenen Lebenszyklusphasen in ein sogenanntes *Digital Factory Repository*. Alle digital abbildbaren Aspekte jedes Assets werden semantisch in das Modell der *Digitalen Fabrik* gemappt. Dadurch kann in jedem Kontext auf alle passenden Dokumente und Daten zugegriffen werden, in dem sie möglicherweise benötigt werden.

Auch sind in verschiedenen Roadmaps und Positionspapieren [255, 106, 108, 107] Anwendungsszenarien für die Nutzung der digitalisierten Produktion dargelegt. Mit Schlagwörtern wie *Cyber Physical Systems* und *Industrie 4.0* wird die voll vernetzte, sich selbst konfigurierende und selbst organisierende Fabrikhalle beschrieben, die individuelle Produkte mit der Losgröße 1 fertigt. Auch soll die gesamte Lieferkette automatisiert werden. Dabei wird festgestellt, dass diese Zukunftsvision nur auf Basis von standardisierten Funktionalitäten und Schnittstellen mit eindeutiger Semantik möglich sind. Hierzu wurde Forschungsbedarf identifiziert und kategorisiert.

Die NE 139 [256] beschreibt die Komplexität von einfachen Schnittstellen unter den spezifischen Anforderungen der Prozessindustrie und zeigt damit auch, wie weit eine



maschinelle Umsetzung auf Basis heutiger Technik und mit den heutigen Werkzeugen entfernt ist. Daten und Zugangsmöglichkeiten sind aber prinzipiell vorhanden.



## 3 Konzept zur Modellintegration

Im Umfeld der Prozessautomatisierung soll auf Basis der in Kapitel 2 ab Seite 7 vorgestellten Technologien und Arbeiten ein Konzept zur Apparatemodell-Integration ausgearbeitet werden. Dazu soll die daraus resultierende Kernfrage beantwortet werden: *Wie kann eine Infrastruktur auf Basis von Standardtechnologien aussehen, sodass alle Schritte des Engineerings für Apparatemodelle weitestgehend automatisiert werden können?* Dabei soll explizit nicht auf Modul- und *Package-Unit*-Integration wie z. B. in [212] oder modulare, herstellerseitig zusammengestellte komplexe Assets wie z. B. in [172] beschrieben eingegangen werden. In beiden Fällen sind mehr a-priori Informationen vorhanden die einen Lösungsansatz vereinfachen.

Das Grundkonzept wurde bereits in [1] veröffentlicht und einige Werkzeugaspekte wurden in [2] und [3] weiter vertieft. Erste Ideen zur abstrakten übergreifenden Verwendung von Messstellen wurden in [8] aus einer anderen Arbeit heraus aufgegriffen. Zur technologischen Basis wurde in [3] und [5] beigetragen.

In diesem Kapitel sollen Designentscheidungen zu Werkzeugkette und Technologien diskutiert und begründet werden. Dabei sollen Vorteile wie auch vorhersehbare Schwierigkeiten bewertet werden.

### 3.1 Abstraktion der Domäne

Um die Vorgänge bei der Umsetzung des Konzepts automatisieren zu können, wird zunächst der relevante Teil des Domänenmodells notiert. Hierbei sind verschiedene Partialmodelle mit unterschiedlichen Modellgegenständen involviert. Diese können physisch, als abgelegte oder dynamische Daten oder als abstraktes Modell ausgeprägt sein. Aus diesem Grund sind die Modelle ohne Details und Attribute ausgeführt und als Vertreter für ihre sehr komplexen hinterlegten Modellgegenstände zu verstehen. Allen Beziehungen zwischen den Partialmodellen werden Datenbestände oder Technologien zugeordnet, mit denen diese verknüpft werden können. Die Betrachtungen ähneln den Merkmalsbeschreibungen aus [33] und verwenden Begriffe aus der Richtlinie zu formalen Prozessbeschreibungen [15].

#### 3.1.1 Verfahrensbilanzraum

Anhand von Eigenschaften und Größen in einem Bilanzraum lassen sich Aussagen über die Qualität des Prozesses [208] und den Zustand der technischen Ressourcen [194] ableiten. Jeder Bilanzraum hat ein- und ausgehende Edukt- und Produktströme sowie Energieströme. Deren Eigenschaften sind Prozessgrößen, die sich direkt oder indirekt messen lassen. Sowohl mechanische Anlagenteile als auch technische Einrichtungen zur Erfassung und Beeinflussung werden als technische Ressourcen bezeichnet. Abbildung 3.1 auf Seite 36 zeigt das Mengengerüst mit UML-Klassennotation. Ein Bilanzraum ist nach

gewünschtem Detaillierungsgrad gemäß [15] komponier- und dekomponierbar. Dies passiert gleichzeitig mit den verbundenen technischen Ressourcen.

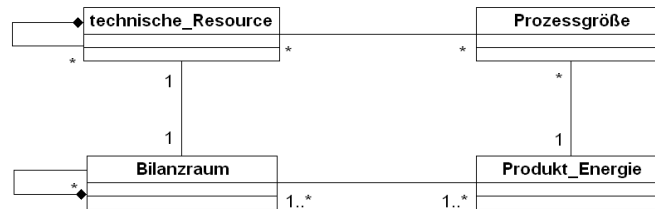


Abbildung 3.1: Bilanzraummodell

Realisiert ist dieses Partialmodell durch die Daten aus dem verfahrenstechnischen Engineering, vor allem den R&I-Fließbildern. Diese können in proprietären Formen oder als (standardisierte) Exporte vorliegen (siehe Abschnitt 2.1.1 ab Seite 8 und Abschnitt 2.1.2 ab Seite 9). Die technischen Ressourcen sind zu diesem Zeitpunkt lediglich Anforderungen an den Anlagenbau und die Automatisierung. Deren konkrete technische Umsetzung wird erst später festgelegt.

#### 3.1.2 Geräteanforderungen

In der IEC 62424 [17] sind verschiedene sog. Process Control Engineering (PCE)-Kategorien definiert, die mit einem Kennbuchstaben die üblichen Prozessgrößen kategorisieren. Diese sind jedoch teilweise mehrfach belegt (z. B. Analyse (A) mit pH-Wert, Konzentration, etc.). Die Aufnahme einer solchen Prozessgröße oder die Einflussnahme auf eine Prozessgröße sind abstrakte Grundaufgaben, die durch eine standardisierte Rolle repräsentiert werden. Jede dieser Rollen garantiert den Zugriff auf bestimmte Messwerte oder Stellfunktionen sowie die benötigten dazu gehörigen Zusatzfunktionen (Parametrierung von Alarmen, etc.). Diese sind im Modell (Abbildung 3.2) unter Rollenparameter zusammengefasst.



Abbildung 3.2: Geräteanforderungsmodell

#### 3.1.3 Konkrete Geräte

Ein Mess- oder Stellgerät (Abbildung 3.3) mit einer spezifischen Aufgabe besitzt für dessen Kernfunktion eine Schnittstelle zur Übertragung des Mess- oder Stellwertes. Zusätzlich können bei modernen Geräten viele verschiedene Parameter eingestellt und Zusatzfunktionen bedient werden. Diese werden zur Bedienung und Parametrierung mittels generischer Werkzeuge in Gerätebeschreibungen abgelegt (siehe Abschnitt 2.1.5 ab Seite 17).



Abbildung 3.3: Gerätemodell

### 3.1.4 Apparatemodell

In jedem Bilanzraum wird ein zu einem bestimmten Detaillierungsgrad herunter gebrochener Verfahrensschritt bearbeitet. Oft wiederkehrende, bekannte Verfahrensschritte sind Grundoperationen. Dafür werden im Detail Engineering Teilanlagen, Anlagenteile und schließlich Apparate vorgesehen. Je nach dem Zweck eines Bewertungsmodells ist das Ziel der Betrachtung entweder der Verfahrensschritt (Prozessdiagnose) oder der Apparat (PAM). Diese wurden im Partialmodell als *Betrachtungsgegenstand* abstrahiert.

Ein Bewertungsmodell formalisiert Expertenwissen über einen Betrachtungsgegenstand. Es benötigt Informationen über Prozessgrößen innerhalb des Bilanzraums des Apparates oder Verfahrensschritts. Ein generisches Bewertungsmodell für einen konkreten Betrachtungsgegenstand fordert also bei seiner Instanziierung Rollen, um auf die für seine Berechnungen benötigten Prozessgrößen zugreifen zu können (Abbildung 3.4).

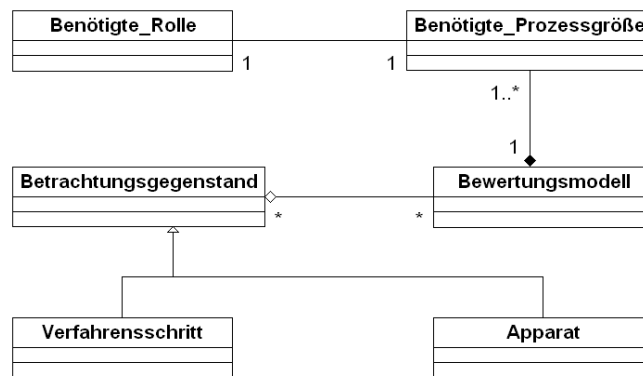


Abbildung 3.4: Prozessschritt- oder Apparatemodell

### 3.1.5 Gesamtmodell

Um ein Bewertungsmodell integrieren zu können, müssen die Partialmodelle mit den im Betrieb zu Verfügung stehenden Datenbeständen miteinander verbunden werden. Dies wird in den folgenden drei Stufen beschrieben. Die Einbindung der Ergebnisse von Bewertungsmodellen zur Weiterverwendung wird hierbei vorerst nicht betrachtet, da sie von der Implementierung abhängig ist.

#### Vorhandene Verbindungen

Die in Abbildung 3.5 auf Seite 38 gezeigten Verbindungen (*A*, *B*, *C*, *D*) zwischen den Partialmodellen sind bereits vorhanden bzw. werden durch heute übliche Engineeringprozesse hergestellt. Teilweise sind diese nicht zwingend für das Engineering der Anlage erforderlich oder werden mit implizitem Wissen beim Übertragen von Daten angewendet. Die sind deshalb unter Umständen nicht formal oder explizit notiert.

- A** Die technischen Ressourcen in den Bilanzräumen werden durch MSR-Anforderungen ergänzt. Die festgelegte Rolle legt den Typ dieser Anforderung fest. Es wird eine Messstellenbezeichnung vergeben.

- B** Im Detail Engineering der Automatisierung werden die konkreten Umsetzungen der MSR-Anforderungen festgelegt. Die Messstellenbezeichnung der Anforderung wird für das Gerät übernommen und für die spätere Parametrierung vorgesehen.
- C** Die Verbindung zwischen einem Rollenparameter einer konkreten Rolle und der Prozessgröße der dieser konkreten Rolle zugeordneten technischen Ressource ergibt sich aus Verbindung *A* implizit. Dieses Wissen ist zumeist in den Engineeringwerkzeugen vorhanden und hart codiert.
- D** Die Verbindung zwischen einen konkreten Rollenparameter und dessen Umsetzung in einem konkreten Gerät ist in Applikationsprofilen (z. B. das PROFIBUS-PA-Profil [116]) festgelegt. Der tatsächliche Zugriff wird erst mithilfe der Gerätebeschreibungen für das Gerät (z. B. EDDL [124]) möglich.

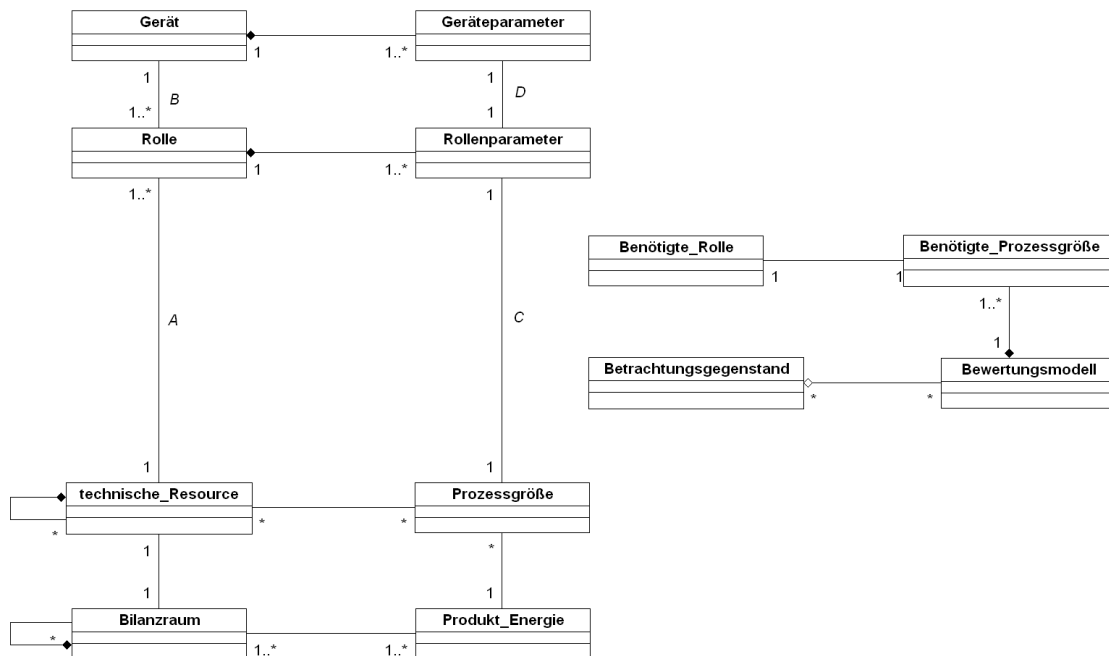


Abbildung 3.5: Gesamtmodell, Verknüpfungsschritt 1

### Auffinden der Bewertungsmodellinstanz

Bei der Instanziierung des Bewertungsmodells für eine Apparateinstanz müssen zunächst der Modellgegenstand und dessen Anforderungen im Anlagenmodell gefunden werden. Hierdurch entstehen weitere Verbindungen (*E*, *F*, *G*) zwischen den Partialmodellen (siehe Abbildung 3.6).

- E** Zuerst muss der Bilanzraum gefunden werden, auf den das Modell anwendbar ist. Für Apparate ist dies die Zuordnung vom angegebenen Apparatetyp im Modell und dem Apparatetyp in den Engineeringdaten. Bei Prozessmodellen ist wahrscheinlich eine manuelle Zuweisung notwendig.

**F** Dann müssen die Anforderungen mit den technischen Ressourcen abgeglichen werden. Die benötigten Rollen sind nicht nur eine einfache Typ-Typ Zuordnung, wobei diese eine Vorauswahl ermöglicht [10]. Eine genauere Einschränkung lässt sich über eine Struktursuche realisieren ([239, 257, 211]). Bei Apparatemodellen kann über die abgehenden Prozessanschlüsse (modell- und apparateabhängig) zwischen zwei technischen Ressourcen mit der gleichen Rolle auch dasjenige mit dem korrekten Zweck (z. B. Messung der Temperatur des Zulaufs, nicht des Ablaufs) bestimmt werden.

**G** Aus (F) ergibt sich implizit die Zuordnung der Prozessgröße im Modell.

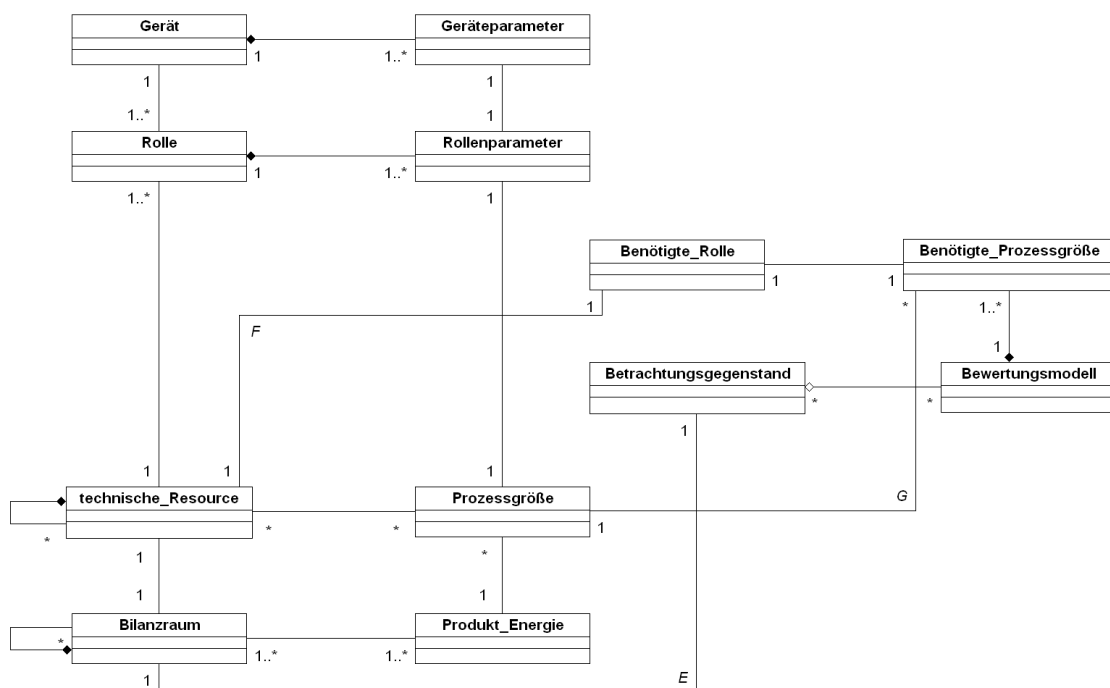


Abbildung 3.6: Gesamtmodell, Verknüpfungsschritt 2

### Einbindung der Bewertungsmodellinstanz

Zur Einbindung der Apparatemodelle müssen weitere Verbindungen (*H*, *I*) hergestellt werden, die indirekt bereits durch vorhandenen Verknüpfungen bestehen. Sie müssen jedoch dauerhaft und direkt mittels eines Zugriffs über Kommunikationstechnologien verfügbar sein (siehe Abbildung 3.7 auf Seite 40).

**H** Die Zuordnung des tatsächlichen Gerätes kann über die Verbindungen *A* und *B* nachvollzogen werden. Da diese Verbindung in der Betriebsphase nicht mehr explizit nachverfolgbar ist, kann man sich mit dem Matching der MSR-Bezeichnung bzw. PLT-Stellenbezeichnung behelfen.

- I Der Zugriffsweg ergibt sich aus den in den Partialmodellen vorhandenen Verbindungen und mit Hilfe der darin verwendeten Technologien implizit. Die konkreten Beschreibungsdokumente müssen gefunden und nach der gesuchten Information durchsucht werden. Wissen zu Applikationsprofilen macht diese Aufgabe automatisierbar.

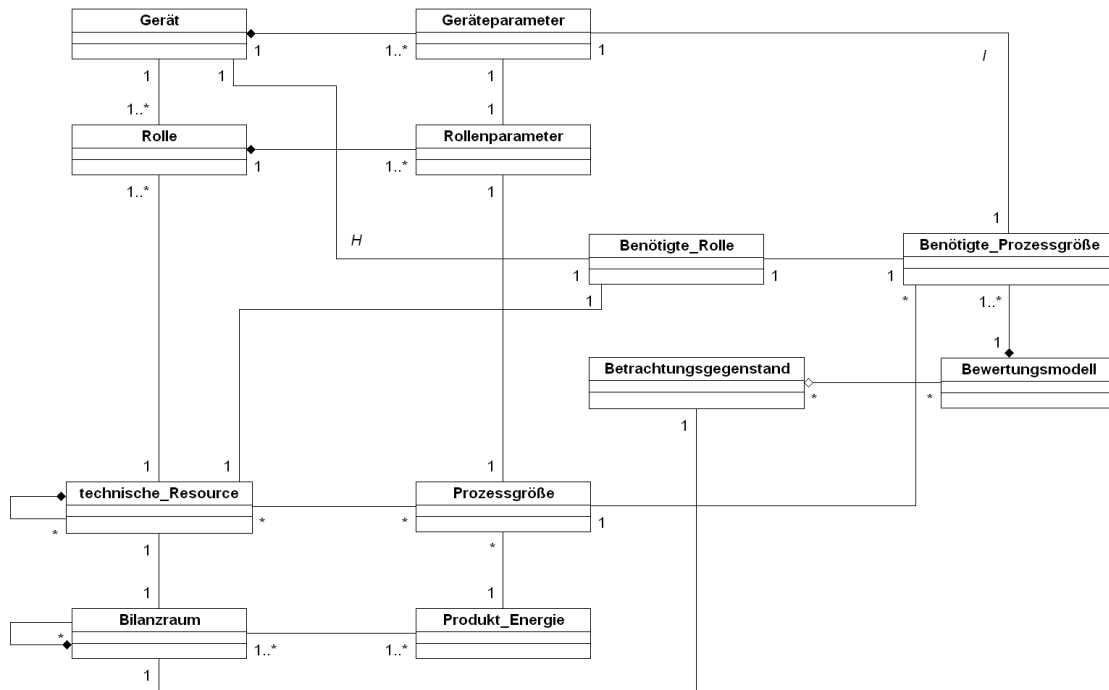


Abbildung 3.7: Gesamtmodell, Verknüpfungsschritt 3

## 3.2 Beteiligte Akteure

Die im zuvor beschriebenen Gesamtmodell benötigten Voraussetzungen und Ergebnisse werden von verschiedenen Akteuren geschaffen. Dies sind zum einen Aufgaben, die heute schon von diesen Akteuren gefordert werden, und zum anderen neue Aufgaben, die für die Umsetzung des Konzepts zur Modellintegration zusätzlich notwendig werden. Teilweise ist auch eine höhere Datenqualität notwendig, da automatisierte Algorithmen meist formal korrekte Daten benötigen (siehe [38, S. 12], [258, F. 15ff.]

Für die Integration ist eine offene und standardisierte Infrastruktur notwendig. Diese beinhaltet Werkzeuge für die Herstellung und die Integration der Modelle sowie Software für den laufenden Betrieb.

### Feldgerätehersteller

Hersteller von Feldgeräten für verfahrenstechnische Anlagen sehen sich hohen Anforderungen an Funktionsumfang und verschiedenen Integrationstechniken für Geräte gegenüber. Die zum Teil sehr komplexen Geräte müssen mit Gerätebeschreibungen und Bediensoftware für verschiedene Kommunikationstechnologien (siehe Abschnitt 2.1.4.1 ab Seite 15)



und dabei unter Umständen noch mehrere Integrationstechnologien (siehe Abschnitt 2.1.5 ab Seite 17) bereitstellen und warten. Gleichzeitig ist es großen Anwendern wichtig, dass die Feldgeräte Applikationsprofilen gehorchen. Sie können damit aus einem PLS typ- und herstellerunabhängig bedient werden, jedoch nur mittels eines gemeinsamen und damit geringeren Funktionsumfangs. Insbesondere diese Kompatibilität zu Applikationsprofilen ist für die automatisierte Integration von Bewertungsmodellen essentiell.

### **Apparatehersteller**

Hersteller von Apparaten stellen diese meist nur auf Anwenderaufträge her. Es gibt einerseits Serien für Tanks, Wärmeübertrager oder ähnlich üblichen Equipment und andererseits Einzelstücke. In letzterem Falle hat der Anwender auch das Know-How zum Prozess und zu möglichen Bewertungsmodellen. Bei Komponenten mit höherer Stückzahl und ohne dedizierter Automatisierung ist die Beschreibung von typspezifischen Modellen am sinnvollsten. Für die Beschreibung der Apparate ist es wichtig, dass die Anschlüsse unterscheidbar sind und auch deren Zweck annotiert wird. Dadurch ist später eine Untersuchung der Flussweg zum Auffinden von Messstellen automatisierbar. Ein entsprechendes Werkzeug zur Erstellung einer solchen Beschreibung ist für den Apparatehersteller vorzusehen.

### **Apparatemodellhersteller**

Der Hersteller eines Bewertungsmodells ist nicht notwendigerweise der Hersteller des Apparates. Er benötigt das entsprechende Prozesswissen und eine formale Beschreibung der Apparateschnittstellen. Das Modell muss mit allen zur Integration notwendigen Dateien und Dokumenten ausgeliefert werden. Sowohl zur Notation als auch zum Export ist Werkzeugunterstützung erforderlich.

### **Anlagenplaner**

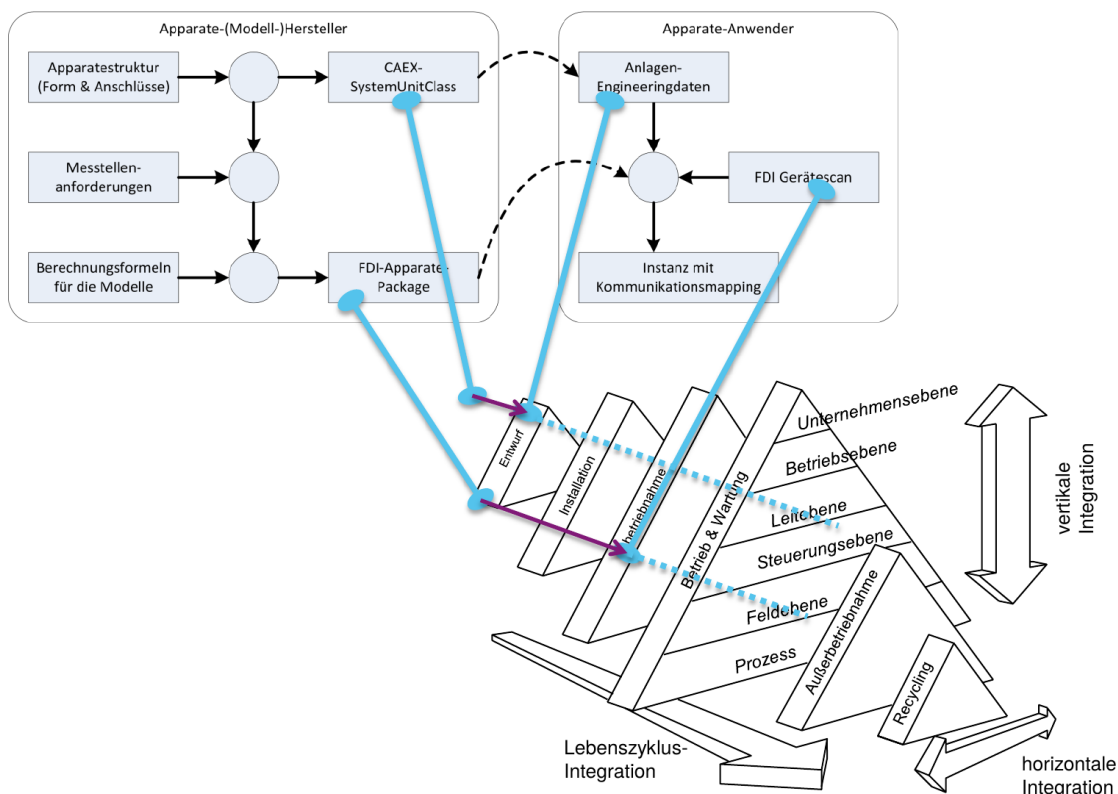
Anlagenplanung ist ein arbeitsteiliger Vorgang, in dem verschiedene Schritte durch verschiedene Gewerke und Bearbeiter unter Zeitdruck ausgeführt werden. Aus diesem Grund ist die Konsistenz, Vollständigkeit und die formale Korrektheit der Daten immer ein Problem. Je besser die Qualität der Planung, desto besser lassen sich die Ergebnisse über ihren angedachten Zweck hinaus nutzen. Auch das Nachführen der Engineeringdaten bei Änderungen an der Anlage ist ein Qualitätsmerkmal. Anlagenstruktur- und Instrumentierungspläne, wie z. B. R&I-Fließbilder, sollten in einem standardisierten Format vorliegen oder exportierbar sein, um sie automatisiert auslesen zu können.

### **Anlagenbetreiber**

Durch die Berechnung von Bewertungsmodellen soll der Anlagenbetreiber beim Fahren, der Optimierung oder der Wartung der Anlage unterstützt werden. Dafür muss er für die Modellintegration notwendigen Dokumente verfügbar haben und die Infrastruktur für den Betrieb der Modelle vorhalten. Die Integration der Modelle und die Darstellung oder Weiterverwertung der Ergebnisse der Berechnungen ist Voraussetzung für eine nutzbringende Anwendung.

### 3.3 Werkzeugkette

Unter Berücksichtigung des durch die Akteure üblicherweise vorgehaltenen Know-Hows und der zeitlichen Abhängigkeiten im Anlagen- und Apparate-Lebenszyklus wurde die Werkzeugkette in Abbildung 3.8 auf Seite 42 entworfen und in [2] vorgestellt. Sie beschreibt die Tätigkeiten bestimmter Akteure und Integrationspunkte im Lebenszyklus eines Bewertungsmodells. Dieser Lebenszyklus ist in den Lebenszyklus der Anlage, in der er instanziiert werden soll, eingeordnet. Mittels verschiedener Datenbestände und standardisierter Modelle sollen alle Schritte der Werkzeugkette nach Möglichkeit automatisiert werden, um eine einfache Erstellung und Integration der Modelle zu erreichen.



**Abbildung 3.8:** Werkzeugkette zur Integration von Apparatemodellen (aus [2]) mit Einordnung in den Anlagenlebenszyklus (hellblau – Einordnung der Bereitstellung in Lebenszyklusphasen, violett – Abhängigkeit, hellblau gestrichelt – Möglichkeiten der späteren Einführung)

Bei oft verwendeten Apparaturen wird üblicherweise im Basic Engineering der Anlage mit der Rolle des Apparates geplant. Diese soll vom Modellhersteller auch angegeben werden. Sobald die Struktur und die Schnittstellen des Apparates festgelegt worden sind, kann er im Detail Engineering eingeplant werden. Um später das Auffinden der Instanzen zu vereinfachen, kann dem Planer ein Planungsdatenartefakt für den Apparatetyp zur Verfügung gestellt werden. Es kann aus Strukturinformationen automatisch generiert werden. Dieses Artefakt sollte in einem Exportformat vorliegen, dessen Semantik definiert bzw. standardisiert ist und somit verwendet werden kann, ohne Mehrdeutigkeiten zu generieren.

Im besten Fall ist dieses Artefakt zu Beginn des Detail Engineering verfügbar. Wenn es später verfügbar wird, ist es auch möglich, die Instanzen über die Rollenbeschreibung zu finden. Die Ergebnisse einer Suche wären jedoch ungenauer im Gegensatz zu der Situation, wenn das Artefakt in den Engineeringdaten hinterlegt ist. Der violette Pfeil in Abbildung 3.8 zeigt den Zeitbereich im Anlagenlebenszyklus, in dem die Bereitstellung das einfachste Engineering verspricht. Die gepunktete Linie deutet an, dass die Bereitstellung auch später erfolgen kann.

Nach der Definition der Messstellenanforderungen kann der Modellhersteller sein Modell mittels eines Werkzeuges beschreiben. Dieses Modell kann dann über ein weiteres Werkzeug als FDI Apparatepackage exportiert werden. Dieses Apparatemodell kann ab der Inbetriebnahme der Feldgeräte in der Anlage eingesetzt werden. Die Instanziierung ist ab diesem Zeitpunkt immer mit dem gleichen Aufwand möglich.

Die Instanziierung wird beim Anlagenbetreiber durchgeführt. Voraussetzung ist neben dem Zugriff auf die Feldgeräte eine Softwarekomponente zur Integration komplexer Apparatepakete. Bei der Instanziierung werden über einen Gerätescan alle verfügbaren Feldgeräte identifiziert und deren Messstellenbeschreibung (*tag description*) ausgelesen. Danach werden sie mit den Messstellenanforderungen aus den Engineeringdaten abgeglichen. Das Setzen der Messstellenbeschreibung während der Geräteinbetriebnahme wurde nicht immer konsequent gehandhabt. Durch die Möglichkeiten der elektronischen Abwicklung der Bestellung individuell vorkonfigurierter Feldgeräte mittels eCl@ss ist diese Schwäche als überwunden zu betrachten.

Über Engineering-Artefakt oder Rollenbeschreibung wird in den Engineeringdaten nach möglichen Modellinstanzen gesucht. Sind diese identifiziert und bestätigt, können im Umfeld dieser Instanzen gemäß der Messstellenanforderungen aus dem Modell die Feldgeräte bestimmt werden. In den Feldgeräten wird dann das benötigte Datum identifiziert und der Messstellenanforderungen in der Modellinstanz zugeordnet. Das Modell kann aus der Verfügbarkeit bestimmter Messstellen die berechenbaren Modellkennzahlen selbst bestimmen. Es ist möglich, dass aufgrund des Vorhandenseins oder der Positionierung der Messstellen in der Anlage keine Kennzahl berechnet werden kann. In diesem Fall ist ein Zugriff auf indirekt oder per Simulation bestimmte Ersatzwerte möglich [211]. Anderenfalls kann das Modell nicht verwendet werden.

### 3.3.1 Apparatemodelle und FDI Pakete

Wie bereits in der Beschreibung der Werkzeugkette angedeutet, besteht das Apparatemodell aus drei Teilen: der Struktur des Apparates, den benötigten Messstellen und den Berechnungsvorschriften für die Kennzahlen (Abbildung 3.9 auf Seite 44). Weiterhin können Parameter für die Adaption an statische Prozessbedingungen eingeführt werden. Da der Hersteller eines solchen Modells üblicherweise keine Expertise bei der Herstellung von Gerätebeschreibungen besitzt, ist eine durchgängige Werkzeugunterstützung notwendig ([2] / Abschnitt 4.3 ab Seite 75).

#### Modell

Die Basis für ein Modell ist eine Anforderung an Equipment, die – genauso wie bei den Feldgeräten die Funktionalität – als Rolle standardisiert ist. Diese Rolle legt die Funktion

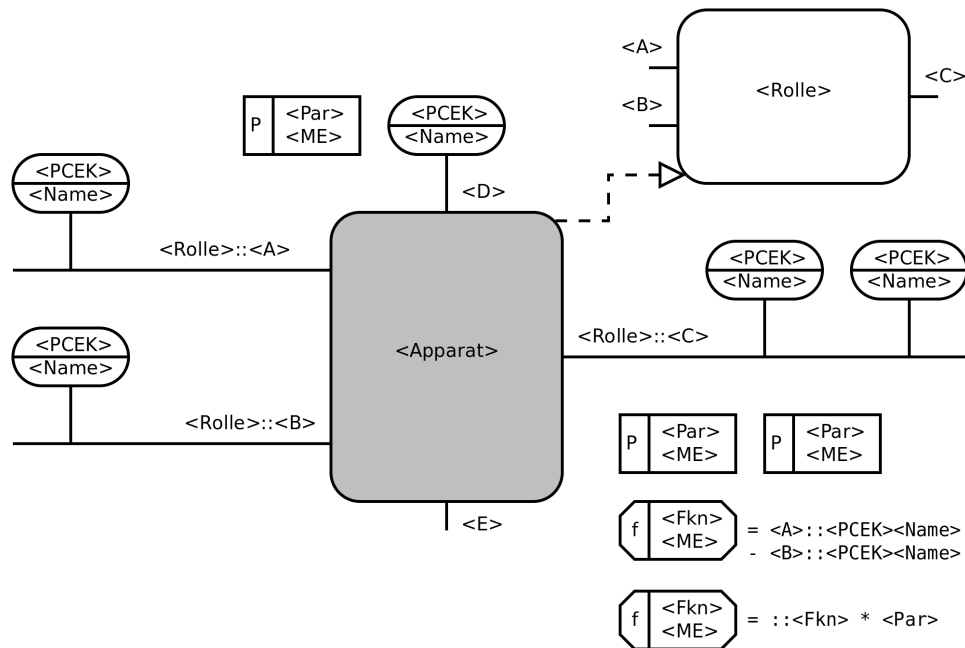


Abbildung 3.9: Generische Darstellung von Apparatemodellen

des Apparates fest und vergibt Namen für Schnittstellen mit bestimmten Funktionen wie z. B. *Zufluss*. Es ist ratsam, diese Bezeichnungen in die Implementierung dieser Rolle zu übernehmen, damit die standardisierte Schnittstellensemantik der Rolle mit dem standardisierten Namen eindeutig adressiert werden kann. Zusätzliche Anschlüsse können problemlos hinzugefügt werden. Namen, die bereits eine andere standardisierte Funktion repräsentieren, sind zu vermeiden. Gleichsam sollten standardisierte optionale Anschlüsse in der bekannten Form verwendet und nicht zweckentfremdet werden.

Die Daten zu den beschriebenen Modellelementen sind gemeinsam mit einem aussagekräftigen Symbol für den Apparat für die Strukturinformationen im Engineering-Artefakt hinreichend. In der vorgestellten Lösung erfolgte die Umsetzung mittels CAEX-SystemUnitClass, da entsprechende Vorarbeiten die meisten offen verfügbaren und unmittelbar verwendbaren Bibliotheken zur Verfügung stellten [40, 41, 42, 49, 47]. Auch die Entscheidungskriterien für das Datenformat CAEX bzw. AML aus [43, Tab. 2, S. 41] bestätigen diese Wahl.

Auf der Struktur aufbauend werden die Messstellenanforderungen für die einzelnen Schnittstellen festgelegt. Dabei werden die Typen (PCE-Kategorie [17]) und auch eine Maßeinheit, in die der Prozesswert für die Berechnungen konvertiert werden soll, festgelegt. Wenn Parameter zur Adaption des Modells benötigt werden, werden auch diese mit einer Maßeinheit versehen. Somit können die Maßeinheiten aller Berechnungen automatisch abgeleitet werden.

Zuletzt werden die Berechnungen für Kennwerte beschrieben. Im einfachsten Fall sind dies Gleichungen zur Bestimmung statischer Größen. Ist eine Eingangsgröße für eine Gleichung nicht verfügbar, kann die Gleichung später aus der Berechnung ausgeschlossen werden.

## FDI Paket

Die Entscheidung für eine Integrationstechnologie wurde aufgrund der Struktur, der Möglichkeit, Modelle einzubinden, und der unterschiedlichen Zugriffsmethoden auf die Feldgeräte getroffen. Gerätezugriff und Einbindung können mit den offenen Schnittstellen des Systems realisiert werden, ohne in das System eingreifen zu müssen. Damit wird die Rückwirkungsfreiheit, eine der Kernanforderungen, erfüllt.

Im FDI Package werden alle Definitionen des Modellherstellers in EDDL abgebildet [2]. Die Struktur lehnt sich an der von PA-Feldgeräten an. Alle Messstellenanforderungen werden als *Transducer-Block* eingeführt. Neben den Standardparametern sind weitere für den Primärmesswert und die Speicherung von dessen Verfügbarkeit vorgesehen. Wenn bei der Instanziierung ein angeforderter Messwert nicht aus einem Feldgerät bezogen werden kann, kann er mittels EDD-*Validity*-Mechanismus ausgeblendet werden. Dieser Mechanismus kann durchgängig für alle Abhängigkeiten automatisch implementiert werden. Alle Schnittstellen werden in je einen *Function-Block* zusammengefasst. Die darin definierten Parameter und Funktionsergebnisse werden als EDD-Parameter eingebettet. Der Apparat selbst erhält ebenfalls einen *Function-Block* zur Strukturierung der Parameter.

Die Berechnungen und Lesevorgänge auf den Geräten werden in eine Methode zusammengefasst, die mit EDD-Mitteln eine Caching-Funktion implementiert, um unnötig häufige Lesevorgänge abzufangen. Alle Kennzahlen werden in einer mit EDDL beschriebenen Benutzungsoberfläche zusammengefasst und angezeigt. Ebenfalls werden für alle Blöcke Menüs generiert.

Zum Schluss werden die generierte EDD-Datei und die generierte SystemUnitClass mittels der FDI-Packaging-Werkzeugkette [259, 260] exportiert und dem Anwender zur Verfügung gestellt.

### 3.3.2 Instanziierung

Die Instanziierung der Apparatemodelle erfolgt auf Basis von Engineeringdaten und der Identifikationsdaten der in Betrieb genommenen Feldgeräte. Dabei wird aufgrund der in Abschnitt 3.3.1 ab Seite 43 getroffenen Entscheidung für CAEX dieses Format verwendet. Die Anlagenstruktur wird wie in [261] als Instanzhierarchie exportiert und kann nun nach den SystemUnitClasses der zu instanzierenden Modelle durchsucht werden. Können dabei nicht alle erwarteten Apparate gefunden werden, ist ebenfalls eine Suche nach der implementierten Rolle möglich. Dann ist jedoch das Ergebnis auf jeden Fall händisch zu bewerten.

Die Schnittstellen bzw. Prozessanschlüsse des Apparates sollten, wenn die Rolle oder die SystemUnitClass ohne Änderungen verwendet wurde, mit denen im Apparatpaket übereinstimmen. Somit können die Anschlüsse zugeordnet werden. Den entsprechenden Rohleitungen kann man nun in den Engineeringdaten folgen und mit ihnen verbundene Messstellenanforderungen sammeln [261, 239, 211]. Diese werden anhand ihrer PCE-Kategorie mit den Anforderungen des Modells an diesem Anschluss abgeglichen. Bei Übereinstimmung kann das Gerät gesucht werden.

Auch hier wird ein Vorteil der verwendeten FDI-Technologie genutzt: Die Geräte sind mittels eines generischen OPC UA-Clients einheitlich auffindbar und zugreifbar. Aufgrund des Wissens über die Struktur der von FDI unterstützten Technologien FF, HART, PA,

DP und PN IO können die in den Feldgeräten abgelegten Messstellenbezeichnungen bei allen Technologien nach festgelegten Regeln ausgelesen werden. Somit sind die Geräte mit ihrer Zugriffsschnittstelle identifiziert. Spätere Änderungen im Namensschema, welche diese Identifikation unsicher machen würden, sollten aus Gründen der Praktikabilität ausgeschlossen sein. Solche Eingriffe würden Engineeringdaten aus unterschiedlichen Planungsphasen inkonsistent machen.

Alle Parameter des identifizierten Gerätes, und damit auch der in einen Parameter gemappte Prozesswert, sind mittels eines einheitlichen Zugriffsmechanismus' über OPC UA auslesbar. Auch jetzt kann das spezifische Wissen über die Kommunikationstechnologien verwendet werden, um in den Geräten den richtigen Parameter zu finden (vgl. [76]). Dafür sind beispielsweise in HART feste Parameternamen vorgeschrieben. In PA wird die Gerätestruktur und die Namen und Funktion der Parameter vom Profil [116] festgelegt.

#### 3.3.3 Betrieb

Die bei der Instanziierung festgelegten Mappings zwischen Modellanforderungen und Parametern in Geräten können nicht innerhalb des Paketes aufgelöst werden. Auch ein Zugriff quer durch das Objektmodell des FDI Server ist sowohl implementierungsabhängig und damit unzuverlässig als auch ein Sicherheitsrisiko. Deswegen wird der Zugriff auf die Geräte über die definierten Schnittstellen gelöst. Die folgenden Einzelschritte sind exemplarisch in Abbildung 3.10 dargestellt.

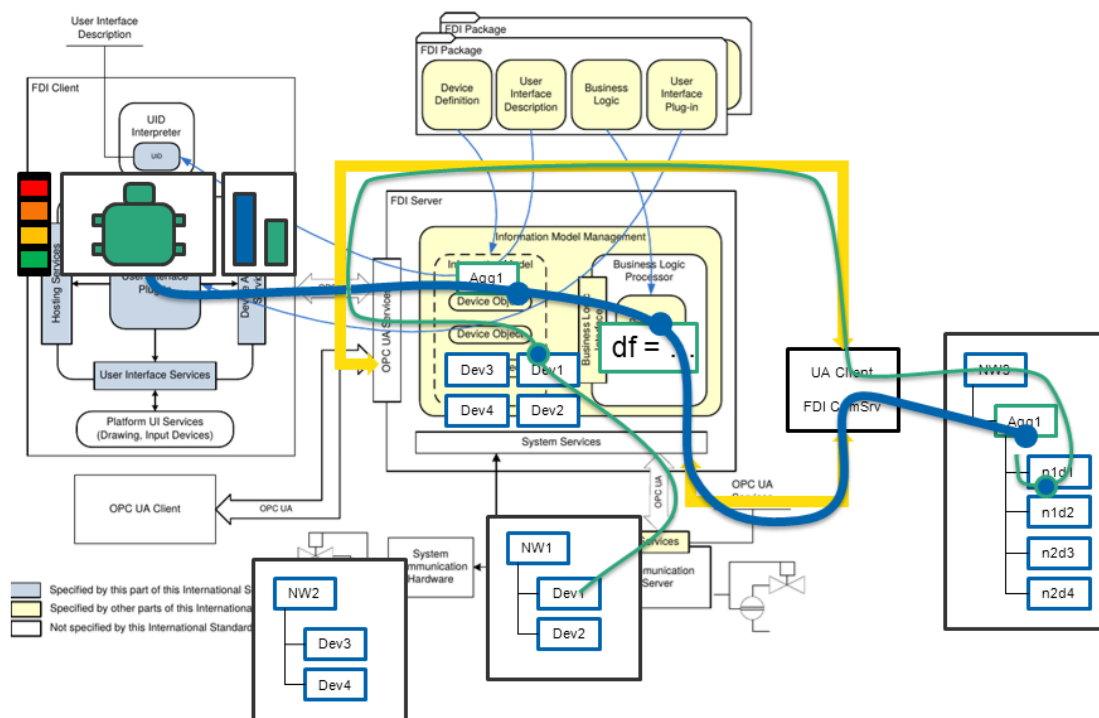


Abbildung 3.10: Lesevorgänge bei Aktualisierung von Kennzahlen für das Apparatemodell

Das Paket greift über ein virtuelles Netzwerk auf virtuelle Feldgeräte, die die Apparate repräsentieren, zu. Dies wird über die FDI Communication Server-Schnittstelle realisiert.

Die Zugriffe auf die verschiedenen Blöcke der Geräte werden mittels der Mappinginformationen auf die Objektmodelle der realen Feldgeräte weitergeleitet. Dies wird mittels eines OPC UA-Clients und der OPC UA-Schnittstelle des FDI Servers umgesetzt. In [262, Abs. 4.5f, S. 47] wird beschrieben, dass das FDI Package nachgeladen wird, wenn auf das Informationsmodell des entsprechenden Feldgerätes zugegriffen wird. Zur Optimierung kann der Client OPC UA-Lesevorgänge bündeln.

Die berechneten Kennzahlen müssen zuletzt in den Systemen eingebunden werden, in denen sie angezeigt, verwendet oder ausgewertet werden sollen. Durch die Verwendung derselben Technologie (FDI) wie für die Feldgeräte, ist eine Einbindung in gerätezentrierte PAM-Tools [142, 140, 141] gleichartig möglich, wenn diese eine FDI Client-Komponente besitzen. Genauso können die Kennzahlen mit standardisierten Zugriffsmethoden aus dem OPC UA-Objektmodell gelesen werden. HMIs mit nativer OPC UA-Unterstützung können dies ohne weitere Softwarekomponenten realisieren. Zur Einbindung ist in diesem Fall ein zusätzlicher Engineeringaufwand zu erbringen.

## 3.4 Modell-Anwendungsfälle

Über die verschiedenen von FDI unterstützten Feldbustechnologien sind nicht nur die Prozesswerte, sondern auch Grenzwerte und andere zusätzliche Daten zugreifbar. Wenn die Applikationsprofile umgesetzt sind und die entsprechenden Daten bei der Inbetriebnahme vollständig und konsistent in die Geräte übertragen wurden, ist eine größere Vielfalt an Anwendungsfällen möglich.

Drei Anwendungsfälle für Bewertungsmodelle sind inzwischen in den Produkten der PLS-Hersteller integriert und dadurch gut abgedeckt. Ein Modell überwacht für Wärmeübertrager Anlagerung, Verstopfung und die Effektivität. Dafür werden die Prozesswerte aus Temperatur- und Durchflussmessgeräten verwendet [221, 141, 224]. Ein zweites überwacht Pumpen auf Effizienz. Es verbindet Daten aus dem Antrieb mit denen aus nachgeschalteten Regelventilen oder Durchflussmessern [206, 221, 141, 224]. Der dritte Standard-Anwendungsfall ist eine Ventilüberwachung. Diese schließen aus den Unterschieden zwischen Sollwert und tatsächlicher Position sowie ihren dynamischen Änderungen auf Schwergängigkeit, Ablagerung, etc. [207, 141, 224].

### Parametrierarme Grundoperationen

Alle aufgeführten, im PLS implementierten Grundoperationen benötigen einen erheblichen Parametrieraufwand. Die generischen Modelle benötigen beispielsweise die Eingabe der Kennlinien von Antrieben und anderen, zur korrekten Berechnung notwendigen Parametern. Spezialisierte Modelle könnten diese Parametersätze fertig mitbringen oder spezifisch für bestimmte Ausprägungen implementiert sein.

### Kennzahlen zum Arbeitspunkt

Wie im Beispiel des Wärmeübertragers können bei anderen statisch überwachbaren Prozessen aus Sätzen von Prozesswerten aus der Umgebung des Apparates Kennzahlen berechnet werden. Diese können sowohl für Apparate- also auch Prozessdiagnosen verwendet werden.

#### **Einhaltung von der Auslegung entsprechenden Grenzwerten**

Modelle können unter gegebenen Grenzwerten für z. B. Medieneigenschaften die richtige Einstellung der Alarmwerte in Feldgeräten prüfen, um unerlaubte Zustände als Gerätealarm gemeldet zu bekommen. Somit könnte ein Betrieb außerhalb der Spezifikation leichter bemerkt werden. Eine solche Prüfung kann sowohl einmalig als auch regelmäßig vorgenommen werden.

#### **Geräte Diagnose mit Daten aus Nachbargeräten**

Auch in der gerätezentrierten Selbstdiagnose können Messwerte aus benachbarten Feldgeräten für eine Verbesserung der Einschätzung des Zustandes eines Feldgerätes verwendet werden. So könnte ein Pumpenantrieb, wie auch in den PLS-Beispielen aufgeführt, mit Druck-, Temperatur- oder Durchflussmesswerten aus ihrem Wirkungsbereich besser beobachtet werden. Ihr Diagnosemodul müsste in diesem Fall zweigeteilt sein (Antrieb und übergreifende Beurteilung).

### **3.5 Verwertungsszenarien**

Im Folgenden soll kurz umrissen werden, wie ein Hersteller von Apparatemodellen diese vermarkten könnten. Voraussetzung ist, dass die Softwarekomponenten zur Integration der FDI Apparatepackages (siehe Abschnitt 3.3.3 ab Seite 46 und Abschnitt 4.4 ab Seite 86) frei verfügbar und mit allen am Markt befindlichen FDI Servern kompatibel sind.

**Mehrwert zu einem Apparateprodukt:** Ein Hersteller eines Apparates mit einer Standardfunktion kann ein Modell als Alleinstellungsmerkmal / Unique Selling Proposition zum Produkt anbieten.

**Eigenes Produkt:** Ein Modell mit geringem oder ohne Parametrierungsaufwand kann von Spezialisten erstellt und im Rahmen von Beratungen vermarktet werden.

**Absicherung von Gewährleistungszusagen:** Ein Hersteller eines Apparates kann erweiterte Garantieleistungen anbieten, wenn sein Modell mit Grenzwertüberwachung dauerhaft die Einhaltung der garantierten Prozessbedingungen überwacht.

### **3.6 Migrationsstrategie**

Das vorgestellte Konzept zur Apparatemodell-Integration basiert auf der FDI-Technologie, die sich zu Beginn der vorliegenden Arbeit noch in der Spezifikationsphase befand. Durch Verzögerungen im Projektplan gibt es Abhängigkeiten die bis zur Fertigstellung der vorliegenden Arbeit nicht aufgelöst werden konnten. Tabelle 3.1 zeigt die zeitlichen Abhängigkeiten, welche von Einführung FDI-Technologie herrühren. Eine weitere Abhängigkeit ergibt sich aus einem Feature des verwendeten OPC UA-Servers. Der generische *NodeManager*, der eine XML-NodeSet-Datei laden kann, wie in [12] gefordert, wurde mit dem *Unified Automation C++ Server 1.4* im November 2014 ausgeliefert.



**Tabelle 3.1:** Technologien und ihre Verfügbarkeit laut [154] und [156]

Komponente	geliefert mit	Zeitraum ab Q2 2013 [154]	Zeitraum ab Q4 2014 [156]
Parser für die harmonisierte EDD	Betatestphase Reference Host	Q2 2014	ok
FDI Server	Betatestphase Reference Host	Q2 2014	ok
Kommunikation über FDI Communication Server	nicht unterstützt	„Zukunft“	single channel ok; multi channel „Zukunft“
Gerätespezifisches Informationsmodell	Betatestphase Reference Host	Q2 2014	ok
Unterstützung modularer Feldgeräte	nicht verfügbar	„geplant“	„Zukunft“
FDI Server mit OPC UA-Schnittstelle	nicht verfügbar	„Zukunft“	„geplant“

Um die in Kapitel 4 ab Seite 57 beschriebenen prototypischen Implementierungen dennoch testen zu können, wurden die verwendeten Schnittstellen der FDI-Komponenten ohne Business Logic implementiert und mit Testdaten versehen. Im Folgenden wird auf diese Dummy-Komponenten eingegangen und die Migration zu später verfügbaren FDI-Produkten beschrieben.

### 3.6.1 FDI Projektplan und Verzögerungen

Ein Zeitplan des FDI Projektes von 2008 [263] sah Spezifikation und Validierung bis Mitte 2010 vor. Die finale Spezifikation lag jedoch erst 2014 vor, die Validierung begann zu diesem Zeitpunkt und wurde 2015 abgeschlossen [259]. Unter anderem war die Harmonisierung der EDDL [147] eine Aufgabe, die das Projekt blockiert hat. Auch zum Stand 2013 [154] sind einige Abhängigkeiten noch nicht aufgelöst. Der Reference Host wurde im ersten Quartal 2015 freigegeben. Es fehlen aber weiterhin die OPC UA-Schnittstelle und FDI Communication Server-*multi channel*-Unterstützung [259, 156]. Die erste Umsetzung des FDI Servers mit OPC UA-Schnittstelle [157] war erst im zweiten Quartal 2015 verfügbar und wurde nicht mehr für Untersuchungen berücksichtigt.

### 3.6.2 Ohne voll funktionalen FDI Server

Um einen FDI Server mit OPC UA-Unterstützung zu emulieren, muss ein OPC UA-Informationsmodell gemäß [190] und [159, 160, 158] erstellt und die benötigten Feldgerätedaten hinterlegt werden. Dies wurde in [11] realisiert. Um in einen solchen Server auch die FDI Apparatepackages zu emulieren, müssen die Server zu einem emulierten FDI Server aggregiert werden. Eine Umsetzung ist in [12] beschrieben. Diese Umsetzung

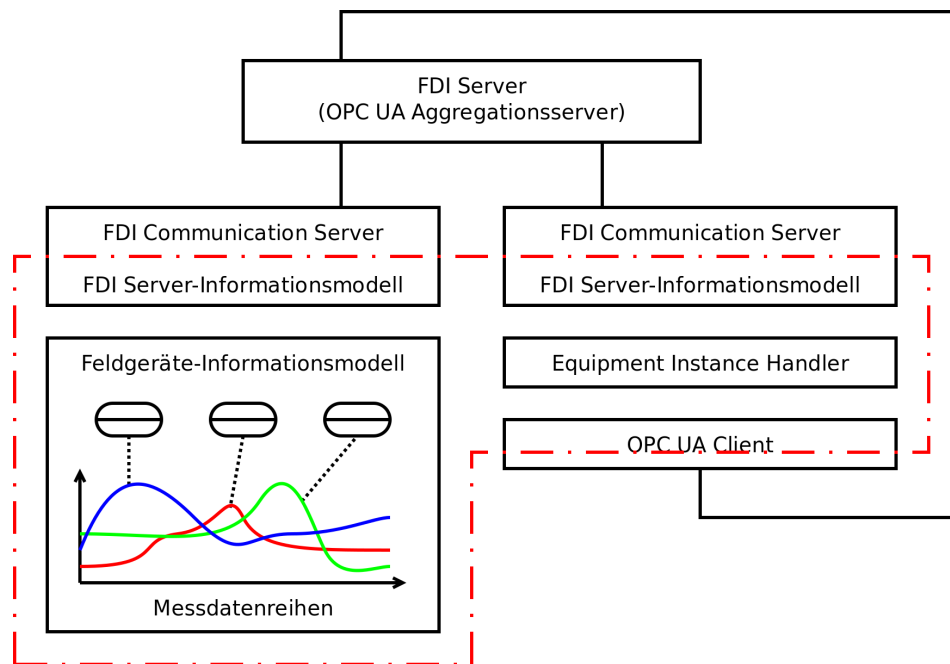


Abbildung 3.11: Architektur und Informationsmodell ohne OPC UA-fähigen FDI Server

konnte noch nicht auf ein generisches Informationsmodell im verwendeten OPC UA-Server zurückgreifen. Eine auf das später verfügbare Feature aufsetzende Weiterentwicklung ist in Abschnitt 4.1.2 ab Seite 62 und Abschnitt 4.4.2 ab Seite 88 dokumentiert. Das Testsetup für den Aggregationsserver ist in Abbildung 3.11 auf Seite 50 dargestellt. Die mit einer roten Strichpunktlinie umrandeten Komponenten sind im Rahmen dieser Arbeit entstanden, die restlichen Komponenten wurden im Rahmen von [11] und [12] entwickelt.

### 3.6.3 Mit einem marktreifen FDI Server

Aufgrund der Verschiebungen im Projektplan von FDI ist ein Testen mit diesem Szenario nicht möglich. Wegen der ausschließlichen Verwendung von spezifizierten Schnittstellen, kann jedoch davon ausgegangen werden, dass sich die Lösung problemlos integriert. Abbildung 3.12 zeigt die geänderte Situation mit nur noch den zusätzlichen Komponenten, die für die Integration von Apparatemodellen in der Standardarchitektur notwendig sind. Dabei muss eine Komponente verändert werden: Um das vollständig FDI-Informationsmodell im Aggregationsserver halten zu können, wurde im FDI Communication Server ein FDI Server-Informationsmodell implementiert. Dieses muss mit der Umstellung durch das schmalere FDI Communication Server-Informationsmodell ersetzt werden. Diese Anforderung muss in der Implementierung in Abschnitt 4.4 ab Seite 86 beachtet werden. Die mit einer roten Strichpunktlinie umrandeten Komponenten wurden bis auf den OPC UA-Client ([11]) in dieser Arbeit entworfen oder entwickelt. Alle weiteren sind zukünftige Standardkomponenten.

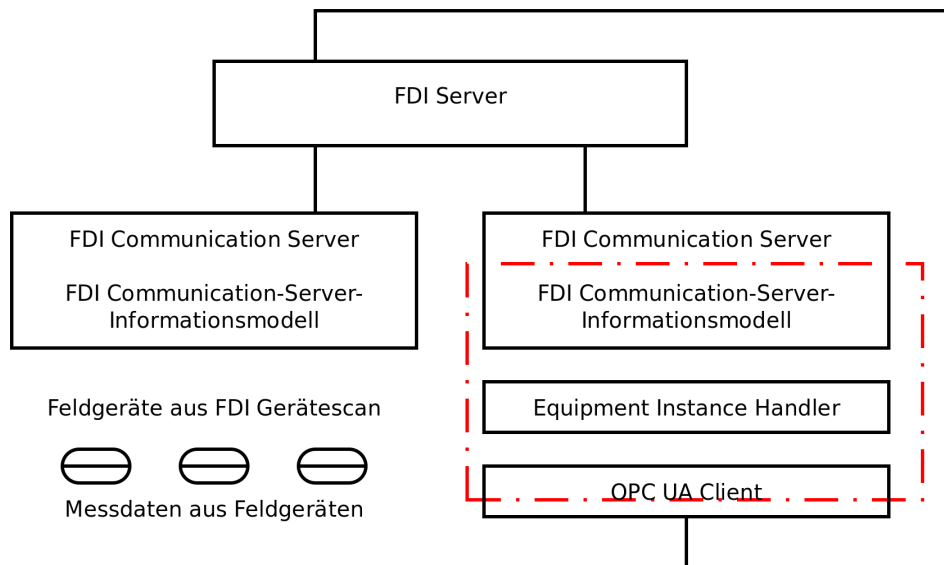


Abbildung 3.12: Architektur und Informationsmodell mit OPC UA-fähigen FDI Server

## 3.7 Diskussion

Zum Abschluss soll das beschriebene Konzept zur Modellintegration unter verschiedenen Gesichtspunkten kritisch betrachtet werden. Dabei sollen die technischen Grenzen und die praktische Anwendbarkeit diskutiert werden.

### 3.7.1 Betrachtung der beteiligten Akteure

Beim Einsatz des vorgestellten Konzepts sind maßgeblich der Infrastrukturhersteller, der Apparate- bzw. Apparatemodell-Hersteller und der Anwender beteiligt. Alle müssen einen Nutzen aus dem Einsatz ziehen können.

#### Infrastrukturhersteller

Die zusätzlich benötigte Infrastruktur, bestehend aus dem Equipment Instance Handler und den notwendigen Kommunikationskomponenten, muss entwickelt und gepflegt werden. Es ist erforderlich, dass sie mit allen am Markt befindlichen FDI Servern kompatibel sind. Somit ergeben sich erhebliche Kosten, die auf den Anwender der Modelle, der diese Komponenten einsetzt, umgelegt werden müssen. Ebenfalls muss die Software zur Erstellung von Paketen entwickelt und gewartet werden. Diese Kosten müssen auf die Hersteller der Apparatemodelle umgelegt werden. Dies ist ein erhebliches Risiko, da der Erfolg nicht nur von einer kritischen Masse an verfügbaren Modellen, sondern auch von der Einheitlichkeit der Infrastruktur abhängt. Demzufolge ist eine Standardisierung von Modellstruktur und Kommunikationswegen notwendig, um verschiedene Lösungen bei gleichzeitigem Anwendernutzen zu ermöglichen. Die Konkurrenz zwischen EDD und FDT bei den Geräteintegrationstechnologien hat gezeigt, dass die Akzeptanz mehrerer Lösungen niedrig ist.

#### **Apparatemodellhersteller**

Ob aus einem Bewertungsmodell ein Nutzen gezogen werden kann, hängt von der Funktion und Einsparpotential beim Betrieb des bewerteten Apparates ab. Wenn sich gemäß der Anwendungsfälle und Geschäftsmodelle ein Nutzen für den Apparatehersteller ergibt, kann er ein Modell implementieren. Die Werkzeuge hierfür sollten ihn bei der Erstellung unterstützen, insbesondere bei der Verwendung der Technologien, mit denen er eigentlich nicht in Berührung kommt. Dabei ist vor allem die Investitionssicherheit wichtig. Das Modell sollte in allen Anwendersystemen einsetzbar sein und für einen angemessenen Zeitraum unterstützt werden. Dies ist eine Forderung an den Infrastrukturhersteller. Gleichzeitig sollte das Modell vor Nachahmung geschützt werden. Letzteres wird vom Binärformat der EDDL weitestgehend geleistet.

#### **Apparatemodellanwender**

Ein Anwender von Apparatemodellen hat – direkt und anteilig indirekt – den gesamten Aufwand für die Infrastruktur und Modellentwicklung zu erbringen. Gleichzeitig hat er nur einen indirekten Nutzen, indem er Stillstände vermeiden, seine Wartung besser planen und seine Produktqualität steigern kann. Trotzdem ist der Nutzen solcher Methoden im gewissen Maß nachgewiesen, jedoch schwierig messbar [264, Kap. 3, S. 71f]. All diese Arten des Nutzens lassen sich auch nicht direkt auf den Einsatz eines solchen Modells zurückführen, da weitere Tätigkeiten mit dem gleichen Ziel durchgeführt werden. Deswegen ist es wichtig, dass keine unnötigen Aufwände beim Engineering und im Betrieb der Modellinstanzen entstehen. Dafür wurde ein Konzept zum automatischen Engineering entwickelt. Dennoch ist ein gewisser Aufwand bei der Einrichtung und bei der Auswertung der von den Modellen gelieferten Kennwerte zu erbringen. Die Integration der Kennzahlen funktioniert mit den in anderen Werkzeugen üblichen Mechanismen. Der Einsatz von zusätzlichen Softwarekomponenten darf keine Rückwirkungen auf die Anlage haben. Dies passiert nur beim Lesen von Daten aus den Feldgeräten. Zur Sicherstellung des Produktivdatenverkehrs besitzen alle Feldbusse Mechanismen zur Priorisierung des zyklischen Verkehrs. Zusätzlich wurde eine Methode zur Vermeidung von Überlastungen vorgesehen. Ein Schreibzugriff, also die Manipulation von Feldgeräte über die Modelle, kann durch die OPC UA-Schnittstelle des FDI-Servers unterbunden werden. Somit kann ein schädlicher Einfluss ausgeschlossen werden.

#### **3.7.2 Technische Voraussetzungen in der Anlage**

Für die Instanziierung der Modelle ist es erforderlich, dass genügend die Anforderungen des Modells erfüllende Messstellen in der Umgebung der zu überwachenden Apparate vorhanden sind. Üblicherweise werden zur Prozessführung deutlich weniger MSR-Stellen benötigt, als für eine Prozessdiagnose, eine CM- oder PAM-Aufgabe. Auch ist ein Zugriff auf Daten aus der Sicherheitstechnik nicht erlaubt. Da die Planung und der Betrieb von Messstellen mit Anschaffungs- und Wartungskosten behaftet ist und der Mehrwert von zusätzlichen, unter Umständen redundanten Geräten nicht einfach zu beziffern ist, spricht kurzfristig betriebswirtschaftlich nichts für eine vollständige Ausrüstung. Eine Abwägung zwischen geringeren Investitionskosten und späterer größerer Verfügbarkeit ist sowohl

von Budget und Automatisierungsaufgabe als auch der Einschätzung der Planer und Entscheidungsträger abhängig.

Gleichzeitig hängt die Einsetzbarkeit der Modelle von der Anzahl der geforderten Messstellen ab. Hier liegt es am Geschick des Modellherstellers, möglichst wenige Abhängigkeiten zu nutzen. Auch kann über einen gut angelegten Abhängigkeitsgraph von Messstellen und Zwischenergebnissen ein möglichst großes Teilmodell betrieben werden, auch wenn nicht alle Anforderungen instanziiert werden können.

Eine generelle Aussage zum Grad der Abdeckung der Messstellenanforderungen lässt sich nicht treffen, da dies maßgeblich von den Modellen abhängt, deren technische Umsetzung nicht Teil dieser Arbeit ist.

Weiterhin wurde vorgeschlagen, fehlende Messstellen über Berechnungen oder Simulationen indirekt zu bestimmen [211]. Dies erfordert jedoch einen höheren Aufwand, der in der vorliegenden Arbeit nicht implementiert wurde. Ob die Integration von Ersatzwertmechanismen in die Modell-Pakete oder in den Equipment Instance Handler (siehe Abschnitt 4.4 ab Seite 86) sinnvoller wäre, soll nicht diskutiert werden.

### 3.7.3 Technologische Voraussetzungen in der Anlage

Wie im Konzept bereits beschrieben, sind für den Einsatz in einer Anlage gewisse Voraussetzungen zu erfüllen. Für die Integration sind konsistente und vollständige Engineeringdaten erforderlich. Diese müssen in geeigneter Form exportiert werden. Die Konsistenz der Daten ist vom Anlagenbetreiber abhängig. Die Pflege der Planungsdaten bzw. Dokumentation einer Anlage wird von Betreibern je nach Größe, Komplexität und einzuhaltenden Vorschriften für mehr oder weniger wichtig erachtet. Die Abdeckung ist hier nicht generalisiert einschätzbar. Die Vollständigkeit der Engineeringdaten hängt von den verwendeten Planungstools ab. Prinzipiell ist auch ein anderes Exportformat als das gewählte CAEX einsetzbar, sodass hier mit einer Abdeckung zu rechnen ist.

Weiterhin gibt es Anforderungen an die technische Ausstattung der Anlage. Die MSR-Technik muss digital eingebunden sein. Das heißt, Feldgeräte müssen über digitale Kommunikationstechnologien angebunden sein. Dies ist die Voraussetzung für eine weitere technische Randbedingung, nämlich den Einsatz der FDI-Technologie. Die Abdeckung von Anlagen mit digitaler Anschaltung ist, auch geschuldet der langen Anlagenlaufzeit, immer noch nur zwischen 5 und 10 % [265]. Wenn die Technologien eingesetzt werden, dann jedoch konsequent und durchgängig. Durch die Harmonisierung von EDD- und FDT-Technologie zu FDI werden die Vorteile der digitalen Feldanbindung für größere Anteile sorgen.

Durch den Einsatz eines FDI Servers werden etliche Voraussetzungen automatisch erfüllt. Da die Feldgeräte über diesen Server parametrisiert und konsequenterweise ins PAM-System eingebunden werden, sind alle Geräte im Informationsmodell vorhanden. In [266] wird zusätzlich eine Lösung vorgeschlagen, die weitere Gerätezugriffstechnologien in ein gleichartiges Datenmodell übernehmen kann und damit eine größere Durchgängigkeit schafft. Mit der bisher ausstehenden Unterstützung für modulare Geräte und der Fertigstellung des Applikationsprofils *Remote IO for Process Automation* ist auch ein Zugriff auf Messstellen, die mittels Remote IO [90] angebunden sind, möglich. Mit dem FDI-Gerätescan kann der Gerätebestand jederzeit aktualisiert werden. Auch kann dann die Einbindung der Apparatemodelle ins PAM gleichsam erfolgen.

Die NAMUR hat sich bereits länger für den alleinigen Einsatz von FDI ausgesprochen [267]. Hervorgehoben wurde dabei auch die Notwendigkeit der OPC UA-Schnittstelle im FDI Server. Es ist davon auszugehen, dass diese – trotz der niedrigen Priorität im Reference Host [154] – aufgrund der vielen Anwendungsmöglichkeiten [1, 268] in den Produkten integriert wird. Die Validierung der Spezifikation der OPC UA-Informationsmodelle für FDI ist erfolgreich abgeschlossen [156]. Im Quartal 2 2015 hat *ABB* einen FDI Server mit OPC UA-Schnittstelle [157] vorgestellt.

#### 3.7.4 Geforderte Dynamik

Für Modelle, die eine kontinuierliche Überwachung anbieten, ist ein wichtiges Kriterium die Zykluszeit, in der die Daten abgefragt werden müssen. Dabei ist es wichtig, dass dieser Zyklus nicht zu lang ist, um keine besonderen Ereignisse oder Verlaufsänderungen zu verpassen. Ebenfalls muss ein Satz von gelesenen Messdaten möglichst gleichzeitig erfolgen, um einen konsistenten Datensatz zu erhalten. Da sich die für ein Modell abgefragten Geräte notwendigerweise in örtlicher Nähe zueinander befinden, ist auch nicht unwahrscheinlich, dass sie auf dem gleichen Feldbussegment oder den gleichen Multiplexer (je nach Technologie) angeschaltet sind. Von diesem Worst-Case ist bei der Abschätzung der Modellgenauigkeit auszugehen, weil dies eine exakte Gleichzeitigkeit ausschließt. Die reine Übertragungszeit für die notwendigen Anfrage- und Antworttelegramme in PA und FF ist bei der Baudrate 31.25 kBaud je nach Länge der Daten zwei bis drei Millisekunden. Wenn vorher eine Verbindung aufgebaut oder die Anfrage wiederholt werden muss, weil die Daten nicht sofort im Gerät bereit zum Senden sind, kommen ein bis drei Buszyklen hinzu. Bei HART mit einer Bitrate von 1200 Bit/s dauert eine Leseanfrage mit Antwort ungefähr eine halbe Sekunde. Je nachdem, wie viele Geräte über dasselbe Medium abgefragt werden müssen, addiert sich diese Zeit, wobei Wartezeiten, durch parallele Vorgänge nicht zwingend mehrfach auftreten. Das Zeitintervall, in dem gleichzeitige Messungen aufgenommen werden können, ist also auch von der verwendeten Feldbus-Technologie abhängig.

Es gibt auf dem Medium eine vorhandene Bandbreite, die mit dem zyklischen Verkehr (nicht bei HART) und anderen azyklischen Zugriffen, z. B. vom PAM-System, geteilt werden muss. Diese ist zugunsten der Verfügbarkeit zu schonen, das heißt, es ist ein Kompromiss zwischen Busauslastung und schnellen Abfragezyklen zu finden. Über in den FDI Apparatepackages erzeugten Mechanismen werden Methoden zum Cachen der Werte und zur Sicherstellung einer minimalen Abfrage-Zykluszeit implementiert. Dies gibt die Möglichkeit einer Adaption der Zugriffsstrategie an die Kommunikationsstruktur der Anlage.

Insgesamt lässt sich schlussfolgern, dass sich ein Abfragezyklus unter zehn Sekunden verbietet. Beim Einsatz mehrerer Apparatemodelle ist eine Minute möglich, fünf Minuten realistisch. Bei vielen HART-Geräten müssen diese Zeiten noch erhöht werden. Aufgrund der langsamen Vorgänge in der Verfahrenstechnik sind auch mit derartig langen Zyklen Modelle betreibbar.

### 3.7.5 Skalierbarkeit

Mit steigender Anzahl von instanziierten Apparatemodellen mit kontinuierlicher Überwachung steigt der Kommunikationsbedarf auf den Feldbussen und der Ressourcenbedarf im FDI Server. Die Kommunikation aufgrund von Anfragen aus dem Apparatemodell ist im Konzept nicht übergreifend gesteuert oder synchronisiert. Somit kann es zu Engpässen bei der Bandbreite kommen, insbesondere, wenn mehrere Apparatemodelle auf dieselben Geräte zugreifen oder in denselben Feldbussegmenten kommunizieren. Alternativ könnte die Umleitung der Zugriffe auf Primärmesswerte in ein PIMS die Netzwerklast deutlich senken. Dafür ist jedoch ein erheblicher Implementierungsaufwand im FDI Server bzw. Equipment Instance Handler (EIH) notwendig. Zudem sind Schnittstellen der PIMS, oder wenn OPC UA verwendet wird, das Informationsmodell nicht standardisiert. Deshalb wird diese Variante nicht weiter betrachtet.

Je nach Implementierung des FDI Server ist auch der Speicherbedarf für kontinuierlich im *Online*-Modus befindliche Feldgeräte und die Apparatemodelle entsprechend einzuschätzen. Der in [6] verwendete EDD-Parser benötigt zweieinhalb bis drei Megabyte Hauptspeicher für eine geladene und zur Kommunikation angebundene EDD. In einem FDI Server kommt das OPC UA-Informationsmodell hinzu. Bei einer größeren Menge von Geräten und Apparatemodellen muss eine effizientere Methode zur Implementierung gefunden werden. Dies zeigt auch eine Untersuchung an einem aktuellen OPC UA-Aggregationsserver [269]. In dieser Implementierung wurden bei der Massenverarbeitung noch deutliche Performanceschwierigkeiten aufgedeckt.

### 3.7.6 Verhältnis von Aufwand und Nutzen

Wie in jedem Projekt ist auch bei der Einrichtung einer Infrastruktur wie der vorgestellten die Frage zu stellen, ob der notwendige Aufwand den möglichen Mehrwert rechtfertigt. Analog zu dem in [180] beschriebenen quantifizierbaren und nicht-quantifizierbaren Nutzen für ein MES-System kann auch der Nutzen der Apparatemodelle in Frage gestellt werden. Als Beispiel wird ein CM-Apparatemodell diskutiert:

Dessen Aufgabe ist es, den „Gesundheits“-Zustand eines Apparates zu bestimmen. Aus seinen Kennzahlen kann bei entsprechender Auswertung unter anderem auf höheren Energieverbrauch, einen drohenden Totalausfall oder andere Folgen der Nichtbeachtung des Apparatzustands geschlossen werden. Ob ein solches Apparatemodell verwendet werden soll, wird anhand der wahrscheinlich vagen Antworten auf folgende Fragen entschieden:

- Wann fällt ein schlechter Zustand mit und wann ohne Modell auf?
- Welche Auswirkungen hat der zu detektierende Zustand?
- Wie hoch ist die Wahrscheinlichkeit von Fehlalarmen oder nicht erkannten kritischen Zuständen?
- Welche Kosten lassen sich überhaupt vermeiden?
- Ändern sich Wartungsintervalle zur Behebung der erkannten Zustände?
- Welche Mehrkosten entstehen durch Einrichtung und Auswertung der Modelle?

Die Frage, ob sich der Einsatz eines Bewertungsmodells lohnen wird, kann im Rahmen dieser Arbeit nicht beantwortet werden, weil sowohl das Modell als auch die Verwendung der Ergebnisse einen erheblichen Einfluss auf die Antwort haben. Ebenfalls kann eine analoge Diskussion zu MES-Systemen herangezogen werden [264]. Hier wird der Aufwand gegen Wertschöpfungsmöglichkeiten abgewogen. Ein Ergebnis ist dabei, dass mit zunehmender Automatisierung der Engineeringaufwände auch kleinere Verbesserungen lohnenswert werden.



## 4 Entwurf und Implementierung

Dieses Kapitel beschreibt eine Möglichkeit der technischen Umsetzung des in Kapitel 3 ab Seite 35 ausgearbeiteten Konzepts zur Modellbeschreibung und Modellintegration. Der Entwurf und die Implementierung der zuvor beschriebenen Softwarekomponenten oder -applikationen wird dokumentiert und auf die Umsetzung der getroffenen Designentscheidungen eingegangen. Die Umsetzung ist prototypisch. Es wurde versucht, die Umsetzbarkeit und Durchgängigkeit des Konzepts nachzuweisen. Darum wurde, wenn für eine Aufgabe mehrere Technologien einsetzbar waren, nur eine ausgewählte angewendet. Eine industrietaugliche Lösung benötigt eine deutlich breitere technologische Unterstützung.

### 4.1 Bereitstellung des Informationshaushalts

Um einen Prototypen für die Umsetzung des Konzepts zur Modellintegration entwickeln und testen zu können sind technische Voraussetzungen zu erfüllen. Da in der Praxis die Durchdringung mit einigen verwendeten Technologien nicht so weit fortgeschritten war, dass Produkte verfügbar waren, die eingebunden bzw. deren Ausgaben verwendet werden konnten, mussten diese Voraussetzungen geschaffen werden. Das sind zum einen vollständige Engineeringdatenexporte und zum anderen ein FDI Server, deren Erstellung bzw. dessen Emulation in den folgenden Abschnitten behandelt wird.

#### 4.1.1 Engineeringdaten Anlage

Für die spätere automatisierte Instanziierung der Apparatemodelle werden Anlagenstrukturinformationen benötigt. Im beschriebenen Proof-of-Concept wurden diese Daten mit CAEX (IEC 62424 [17]), [38, 270] abgebildet.

##### 4.1.1.1 CAEX

In Arbeiten wie [38] wurde bereits gezeigt, dass sich die proprietären Engineeringdatenbanken in das CAEX-Format abbilden lassen. Für eine eindeutige Festlegung der Semantik der Daten werden standardisierte Bibliotheken benötigt. Das sind in den verwendeten R&I-Fließbildern:

**Schnittstellenbibliotheken**, die einheitliche Verbindungstypen definieren. Mit ihnen kann man zwischen kompatiblen und inkompatiblen (z. B. Stoff transportierenden und informationstechnischen) Schnittstellen unterscheiden und sie attributieren (z. B. Nenndurchmesser von Rohren).

**Rollenbibliotheken**, die verfahrenstechnische und MSR-Anforderungen abstrakt abbilden. Diese repräsentieren unter anderem die PLT-Stellen gemäß IEC 62424 [17] und die Apparate und Verrohrung gemäß ISO 10628 [27, 28] ab.

**Systemeinheiten** (`SystemUnitClasses`), die konkrete Umsetzungen der verwendeten Rollen repräsentieren. Das sind üblicherweise Produkte, die im Detail Engineering für die abstrakte Funktion, welche die Rolle beschreibt, eingesetzt werden.

**Instanzhierarchien**, die R&I-Fließbilder von Anlage, Teilanlage oder Anlagenteilen detailliert darstellen. Sie definieren eine Hierarchie von Objekten, die mithilfe von Rollenklassen und Systemeinheiten getypt und mittels deren Schnittstellen miteinander verknüpft werden.

**Attribut-Bibliotheken** (ab Version 3, nicht verwendet), die den Attributen in Schnittstellen, Rollen und Systemeinheiten eine Wertsemantik (z. B. Länge in Metern) und eine Verwendungsemantik (z. B. Durchmesser oder Höhe) geben kann. Dies ist eine Neuerung in CAEX Version 3, weswegen es zum Zeitpunkt der Bearbeitung noch keine verwendbaren Bibliotheken gab. Demzufolge wurde nicht auf Attributbibliotheken zurückgegriffen.

### 4.1.1.2 PandIX

Neben den Rollenbibliotheken für die Fertigungstechnik, welche durch AML bereitgestellt werden, waren die P&I Exchange (PandIX)-Bibliotheken vom Lehrstuhl Prozessleittechnik der RWTH Aachen zum Zeitpunkt der Untersuchung die einzigen frei verfügbaren Bibliotheken für CAEX [40]. Sie stellen alle für das Basic Engineering notwendigen Elemente bereit, bei der technischen Ausrüstung jedoch nur eine Teilmenge. Die Bibliotheken wurden in Kooperation mit der PLT/RWTH Aachen und unter Verwendung des im nächsten Abschnitt behandelten Prüfwerkzeuges überarbeitet [3]. Die PandIX-Bibliotheken wurden für alle Arbeiten mit CAE-Daten verwendet. Sie enthalten:

**Schnittstellen** für

**Prozessverbindungen** gemäß IEC 62424 `ProcessConnectionInterface` und Spezialisierungen für Rohranschlüsse, Messgeräteanschlüsse, etc. Für Rohre kann der Nenndurchmesser und die Druckbelastbarkeit angegeben werden.

**Signalverbindungen** zur Darstellung von Regelkreisen und Sicherheitsverschaltungen.

**PCE-Requests** Unter einer gemeinsamen Oberklasse definiert PandIX Messstellen-, Differenzmessstellen-, Aktuator- und Regelfunktionsanforderung. Hierbei können PCE-Kategorie und verschiedene Zusatzkennzeichen angegeben werden.

**Verrohrung** Es werden `RoleClasses` für Rohre, Verbinder und Abzweigungen definiert.

**Apparate/Equipment** Reaktoren, Tanks, Wärmeübertrager, Ventile und Pumpen sind gemäß ISO 10628 festgelegt.

### 4.1.1.3 CAEX-Checker

Das CAEX-XML-Format ist ein Metaformat, welches wenige Möglichkeiten zur Prüfung mit XML-Standardwerkzeugen erlaubt. Anhand der im Standard [17] definierten XML Schema Description (XSD) kann jede CAEX-Datei validiert werden. Diese Validierung

ist ausschließlich eine Syntaxprüfung, die sicherstellt, dass jeder XML-Parser die Datei fehlerfrei und eindeutig einlesen kann (die Datei ist *wohlgeformt* und *gültig*). Insbesondere die unter den Objekten über eindeutige Namen oder eindeutige IDs vorgenommenen Verknüpfungen können anhand der aktuellen Schemadefinition nicht geprüft werden. Auch dateiübergreifende Abhängigkeiten können mit einem solchen Werkzeug generell nicht überprüft werden.

Um mit dem zu entwickelnden Instanziierungswerkzeug auf korrekten Daten arbeiten zu können, wurde ein Prüfwerkzeug entwickelt, welches die übergreifenden Abhängigkeiten und die im Standard informal formulierten syntaktischen und semantischen Vorgaben prüfen kann. Das Werkzeug wurde in [2] erwähnt und in [3] vorgestellt. Zu den prüfbaren Konstrukten sei auf [3] verwiesen, der relevante Abschnitt ist in Anhang A, Abschnitt A.1 ab Seite 169 erneut abgedruckt. Diese Ausführungen entsprechen dem aktuellen Stand.

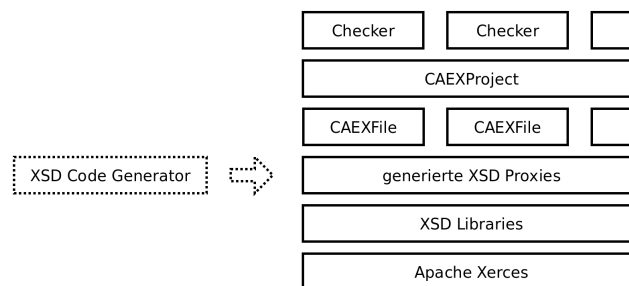


Abbildung 4.1: Architektur des CAEX-Checkers

**Anwendungsarchitektur** Die Architektur des CAEX-Checkers ist in Abbildung 4.1 dargestellt. Er basiert auf automatisch generierten XML-Proxies, die mittels des Tools *CodeSynthesis XSD* Version 3.3.0 [271] aus dem CAEX-XSD-Schema [17] Version 2.15 generiert wurden. Diese basieren auf den mitgelieferten Basisbibliotheken und dem XML-Framework *Apache Xerces* Version 2.8. Darauf wurden eine Abstraktion für einzelne CAEX-Dateien und Projekte, die aus mehreren solchen Dateien bestehen, aufgesetzt. Beide bieten eine generische Schnittstelle zum Zugriff auf die in den Proxies hinterlegten XML-Daten an.

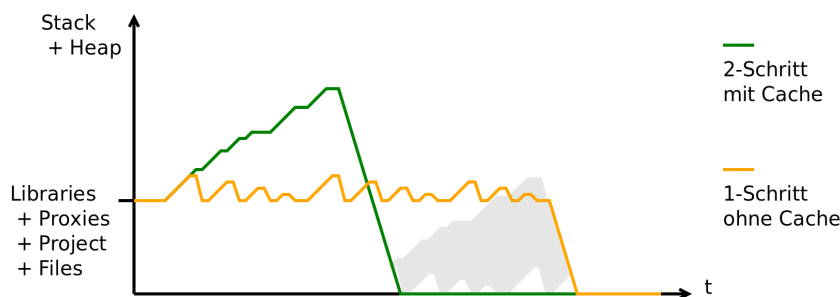
**Implementierungsdetails** Nach der Initialisierung eines Projektes, versucht dieses, die Haupt-CAEX-Datei zu öffnen. Dies wird transparent von der Schnittstelle der XSD- und der Xerces-Bibliotheken übernommen. Danach werden aus diesen Dateien alle direkten und indirekten Referenzen auf andere CAEX-Dateien (*ExternalReference*) aufgelöst und ebenfalls geladen. Die *Xerces*-Bibliotheken übernehmen gleichzeitig eine Validierung der Eingabedateien gegen das CAEX-XSD-Schema.

Die Schnittstelle zwischen Checkern und Projekt bzw. Dateien wurde als Visitormuster implementiert, um das Traversieren im Objektmodell von den Checkern trennen zu können. Die auszuführenden Checker werden an das Projekt und die Dateien übergeben, welche die Checkern mit den zu prüfenden Konstrukten aufrufen. Alle gemäß den angegebenen Kommandozeilenparametern aufzurufende Checker werden nacheinander instanziiert und

der Objektbaum mit ihnen traversiert. Im Rahmen der Arbeit wurden folgende Checker implementiert:

- Elementsuche (nach Pfad)
- Name & UUID Prüfung
- UUID Generierung und Vervollständigung
- Referenzen und Modellintegrität (Version 1, verworfen)
- Objektsammler (Cachemechanismus)
- Referenzen und Modellintegrität (Version 2)

Die letzten beiden Checker implementieren gemeinsam ein Zweischrittverfahren und sind mit Abstand die komplexesten Module der Anwendung. Um die Wiederholung ständig gleicher Suchvorgänge nach Objekten und Objekthierarchien zu vermeiden, wurde ein Cachemechanismus entworfen, der alle verwendeten *Modellaggregationen* zwischenspeichert. Somit werden doppelte Fehlerausgaben vermieden und die Laufzeit erheblich verkürzt. Der Speicherbedarf ist dadurch in der Spitze höher, jedoch in der Zeitnutzung geringer. Abbildung 4.2 verdeutlicht dies qualitativ. Die geringere Laufzeit ergibt sich durch die Einsparung von redundanten Prüfungen. Der erhöhte Speicher folgt aus dem längeren Halten von kombinierten Elementen.



**Abbildung 4.2:** Qualitative Speichernutzung der verschiedenen Implementierungen der Referenz- und Integritäts-Checker (bei gleicher verwendeter Prozessorleistung)

**Caching von Modellaggregationen** Alle Teilmodelle lassen sich, abgesehen von ihren Namen und Typen, auf ihre Attribute bzw. ihre Attributstruktur zurückführen. Unabhängig davon, ob ein Element einer Hierarchie *InternalElements*, Systemklassen, Rollenklassen, Schnittstellen, Schnittstellenklassen oder Attribute sind, sind die Namen der Hierarchieelemente einzigartig und die Attributzuordnung eindeutig. Der Cache verwirft alle Typinformationen und speichert ausschließlich ein Mapping von relativen, hierarchischen Namen auf *AttributCache*-Objekte. Um wieder auf die Originalobjekte zugreifen zu können, speichern die Cache-Elemente alle Referenzen auf die Proxies, aus denen sie generiert wurden. Auch wird jede Cache-Stufe behalten, um Cache-Ketten von wiederverwendbaren Typen nicht mehrfach berechnen zu müssen.

Abbildung 4.3 zeigt ein Beispiel einer Cachestruktur für eine exemplarische CAEX-Proxystruktur. Die dunklen Kästen mit den breit gezeichneten Beziehungen sind die Proxies, die hellen Kästen mit den schmalen Referenzen die Cache-Elemente. Der Übersichtlichkeit halber wurde auf die Verwendung einer Schnittstellenklasse als Typ für IF1 verzichtet, genauso wie beim Übergang von Rollenklasse zur Systemklasse gäbe es keine neue Hierarchieebene, jedoch ein zusätzliches, oft wiederverwendbares Cache-Element. Die Cache-Elemente sind in als Container der *C++ Standard Template Library* abgelegt.

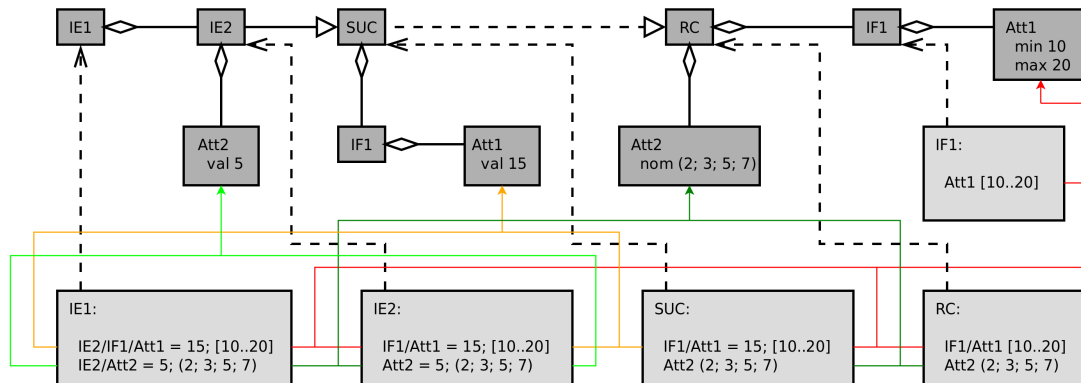


Abbildung 4.3: Einfaches Beispiel einer Cache-Struktur

#### 4.1.1.4 Dia2CAEX

Die Arbeiten zur schnellen Generierung von CAEX-Exporten von R&I-Fließbildern wurde online frei zugänglich veröffentlicht [4]. Auf frühere Versionen wurde bereits in [3] verwiesen. Die R&I-Fließbilder wurden in *Dia* [272] gezeichnet (siehe Abbildung 4.4 auf Seite 62).

Dafür wurden die mitgelieferten Symbole der *ChemEng*-Bibliothek um eine weitere Bibliothek mit Symbolen für PCE-Requests, Ventile, Pumpen, Rohrleitungsverbindern und Reaktoren erweitert. Dabei wurden die Symbole für Equipment gemäß der ISO 10628 [27, 28] verwendet. Die PCE-Requests sind nach der IEC 62424 [17] dargestellt, eine kleine Erweiterung zur Unterscheidung von Sensoren und Aktuatoren wurde angefügt. Für Differenzsensoren wurde ein zusätzliches Symbol erstellt. Weiterhin gibt es Symbole für Bibliotheksnamen, um die *ExternalReference* beim Export sofort mit einbinden zu können. Dies sollten die *PandIX*-Bibliotheken [40] in der aktuellen Version [3] sein. Um in einem großen R&I-Fließbild mehrere Teilanlagen abgrenzen zu können und diese in der Instanzhierarchie auch separieren zu können wurde *Dia* befähigt, Gruppen in Gruppen erstellen zu können, ohne die Verbinder von den Objekten zu lösen [4]. Auch ist zum expliziten Erstellen von Schnittstellen von solchen Gruppen ein nicht standardkonformes Symbol eingeführt worden. An ihm kann man auch eine Schnittstellenklasse angeben. Rohrleitungen werden mit durchgezogenen Polylinien (Polygonzügen) gezeichnet, Signalverbindungen mit gestrichelten (*dashed*) Polylinien. An jeden Zwischenpunkt der Rohrlinien kann eine Messstelle, jedoch kein anderes Rohr angedockt werden. Für Verzweigungen müssen die entsprechenden Symbole aus der Bibliothek verwendet werden. Alle Elemente, die Anlagenteile repräsentieren, müssen in hierarchisch geschachtelten Gruppen zusammengefasst sein. Es darf nur eine Top-Level-Gruppe geben.

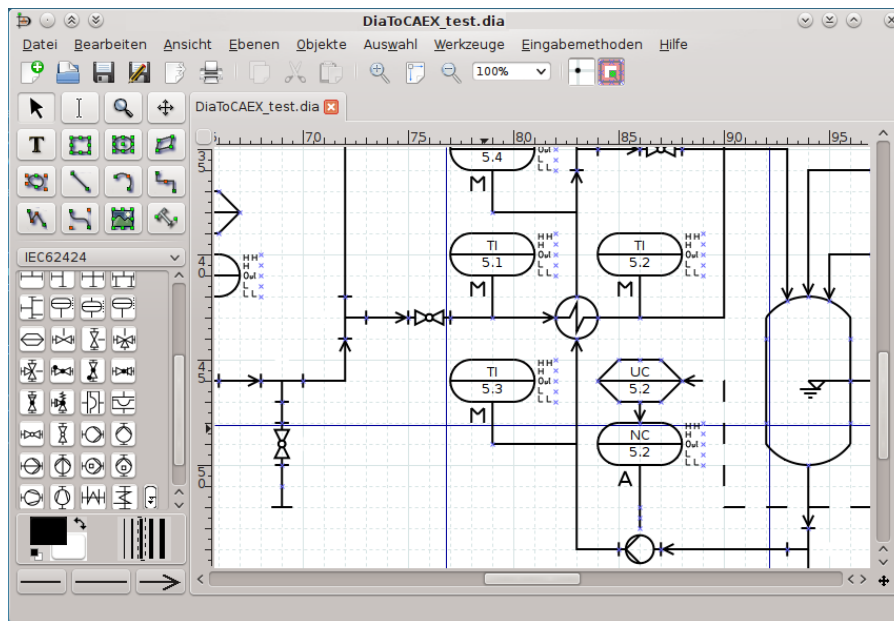


Abbildung 4.4: Screenshot von Dia mit einem nach CAEX umwandelbaren Diagramm (aus [4])

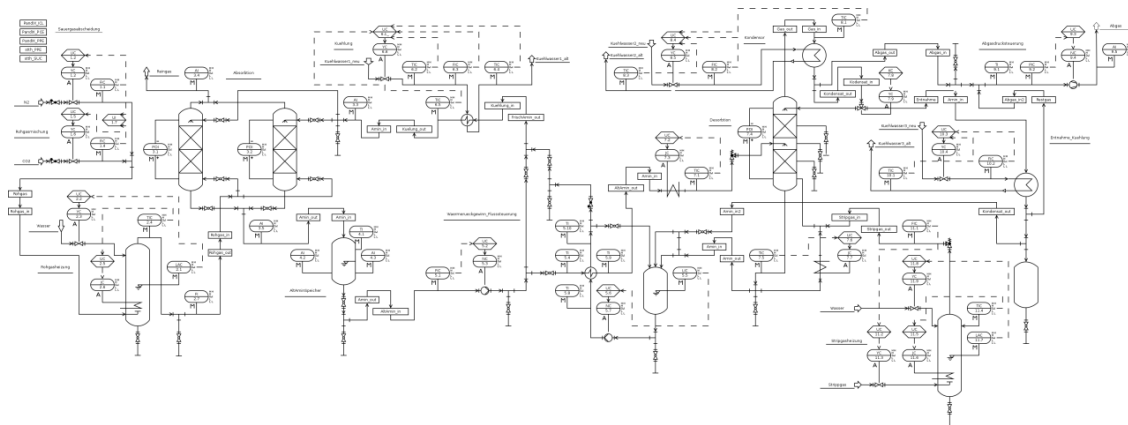
All diese Änderungen vereinfachen die Transformation der *Dia*-Datei, welche ein mit *GZIP* gepacktes XML-Format ist. Die Umwandlung geschieht mittels einer *XSL*-Transformation [273]. Diese Transformation erstellt eine CAEX-Datei, bindet die Bibliotheken aus den ExternalReference-Blöcken ein, erstellt eine Instanzhierarchie und sucht nach Inhalten für diese. Dafür wird in der XML-Hierarchie in die Gruppen von Objekten abgestiegen und alle bekannten Objekttypen umgewandelt. Dabei ist für jeden Typ ein separater Satz von Regeln erforderlich, um die Rollennamen zuzuordnen und die in *Dia* fest durchnummerierten Objektverbinder korrekt mit den richtigen, angedockten Rohrleitungen mittels InternalLinks zu verlinken. Der Quelltext B.1 in Anhang B, Abschnitt B.1 ab Seite 177 zeigt den Regelsatz für einen Wärmetauscher. Die darauf folgenden Listings sind beispielhafte Ein- und Ausgabedaten der Transformation.

Als Beispielanlage wurde die Versuchs-Gaswaschanlage des Instituts für Verfahrenstechnik und Umwelttechnik der TU Dresden mittels *Dia* modelliert. Abbildung 4.5 zeigt einen Überblick, die detaillierte Abbildung befindet sich in Anhang C, Abschnitt C.1 ab Seite 185.

#### 4.1.2 FDI Server Emulation

Zum Zugriff auf Daten der von Apparatemodellen verwendeten Feldgeräte soll im Konzept ein FDI Server mit OPC UA-Schnittstelle verwendet werden. Da die Implementierung der Schnittstelle nicht absehbar war [154, 156], wurde ein emulierter FDI Server verwendet. Bei einem Übergang auf eine FDI Server-Implementierung mit der benötigten Schnittstelle soll die in Kapitel 3, Abschnitt 3.6 ab Seite 48 beschriebene Migrationsstrategie angewendet werden.

Die Emulation des FDI Servers soll gleichzeitig den Zugriff auf die Daten der Feldgeräte vereinfachen. Die in Abschnitt 4.1.1.4 ab Seite 61 modellierte Beispielanlage ist als



**Abbildung 4.5:** Überblick über die in *Dia* modellierte Gaswaschanlage (aus [4] auf Basis von [274])

Laboranlage automatisiert und besitzt keine in Großanlagen üblichen Feldgeräte, sondern nur einfache Sensoren. Statt eine Schnittstelle zu implementieren, wurden die Werte aus Zeitreihen in die Gerätemodelle importiert. Mithilfe eines Gerätemodells wird das Objektmodell eines FDI Server einer mit intelligenten Feldgeräten ausgestatteten Anlage mit dynamischen Prozessdaten emuliert.

#### 4.1.2.1 FDI OPC UA Informationsmodell

Auf Basis eines OPC UA-Servers und eines OPC UA-Modellierungswerkzeug wurde in einer studentischen Arbeit [11] das Informationsmodell für den FDI Server implementiert. Das *C++ Server Software Development Kit (SDK)* und der *UAModeler* von *Unified Automation* unterstützen das Informationsmodell OPC UA-DI [190] nativ. Aus diesem Grund konnten viele Modellelemente weiter verwendete werden. Neben kleineren Fehlern im mitgelieferten Modell bereitete vor allem die vorgesehene Werkzeugkette Schwierigkeiten. Im Zeitraum der Bearbeitung konnte das Modell nur in C++-Quelltext exportiert werden. Die Randbedingungen des SDK ließ keine dynamische Erstellung von Instanzen auf Basis von im *UAModeler* beschriebenen Typen zu [11]. Demzufolge müssten alle Instanzen im Modellierungswerkzeug per Hand erstellt werden, was nicht nur große Mengen redundanten Quelltexts sondern auch erheblichen Aufwand bedeutet. Es wurde jedoch bereits darauf hingewiesen, dass ein Feature zum Laden eines Informationsmodells aus einer XML-Beschreibung geplant war. In anderen Servern ist dieses Feature bereits vorhanden gewesen [68, 69]. Die Vorteile durch das mitgelieferte OPC UA-DI-Modell und die Möglichkeit der Verwendung eines grafischen Modellierungswerkzeugs überwogen jedoch diesen Nachteil.

#### 4.1.2.2 OPC UA Client

Eine weitere Anforderung war der wiederholte Zugriff auf Nodes in einem OPC UA-Informationsmodell. Die Nodes mit den Prozessdaten der Feldgeräte müssen nicht nur anhand der *NodeID* im OPC UA-Server sondern auch anhand des *Browsepaths* wieder erkannt werden. Hierfür wurde als Fortführung der oben genannten studentischen Arbeit [11] ein Client-API auf Basis der *C++ Client SDK* von *Unified Automation* implementiert.

Diese legt die Datenpunkte anhand des *Browsepaths* ab und kann auch bei dynamisch vergebenen *NodeIDs* die richtigen Variablen im OPC UA-Server wieder finden.

### 4.1.2.3 FDI OPC UA Server

Aufbauend auf die vorgenannten Arbeiten wurde in einer weiteren studentischen Arbeit [12] ein emulierter FDI Server entwickelt. Dieser ist ein OPC UA-Aggregationsserver gemäß der Spezifikation der *OPC Foundation* [191]. Da ein FDI Server kein Aggregationsserver ist, kann jedoch auf das zusätzliche Informationsmodell verzichtet werden.

Die größte Herausforderung ist der Umgang mit gleichen Typen in den unterschiedlichen aggregierten Servern. Die FDI-Spezifikation legt *NodeIDs* für die im Standard spezifizierten Typen fest. Jedoch werden auch durch FDI Packages weitere Typen eingeführt, die nicht vorherzusehen sind und je nach angeschlossenen Geräten unterschiedliche *NodeIDs* erhalten. Demzufolge muss der Aggregationsserver gleiche Typen erkennen und bei der Aggregation der Informationsmodelle Referenzen vom Typ `hasSubtype` und `hasTypeDefinition` manipulieren. In [12] wird dies über ein *Namensraum-Mapping* gelöst, welches beim Übertragen eines aus den zu aggregierenden Servern ausgelesenen Zwischenmodell in das aggregierte Informationsmodell angewendet wird. [262] löst diese Aufgabe nicht nur über Regeln sondern auch über den Vergleich von konstanten Properties. Die gemäß [11] in den unterlagerten OPC UA-Servern zur Verfügung gestellten Daten werden über die Clientkomponente im Aggregationsserver aus den unterlagerten Servern gelesen, wenn auf die entsprechenden aggregierten Nodes zugegriffen wird. Ein Schreibzugriff funktioniert genauso, ist aber im Anwendungsfall nicht vorgesehen. Somit ersetzt im emulierten FDI Server der Aggregationsmechanismus die Hierarchie von FDI Server und FDI Communication Servern. Diese Abweichung vom Standard wird mit der Umsetzung der Migrationsstrategie wieder zurückgenommen.

### 4.1.3 Feldgeräte-Informationsmodell

In [11] und [12] wurde zum Proof-of-Concept zum FDI-Informationsmodell ein einzelnes PROFIBUS-PA-Gerät modelliert. Es handelt sich um einen Temperaturmessumformer gemäß PA-Profil [116].

Darauf aufbauend wurden alle im Anlagenbeispiel (Abschnitt 4.1.1.4 ab Seite 61) verwendeten Gerätetypen gemäß PA-Profil modelliert und in den *Unified Automation* OPC UA-Server integriert.

#### 4.1.3.1 Informationsmodell

Das vorgegebene Informationsmodell wurde weiter aufgeteilt. Auch wurden zusätzlich alle optionalen Modellelemente angelegt. Die konkreten Gerätetypen, wie sie in einem FDI Server aus den FDI Packages geladen werden, wurden in einem separaten Namensraum gespeichert. Somit ergibt sich im finalen Informationsmodell folgende Namensraum-Struktur:

- 1) OPC UA-Basisinformationsmodell (von SDK und Modellierungswerkzeug bereitgestellt)



- 2) OPC UA-DI-Informationsmodell (von SDK und Modellierungswerkzeug bereitgestellt)
- 3) FDI-Informationsmodell ([11])
- 4) PA-FB-Typen
- 5) PA-Gerätetypen
- 6) Geräteinstanzen

In Abbildung 4.6a ist ein Teil der für den Namensraum 4) entworfenen Blocktypen abgebildet. Abbildung 4.6b zeigt die Gerätetypen aus Namensraum 5), die diese Blöcke verwenden. Von diesen Gerätetypen wurden im Namensraum 6) die Instanzen für das Anlagenbeispiel abgeleitet (siehe Abbildung 4.6c).



Abbildung 4.6: Struktur des Geräte-Informationsmodells (zur Übersichtlichkeit wurden Nodes entfernt)

#### 4.1.3.2 Integration als NodeSet-XML

Die Vorteile durch die Instanziierung eines NodeSets im OPC UA-Server, die in [11] und [12] heraus gearbeitet wurden, konnten ab Q4 2014 genutzt werden. Ab Version 1.4 des

*C++ Server SDK* von *Unified Automation* ist diese Funktion verfügbar. Somit konnten die erstellten Informationsmodelle ohne Programmieraufwand in den Server geladen werden. Diese werden aus dem Modellierungswerkzeug als je eine XML-Datei je Namensraum exportiert. Die XML-Dateien folgen der zweiten Version des NodeSet-XSD der *OPC Foundation* mit dem Namensraum

<http://opcfoundation.org/UA/schemas/1.02/UANodeSet.xsd>.

Sie verwenden die Nodes aus den Spezifikationen gemäß des offiziellen Basis-NodeSet Namensraums

<http://opcfoundation.org/UA/schemas/1.02/Opc.Ua.NodeSet2.xml>.

Somit könnten diese Exportdateien auch anderweitig eingesetzt werden.

Das *C++ Server SDK* enthält ein Beispiel für die Integration von beliebigen XML-NodeSets. Es enthält zwei zusätzliche Header und zwei zusätzliche Quelltextdateien mit sieben aus dem SDK abgeleiteten Klassen einschließlich deren Implementierung. Unter ihnen befindet sich ein *NodeManager*. Für jedes der NodeSets wird beim Start des Servers ein solcher *NodeManager* erzeugt, und ihm die XML-Datei übergeben.

### 4.1.3.3 Anbindung der Daten im dynamischen Informationsmodell

Um in das durch XML beschriebene NodeSet dynamisch Daten einbinden zu können, müssen zuerst die entsprechenden Nodes in der Datei mit einer *Extension* erweitert werden. Zum Auslesen der einzuspeisenden Messdaten aus *CSV*-Dateien muss der Spaltenname bekannt sein. Dieser wird im Beispiel in Quelltext 4.1 in Zeile 12 festgelegt.

**Quelltext 4.1:** Variable-Element eines NodeSets mit einer Extension zum Zuweisen von Messdaten

```
<UAVariable DataType="Float" ParentNodeId="ns=1;i=5001"
  NodeId="ns=1;i=6002" BrowseName="1:Value" UserAccessLevel="3" AccessLevel="3">
  <DisplayName>Value</DisplayName>
  <References>
    <Reference ReferenceType="HasTypeDefinition">i=63</Reference>
    <Reference ReferenceType="HasComponent"
      IsForward="false">ns=1;i=5001</Reference>
  </References>
  <Extensions>
    <Extension>
      <CHAName>
        <ColumnName>T11</ColumnName>
      </CHAName>
    </Extension>
  </Extensions>
</UAVariable>
```

Damit die *Extension* ausgewertet wird, muss der *NodeManager*, der das Informationsmodell aus der XML-Beschreibung instanziiert, verändert werden. Anhand des Beispiels wurde der *NodeManager* wie in Abbildung 4.7 adaptiert und umgesetzt.

## 4.2 Datenmodell für Apparatepakete

Wie in Abschnitt 3.3.1 ab Seite 43 beschrieben, sollen die Apparatepakete als FDI Package ausgeliefert werden. Dafür müssen ihre Struktur und ihre Berechnungsmodelle in einer EDD-Datei wiedergegeben werden. Zur einfacheren Integration wird die Struktur des Apparates ebenfalls mit einer CAEX-SystemUnitClass dargestellt.

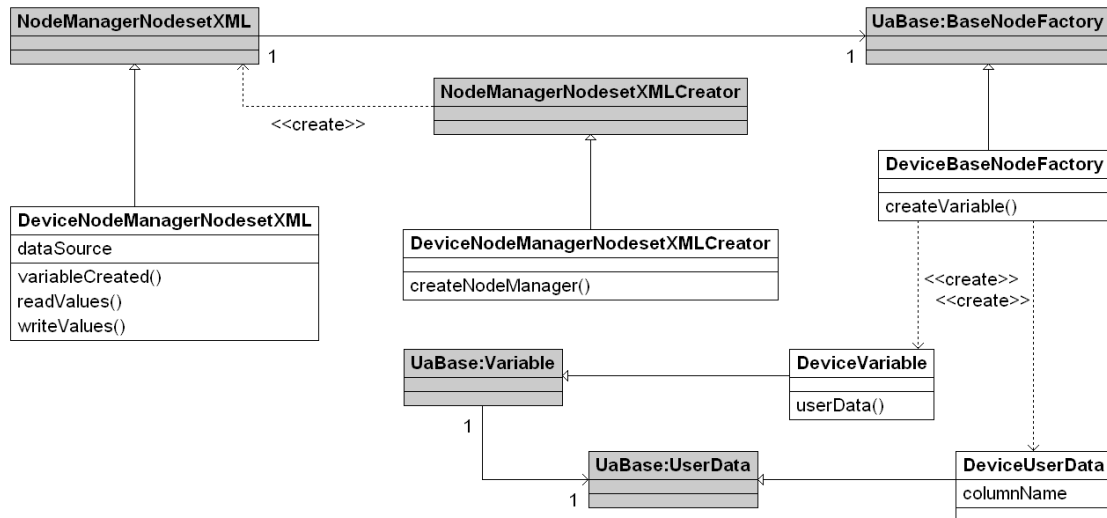


Abbildung 4.7: Klassendiagramm mit den Klassen der *Extension* zum Einbinden von Messdaten

Zur einfacheren Verständlichkeit sei auf folgende Nomenklatur bei *Parametern* und *Variablen* hingewiesen: Das EDDL-Konstrukt heißt *Variable*. Die OPC UA-Nodeklasse wird als *Variable* bezeichnet. Konsistent zu [116] wird eine *Variable* eines Gerätemodells (auch in seiner EDD-Darstellung) als *Parameter* bezeichnet. Das Modellelement eines Apparatmodells heißt ebenfalls *Parameter*.

Abbildung 4.8 zeigt das abstrakte Modell für die Apparatmodelle. Ein Apparat kann viele Anschlüsse haben. An diesen Anschlüssen sind beliebig viele Messstellen(-anforderungen) vorhanden. Sowohl auf der Ebene eines Anschlusses als auch global (auf Ebene des Apparates) können Parameter und Funktionen hinzugefügt werden. Die in den folgenden Abschnitten beschriebenen Zuordnungen zu Blöcken wurde in [1, Fig. 4, S. 5] beschrieben.

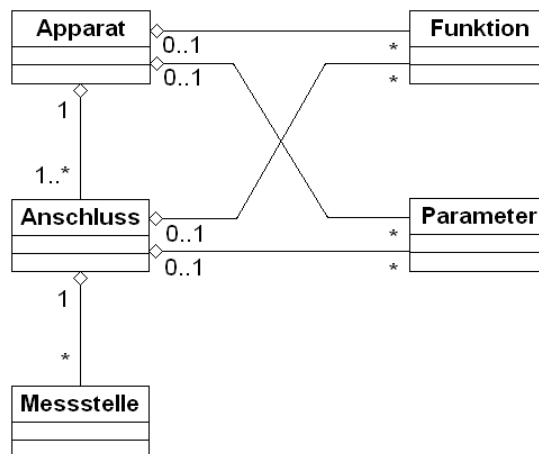


Abbildung 4.8: Struktur eines Apparatmodells

Die Struktur des Apparatmodells spiegelt sich sowohl in der CAEX-SystemUnitClass als auch in der EDD im FDI Apparatpackage wieder. In der SystemUnitClass sind neben

dem Apparat, welcher das Hauptelement ist, noch jeder Anschluss mit seinem Namen und seiner Schnittstellenklasse, die den Typ des Anschlusses festlegt, enthalten (z. B. für eine Rohrleitung `ProductConnectionPoint`). Quelltext 4.2 zeigt die wichtigsten Aspekte dieser Datei. Eine vollständige Version ist in Anhang B ab Seite 177, Abschnitt B.2 ab Seite 181 abgedruckt. Wichtig sind insbesondere die Namen der `ExternalInterfaces`, da sie später für die Instanziierung benötigt werden.

**Quelltext 4.2:** Gekürztes Beispiel für eine exportierte `SystemUnitClass` (neuere Version des Beispiels aus [2])

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<CAEXFile FileName="SystemUnitLib_HE_example.aml" SchemaVersion="2.15"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd">
[... Header]
  <ExternalReference Alias="PandIX_ICL" Path="PandIX_ICL_V600.xml"/>
[... (3 weitere Referenzen)]
  <SystemUnitClassLib ID="2111c4a1-b2f4-41f3-a231-c1bd7679e4c1"
    Name="SystemUnitLib_HE_example">
    <SystemUnitClass ID="3606689b-2e2e-4a20-b0d0-b9f4b2af780a"
      Name="HE_example">
      <Attribute Name="ISO10628Type" AttributeDataType="xs:string" >
        <Value>X2501</Value>
      </Attribute>
[... (4 weitere Attribute)]
      <ExternalInterface Name="PIn"
        RefBaseClassPath="PandIX_ICL@PandIX_ICL/ProductConnectionPoint">
        <Attribute Name="DN" AttributeDataType="xs:double" Unit="mm">
          <Value>32</Value>
        </Attribute>
[... (1 weiteres Attribut)]
        </ExternalInterface>
[... (3 weitere ExternalInterfaces mit 2 Attributen)]
      <SupportedRoleClass
        RefRoleClassPath="PandIX_PPE@PPE_BasicLib/HeatExchangerRequest"/>
    </SystemUnitClass>
  </SystemUnitClassLib>
</CAEXFile>
```

Um eine einfache Integration und gleichartige Handhabung der Apparatepakete im EIH (Abschnitt 4.4 ab Seite 86) zu ermöglichen, wird eine einheitliche Struktur und Funktion der Pakete vorgeschrieben. Diese ermöglichen einen einheitlichen Zugriff auf Integrationsdaten (CAEX-`SystemUnitClass`) und das zu integrierende FDI-Informationsmodell. Bei der Modellierung wurde auf die Struktur des PROFIBUS-PA-Profiles [116] zurückgegriffen. Es hat einerseits die flexibelste Struktur der unterstützten Technologien und andererseits ist es weit genug verbreitet [265, F. 19], sodass eine einfache oder sogar automatische Integration der Ergebnisse der Modelle in ein PAM- oder CM-System wahrscheinlich ist. Durch die übliche Aufteilung in mehrere FB ist das Modell leicht strukturierbar.

Die maximale Anzahl der in einem PA-Feldgerät vorhanden Blöcken ist durch die Größe des *Device Directory* und der maximalen Telegrammgröße begrenzt. Die im *Device Directory* enthaltenen Blockbeschreibungen müssen in acht Indizes passen [116, Abs. 9.4.3, S. 122]. Eine Blockreferenz besteht aus zwei `Unsigned16`-Werten (zusammen 4 Byte). Die maximal in einem Telegramm übertragbaren Anzahl von Blockreferenzen berechnet sich aus  $244 \text{ Byte} \div (2 \cdot 2 \text{ Byte}) = 61$ . Somit ergeben sich maximal  $8 \cdot 61 = 488$  Blöcke. Die Blöcke jedes der vier FB-Typen werden mittels der `TR_ID` von 1 bis 255 durchnummeriert. Dies beschränkt die Anzahl von Blöcken weiter, die Obergrenze erscheint aber

aufgrund der Struktur der Apparatemodelle nicht erreichbar. Bei der vollen Ausnutzung der maximalen Anzahl von Blöcken entfallen aus dem Adressbereich des Gerätes  $[253 \cdot 255 \div 488] = 132$  Indizes auf einen Block. Diese Anzahl stellt ebenfalls keine praktisch erreichbare Begrenzung dar. So kann die Verwendung des noch größeren PN IO-Adressraums vermieden werden. Eine weitere Modellalternative, PROFINET für PA, ist erst in der Spezifikationsphase [265, F. 29]. Deswegen wäre mit diesem Modell die Integration der durch die Berechnungsmodelle erstellten Kennzahlen wahrscheinlich schwieriger.

Da genügend Blöcke adressiert werden können wurde für die Abbildung der Modellstruktur in FB in der EDD folgende Festlegung getroffen: Jede Messstelle erhält einen separaten *Transducer Block*; jeder Anschluss erhält einen separaten *Function Block*; der Apparat erhält einen separaten *Function Block* zur Strukturierung seiner Parameter. Weiterhin sollte die EDD Interoperability Guideline [275] der PNO beachtet werden.

Um Funktionen von Geräten partiell zu verwenden, ist in der Standardisierung und in Applikationsprofilen die Aufteilung in *mandatory* und *optional* üblich. Dieser Mechanismus ist jedoch für die im Umfang der Instrumentierung zu unsicheren Abdeckung der Anforderungen zu unflexibel. Die in [1] vorgestellte Berechnungsstruktur wurde in [2] verfeinert, sodass jeder Wert optional sein kann. Dies wird in der EDD über je einen zweiten Parameter realisiert, der abspeichert, ob der Wert tatsächlich verfügbar ist. Alle Berechnungen und Werte, die auf diesem Wert basieren, werden mittels des *Validity*-Konstrukts ausgeblendet, wenn der Wert nicht verfügbar ist. Quelltext 4.3 zeigt die Definition eines Parameters, der auf diese Weise ausgeblendet werden kann. In Quelltext 4.4 wird gezeigt, wie die zugehörigen Berechnungen ebenfalls ausgelassen werden können. Die *storage class* des *\*\_IS\_VALID*-Parameters ist *LOCAL* und sein *HANDLING* nur-lesend. Das heißt, dass sein Wert im FDI Server abgespeichert wird und vom Nutzer eines FDI Clients oder aus der BL heraus nicht geändert werden kann. Der Wert wird bei der Instanziierung des Modells gesetzt, wenn die Verfügbarkeit des Parameters, z. B. der Messwert einer Messstelle geprüft und ihre Zuordnung vorgenommen wird.

**Quelltext 4.3:** EDD-Definition eines ausblendbaren EDD-Parameters (aus [2])

```
[...]
VARIABLE FpVTB__OUT_Value {
  LABEL [FpVTB__OUT_Value_lbl];
  CLASS CONTAINED & DYNAMIC;
  TYPE FLOAT;
  HANDLING READ;
  VALIDITY IF (FpVTB__IS_VALID == 1) {TRUE;} ELSE {FALSE;}
}
[...]
VARIABLE FpVTB__IS_VALID {
  LABEL [FpVTB__IS_VALID_lbl];
  CLASS LOCAL;
  TYPE ENUMERATED (1) {
    DEFAULT_VALUE 1;
    {0, [var_not_valid]},
    {1, [var_valid]}
  }
  HANDLING READ;
}
[...]
```

**Quelltext 4.4:** EDD-Code zur Verwendung eines ausblendbaren EDD-Parameters (erweitert aus [2])

```
[...]
```

```
if (FptVTB__IS_VALID == 1)
{
  if (FptVTB__OUT_STATUS == 0x80)
  {
    [...] = [...] FpVTB__OUT_Value [...];
  }
  else
  [...]
}
[...]
```

Im Folgenden wird auf die Strukturierung und die Parameter der Messstellenanforderungen, der Parameter, des Berechnungsmodells und der Menüs eingegangen. Es wird zudem eine in der Business Logic implementierte Variante zur Sicherstellung einer minimalen Lese-Zykluszeit beschrieben, die eine zu hohe Belastung der Feldbusse und Feldgeräte durch die Berechnungsmodelle verhindern soll.

### 4.2.1 Messstellen

Wie bereits im Modell in Abbildung 4.8 auf Seite 67 dargestellt werden die Messstellenanforderungen an einen bestimmten Anschluss des Apparates gebunden. Eine genauere Festlegung des Ortes ist nicht möglich und auch nicht praktikabel. Wie auch in [211] beschrieben, werden für bestimmte Messstellentypen Umgebungsinformationen benötigt, um für den Apparat gültige Werte berechnen zu können (z. B. Druckmessgeräte). Diese können im Modell mittels Parametern und Funktionen realisiert werden.

Im Engineering werden für Messstellen die folgenden Informationen benötigt:

**Name** (eindeutig)

**Funktionskennzeichen** bzw. PCE-Kategorie. In der IEC 62424 [17, 37] sind die Bedeutungen von Buchstaben standardisiert. Hierbei steht beispielsweise das *F* für eine Durchflussmessung.

**Maßeinheit** Bestimmte PCE-Kategorien grenzen die mögliche Funktion ein, sind aber nicht eindeutig. Im Beispiel einer Durchflussmessung ist durch das Funktionskennzeichen nicht festgelegt, ob ein Masse- oder Volumenstrom gemessen werden soll. In der vorliegenden Arbeit wird diese Eingrenzung über die Maßeinheit vorgenommen. Sie ist bei der Unterscheidung eindeutig, schließt aber keine neuen Messverfahren aus. Zusätzlich ermöglicht dies eine spätere Prüfung der Funktionen.

Mit der Zuordnung zu einem Anschluss des Apparates können mit den aus den späteren Messgeräten zu lesenden Werten Berechnungen angestellt und Kennzahlen bestimmt werden.

Bei der Instanziierung einer Messstelle müssen weitere Informationen zur Messstelle angelegt werden.

**Validity** speichert ab, ob die Anforderung mit einem in der Anlage vorhandenen und mittels FDI Server zugreifbaren Feldgerät erfüllt werden kann.

**Referenzkennzeichnung.** Eine zumeist unternehmensweit eindeutige Betriebsmittelkennzeichnung [276, 29] dieses Gerätes.

**Messstellenbezeichnung**, üblicherweise auch als TAG, TAG\_ID oder TAG\_DESC im Feldgerät gespeichert.

**Gerätename im FDI Server.** Dieser Name könnte auch mit der Referenzkennzeichnung oder der Messstellenbezeichnung übereinstimmen. Dies kann aber nicht garantiert werden. Der *BrowsePath* zu dieser Node im Informationsmodell des FDI Server ist, wie in Abbildung 4.6c auf Seite 65 ersichtlich, standardisiert:  
 <root>/Objects/DeviceSet/<Gerätename>

**BrowsePath zum Parameter** mit dem gesuchten Wert. Dieser Parameter ist nur verfügbar, wenn das FDI Package des Gerätes im FDI Server geladen ist [262]. Da sich nicht bei jedem Laden die Verwendung derselben *NodeID* für dieselben Nodes erzwingen lässt, ist dies die einzige Möglichkeit, dauerhaften Zugriff zu ermöglichen.

Diese Daten sollten im Informationsmodell des instanziierten Apparatmodells ebenfalls abgelegt werden können.

Alle Informationen zu einer Messstelle bzw. Messstellenanforderung werden in einem *Transducer Block* zusammengefasst. Dabei werden zu den PA-Standardparametern, die in jeden Block gehören [116], die in Tabelle 4.1 auf Seite 72 aufgezählten Parameter aufgenommen. Alle Parameternamen haben noch einen Präfix, der die FB-Zuordnung und den Namen der Messstellenanforderung im Modell widerspiegelt (z. B. F1TB\_).

#### 4.2.2 Parameter

Parameter können zur Einstellung des Modells statische Apparat-, Umgebungs- oder Prozesseigenschaften speichern [217, Abs. 4, S. 6]. Sie können im Modell zu einem Anschluss oder zum Apparat gehören. Dadurch wird eine Zuordnung in die verschiedenen Funktionsblöcke realisiert. Zu jedem Parameter wird im Engineering des Apparatmodells folgende Eigenschaften festgelegt:

**Name** (eindeutig)

**Maßeinheit** um die Semantik des Parameters zu verdeutlichen. Zusätzlich ermöglicht dies eine spätere Prüfung der Funktionen.

Bei der Instanziierung des Modells muss neben dem Wert des EDD-Parameters auch dessen Verfügbarkeit gespeichert werden.

**Validity** speichert ab, ob der Wert des Parameters gesetzt wurde und einen verwendbaren Wert enthält.

Daraus ergibt sich die Liste von Parametern in der EDD-Datei in Tabelle 4.2 auf Seite 73. Diese Parameter gehören zu dem FB zu dessen Anschluss oder Apparat der Parameter zugeordnet ist. Der Präfix ergibt sich aus dem Namen und der Zuordnung.

#### 4.2.3 Berechnungsmodell

In der prototypischen Umsetzung werden nur einfache arithmetische Berechnungen betrachtet, die mit den aktuellen Werten aus den Geräten und eventuell festgelegten Parametern

Tabelle 4.1: EDD-Parameter im Transducer Block für eine Messstellenanforderung

Name	Typ	Speicherklasse	Beschreibung
(TAG_DESC)	VisibleString	CONTAINED	(unter den Standardparametern definiert)
REF_DESC	VisibleString	CONTAINED	Referenzkennzeichnung
OUT_Value	FLOAT	CONTAINED & DYNAMIC	Messwert
OUT_Status	Unsigned8	CONTAINED & DYNAMIC	Status des Messwertes
OUT_UNIT	ENUMERATED(2) (Unsigned16)	LOCAL	PA-Unit-Code der Maßeinheit des Messwertes
NAME	VisibleString	LOCAL	Name im Modell
ANSCHLUSS	VisibleString	LOCAL	Anschlussname im Modell
IS_VALID	ENUMERATED	LOCAL	Verfügbarkeit des Messwertes
FDI_DEV_NODE	VisibleString	CONTAINED	Name der Geräte-Node im FDI Server
FDI_VAR_PATH	VisibleString	CONTAINED	relativer Pfad von der Geräte-Node zur OUT-Variablen im FDI Server

durchgeführt werden können. Die Funktionen können im Modell (Abbildung 4.8 auf Seite 67) einem Anschluss oder dem Apparat zugeordnet werden. Es kann die gesamte Mächtigkeit der in EDDL für *Methoden* definierten, C-ähnlichen Scriptsprache verwendet werden. Zum Ende muss ein Wert bzw. ein Funktionsergebnis zugewiesen werden. Dieser Wert kann in anderen Berechnungen weiter verwendet werden. Ergebnisse von Berechnungen mit zyklischen Abhängigkeiten sind von der Ausgabereihenfolge der Funktionen abhängig und verwenden Eingangsdaten aus dem vorherigen Datensatz. Sie sind deswegen als undefiniertes Verhalten zu betrachten. Bei der Definition einer Funktion im Apparatmodell müssen folgende Angaben gemacht werden:

**Name** (eindeutig) er ist gleichzeitig der Name des EDD-Parameters, der das Funktionsergebnis speichert und in weiteren Berechnungen weiterverwendet werden kann.

**Maßeinheit** um eine erste Prüfung der Berechnung vornehmen zu können und später das Ergebnis interpretieren zu können, ist eine Maßeinheit erforderlich.

**KPI-Option** Es kann markiert werden, ob ein Berechnungsergebnis eine Kennzahl des Modells ist.



**Tabelle 4.2:** EDD-Parameter für einen Modell-Parameter

Name	Typ	Speicherklasse	Beschreibung
VALUE_Value	FLOAT	LOCAL	Modellparameter
VALUE_Status	Unsigned8	LOCAL	Status des Modellparameters
VALUE_UNIT	ENUMERATED(2) (Unsigned16)	LOCAL	PA-Unit-Code der Maßeinheit des Modellparameters
NAME	VisibleString	LOCAL	Name im Modell
ANSCHLUSS	VisibleString	LOCAL	Anschlussname im Modell (optional)
IS_VALID	ENUMERATED	LOCAL	Verfügbarkeit des Modellparameters

**Berechnungsvorschrift** es ist anzugeben, wie das Funktionsergebnis bestimmt wird. Dabei können bei Funktionen, die einem Anschluss zugewiesen sind, nur Messwerte, Parameter und Ergebnisse anderer dort definierter Funktionen als Eingangswerte verwendet werden (Kapselung). Die Funktionen des Apparates können auf alle anderen Werte zugreifen.

Bei der Instanziierung sind keine Angaben weiter erforderlich. In der EDD werden für jede Funktion die in Tabelle 4.3 genannten Parameter definiert. Auch hier wird mit einem Präfix die FB-Zuordnung und der Name hinzugefügt.

**Tabelle 4.3:** EDD-Parameter für eine Funktion eines Berechnungsmodells

Name	Typ	Speicherklasse	Beschreibung
VALUE_Value	FLOAT	LOCAL	Ergebnis
VALUE_Status	FLOAT	LOCAL	Status des Ergebnisses
VALUE_UNIT	ENUMERATED(2) (Unsigned16)	LOCAL	PA-Unit-Code des Ergebnisses
NAME	VisibleString	LOCAL	Name im Modell
ANSCHLUSS	VisibleString	LOCAL	Anschlussname im Modell (optional)
IS_VALID	ENUMERATED	LOCAL	Verfügbarkeit des Ergebnisses

Aus der Berechnungsvorschrift kann abgelesen werden, welche Eingangswerte benötigt werden. Zu allen Eingangswerten kann über die \*\_IS\_VALID-Parameter die Verfügbarkeit geprüft werden. Die Berechnung kann dann, wie im Beispiel in Quelltext 4.5 auf Seite 74 dargestellt, ausgelassen werden. Das Funktionsergebnis darf dann ebenfalls nicht verwendet

werden. Dafür wird der Status des Wertes auf 0x08 (*BAD – not connected*) statt 0x80 (*GOOD*) [116] gesetzt.

**Quelltext 4.5:** Ausblendbare Berechnung eines Funktionsergebnisses in einer EDD-Methode (neuere Version des Beispiels aus [2])

```
if (( FptVTB__IS_VALID == 1 ) && ( PTempInFB__rho_pt_IS_VALID == 1 )) {
    if (( FptVTB__OUT_Status == 0x80 )) {
        float tmp1;
        tmp1 = FptVTB__OUT_Value * PTempInFB__rho_pt_VALUE_Value;
        assign_float(PTempInFB__FptMb_VALUE_Value, tmp1);
        assign_int(PTempInFB__FptMb_VALUE_Status, 0x80);
    }
    else {
        assign_int(PTempInFB__FptMb_VALUE_Status, 0x08);
    }
}
```

#### 4.2.4 Steuerung des Timings der Leseaufrufe im FDI Apparatpaket

Eine Steuerung des Timings der durch Berechnungsmodelle hervorgerufenen Leseaufrufe hat zwei wesentliche Gründe. Zum einen soll das Modell mit konsistenten Daten versorgt werden. Das heißt, dass alle Messwerte vom gleichen Zeitpunkt stammen. Genauso sollten diese Sätze von Messwerten möglichst mit einem konstanten Zyklus gelesen werden. Zum anderen soll die Last des FDI Servers, der FDI Communication Server, der Feldbusse und nicht zuletzt der Feldgeräte unnötig erhöht werden. Dies lässt sich im Berechnungsmodell oder durch einen Cache im EIH realisieren. Erstere Variante ist zu bevorzugen, da hierdurch weniger Last erzeugt wird. Die zweite Variante ist dennoch sinnvoll, wenn FDI Apparatpackages eine solche Funktion nicht besitzen.

In [1] wurde eine Möglichkeit hierfür vorgestellt, und in [2] verbessert. Die in Quelltext 4.6 abgedruckte Variante stellt den letzten Stand der Entwicklung dar.

**Quelltext 4.6:** EDD-Mechanismus zum gleichzeitigen zyklischen Lesen von EDD-Parametern aus Geräten in einer Methode (neuere Version des Beispiels aus [2])

```
METHOD method_read_pv {
    LABEL [method_read_pv_lbl];
    ACCESS ONLINE;
    CLASS INPUT;
    DEFINITION {
        if (FptVTB__IS_VALID == 1)
        {
            // _Value & _Status
            ReadCommand(Read__FptVTB__OUT);
        }
    }
    [...] (weitere Parameter)
}

METHOD method_calc_functions {
    LABEL [calc_functions_lbl];
    ACCESS OFFLINE;
    CLASS INPUT;
    DEFINITION {
        long cur_time;
        cur_time = [...] GetCurrentTime() [...];
        if (calc__last_refresh + calc__refresh_cycle < cur_time) {
            assign_int(calc__last_refresh, cur_time);
            method_read_pv();
        }
    }
}
```

```
[... (weitere Berechnungen)]
  }
}
```

#### 4.2.5 Menüstruktur für Apparatepakete

Alle in den verschiedenen Funktionsblocks definierten Parameter müssen noch in eine Menüstruktur, die UID eingeordnet werden. Die UID ist ein Teil der EDD des Apparatepaketes. Eine einheitliche Organisation der Menüs des UID schreibt der Usability-Styleguide [277] vor. Dabei sind drei wesentliche *Views* bzw. Hauptmenüs anzulegen: *Operate*, *Maintenance* und *Specialist*. Das *Operate*-Menü ist für die Anzeige aller für einen Operator bestimmten Daten und Zustandsinformationen bestimmt. Dabei werden alle als Kennzahl (KPI) markierten Berechnungsergebnisse hinzugefügt. Es könnte auch eine grafische Darstellung der dort angegebenen Werte hinzugefügt werden. Das *Maintenance*-Menü enthält alle instanziierten Messstellen und validen Zwischenergebnisse. *Transducer Blocks* und Parameter, die ausgeblendet wurden, sind nicht enthalten. Alle Elemente sind gemäß Modellstruktur in Submenüs eingeordnet. Im *Specialist*-Menü sind alle Parameter enthalten, insbesondere auch die `*_IS_VALID`-Parameter. Die Hierarchie ist dieselbe wie im *Maintenance*-Menü.

Zur EDD gehört eine Datei mit dem Wörterbuch für die Menüeinträge und Beschriftungen der `VARIABLE`-Konstrukte. Diese kann in beliebig viele Sprachen übersetzt werden, auch eine partielle Übersetzung ist möglich. Das Ausfüllen der Beschreibungen in der Standardsprache (Englisch) ist zwingend erforderlich.

### 4.3 Apparatepaket-Werkzeug

Aufgrund der Komplexität der EDDL-Sprache, der strikten Randbedingungen bei der Integration von Apparatepaketen und der Komplexität von Modell- und Menüstruktur ist das Schreiben einer EDD einem Hersteller von Apparaten oder Apparatemodellen nicht zuzumuten bzw. zuzutrauen. Demzufolge muss dem Modellhersteller eine entsprechende Werkzeugunterstützung angeboten werden. Hierfür wurde im Rahmen dieser Arbeit ein Prototyp als Proof of Concept erstellt und in [2] veröffentlicht. Die Ergebnisse sollen hier genauer beschrieben und eingeordnet werden. Mit Veröffentlichung des neuen EDD-Standards [126] und des Style-Guides [277] wurden auch noch Änderungen zu diesem Arbeitsstand notwendig. Auch sollte ein Werkzeug beim Erstellen von Modellen den Anwender dabei unterstützen, die Fachbegriffe aus den unterschiedlichen Gewerken korrekt und konsistent zu verwenden [278].

#### 4.3.1 Erstellen von Modellhierarchie und Elementen

Das Apparatepaket-Werkzeug wurde in *Java* entwickelt. Das in Abbildung 4.8 auf Seite 67 dargestellte Modell muss in der Applikation abgebildet werden. Dafür wurden noch zusätzliche Klassen eingeführt und Verfeinerungen zum Abspeichern aller in Abschnitt 4.2 ab Seite 66 beschriebenen Eigenschaften vorgenommen.

Abbildung 4.9 auf Seite 76 zeigt das Datenmodell für einen Apparat im Werkzeug. Die zentrale Abstraktion ist das `ApparateModellelement`, das eine gemeinsame Oberklasse

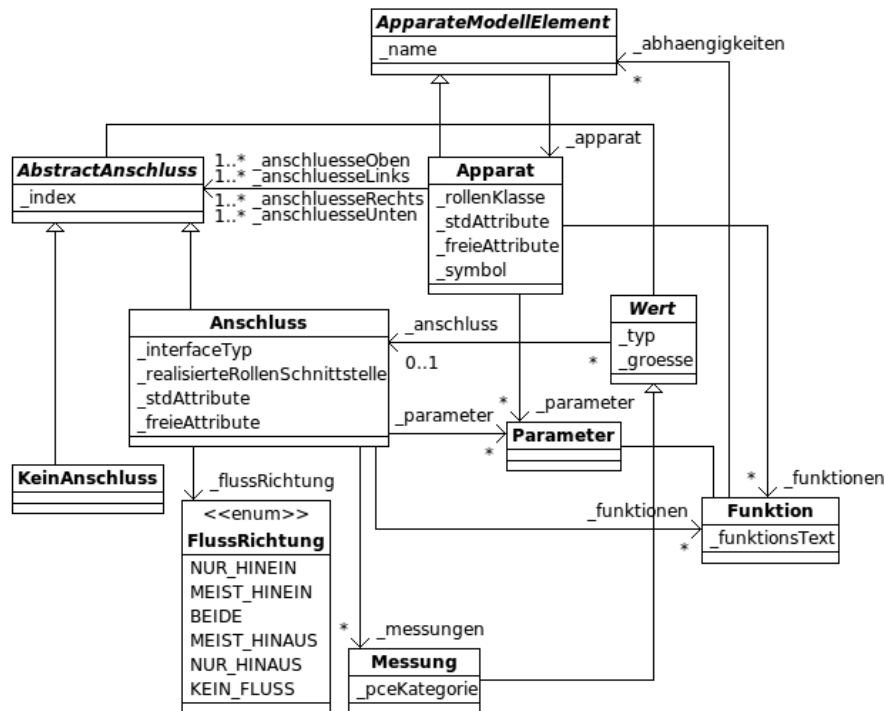


Abbildung 4.9: Datenmodell des Apparatemodells im Werkzeug (aus [2])

darstellt. Alle Elemente, die in einer Berechnung verwendet werden können, sind von einer zusätzlichen gemeinsamen Oberklasse `Wert` abgeleitet. Der `Apparat` ist eine Menge von Kacheln, die in je mindestens einer Zeile und Spalte angeordnet sind. An jeder freien Kante einer Randkachel kann ein `Anschluss` angefügt werden. Dies ist entweder ein `Anschluss`-Objekt oder ein Dummy-Objekt von der Klasse `KeinAnschluss`. Beide sind in einer gemeinsamen Oberklasse `AbstractAnschluss` zusammengefasst. `Anschluss`-Objekte haben ebenfalls eine Flussrichtung, die als ausschließliche oder als Hauptflussrichtung sowie in beide Richtungen festgelegt werden kann. Der Wert `KEIN_FLUSS` ist ein Messstellenanschluss. An einem solchen `Anschluss` kann nur eine Messstelle vorhanden sein, die direkt am `Apparat` und nicht über ein Rohr angeschlossen ist (z. B. Füllstand).

Die beschriebenen Attribute entsprechen den in Abschnitt 4.2 ab Seite 66 beschriebenen Anforderungen. Das Modell trägt alle von Anwender angegebenen Informationen und wird zum Speichern und Laden mittels *thoughtworks XStream* serialisiert und deserialisiert. Es wird eine XML-Datei ausgegeben, die das Objektmodell mit allen seinen Referenzen abbildet.

Die Anbindung des Apparatemodells in das Graphical User Interface (GUI) des `Apparatepaket`-Werkzeugs wird über das *Adapter*-Entwurfsmuster realisiert. Gleichzeitig kann das Modell gemäß *Visitor*-Muster für unbekannte Clients traversiert und ausgelesen werden. Somit sind die verschiedenen Prüf- und Exportaufgaben in Module kapselbar und vom Modell unabhängig. Abbildung 4.10 zeigt die *Adapter*- und *Visitor*-Klassen, die für das Datenmodell des Apparatemodells entwickelt wurden. Nach dem Erstellen eines `ApparateModellElement` in einem bestimmten Kontext wird die `ApparateModellElementAdapterFactory` des Kontexts aufgerufen, um dem Element den für den Kontext



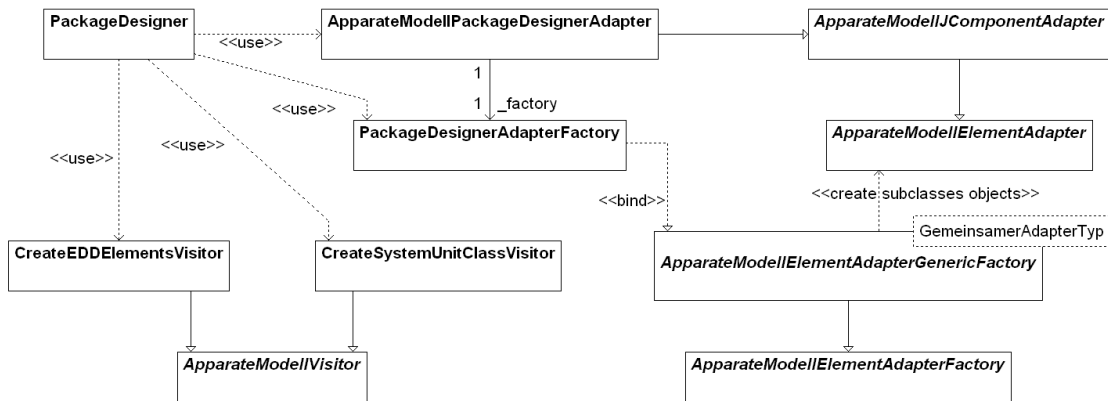


Abbildung 4.11: Einbindung des Datenmodells des Apparatemodells in die GUI des Apparatepaket-Werkzeugs

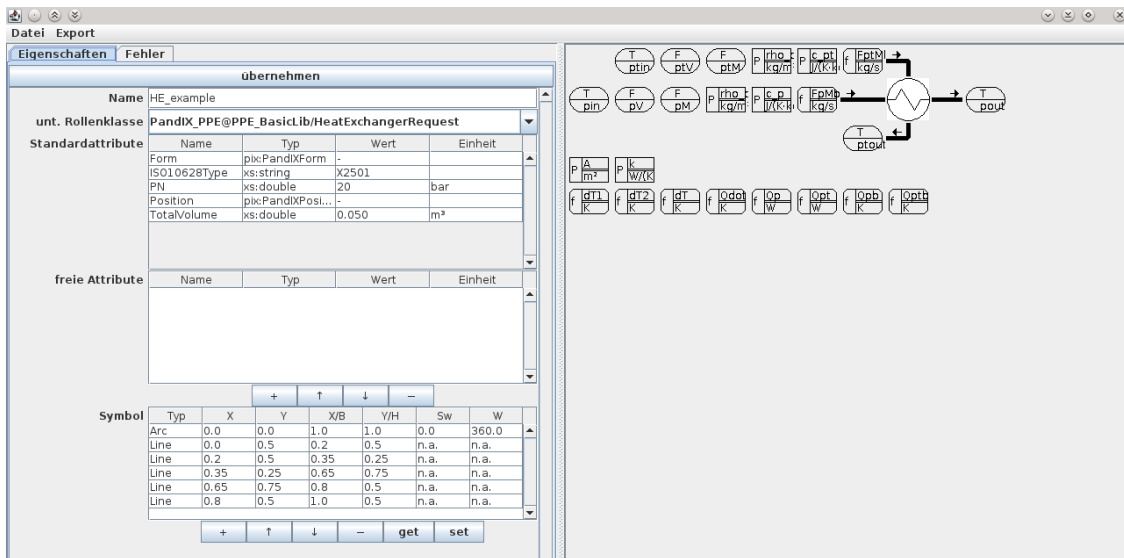


Abbildung 4.12: Übersicht über die GUI des Apparatepaket-Werkzeugs (aus [2])

[2] verwiesen, der relevante Auszug (Abschnitt 3) ist im Anhang A, Abschnitt A.2 ab Seite 172 abgedruckt.

Die Auswahl von PCE-Kategorien wurde gemäß IEC 62424 [17] modelliert. Zur Auswahl der Rollenbibliotheken wurden einige weit verbreitete Standardrollen mit ihren Schnittstellen ins Werkzeug implementiert. Um eine große Menge solcher Rollen anbieten zu können, müssten die Rollendefinitionen aus CAEX-Rollenbibliotheken geladen werden.

Für die Maßeinheiten wurde das *JScience*-Framework [279] verwendet, eine Implementierung des *Units of Measurement API for Java*. Es erlaubt eine typbasierte (statische) und objektbasierte (dynamische) Verarbeitung von Maßeinheiten. Als Alternative kam *UOMu* [280] in Betracht, wurde aber mangels einer guten Dokumentation oder Verwendungsbeispielen nicht verfolgt. Für alle PCE-Kategorien wurden alle für Messwerte in verfahrenstechnische Anlagen relevante, auf *SI*-Basis-Einheiten basierende Maßeinheiten [281, 282, 283] zur Auswahl vorgesehen. Die Größenbezeichnungen und Formelzeichen soll-

ten gemäß DIN EN 60027-1 [284] gewählt werden. Beispielsweise wurden für die Kategorie  $F$  (Durchfluss)  $\frac{\text{kg}}{\text{s}}$ ,  $\frac{\text{m}^3}{\text{s}}$  und  $\frac{\text{mol}}{\text{s}}$  ausgewählt. Der Quelltext 4.7 zeigt die Definition dieser Maßeinheiten zu ihren physikalischen Größen.

**Quelltext 4.7:** Drei physikalische Größen für die PCE-Kategorie F

```
/**
 * Volumenstrom
 */
public static final PhysikalischeGroesse VOLUMENSTROM = new PhysikalischeGroesse(
    "Volumenstrom", "Q", null, SI.CUBIC_METRE.divide(SI.SECOND));

/**
 * Massenstrom
 */
public static final PhysikalischeGroesse MASSENSTROM = new PhysikalischeGroesse(
    "Massenstrom", "Q_m", null, SI.KILOGRAM.divide(SI.SECOND));

/**
 * Stoffmengenstrom
 */
public static final PhysikalischeGroesse MENGENSTROM = new PhysikalischeGroesse(
    "Mengenstrom", "Q_n", null, SI.MOLE.divide(SI.SECOND));
```

Somit kann jeder Massen-, Volumen- und Stoffmengenstrom abgedeckt werden. Zur Vereinfachung werden alle Berechnungen im Modell mit diesen Einheiten ausgeführt. Damit können die Ausdrücke bis zu einem gewissen Maß auf ihre Korrektheit geprüft werden.

### 4.3.2 Notation der Modelle

In Funktionen können die Berechnungen als mathematische Gleichung eingegeben werden. Die Anwendung unterstützt den Nutzer dabei durch die Kontextvervollständigung (siehe Anhang A, Abbildung A.7 auf Seite 175). Hierfür werden alle im Modell verwendeten Namen und alle in EDDL definierten BuiltIn-Funktionen [127] in einer Liste angezeigt, die während des Tippens gefiltert wird. Die Bezeichner aus dem Modell werden während der Bearbeitung ständig aktualisiert und enthalten alle Objektamen ihrer Hierarchie. Die BuiltIns wurden inklusive Dokumentation aus dem Standard in *exp4j*-Funktionen übertragen. Zwei Beispiele sind in Quelltext 4.8 abgedruckt.

**Quelltext 4.8:** Beispiele für EDD-Builtins gemäß IEC 61804-5 [127] mit und ohne JScience-Implementierung [279]

```
/**
 * Builtin abs.<br>
 * Purpose<br>
 * The Builtin abs returns the absolute value of a floating point value.
 */
public static final EDDLBuiltInFunction abs = getAbs();

/**
 * @return abs
 * @see #abs
 */
private static EDDLBuiltInFunction getAbs()
{
    try
    {
        return new EDDLBuiltInFunction("abs", 1, new CustomFunction("abs", 1)
```

```
    {
        @Override
        protected Amount<? > applyFunctionWith(Amount<? >... values)
            throws ConversionException
        {
            return values[0].abs();
        }
    });
}
catch (InvalidCustomFunctionException e)
{
    e.printStackTrace();
}
return null;
}

[...]
/**
 * Builtin _ERROR<br>
 * Purpose<br>
 * The Builtin _ERROR puts a textual information in a log. This information is
 * put into the category error. The Builtin {@link #_TRACE} and the Builtin
 * {@link #_WARNING} write information to this same log.<br>
 * Any variable reference contained in the format string will be expanded to the
 * current value. Method local variables and DD variables can be referenced in
 * the message parameter. For more information, refer to the Builtin
 * {@link #PUT_MESSAGE}.
 */
public static final EDDLBuiltInFunction _ERROR = new EDDLBuiltInFunction(
    "_ERROR", Void.class, new Class[] {String.class, Integer[].class});

[...]
```

Die fertigen Ausdrücke werden dann mittels einer Erweiterung von *exp4j* [285] auf syntaktische Korrektheit (Basisfunktion von *exp4j*) und Konsistenz der Maßeinheiten (eigene Erweiterung mit *JScience* [279]) geprüft. Dabei wurde im Objektmodell von *exp4j*-Gleichungen der Wert von einem `double`-Wert auf ein *JScience*-Amounts umgestellt. Beim Ausrechnen einer Gleichung wird dadurch automatisch die Maßeinheitenkompatibilität geprüft. Da die Gleichungen noch keine Werte besitzen, wird für jeden Wert eine Zufallszahl zwischen 0.5 und 1.5 eingesetzt (Quelltext 4.9). Das verhindert unnötige Fehler wie Divisionen durch null, Gleitkommaüber- und -unterläufe. Der errechnete Wert hat keine Bedeutung. Dass bei dessen Berechnung kein Fehler aufgetreten ist, zeigt aber, dass die Formel mit korrekten Maßeinheiten gerechnet wird.

**Quelltext 4.9:** Bereitstellung eines Wertes für den *exp4j*-ExpressionBuilder

```
// abhaengigkeiten als zufalls-variablenwert in den expressionbuilder einspeisen
double dv = Math.random() + 0.5;
e.withVariable(n, Amount.valueOf(dv, w.getPhysikalischeGroesse().getEinheit()));
```

Um später Eingangswerte aus den Feldgeräten automatisch auf die im Modell verwendete Standard-*SI*-Einheit [281, 282] konvertieren zu können wurde eine Abbildung von PA-UnitCodes [116] auf *JScience*-Einheiten erstellt (Quelltext 4.10). Dabei kann der zum Wert `<value>` gehörende Parameter `<value>_UNIT` ausgelesen werden. Diese Nummer zeigt auf einen Eintrag in der Einheitentabelle. Mithilfe der Zielmaßeinheit im Berechnungsmodell (die kompatible *SI*-Einheit) kann ein Konverterobjekt erzeugt werden, welches die Maßeinheit stetig umwandelt. Im Betrieb wird diese Funktionalität vom EIH (Abschnitt 4.4 ab Seite 86) bereitgestellt.



Quelltext 4.10: Auswahl aus PA-UnitCodes [116] für *JScience* [279]

```

/**
 * Erstellt die Tabelle
 *
 * @return die Tabelle mit allen Einträgen aus der PI Spezifikation
 */
private static Map<Integer, Unit<? >> init()
{
    Map<Integer, Unit<? >> tab = new HashMap<>();

    // Profile-PA-Devices_3042_V301_Dec04.pdf
    // Seiten 96 bis 108

    // Seite 96
    tab.put(1000, SI.KELVIN);
    tab.put(1001, SI.CELSIUS);
    tab.put(1002, NonSI.FAHRENHEIT);
    tab.put(1003, NonSI.RANKINE);
    [...]
    tab.put(1186, SI.WATT);
    tab.put(1187, SI.TERA(SI.WATT));
    tab.put(1188, SI.GIGA(SI.WATT));
    tab.put(1189, SI.MEGA(SI.WATT));
    tab.put(1190, SI.KILO(SI.WATT));
    tab.put(1191, SI.MILLI(SI.WATT));
    tab.put(1192, SI.MICRO(SI.WATT));
    tab.put(1193, SI.NANO(SI.WATT));
    tab.put(1194, SI.PICO(SI.WATT));
    tab.put(1195, SI.JOULE.times(4184000).divide(NonSI.HOUR));
    tab.put(1196, SI.JOULE.divide(NonSI.HOUR));
    // tab.put(1197, );
    tab.put(1198, NonSI.HORSEPOWER);
    tab.put(1199, SI.WATT.divide(SI.METRE).divide(SI.KELVIN));
    tab.put(1200, SI.WATT.divide(SI.METRE.pow(2)).divide(SI.KELVIN));
    [...]
}

```

### 4.3.3 Automatische Eingrenzung der ausführbaren Modelle

Der EDD-Quelltext (Abschnitt 4.3.7 ab Seite 84) wird gemäß dem letzten Stand der Struktur des FDI Apparatepackages ausgegeben (Abschnitt 4.2 ab Seite 66). Hierdurch wird die automatisch Eingrenzung der ausführbaren Berechnungen automatisch implementiert. Im Werkzeug sollten dem Apparatemodell-Hersteller auch die Auswirkungen des Fehlens bestimmter Messstellenanforderungen auf die Berechenbarkeit des Modells angezeigt werden. Somit könnte die Anwendbarkeit in verschiedenen Szenarien geprüft werden.

Abbildung 4.13 auf Seite 82 und Abbildung 4.14 auf Seite 82 zeigen ein Beispiel eines Wärmeübertragers mit acht Messstellenanforderungen (an jedem der vier Anschlüsse ein  $T$ , und auf jedem Eingang zwei  $F$  [je mit Volumen- und Massenstrom]). Die verfügbaren Messstellen und berechenbaren Funktionen sind grün hinterlegt, die vorhandenen bzw. berechenbaren rot. In beiden Szenarien sind je ein Volumenstrom- und Massenstrom-Messaufnehmer als vorhanden angenommen. Somit kann über die Dichte der Massenstrom bestimmt werden, wenn er nicht direkt gemessen wird. In der ersten der beiden Abbildungen sind alle Temperaturmessgeräte vorhanden. Das Modell ist berechenbar. In der zweiten Abbildung fehlt  $T_{pin}$  und somit kann keiner der KPI berechnet werden. Das Modell wäre noch flexibler, wenn auch an den Abflüssen zwei wahlweise notwendigen  $F$ -Anforderungen vorhanden wären und diese mit denen der entsprechenden Zufüsse der Produkträume

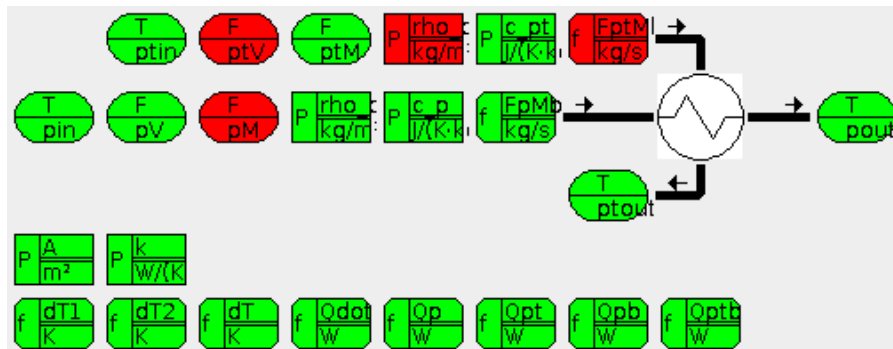


Abbildung 4.13: Mögliche Darstellung der Berechenbarkeit im Apparatepaket-Werkzeug (Gut-Fall, auf Basis von [2, Abb. 5])

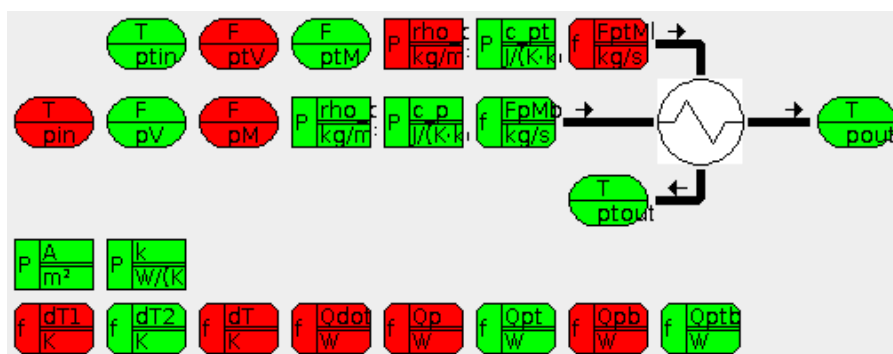


Abbildung 4.14: Mögliche Darstellung der Berechenbarkeit im Apparatepaket-Werkzeug (Schlecht-Fall, auf Basis von [2, Abb. 5])

verknüpft würden. In diesem Fall wäre jeweils einer von vier dieser Messaufnehmer notwendig.

Im prototypischen Apparatepaket-Werkzeug ist diese Funktion nicht umgesetzt. Abbildung 4.13 und Abbildung 4.14 sind aus Abbildung A.2 in Anhang A ab Seite 169 [2, Abb. 5] erstellt.

#### 4.3.4 User Interface Description/Plugin

Um die Ergebnisse der Berechnungen der Apparatemodelle wie angedacht in den gerätezentrierten PAM- oder CM-Tools anzeigen zu können wird ein User Interface Description (UID) und ggf. ein User Interface Plugin (UIP) benötigt.

Da ein UIP die Anzahl der notwendigen Werkzeuge erheblich vergrößert und die Integration in eine Werkzeugkette erheblich erschwert, wurde auf eine programmierte Benutzungsoberfläche verzichtet. Auch müssten die Vorgaben aus den Style-Guides für FDT [138] bzw. [277] FDI beachtet werden. Darüber hinaus ist die werkzeuggestützte Generierung von GUIs wenig individuell, was der Hauptvorteil eines UIP wäre. Eine Möglichkeit wäre, modellbasierte Ansätze zur Generierung von GUIs zu verwenden [286].

Das UID besteht aus unterschiedlichen, per EDD beschriebenen MENU-Konstrukten. Diese können ineinander geschachtelt werden. Der FDI-Style-Guide [277] beschreibt drei standardisierte Hauptmenüs. Diese werden gemäß der Beschreibung in Abschnitt 4.2.5

ab Seite 75 in die EDD exportiert (Abschnitt 4.3.7 ab Seite 84). Quelltext 4.11 zeigt ein Beispiel für ein Submenüs in der *Maintenance*-Ansicht.

**Quelltext 4.11:** Beispiel für ein Untermenü eines Transducers in der EDD

```
MENU Tab_s_PIn_Tpin
{
  LABEL [Menu_Tab_s_PIn_Tpin_lbl];
  HELP [Menu_Tab_s_PIn_Tpin_hlp];
  ITEMS
  {
    TpinTB__BLOCK_OBJECT ,
    TpinTB__ST_REV ,
    TpinTB__TAG_DESC ,
    TpinTB__STRATEGY ,
    TpinTB__ALERT_KEY ,
    TpinTB__TARGET_MODE ,
    TpinTB__MODE_BLK ,
    TpinTB__ALARM_SUM ,
    TpinTB__REF_DESC ,
    TpinTB__OUT ,
    TpinTB__OUT_STATUS ,
    TpinTB__OUT_UNIT ,
    TpinTB__NAME ,
    TpinTB__ANSCHLUSS ,
    TpinTB__IS_VALID ,
    TpinTB__FDI_DEV_NODE ,
    TpinTB__FDI_VAR_PATH
  }
}
```

### 4.3.5 SystemUnitClass Export

Wie bereits in Abschnitt 4.2 ab Seite 66 beschrieben, kann die Struktur des Apparates in einer CAEX-SystemUnitClass repräsentiert werden. In Quelltext 4.2 auf Seite 68 bzw. Anhang B, Abschnitt B.2 ab Seite 181 ist ein Beispiel abgedruckt [2]. Die Dateien können bereits vor dem Erstellen der Messstellenanforderungen, Parameter und Berechnungsmodelle allein aus der vollständigen Struktur des Apparates im Werkzeug exportiert werden. Dabei sind verschiedene Informationen, die zur Integration in das Anlagenengineering wichtig sind, automatisch eingetragen. Die Vollständigkeit dieser Informationen und die Kompatibilität zur Rollenklasse kann mittels einer CAEX-Prüfung nachgewiesen werden [3], Abschnitt 4.1.1.3 ab Seite 58. In Ermangelung einer CAEX-Bibliothek oder eines CAEX-Datentyps zur Beschreibung der Darstellung eines Objektes wird das Symbol nicht mit exportiert. Eine Grafik könnte in das FDI Package hinzugefügt werden.

Die exportierte SystemUnitClass sollte im Engineering der Anlage zur Verfügung stehen, damit später die Suche nach Instanzen (Abschnitt 4.5.1 ab Seite 95) besser automatisierbar ist.

### 4.3.6 ProList und PAM-Methodenblatt Export

Für die elektronische Bestellabwicklung [287, S. 19] und weitere in PROLIST bzw. eCl@ss abgedeckte Anwendungsfälle könnte aus der Apparatestruktur und deren Eigenschaften eine Merkmalsbeschreibung exportiert werden. In [63] wird die Verwendung von Merkmalen in Systemeinheiten beschrieben. Eine Struktur ist bereits in der IEC 61987-11 [85, Abs. 5.6, S. 19] vorgegeben. An dieser Stelle wird eine Kopplung von genormten Rollenklassen

äquivalenten Merkmalleisten benötigt. Auf diese Weise kann dem Bearbeiter im Werkzeug eine Liste von zu setzenden Merkmalen angeboten werden. Sind diese eingetragen, kann eine Merkmalsbeschreibung exportiert werden. Diese kann ebenfalls auf die vorhandene `SystemUnitClass` verweisen. Somit können bei einer Auswahl eines Produktes über `eCl@ss` automatisch die `SystemUnitClass` hinterlegt und somit eine spätere Integration des Modells vereinfacht werden.

Auch die Einsatzmöglichkeit und die Funktion des Modells kann beschrieben werden. Die VDI/VDE-Richtlinie 2651 Blatt 2 [198] beschreibt dafür eine Methodik und ein (nicht-formales) Ausgabeformat für PAM-Funktionen. Die Beschreibungen für komplexe Assets beziehen sich auf konkrete Teilanlagen, können aber auch für generische Anwendungsfälle notiert werden. Einige der notwendigen Informationen, z. B. die Struktur, können bereits mit den für das Apparatemodell zwingend notwendigen Informationen automatisch ausgefüllt werden. Weitere Informationen, wie z. B. die Beschreibung der Funktion, müssten jedoch in zu erstellenden Eingabemasken eingetragen werden. Für eine solche Funktion ist also für den Ersteller des Modells im Werkzeug zusätzlicher Aufwand notwendig. Dieser ist jedoch, um das Modell für dessen Anwender zu dokumentieren, auf andere Weise ebenfalls notwendig.

### 4.3.7 EDDL Export

Genauso wie der Export der `SystemUnitClass` (Abschnitt 4.3.5 ab Seite 83) wird die EDD des Apparatepaketes vollständig aus dem Werkzeug exportiert. Dabei wird mittels eines speziellen Visitors (Abschnitt 4.3.1 ab Seite 75) über das Modell des Paketes traversiert. Es werden die Daten gemäß dem Modell in Abschnitt 4.2 extrahiert und in der beschriebenen Form ausgegeben. Die Implementierung basiert auf Stümpfen und Templates, in welche die aktuellen Bezeichner, Datentypen, usw. eingesetzt werden. Gleichzeitig werden alle anzuzeigenden Texte mit Bezeichnern versehen und ein Template für eine Übersetzungsdatei (Dictionary) angelegt. Dies wird in Quelltext 4.12 und Quelltext 4.13 verdeutlicht. Mithilfe der *TODO*-Markierungen können alle nicht beschriebenen bzw. nicht übersetzten Elemente leicht gefunden werden.

Eine Funktion zur Übernahme der alten Übersetzung bei einem erneuten Export wurde nicht implementiert, sollte aber in einer vollständigen Implementierung vorgesehen werden. Auch könnte eine Auswahl an Sprachen integriert und eine Übersetzung mittels elektronischem Wörterbuch implementiert werden. Genauso könnten viele paketunabhängige Standardparameternamen – wie die Manufacturer ID im Beispiel – gespeichert und wiederverwendet werden.

**Quelltext 4.12:** Label einer Variablen mit Verweis ins Dictionary

```
[...]  
  VARIABLE PhysB__DeviceManID  
  {  
    LABEL      [PhysB__DeviceManID_lbl];  
    HELP       [PhysB__DeviceManID_help];  
  }  
[...]
```

**Quelltext 4.13:** Automatisch erzeugte Dictionaryeinträge

```
[...]  
[0,0] PhysB__DeviceManID_lbl
```

```

"TODO PhysB__DeviceManID"
"|de|TODO PhysB__DeviceManID"

[0,0] PhysB__DeviceManID_help
"TODO PhysB__DeviceManID"
"|de|TODO PhysB__DeviceManID"

[...]

```

Die exportierten Dateien sollen dem IEC 61804-3 [124] Standard genügen, in der finalen Version der harmonisierten Variante [126]. Da die EDD des FDI Apparatepackages die Struktur von PA Feldgeräten gleicht, sollte hier zusätzlich die *EDD Interoperability Guideline* [275] beachtet werden.

Die ausgegebenen Dateien können mit Werkzeugen wie der *is EDD Workbench* [129] oder der FDI Referenzimplementierung [259] überprüft werden. Letzteres Werkzeug kann in der vollständigen Werkzeugkette auch die Übersetzung der Klartextdateien in das Binärformat (Tokenizing) [288] vornehmen.

### 4.3.8 Package Export

Der FDI Standard Teil 4 [163] beschreibt Aufbau (gemäß [164]) und Inhalt der FDI Packages. Das Paket ist ein *ZIP*-gepackter Container mit bestimmten Unterordnern. Die enthaltenen Dateien werden im *Package Catalog*, einer Art Manifest-Datei, aufgelistet. Darunter sind die EDD-Datei im Binärformat [288], UIPs für unterschiedliche Anzeigeplattformen, Abbildungen und Dokumentationen und Feldbus-spezifische Integrationsdaten (z. B. die GSD für PROFIBUS-Geräte).

Abbildung 4.15 auf Seite 86 zeigt die Werkzeugkette aus [259] zum Erstellen von FDI Packages. Die Werkzeugkette ist als *Eclipse-Plugin* implementiert und ruft seine Werkzeuge als Kommandozeilen-Programme auf. Somit könnten diese Werkzeuge auch in das Apparatepakete-Werkzeug integriert werden. Da im aktuellen Arbeitsstand Dateien (z. B. das EDD-Dictionary) nachbearbeitet werden müssen, ist eine vollständige Integration der FDI-Werkzeugkette nicht praktikabel. Sollte die (Nach-)Bearbeitung der exportierten Dateien und eine Versionsmigration im Werkzeug umgesetzt sein, wird dies jedoch möglich.

Ein FDI Apparatepackage enthält notwendigerweise folgende Dateien:

**Package Catalog** XML-Datei mit Inhalts- und Versionsangaben

**EDD** im Binärformat

**Dictionary** nachbearbeitet

**GSD** als Anforderung des FDI-PA-Profiles

**Apparatesymbol** aus dem Werkzeugsymbol generierte Bitmapdatei

**SystemUnitClass** zur Unterstützung der Integration

Da ein FDI Apparatepackage einem FDI Package eines PROFIBUS PA-Gerätes gleicht, sollten auch die Konventionen aus dem entsprechenden FDI-Profil [169] beachtet werden.

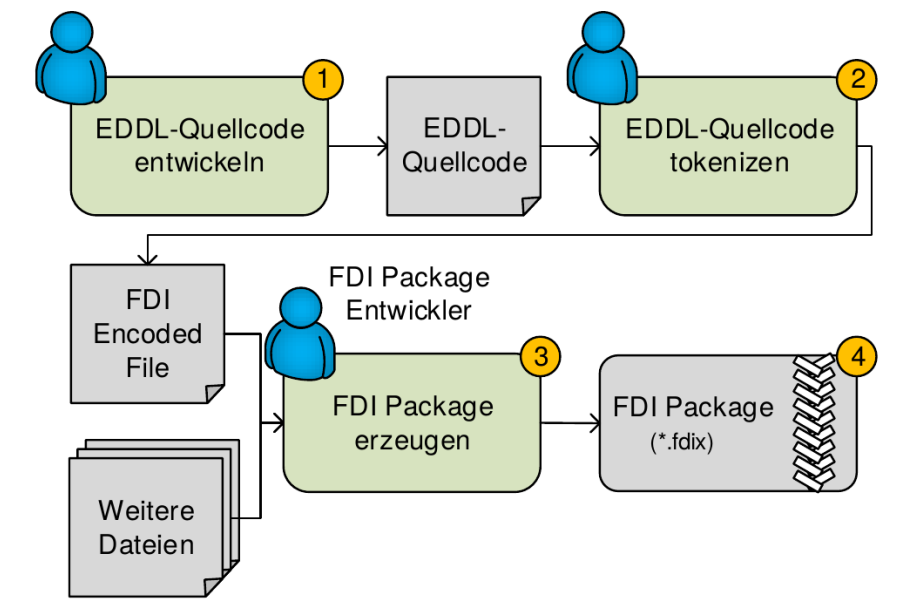


Abbildung 4.15: Werkzeugkette zum Packen von FDI Packages (aus [259])

## 4.4 Equipment Instance Handler

Der Equipment Instance Handler (EIH) bindet die FDI Apparatepackages zur Laufzeit in den FDI Server ein und realisiert die für die Bewertungsmodelle notwendigen Lesezugriffe auf Feldgeräte mittels des FDI Servers. Abbildung 3.10 auf Seite 46 zeigt diese Einbindung. Entwurf und Implementierung dieser Komponente wird im Folgenden vorgestellt. Gemäß der Migrationsstrategie (Abschnitt 3.6 ab Seite 48) wird die Variante ohne FDI Server mit OPC UA-Unterstützung beschrieben.

Abbildung 4.16 zeigt die Architektur dieser Softwarekomponente. Die Konfigurationskomponente (Abschnitt 4.4.1) verarbeitet die globalen Einstellungen und die Instanzdaten, die vom Instanzierungswerkzeug (Abschnitt 4.5 ab Seite 94) geliefert werden. In Abschnitt 4.4.2 ab Seite 88 wird die vom EIH angebotene OPC UA-Schnittstelle, welche ein FDI Communication Server-Informationsmodell umsetzt, beschrieben. Die darin empfangenen Anfragen werden an die entsprechenden Instanzen weitergeleitet. In diesen Instanzen erfolgen die konfigurierte Abbildung auf konkrete Feldgeräte und der Aufruf von Lesebefehlen. Diese werden vom OPC UA-Client umgesetzt (Abschnitt 4.4.3 ab Seite 91). Die Suchstrategie zum Auffinden der benötigten Variablen-Nodes wird in Abschnitt 4.4.4 ab Seite 92 beschrieben. Auch eventuell notwendige Umrechnungen von Werten in andere Maßeinheiten erfolgen in den Instanzen. Eine von der Implementierung in FDI Apparatepackages unabhängiger Mechanismus zur Kontrolle der Netzwerklast könnte implementiert werden. Da alle im Rahmen der Arbeit entwickelten FDI Apparatepackages einen paketinternen Mechanismus für diese Lastkontrolle implementieren, ist diese für die Evaluation nicht notwendig. Sie wurde im Prototyp deswegen nicht umgesetzt. Um unterschiedlich arbeitende FDI Apparatepackages sicher nebeneinander betreiben zu können, ist ein solcher Mechanismus erforderlich. Darum wurden verschiedene Umsetzungsideen dafür diskutiert (Abschnitt 4.4.5 ab Seite 93). Der EIH kann auf einer dedizierten Maschine

oder auf dem gleichen Rechner wie der FDI Server ausgeführt werden. Es wird nur eine Netzwerkverbindung zum FDI Server benötigt, über die ein OPC UA-Zugriff effizient und in beiden Richtungen möglich ist.

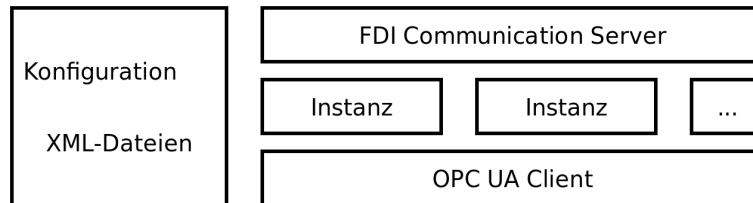


Abbildung 4.16: Komponenten des Equipment Instance Handlers

#### 4.4.1 Datenmodell für die Konfiguration des Equipment Instance Handlers

In Abschnitt 4.2 ab Seite 66 und den entsprechenden Unterabschnitten wurden die notwendigen Daten für den Betrieb eines Bewertungsmodells identifiziert und festgelegt. Alle Daten, die eine direkte Abhängigkeit zu einem Feldgerät haben, insbesondere wenn sie sich im Betrieb ändern können, müssen im Datenmodell für die Instanzen auf dem EIH verwaltet werden (z. B. EDDL-Speicherklassen *CONTAINED*, *INPUT*, *DYNAMIC*). Daten, die ausschließlich zur Konfiguration des Modells im FDI Server notwendig sind (z. B. EDDL-Speicherklasse *LOCAL*), müssen nicht im EIH vorhanden sein. Diese Daten sind im Gerätespeicher des FDI Servers abgelegt. Hierdurch ergibt sich der Nachteil, dass der Datensatz für eine Instanz eines FDI Apparatepackages auf zwei Stellen verteilt ist. Würde man konsequent alle Parameterwerte auch im EIH ablegen, wäre ein vollständiger Datensatz vorhanden. Jedoch könnten durch Änderung lokaler Parameter auf dem FDI Server Inkonsistenzen auftreten. Dies könnte behoben werden, wenn alle Parameter als Geräteparameter deklariert würden. Dies ist jedoch, besonders bei den *\*\_IS\_VALID*-Variablen nicht praktikabel. Aufgrund des anfallenden Datenverkehrs zwischen FDI Server und FDI Communication Server und den damit unvermeidbaren Verzögerungen in den Berechnungen, wurde die verteilte Variante gewählt.

Die Konfiguration wird mittels einer XML-Datei auf dem EIH abgelegt. Das Dateiformat definiert eine Liste von Instanzen, in denen die Parameter mit ihren Werten gemäß der Modellstruktur abgelegt sind. Somit bleibt die Zuordnung der Parameter zu den Funktionsblöcken erhalten. Über den FDI Server können später die Werte der Parameter, die als *HANDLING READ & WRITE* deklariert sind und die Speicherklassen *CONTAINED* oder *LOCAL* haben, geändert werden. Schreibzugriffe auf diese Parameter werden in die Konfigurationsdatei zurück gespeichert. Das XSD-Schema für die XML-Dateien zur Konfiguration des EIH ist in Anhang B, Abschnitt B.3 ab Seite 182 abgedruckt. Anhand dessen kann jede Datei auf Gültigkeit (Syntax) überprüft werden. Das Konfigurationsmodell ist im rechten Teil von Abbildung 4.24 auf Seite 101 dargestellt.

Da die FDI Apparatepackages nach dem FDI-Profil für PROFIBUS PA aufgebaut sind, ist für die Adressierung der Variablen noch die Position der FB im Slot-Index-Adressraum festzulegen. Dafür wird anhand der Anzahl der Parameter im FB festgelegt, in welchen Slots die FB liegen und an welchem Index (Offset) sie beginnen. Slot und Index für die Adressierung ergeben sich aus den im *Device Directory* abgespeicherten

Slot und dem Offset sowie dem relativen Index in der EDD (Displacement). Da die FDI Apparatepackages bislang keinem Applikationsprofil genügen, ist die Platzierung im Adressraum weitestgehend wahlfrei, die Vorgaben aus dem PA-Profil [116] sind einzuhalten bzw. schon durch die Struktur der FB gegeben. So soll z.B. der *Physical Block* an Slot 0, Index 16 liegen. Das *Device Directory* selbst ist in einer Struktur definiert und in Slot 1, Index 0-13 im Adressraum abgelegt. Der EIH erzeugt diese Struktur für jede Instanz und gibt sie auf Leseanfragen aus. Mit dem PN IO-Adressraum wären wesentlich mehr Parameter bei gleicher Flexibilität adressierbar [170]. Dies scheint jedoch nur dann praktikabel, wenn ein Apparate-Applikationsprofil standardisiert werden sollte, da dann Bereiche fest einer standardisierten Funktion zugewiesen werden können.

### 4.4.2 FDI Communication Server

Der FDI Communication Server bietet als Komponente des EIH den Zugriff auf die Apparate-Feldgeräte [158, 169, 289] an.

Das Informationsmodell für den FDI Communication Server wurde in [11] und Abschnitt 4.1.2.1 ab Seite 63 modelliert. Nach der Freigabe des OPC UA-SDK in der Version 1.4 von *Unified Automation* konnte dieses Modell auch generisch als NodeSet-XML-Datei geladen und mittels *Extensions* an die Serverfunktionalität angebunden werden. Dies hat den Vorteil einer besseren Trennung von OPC UA-Server-spezifischem Code und dem Code des EIH. Andererseits wird bei der Konfiguration immer ein zusätzlicher Schritt benötigt, welcher das Adressraummodell aktualisiert. Dabei ist insbesondere die Wiedergabe des Servermodells im NodeSet-XML-Format mit den korrekten und eindeutigen *NodeIDs* sehr aufwändig. Die Verwendung einer XSLT zur Erzeugung der NodeSet-XML aus der Konfigurations-XML-Datei wäre passend, ist jedoch wegen der komplexen Anforderung an die Vergabe der *NodeIDs* nicht mit vertretbarem Aufwand zu erstellen.

Stattdessen wurde die NodeSet-XML für die Typ-Beschreibung und eine weitere NodeSet-XML-Datei für die Instanz im Instanzierungswerkzeug erzeugt. Auch hierfür wurde eine auf der Klassenstruktur des Apparatepaket-Werkzeugs laufende *Visitor*-Klasse implementiert. Dieser Visitor arbeitet *Template*-basiert. Im *UA Modeler*-Tool wird für einen Beispiel-Typ exemplarisch eine Beispiel-Instanz angelegt und beide NodeSets exportiert. Aus diesen wurde das *Template* erstellt, welches Strukturen kopiert, in welche spezifische Inhalte wie Namensraum, *NodeIDs* und Namen sowie die instanzspezifischen NodeSet-*Extensions* eingesetzt werden.

Abbildung 4.17 zeigt die Einbindung und den Zugriff auf Geräte in einem FDI Communication Server. Das Informationsmodell des FDI Communication Servers wird im FDI Server basierend auf EDD-Konstrukten aus dem FDI Package des FDI Communication Servers aufgebaut. Die entsprechenden Konstrukte sind in Quelltext 4.14 dargestellt. Abbildung 4.18 auf Seite 90 zeigt ein Beispiel für den Adressraum eines FDI Communication Servers für Apparatemodelle.

**Quelltext 4.14:** Einbindung des FDI Communication Servers in den FDI Server (aus [169])

```
COMPONENT Profibus_Communication_Device
{
    LABEL "PROFIBUS communication device";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { Profibus_Service_Provider_Relation }
```



```

    BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION Profibus_Service_Provider_Relation
{
    LABEL "Relation to communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
}

COMPONENTS
{
    Profibus_Service_Provider{AUTO_CREATE}
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

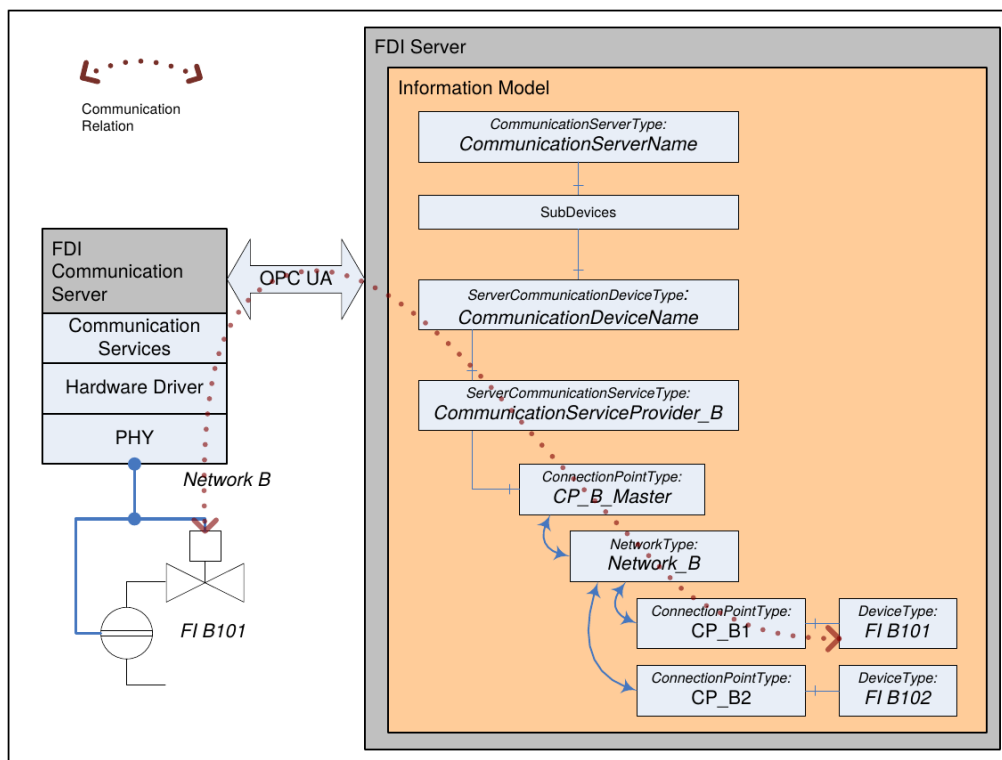
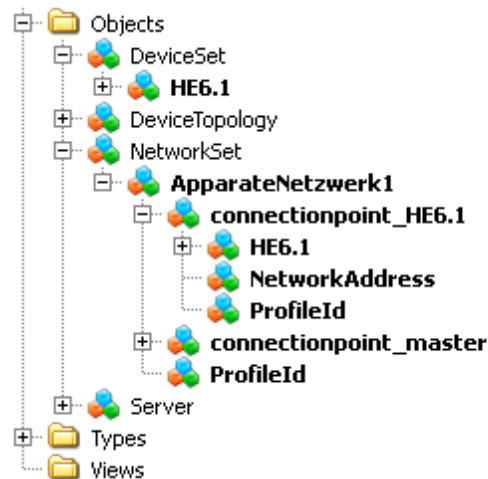


Abbildung 4.17: Einbindung eines Netzwerkes über einen FDI Communication Server (aus [159])

Die Repräsentation eines Apparates wird in den Abbildungen 4.19a-c auf Seite 91 dargestellt. Die Typen werden in einem FDI Server aus dem FDI Apparatepackage generiert und die Instanzen beim Erstellen jedes konkreten Apparatemodells daraus abgeleitet. Die Blöcke in Abbildung 4.19b auf Seite 91, die mit `trans_` beginnen, genügen dem Typ in Abbildung 4.19a. Das virtuelle Apparat-Feldgerät `HE_6.1` in Abbildung 4.19c ist eine Instanz des Typs `PAHeatExchangerAquisistionType` in Abbildung 4.19b.

Beim Erstellen der Instanzen wird ein virtuelles PROFIBUS PA-Segment und je ein virtuelles Feldgerät für eine Instanz eines Apparatemodells angelegt (Abbildung 4.18 auf Seite 90). Hierbei sind die Begrenzungen der Busadressen der realen Netzwerke ebenfalls zu beachten. In einem PROFIBUS-Netzwerk existieren nur 126 (+1) Adressen (vgl. Quelltext 4.15 auf Seite 90). Sollen mehr virtuelle Feldgeräte erzeugt werden, muss



**Abbildung 4.18:** Adressraummodell des FDI Communication Servers des Equipment Instance Handlers (zur Übersichtlichkeit wurden Nodes entfernt)

weiteres virtuelles Netzwerk angelegt werden. Die Adressen werden vom EIH vergeben. Dass diese über das Netzwerk nicht geändert werden können, kann in der GSD im FDI Apparatepackage festgelegt werden (siehe Quelltext 4.16 auf Seite 91). Ebenfalls kann der Parameter **Address** als read-only deklariert werden (Quelltext 4.15: **HANDLING READ;**). Ebenfalls sollten die Adressen von gelöschten Instanzen nicht neu vergeben werden.

**Quelltext 4.15:** Einbindung des ConnectionPoint in der EDD eines Feldgerätes oder Apparatepakets (aus [169])

```

COMPONENT ConnectionPoint_Profibus_PA
{
  LABEL "PROFIBUS PA Connection Point";
  CLASSIFICATION NETWORK_CONNECTION_POINT;
  CAN_DELETE FALSE;
  PROTOCOL Profibus_PA;
}

VARIABLE Address
{
  LABEL "Station address";
  HELP "Address of the PROFIBUS slave";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(1)
  {
    INITIAL_VALUE 126;
    MIN_VALUE 0;
    MAX_VALUE 126;
  }
  HANDLING READ; // & WRITE;
}

COLLECTION ConnectionPoint
{
  LABEL "PROFIBUS PA Connection Point data";
  MEMBERS
  {
    CONNECTION_POINT_ADDRESS, Address;
  }
}

```



Abbildung 4.19: Struktur des Apparate-Informationsmodells (zur Übersichtlichkeit wurden Nodes entfernt)

Quelltext 4.16: GSD-Fragment zum Verhindern der Adresszuweisung über den Bus

```
Set_Slave_Add_Supp = 0
```

#### 4.4.3 FDI-Discovery und Client für den wiederholbaren Zugriff

Um die Apparatepakete instanziierten zu können, müssen alle von einem FDI Server verwalteten Feldgeräte gefunden werden. Hierfür wird das Informationsmodell durchsucht und in allen gefundenen Nodes vom Typ `ServerCommunicationDeviceType` die `Scan`-Methode aufgerufen. Danach ist das `DeviceSet` des FDI Servers aktuell. Der Browse-Path der einzelnen Geräte ist damit bestimmt (ein Browse-Path ist die Liste der Browse-Namen der Nodes, die ausgehend vom Server-root über hierarchischen Referenzen zu einer gesuchten Node führen). Unterhalb der Geräte-Nodes können die benötigten Variablen durch `browse` der hierarchischen Referenzen gesucht werden. Die Suchstrategie wird in Abschnitt 4.4.4 ab Seite 92 erläutert. Benötigt werden die Messstellenbezeichnung,

die zwingend mit dem `DisplayName` der Geräte-Node übereinstimmen sollte, und der Messwert des Feldgerätes inklusive Skalierung und Maßeinheit.

Mithilfe des *Unified Automation C++ Client SDK* wurde ein Client [12, Abs. 1.2, S. 3] entwickelt, der anhand eines Browse-Paths eine Variable von der *NodeID* unabhängig wiederfinden und auslesen kann (Abschnitt 4.1.2.2 ab Seite 63). Dies ist notwendig, da der FDI Server die Informationsmodelle nach der Nutzung wieder entlädt und die *NodeIDs* für die Elemente des Informationsmodells nicht deterministisch vergibt. Diese Komponente wird vom EIH wiederverwendet, um die Lesezugriffe im Betrieb durchzuführen.

### 4.4.4 Datenmodell der zu verlinkenden Geräte

Die Engineeringdaten-Exporte, die in Abschnitt 4.1.1 ab Seite 57 beschrieben wurden, enthalten unter anderem die Anforderungen für die zu verwendenden Feldgeräte. Aus den *Dia*-Dateien konnten nur die Messstellenbezeichnungen angegeben werden, welche in den Engineeringdatenbanken durchgängig verwendet werden. Diese werden mit anderen Parametrierdaten bei der Inbetriebnahme in die Feldgeräte übertragen und können dort ausgelesen werden.

Die verwendeten Feldgeräte stammen von unterschiedlichen Herstellern und werden eventuell auch über unterschiedliche Feldbus-Technologien angebunden. Alle für die Feldgeräteanbindung in der Prozesstechnik relevanten Technologien, die derzeit von FDI unterstützt werden, folgen dabei dem Funktionsblockprinzip der IEC 61804-2 [119]. Darin ist abstrakt das Datenmodell für die Geräte eingegrenzt (siehe Abbildung 4.20). Zur Strukturierung werden ausschließlich Funktionsblöcke verwendet. Bei modularen Geräten können sich die Blöcke auch in Modulen befinden.

Verschiedene Geräte-Identifikationsdaten (z. B. PROFIBUS I&M [104]) und Prozessdaten werden in ein digitales Datenmodell gespeichert, auf welches über die Feldbus-Schnittstelle zugegriffen werden kann. Die Feldgeräte unterschiedlicher Feldbus-Technologien besitzen verschiedene Strukturen und verschiedene Adressierungsschemata für diese Daten. Auch sind verschiedene zusätzliche Spezifikationen wie [290] für EDDL für PROFIBUS zu beachten. Dies ist für den Zugriff ein kleineres Problem, da die Anzahl der Technologien begrenzt und auch klein ist. Zudem ist die Beschreibungssprache für alle Technologien weitestgehend harmonisiert [126]. Weiterhin soll für den Zugriff ein FDI Server verwendet werden, durch den die Zugriffstechnologie transparent ist. Abbildung 4.6 auf Seite 65 zeigt eine beispielhafte Ausprägung ohne die im Folgenden beschriebenen Erkenntnisse. Die Semantik der Daten im OPC UA-Informationsmodell ist möglicherweise durch den verwendeten Namensraum festgelegt. Die richtige Interpretation hängt hingegen vom Client ab. Der gleichartige Zugriff ist jedoch durch die OPC UA-Technologie bereits möglich.

Unterschiedliche Hersteller entwickeln bei gleicher Gerätefunktion üblicherweise eine unterschiedliche Struktur und Benennung für die abgelegten Daten. Dies ist für die gleichartige Behandlung von Geräten ein großes Problem, da die Vielfalt mit der Größe der Anlage wächst. Durch Applikationsprofile [112, 291] wird zumindest für die Kernfunktionalität eine Teilmenge der Daten in Adressierung und Semantik standardisiert (z. B. dem PA-Profil [116, 117, 103]). Mit diesem Applikationsprofil können die Messwerte in einem ausgelesenen Informationsmodell eines PROFIBUS-PA-Gerätes regelbasiert zuverlässig aufgefunden werden. Für die unterschiedlichen *Transmitter*-Typen wurden unterschiedli-

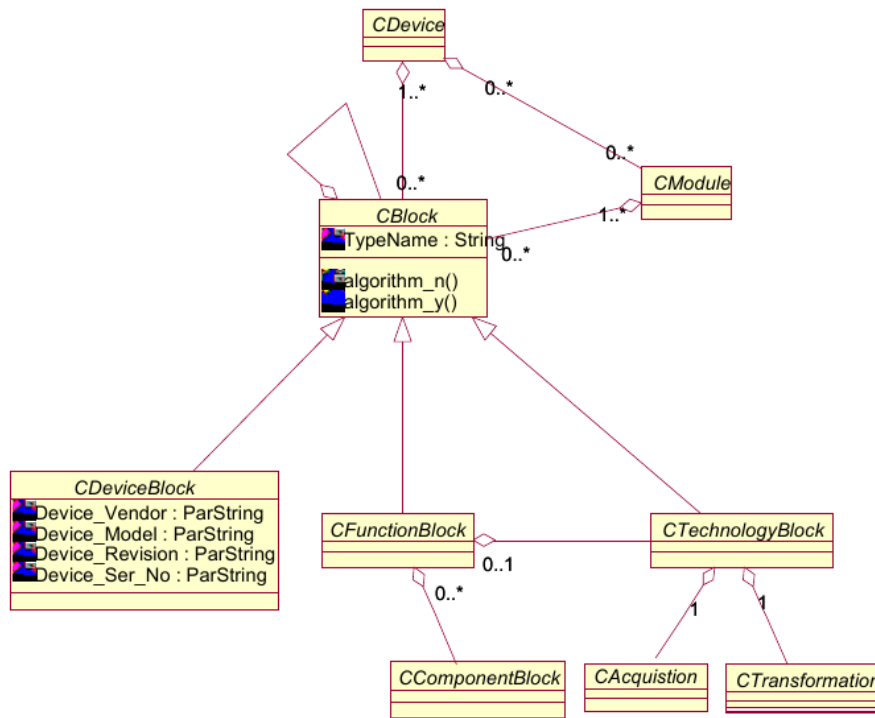


Figure 8 – UML class diagram of the device model

Abbildung 4.20: Gerätemodell gemäß IEC 61804-2 (aus [119, Fig. 8, S. 24])

che Transducerblöcke mit standardisierten Parametern definiert. Hierdurch entsteht zwar eine gewisse Vielfalt, diese wird jedoch mittels Functionblocks, die eine Nachbearbeitung realisieren (meistens ein *AnalogInputFunctionBlock*), an der Schnittstelle wieder reduziert. Somit können für PROFIBUS PA Sensor-Feldgeräte mittels der Parameter *OUT*- und *OUT\_SCALE* der Prozesswert und dessen Maßeinheit sicher bestimmt werden.

Ähnlich kann bei HART- und FF-Geräten vorgegangen werden. Diese verwenden in der EDD ein einfacheres Blockmodell [292, 126]. Für die Messstellenbezeichnung von HART-Geräten wird in den *PDM-Standard-Includes* der Parameter *TAG* verwendet. Der Prozesswert, der über das 4-20 mA-Signal übertragen wird, ist von dessen Skalierung unabhängig als PV über das HART-Interface verfügbar. Gleichsam kann in FF-Feldgeräten auf die PV oder *OUT\_D\** zugegriffen werden. Die Bedeutung der Werte aus weiteren möglichen Funktionsblöcken herauszufinden wäre jedoch nur mit dem Wissen aus FF-Geräteprofilen möglich [102].

#### 4.4.5 Möglichkeiten der Steuerung des Timings der Leseaufrufe im Equipment Instance Handler

Wie bereits in Abschnitt 4.4 ab Seite 86 erwähnt, wurde eine Lösung zur Kontrolle der Netzwerklast und Datenkonsistenz in den FDI Apparatepackages implementiert. Darum soll lediglich beschrieben werden, welche Möglichkeiten es für eine Umsetzung im EIH gibt und welche Randbedingungen zu beachten sind. Insbesondere soll auf die Verträglichkeit mit einem im Modell umgesetzten Mechanismus eingegangen werden.

Vollständigkeit und Gleichzeitigkeit (Konsistenz) sind die wichtigsten Anforderungen an Messwerte aus den verschiedenen, einem Apparatemodell zugeordneten Feldgeräten. Dies wird im FDI Apparatepackage durch eine Methode realisiert, die Werte nacheinander (quasi gleichzeitig) liest und für die Verarbeitung zwischenspeichert. Der Zwischenspeicher wird mit einer Zeitspanne konfiguriert, der ein erneutes Lesen vor dem Ablauf dieser Zeit verhindert (Abschnitt 4.2.4 ab Seite 74).

Ähnliches muss ein Mechanismus im EIH realisieren. Bei der Konfiguration der Apparatemodelle werden alle Geräte bekannt gegeben. Wird ein Messwert für ein Modell gelesen, könnte ein Leseaufruf für alle weiteren ausgelöst werden. Dies würde auch die folgenden Leseaufrufe aus dem Modell vorweg nehmen und auf diese Weise die Zeitdifferenz zwischen den Messwerten senken. Der EIH kann Leseaufrufe auf mehrere Variablen (Nodes) in einem OPC UA-Leseaufruf bündeln [293, Abs. 5.10.2]. Die FDI Communication Server können mehrere Kommunikationsbeziehungen auf einem Netzwerk gleichzeitig verarbeiten [158, Abs. 8.4.8] und dadurch parallele Anfragen bearbeiten. Wenn der FDI Server seine Anfragen ebenfalls bündelt, kann die Gleichzeitigkeit der Messwerte gut realisiert werden.

Problematisch ist die Mehrfachverwendung von Messstellen in mehreren Modellen. Die Lesezyklen der Modelle sind wahrscheinlich unterschiedlich, sodass für das Einhalten der Anforderungen für beide Modelle ohne zu großen Overhead Lösungen gefunden werden müssen. Je nach Anzahl der Mehrfachverwendungen sind unterschiedliche Lösungen denkbar:

**sehr wenig Überschneidung:** Wenn nur ein oder zwei Messgeräte gemeinsam genutzt werden, können diese problemlos mehrfach abgefragt werden (triviale Lösung).

**große Überschneidung, in Clustern auftretend:** Wenn eine geringe Zahl an Apparatemodellen sich eine große Menge Messgeräte teilt, erscheint es sinnvoll, die Leseaufrufe für die Modelle zu synchronisieren und in einem Lesezugriff zu bündeln. Die Anzahl an gleichzeitig unter Einhaltung der Timing-Anforderungen lesbaren Variablen wäre experimentell zu untersuchen.

**flächendeckende Überschneidung:** Bei einer großen Anzahl an Modellen und mehrfach verwendeten Messstellen ist eine allgemeine Lösung nicht zu finden. Es könnten anhand von Zykluszeiten und Mengenmodellen sinnvolle Cluster zum Synchronisieren gesucht werden oder durchgängig alle angeforderten Messwerte in einen Zwischenspeicher gelesen und die Leseaufrufe ausschließlich aus dem Zwischenspeicher bedient werden. Letztere Strategie ist wahrscheinlich für einen FDI Server eine zu große Last. Hier wäre die Anbindung des EIH an ein PIMS und die Abbildung der dynamischen Werte auf die zyklisch übertragenen Werte eine bessere Lösung.

## 4.5 Instanziierungswerkzeug

Um ein FDI Apparatepackage für einen Apparat in Betrieb zu nehmen, sind verschiedene Konfigurationsaufgaben in FDI Server und EIH durchzuführen. Hierfür müssen zusätzlich Daten bereitgestellt und analysiert werden. Um diese Aufgabe zu erleichtern, wurden in Abschnitt 3.3.2 ab Seite 45 ein Konzept zur (Teil-)Automatisierung dieser Vorgänge

erarbeitet. Abbildung 4.21 zeigt die einzelnen Schritte der Instanziierung mit Fokus auf die Ablage der Daten.

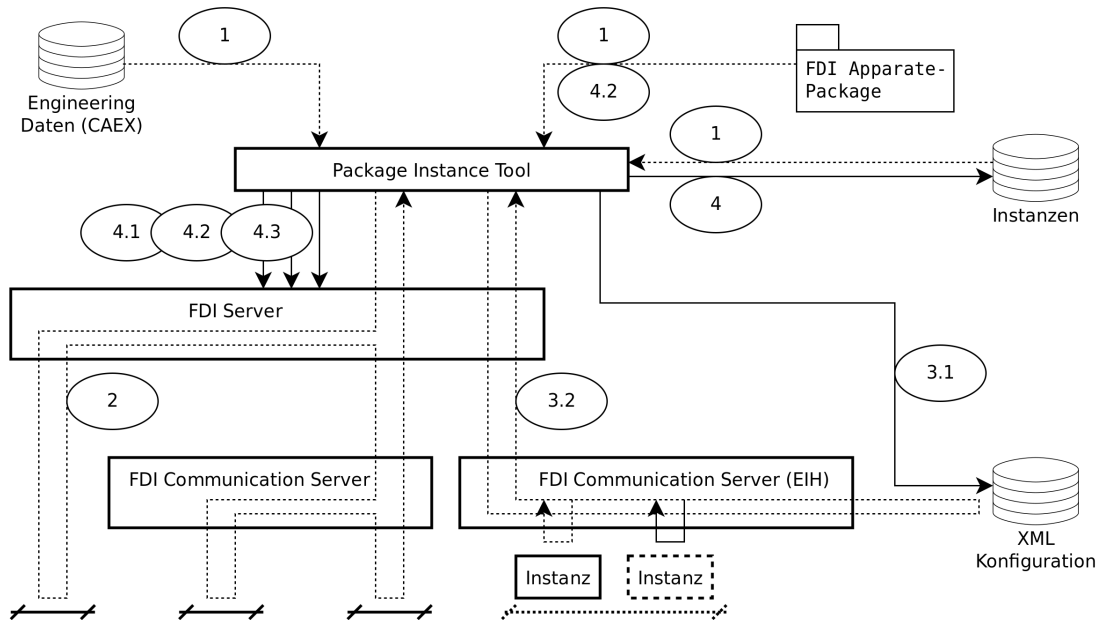


Abbildung 4.21: Schritte bei der Verteilung der Instanzdaten für ein Apparatmodell

Ein Instanziierungswerkzeug soll eine GUI-Unterstützung für die einzelnen Schritte bieten. Es wurde versucht, Softwarekomponenten aus dem Apparatpaket-Werkzeug (siehe Abschnitt 4.3 ab Seite 75) wiederzuverwenden. Dies sorgt auch dafür, dass die Apparatmodelle in beiden Anwendungen gleich dargestellt werden. Abbildung 4.22 auf Seite 96 zeigt einen Überblick über die adaptierten Komponenten des Instanziierungswerkzeugs (vgl. Übersicht des Apparatpaket-Werkzeugs in Abbildung 4.11 auf Seite 78). Zur Integration und Darstellung des Apparatmodells verwendet es ebenfalls das in Abbildung 4.10 auf Seite 77 dargestellte Muster.

Um die Suche nach verwendbaren Messstellen in CAEX-Engineeringdaten durchzuführen, wurde eine auf *XPath* basierende Abstraktion (*CaexQuery*) entwickelt. Die Suchstrategie ist in Abschnitt 4.5.1 beschrieben. Für die Suche nach korrespondierenden Feldgeräten im FDI Server wurde der Java-OPC UA-Stack der Firma *Prosys* integriert und ein enthaltenes Beispiel zu einem Browse-Client erweitert (siehe Abschnitt 4.5.2 ab Seite 100). Die drei *Visitor*-Klassen erstellen die Konfigurationsdateien für den EIH (siehe Abschnitt 4.5.3 ab Seite 100). Die Konfiguration des Apparatmodells im FDI Server wird in Abschnitt 4.5.4 ab Seite 102 beschrieben.

#### 4.5.1 CAEX-Suche und Matching

Eine saubere strukturelle Suche in Engineeringdaten ist möglich, aber sehr aufwendig [294, 71]. Sie stellt Anforderungen an Vollständigkeit der Datensätze und Eindeutigkeit der Kennzeichen [295, 296]. Sehr wichtig ist weiterhin, dass diese formal notiert vorliegen.

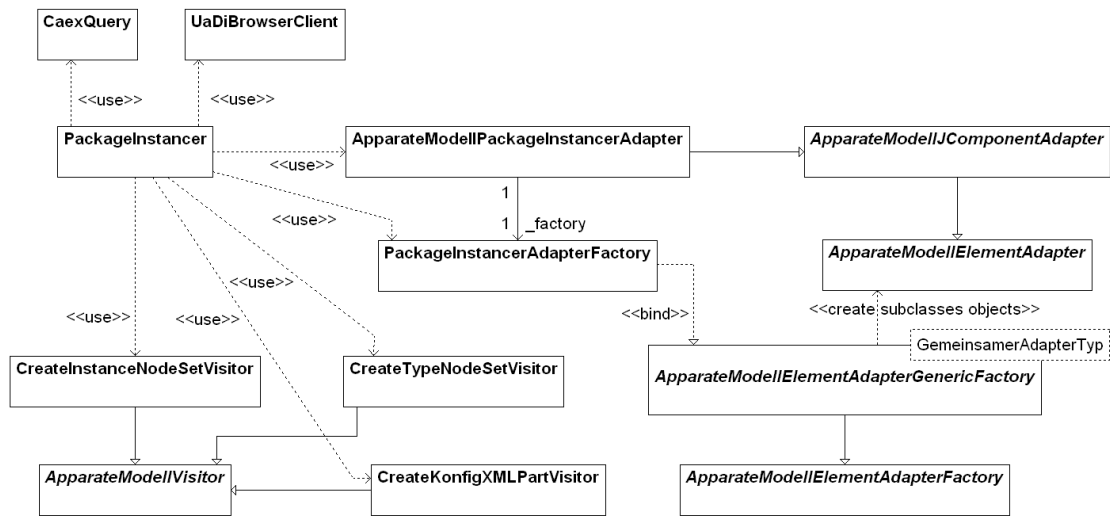


Abbildung 4.22: Einbindung des Datenmodells des Apparatemodells in die GUI des Instanzierungswerkzeugs

Die erste und zweite Anforderung sind methodisch nur schwer nachzubessern, wenn sie nicht im Engineering erfüllt wurde [258, 38]. Die Zweite ist durch einen wohlgeformten und geprüften CAEX-Export umgesetzt (siehe Abschnitt 4.1.1.3 ab Seite 58).

Ein Beispiel für eine solche Suche wurde in [2] beschrieben und ist in Abbildung 4.23 wiedergegeben. Eine vollständige Suche wird gemäß PandIX-Schema folgendermaßen umgesetzt.

1) **Ausgangspunkt** Es gibt zwei Optionen:

**Suche nach dem Apparat:** Ist das Apparatemodell bzw. FDI Apparatepackage, welches instanziiert werden soll, bekannt, werden die Engineeringdaten nach einem passenden Apparat durchsucht. Hierfür wird die `SystemUnitClass` in der CAEX-Bibliothek aus dem Paket geparkt und ihr Name als Suchbegriff verwendet. Gesucht werden muss ausschließlich in der Instanzhierarchie und in Unterelementen von komplexen Systemklassen (z. B. Modulen). Wenn der Name der `SystemUnitClass` im XML-Attribut `RefBaseSystemUnitPath` mit dem Suchbegriff übereinstimmt und dessen Definition in den Engineeringdaten (Datei-Alias) mit der im FDI Apparatepackage gegebenen `SystemUnitClass` übereinstimmt, ist eine Instanz sicher gefunden. Ohne die Übereinstimmung beim Datei-Alias muss der Fund händisch bestätigt werden.

Wurde dabei nichts gefunden oder soll weitergesucht werden, kann die `SupportedRoleClass` der `SystemUnitClass` im Apparatepaket gesucht werden. Im gleichen Suchkontext im Element `RoleRequirements` wird nach Attributen `RefBaseRoleClassPath` mit demselben Wert von `RefBaseRoleClassPath` in der `SupportedRoleClass` gesucht. Fundstellen, welche diese alternative Methode hervorgebracht hat, müssen händisch überprüft werden.

**Suche nach dem Apparatemodell:** Ist der Apparat bekannt, wird dieser über seine Referenzkennzeichnung in den Engineeringdaten gesucht. Ist dort der `RefBase-`



```

[...]
```

The diagram shows XML code for a CAE tool instance hierarchy. It consists of several blocks of code, each starting with an ellipsis [...].

- The first block is an `<InternalElement Name="HE2">` element. Its `RefBaseSystemUnitPath` attribute is set to `"HE_example_lib@SystemUnitLib_HE_example/HE_example"`. It contains a `<RoleRequirements RefBaseRoleClassPath="PandIX_PPE@PPE_BasicLib/HeatExchangerRequest" />` child element.
- The second block is an `<InternalElement Name="FI27.1">` element. It contains an `<ExternalInterface Name="P" RefBaseClassPath="PandIX_ICL@PandIX_ICL/SensorProcessInterface">` child element.
- The third block is an `<InternalLink Name="link_HE2_Pipe26">` element. Its `RefPartnerSideA` is `"HE2:PIIn"` and its `RefPartnerSideB` is `"Pipe26:POut"`.
- The fourth block is an `<InternalLink Name="link_FI27.1_Pipe26">` element. Its `RefPartnerSideA` is `"[FI27.1]:P"` and its `RefPartnerSideB` is `"Pipe26:M1"`.

Arrows in the diagram indicate the following relationships:

- An arrow points from the `RefBaseSystemUnitPath` attribute of the `HE2` element to the `RefBaseClassPath` attribute of the `ExternalInterface` element.
- An arrow points from the `RefPartnerSideA` attribute of the `link_HE2_Pipe26` element to the `RefPartnerSideA` attribute of the `link_FI27.1_Pipe26` element.
- An arrow points from the `RefPartnerSideB` attribute of the `link_HE2_Pipe26` element to the `RefPartnerSideB` attribute of the `link_FI27.1_Pipe26` element.
- An arrow points from the `RefPartnerSideA` attribute of the `link_FI27.1_Pipe26` element to the `RefBaseClassPath` attribute of the `ExternalInterface` element.

```

[...]
```

**Abbildung 4.23:** Teile einer aus einem CAE-Tool exportierten Instanzhierarchie (korrigierte Darstellung aus [2])

`SystemUnitPath` angegeben, wird das FDI Apparatepackage gesucht, welches diese Klasse definiert, und ebenfalls der Datei-Alias untersucht.

Ist das Attribut nicht angegeben, kann ebenfalls auf Rollenübereinstimmung in den `RoleRequirements` und der `SupportedRoleClass` geprüft werden. Funde sind, wie in der ersten Option, zu überprüfen.

- 2) **Suche nach den Anschlüssen** Je nach Definitionsstil sind die Anschlüsse eines Apparates polymorph aus der Rolle abgeleitet (objektorientiert) oder in der Definition der `SystemUnitClass` wiederholt (AML-konform). Für beide Varianten ist sicherzustellen, dass die im Apparatemodell referenzierten Anschlüsse definiert und richtig zugeordnet werden können. Dies sollte – durch den Workflow bedingt – über den Bezeichner eindeutig möglich sein.
- 3) **Suche nach den Messstellen entlang der Flusswege** Von den Anschlüssen aus werden nun Messstellen gesucht, die auf dem Flussweg des durch den Anschluss in den Apparat ein- oder austretenden Mediums liegen. Hierbei kann bis zu anderen Apparaten, Rohrverzweigungen oder Mehrwegeventilen an Rohrverbindungen navigiert werden, ohne dass die Messwerte am Messgerät von denen, die am Apparat gemessen würden, abweichen. In bestimmten Fällen, z. B. bei der Suche nach einer Durchflussmessstelle bei flüssigen Medien, kann auch über Pumpen hinweg gesucht werden. Andererseits sind bei Anforderungen an Druckmessstellen Höhenunterschied und Druckabfall in Leitungen schnell nicht mehr aus Anlagendaten allein zu bestimm-

men, was den Suchbereich erheblich einschränkt. Je nach Abstand und Durchsatz ist mit einer negativen (Eintritt) oder positiven (Austritt) Totzeit zu rechnen.

In CAEX-Exporten wird die Suche durch `ProcessConnectionPoints` hindurch geführt. Ziel ist es, die `ProcessConnectionInterfaces`, insbesondere die mit der konkreten Klasse `MeasurementPoint`, zu finden, an denen Messstellen angeschlossen sein können. Das Funktionskennzeichen (bzw. PCE-Kategorie) der angeschlossenen Messstelle wird mit den Anforderungen auf Flussweg dieses Anschlusses im Apparatemodell verglichen. Wird sie benötigt, wird ihre Referenzkennzeichnung ausgelesen und als Suchergebnis vermerkt. Technisch wird diese Suche mit den folgenden Schritten ausgeführt:

**Navigieren über (Rohr-)Verbindungen** Das Ausgangsobjekt *O1* befindet sich in einer beliebigen Unterebene einer Instanzhierarchie. In der Elternebene wird nach einem `InternalLink` gesucht, bei dem die `RefPartnerSideA` oder die `RefPartnerSideB` einen relativen Link zu dem betrachteten Anschluss (einem `ExternalInterface`) *A1* besitzt:

```
<InternalLink RefPartnerSideA="O1:A1" RefPartnerSideB="O2:A2">
```

Existiert ein solcher Link nicht, kann eine Ebene oberhalb unter Angabe des direkten Eltern-`InternalElement` von *O1* – genannt *P1* – ein `InternalLink` mit einer Referenz *P1.O1:A1* gesucht werden. Dieses Vorgehen lässt sich in der Hierarchie nach oben fortführen (*PP1*, *PPP1*, ...). Kann bis zum Wurzelement der Instanzhierarchie keine passende Referenz gefunden werden, ist keine Verbindung vorhanden.

In der entsprechenden anderen `RefPartnerSide` des gefundenen `InternalLinks` ist der Name des verbundenen Objektes *O2* und der verbundene Anschluss *A2* sowie gegebenenfalls die Elemente der Elternhierarchie *P2*, *PP2.P2*, ... vorhanden.

**Finden aller definierten Prozessanschlüsse** Das verbundene Objekt hinter einer Verbindung *O2* ist wiederum ein Prozessraum, an dem sich Anschlüsse befinden. Diese werden wie oben beschrieben bestimmt. Der Suchmenge ist zunächst der Anschluss *A2*, von dem aus das Objekt erreicht wurde, zu entnehmen. Alle weiteren Anschlüsse können auf ein angeschlossenes Messgerät oder ein weiteres Objekt untersucht werden. Wenn mehrere `ProcessConnectionPoints` verblieben sind, muss die Suche abgebrochen werden, da eine Trennung oder Zusammenführung von Flusswegen vorliegt.

Eine Ausweitung wäre möglich, wenn man über die an allen verbleibenden Pfaden vorhandenen Messgeräte mittels Bilanzgleichungen auf einen virtuellen Messwert schließt [211] oder Verschaltungen von Absperrventilen untersucht. Für letzteren Fall können dynamische Abhängigkeiten modelliert zur Laufzeit ausgewertet werden, wie z. B. Flusswege an Drei-Wege-Ventilen [257] oder Verteiler-Ventilgruppen mit gegenseitigem Ausschluss.

An gefundenen Anschlüssen vom Typ `MeasurementPoint` wird mit dem selbem Mechanismus wie dem zum Verfolgen von Stoffverbindungen ein angeschlossenes Messgerät bestimmt.

```
<InternalLink RefPartnerSideA="02:M1"
  RefPartnerSideB="[FIC27.1]:P">
```

Der entsprechende `InternalLink` referenziert den Anschluss  $P$  vom Typ *Sensor-ProcessInterface*. Das entsprechende Objekt *FIC27.1* ist ein Messgerät dessen XML-Attribut `Name` das eindeutige Referenzkennzeichen enthält und dessen CAEX-Attribute `FunctionCode` und `ReferenceCode` den Typ des Messgerätes festlegen. Fehlende Informationen können auch an dessen Messwertanschluss *Out* vom Typ `SignalSource` ausgelesen werden. Eine weiterführende Standardisierung von CAEX-Attribut-Bibliotheken erleichtert dabei die Interpretation der im Engineering festgelegten Eigenschaften.

**Blattübergänge** Sind Anschlüsse an explizit in der Wurzelebene der Instanzhierarchie deklarierte `ExternalInterfaces` angeschlossen, liegt eine Verbindung zu einem anderen Blatt eines R&I-Fließbildes oder in die nächsthöhere Strukturierungsebene der Anlage vor. In beiden Fällen kann die Suche in der referenzierten Instanzhierarchie fortgesetzt werden. Im zweiten Fall ist sie unter Umständen nicht praktikabel, da ebenfalls eine andere, von der vorherigen unabhängige Vernetzungsstruktur referenziert werden könnte.

Sind keine sauber definierten Engineeringdaten vorhanden, kann zumindest auf verschiedene Messgeräte-Typen gefiltert werden. Auch solche Filter können auf Basis von CAEX bzw. AML implementiert werden [10].

Im Instanziierungswerkzeug wird die im ersten Schritt (Abbildung 4.21 auf Seite 95) bereitgestellte (oder aus früheren Suchen noch bekannte) CAEX Datei untersucht. Alle XML-Abfragen sind mittels *XPath*-Ausdrücken umgesetzt. Quelltext 4.17 und Quelltext 4.18 zeigen Beispiele für solche Ausdrücke. Bekannte Apparatemodellinstanzen aus dem Speicher werden nicht erneut gefunden. Wenn ein Apparat gefunden wurde, wird die beschriebene Suchstrategie schrittweise mittels der dynamisch angepassten *XPath*-Ausdrücke ausgeführt. Die Ergebnisse werden als Abbildung von Anforderung auf gefundene Messstellen an den nächsten Schritt übergeben.

**Quelltext 4.17:** Code zum Suchen nach definierten Rollenklassen mittels *XPath*

```
for(String rcfile: roleClassLibraries)
{
    CaexQuery roleclass = new CaexQuery();
    roleclass.setCaexFile(rcfile);
    List<String> rcs = roleclass
        .queryXPathToConcreteXPaths("/CAEXFile/RoleClassLib//RoleClass");
    for(String rc: rcs)
    {
        String name = roleclass.queryXPathToCAEXClassName(rc);
        queryRCForRole.put(name, new Tupel<String, CaexQuery>(rc, roleclass));
        System.out.println("RC: " + name + " -> " + rc);
    }
}
```

**Quelltext 4.18:** Code zum Suchen nach `InternalLinks`, die eine bestimmte Schnittstelle eines bestimmten `InternalElements` verbinden, mittels *XPath*

```
List<String> ilsA = engineering.queryXPathToConcreteXPaths(
    "/CAEXFile/InstanceHierarchy//InternalElement/InternalLink"
    + "[@RefPartnerSideA=\"\" + ieName + \":\" + extIfName + "\"]")
```

```
);  
List<String> ilsB = engineering.queryXPathToConcreteXPaths(  
    "/CAEXFile/InstanceHierarchy//InternalElement/InternalLink"  
    + "[@RefPartnerSideB=\"\" + ieName + \":\" + extIfName + \"\"]"  
);
```

### 4.5.2 Node-Suche nach Prozesswert

Nachdem die Messstellenanforderungen einer Apparatemodellinstanz zu konkreten Feldgeräten zugeordnet wurden, ist der Zugriff auf deren Prozesswerte einzurichten. Dabei kommt ein Vorteil der FDI-Technologie zum Tragen, der auch die Entscheidung für diese Technologie stark beeinflusst hat. Der FDI-Server stellt ein Informationsmodell der verwalteten Feldgeräte über eine offene Schnittstelle (OPC UA) zur Verfügung. Diese Schnittstelle ist von Kommunikationstechnologie und Anlagenarchitektur unabhängig. Somit kann der Zugriff auf die Prozesswerte mit dem Wissen über Gerätestrukturen robust automatisiert werden. Das Instanzierungswerkzeug verwendet hierfür einen Standard-OPC UA-Client.

Die DIN EN 61804-2 [119] gibt mit den Definitionen für Block und Parameter die Struktur des Informationsmodells vor. In den Blöcken kann gemäß dem OPC UA-DI-Informationsmodell [190] eine flache Liste aller enthaltenen Parameter ausgelesen werden (Browse-Funktion von OPC UA). Die Information für die Bestimmung der Semantik der OPC UA-Nodes im Informationsmodell der Feldgeräte kann aus Profilen gewonnen werden [8, 199]. Auch Standardimplementierungen, wie in der NE 131 [297] beschrieben, können einen einheitlichen Zugriff vorschreiben. [122] beschreibt ebenfalls ein Modell zum Zugriff auf den richtigen Wert. In [292, Abs. 4.2, S. 55] wird der Zugriff für die drei für FDI relevanten Kommunikationstechnologien FF, HART und PROFIBUS im Einzelnen beschrieben.

Im Instanzierungswerkzeug wurde die Node-Suche für PROFIBUS PA implementiert. Die Suche erfolgt aufgrund der Verallgemeinerung der Festlegungen in [116, Abs. 15 und 16, 19 und 20, 23 und 24, 27 und 28 sowie 32]. Die in den verschiedenen Gerätetypen angegebenen Strukturen und Parameternamen gleichen sich zu großen Teilen, sodass nur wenige Typ-Abfragen umgesetzt werden mussten, um die korrekten Variablen-Nodes zu bestimmen. Insbesondere bei den in der Prozessindustrie üblichen Standardmessgeräten [116, Abs. 16] kann immer auf den OUT-Wert des *AnalogInputFunctionBlocks* [116, Abs. 15.2] zugegriffen werden.

### 4.5.3 Übergabe der Konfiguration an den EIH

Der EIH hat die Aufgaben, den Zugriff auf die Feldgeräte mit den Prozesswerten zu vermitteln und den Gerätespeicher für die virtuellen Apparate-Feldgeräte umzusetzen und persistent abzulegen. Diese Konfigurationsdaten ergeben sich aus der EDD, welche vom Apparatemodell im Apparatepakete-Werkzeug abgeleitet wurde. Da das Apparatemodell ebenfalls im FDI Apparatepackage abgelegt wurde, kann das Instanzierungswerkzeug eine Konfigurationsdatei auch direkt aus dem Modell erzeugen.

Die Konfigurationsdatei folgt dem XSD-Schema in Anhang B, Abschnitt B.3 ab Seite 182. Sie speichert die Werte der Variablen des Informationsmodells im FDI Communication

Server des EIH. Diese besitzt die gleiche hierarchische Struktur wie die EDD des Apparatemodells. Deswegen konnte die Export-Komponente (ein Apparatemodell-Visitor) für diese XML-Datei aus dem EDD-Exporter des Apparatepaket-Werkzeuges weiterentwickelt werden. Dessen Ausgabe wird in die Gesamt-Konfigurationsdatei mit allen Instanzen integriert.

Zur Konfiguration des FDI Communication Servers des EIH werden für jeden Apparatemodell-Typ und für jede Apparatemodell-Instanz je eine NodeSet-XML-Datei verwendet. Diese ähnelt dem Feldgerätemodell in Abschnitt 4.1.3 ab Seite 64. Sie wird aus dem Instanziierungswerkzeug durch einen speziellen *Visitor* erzeugt. Die Struktur ist deutlich komplexer als die der Konfigurations-XML-Datei oder der EDD des Apparatemodells mit den Daten, da alle Referenzen zwischen den Objekt- oder Variablen-Nodes explizit als Referenz-Nodes angelegt werden müssen. Dabei war zu beachten, dass die Menge an OPC UA-NodeIDs innerhalb eines Namensraums bei mehreren Modellen nicht ausreichen könnte. Ebenfalls müssten bei der unabhängigen Erstellung der NodeSet-XML-Dateien Dopplungen verhindert werden. Deswegen wurde ein separater Namensraum für jede erzeugte NodeSet-XML-Datei erstellt. Dies hat auch den Vorteil, dass die Modelle im FDI Communication Server einzeln geladen und entladen werden können. Um den Namen des Namensraums eindeutig wählen zu können, wurden die eindeutige Referenzkennzeichnung des Apparates aus den Engineeringdaten verwendet. Der Namensraum für den NodeSet mit dem Apparatetyp wurde aus der Typ-Bezeichnung aus dem FDI Apparatepackage abgeleitet. Die Struktur der Konfigurations-XML-Datei und der NodeSet-XML-Dateien ist in Abbildung 4.24 dargestellt. Die PROFIBUS-Datenadressen Slot und Index in den Variablen werden absolut angegeben, um die Berechnung des Funktionsblock-Layouts im PROFIBUS-Adressraum des Apparatemodells (VApparat) nicht im EIH ausführen zu müssen. Deswegen werden die Daten des Device Directorys einmal berechnet und statisch als Hex-String in DDEintrag-Elementen abgelegt.

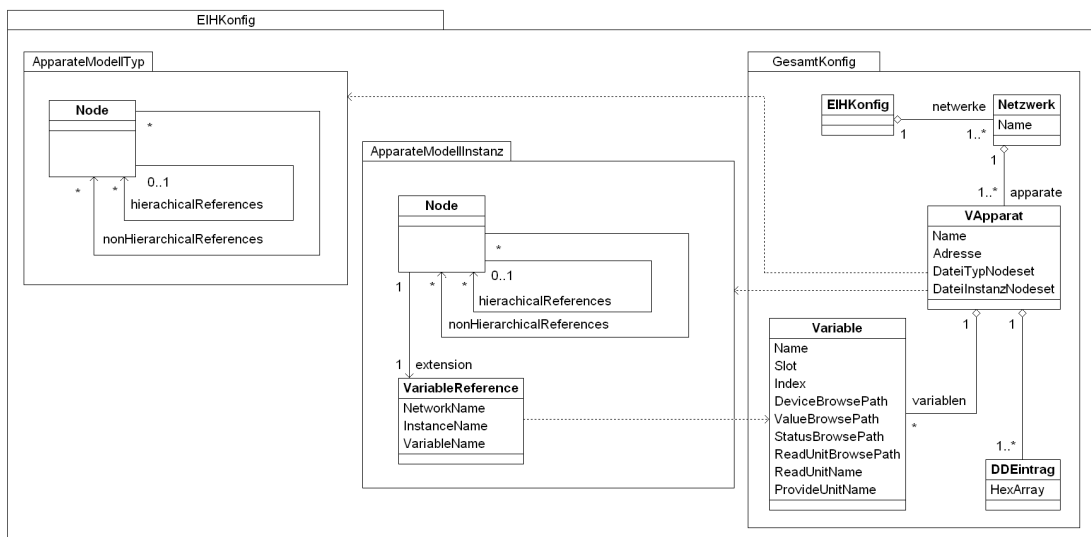


Abbildung 4.24: Verteilung der Konfigurationsartefakte des Equipment Instance Handlers

Die Übertragung der Konfigurationsdateien auf den EIH wurde in der vorliegenden Umsetzung händisch vorgenommen. Eine automatische Übertragung ist wie in [6] beschrieben möglich. Um die aktualisierte Konfiguration zu übernehmen, wird über die OPC UA-Schnittstelle des FDI Communication Servers des EIHs in der entsprechenden Node die `Scan`-Methode aufgerufen. Dies kann auch automatisch vom Instanziierungswerkzeug durchgeführt werden.

### 4.5.4 Instanz-Konfiguration im FDI Server

Zur Integration einer Instanz eines FDI Apparatepackages sind im FDI Server drei Schritte erforderlich (4.1–4.3 in Abbildung 4.21 auf Seite 95), bevor online auf diese Instanz zugegriffen werden kann. Die Einrichtung des FDI Communication Servers des EIHs ist nur dann notwendig, wenn ein neuer EIH verwendet wurde oder in einem bestehenden EIH ein neues Netzwerk angelegt wurde. Das Instanziierungswerkzeug legt die Struktur von EIH und FDI Communication Server fest und gibt sie allen Komponenten bekannt. Der FDI Communication Server wird dem FDI Server mit dem Aufruf des `Services AddReferences` (4.1) eingerichtet. Dieser wird über die OPC UA-Schnittstelle gerufen. Die Integration ins Informationsmodell geschieht dann automatisch.

Der zweite Schritt (4.2) ist die Suche nach der neuen Geräteinstanz im Netzwerk. Wenn diese im optionalen ersten Schritt nicht automatisch gesucht wurde, wird die Suche ebenfalls über die OPC UA-Schnittstelle ausgelöst. Im Informationsmodell wird die Node gesucht, die das Netzwerk repräsentiert, welches die gesuchte Instanz enthält. Auf dieser wird die Methode `Scan` ausgeführt. Wenn die aktualisierte Konfiguration des EIH angewendet wurde, muss das neue Gerät vom FDI Communication Server des EIH gemeldet und somit vom FDI Server gefunden werden. Das FDI Package des Apparatemodells wird geladen und das Informationsmodell instanziiert.

An dieser Stelle entsteht unter Umständen ein Abhängigkeitsproblem. Die Instanz mit allen *CONTAINED* und *DYNAMIC* Parametern ist bereits betriebsbereit. Die *LOCAL* Parameter, insbesondere die *\*\_IS\_VALID*-Variablen, welche bestimmte, nicht definierte Parameter aus den Berechnungsmethoden ausblenden, sind noch nicht korrekt eingestellt. Sollte der FDI Server sofort nach der Erstellung des Informationsmodells in den *ONLINE*-Modus wechseln, könnte es undefiniertes Verhalten geben. Dies ist von der Implementierung des FDI Servers abhängig. Der Reference-Host [259] besitzt dieses Verhalten nicht.

Der dritte Schritt besteht dann aus dem Setzen der *LOCAL* Parameter im Informationsmodell des FDI Servers mittels der OPC UA-Schnittstelle. Danach kann sicher in den *ONLINE*-Modus gewechselt werden.

## 4.6 Beispiele und Evaluation

Abschließend soll anhand von zur Verfügung stehenden Engineeringdaten die Anwendung der vorgestellten prototypischen Lösung dokumentiert und bewertet werden. In Abschnitt 4.6.1 werden Beispiel-Apparatemodelle mit ihren Anforderungen beschrieben. Dazu wird auf weiterreichende Anforderungen an das Apparatepaket-Werkzeug und die Berechnungsmethodik eingegangen. Abschnitt 4.6.2 ab Seite 106 behandelt die Möglichkeit, die zuvor beschriebenen Modelle in Anlagen tatsächlich instanziiieren zu können. In Abschnitt 4.6.3 ab Seite 115 wird die Möglichkeit diskutiert, die im Apparatemodell

berechneten Kennwerte in Tools einbinden zu können. Abschnitt 4.6.4 ab Seite 115 beschreibt die erwarteten Einflüsse der Lösung auf die verwendeten Kommunikationskanäle.

#### 4.6.1 Generierung von Modellen bzw. Paketen

Im Folgenden werden fünf Beispiele für Grundoperationen als Apparatemodell im Apparatepakete-Werkzeug umgesetzt. Wie in anderen Lösungen [66, Abs. 7] werden die Berechnungsfunktionen für Kennzahlen über bekannte Zusammenhänge hergeleitet und allgemein notiert. Die Anwendungsregeln werden implizit durch die Messstellen-Anforderungsmuster festgelegt. Als Beispiele wurden hinreichend verbreitete oder offensichtliche Modelle ausgewählt, da für speziellere Modelle vertiefte verfahrenstechnische Kenntnisse erforderlich wären (siehe Abschnitt 3.4 ab Seite 47).

Unter den Modellen befinden sich auch zwei der drei Modelle, die in Literatur oft beschrieben und in PLS-Produkten implementierte wurden (Wärmeübertrager und Pumpe, Abschnitt 2.3.2 ab Seite 30, [198]). Auf das Ventilmodell wurde verzichtet, da in der Infrastruktur die Verfahrensgeschwindigkeit zur Erkennung von Schwergängigkeit nicht genau bestimmt werden kann. Eine vollständige Blockade (detektiert durch einen Fluss trotz geschlossenem Ventil oder einem dauerhaften Unterschied zwischen Stellwert und Position) sollte ohnehin schon durch einen im Pumpenantrieb implementierten Algorithmus mit einer Abbildung auf einen Alarm abgedeckt sein.

#### Wärmeübertrager

Das erste Beispiel (Abbildung 4.25) ist das Wärmeübertragermodell aus Abschnitt 4.3 ab Seite 75. Es gehört zu den verbreiteten Standard-Apparatemodellen. Anhand der Masse- oder Volumenströme sowie der Ein- und Ausgangstemperaturen der beiden Produkträume kann die Effektivität des Wärmeübertragers bestimmt werden. Auch Zwischenergebnisse wie die Temperaturdifferenzen können als Kennzahlen verwendet werden.

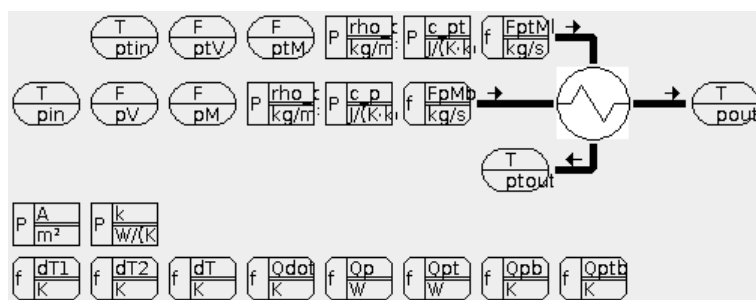


Abbildung 4.25: Darstellung des Apparatemodells für einen Wärmeübertrager (aus [2])

#### Kolonnen mit Festbetschüttung

Aus dem Instanziierungsbeispiel (siehe Abschnitt 4.1.1.4 ab Seite 61 bzw. Abschnitt 4.6.2 ab Seite 106) wurden zwei Gas-Flüssigkeits-Sorptionskolonnen übernommen. Die erste besitzt einen Zufluss und die zweite zwei Zuflüsse für die flüssige Phase. Beide Modelle bestimmen das Gegenstromverhältnis zwischen Gas- und Flüssigphase. Ebenfalls kann

die Druckdifferenz über der Festbettschüttung als Kennzahl ausgewertet werden. Die Abbildungen 4.26 und 4.26 zeigen die Modelle im Apparatpaket-Werkzeug.

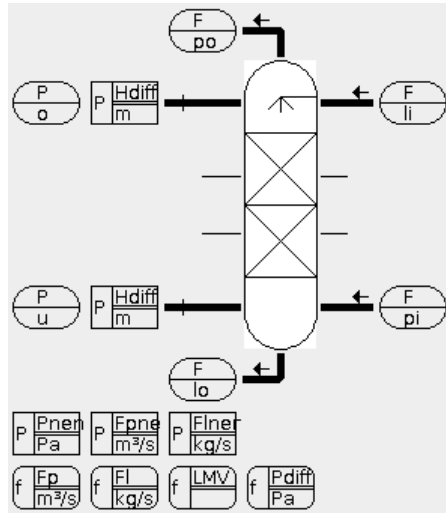


Abbildung 4.26: Darstellung des Apparatmodells für eine Sorptionskolonne (1)

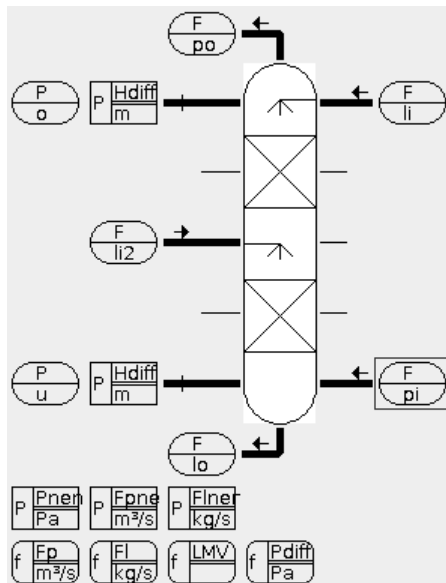


Abbildung 4.27: Darstellung des Apparatmodells für eine Sorptionskolonne (2)

Im Gegensatz zum Apparatmodell des Wärmeübertragers können im ersten Modell die Flüsse sowohl an Einlässen wie auch an den Auslässen bestimmt werden. Sie sind also flexibler instanziiert. Im zweiten Kolonnenmodell kann für den Flüssigkeitsstrom die Summe der Eingangsflüsse oder der Ausgangsfluss verwendet werden. Aufgrund der Möglichkeit eines Rückstaus im Sumpf sind die Eingangsflüsse zur Messung zu bevorzugen.



## Pumpe

Ein weiteres Standardmodell ist das Pumpenmodell in Abbildung 4.28. Es werden drei Kennzahlen bestimmt. Es kann eine Unterschreitung des minimalen Saugdrucks detektiert werden. Die Druckdifferenz zwischen Saug- und Druckseite kann ebenfalls ausgewertet werden. Weiterhin kann das Verhältnis von Förderleistung und Drehzahl überwacht werden.

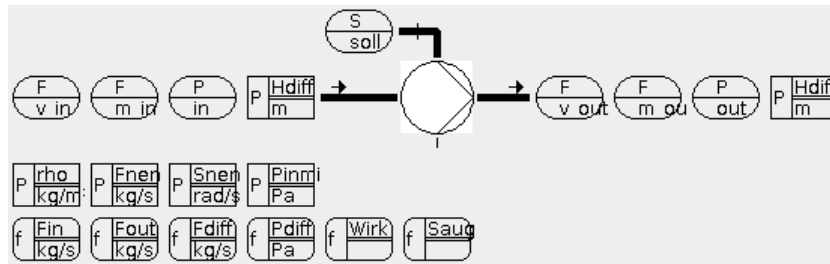


Abbildung 4.28: Darstellung des Apparatemodells für eine Pumpe

## Rohrheizung

Ein weiteres Modell, das aus dem Instanziierungsbeispiel abgeleitet wurde, ist eine Rohrheizung. Diese wird üblicherweise nicht als dedizierter Apparat aufgefasst, sondern ist ein Rohr, welches mit Heizelementen und Isolierung versehen wird. Mit dem Modell wird die Erkennung der Überhitzung bei zu geringem Durchfluss ermöglicht. Als einfacher Fall kann auch eine zu hohe Temperatur anhand eines einstellbaren Schwellwertes gemeldet werden (z. B. der Maximaltemperatur der Elektro- und Wärmeisolierung).

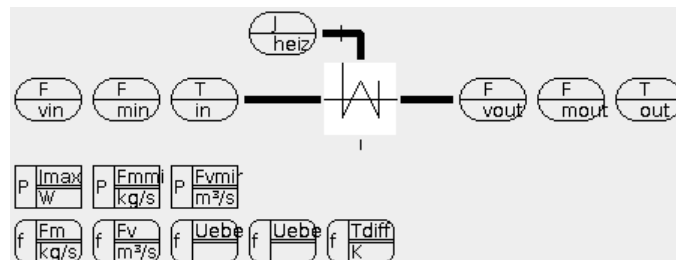


Abbildung 4.29: Darstellung des Apparatemodells für eine Rohrheizung

## Zusammenfassung

Während der Erstellung der Beispiel-Apparatemodelle wurden hilfreiche Features und Erweiterungsmöglichkeiten erfasst und folgend kurz zusammengefasst. Die Erstellung der Apparatesymbole in der implementierten Form ist nicht für den Produktivbetrieb geeignet. Die Variante wurde gewählt, um keine komplexen Nutzerinteraktionen mit graphischen Elementen umsetzen zu müssen. Ihre Bedienung ist deswegen zu umständlich. Die Verwendung von Rollenklassen und Standardattributen reduziert den Aufwand zur Modellierung von Anschlüssen erheblich. Dagegen ist der Formeditor eine Beschränkung für die Vielfalt und Flexibilität der Modelle. Die Berechnungsformeln in der bestehenden

Form können keine Alternativen abbilden, da *ext4j* ausschließlich arithmetische Ausdrücke prüfen kann. Für die Auswahl zwischen wahl-obligatorischen Messwerten bzw. Zwischenergebnissen (z. B. einem gemessenem Massenstrom und einem aus einem gemessenem Volumenstrom und der Dichte berechnetem Massenstrom) müsste die *ext4j*-Grammatik zumindest *Conditionals* unterstützen. Diese müssten dann jedoch in EDD-IF-Konstrukte übersetzt werden. Positiv kann herausgestellt werden, dass die Maßeinheitenprüfung zur schnellen Kontrolle von eingegeben Formeln beiträgt. Eine Erweiterungsmöglichkeit wäre, Sätze von Instanziierungsszenarien und für diese Sätze von Eingangsgrößen zu laden, um ein Modell zur Entwicklungszeit systematisch testen zu können.

### 4.6.2 Instanziierung

Um die Abdeckung der Berechnungsfunktionen mit tatsächlichen Messstellen in den Apparatemodellen einschätzen zu können, wurden die im vorherigen Abschnitt definierten Beispielmole instanziiert. Dafür sind in CAEX notierte Engineeringdaten notwendig, in denen das Instanzierungswerkzeug die Apparateinstanzen finden und passende MSR-Anforderungen zuordnen kann.

Die in [274, Bild 4.2, S. 54] beschriebene Anlage wurde anhand des Bedienbildes in *Dia* nachgebildet und mittels *Dia2CAEX* [4, 3] (Abschnitt 2.1.2 ab Seite 9) in eine CAEX-Datei gewandelt. Das Original-R&I-Fließbild und die detailliertere Nachbildung sind in Abschnitt C.1 ab Seite 185 abgebildet. Zusätzlich wurde die Versuchsanlage des Lehrstuhls Prozessleittechnik der RWTH Aachen (*ACPLT Pumpwerk*) [298] herangezogen. Da darin die CAEX-Klassen codiert vorlagen, wurden die instanziierten Pumpenmodelle lediglich theoretisch diskutiert.

#### 4.6.2.1 Instanzen in der $CO_2$ -Absorptionsanlage

In der  $CO_2$ -Absorptionsanlage wurden je eine oder mehrere Instanzen für alle fünf in Abschnitt 4.6.1 ab Seite 103 beschriebenen Apparatemodelle gefunden. Die Abdeckung mit Messstellen wurde tabellarisch und graphisch dokumentiert. Messstellen, die genutzt werden könnten, aber mit den implementierten Methoden nicht gefunden wurden, sind in den Tabellen beschrieben und in den Grafiken als nicht gefunden dargestellt. In den Abbildungen sind die automatisch gefunden Messstellen grün und alle anderen rot. Die Parameter sind grün hinterlegt, wenn ein Wert angegeben wurde. Auch die Abhängigkeiten sind in den Grafiken derart gekennzeichnet. Die Funktionen sind jedoch gemäß der alternativen Berechnungswege (Abschnitt 4.6.1, letzter Absatz) eingefärbt, die mit den Modellen in der Implementierung nicht abgebildet werden können.

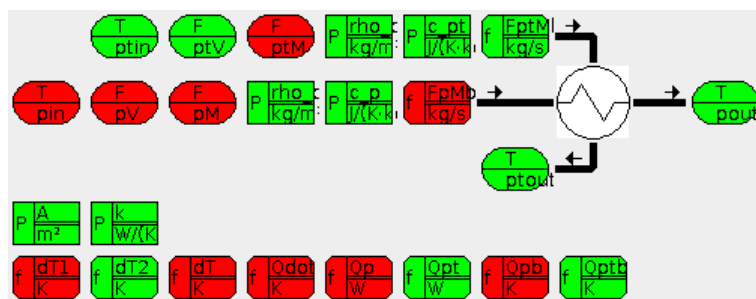
**Wärmeübertrager** Es wurden zwei Wärmeübertrager (*W 1* und *W 5*) in den Engineeringdaten der Anlage erkannt. Mit den gefundenen Wärmeübertragern in den Teilanlagen 5 und 6 wurde versucht, das Wärmeübertragermodell zu instanziiieren. *W 1* (Tabelle 4.4 und Abbildung 4.30) ist nicht voll ausgestattet. Die Effektivität kann deswegen nicht berechnet werden. Jedoch kann der Wärmeübergang anhand des zweiten Produktraums und dessen Temperaturgefälle bestimmt werden. *W 5* (Tabelle 4.5 auf Seite 108 und Abbildung 4.31 auf Seite 108) besitzt alle notwendigen Temperatur-Messstellen. Hierbei fehlen jedoch die

Durchflusssensoren, weswegen Wärmeübergang und Effektivität nicht bestimmt werden können. Nur die Temperaturgefälle in beiden Produkträumen können ausgewertet werden.

Der Kopfkühler  $W 3$  der Kolonne in Teilanlage 8 und der Kondensator und  $W 7$  in Teilanlage 10 könnten strukturell ebenfalls passen. Diese Kandidaten wurden nicht weiter betrachtet, weil die darin vonstatten gehenden Phasenübergänge im Modell nicht abgebildet sind. Aufgrund des Matchings nach SystemUnitClasses und RoleClasses werden diese scheinbaren Kandidaten auch nicht gefunden.

Tabelle 4.4: Instanziierung  $W 1$ 

Messstellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
T_pin	TI 5.10	nein / unter semantischer Kenntnis des zwischengelagerten Bypass-Ventils
F_pV	(NC 5.1)	nein / über berechneten Volumenstrom der Pumpe
F_pM	(NC 5.7)	nein / über berechneten Volumenstrom der Pumpe
T_pout	TIC 6.5	ja
T_ptin	TIC 6.2	ja
F_ptV	FIC 6.3	ja
F_ptM	-	nein / wegen F_ptV nicht notwendig
T_ptout	TIC 6.4	ja

Abbildung 4.30: Instanziierung  $W 1$ 

**Kolonnen** Aufgrund der Ausführung als zwei parallele Kolonnen ( $K 1$  und  $K 2$ ) in Teilanlage 3 ist eine Anwendung des Modells in seiner einfachen Form nicht möglich, da die Aufteilung der Stoffströme nach den Messungen geschieht. Die Bestimmung des Differenzdrucks bringt durch die Ausstattung mit einem Differenzdruckmessers ebenfalls keinen Mehrwert. Wendet man das Modell auf beide Instanzen als einen Apparat an, sind ebenfalls wenige Messstellen instanziiierbar. Zusätzlich gibt es das Problem, dass beide Instanzen konkurrierende Werte für die Drücke bzw. Druckdifferenzen liefern.

Bei der Instanziierung des zweiten Kolonnenmodells in der Beispielanlage können ebenfalls nicht alle angeforderten Messstellen gefunden werden ( $K 3$ , Tabelle 4.6 auf

Tabelle 4.5: Instanziierung *W 5*

Messtellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
T_pin	TIC 5.4	ja
F_pV	-	nein / wegen F_pM nicht notwendig
F_pM	(FIC 5.1)	nein / unter semantischer Kenntnis des zwischengelagerten Bypass-Ventils
T_pout	TI 5.9	ja
T_ptin	TIC 5.8	ja
F_ptV	(NC 5.7)	nein / über berechneten Volumenstrom der Pumpe
F_ptM	(NC 5.7)	nein / über berechneten Massenstrom der Pumpe
T_ptout	TIC 5.10	ja

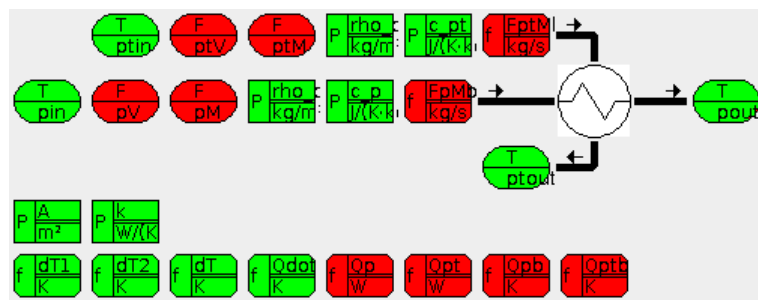
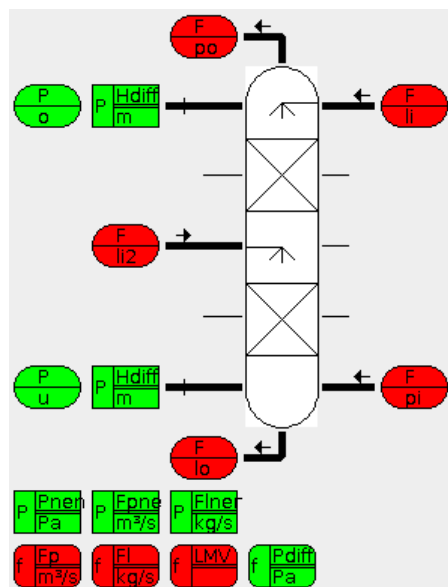


Abbildung 4.31: Instanziierung *W 5*

Seite 109 und Abbildung 4.32 auf Seite 109). Bis auf die Druckdifferenz, die ohnehin von einem Differenzdruckmesser gemessen wird, ist kein Kennwert bestimmbar. Auch mit den indirekt nutzbaren Durchflussmessstellen kann keine weitere Kennzahl berechnet werden, da eine von ihnen nur bedingt benutzbar ist, eine durch die Mischstelle ungenau ist und eine ganz fehlt.

Tabelle 4.6: Instanziierung  $K_3$ 

Messtellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
F_pi	(FIC 11.1)	nein / FIC 11.1 befindet sich hinter der Mischstelle der Sumpfheizung
F_po	-	nein
F_li	-	nein
F_li2	(FIC 5.1)	nein / unter semantischer Kenntnis des zwischengelagerten Bypass-Ventils
F_lo	-	nein
P_o	PDI 7.4	ja
P_i	PDI 7.4	ja

Abbildung 4.32: Instanziierung  $K_3$ 

**Pumpe** In den Engineeringdaten können zwei Flüssigkeitspumpen ( $P_1$  und  $P_3$ ) gefunden werden, die Vakuumpumpe  $P_4$  wird durch eine andere Rollenklasse ausgeschlossen. Für die Pumpe  $P_1$  (Teilanlage 5, NC 5.3, Tabelle 4.7 auf Seite 110 und Abbildung 4.33

auf Seite 110) können zwar nur zwei Messstellen instanziiert werden, jedoch ist mit diesen bereits eine Aussage über das Verhältnis von Drehzahl zu Fördermenge möglich. Da für die Pumpe *P 3* in Teilanlage 5 (*NC 5.7*) nur der Aktuator instanziiert werden kann, wurde auf Tabelle und Abbildung verzichtet. Es kann keine Kennzahl bestimmt werden.

Tabelle 4.7: Instanziierung *P 1*

Messstellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
S_soll	<i>NC 5.3</i>	ja
F_v_in	-	nein
F_m_in	<i>FIC 5.1</i>	ja
P_in	-	nein
F_v_out	-	nein
F_m_out	-	nein
P_out	-	nein

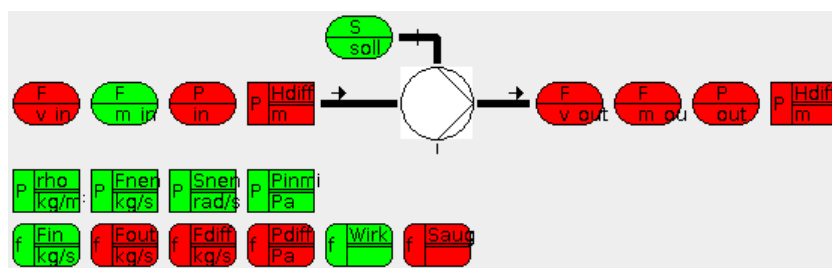


Abbildung 4.33: Instanziierung *P 1*

**Rohrheizung** Für die Rohrheizungsmodelle können zwei Instanzen (*W 6* und *W 4*) bestimmt werden. In Teilanlage 7 ist der Vorheizter *W 6* (*JC 7.3*, Tabelle 4.8 und Abbildung 4.34) bestimmbar. Automatisch können nicht alle Messstellenanforderungen gefunden werden. Mit der Zuordnung von *FIC 5.1* zu *F\_min* können jedoch alle Kennwerte berechnet werden. Weiterhin könnte das Modell auf die Sumpfheizung *W 4* von *K 3* in Teilanlage 7 (*JC 7.7*, Tabelle 4.9 und Abbildung 4.35 auf Seite 112) angewendet werden. Dafür fehlt jedoch der Durchflusssensor und somit kann nur die generelle Übertemperaturwarnung berechnet werden. Dies stellt jedoch keinen besonderen Mehrwert gegenüber einem Geräte- oder Prozessalarm dar, der höchstwahrscheinlich ohnehin ins PLS gemeldet wird.

Das Heizungsmodell lässt sich auch auf weitere Apparate in der Anlage sinnvoll einsetzen. Ein automatisches Finden dieser beiden Instanzen ist jedoch nicht möglich. Hierfür müsste das Modell mit einer anderen Rollenklasse neu erstellt werden. Darauf wurde verzichtet, um zu zeigen, dass die Modelle auch in ähnlichen Anwendungsfällen funktionieren können. Für die indirekten Gasheizungen *W 8* in Teilanlage 2 (*JC 2.6*, Tabelle 4.10 auf Seite 112 und Abbildung 4.36 auf Seite 112) und *W 9* in Teilanlage 11 (*JC 11.6*, Tabelle 4.11 auf

Tabelle 4.8: Instanziierung Heizung W 6

Messtellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
J_heiz	JC 7.3	ja
F_vin	-	nein
F_min	(FIC 5.1)	nein / unter semantischer Kenntnis des zwischengelagerten Bypass-Ventils
T_in	TI 5.9	ja
F_vout	-	nein
F_mout	-	nein
T_out	TIC 7.1	ja

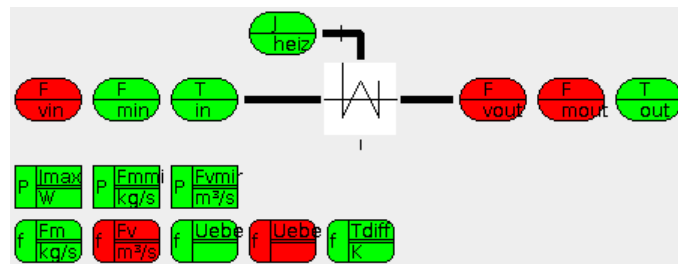


Abbildung 4.34: Instanziierung Heizung W 6

Tabelle 4.9: Instanziierung Heizung W 4

Messtellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
J_heiz	JC 7-7	ja
F_vin	-	nein
F_min	-	nein
T_in	TIC 7.5	ja
F_vout	-	nein
F_mout	-	nein
T_out	-	nein

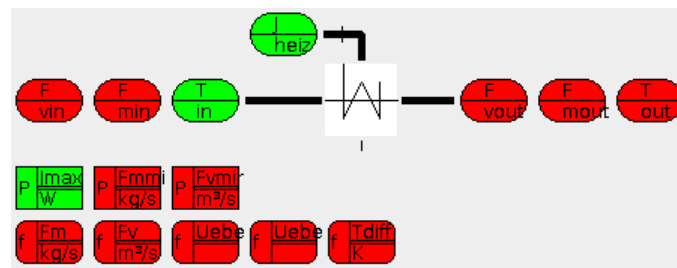


Abbildung 4.35: Instanziierung Heizung  $W_4$

Seite 113 und Abbildung 4.37 auf Seite 113) können ebenfalls die geheizten Stoffströme und die Heizungen überwacht werden. Beide Instanzen besitzen eine Durchflussmessung und eine Ausgangstemperaturmessung, jedoch keine Eingangstemperaturmessung. Demzufolge können alle Kennzahlen berechnet werden außer der Temperaturdifferenz. Es bleibt jedoch festzustellen, dass ein spezialisiertes Modell für die beiden Apparate mehr Kennzahlen liefern könnte.

Tabelle 4.10: Instanziierung Heizung  $W_8$

Messtellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
J_heiz	JC 2.6	ja
F_vin	(FIC 1.1 + FIC 1.4)	nein / über Summation von Teilflüssen
F_min	-	nein
T_in	-	nein
F_vout	FI 2.7	ja
F_mout	-	nein
T_out	TIC 2.4	ja

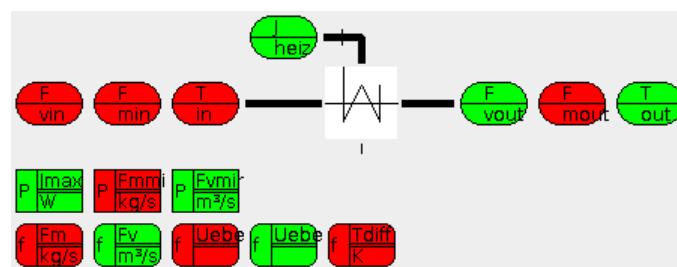


Abbildung 4.36: Instanziierung Heizung  $W_8$

#### 4.6.2.2 Instanzen im ACPLT Pumpwerk

Abbildung 4.38 auf Seite 114 zeigt die Versuchsanlage des Lehrstuhls Prozessleittechnik der RWTH Aachen (ACPLT Pumpwerk) [298]. In dieser sind fünf mögliche Instanzen für



Tabelle 4.11: Instanziierung Heizung W 9

Messstellenanforderung	mögliche Messstelle	gefunden / auffindbar unter der Bedingung
J_heiz	JC 11.6	ja
F_vin	-	nein
F_min	-	nein
T_in	-	nein
F_vout	FIC 11.1	ja
F_mout	-	nein
T_out	TIC 11.4	ja

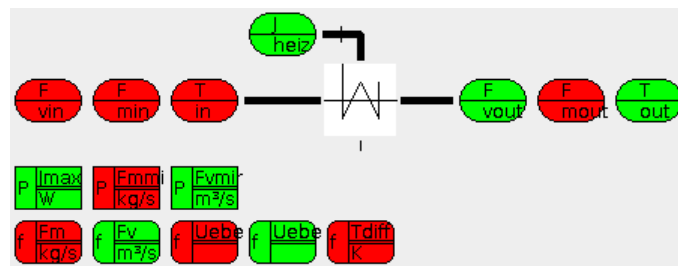


Abbildung 4.37: Instanziierung Heizung W 9

das Pumpenmodell zu finden. Bei den zuordenbaren Messstellen ergeben sich folgende Muster: Alle Instanzen besitzen einen Antrieb, aus dem das Stellsignal ausgelesen werden kann. Pumpe *P 13* (Antrieb *N13*), Pumpe *P18* (Antrieb *N18*), Pumpe *P29* (Antrieb *N29*) besitzen ein ausgangsseitiges Durchfluss-Messgerät. Pumpe *P36* (Antrieb *N36*) und Pumpe *P38* (Antrieb *N38*) teilen sich einen ausgangsseitigen Durchfluss-Sensor. Somit kann eine Kennzahl (Fördermenge je Drehzahl) berechnet werden. Über eine logische Auswertung der Ventilstellungen von *Y37* und *Y39* sowie der Funktion der Rückschlagklappen *K45* und *K46* könnte im wechselseitigen Betrieb ein Wert zugeordnet werden. Die Modelle von Pumpe *P13* (Antrieb *N13*) und Pumpe *P18* (Antrieb *N18*) können ebenfalls den ausgangsseitigen Drucksensor instanzieren. Diese allein sind jedoch nicht hinreichend, um weitere Kennzahlen berechnen zu können. Somit ist das Modell mit der vorliegenden Instrumentierung in drei von fünf Instanzen verwendbar.

#### 4.6.2.3 Auswertung

Aus der Instanziierung der in Abschnitt 4.6.1 ab Seite 103 beschriebenen Apparatemodelle wurden vor allem drei Schlüsse gezogen:

- 1) Die alternativen Messstellenanforderungen erhöhen die Anzahl der Suchoptionen für die benötigten Messstellen. Dadurch können Berechnungsfunktionen für Kennzahlen in mehr Instanziierungssituationen ausgeführt werden und damit erhöht sich die Abdeckung durch die Apparatemodelle. Wenn der Messort für die Erfassung eines

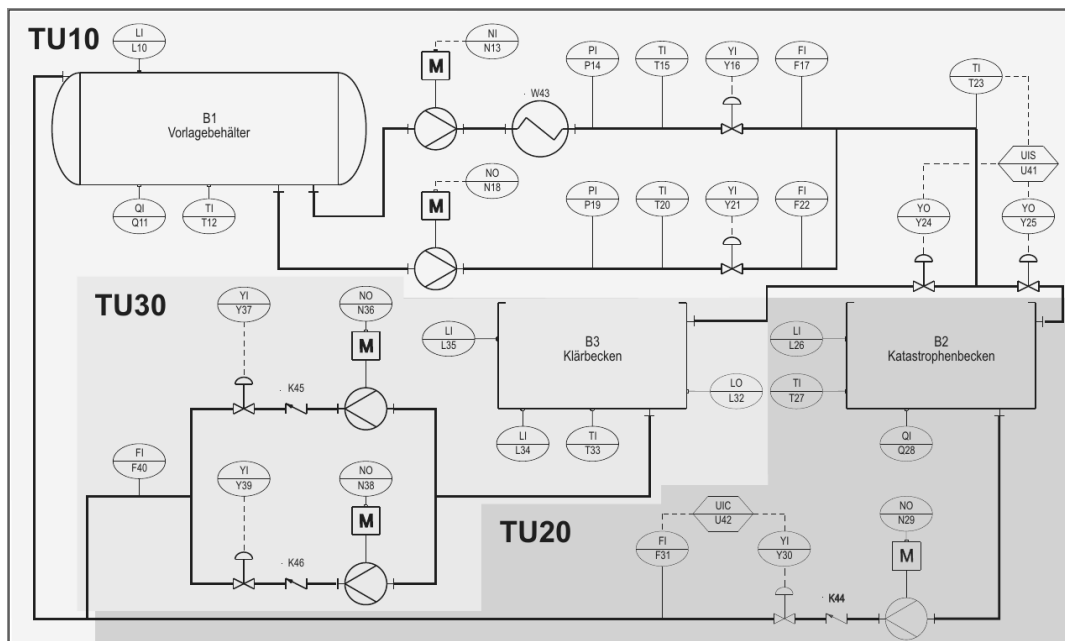


Abbildung 4.38: R&I-Fließbild des ACPLT Pumpwerks (aus [298])

Messwertes irrelevant ist, erhöht sich die Wahrscheinlichkeit, dass eine benötigte Größe in der Anlage tatsächlich von einem Messgerät erfasst wird. Das Negativbeispiel ist das erstellte Apparatemodell für den Wärmeübertrager. Bei diesem werden nur an den Einlässen der beiden Produkträume Durchfluss-Messstellen gesucht. Messstellen hinter den Auslässen könnten den Wert jedoch genauso bestimmen. Da aber die nicht vorhandene Redundanz durchaus einen modellrelevanten Grund haben kann, darf nicht automatisch an Auslässen gesucht werden, wenn an den Einlässen keine Messgeräte gefunden wurden. Deswegen sollte die Anforderungen wie im Apparatemodell für die Rohrheizung oder die Pumpe angelegt werden. Da es in diesem Fall unerheblich ist, ob der Messwert an Ein- oder Auslass aufgenommen wird, ist das Modell in mehr Instrumentierungssituationen verwendbar.

- 2) Wenn nicht genügend Messstellen vorhanden sind, kann ein Apparatemodell nicht verwendet werden. In einigen Situationen ist mit Wissen über Flusswege, die automatisch nicht eindeutig bestimmt werden können, das eine oder andere Messgerät mehr zuzuordnen. Auch modellbasierte Simulationen von nicht vorhandenen Messwerten [211] können weitere Daten liefern. Jedoch kann es Apparate geben, für deren Betrieb Messstellen nicht benötigt werden, ohne die ein passendes Modell aber keinen Mehrwert bringen kann. Dieser Nachteil ist von der Akquise-Infrastruktur unabhängig. Auch ein für ein PLS implementiertes Modell könnte in einem solchen Fall nicht verwendet werden. Diese Möglichkeit sollte von einem Hersteller von Apparatemodellen in Betracht gezogen werden.
- 3) Wenn die Engineeringdaten konsistent sind, kann die Suche nach Instanzen über die SystemUnitClass des Apparates bzw. Apparatemodells keine fehlerhaften Treffer liefern. Ist die SystemUnitClass nicht hinterlegt, gibt es die Alternative, über die

Rollenklasse zu suchen. Hiermit sollten alle tatsächlichen Instanzen des vom Modell behandelten Apparatetyp gefunden werden, jedoch auch die Instanzen anderer Typen, die die gleiche Rollenklasse unterstützen. Deswegen kann nicht in jedem Fall die Instanzzuordnung automatisch erfolgen.

### 4.6.3 Anbindung der Kennzahlen

Aktuelle PAM- und CM-Werkzeuge für Feldgeräte können die Selbstüberwachungsfunktionen gemäß NE 107 [201] integrieren [299]. Diese Funktionen sind mittels der bestehenden Integrationstechnologien EDD und FDT beschrieben bzw. eingebunden. Bei einer Migration auf FDI als Integrationstechnologie ändern sich Werkzeuge und Schnittstellen, die Funktion bleibt jedoch erhalten. Durch die Infrastruktur könnte die redundante Anbindung der PAM/CM-Werkzeuge und des Geräteintegrationswerkzeugs an die Feldebene wegfallen. Beide können dann über den zentralen FDI-Server auf die Feldgeräte zugreifen. Dies reduziert auch den Pflegeaufwand bei Updates von Gerätesoftware und den dazugehörigen EDD-Dateien bzw. DTMs.

Durch die Verwendung von FDI für die Einbindung von Feldgeräte-Modellen kann unter Zuhilfenahme der vorgestellten Infrastruktur jede Apparatemodell-Instanz wie ein Feldgeräte-Modell eingebunden werden. Es wird keine weitere Software benötigt. Je nach Implementierung des PAM- und CM-Werkzeuges ist darin kein Integrationsaufwand notwendig. Neue Apparatemodell-Instanzen können genauso wie neue Feldgeräte automatisch durch ein erneutes Durchsuchen der Geräteliste im FDI Server detektiert und automatisch integriert werden. Wenn die KPI im FDI Apparatepackage genauso ausgezeichnet sind, wie die NE 107-KPI in Feldgeräte-FDI Packages, können diese genauso automatisch integriert werden wie bei Feldgeräten.

Eine weitere Möglichkeit der Einbindung der Kennzahlen aus FDI Apparatepackages besteht mittels des Zugriffs über die OPC UA-Schnittstelle des FDI Servers. Hierbei ist zu beachten, dass der FDI Server in der Lage ist, ein Gerät in den *Offline*-Modus zu setzen oder ganz zu entladen, wenn es nicht verwendet wird. Der verwendete OPC UA-Client muss demzufolge den FDI Mechanismus zum erneuten Laden eines Gerätes beherrschen. Hierfür muss zuerst auf die Node im *DeviceSet* zugegriffen werden und das Gerät in den *Online*-Modus versetzt werden.

Generell muss bei der Einbindung der Meldungen auf die korrekte Priorisierung und Einbindung in die Alarmhierarchie geachtet werden, um Alarmfluten zu verhindern [300, 264].

### 4.6.4 Dynamik und Skalierbarkeit

Aufgrund der noch nicht verfügbaren FDI Server mit OPC UA und FDI Communication Server-Unterstützung konnten die Kommunikationskanäle nicht in der designierten Konfiguration untersucht werden. Durch die fehlenden Verzögerungszeiten in der Kommunikation mit realen Feldgeräten über Feldbus konnten daher keine aussagekräftigen Laufzeitmessungen durchgeführt werden. Die Skalierbarkeit und zu erwartende Auslastung der Netzwerke und Feldbusse lässt sich hingegen relativ zur durch klassische PAM- und CM-Aufgaben ausgelösten Auslastung [9] analytisch abschätzen.

Für diese Abschätzung wird von einer Grundlast im Kommunikationssystem und auf den Servern (FDI Server und FDI Communication Server), hervorgerufen durch CM- und PAM-Aufgaben, ausgegangen. Wird diese Auslastung bestimmt, kann anhand der folgenden Abschätzung aus der Systemlast ohne Apparatemodell-Infrastruktur auf die Systemlast mit der Apparatemodell-Infrastruktur geschlossen werden. Für die Berechnung der Spannen wurde Intervallarithmetik [301] verwendet. Zur leichteren Interpretierbarkeit der Intervalle ist immer ein zusätzlicher Wert in der Mitte des Intervalls notiert. Dieser Wert hat keinen Einfluss auf die Intervallgrenzen und wird mit gemittelten oder angenommenen Werten belegt.

$$x \in [x_{min}; \mathbf{x}_{med}; x_{max}] \quad (4.1)$$

Die Operationen sind wie folgt abgewandelt:

$$\begin{aligned} x \circ y &= [x_{min}; \mathbf{x}_{med}; x_{max}] \circ [y_{min}; \mathbf{y}_{med}; y_{max}] \\ &= [\min \{x_{min} \circ y_{min}, x_{min} \circ y_{max}, x_{max} \circ y_{min}, x_{max} \circ y_{max}\}; \\ &\quad \mathbf{x}_{med} \circ \mathbf{y}_{med}; \\ &\quad \max \{x_{min} \circ y_{min}, x_{min} \circ y_{max}, x_{max} \circ y_{min}, x_{max} \circ y_{max}\}] \end{aligned} \quad (4.2)$$

Zur Bestimmung von Kommunikationsaufwänden wurden die Beispiel-Apparatemodelle aus Abschnitt 4.6.1 ab Seite 103 und deren Instanzen aus Abschnitt 4.6.2 ab Seite 106 betrachtet. Tabelle 4.12 zeigt die Anzahlen von Messstellen, die von Apparatemodellen angefordert werden und die in den Beispielen tatsächlich instanziiert werden konnten. Im Durchschnitt wurden 60% der Messstellenanforderungen instanziiert. Dieser Faktor von  $0.6 \frac{\text{Messst}}{\text{Messst}}$  kann zur Abschätzung der Mehraufwände herangezogen werden. Dieser und alle weiteren Werte wurden in der Darstellung auf eine Nachkommastelle gerundet. Die Berechnungen, die diese Werte weiter verwenden, benutzen den genauen Wert.

**Tabelle 4.12:** Anzahlen von Messgeräteanforderungen in den Apparatemodell-Beispielen

Apparatemodell	Anzahl Messstellen (ohne Redundanzen)	durchschnittlich instanziierte Messstellen	Verhältnis Anzahl instanziierte zu Gesamtanzahl
Wärmeübertrager	6	4	0.7
Kolonne 1	4	-	-
Kolonne 2	4.5	2	0.2
Pumpe	5	2	0.2
Rohrheizung	4	3	0.8
<i>Minimum</i>	4	2	0.3
<i>Durchschnitt</i>	4.7	2.8	<b>0.6</b>
<i>Maximum</i>	6	4	1

In der Konfiguration mit den OPC UA Aggregationsserver und den über NodeSets nachgebildeten FDI Communication Servern (gemäß Abbildung 3.11 auf Seite 50) ergibt sich der in Abbildung 4.39 oben abgebildete Kommunikationspfad. Darunter ist der Pfad für einen Zugriff in der migrierten Architektur (gemäß Abbildung 3.12 auf Seite 51). In beiden Fällen erhält der FDI Server durch eine Client-Anfrage zwei OPC UA-Anfragen

(gefüllte Pfeilspitze), wobei die zweite ein Aufruf auf mehrere Nodes ist. Unten in der Abbildung ist zum Vergleich ein Zugriff auf ein Feldgerät dargestellt. Die Feldbuszugriffe (offene Pfeilspitze) werden nicht gebündelt, aber parallel ausgeführt. Somit ergibt sich der Kommunikationsaufwand bei den beiden oberen Szenarien für eine Anfrage an ein Apparatemodell im Gegensatz zu einer CM- oder PAM-Anfrage auf ein einzelnes Feldgerät (unten) wie folgt ( $FgMod = \text{Anzahl Feldgerätemodelle}$ ,  $AppMod = \text{Anzahl Apparatemodelle}$ ,  $Messst = \text{Anzahl Messstellen}$ ,  $UAAnf = \text{Anzahl OPC UA-Anfragen}$ ,  $FgRead = \text{Anzahl Lesezugriffe auf Feldgeräte}$ ):

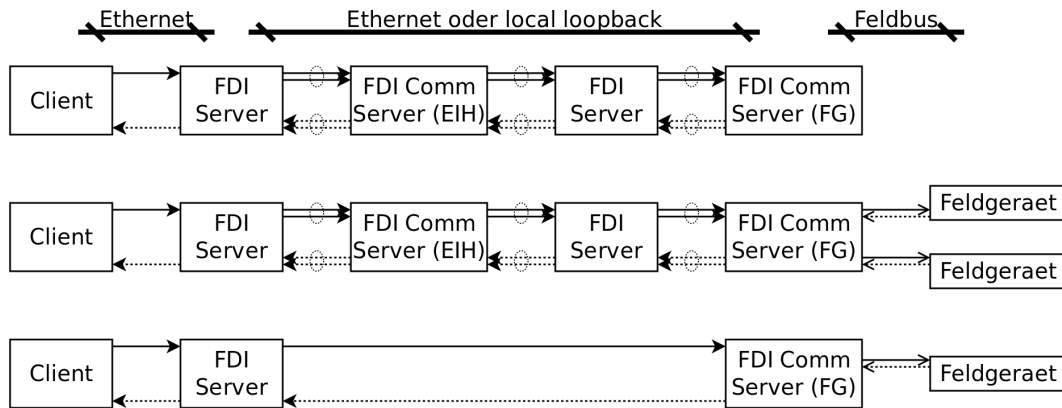


Abbildung 4.39: Kommunikationspfade in den verschiedenen Ausprägungen der Infrastruktur

**OPC UA Anfragen:** Faktor  $[2.0; 2.0; 2.0] \frac{FgMod}{AppMod} := \frac{4 UAAnf/AppMod}{2 UAAnf/FgMod}$

Da mehrere Leseanfragen in einen Serviceaufruf gebündelt werden können, sind vier statt zwei OPC UA-Serviceaufrufe im Kommunikationspfad. Somit produziert ein Apparatemodell die doppelte Netzwerklast am FDI Server eines Feldgerätemodelles (bei gleichem Abfragezyklus).

**zu lesende Nodes in den FDI Servern:** Faktor  $\left[ \frac{FgMod}{AppMod} \right] \leq$  maximale Anzahl der instanziierten Messgeräteanforderungen

In den Beispielen sind bei vollständiger Instrumentierung außer der redundanten Messgeräteanforderungen vier bis sechs Messstellen instanziiert (siehe Tabelle 4.12). Da in einem PAM-Modell für ein einzelnes Feldgerät üblicherweise mehr als ein Parameter gelesen werden muss, ist eine Abschätzung, die den maximalen Faktor verwendet, sehr defensiv.

Das Maximum kann durch eine geschätzte Anzahl von für ein Feldgeräte-PAM-Modell benötigten Aufrufen geteilt werden. Ein relativ komplexes Beispiel ist ein Modell zum Erkennen einer Schwergängigkeit in einem Ventil: Die Ansteuerung des Stellantriebs ist das Feldgerät, das vom Apparatemodell bzw. Feldgerätemodell angesprochen wird. Es muss die aktuelle Lage ausgelesen werden, in den *Manual-Mode* gesetzt werden, ein anderer Stellwert gesetzt werden, nach einer Messzeit die Position oder eine im Gerät bestimmte Verfahrzeit ausgelesen werden und der Modus wieder in den *Normal-Modus* versetzt werden. Dies sind fünf separate Zugriffe auf

ein Gerät. Minimal werden zwei Zugriffe angenommen. Die Spanne für diesen Faktor kann also als  $[0.8; \mathbf{1.7}; 3.0] \frac{\text{FgMod}}{\text{AppMod}} := [\frac{4}{5}; \mathbf{5}; \frac{6}{2}] \frac{\text{Messst}/\text{AppMod}}{\text{Messst}/\text{FgMod}}$  angenommen werden.

**zu lesende Parameter in Feldgeräten:** Faktor  $\left\lceil \frac{\text{FgMod}}{\text{AppMod}} \right\rceil \leq$  maximale Anzahl der instanziierten Messgeräteanforderungen

Da durch jeden Zugriff auf eine Node im FDI Communication Server ein Feldbuszugriff ausgelöst wird, wird die gleiche Anzahl wie die der zu lesenden Nodes angenommen. Die Spanne ist  $[0.8; \mathbf{1.7}; 3.0] \frac{\text{FgMod}}{\text{AppMod}} := [0.8; \mathbf{1.7}; 3.0] \frac{\text{FgRead}/\text{AppMod}}{\text{FgRead}/\text{FgMod}}$ . Aus jeder Messstelle werden `OUT_Value` und `OUT_Status` gelesen. Diese befinden sich in PA-Geräten in einem *Value & Status*-Datentyp (101) [116] und werden mit einem azyklischen Lesezugriff über den Feldbus gelesen.

Um die Lasterhöhung in der gesamten Anlage abschätzen zu können, ist noch das Verhältnis aus in Betrieb befindlichen Apparatemodellen und Feldgerätemodellen zu bestimmen. Hierfür werden die Engineeringdaten der beiden Beispielanlagen aus Abschnitt 4.6.2 ab Seite 106 verwendet. Es werden daraus alle Feldgeräte und Apparate erfasst, unabhängig davon, ob ein Modell dafür vorliegt oder nutzbringend wäre. Die Anzahlen sind in Tabelle 4.13 aufgeschlüsselt. Die Sensoren und Aktuatoren wurden separat erfasst und ihre Summe bestimmt (MSR-Anforderungen ohne die Regler). Da PAM- bzw. CM-Modelle für Ventile – wie bereits angedeutet – auch als Feldgerätemodell ohne übergreifende Betrachtung ausgeführt sein kann, wurden diese neben allen weiteren Apparaten gesondert erfasst. Somit lassen sich die Abschätzungen für verschiedenen Szenarien vornehmen. Dabei wird angenommen, dass die in den Szenarien betrachtete Menge von Feldgeräten mittels eines FDI-basierten PAM- oder CM-Modell regelmäßig bzw. automatisch überwacht werden.

Die Erhöhung des Lastanteils für PAM- und CM-Aufgaben [9] errechnet sich aus dem Faktor oder der Faktorenspanne des betrachteten Auslastungsaspekts und dem Verhältnis aus zu erwartenden Apparatemodell-Instanzen und Feldgerätemodellinstanzen. Dieses Verhältnis wird als Intervall um den Quotienten der entsprechenden Summen aus Apparatemodellen  $\Sigma_A$  und Feldgerätemodellen  $\Sigma_F$  plus/minus der Standardabweichung der Quotienten der Teilanlagen angegeben  $[\frac{\Sigma_A}{\Sigma_F} - \sigma_{TA}; \frac{\Sigma_A}{\Sigma_F}; \frac{\Sigma_A}{\Sigma_F} + \sigma_{TA}]$ . Die Pseudo-Einheiten  $\frac{\text{FgMod}}{\text{AppMod}}$  und  $\frac{\text{AppMod}}{\text{FgMod}}$  kürzen sich bei der Multiplikation weg:  $\frac{\text{FgMod}}{\text{AppMod}} \cdot \frac{\text{AppMod}}{\text{FgMod}} \cdot \frac{\text{Messst}}{\text{Messst}}$ . Diese Berechnung ist nur dann aussagekräftig, wenn bereits eine Grundlast durch Feldgerätemodelle besteht. Sie beziehen sich auf eine Vollausrüstung mit Apparatemodellen, die vollständig mit Messstellen instanziiert sind. Wie die Untersuchungen in Abschnitt 4.6.2 ab Seite 106 gezeigt haben, ist dies eine sehr defensive Annahme. Weniger gefundene Messstellen und Apparatemodelle, die gar nicht instanziiert werden können, verbessern die Lastsituation. Das durchschnittliche Verhältnis von instanziierten und definierten Messstellenanforderungen war  $0.6 \frac{\text{Messst}}{\text{Messst}}$ . Dabei kann ohne großes Risiko für die Sicherheit der Aussage ein Korrekturfaktor von  $[0.5; \mathbf{0.6}; 0.8]$  multipliziert werden.

**(8) ÷ (1)** Verhältnis zwischen der Anzahl aller Apparate mit Ventilen als Apparatemodell und allen Aktuatoren ohne Ventile (worst case):

Dieses Szenario geht davon aus, dass ausschließlich PAM- und CM-Modelle für Aktuatoren (keine Ventilantriebe) regelmäßig oder automatisch geprüft werden.

**Tabelle 4.13:** Anzahlen von Mess- und Stellgeräten und Apparaten in den Beispielanlagen sowie deren Verhältnisse  $\left[\frac{\text{AppMod}}{\text{FgMod}}\right]$

<i>CO<sub>2</sub>-Waschanlage TU Dresden</i>													...
Teilanlage	1	2	3	4	5	6	7	8	9	10	11	Σ	...
Aktuator (1)	0	1	0	0	2	0	2	0	1	0	1	7	
Ventil (2)	2	1	0	0	0	1	1	1	0	1	2	9	
Sensor (3)	2	3	5	3	6	4	3	3	3	2	3	37	
(1)+(2) = (4)	2	2	0	0	2	1	3	1	1	1	3	16	
(1)+(3) = (5)	2	4	5	3	8	4	5	3	4	2	4	44	
(4)+(3) = (6)	4	5	5	3	8	5	6	4	4	3	6	53	
Apparat (7)	0	1	2	1	4	1	3	1	1	2	1	17	
(2)+(7) = (8)	2	2	2	1	4	2	4	2	1	3	3	26	
(8)÷(1)	-	2.0	-	-	2.0	-	2.0	-	1.0	-	3.0	<b>3.7</b>	
(8)÷(5)	1.0	0.5	0.4	0.3	0.5	0.5	0.8	0.7	0.3	1.5	0.8	<b>0.6</b>	
(7)÷(4)	-	0.5	-	-	2.0	1.0	1.0	1.0	1.0	2.0	0.3	<b>1.1</b>	
(7)÷(6)	-	0.2	0.4	0.3	0.5	0.2	0.5	0.3	0.3	0.7	0.2	<b>0.3</b>	

...	<b>ACPLT Pumpwerk</b>					<b>alle</b>	
...	Teilanlage	10	20	30	Σ	$\sigma_{TA}$	$\frac{\Sigma_A}{\Sigma_F}$
	Aktuator (1)	2	1	2	5		
	Ventil (2)	4	1	2	7		
	Sensor (3)	10	4	4	18		
	(1)+(2) = (4)	6	2	4	12		
	(1)+(3) = (5)	12	5	6	23		
	(4)+(3) = (6)	16	6	8	30		
	Apparat (7)	4	2	3	9		
	(2)+(7) = (8)	8	3	5	16		
	(8)÷(1)	4.0	3.0	2.5	<b>3.2</b>	0.9	3.5
	(8)÷(5)	0.7	0.6	0.8	<b>0.7</b>	0.3	0.6
	(7)÷(4)	0.7	1.0	0.8	<b>0.8</b>	0.5	0.9
	(7)÷(6)	0.3	0.3	0.4	<b>0.3</b>	0.1	0.3

Dazu sollen mittels der Apparatemodell-Infrastruktur für alle Apparate inklusive der Ventile Apparatemodelle integriert werden. Aufgrund der vielen Divisionen durch Null in dieser Zeile im ersten Beispiel weichen die Einzelergebnisse deutlich vom Quotienten der Summen ab. Aus diesem Grund liegt das Maximum des Intervalls über dem größten Einzelwert (4.0).  $\Sigma_A/\Sigma_F$  ist gleich [2.6; **3.5**; 4.4].

**OPC UA Anfragen:**

$$[2.0; \mathbf{2.0}; 2.0] \cdot [2.6; \mathbf{3.5}; 4.4] \cdot [0.5; \mathbf{0.6}; 0.8] = [2.6; \mathbf{4.1}; 7.0]$$

**zu lesende Nodes / zu lesende Parameter:**

$$[0.8; \mathbf{1.7}; 3.0] \cdot [2.6; \mathbf{3.5}; 4.4] \cdot [0.5; \mathbf{0.6}; 0.8] = [1.0; \mathbf{3.5}; 10.5]$$

Anhand dieses Szenarios lässt sich nur sehr schlecht eine Aussage ableiten. Der Faktor von 10.5 bedeuten eine erhebliche Erhöhung des azyklischen Datenverkehrs. Gleichzeitig ist die Netzwerklast, auf deren Basis die zu erwartende Last bestimmt wird, relativ gering. Die einzige Aussage, die sicher getroffen werden kann, ist eine allgemeine, die jedoch für alle vier Szenarien gilt: Wenn die Feldbus- und FDI-Infrastruktur ohne Apparatemodelle ausgelastet ist, kann das Konzept in diesem Szenario nicht angewendet werden.

- (8) ÷ (5)** Verhältnis zwischen der Anzahl aller Apparate mit Ventile als Apparatemodell und allen Sensoren und Aktuatoren ohne Ventile:

Im Gegensatz zum vorherigen Szenario wird in diesem davon ausgegangen, dass auch für die Sensoren CM- oder PAM-Modelle in Betrieb sind.  $\Sigma_A/\Sigma_F$  ist gleich [0.3; **0.6**; 1.0].

**OPC UA Anfragen:**

$$[2.0; \mathbf{2.0}; 2.0] \cdot [0.3; \mathbf{0.6}; 1.0] \cdot [0.5; \mathbf{0.6}; 0.8] = [0.3; \mathbf{0.8}; 1.5]$$

**zu lesende Nodes / zu lesende Parameter:**

$$[0.8; \mathbf{1.7}; 3.0] \cdot [0.3; \mathbf{1.0}; 1.4] \cdot [0.5; \mathbf{0.6}; 0.8] = [0.1; \mathbf{0.7}; 2.3]$$

Mit einer durchschnittlichen Erhöhung der Aufwände auf weniger als das doppelte ist die Verwendbarkeit der Infrastruktur wahrscheinlich.

- (7) ÷ (4)** Verhältnis zwischen der Anzahl aller Apparate ohne Ventilen und allen Aktuatoren mit Ventilen als Feldgerätemodell:

Dieses Szenario geht davon aus, dass die Ventile als Feldgerätemodell bereits installiert sind und kein zusätzliches Apparatemodell für sie instanziiert werden soll.

$\Sigma_A/\Sigma_F$  ist gleich [0.4; **0.9**; 1.4].

**OPC UA Anfragen:**

$$[2.0; \mathbf{2.0}; 2.0] \cdot [0.4; \mathbf{0.9}; 1.4] \cdot [0.5; \mathbf{0.6}; 0.8] = [0.5; \mathbf{0.8}; 1.4]$$

**zu lesende Nodes / zu lesende Parameter:**

$$[0.8; \mathbf{1.7}; 3.0] \cdot [0.4; \mathbf{0.9}; 1.4] \cdot [0.5; \mathbf{0.6}; 0.8] = [0.5; \mathbf{0.8}; 1.4]$$

Der Durchschnittswert liegt in der gleichen Größenordnung wie im vorherigen Szenario.



**(7)÷(6)** Verhältnis zwischen der Anzahl aller Apparate ohne Ventilen und allen Sensoren und Aktuatoren mit Ventilen als Feldgerätemodell:

Im Gegensatz zum vorherigen Szenario sind auch für PAM- oder CM-Modelle für Sensoren in Betrieb, wodurch eine hohe Grundlast für diese Aufgaben vorhanden ist.

$\Sigma_A/\Sigma_F$  ist gleich [0.2; **0.3**; 0.5].

**OPC UA Anfragen:**

$$[2.0; \mathbf{2.0}; 2.0] \cdot [0.2; \mathbf{0.3}; 0.5] \cdot [0.5; \mathbf{0.6}; 0.8] = [0.2; \mathbf{0.4}; 0.7]$$

**zu lesende Nodes / zu lesende Parameter:**

$$[0.8; \mathbf{1.7}; 3.0] \cdot [0.2; \mathbf{0.3}; 0.5] \cdot [0.5; \mathbf{0.6}; 0.8] = [0.1; \mathbf{0.3}; 1.1]$$

Selbst die maximale Schätzung für dieses Szenario ist in einem realistischen Bereich, weswegen in einem solchen Szenario keine Schwierigkeiten zu erwarten sind.

Zusammenfassend kann festgestellt werden, dass, wenn beim Engineering der Feldbusse bzw. Kommunikationsstrukturen Bandbreite für PAM- und CM-Aufgaben vorgesehen wurde und genügend Reservekapazitäten eingeplant wurden, nichts einer Einrichtung der beschriebenen Infrastruktur entgegen spricht. In einer Anlage, in der noch keine PAM- oder CM-Aufgaben für Feldgeräte etabliert sind, ist eine Einführung einer Infrastruktur zur Apparatemodell-Integration wahrscheinlich nicht der nächste Schritt. Deswegen sollte der zu erwartende Mehraufwand in der Kommunikation auf den Netzwerken immer auf Basis von Erfahrungswerten abschätzbar sein.



## 5 Zusammenfassung der Ergebnisse und Ausblick

Die zu Beginn gestellte Forschungsfrage „Wie könnte eine Infrastruktur auf Basis von verbreiteten Standardtechnologien aussehen, welche alle Schritte des Engineeringprozesses für freie Apparatemodelle automatisieren kann?“ wurde durch die beschriebene Infrastruktur beantwortet. Mithilfe der umgesetzten Komponenten konnte nachgewiesen werden, dass das beschriebene Konzept zur Modellintegration unter den gegebenen Vorbedingungen angewendet werden kann. Durch die Verwendung des Computer Aided Engineering eXchange-Austauschformats können das Engineering und die Integration der Modelle vereinfacht und automatisiert werden. Als Ausführungs- und Kommunikationsplattform wurde die Field Device Integration-Technologie verwendet. Die verwendeten Technologien sind zukunftsfähig und die durch sie eingeführten Abhängigkeiten sind durch verfügbare Produkte auflösbar. Mit den Anwendungsbeispielen wurde gezeigt, dass Modelle abstrakt definiert und auf konkrete Strukturen angewendet werden können.

Darüber hinaus werden einige Anforderungen aus Forschungsbedarfen im Bereich *Industrie 4.0* für die konzeptionierte und prototypisch umgesetzte Infrastruktur erfüllt. Die Einbindung der Modelle ist eine Selbstkonfigurationsfunktion; die Infrastruktur selbst ermöglicht eine Einbindung unabhängiger, externer Komponenten ohne die Basisfunktion zu beeinflussen. Dabei wurden ausschließlich Technologien verwendet, die vor und außerhalb der *Industrie 4.0* entstanden sind. Hiermit wurde ebenfalls gezeigt, dass unabhängig von konkreten Umsetzungen viele Voraussetzungen für die „digitale Revolution“ in der Automatisierung bereits erfüllt sind.

### 5.1 Ergebnisse

In einer Analyse des Standes der Technik und der Forschung wurden verschiedenste Techniken und Technologien mit Bezug zur Kommunikation in Automatisierungsanlagen untersucht (Kapitel 2 ab Seite 7). Diese wurden gemeinsam mit den Anforderungen an eine Lösung in Partialmodelle abstrahiert und diese in einem generischen Domänenmodell miteinander verknüpft. Aus den zur Verfügung stehenden und praxisrelevanten Technologien wurden Informationsträger für alle Partialmodelle und Verknüpfungen im Domänenmodell ausgewählt. Dabei sind vor allem diejenigen Standards wichtig, die die Struktur und die Semantik der Informationen allgemein zugänglich definieren. Auf Basis dieser ausgewählten Technologiebasis wurde ein Konzept zur Modelldefinition, Modellintegration und zum Betrieb der Modelle entwickelt (Kapitel 3 ab Seite 35). Dazu wurde auch ein Format für die Apparatemodelle entwickelt (Abschnitt 4.2 ab Seite 66).

Die anvisierte Infrastruktur wurde in der Form dreier Softwarekomponenten umgesetzt. Die erste Komponente ist ein Werkzeug zur Erstellung von Apparatemodellen. Sie ist geeignet, die Struktur der Modelle grafisch zu notieren und die Berechnungsmodelle

in einem Formeleditor zu definieren. Alle Bestandteile des Apparatemodells werden automatisch erzeugt. Der letzte Integrationsschritt, das Verpacken in ein FDI Package, war nur mit dem Standalone-Entwicklungstool der FDI-Referenzimplementierung der *FDI-Corporation* möglich und wurde nicht in das Werkzeug integriert. Er ließe sich prinzipiell über die *Java*-Klassen des *Eclipse*-Plugins automatisieren (Abschnitt 4.3 ab Seite 75).

Als zweite Komponente wurde das Instanzierungswerkzeug entwickelt. Es kann mithilfe von *XPath*-Anfragen in CAEX-Exporte von Engineeringdaten die Instanzen von Apparaten auffinden, auf die ein ausgewähltes Apparatemodell angewendet werden kann. Mit einem OPC UA-Client durchsucht sie das Informationsmodell des FDI Servers der Anlage nach den für das Apparatemodell benötigten Feldgeräten (Abschnitt 4.5 ab Seite 94). Mithilfe der gewonnenen Daten erzeugt es die Konfigurationsdaten für die dritte Komponente.

Diese dritte Komponente – der Equipment Instance Handler – ergänzt die FDI-Infrastruktur für den laufenden Betrieb der Apparatemodell-Infrastruktur. Der EIH vermittelt die gebündelten Leseanfragen nach Messdaten aus Feldgeräten an deren Informationsmodelle im FDI Server. Dabei verhält sie sich wie ein FDI Communication Server. Er lässt Apparatemodellinstanzen wie Feldgeräte aussehen, damit der FDI Server sie automatisch integrieren kann. Die Weiterleitung der Anfragen erfolgt mittels eines OPC UA-Clients (Abschnitt 4.4 ab Seite 86).

Aufgrund der zu späten Verfügbarkeit eines FDI Servers mit OPC UA-Schnittstelle und FDI Communication Server-Unterstützung sind verschiedene Lösungen entwickelt worden, um die Anwendbarkeit von Konzept und Infrastruktur auch ohne diese Komponente nachweisen zu können. Dabei wurden OPC UA-Informationsmodelle zur Emulation der zu erwartenden FDI Server-Infrastruktur entwickelt (Abschnitt 4.1.2 ab Seite 62). Die abschließende Integration konnte damit nicht vorgenommen werden, dennoch konnten die funktionalen Anforderungen weitestgehend erfüllt werden.

Weiterhin sind Werkzeuge zur Bereitstellung und Aufarbeitung von Daten entstanden. Mit *Dia2CAEX* wurde es ermöglicht, CAEX-Exporte von R&I-Fließbildern zu erstellen und mit diesen das Instanzierungswerkzeug zu testen (Abschnitt 4.1.1.4 ab Seite 61). Zur Prüfung von solcher CAEX-Exporte wurde ebenfalls ein Werkzeug entwickelt (Abschnitt 4.1.1 ab Seite 57).

## 5.2 Bewertung der Erfüllung gesteckter Ziele

Im Folgenden sollen die in Abschnitt 1.3 ab Seite 4 formulierten Ziele anhand der Eigenschaften der vorgestellten prototypischen Lösung abgeglichen werden.

### Automatisierbarkeit

Diese Infrastruktur beantwortet die gestellte Forschungsfrage positiv. Bei der Erstellung der Pakete im Apparatepaket-Werkzeug kann die gesamte Ausgabekette automatisiert werden. Die Definition des Modells ist in keinem Fall automatisierbar. Einige Arbeitsschritte könnten besser unterstützt werden, wie in Abschnitt 5.3 ab Seite 127 beschrieben wird.

Die Integration von Modellen kann vollständig automatisch erfolgen, wenn die System-UnitClass des Apparates in den Engineeringdaten verwendet wurde, genügend Messstellen vorhanden und sauber in den Engineeringdaten notiert sind und das Modell keine einzustellenden Parameter enthält. Ansätze für das automatische Setzen der Werte von Parametern

werden in Abschnitt 5.3 ab Seite 127 umrissen. Der Equipment Instance Handler muss einmalig eingerichtet werden und wird dann automatisch vom Instanzierungswerkzeug konfiguriert.

## Verwendung offener Standards und zukunftssicherer Technologien

Diese Anforderung sollte sicherstellen, dass keine Abhängigkeit von einem bestimmten Hersteller bzw. Produkt die Zukunftsfähigkeit der Lösung infrage stellen kann.

Die Beschreibung der Modellstruktur und die Integration von Instanzen erfolgte über ein herstellerneutrales, offenes und standardisiertes Austauschformat: CAEX [17, 37]. Seine Verwendung in AutomationML [50] und die weiterführende Standardisierung von Rollenklassen [51] sowie die Verwendung für den NAMUR-*PCE-Container* [43] ist ein starkes Indiz für eine Weiterverbreitung der Technologie in der Prozessautomatisierung.

Die Verwendung von FDI als Zugriffstechnologie für Feldgeräte ist ebenso zukunftsweisend. Branchenverbände wie die NAMUR haben sich auf die Verwendung dieser Technologie für die Geräteintegration als einzige Lösung festgelegt [267], [155, F. 4]. Mithilfe der OPC UA-Schnittstelle – ebenfalls einer bereits stark verbreiteten Basistechnologie – im FDI Server ist ein einheitlicher Zugriff auf Feldgerät möglich und automatisierbar. Dies ermöglicht auch weiteren Mehrwertdiensten den Zugriff. Somit bietet FDI auch Anreize zur Migration, nicht nur zur Verwendung in Neuanlagen. Bei der Umsetzung wurde Wert darauf gelegt, ausschließlich über die offenen und standardisierten Schnittstellen der in der Anlage vorhandenen Systeme zu kommunizieren.

Die Definition des Informationsmodells für Apparatemodelle musste selbst vorgenommen werden, da hierfür kein Standard existiert. Jedoch folgt es strukturell einem Informationsmodell [116, 190], welches in den PAM- und CM-Werkzeugen bekannt und der Zugriff darauf implementiert ist. Durch die Bereitstellung eines Werkzeugs für die Modellgenerierung wurde dennoch eine gleichartige Struktur der Modelle sichergestellt.

## Rückwirkungsfreiheit

Aufgrund der hohen Anforderungen an Verfügbarkeit und Sicherheit darf eine Zusatzfunktion keinen Einfluss auf die Basisfunktionalität der Anlage haben.

Dies wird insbesondere dadurch sichergestellt, dass die bestehenden Komponenten spezifikationskonform verwendet wurden. Die Zugriffe über die standardisierten Schnittstellen sind getestet und gewährleisten eine korrekte Funktion bzw. konsistenzsichernde Ausnahmenbehandlung.

Auch beim Datenzugriff über die Feldbusse ist durch die Protokolle eine Beeinträchtigung der Kommunikation der Produktivfunktionen technisch ausgeschlossen. Die Zykluszeiten auf den Medien werden durch erhöhten Datenverkehr zwar vergrößert, sie können jedoch ein im Engineering festgelegtes Maximum nicht überschreiten.

Gleichzeitig sorgt ein in FDI Apparatepackages implementierter und für den EIH entworfener Mechanismus zur Lastbegrenzung und -verteilung dafür, dass unnötige Lesezugriffe unterbunden werden und eine zeitliche Häufung von Anfragen vermieden wird.

### Zuverlässigkeit

Eine Struktursuche birgt immer die Möglichkeit, falsche Ergebnisse zu finden oder Vorkommen zu übersehen. Die im Instanziierungswerkzeug implementierte Suche über die `SystemUnitClasses` der Apparate schließt solche Fehler aus. Bei der alternativen Suche über `RoleClasses` können zwei Probleme auftreten. Einerseits kann das Apparatemodell für einen gefundenen Apparat nicht anwendbar sein, andererseits könnten redundante Rollenklassen-Definitionen ein Auffinden verhindern. Eine Standardisierung der Rollenklassen wie in [51] beseitigt das zweite Problem.

Das Auffinden der Modelle im generierten OPC UA-Informationsmodell des FDI Servers wurde erfolgreich nachgewiesen. Die Funktion der geplanten Kommunikationswege wurde ebenfalls überprüft [12].

Die Zuverlässigkeit konkreter Modelle kann hier nicht bewertet werden, da deren Erstellung und Validierung nicht im Fokus dieser Arbeit liegt.

### Offenheit

Die Struktur der Modelle ist frei zugänglich. Die Berechnungsmodelle können durch die Verwendung von EDDL produktunabhängig notiert werden. Die Sprache ist standardisiert und mehrere Interpreter sind verfügbar. Die Werkzeuge verstecken die Komplexität der EDDL vor den Herstellern eines Apparatemodells. Die im Formeleditor angegebenen Berechnungsmodelle werden in EDD übersetzt. Bei Bedarf besteht dennoch die Möglichkeit, die generierte EDD nachzubearbeiten.

Das Informationsmodell des FDI Communication Servers im EIH ist strukturell durch die IEC 62769-7 [158] vorgegeben, die Instanzausprägung wird durch das FDI Apparatepackage festgelegt. Das Quellmodell und die Abbildungsvorschriften zur EDD sind in dieser Arbeit dokumentiert.

### Verwendbarkeit

Die Integrierbarkeit der Berechnungsergebnisse in gewerkespezifische Werkzeuge wurde durch die Wahl der Technologien sichergestellt. Die Apparatemodelle integrieren sich in die FDI-Infrastruktur wie Feldgerätemodelle. Somit können spezifische Tools über die für sie üblichen Schnittstellen auf die Apparatemodelle zugreifen. Gleichzeitig ist auch ein generischer Zugriff über die OPC UA-Schnittstelle des FDI Servers möglich. In diesem Fall muss jedoch der zu lesende Kennwert aus dem Informationsmodell heraus gesucht werden. Durch eine Standardisierung von OPC UA-Informationsmodellen für CM-Modelle wie in [231] kann auch diese Suche beschränkt werden.

Bei der prototypischen Implementierung der zwei Werkzeuge spielte die eingängige Benutzbarkeit eine untergeordnete Rolle. Dennoch konnte bei der Anwendung auf die vorgestellten Beispiele gezeigt werden, dass Wissen über die Struktur des Apparates für beide Werkzeuge, über Apparat und Modell im Apparatepaket-Werkzeug und über die Anlage im Instanziierungswerkzeug für die Erstellung und Integration des FDI Apparatepackages ausreicht. Es werden außer Hinweise zur Bedienung keine Informationen benötigt, die beim Herstellen eines Modells nicht ohnehin zu diesem Zweck vorlägen.

## Nutzen

Der Nutzen, der durch die Infrastruktur gelieferten Mehrwertdienste erreicht wird, ist nicht allgemein quantifizierbar, denn er ist von der Qualität der Apparatemodelle, von der Verwendbarkeit der Berechnungsergebnisse sowie von der konkreten Verwendung derselben abhängig. Generell ist es möglich, dass Instanzen von Apparatemodellen durch Kombination mehrerer Datenquellen Zustände quantifizieren, die ohne sie nicht bestimmt und erkannt würden. Ob die Anwendungsfälle dieser Infrastruktur in Zukunft durch andere Systeme abgedeckt werden, wie zum Beispiel die Standard-Modelle für PLS, sollte in Frage gestellt werden. Beispielsweise ist es bei der Modularisierung [212, 25] von verfahrenstechnischen Anlagen möglich, herstellerseitig Bewertungsmodelle in die Steuerungs- und Kommunikationstechnik des Moduls zu integrieren. Dem entgegen ist es wiederum schwieriger, weitere Mehrwertdienste zu integrieren, da die Automatisierungstechnik eines Moduls möglicherweise proprietär ausgeführt ist und das Modul nur über seine bei der Planung vorgesehene Schnittstelle ansprechbar ist. Deswegen könnte eine nachträgliche Integration von nicht anfänglich vorgesehenen Mehrwertdiensten unmöglich sein.

## 5.3 Ansätze für Erweiterung und Fortführung

Für eine produktiv einsetzbare Lösung sind die vorgestellten Werkzeuge noch nicht geeignet. Jedoch konnten mit ihrer Hilfe Modelle mit Basisfunktionalitäten erstellt und bearbeitet werden.

### Konzept und Infrastruktur

Das in Kapitel 3 ab Seite 35 vorgestellte Konzept bietet wenig substanzielles Erweiterungspotential unter den für die Infrastruktur angenommenen Annahmen und Entscheidungen bezüglich Technologien und Paradigmen sowie der Wichtung von domänenspezifischen Anforderungen. Durch technische Weiterentwicklungen, zukünftig verfügbare Wissensrepräsentationen, zukünftige Technologien und nicht zuletzt einer anderen Ausrichtung bei den vorgenannten Bedingungen würde sich das in Abschnitt 3.1 ab Seite 35 beschriebene Domänenmodell ändern. Somit wird eine unter anderen Vorbedingungen entworfene Lösung für die Integration von Apparatemodellen wahrscheinlich anders aussehen. Insbesondere bei der Betrachtung von modularisierten Anlagen gibt es neue Strukturen, Automatisierungsparadigmen und Anforderungen [25]. Aufgrund der in der Prozessindustrie üblicherweise langen Innovationszyklen bleibt die vorgestellte Lösung mindestens mittelfristig tragfähig.

Auf einfachere Ansätze zur Weiterentwicklung des Konzepts, die sich auf eine spezielle Komponente beziehen und in Kapitel 4 ab Seite 57 teilweise bereits herausgearbeitet wurden, wird in den folgenden Abschnitten eingegangen.

### Apparatepaket-Werkzeug

Wie bereits in Abschnitt 4.3 ab Seite 75 angesprochen wurde, gibt einige Möglichkeiten das Apparatpaket-Werkzeug weiter zu verbessern. Wenn zu einem Apparat bereits eine SystemUnitClass existiert, könnte mit dieser die Apparatstruktur in das Werkzeug geladen

werden. Dies erspart dem Nutzer die Eingabe der Apparatestruktur. Sonst erzeugt das Apparatpaket-Werkzeug diese Daten aus den Eingaben. Weiterhin wäre es erforderlich, ein zu einer früheren Version erzeugtes EDD-Dictionary wieder importieren zu können, um nicht alle Übersetzungen erneut in diese Datei übertragen zu müssen.

Beim manuellen Erstellen der Messgeräteanforderungen kann eine automatische Generierung von Anforderungen für eine alternative Messgröße oder am zweiten Anschluss eines Produktraums angeboten werden. Dies wäre bei der Erstellung der Beispiele in folgenden Fällen hilfreich gewesen: Zum einen könnte beim Erstellen der Durchfluss-Messgeräteanforderung am Eintritt eine gleichartige Anforderung am Austritt des Produktraums erstellt werden. Zum anderen könnte bei der Erstellung einer Anforderung nach einer Massenstrom-Messgeräteanforderung ebenfalls eine Volumenstrom-Messanforderung und ein Parameter für die Dichte vorgeschlagen werden. Beide benötigen dann noch eine Funktion, welche den geforderten Messwert je nach Instanziierungsvariante mit einer festen Maßeinheit zur Verfügung stellt. Wird das Modell dann mit einem Massenstrom-Messgerät instanziiert, wird der Parameter nicht benötigt, sonst muss dieser bei der Instanziierung festgelegt werden.

Auch der Formeleditor sollte deutlich erweitert werden. EDDL definiert eine C-ähnliche Programmiersprache. Deren Kontrollstrukturen sollten im Editor für Berechnungsformeln genutzt werden können. Dies macht jedoch die automatische Erstellung des Codes zur Prüfung der \*\_IS\_VALID-Variablen schwieriger. Auch müssen die Gültigkeitsbereiche der Variablen im erstellten EDD-Code beachtet werden, wo die bis dahin getrennten Berechnungsformeln hintereinander im gleichen *Scope* ausgegeben werden. Für dieses Feature wird kein vollständiger EDDL-Parser benötigt, sondern nur der Subset für die Methoden.

Für die Erstellung eines Symbols für das FDI Apparatpackage wird ein besserer grafischer Editor benötigt. Das Laden eines Bildes könnte ebenfalls eine Alternative sein.

Wenn die FDI Package-Toolchain [259] voll eingebunden werden soll, müssen alle Artefakte des FDI Apparatpackages im Werkzeug ausgewählt und integriert werden (z. B. *PDF*-, *CAD*- und ähnliche Dateien).

Eine deutliche Erweiterung der Möglichkeiten des Einsatzes der Modelle wäre durch die Verfügbarkeit von Historien von Messwerten erreichbar. Dafür wäre es notwendig, andere Datenquellen einzubinden (z. B. mittels OPC UA *Historical Access* [302]). Die höhere Belastung des FDI Servers durch den deutlich größeren Datendurchsatz würde jedoch die Nutzbarkeit beschränken.

Um die technische Zuverlässigkeit der Apparatemodelle im Betrieb abzusichern, kann die EDD des FDI Apparatpackagees wie die eines Feldgerätes getestet werden. Hierfür kann die PA-Testspezifikation [303, Abs. 7] weitestgehend verwendet werden. Da der Apparat nicht als Feldgerät an den PA-Feldbus angeschlossen werden kann, müsste ein virtuelles Feldgerät wie in [304, 121] zwischengeschaltet werden, welches einen Lesezugriff auf einen OPC UA-Zugriff umsetzt. Eine erweiterte Testmöglichkeit könnte mithilfe von Instanziierungsszenarien wie die Beispiele in Abbildung 4.13 und Abbildung 4.14 auf Seite 82 realisiert werden. Mit Datenverläufen von Eingangsgrößen der Messgeräteanforderungen für verschiedene Situationen können die Berechnungsformeln systematisch getestet werden.



## Instanziierungswerkzeug

Die strukturelle Suche im Werkzeug könnte robuster gegen unsaubere Strukturinformationen sein (wie z. B. in [38] beschrieben). Bisher wurden keine CAEX-Dateien akzeptiert, die nicht vom Prüfwerkzeug als fehlerfrei deklariert wurden. Die Verarbeitung nicht formal korrekter Eingangsdaten birgt jedoch auch die Gefahr, falsche Messgeräte zu erkennen.

Die volle Automatisierung der Instanziierung könnte durch das automatische Festlegen der Werte der in den Apparatemodellen definierten Parameter ermöglicht werden. Eine einfache Variante wäre, die Parameter in die exportierte `SystemUnitClass` zu übernehmen und sich darauf zu verlassen, dass deren Werte im Detail Engineering, wenn die beschriebene Rolle durch die konkrete Umsetzung ersetzt wird, vom Planer festgelegt wird. Ein solches Verfahren ist zwar sehr sicher bezüglich der Zuordnung der Parameter, es hat dennoch einige Unsicherheiten. Insbesondere, wenn nicht klar ist, wofür ein Wert benötigt wird, wird dieser entweder nicht gesetzt oder nur grob überschlagen. Will man diese Unwägbarkeiten umgehen, benötigt man zuerst eine semantische Beschreibung des Parameters. Hier wären standardisierte Attributklassen, wie in CAEX V3 [37] vorgesehen, geeignet. Diese könnten z. B. auf Festlegungen in der IEC 61360 [283] verweisen. Nach solchen CAEX-Attributen könnte dann in einem größeren Suchraum in der Umgebung des Objektes, zu dem der Apparatemodell-Parameter gehört, gesucht werden.

Werte, die fest in die Engineeringdaten gehören wie z. B. Rohrdurchmesser wären dann einfach zu finden und automatisch zu setzen. Materialspezifische Parameter für Grenzwerte könnten ebenfalls auffindbar sein. Der Schluss auf den Wert benötigt bereits mehrere Schritte, weswegen ein solcher Automatismus recht komplex würde. Werte aus Feldgeräten könnten ebenfalls verwendet werden, da diese aber einen eigenen Alarmmechanismus besitzen sollten, hat das Teilmodell wahrscheinlich keinen Mehrwert. Auch auf die Werte für Parameter, die Eigenschaften von Produkten ablegen, müsste in mehreren Schritten geschlossen werden. Es zeigt sich, dass eine solche Komponente viele Wissensrepräsentationen kombinieren müsste und deswegen relativ komplex werden würde.

## Equipment Instance Handler

Wie bereits in Abschnitt 4.4.5 ab Seite 93 beschrieben kann auch im EIH ein Mechanismus zur Lastverteilung und Lastbeschränkung implementiert werden. Insbesondere, wenn die Möglichkeit besteht, dass FDI Apparatepackage zur Anwendung kommen, die keine eigene Begrenzung der Leseaufrufe in die zugeordneten Feldgeräte implementieren, ist ein solcher Mechanismus notwendig. Effekte durch gegenseitige Beeinflussung von paketinternem und im EIH implementiertem Mechanismus wären noch zu untersuchen.

Eine weiterer Ansatz ist, den FDI Server, den FDI Communication Server und die Feldbusse zu entlasten, indem die Werte, die ohnehin über den Echtzeitdatenverkehr übertragen werden, nicht zusätzlich azyklisch zu lesen. Da ein Eingriff in das PLS gemäß Anforderungen nicht erlaubt ist, können die Daten auch über ein PIMS bezogen werden. Hier gibt es jedoch noch die Hürde einer fehlenden Standardisierung der PIMS-Schnittstelle.

Eine solche Datenquelle würde den Zugriff auf Werthistorien ermöglichen (z. B. bei einer OPC UA-Anbindung mittels *Historic Data* [302]). Die größere Menge an zu verarbeitenden Daten würde jedoch (trotz der geringeren Anzahl von Zugriffen auf die Schnittstellen und die FDI Communication Server) die gesamte Auslastung des FDI Servers steigern.

Es ist also in Frage zu stellen, ob ein separater FDI Server für FDI Apparatepackages vorzusehen wäre oder der Overhead generell als zu hoch eingeschätzt würde.

### Einbindung und Beschreibung

Die vorgestellte Variante zur Einbindung verlässt sich auf die Funktion zur automatischen Erkennung und Einbindung von Feldgerätemodellen in FDI Packages. Jedoch sind auch weitere Möglichkeiten vorhanden, um die berechneten KPI der Apparateinstanzen einzubinden. Voraussetzung sind dafür eine semantische Annotation der gemäß einer standardisierten Bedeutung oder eine Einbettung in ein standardisiertes Informationsmodell. Hierfür könnte die *Business-To-Machine Markup Language (B2MML)* [305] bzw. die dazu entwickelte OPC UA-Spezifikation [306] oder die *Key Performance Indicator Markup Language (KPI-ML)* [307] verwendet werden.

Es gäbe auch die Möglichkeit, durch den EIH den Alarmmechanismus eines Feldbus-segments simulieren zu lassen und die Meldungen der Apparatemodelle als Gerätealarm in das PLS einzubinden. Auf diese Weise ist die Repräsentation als virtuelles Feldgerät authentischer. Diese Variante der Integration widerspricht jedoch der Anforderung, nicht in das Produktivsystem einzugreifen.

Ein wichtiger Aspekt für die einfache Nutzbarkeit von Bewertungsmodellen ist nicht nur die Anzeige des Apparatzustands, sondern auch die Unterbreitung von Vorschlägen für die Behebung von „schlechten“ Zuständen. Da den apparatezentrierten Bewertungsmodellen deren weitere Umgebung unbekannt ist, sind apparateübergreifende Zusammenhänge nicht in jedem Fall genau untersuchbar.

Eine weitere Möglichkeit zur Weiterverwendung in überlagerten Systemen ist eine Aggregation mehrerer Zustandsbewertungen in größeren Zusammenhängen, z. B. Teilanlagen, die mehrere Apparate enthalten. Das VDMA-Einheitsblatt 24582 [231] beschreibt derartige Aggregations-FBs zur gewichteten Verknüpfung von vorverarbeiteten Kennzahlen. Mittels einer solchen Hierarchie von Zustandsinformationen kann eine übersichtlichere Anzeige erreicht werden.

Die VDI/VDE-Richtlinie 2651 Blatt 2 [198] definiert Beschreibungsmittel für die Spezifikation von PAM-Anforderungen und die Dokumentation von PAM-Funktionen. Solche Dokumente könnten vom Apparatpaket-Werkzeug ausgegeben werden. Bis jetzt fehlt hierfür jedoch noch ein einheitliches elektronisches Format.

## 5.4 Fazit

Die Anwendung von PAM- und CM-Methoden zur Optimierung von Prozessen und Betriebsabläufen erreicht eine immer breitere Anwendung in der Prozessindustrie. Gerade bei einem zeitweisen Einsatz oder der Nachrüstung von entsprechenden Systemen ist eine rückwirkungsfreie Integration unerlässlich. Mit der vorgestellten Infrastruktur lassen sich diese Anforderungen mit geringem Integrationsaufwand umsetzen. Die Integration kann dynamisch im Betrieb und in jeder Lebenszyklusphase, in der Feldgeräte und FDI-Server verfügbar sind, vorgenommen werden. Es wurde gezeigt, dass die Bewertung von Apparatzuständen mit generischen Modellen ohne Annahmen über den konkreten Kontext technisch möglich ist. Diese Flexibilisierung adressiert auch künftige Nutzungsszenarien, wie sie durch *Industrie 4.0* definiert werden.

# Akronyme

<b>ABK</b> Anzeige-Bedien-Komponente .....	15, 24
<b>AML</b> AutomationML .....	4, 12, 14, 30, 44, 58, 97, 99, 125
<b>API</b> Application Programming Interface .....	63, 78
<b>BL</b> Business Logic .....	21, 49, 69, 70
<b>CAE</b> Computer Aided Engineering .....	9, 10, 12, 13, 15, 58, 97
<b>CAEX</b> Computer Aided Engineering eXchange .	9, 11, 12, 14, 44, 45, 53, 57–59, 61, 62, 66, 68, 78, 83, 95, 96, 98, 99, 106, 123–125, 129
<b>CAN</b> Controller Area Network .....	16
<b>CFF</b> Common File Format .....	17
<b>CIP</b> Common Industrial Protocol .....	18
<b>CM</b> Condition Monitoring .	1, 3, 27, 30, 31, 52, 55, 68, 82, 115–118, 120, 121, 125, 126, 130
<b>COM</b> Component Object Model .....	19, 25
<b>DD</b> Device Definition .....	21
<b>DP</b> (PROFIBUS) Dezentrale Peripherie .....	18, 46
<b>DIN</b> Deutsches Institut für Normung e.V. ....	9, 14, 79, 100
<b>DTM</b> Device Type Manager .....	19, 20, 26, 27, 31, 115
<b>EDD</b> Electronic Device Description ....	19, 21, 22, 26, 27, 45, 49, 51, 53, 55, 66–75, 79, 81–85, 88, 90, 93, 100, 101, 106, 115, 126, 128
<b>EDDL</b> Electronic Device Description Language	18–21, 38, 45, 49, 52, 67, 72, 75, 79, 87, 92, 126, 128
<b>EDS</b> Electronic Data Sheet .....	17
<b>EIH</b> Equipment Instance Handler ....	51, 53, 55, 68, 74, 80, 86–88, 90, 92–95, 100–102, 124–126, 129, 130, 182, 184
<b>ERP</b> Enterprise Resource Planning .....	25
<b>FB</b> Funktionsblock .....	18, 19, 30, 45, 65, 68, 69, 71, 73, 75, 87, 88, 92, 93, 101, 130

<b>FDI</b> Field Device Integration	18, 20–23, 26, 45, 47–50, 52–54, 64, 65, 68, 82, 85, 87, 92, 100, 115, 118, 120, 123–126, 130
<b>FDT</b> Field Device Tool	19–23, 51, 53, 82, 115
<b>FDT2</b> Field Device Tool Version 2	23
<b>FF</b> Foundation Fieldbus	16–18, 21, 22, 45, 54, 93, 100
<b>GMA</b> VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik	14
<b>GSD</b> General Station Description	17, 85, 90, 91
<b>GUI</b> Graphical User Interface	76–78, 82, 95, 96
<b>HMI</b> Human Machine Interface	24, 31, 47
<b>I&amp;M</b> Identification&Maintenance	17, 18, 92
<b>IEC</b> International Electrotechnical Commission	9, 11, 12, 14, 18, 19, 21–23, 26, 30, 32, 36, 57, 58, 61, 70, 78, 79, 83, 85, 92, 93, 126, 129
<b>ISA</b> The Instrumentation, Systems, and Automation Society	24, 25, 27
<b>ISO</b> International Organisation for Standardization	9, 12, 13, 18, 30–32, 57, 58, 61
<b>KPI</b> Key Performance Indicator	4, 28, 30, 43, 45–47, 52, 55, 69, 70, 72, 75, 81, 103–105, 109, 110, 112, 113, 115, 130
<b>MES</b> Manufacturing Execution System	25, 55, 56
<b>MSR</b> Messen-Stellen-Regeln	1, 9, 13, 37–39, 52, 53, 57, 106, 118
<b>NA</b> NAMUR-Arbeitsblatt	28, 30, 32
<b>NAMUR</b> Interessengemeinschaft Automatisierungstechnik der Prozessindustrie	16, 18, 21, 26, 54, 125
<b>NE</b> NAMUR-Empfehlung	14, 18, 27, 32, 100, 115
<b>.NET</b> .NET-Framework	22
<b>OLE</b> Object Linking and Embedding	25
<b>OPC</b> OLE for Process Control	4, 25, 26, 31
<b>OPC UA-DI</b> (OPC UA) for Device Integration	21, 26, 63, 65, 100
<b>OPC UA</b> OPC Unified Architecture	4, 13, 21, 25, 26, 45–52, 54, 55, 62–65, 67, 86–88, 92, 94, 95, 100–102, 115–117, 120, 121, 124–126, 128–130, 132, 140
<b>PAM</b> Plant Asset Management	1, 3, 17, 20, 24, 26, 27, 30, 32, 37, 47, 52–54, 68, 82–84, 115–118, 120, 121, 125, 130

---

<b>PandIX</b> P&I Exchange .....	58, 61, 96
<b>PCE</b> Process Control Engineering .....	36, 44, 45, 58, 61, 70, 78, 79, 98, 125
<b>PIMS</b> Process Information and Management System .....	30, 55, 94, 129
<b>PLS</b> Prozessleitsystem .....	1, 2, 13–17, 23–31, 41, 47, 48, 103, 110, 114, 127, 129, 130
<b>PLT</b> Prozessleittechnik .....	9, 11, 13, 14, 39, 57
<b>PNK</b> Prozessnahe Komponente .....	15, 24
<b>PNO</b> PROFIBUS Nutzerorganisation (jetzt PROFIBUS International (PI)) .....	20, 69
<b>PN IO</b> PROFINET IO .....	18, 46, 69, 88
<b>PA</b> (PROFIBUS) Process Automation .	16, 18, 21, 38, 45, 46, 54, 64, 65, 68, 69, 71–73, 80, 81, 85, 87–89, 92, 93, 100, 118, 128
<b>PROFIBUS</b> Process Field Bus .....	17, 22, 38, 64, 68, 85, 87, 89, 92, 93, 100, 101
<b>SCADA</b> Supervisory Control and Data Aquisition .....	24
<b>SDK</b> Software Development Kit .....	63–66, 88
<b>SPS</b> Speicherprogrammierbare Steuerung .....	24
<b>STEP</b> Standard for the Exchange of Produkt Model Data .....	12
<b>UID</b> User Interface Description .....	21, 22, 75, 82
<b>UIP</b> User Interface Plugin .....	21–23, 82, 85
<b>VDE</b> Verein der Elektrotechnik Elektronik Informationstechnik	8, 13, 26, 27, 30, 84, 130
<b>VDI</b> Verein Deutscher Ingenieure .....	8, 13, 25–27, 30, 32, 84, 130
<b>VDMA</b> Verband Deutscher Maschinen- und Anlagenbau .....	30, 32, 130
<b>WPF</b> Windows Presentation Foundation .....	22
<b>XML</b> eXtensible Markup Language	11–13, 48, 58, 59, 62, 63, 66, 76, 85, 87, 88, 96, 99, 101
<b>XSD</b> XML Schema Description .....	58, 59, 66, 87, 100
<b>XSLT</b> eXtensible Stylesheet Language Transformation .....	62, 88, 177



# Glossar

- Anlage** verfahrens- bzw. prozesstechnische Produktionsanlage .....  
1–3, 5, 7–15, 24, 25, 27–29, 31, 32, 37, 41–43, 45, 53–55, 57, 58, 63–65, 71, 77, 84,  
93, 99, 100, 103, 106, 107, 111, 114, 118, 121, 124–127
- Anschluss** Modellelement eines Apparate-Bewertungsmodells .....  
44, 67, 69–74, 76, 82, 98, 128
- Anzeige-Bedien-Komponente** (ABK) Hardwarekomponente eines Prozessleitsystems .  
25, 133, 140
- Apparat** zentrales Modellelement eines Apparate-Bewertungsmodells .....  
43–45, 67, 69–73, 76, 77, 83
- Apparat** (= (piece of) Equipment) Anlagenausrüstung wie Tanks, Kessel, Wärmeübertra-  
ger, etc. ....  
2, 4, 8, 9, 29, 30, 32, 33, 37–39, 41, 42, 44–46, 48, 52, 53, 56, 58, 67, 71, 75, 83, 84,  
88, 89, 92, 95–99, 101, 105, 106, 109, 111, 112, 114, 115, 118–121, 124, 126–128, 130,  
159, 163
- Apparatemodell** Bewertungsmodell zur Berechnung von Kennzahlen für einen Apparat  
1, 7, 29–31, 35, 37, 39, 42–45, 47–49, 51, 52, 54–57, 63, 67–69, 71, 73, 75–79, 81, 83,  
84, 89, 94–106, 113–118, 120, 121, 123–130
- Apparatepaket** technologieneutraler Begriff für FDI Apparatepackages) .....  
43, 45, 67, 68, 75–77, 79, 82, 84, 85, 88, 89, 91, 95–97, 101, 103, 104, 124, 126–128,  
130
- Application Programming Interface** (API) Dokumentierte Schnittstelle zwischen Pro-  
grammbibliotheken und Anwenderprogrammen .....  
133
- Asset** für (Plant) Asset Management relevante Ressource .....  
27–29, 31, 33, 35, 84
- Business Logic** (BL) Teil eines FDI Packages, Lese- und Schreibkommandos, der mit-  
telsEDD Abhängigkeiten zwischen Parametern sowie Methoden beschreibt .....  
22, 23, 49, 69, 70, 133
- CAEX-Attribute** CAEX-Konstrukt, welches die Eigenschaften aller CAEX-Klassen defi-  
niert .....  
11, 57, 58, 60, 97, 129
- Common File Format** (CFF) Eine Gerätebeschreibungsdatei mit den Daten für die Inbe-  
triebnahme von Foundation Fieldbus-Geräten .....  
17, 133

<b>Common Industrial Protocol (CIP)</b> Spezifikation und Protokoll eines gemeinsamen Geräte-Informationsmodell für verschiedene Feldbusse ( <i>DeviceNet, ControlNet, EthernetIP</i> ) .....	18, 133
<b>Component Object Model (COM)</b> <i>Microsoft Windows</i> -spezifisches Objekt- und Schnittstellenframework .....	20, 133
<b>Computer Aided Engineering eXchange (CAEX)</b> XML-basiertes CAE-Austauschformat nach IEC 62424 [17] .....	9, 11, 12, 14, 44, 45, 54, 57–59, 61, 62, 67, 68, 77, 83, 84, 96, 98, 99, 106, 123–125, 129, 133, 142
<b>Condition Monitoring (CM)</b> Zustandsüberwachung von Assets .....	1, 3, 28, 31, 32, 53, 56, 69, 83, 115–118, 120, 121, 125, 126, 130, 133
<b>Controler Area Network (CAN)</b> Feldbustechnologie .....	17, 133
<b>Device Definition (DD)</b> Teil der EDD in einem FDI-Paket, der die Gerätestruktur beschreibt .....	23, 133
<b>Device Directory</b> Speicherbereich (nur lesbar) eines PROFIBUS PA-Gerätes, der die Struktur des Gerätes beschreibt .....	101
<b>Device Type Manager (DTM)</b> gerätespezifische Softwarekomponente zum Zugriff mittels FDT-Technologie .....	19, 20, 27, 28, 31, 115, 133
<b>(PROFIBUS) Dezentrale Peripherie (DP)</b> Feldbustechnologie .....	19, 46, 133, 143
<b>eCI@ss</b> mit PROLIST zusammengeführte Beschreibung von Produktdaten (u. A. Merkmalleisten von Feldgeräten) .....	12, 14, 15, 43, 84
<b>Electronic Data Sheet (EDS)</b> Beschreibung für den Parametersatz eines intelligenten Feldgerätes .....	17, 133
<b>Electronic Device Description (EDD)</b> Funktionsblock-orientierte Beschreibung für den Parametersatz und die Bedienoberfläche eines intelligenten Feldgerätes .....	19, 21, 23, 27, 28, 45, 49, 52, 54, 55, 67, 69, 70, 72–76, 80, 81, 83–86, 88, 89, 94, 101, 106, 115, 126, 128, 133
<b>Electronic Device Description Language (EDDL)</b> Integrationstechnologie für intelligente Feldgeräte .....	19, 21–23, 38, 45, 49, 52, 67, 72, 75, 79, 87, 92, 126, 128, 133, 140, 144



- 
- Enterprise Resource Planning (ERP)** Software zur Unternehmensverwaltung .....  
25, 133
- Equipment** (= Apparat bzw. Apparate) .....  
41, 44, 58, 61
- Equipment Instance Handler (EIH)** Zusatzkomponente zur Umsetzung des in dieser Arbeit beschriebenen Konzepts der Modellintegration .....  
52, 53, 55, 68, 74, 81, 86–89, 91, 94–96, 101, 102, 124–126, 129, 130, 133, 160, 162
- eXtensible Markup Language (XML)** selbstbeschreibendes Datenformat .....  
11–13, 49, 59, 62, 63, 65–67, 77, 86, 88, 97, 99, 101, 135
- eXtensible Stylesheet Language Transformation (XSLT)** in XML notierte Scriptsprache zur Transformation von XML-Dokumenten in ein beliebiges Zieldokument .....  
88, 135, 155
- ExternalInterface** CAEX-Konstrukt, welches Schnittstellen von InternalElements, SystemUnitClasses und RoleClasses beschreibt .....  
60, 68, 98–100
- ExternalReference** CAEX-Konstrukt, welches weitere Dateien referenziert .....  
59, 61, 62
- FDI Apparatepackage** (= Equipment Package, Apparatepaket) FDI Package für die Integration von Bewertungsmodellen für Apparate in FDI Servern .....  
43, 48, 50, 55, 67, 74, 81, 85–89, 94–97, 101, 102, 115, 125, 126, 128–130
- FDI Client** Komponente der Field Device Integration-Technologie zur Anzeige der in den FDI Packages definierten Benutzungsoberflächen .....  
23, 46, 69
- FDI Communication Server** Komponente der Field Device Integration-Technologie zur Anbindung von Feldgeräten an den FDI Server .....  
22, 46, 49–51, 64, 74, 87–91, 94, 101, 102, 115, 116, 118, 124, 126, 129, 130, 162
- FDI Package** Container für Gerätetyp-spezifische Beschreibungen und Softwarekomponenten für die Field Device Integration-Technologie .....  
21, 23, 24, 45, 46, 64, 65, 67, 71, 84–86, 89, 102, 115, 124, 128, 130
- FDI Server** zentrale Komponente der Field Device Integration-Technologie zur Speicherung und Bereitstellung der Geräteinformationsmodelle .....  
15, 22, 46, 48–52, 54, 55, 57, 63–65, 69, 71, 72, 74, 86–89, 91, 93–96, 102, 103, 115–117, 124–126, 128–130
- Feldbus** Variante zur Verkabelung von Feldgeräten; ermöglicht eine integrierte Inbetriebnahme, zentrales Parametrieren und ständigen Zugriff auf die Gerätedaten .....  
15–19, 23, 25, 28, 47, 53–55, 70, 74, 85, 92, 115–118, 120, 121, 125, 128–130
- Feldgerät** Intelligenter Sensor oder Aktuator mit eigener Kommunikationsschnittstelle  
2, 13–20, 22, 24, 25, 27, 28, 41, 43–46, 48–50, 53–55, 63, 64, 69–71, 74, 81, 85–89, 91–94, 96, 100, 101, 115, 117, 118, 121, 124–126, 128–130

<b>Field Device Integration (FDI)</b> Integrationstechnologie für intelligente Feldgeräte, kombiniert Electronic Device Description Language und Field Device Tool .....	19–23, 27, 45–47, 49, 51, 53, 54, 64, 65, 68, 83, 85, 86, 88, 92, 100, 115, 118, 120, 123–126, 130, 134, 139
<b>Field Device Tool (FDT)</b> Integrationstechnologie für intelligente Feldgeräte .....	19–21, 23, 52, 54, 83, 115, 134, 140
<b>Field Device Tool Version 2 (FDT2)</b> Integrationstechnologie für intelligente Feldgeräte, Nachfolger von Field Device Tool .....	23, 24, 134
<b>Foundation Fieldbus (FF)</b> Feldbustechnologie .....	16, 17, 19, 22, 23, 46, 54, 94, 100, 134, 137
<b>Funktion</b> Modellelement für eine Berechnung innerhalb eines Apparat-Bewertungsmodells .....	43, 45, 67, 70–74, 79, 82, 83, 106, 128
<b>Funktionsblock (FB)</b> (= Function Block) in der Automation mit verschiedenen Ausprägungen weit verbreitete Abstraktion (z. B. als Steuerungssoftwaremodule mit innerem Zustand gemäß IEC 61131-3, gemäß IEC 61499 oder für Prozessleitsysteme; als Modell und Strukturierungsmethode in der IEC 61804-2 [119] und IEC 61804-3 [124] sowie dem PA-Profil [116]; etc.) .....	30, 65, 69, 71–73, 75, 88, 130, 133, 138
<b>General Station Description (GSD)</b> früher Gerätestammdaten; Datei mit Daten für die Inbetriebnahme von PROFIBUS-Geräten .....	17, 85, 86, 89, 91, 134
<b>Graphical User Interface (GUI)</b> grafische Benutzungsoberfläche von Softwareanwendungen .....	77, 79, 83, 95, 96, 134
<b>HART</b> Digitalprotokoll über 4-20 mA-Zweidrahtleitungen .....	15–17, 19, 23, 46, 54, 55, 94, 100
<b>Human Machine Interface (HMI)</b> siehe Anzeige-Bedien-Komponente .....	25, 32, 46, 134, 142
<b>Identification&amp;Maintenance (I&amp;M)</b> Gerätedaten zur Identifikation und Wartungsunterstützung .....	18, 92, 134
<b>Informationsmodell</b> Definition aus [308]: „Ein Informationsmodell definiert Regeln, wie Informationen strukturiert werden können. Dies beinhaltet sowohl die Darstellung von Elementen des Informationsmodells, als auch deren Beziehungen untereinander. Informationsmodelle sind für	

- den Menschen interpretierbar. Beispiele für Informationsmodelle sind die UML-Diagrammtypen, R&I-Fließbilder oder PLT-Stellenpläne. Für eine digitale Verarbeitung ist es notwendig, Informationsmodelle in rechnerinterpretierbare Datenmodelle abzubilden.“ .....  
13, 14, 19–23, 26, 27, 46, 49–51, 54, 55, 63–68, 71, 87–89, 91–93, 100–103, 124–126, 130, 162
- InstanceHierarchy** CAEX-Konstrukt, welches tatsächliche Elemente einer Anlage als InternalElements abbildet .....  
11, 45, 58, 61, 62, 97–99
- Interessengemeinschaft Automatisierungstechnik der Prozessindustrie (NAMUR)**  
früher Normenarbeitsgemeinschaft für Meß- und Regeltechnik .....  
17, 18, 21, 27, 54, 125, 134
- InterfaceClass** CAEX-Konstrukt, welches Schnittstellen-Typen definiert .....  
11, 45, 57, 58, 60–62, 67, 77
- InternalElement** CAEX-Konstrukt, welches Teile einer InstanceHierarchy oder SystemUnitClass definiert .....  
60, 98, 100
- InternalLink** CAEX-Konstrukt, welches zwei ExternalInterfaces von Unterelementen miteinander verbindet .....  
62, 98–100
- Key Performance Indicator (KPI)** (= Kennzahl) durch ein Bewertungsmodell bestimmtes Maß für einen bestimmten Bewertungsaspekt; bildet eine komplexe Bewertung auf eine lineare Skala mit gut- und schlecht-Bereichen ab. ....  
75, 82, 115, 130, 134
- Lebenszyklus** Life-Cycle, Zeitabschnitt zwischen Planung und Recycling von Automatisierungsanlagen und ihren Komponenten (Instanz), vgl. Produktlebenszyklus: von der Produktplanung über Auslieferung und Abkündigung zum Supportende (Typ)  
7, 11, 13, 27, 29, 33, 42, 43
- Manufacturing Execution System (MES)** Softwaresystem zur Organisation der Ausführung von Produktionsprozessen .....  
25, 56, 134
- Messstelle** Modellelement für eine MSR-Anforderung in einem Apparatemodell (= Messstellenanforderung) .....  
43–45, 53, 67, 69–73, 75, 76, 82, 83, 100, 103, 107–114, 116–118, 128
- Messstelle** Anforderung in Engineeringdaten oder Geräteinstanz in einer Anlage für eine bestimmte Messaufgabe .....  
35, 41, 43, 45, 53, 54, 58, 62, 70, 94–96, 98, 99, 106–114, 118, 124, 157, 158, 163

- Namensraum** (= Namespace) Konzept zum Sicherstellen von eindeutigen Bezeichnern über Domänengrenzen hinweg. Für jede Domäne wird ein Name definiert (Namensraum), der mit den eindeutigen Namen innerhalb der Domäne zu global eindeutigen Bezeichnern zusammen gesetzt wird. ....  
64, 65, 89, 93, 101
- .NET-Framework** (.NET) Applikationsframework für *Microsoft Windows* .....  
23, 134
- Node** Oberbegriff bzw. abstrakte Oberklasse der Elemente des OPC Unified Architecture-Adressraummodells bzw. eines OPC Unified Architecture-Informationsmodells. Ihre Basissemantik erhält eine Node durch die vorgegebenen, ebenfalls abstrakten Unterklassen *Objekttyp*, *Objekt*, *Variablentyp*, *Variable*, *Datentyp*, *Referenztyp*, *Referenz*, *Methode* oder *View*. ....  
26, 64–67, 71, 72, 87, 91, 92, 94, 100, 101, 115, 117, 118, 120, 121
- NodeSet** Modell oder Teilmodell eines OPC Unified Architecture-Informationsmodells bzw. dessen Beschreibung (siehe Node) .....  
49, 65, 66, 88, 89, 101, 116, 162
- Object Linking and Embedding** (OLE) Auf proprietärer Technologie basierendes Framework zum Remote-Zugriff auf Softwareobjekte .....  
26, 134
- OLE for Process Control** (OPC) IT-Technologie für den Zugriff von Human Machine Interfaces oder Web-Schnittstellen auf Prozessdaten .....  
25, 26, 32, 134, 142
- (OPC UA) for Device Integration** (OPC UA-DI) OPC UA-Informationsmodell für die Abbildung von Feldgeräten und deren Vernetzung, als IEC 62541-100 [161] standardisiert .....  
22, 27, 63–65, 100, 134
- OPC Unified Architecture** (OPC UA) Plattform-unabhängiger Technologienachfolger von OLE for Process Control .....  
13, 21, 22, 25–27, 46, 49–51, 53–55, 63–65, 67, 86–88, 93, 94, 96, 100–103, 115–117, 120, 121, 124–126, 128–130, 134, 142
- P&I Exchange** (PandIX) Computer Aided Engineering eXchange-Bibliotheken der RWTH Aachen .....  
58, 61, 96, 135
- Parameter** Geräteparameter, abgebildet als Variable .....  
19, 20, 22, 23, 36, 45, 46, 67, 69–75, 81, 87–89, 93, 94, 100, 102, 103, 117, 118, 120, 121
- Parameter** Modellelement zur Adaption eines Apparate-Bewertungsmodells .....  
43–45, 67, 69–73, 83, 106, 124, 128, 129

- Plant Asset Management (PAM)** Definition aus [193]:  
 „Aspekt 1: Verwalten der Assets über den gesamten Lebenszyklus hinweg. Von besonderem Interesse sind Identifikation, Asset-Historie, betriebswirtschaftliche und technische Daten.  
 Aspekt 2: Organisation des Einsatzes und Zustanderhaltens der Assets  
 Aspekt 3: Erzeugen und Bereitstellen von Informationen, insbesondere über Verlauf und Prognose der Asset-Gesundheit zur Entscheidungsunterstützung“ .....  
 1, 3, 18, 20, 24, 27, 28, 30, 31, 33, 37, 46, 53–55, 69, 83, 84, 115–118, 120, 121, 125, 130, 134
- (PROFIBUS) Process Automation (PA)** Eine Feldbustechnologie .....  
 16, 18, 19, 22, 38, 45, 46, 54, 64, 65, 68, 69, 71–74, 81, 85, 86, 88, 89, 93, 100, 118, 128, 135, 143
- Process Field Bus (PROFIBUS)** Familie mehrerer Feldbustechnologien ((PROFIBUS) Dezentrale Peripherie, (PROFIBUS) Process Automation) .....  
 23, 38, 64, 68, 85, 86, 88, 89, 92, 93, 100, 101, 135, 140, 143
- Process Information and Management System (PIMS)** NAMUR-Arbeitsblatt 96 [208]:  
 „Ein beliebiges Prozessdatenarchiv als Informationssammler mit Visualisierungsmöglichkeiten“ .....  
 31, 55, 95, 129, 135
- PROFIBUS Nutzerorganisation (jetzt PROFIBUS International (PI)) (PNO)** PROFIBUS & PROFINET International .....  
 20, 69, 135
- Profil** (Applikationsprofil, Geräteprofil) Beschreibung einheitlicher Struktur und einheitlichen Verhaltens von Geräten (z. B. Feldgeräten) mit demselben Zweck mit dem Ziel besserer Austauschbarkeit für den Anwender .....  
 14, 15, 18, 22, 38, 40, 41, 46, 47, 54, 64, 68, 69, 86, 88, 93, 94, 100
- PROFINET** Familie mehrerer Industrial Ethernet Technologien (PROFINET IO, *PROFINET CBA*) .....  
 19, 23, 69
- PROFINET IO (PN IO)** Industrial-Ethernet-Technologie .....  
 18, 46, 69, 88, 135, 143
- PROLIST** Initiative zur elektronischen Beschreibung von Geräte-Merkmalen (2013 endgültig abgelöst durch eCl@ss) .....  
 14, 15, 84
- Prozess** Automatisierungsprozess; zu steuernder Ablauf; die Prozessdaten (Mess- und Stelldaten) werden von der Steuerung (z. B. SPS) im Prozessabbild gehalten ....  
 1, 2, 4, 8, 25, 29, 30, 33, 35, 37, 41, 48, 130
- Prozessleitsystem (PLS)** Verteiltes Bedien-Beobachtungssystem für prozesstechnische Anlagen .....  
 1, 2, 13–17, 24–32, 41, 47, 48, 103, 111, 114, 127, 129, 130, 135, 137, 140, 144

- Prozessleittechnik** (PLT) Gesamtheit der Überwachungs-, Bedien-, Steuer-, Regel-, und Sicherheitstechnik einer Anlage .....  
9, 12, 14, 15, 40, 57, 135
- Prozessnahe Komponente** (PNK) Hardwarekomponente eines Prozessleitsystems ....  
25, 135
- Remote IO** Modulares Feldgerät, das u. a. digitale und analoge Ein- und Ausgangsmodule besitzen kann .....  
15, 17, 54
- RoleClass** CAEX-Konstrukt, welches Funktionalitäten umsetzungsunabhängig festlegt  
9, 11, 12, 32, 36, 43, 45, 57, 58, 60–62, 77, 84, 97, 98, 106, 107, 109, 112, 115, 125,  
126
- R&I-Fließbild** Rohrleitungs- und Instrumentenfließbild .....  
9, 12, 29, 36, 41, 57, 58, 61, 99, 106, 114, 124, 163
- Speicherprogrammierbare Steuerung** (SPS) (mit IO-Baugruppen und Feldgeräten verbundene) Steuer- und Regelbaugruppe .....  
135
- Standard for the Exchange of Produkt Model Data** (STEP) Standardserie ISO 10303 zum Austausch von Produkt-Daten .....  
12, 135
- SystemUnitClass** CAEX-Konstrukt, welches konkrete wiederverwendbare Anlagenteile definiert .....  
9, 32, 44, 45, 58, 60, 61, 67, 68, 83, 84, 86, 96–98, 107, 115, 124, 126, 127, 129, 159
- User Interface Description** (UID) Teil eines FDI Packages, der die (mittels EDD notierte) Beschreibung von grafischen Benutzungsoberflächen für FDI Clients enthält .....  
23, 75, 83, 135
- User Interface Plugin** (UIP) Softwarekomponente auf .NET-Basis mit einer grafischen Benutzungsoberfläche für FDI Clients .....  
23, 83, 85, 135
- Variable** Konstrukt in der Electronic Device Description Language zur Deklaration von Speicherstellen im online- und offline-Gerätemodell (= Parameter) .....  
67, 85, 87, 88, 94, 102, 128
- Variable** Node zum Zugriff auf einen Wert im OPC UA-Adressraum .....  
26, 64, 66, 67, 72, 87, 91, 94, 101
- Windows Presentation Foundation** (WPF) deskriptive Technologie für grafische Benutzungsoberflächen für *Microsoft Windows* bzw. .NET .....  
23, 135

**XML Schema Description** (XSD) in XML notierte Formatbeschreibung und -einschränkung (Syntax) mit der impliziten Festlegung der Bedeutung der Elemente (Semantik) für XML-Dokumente ..... 59, 65, 88, 101, 135





# Literaturverzeichnis

- [1] S. Theurich, M. Stöß, M. Wollschlaeger und L. Urbas. „Communication and Information Engineering of FDI Equipment Packages“. In: *Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on*. Sep. 2012, S. 1–8. DOI: 10.1109/ETFA.2012.6489551.
- [2] S. Theurich und M. Wollschlaeger. „Entwicklung und Integration von FDI-Paketen für Apparate“. In: *Jahreskolloquium Kommunikation in der Automation, KomMA 2013*. 14. Nov. 2013.
- [3] S. Theurich, A. Schüller, M. Wollschlaeger und U. Epple. „Dienstbasierte Prüfung von CAEX-Exporten mit standardisierten Bibliotheken“. In: *Automation 2014 – Smart X Powered by Automation*. VDI-Verlag, Juli 2014. ISBN: 978-3-18-092231-7.
- [4] S. Theurich. *Dia2CAEX – Transformation Dia Sketch zu CAEX*. 23. Juni 2014. URL: [http://www.inf.tu-dresden.de/index.php?node\\_id=3601&ln=de](http://www.inf.tu-dresden.de/index.php?node_id=3601&ln=de) (besucht am 08.08.2014).
- [5] S. Theurich und M. Wollschlaeger. „Plant Asset Management und Condition Monitoring mit FDI“. In: *PI Konferenz 2015*. Hrsg. von A. Laubenstein. 12. März 2015.
- [6] S. Theurich. „Entwurf und Implementierung einer web-service-basierten Konfigurationslösung für ein netzwerkzentriertes Asset Management System“. Diplomarbeit. Technische Universität Dresden, Fakultät Informatik, Prozesskommunikation, 30. Okt. 2008.
- [7] S. Theurich, R. Frenzel, M. Wollschlaeger und T. Szczepanski. „Web-based Asset Management for heterogeneous industrial networks“. In: *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*. Juni 2008, S. 267–270. DOI: 10.1109/WFCS.2008.4638757.
- [8] S. Theurich, C. Hahn, R. Frenzel und M. Wollschlaeger. „Field bus abstraction as a means to enable net-work-independent applications Fieldbuses and Networks in Industrial and Embedded Systems“. In: *8th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems (2009)*. Mai 2009, S. 131–138. DOI: 10.3182/20090520-3-KR-3006.00020. URL: <http://www.ifac-paperonline.net/Detailed/46497.html>.
- [9] S. Theurich, R. Lehmann und M. Wollschlaeger. „Network asset models in intelligent field devices“. In: *Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on*. Mai 2010, S. 161–164. DOI: 10.1109/WFCS.2010.5548614.
- [10] S. Bens. „Konfigurationslösung für ein funktionsbasiertes Asset Management System“. Diplomarbeit. Technische Universität Dresden, 31. März 2010.

- [11] A. Neumann. „Umsetzung eines Gerätemodells und Zugriff mittels OPC UA“. Großer Beleg. Technische Universität Dresden, 13. Feb. 2013.
- [12] A. Neumann. „Instanziierung eines verteilten OPC UA-Gerätemodells und Erprobung der Zugriffsmechanismen“. Diplomarbeit. Technische Universität Dresden, Dez. 2013.
- [13] International Organization for Standardization, Hrsg. *ISO 15926-1: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part 1: Overview and fundamental principles*. Nov. 1999.
- [14] K. F. Früh und U. Maier. *Handbuch der Prozessautomatisierung – Prozessleittechnik für verfahrenstechnische Anlagen*. 3. Aufl. Oldenbourg Industrieverlag GmbH, 2004. ISBN: 3-486-27048-6.
- [15] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 3682: Formalisierte Prozessbeschreibungen*. Dez. 2003.
- [16] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62264-1: Integration von Unternehmensführungs- und Leitsystemen – Teil 1: Modelle und Terminologie (IEC 62264-1:2003); Deutsche Fassung EN 62264-1:2008*. Juni 2008.
- [17] International Electrotechnical Commission, Hrsg. *IEC 62424: Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*. Jan. 2010.
- [18] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 3695 Blatt 1: Engineering von Anlagen – Evaluieren und optimieren des Engineerings – Grundlagen und Vorgehensweise*. Nov. 2010.
- [19] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 3695 Blatt 2: Engineering von Anlagen – Evaluieren und optimieren des Engineerings – Themenfeld Prozesse*. Nov. 2010.
- [20] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 3695 Blatt 2: Engineering von Anlagen – Evaluieren und optimieren des Engineerings – Themenfeld Methoden*. Dez. 2010.
- [21] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 3695 Blatt 2: Engineering von Anlagen – Evaluieren und optimieren des Engineerings – Themenfeld Hilfsmittel*. Dez. 2010.
- [22] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 3695 Blatt 2: Engineering von Anlagen – Evaluieren und optimieren des Engineerings – Themenfeld Aufbauorganisation*. Aug. 2013.
- [23] L. Urbas und F. Doherr. „autoHMI: a model driven software engineering approach for HMIs in process industries“. In: *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*. Bd. 3. Juni 2011, S. 627–631. DOI: 10.1109/CSAE.2011.5952755.

- 
- [24] V. Dhole, R. Beck, D. Tremblay, D. Ajikutira und S. Mullick. *Transformation of Process Engineering – Innovations and Best Practices*. Hrsg. von Aspen Technology. 8. Aug. 2012. URL: <http://www.aspentech.com/WorkArea/DownloadAsset.aspx?id=15032387309> (besucht am 22.09.2014).
- [25] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 148: Anforderungen an die Automatisierungstechnik durch die Modularisierung verfahrenstechnischer Anlagen*. 22. Okt. 2013.
- [26] Deutsches Institut für Normung e. V., Hrsg. *DIN 19227 Teil 1:1993-10: Graphische Symbole und Kennbuchstaben für die Prozeßleittechnik*. Okt. 1993.
- [27] Deutsches Institut für Normung e. V., Hrsg. *ISO 10628-1: Schemata für die chemische und petrochemische Industrie – Teil 1: Spezifikation der Schemata (ISO/DIS 10628-1:2013); Deutsche Fassung prEN ISO 10628-1:2013*. Juli 2001.
- [28] Deutsches Institut für Normung e. V., Hrsg. *ISO 10628-2: Schemata für die chemische und petrochemische Industrie – Teil 2: Graphische Symbole (ISO 10628-2:2012); Deutsche Fassung EN ISO 10628-2:2012*. Apr. 2013.
- [29] International Electrotechnical Commission, Hrsg. *DIN EN 81346-2: Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte – Strukturierungsprinzipien und Referenzkennzeichnung – Teil 2: Klassifizierung von Objekten und Kennbuchstaben von Klassen (IEC 81346-2:2009); Deutsche Fassung EN 81346-2:2009*. Aug. 2007.
- [30] A. Wiesner, M. Wiedau, W. Marquardt, H. Richert, H. Temmen und M. Weiten. „Semantische Datenintegration für verfahrenstechnische Planungsdaten“. In: *5. Symposium ???* 4. Apr. 2008.
- [31] U. Epple. „Austausch von Anlagenplanungsdaten auf der Grundlage von Metamodellen“. In: *atp - automatisierungstechnische Praxis* (07 Juli 2003), S. 61–70.
- [32] M. Mertens und U. Epple. „Online Plant Asset Management auf Basis eines Merkmalverarbeitenden Strukturmodells“. In: *VDI-Berichte 1980 – Automation 2007*. VDI-Verlag, Juni 2007, S. 467–479. ISBN: 978-9-18-091980-5.
- [33] M. Mertens und U. Epple. „Plant asset management functions driven by property models“. In: *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*. Sep. 2009, S. 1–8. DOI: 10.1109/ETFA.2009.5347057.
- [34] W. Marquardt und M. Nagl. „Arbeitsprozessorientierte Unterstützung verfahrenstechnischer Entwicklungsprozesse“. In: *atp - automatisierungstechnische Praxis* (04 Apr. 2003), S. 52–58.
- [35] A. W. ans Michael Wiedau, W. Marquardt, H. T. ans Hannes Richert und F. Anhäuser. „Wissensbasierte Integation von Anlagenplanungsdaten – Semantische Technologien bieten großes Potential“. In: *atp edition* (4 Apr. 2010), S. 48–59.
- [36] A. Wiesner, J. Morbach und W. Marquardt. „Information integration in chemical process engineering based on semantic technologies“. In: *Computers and Chemical Engineering* 35 (4 Apr. 2011), S. 692–708.

- [37] International Electrotechnical Commission, Hrsg. *IEC 62424: Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*. 2014.
- [38] S. Schmitz, M. Schlütter und U. Epple. „R&I – Grundlage durchgängigen Engineerings“. In: *VDI Berichte 2032 – Automation 2008*. Juni 2008, S. 55–58.
- [39] S. Runde und A. Fay. „A data exchange format for the engineering of building automation systems“. In: *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. Sep. 2008, S. 303–310. DOI: 10.1109/ETFA.2008.4638411.
- [40] U. Epple, M. Rimmel und O. Drumm. „Modellbasiertes Format für RI-Informationen“. In: *atp edition* (Ausgabe 01-02/2011 Feb. 2011).
- [41] A. Schüller und U. Epple. „PandIX – Exchanging P&I diagram model data“. In: *Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on*. Sep. 2012, S. 1–8. DOI: 10.1109/ETFA.2012.6489537.
- [42] R. Drath und M. Barth. „Concept for managing multiple semantics with AutomationML – Maturity level concept of semantic standardization“. In: *Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on*. Sep. 2012, S. 1–8. DOI: 10.1109/ETFA.2012.6489538.
- [43] A. Schüller, A. Scholz, T. Tauchnitz, R. Drath und T. Scherwietes. „Speed-Standardisierung am Beispiel der PLT-Stelle Datenaustausch mit dem Namur-Datencontainer“. In: *atp edition* (01/02 Feb. 2015).
- [44] A. Lüder, N. Schmidt, O. Graeser, M. Thron und M. John. „Semantikdefinition durch Integration von Klassifikationssystemen in Entwurfsdaten zum verlustfreien Datenaustausch in Werkzeugketten“. In: *Automation 2014 – Smart X Powered by Automation*. Juli 2014.
- [45] AutomationML e. V. c/o IAF, Hrsg. *AutomationML Whitepaper Part 1 – Architecture and general requirements*. V 2.2. Juli 2013. URL: [https://www.automationml.org/o.red.c/dateien.html%20\(AutomationML%20Whitepaper%20Part~1\)](https://www.automationml.org/o.red.c/dateien.html%20(AutomationML%20Whitepaper%20Part~1)) (besucht am 14. 04. 2014).
- [46] R. Drath, A. Lüder, J. Peschke und L. Hundt. „AutomationML - the glue for seamless automation engineering“. In: *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. Sep. 2009, S. 616–623. DOI: 10.1109/ETFA.2008.4638461.
- [47] AutomationML e. V. c/o IAF, Hrsg. *AutomationML Whitepaper Part 2 – Role class libraries*. 7. Aug. 2013. URL: [https://www.automationml.org/o.red.c/dateien.html%20\(AutomationML%20Whitepaper%20Part~2\)](https://www.automationml.org/o.red.c/dateien.html%20(AutomationML%20Whitepaper%20Part~2)) (besucht am 14. 04. 2014).
- [48] AutomationML e. V. c/o IAF. *AutomationML Standard Libraries Part 1 – Part 4*. 12. März 2014. URL: [https://www.automationml.org/o.red.c/dateien.html%20\(AutomationML%20Standard%20Libraries%20Part~X\)](https://www.automationml.org/o.red.c/dateien.html%20(AutomationML%20Standard%20Libraries%20Part~X)) (besucht am 14. 04. 2014).

- 
- [49] AutomationML e. V. c/o IAF. *AutomationML Editor v3*. 30. Juli 2012. URL: [https://www.automationml.org/o.red.c/dateien.html%20\(Automation%20ML%20Editor%20v3\)](https://www.automationml.org/o.red.c/dateien.html%20(Automation%20ML%20Editor%20v3)) (besucht am 30.07.2012).
- [50] International Electrotechnical Commission, Hrsg. *IEC 62714-1: Engineering data exchange format for use in industrial automation systems engineering – Part 1: Architecture and General Requirements*. 27. Jan. 2014.
- [51] International Electrotechnical Commission, Hrsg. *IEC 62714-2: Engineering data exchange format for use in industrial automation systems engineering – Part 2: Role class libraries*. 17. Juni 2013.
- [52] SCRA. *STEP Application Handbook – ISO 10303 – Version 3*. 3. Aufl. 2006.
- [53] International Organization for Standardization, Hrsg. *ISO 13584: Industrial automation systems and integration – Parts library – Part 1: Overview and fundamental principles*. 15. Apr. 2001.
- [54] International Organization for Standardization, Hrsg. *ISO 15926-2: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part 2: Data model*. Nov. 1999.
- [55] International Organization for Standardization, Hrsg. *ISO 15926-3: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part 3: Methodology for the development and maintenance of reference data libraries*. Nov. 1999.
- [56] A. Laud. *ISO 15926 and interoperability using XMpLant technology*. Okt. 2009. URL: <https://www.posccaesar.org/svn/pub/PCA/MemberMeeting/200907/2-6-Adrian%20-%20ISO%2015926%20deployment%20using%20XMpLant%20Technology%20%20PC-1.pdf> (besucht am 15.04.2014).
- [57] Fiatech-PCA, Hrsg. *ISO 15926 P&ID Information Model Overview*. Version 1.1. 20. Aug. 2013. URL: [http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm%20\(2\)](http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm%20(2)) (besucht am 15.04.2014).
- [58] Fiatech-PCA, Hrsg. *ISO 15926 P&ID Profile Schema Specification*. Version 1.6. 26. Sep. 2013. URL: [http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm%20\(1\)](http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm%20(1)) (besucht am 15.04.2014).
- [59] Fiatech-PCA, Hrsg. *ISO 15926 P&ID Profile Template Specification*. Version 1.4. 3. Okt. 2013. URL: [http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm%20\(3\)](http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm%20(3)) (besucht am 15.04.2014).
- [60] M. Riechert. *Degussa PlantXML: Integrierter Planungsprozess mit flexiblen Bausteinen – Struktur, Einführung, Status –*. Foliensatz. 12. Feb. 2004.
- [61] H. Richert und H. Temmen. *Erfahrungen und Perspektiven nach 7 Jahren Degussa PlantXML*. 26. März 2010.
- [62] H. Temmen und C. Wittwer. *Welche Möglichkeiten eröffnet die ISO 15926 zur verbesserten Datenintegration?* Foliensatz. 2011.

- [63] T. Hadlich und C. Diedrich. „Verwendung von Merkmalen für die funktionale Modellierung – Merkmale in frühen Phasen des System-Engineering“. In: *Automation 2014 – Smart X Powered by Automation*. Juli 2014.
- [64] S. Becker und B. Westfechtel. „Integrationswerkzeug für verfahrenstechnische Entwicklungsprozesse – Konsistenzsicherung von Fließbild und Simulationsmodell“. In: *atp - automatisierungstechnische Praxis* (04 Apr. 2003), S. 59–65.
- [65] S. Beez, A. Fay und N. Thornhill. „Automatic generation of bond graph models of process plants“. In: *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. Sep. 2008, S. 1294–1301. DOI: 10.1109/ETFA.2008.4638567.
- [66] T. Schmidberger, A. Fay und R. Draht. „Automatisiertes Engineering von Prozessleitsystem-Funktionen“. In: *atp - automatisierungstechnische Praxis* (02 Feb. 2005), S. 45–51.
- [67] M. Schleipen, R. Drath und O. Sauer. „The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system“. In: *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*. Juni 2008, S. 1786–1791. DOI: 10.1109/ISIE.2008.4676932.
- [68] M. Wesche. „Komplexe Ereignisverarbeitung mittels OPC UA“. Diplomarbeit. Technische Universität Dresden, 22. Feb. 2010, S. 234–240. URL: [http://pure.ltu.se/portal/files/4922758/Proceedings\\_e-maintenance\\_Lulea\\_2010.pdf](http://pure.ltu.se/portal/files/4922758/Proceedings_e-maintenance_Lulea_2010.pdf).
- [69] A. Gössling, S. Theurich und M. Wollschlaeger. „Model-based Data Acquisition for improved eMaintenance strategies“. In: *1st International Workshop and Congress on eMaintenance*. Juni 2010, S. 234–240. URL: [http://pure.ltu.se/portal/files/4922758/Proceedings\\_e-maintenance\\_Lulea\\_2010.pdf](http://pure.ltu.se/portal/files/4922758/Proceedings_e-maintenance_Lulea_2010.pdf).
- [70] M. Barth und A. Fay. „Efficient use of data exchange formats in engineering projects by means of language integrated queries – Engineers LINQ to XML“. In: *IECON 2010 – 36th Annual Conference on IEEE Industrial Electronics Society*. Nov. 2010, S. 1335–1340. DOI: 10.1109/IECON.2010.5675490.
- [71] M. Graube, J. Pfeffer, J. Ziegler und L. Urbas. „Linked Data as integrating technology for industrial data“. In: *Int. J. of Distributed Systems and Technologies* 3 (3 2012). DOI: 10.4018/jdst.2012070104.
- [72] M. Mühlhause. „Konzept zur durchgängigen Nutzung von Engineeringmodellen der Automation“. Diss. Otto-von-Guericke-Universität Magdeburg, 30. Apr. 2012.
- [73] M. Mertens. „Verwaltung und Verarbeitung merkmalsbasierter Informationen: vom Metamodell zur technologischen Realisierung“. Diss. Rheinisch-Westfälischen Technischen Hochschule Aachen, 20. Dez. 2011.
- [74] U. Epple. „Merkmale als Grundlage der Interoperabilität technischer Systeme“. 11. Feb. 2011.
- [75] P. Zgorzelski. „Maschinen und Apparate im PROLIST-Engineering-Workflow“. In: *VDI Berichte 2171 – Automation 2012*. Juli 2012.

- [76] A. Selig. „Informationsmodell zur funktionalen Typisierung von Automatisierungsgeräten“. Diss. Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik der Universität Stuttgart, 2011.
- [77] C. Diedrich und H. Fichtner. *Ergebnisbericht Analyse relevanter Industriestandards*. Projektbericht. ifak Institut für Automatisierung und Kommunikation und Koramis GmbH, 24. Okt. 2005.
- [78] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 100: Nutzung von Merkmalleisten im PLT-Engineering-Workflow*. 31. Aug. 2006.
- [79] J. George. „Die Blaupause als Innovationsbremse – Standardisierter elektronischer Datenaustausch von der Planung bis zur Instandhaltung mittels des PROLIST-Engineering-Workflows“. In: *VDI Berichte 2143 – Automation 2011*. Juni 2011.
- [80] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61987-1: Industrielle Leittechnik – Datenstrukturen und -elemente in Katalogen der Prozessleittechnik – Teil 1: Messeinrichtungen mit analogen und digitalen Ausgängen (IEC 61987-1:2006); Deutsche Fassung EN 61987-1:2007*. Okt. 2007.
- [81] J. George und H. Uiterwyk. „Zukünftige Anwendung von Merkmalleisten und Klassifikation für Geräte der Prozessleittechnik innerhalb eClass“. In: *VDI Berichte 2209 – Automation 2013*. Juli 2013.
- [82] Deutsches Institut für Normung e. V., Hrsg. *PAS 1040 Reihe – Sachmerkmal-Leiste für Maschinen, Apparate und Rohrleitungen in der chemischen Industrie*. Juli 2004.
- [83] J. George. „Maschinenlesbare Merkmalleisten für Prozessautomatisierungsgeräte als Voraussetzung für die Mensch-Maschinen-Kommunikation und Maschinen-Maschinen-Kommunikation für aktuelle und zukünftige Anwendungen“. In: *Automation 2014 – Smart X Powered by Automation*. Juli 2014.
- [84] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61987-10: Industrielle Leittechnik – Datenstrukturen und -elemente in Katalogen der Prozessleittechnik – Teil 10: Merkmalleisten (ML) für den elektronischen Datenaustausch – Grundlagen (IEC 61987-10:2009); Deutsche Fassung EN 61987-10:2009*. Mai 2010.
- [85] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61987-10: Industrielle Leittechnik – Datenstrukturen und -elemente in Katalogen der Prozessleittechnik – Teil 11: Merkmalleisten für Messgeräte für elektronischen Datenaustausch – Allgemeine Strukturen (IEC 65E/152/CD:2009)*. Aug. 2010.
- [86] J. George, D. Großmann und A. Laubenstein. „PROLIST und FDI – Lückenlos von der Bestellung zum Gerätemanagement“. In: *VDI Berichte 2171 – Automation 2012*. Juli 2012.
- [87] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Arbeitsblatt 114: Best Practice Feldbusanwendungen – Auswahl, Planung, Montage, Inbetriebnahme und Betrieb von Feldbussen*. 3. Nov. 2006.
- [88] B. Reißweber. *Feldbussysteme zur industriellen Kommunikation*. 3. Auflage. Oldenbourg Industrieverlag GmbH, 2009. ISBN: 978-3-8356-3143-4.

- [89] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 74: NAMUR-Anforderung an einen Feldbus*. 12. Feb. 1997.
- [90] B. Vogel-Heuser. „Auswahl von Felddbussystemen und Remote I/O-Systemen – Kriterien und Systematik“. In: *atp - automatisierungstechnische Praxis* (04 Mai 2005), S. 87–95.
- [91] U. Enste und J. Müller. *Datenkommunikation in der Prozessindustrie – Darstellung und anwendungsorientierte Analyse*. 1. Aufl. Oldenbourg Industrieverlag GmbH, 2007. ISBN: 978-3-8356-3116-8.
- [92] CAN in Automation (CiA) e. V., Hrsg. *CiA 201 to 207 – CAN Application Layer for industrial applications*. Version 1.1. Feb. 1996.
- [93] M. Felser, Hrsg. *Dokumentensammlung – IEC 61158 Fieldbus*. 2013. URL: [http://www.felser.ch/download/index.htm?iec\\_61158\\_fieldbus.htm](http://www.felser.ch/download/index.htm?iec_61158_fieldbus.htm) (besucht am 29.04.2014).
- [94] HMS Industrial Networks. *Definition der Protokollfamilien*. 21. März 2014. URL: <http://www.felddbusse.de/Normung/protokollfamilien.shtml>.
- [95] International Electrotechnical Commission, Hrsg. *IEC 61158-1: Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 1: Overview and guidance for the IEC 61158 series*. Apr. 2003.
- [96] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 105: Anforderungen an die Integration von Felddbusgeräten in Engineering-Tools für Feldgeräte*. Aug. 2004.
- [97] IO-Link Consortium, Hrsg. *IO-Link Communication – Specification*. Jan. 2009.
- [98] IO-Link Consortium, Hrsg. *IO-Link Interface and System – Specification*. Juli 2013.
- [99] Deutsches Institut für Normung e. V., Hrsg. *DIN ISO 11898: Austausch digitaler Informationen – Steuergerätenetz (CAN) für schnellen Datenaustausch (ISO 11898:1993) (Stand 1994)*. Aug. 1995.
- [100] CAN in Automation (CiA) e. V., Hrsg. *CiA 301 – CANopen application layer and communication profile*. Version 4.2. 21. Feb. 2011.
- [101] F. Doherr, P. Hein, S. Söffing und T. Hauff. „Berücksichtigung von Infrastruktur-faktoren beim automatischen Kommunikationsengineering“. In: *VDI Berichte 2209 – Automation 2013*. Juli 2013.
- [102] K. Gnauck. „Entwicklung eines übergreifenden Modells zur Beschreibung intelligenter Feldgeräte“. Diplomarbeit. Technische Universität Dresden, 13. Apr. 2009.
- [103] PROFIBUS International, Hrsg. *PROFIBUS Specification: Amendment 3 to the Profibus Profile for Process Control Devices V 3.01: Identification and Maintenance Functions (I&M Function for PA)*. V 1.0. Dez. 2004.
- [104] PROFIBUS International, Hrsg. *PROFIBUS Guideline: Profile Guidelines Part 1: Identification & Maintenance Functions*. V 1.1.1. März 2005.



- 
- [105] C. Diedrich und T. Bangemann. *Profibus PA – Instrumentierungstechnologie für die Verfahrenstechnik*. 2. Aufl. Oldenbourg Industrieverlag GmbH, 2006. ISBN: 3-8356-3056-3.
- [106] acatech. *agendaCPS – Integrierte Forschungsagenda Cyber-Physical Systems*. Hrsg. von E. Geisberger und M. Broy. 2012. URL: [www.springerlink.com](http://www.springerlink.com).
- [107] Fachausschuss 7.20 - Cyber Physical Systems. *Stellungnahme Cyber Physical Systems*. Apr. 2013. URL: <http://www.vdi.de/technik/fachthemen/mess-und-automatisierungstechnik/fachbereiche/anwendungsfelder-der-automation/gma-fa-720-cyber-physical-systems/> (besucht am 15.04.2014).
- [108] Promotorengruppe Kommunikation der Forschungsunion Wirtschaft-Wissenschaft, Hrsg. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 – Abschlussbericht des Arbeitskreises Industrie 4.0 – Vorabversion*. 2012.
- [109] Bundesministerium für Bildung und Forschung, Hrsg. *Zukunftsbild „Industrie 4.0“*. Feb. 2014. URL: [www.bmbf.de/pubRD/Zukunftsbild\\_Industrie\\_40.pdf](http://www.bmbf.de/pubRD/Zukunftsbild_Industrie_40.pdf).
- [110] P. Neumann, R. Simon, C. Diedrich und M. Riedl. „Field device integration“. In: *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*. Bd. 2. Okt. 2001, S. 63–68. DOI: 10.1109/ETFA.2001.997672.
- [111] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 2657 Blatt 1: Middleware in der Automatisierungstechnik – Grundlagen*. Dez. 2010.
- [112] International Electrotechnical Commission, Hrsg. *IEC TR 62390: Common automation device – Profile guideline*. Jan. 2005.
- [113] International Organization for Standardization, Hrsg. *ISO 15745-2: Industrial automation systems and integration – Open systems application integration framework – Part 2: Reference description for ISO 11898-based control systems*. 15. Nov. 2003.
- [114] International Organization for Standardization, Hrsg. *ISO 15745-3: Industrial automation systems and integration – Open systems application integration framework – Part 3: Reference description for IEC 61158-based control systems*. 15. Nov. 2003.
- [115] PROFIBUS International, Hrsg. *Guideline for Application Profiles – Basics – Guideline for PROFINET*. Apr. 2009.
- [116] PROFIBUS International, Hrsg. *PROFIBUS Specification: Profile for Process Control Devices*. V 3.01. Dez. 2004.
- [117] PROFIBUS International, Hrsg. *PROFIBUS Specification: Amendment 1 to the Profibus Profile for Process Control Devices V 3.01: PROFIsafe for PA Devices*. V 1.0. Dez. 2004.
- [118] PROFIBUS International, Hrsg. *PROFIBUS Specification: Amendment 2 to the Profibus Profile for Process Control Devices V 3.01: Condensed Status and Diagnostic Messages*. V 1.0. Juni 2005.

- [119] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61804-2: Funktionsbausteine für die Prozessautomation – Teil 2: Festlegung des Funktionsbausteinkonzepts (IEC 61804-2:2006); Deutsche Fassung EN 61804-2:2007, Text Englisch*. DIN EN 61804-2:2007-10. Okt. 2007.
- [120] Open DeviceNet Vendor Association, Inc. und ControlNet International Ltd., Hrsg. *The CIP Networks Library – Common Industrial Protocol*. 3.1. Bd. 1. Nov. 2006.
- [121] S. Mätzler, S. Theurich, M. Wollschlaeger und M. Simros. „Rapid Prototyping von Applikationsprofil-konformen Feldgeräten“. In: *at - Automatisierungstechnik* (Feb. 2013). DOI: 10.1524/auto.2013.0036.
- [122] K. Bender, R. Birkhofer, M. Bregulla, F. Kuttig und H. Meyer. *DEKOS-Engineering-Modell für intelligente Automatisierungskomponenten*. Projektbericht. Informationstechnik im Maschinenwesen, TU München, Sep. 2002.
- [123] M. Riedl, R. Simon und M. Thron. *EDDL – Electronic Device Description*. 1. Aufl. Oldenbourg Industieverlag GmbH, 2002. ISBN: 3-486-27034-6.
- [124] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61804-3: Funktionsbausteine für die Prozessautomation – Teil 3: Elektronische Gerätebeschreibungssprache (EDDL) (IEC 61804-3:2006); Deutsche Fassung EN 61804-3:2007, Text Englisch*. DIN EN 618004-2:2007-10. Okt. 2007.
- [125] International Electrotechnical Commission, Hrsg. *IEC 61804-3: Function Blocks (FB) for Process Control – Part 3: Electronic Device Description Language (EDDL)*. 2nd. 2010.
- [126] International Electrotechnical Commission, Hrsg. *IEC 61804-3: Function Blocks (FB) for Process Control – Part 3: Electronic Device Description Language (EDDL)*. 2012.
- [127] International Electrotechnical Commission, Hrsg. *IEC 61804-5: Devices and integration in enterprise systems – Function blocks (FB) for process control and EDDL – Part 5: EDDL Built-in library*. 2012.
- [128] D. Großmann, D. John und A. Laubenstein. „EDDL Harmonisierung – Der Garant für den Erfolg der Field Device Integration“. In: *atp edition* (10-11 Nov. 2009), S. 6–14.
- [129] ifak system GmbH. *isEDD Workbench*. 2014. URL: <http://www.ifak-system.com/en/communication-automation/tools/products/isedd-workbench/> (besucht am 15.04.2014).
- [130] Siemens AG. *SIMATIC PDM Process Device Manager*. 2014. URL: <http://www.automation.siemens.com/mcms/process-control-systems/de/simatic-pcs-7/simatic-pcs-7-systemkomponenten/process-device-manager-pdm/pages/default.aspx> (besucht am 15.04.2014).
- [131] D. Großmann, K. Bender und B. Danzer. „FDT + EDD + OPC UA = FDD UA - Die einheitliche Formel für das Plant Asset Management in der Prozess- und Fertigungsautomatisierung“. In: *VDI Berichte 2032 – Automation 2008*. Juni 2008, S. 97–100.

- [132] R. Simon, T. Kleegrewe, R. Birkhofer, J. Jeske und O. Merget. *Field Device Tool – FDT*. 1. Aufl. Oldenbourg Industrieverlag GmbH, 2003. ISBN: 3-486-27044-3.
- [133] Deutsches Institut für Normung e. V., Hrsg. *DIN IEC 62453-1: Field Device Tool (FDT)-Schnittstellenfestlegung – Teil 1: Grundlegende Beschreibung und Begriffe (IEC 65C/442/CD:2006), Text Englisch*. DIN IEC 62453-1:2007-04. Apr. 2007.
- [134] Deutsches Institut für Normung e. V., Hrsg. *DIN IEC 62453-2: Field Device Tool (FDT)-Schnittstellenfestlegung – Teil 2: Begriffe und Beschreibung der Einzelheiten (IEC 65C/443/CD:2006), Text Englisch*. DIN IEC 62453-2:2007-04. Okt. 2007.
- [135] Deutsches Institut für Normung e. V., Hrsg. *DIN IEC 62453-301: Field Device Tool (FDT)-Schnittstellenspezifikation – Teil 301: Integration von Kommunikationsprofilen – Kommunikationsprofilfamilie (CPF) 1 nach IEC 61784; Englische Fassung IEC 65E/7/CD:2007*. DIN IEC 62453-301:2008-02. Feb. 2008.
- [136] Deutsches Institut für Normung e. V., Hrsg. *DIN IEC 62453-303-1: Field Device Tool (FDT)-Schnittstellenspezifikation – Teil 303-1: Integration von Kommunikationsprofilen – Kommunikationsprofile (CP) 3/1 und 3/2 nach IEC 61784; Englische Fassung IEC 65E/9/CD:2007*. DIN IEC 62453-303-1:2008-02. Feb. 2008.
- [137] International Electrotechnical Commission, Hrsg. *IEC 61784-1: Digital data communications for measurement and control – Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems*. Mai 2003.
- [138] PROFIBUS International, Hrsg. *PROFIBUS Guideline: DTM Styleguide – Guideline for the implementation of Device Type Managers (DTMs) for field devices*. Version 1.1. Nov. 2000.
- [139] PACTware Consortium e. V. *PACTware White-Paper*. 2010. URL: [http://www.pactware.com/downloads/pactware.html?tx\\_pxddamdownload\\_pi1\[download\]=145&tx\\_pxddamdownload\\_pi1\[file\]=66&tx\\_pxddamdownload\\_pi1\[mode\]=download&chash=ed8c74318f9a51ff0aa445e5a307eaf4](http://www.pactware.com/downloads/pactware.html?tx_pxddamdownload_pi1[download]=145&tx_pxddamdownload_pi1[file]=66&tx_pxddamdownload_pi1[mode]=download&chash=ed8c74318f9a51ff0aa445e5a307eaf4) (besucht am 15. 04. 2014).
- [140] Endress + Hauser Messtechnik. *FieldCare – Anlagen-Asset-Management – Volle Unterstützung im gesamten Anlagen-Lebenszyklus*. Apr. 2004. URL: [www.industrialnetworks.ch/dateien/aussteller/e+h/FieldCare.pdf%E2%80%8E](http://www.industrialnetworks.ch/dateien/aussteller/e+h/FieldCare.pdf%E2%80%8E) (besucht am 29. 04. 2014).
- [141] T. Henrichs und T. Schmidt. *Viel mehr als Instandhaltung: Plant Asset Management ist eine vielschichtige Aufgabe*. 25. Nov. 2008. URL: [http://www.process.vogel.de/msr\\_automatisierung/asset\\_management/fdt/articles/155367/](http://www.process.vogel.de/msr_automatisierung/asset_management/fdt/articles/155367/) (besucht am 29. 04. 2014).
- [142] L. Füchtler. „FieldCare + W@M = Life Cycle Management oder "Die Kunst die richtigen Informationen zur richtigen Zeit zur Verfügung zu stellen zu können"“. In: *VDI Berichte 2032 – Automation 2008*. Juni 2008, S. 379–382.
- [143] K. Bender, D. Großmann und B. Danzer. „FDT + EDD + OPCUA = FDD UA – Die Gleichung für eine einheitliche Gerätebeschreibung?“ In: *atp - automatisierungstechnische Praxis* (02 Feb. 2007), S. 48–54.

- [144] B. Danzer und D. Großmann. „FDI – The future of open device integration“. In: *Fieldbuses and Networks in Industrial and Embedded Systems – FET 2007*. Nov. 2007, S. 285–290. DOI: 10.3182/20071107-3-FR-3907.00040.
- [145] D. Großmann und K. Bender. „Future Device Integration – Eine Vision wird Wirklichkeit“. In: *VDI-Berichte Nr. 1980 – GMA-Kongress 2007 Baden-Baden*. Juni 2007.
- [146] D. Großmann, K. Bender und B. Danzer. „OPC UA based Field Device Integration“. In: *SICE Annual Conference 2008*. Aug. 2008.
- [147] FDI Cooperation. *Field Device Integration*. Foliensatz. Juni 2011.
- [148] Fieldbus Foundation. *Field Device Integration (FDI)*. White Paper. Juni 2013.
- [149] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-1: Field Device Integration (FDI) – Part 1: Overview*. Nov. 2013.
- [150] H. Kupfmüller und R. Lange. „FDI Device Integration – Best of Both Worlds“. In: *atp edition* (6 Juni 2010), S. 16–19.
- [151] FDI Cooperation. *FDI - Field Device Integration Technology*. White Paper. 23. Jan. 2013. URL: [http://www.fdi-cooperation.com/tl\\_files/images/content/Publications/FDI-White\\_Paper.pdf](http://www.fdi-cooperation.com/tl_files/images/content/Publications/FDI-White_Paper.pdf).
- [152] D. Großmann, M. Braun, B. Danzer und M. Riedl. *FDI – Field Device Integration – Handbuch für die einheitliche Intergrationstechnologie*. 1. Aufl. VDE Verlag GmbH, 2013. ISBN: 978-3-8007-3513-6.
- [153] D. Großmann und W. Manke. „Geräteintegration mit FDI und OPC UA – Mit standardisierten Informationsmodell Geräte integrieren“. In: *atp edition* (11 Nov. 2012), S. 56–65.
- [154] FDI Cooperation. „Validation Status for FDI Technical Specification – V 0.9.0“. 5. Nov. 2013.
- [155] M. Pelz. *FDI (Field Device Integration) – Manufacturer survey and actual Status*. Foliensatz. Nov. 2010.
- [156] FDI Cooperation. „Validation Status for FDI Technical Specification – V 1.0.1“. 1. Dez. 2014.
- [157] ABB Asea Brown Boveri Ltd. *ABB Field Information Manager (FIM) – Introduction and key features*. Apr. 2015. URL: [https://library.e.abb.com/public/0f4915474da62cfac1257e3700452b02/2PAA114401\\_C\\_en\\_ABB\\_FIM\\_Customer\\_Introduction\\_and\\_Key\\_features.pdf](https://library.e.abb.com/public/0f4915474da62cfac1257e3700452b02/2PAA114401_C_en_ABB_FIM_Customer_Introduction_and_Key_features.pdf) (besucht am 13.11.2015).
- [158] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-7: Field Device Integration (FDI) – Part 7: Communication Devices*. Nov. 2013.
- [159] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-3: Field Device Integration (FDI) – Part 3: FDI Server*. Nov. 2013.
- [160] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-5: Field Device Integration (FDI) – Part 5: FDI Information Model*. Nov. 2013.
- [161] International Electrotechnical Commission, Hrsg. *IEC 62541-100: OPC Unified Architecture – OPC UA for Devices*. 2013.

- 
- [162] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-2: Field Device Integration (FDI) – Part 2: FDI Client*. Nov. 2013.
- [163] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-4: Field Device Integration (FDI) – Part 4: FDI Packages*. Nov. 2013.
- [164] International Organization for Standardization und International Electrotechnical Commission, Hrsg. *ISO/IEC29500-2: Information technology – Document description and processing languages – Office Open XML File Formats – Part 2: Open Packaging Conventions*. 15. Nov. 2008.
- [165] International Electrotechnical Commission, Hrsg. *IEC 61804-4: Function Blocks (FB) for Process Control and Electronic Device Description Language (EDDL) – Part 4: EDD interpretation*. 2014.
- [166] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-6: Field Device Integration (FDI) – Part 6: Technology Mapping*. Nov. 2013.
- [167] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-101-1: Field Device Integration (FDI) – Profiles – Part 101-1: Foundation Fieldbus H1*. Nov. 2013.
- [168] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-101-2: Field Device Integration (FDI) – Profiles – Part 101-2: Foundation Fieldbus HSE*. Nov. 2013.
- [169] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-103-1: Field Device Integration (FDI) – Profiles – Part 103-1: PROFIBUS*. Nov. 2013.
- [170] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-103-4: Field Device Integration (FDI) – Profiles – Part 103-4: PROFINET*. Nov. 2013.
- [171] International Electrotechnical Commission, Hrsg. *CDV IEC 62769-109-1: Field Device Integration (FDI) – Profiles – Part 109-1: HART and WirelessHART*. Nov. 2013.
- [172] D. Schulz, R. Braun, U. Topp und M. Stöckl. „FDI für elektrische Subsysteme – Modulare Geräte und proprietäre Protokolle“. In: *Automation 2014 – Smart X Powered by Automation*. Juli 2014.
- [173] D. Schulz, R. Braun, U. Topp und M. Stöckl. „Beyond Instrumentation – FDI for Modular Subsystems with Proprietary Protocols“. In: *Emerging Technologies Factory Automation (ETFA), 2014 IEEE 19th Conference on*. Sep. 2014.
- [174] W. Chin. „FDT 2.0 Takes Flight“. In: *ARC view* (März 2011).
- [175] FDT Group, Hrsg. *FDT2 – Technical Specification*. 25. Apr. 2012.
- [176] S. Gizy, M. Gunzert und K.-P. Lindner. „Der Brückenschlag zwischen FDT und FDI“. In: *atp - automatisierungstechnische Praxis* (01 Jan. 2010), S. 69–72.
- [177] U. Epple, M. Gote, G. Hartmann, A. Horch und F. Uecker. *Plant Asset Management in der Praxis – Wunsch und Wirklichkeit*. Hrsg. von ZVEI und NAMUR. Foliensatz. Apr. 2006.

- [178] The Instrumentation, Systems, and Automation Society, Hrsg. *ISA Draft 95.00.3 (S95): Enterprise – Control System Integration – Part 3: Activity Models of Manufacturing Operations Management (Draft 21)*. Jan. 2005.
- [179] Verein Deutscher Ingenieure, Hrsg. *VDI 5600 Blatt 1 (Entwurf): Fertigungsmanagementsysteme (Manufacturing Execution Systems – MES)*. Jan. 2015.
- [180] Verein Deutscher Ingenieure, Hrsg. *VDI 5600 Blatt 2 (Ausgabe deutsch/englisch): Fertigungsmanagementsysteme (Manufacturing Execution Systems – MES) – Wirtschaftlichkeit*. März 2013.
- [181] Verein Deutscher Ingenieure, Hrsg. *VDI 5600 Blatt 3 (Ausgabe deutsch/englisch): Fertigungsmanagementsysteme (Manufacturing Execution Systems – MES) – Logische Schnittstelle zur Maschinen- und Anlagensteuerung*. Juli 2013.
- [182] F. Iwanitz und J. Lange. *OPC – Fundamentals, Implementation, and Application*. 3. Aufl. Hüthig GmbH & Co. KG Heidelberg, 2006. ISBN: 3-7785-2904-8.
- [183] W. Manke, S. Leitner und M. Damm. *OPC Unified Architecture*. 2009. ISBN: 978-3-540-68898-3.
- [184] International Electrotechnical Commission, Hrsg. *DIN CLC/TR 62541-1: OPC Unified Architecture – Teil 1: Übersicht und Konzepte*. Juli 2011.
- [185] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62541-2: OPC Unified Architecture – Teil 2: Modell für die IT-Sicherheit (IEC 65E/93/CDV:2008); Englische Fassung FprEN 62541-2:2008*. Dez. 2008.
- [186] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62541-3: OPC Unified Architecture – Teil 3: Adressraummodell (IEC 62541-3:2010); Englische Fassung EN 62541-3:2010*. Sep. 2011.
- [187] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62541-5: OPC Unified Architecture – Teil 5: Informationsmodell (IEC 62541-5:2011); Englische Fassung EN 62541-5:2011*. Juni 2012.
- [188] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62541-8: OPC Unified Architecture – Teil 8: Zugriff auf Automatisierungsdaten (IEC 62541-8:2011); Englische Fassung EN 62541-8:2011*. Juni 2012.
- [189] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62541-9: OPC Unified Architecture – Teil 9: Alarme und Zustände (IEC 62541-9:2012); Englische Fassung EN 62541-9:2012*. Juni 2013.
- [190] OPC Foundation, Hrsg. *OPC Unified Architecture for Devices (DI) Companion Specification*. Release 1.00. Dez. 2009.
- [191] OPC Foundation, Hrsg. *OPC Unified Architecture Specification Part 13: Aggregates*. 1. Aug. 2012.
- [192] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 91: Anforderungen an Systeme für Anlagennahes Asset Management*. Nov. 2001.
- [193] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 2651 Blatt 1: Plant Asset Management (PAM) in der Prozessindustrie Definition, Modell, Aufgabe, Nutzen*. Sep. 2009.

- [194] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 129: Plant Asset Management*. Okt. 2009.
- [195] A. Horch, G. Gonsior und H. Grieb. „Plant Asset Management – gemeinsames Verständnis und breiter Anwendernutzen“. In: *atp edition* (Ausgabe 10 Okt. 2008). URL: <https://www.di-verlag.de/de/Zeitschriften/atp-edition/2008/10/Plant-Asset-Management-gemeinsames-Verstaendnis-und-breiter-Anwendernutzen>.
- [196] J. Berge. *ISA 108 – Intelligent Device Management (IDM)*. 7. Mai 2014. URL: [http://www.fieldbus.org/images/stories/international/asiapacific/Indonesia/2014-05-07-Jakarta/5\\_idm\\_isa108\\_jonas.pdf](http://www.fieldbus.org/images/stories/international/asiapacific/Indonesia/2014-05-07-Jakarta/5_idm_isa108_jonas.pdf) (besucht am 01.12.2015).
- [197] K.-P. Lindner. *ISA 108 – Intelligent Device Management (IDM)*. Foliensatz. 30. Juni 2015.
- [198] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI/VDE 2651 Blatt 2 (Entwurf): Plant Asset Management (PAM) in der Prozessindustrie – Spezifikation und Methoden*. Sep. 2015.
- [199] M. Wollschlaeger, C. Diedrich, T. Bangemann, J. Müller und U. Epple. „Integration of fieldbus systems into on-line asset management solutions based on fieldbus profile descriptions“. In: *Factory Communication Systems, 2002. 4th IEEE International Workshop on*. 2002, S. 89–96. DOI: 10.1109/WFCS.2002.1159705.
- [200] Honeywell International Inc. *IntelliScout – Automated Evaluation of Equipment Performance*. White Paper. Aug. 2000. (Besucht am 10.04.2012).
- [201] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 107: Selbstüberwachung und Diagnose von Feldgeräten*. 12. Juni 2006.
- [202] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI 2650 Blatt 1: Anforderungen an Selbstüberwachung und Diagnose in der Feldinstrumentierung – Allgemeine Anforderungen*. Aug. 2006.
- [203] Verein Deutscher Ingenieure und Verband der Elektrotechnik, Elektronik, Informationstechnik, Hrsg. *VDI 2650 Blatt 2: Anforderungen an Selbstüberwachung und Diagnose in der Feldinstrumentierung – Allgemeine Fehler und Fehlerzustände von Feldgeräten*. Aug. 2006.
- [204] T. Bierweiler, M. Borrmann, W. Ens, H. Lenz und C. Paulitsch. „Methoden zur Selbstüberwachung von Prozesssensoren“. In: *VDI Berichte 2032 – Automation 2008*. Juni 2008, S. 19–22.
- [205] M. Birkenkamp und B. Peters. „Condition Monitoring: Ein Tool zur operativen oder strategischen Prozessführung? – Condition Monitoring im Umfeld von Plant Asset Management“. In: *VDI Berichte 2032 – Automation 2011*. Juli 2011.
- [206] Siemens AG. *Überwachung von Kreiselpumpen mit dem „PumpMon“-Baustein in SIMATIC PCS 7 – SIMATIC PCS 7 Condition Monitoring Library*. Feb. 2014. URL: <http://support.automation.siemens.com/DE/view/de/42460161> (besucht am 21.03.2014).

- [207] Siemens AG. *Ventilüberwachung und Diagnose mit dem „VlvMon“-Baustein in SIMATIC PCS 7 – SIMATIC PCS 7 Condition Monitoring Library*. Feb. 2014. URL: <http://support.automation.siemens.com/DE/view/de/87604493> (besucht am 21.03.2014).
- [208] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Arbeitsblatt 96: Prozessdiagnose – ein Statusbericht*. Nov. 2002.
- [209] G. Dünnebier und M. v. Felde. „Performance Monitoring – Ein entscheidender Beitrag zur Optimierung der Betriebsführung“. In: *Chemie Ingenieur Technik* 75 (05 Mai 2003), S. 529–533.
- [210] R. Jorewitz, A. Münnemann, U. Epple, W. Böckler, W. Wille und R. Schmitz. „Automation of Performance Monitoring in an Industrial Environment“. In: *2nd European Conference on Electrical and Instrumentation Applications in the Petroleum and Chemical Industry, Conference Proceedings*. Okt. 2005, S. 116–120.
- [211] L. Urbas, M. Obst und M. Stöß. „Formal Models for High Performance HMI Engineering“. In: *Proceedings IFAC MathMod 2012*. Feb. 2012, S. 854–859. DOI: 10.3182/20120215-3-AT-3016.00151.
- [212] M. Obst, S. Runde, G. Wolf und L. Urbas. „Integration requirements of package units – A description approach with FDI“. In: *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*. Sep. 2013, S. 1–8. DOI: 10.1109/ETFA.2013.6647974.
- [213] Chemstations, Inc. *CHEMCAD Feature List*. 13. Juni 2013. URL: <http://www.chemstations.com/> (besucht am 13.11.2015).
- [214] Process Systems Enterprise Limited. *PSE: gPROMS Process Builder - Environment & features*. 13. Nov. 2015. URL: <http://www.psenterprise.com/processbuilder/features.html> (besucht am 13.11.2015).
- [215] M. Barth. „Automatisch generierte Simulationsmodelle verfahrenstechnischer Anlagen für den Steuerungstest“. Diss. Helmut-Schmidt-Universität Hamburg, Fakultät Maschinenbau, 2011.
- [216] J. H. Taylor und A. F. Sayda. „Prototype Design of A Multi-agent System for Integrated Control and Asset Management of Petroleum Production Facilities“. In: *2008 American Control Conference*. Juni 2008.
- [217] T. Müller-Heinzerling. „Online-Diagnose mit Partial Stroke Test – Asset Management für mechanische Anlagenkomponenten“. In: *VDI-Berichte Nr. 1980 – Automation 2007*. Juni 2007.
- [218] T. Müller-Heinzerling. „Asset Management für die „eigentlichen“ Anlagen-Assets – Condition Monitoring ohne aufwendige Zusatzinstrumentierung“. In: *VDI Berichte 2067 – Automation 2014*. Juni 2009.
- [219] S. Kempf und L. Glathe. „Moderne Anlagenleitstände und Bedienkonzepte – HMI+ unterstützt operative Prozessführung industrieller Produktionsprozesse durch benutzerzentrierte Prozessvisualisierung“. In: *VDI-Berichte 2143 – AUTOMATION 2011*. Juni 2011.



- [220] Siemens AG. *Integration von SIMATIC PCS 7 Asset Management in bestehende Projekte – Projektierbeispiel*. 8. März 2010. URL: <http://support.automation.siemens.com/DE/llisapi.dll?func=cslib.csinfo&objId=27833758&objAction=csOpen&nodeid4=20806759&lang=de&siteid=cseus&aktprim=4&extranet=standard&viewreg=DE> (besucht am 21.03.2014).
- [221] Siemens AG. *Anlagennahes Asset Management mechanischer Anlagenkomponenten*. White Paper. Mai 2011. URL: [http://www.siemens.com/press/de/materials/industry/ia/white\\_paper.php](http://www.siemens.com/press/de/materials/industry/ia/white_paper.php) (besucht am 15.01.2014).
- [222] Siemens AG. *PCS 7 in Practice – Plant Asset Management – SIMATIC PCS 7 V7.x – Applikationsbeispiel*. März 2012. URL: <http://support.automation.siemens.com/DE/llisapi.dll?func=cslib.csinfo&objId=56285991&objAction=csOpen&nodeid4=20806759&lang=de&siteid=cseus&aktprim=4&extranet=standard&viewreg=DE> (besucht am 21.03.2014).
- [223] ABB Asea Brown Boveri Ltd. *800xA Asset Optimization – Allgemein*. 2014. URL: <http://www.abb.com/product/seitp334/bb06198da13c99eb8525719a004f4187.aspx?productLanguage=ge&country=00&tabKey=1> (besucht am 21.03.2014).
- [224] ABB Asea Brown Boveri Ltd. *800xA Asset Optimization – Technische Daten*. 2014. URL: <http://www.abb.com/product/seitp334/bb06198da13c99eb8525719a004f4187.aspx?productLanguage=ge&country=00&tabKey=2> (besucht am 21.03.2014).
- [225] ABB Asea Brown Boveri Ltd. *Real-TPI – Allgemein*. 2014. URL: <http://www.abb.com/product/seitp334/b608d498ec126606c125711f00440220.aspx?productLanguage=ge&country=00&tabKey=1> (besucht am 21.03.2014).
- [226] ABB Asea Brown Boveri Ltd. *Real-TPI – Allgemein*. 2014. URL: <http://www.abb.com/product/seitp334/b608d498ec126606c125711f00440220.aspx?productLanguage=ge&country=00&tabKey=2> (besucht am 21.03.2014).
- [227] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61131-3: Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen*. Dez. 2003.
- [228] PLCopen – TC6. *XML intro*. 2014. URL: [http://www.plcopen.org/pages/tc6\\_xml/xml\\_intro/index.htm](http://www.plcopen.org/pages/tc6_xml/xml_intro/index.htm) (besucht am 09.09.2014).
- [229] PLCopen TC 6, Hrsg. *XML Formats for IEC 61131-3 – PLCopenXML*. 8. Mai 2009.
- [230] M. Simros, S. Theurich und M. Wollschlaeger. „Programming embedded devices in IEC 61131-languages with industrial PLC tools using PLCopen XML“. In: *CONTROLO 2012 – 10th Portuguese Conference on Automatic Control*. Funchal, Portugal, Juli 2012.
- [231] Verband Deutscher Maschinen- und Anlagenbauer e. V., Hrsg. *VDMA Einheitsblatt 24582 – Feldbusneutrale Referenzarchitektur für Condition Monitoring in der Fabrikautomation (Entwurf)*. Mai 2013.
- [232] International Organization for Standardization, Hrsg. *ISO 13372: Condition monitoring and diagnostics of machines – Vocabulary*. 1. März 2012.

- [233] International Organization for Standardization, Hrsg. *ISO 13374: Condition monitoring and diagnostics of machines – Data processing, communication and presentation – Part 1: General guidelines*. 15. März 2003.
- [234] International Organization for Standardization, Hrsg. *ISO 13374: Condition monitoring and diagnostics of machines – Data processing, communication and presentation – Part 1: Data processing*. 15. Juli 2007.
- [235] International Organization for Standardization, Hrsg. *ISO 13374: Condition monitoring and diagnostics of machines – Data processing, communication and presentation – Part 1: Communication*. 15. Feb. 2012.
- [236] Deutsches Institut für Normung e. V., Hrsg. *Zustandsüberwachung und -diagnostik von Maschinen – Allgemeine Anleitungen (ISO 17359:2011)*. Sep. 2011.
- [237] D. H. Zhang, J. B. Zhang, M. Luo, Y. Tang und L. Q. Zhuang. „A reference architecture and functional model for monitoring and diagnosis of large automated systems“. In: *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*. Bd. 2. Sep. 2003, 516–523 vol.2. DOI: 10.1109/ETFA.2003.1248742.
- [238] H. Meyer, K. Bregulla und K. Bender. „Einheitliche Gerätefunktion für ein durchgängiges Engineering am Beispiel einer FDT/DTM-Implementierung“. In: *Tagungsband Fachtagung "Verteilte Automation 2000"*. März 2000.
- [239] T. Schmidberger, A. Fay, R. Draht und A. Horch. „Von Anlagenstrukturinformationen automatisch zum Asset-Management“. In: *atp – Automatisierungstechnische Praxis* (06 Juni 2006), S. 54–61.
- [240] R. Isermann. „Wissensbasierte Überwachung und Fehlerdiagnose von Prozessen der Automatisierungstechnik“. In: *VDI Berichte 2067 – Automation 2014*. Juni 2009.
- [241] J. Pathak, Y. Jiang, V. Honavar und J. McCalley. „Condition Data Aggregation with Application to Failure Rate Calculation of Power Transformers“. In: *Proceedings of the 39th Hawaii International Conference on System Sciences*. 2006.
- [242] A. Lee und J. Lastra. „Data aggregation at field device level for industrial ambient monitoring using Web Services“. In: *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*. Juli 2011, S. 491–496. DOI: 10.1109/INDIN.2011.6034929.
- [243] Bayer Technology Services, Hrsg. *PUMon – Ein neues Werkzeug zur Überwachung und Statusermittlung von Anlagenteilen*. 8. Apr. 2010. URL: [http://www.bayer-technology.com/uploads/media/BTS\\_Flyer\\_PUMon\\_080410.pdf](http://www.bayer-technology.com/uploads/media/BTS_Flyer_PUMon_080410.pdf) (besucht am 14.09.2014).
- [244] T. Ross und C. W. Frey. „Statusermittlung nicht-intelligenter Prozessabschnitte mit dem Softwarewerkzeug PUMon“. In: *Jahrbuch Maintenance 2014*. Hrsg. von T.A. Cook. 14. Feb. 2014, S. 78–79.

- [245] M. Barth, M. Strube, A. Fay, P. Weber und J. Greifeneder. „Object-oriented engineering data exchange as a base for automatic generation of simulation models“. In: *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*. Nov. 2009, S. 2465–2470. DOI: 10.1109/IECON.2009.5415229.
- [246] M. Barth, P. Weber, A. Fay und J. Greifeneder. „Semiautomatisch generierte Element- und Prozesssimulationsmodelle für PLS-Tests“. In: *VDI-Berichte 2067 – AUTOMATION 2009*. Juni 2006.
- [247] Deutsches Institut für Normung e. V., Hrsg. *DIN ISO 17359: Zustandsüberwachung und -diagnostik von Maschinen – Beiblatt 1: Erläuterungen zu Fachbegriffen*. Aug. 2007.
- [248] Verein Deutscher Ingenieure, Hrsg. *VDI 2888: Zustandsorientierte Instandhaltung*. Dez. 1999.
- [249] Verein Deutscher Ingenieure, Hrsg. *VDI 2889: Einsatz wissensbasierter Methoden in der Instandhaltung*. Apr. 1998.
- [250] J. Krause, S. Cech, F. Rosenthal, A. Gössling, C. Groba und V. Vasyutynskyy. „Factory-wide predictive maintenance in heterogeneous environments“. In: *Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on*. Mai 2010, S. 153–156. DOI: 10.1109/WFCS.2010.5548616.
- [251] J. Krause. „Kontextsensitive Prognoseverfahren für das Abnutzungsverhalten von technischen Systemen“. Diss. TU Dresden, Jan. 2014.
- [252] Verein Deutscher Ingenieure, Hrsg. *VDI 4499 Blatt 1: Digitale Fabrik – Grundlagen*. Feb. 2008.
- [253] Verein Deutscher Ingenieure, Hrsg. *VDI 4499 Blatt 2: Digitale Fabrik – Digitaler Fabrikbetrieb*. Feb. 2008.
- [254] International Electrotechnical Commission, Hrsg. *IEC CD 62832: Industrial Process measurement, Control and Automation -- Reference model for representation of production facilities (Digital Factory)*. Apr. 2014.
- [255] S. Behrendt und et.al. *Integrierte Technologie-Roadmap Automation 2015 +*. ZVEI – Zentralverband Elektrotechnik- und Elektronikindustrie e. V. Fachverband Automation, Okt. 2006.
- [256] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 139: Informationsschnittstellen in der Prozessautomatisierung – Betriebliche Eigenschaften*. 19. März 2012.
- [257] M. Mertens, G. Quirós und U. Epple. „Unterstützung von Diagnoseanwendungen durch Flussweginformationen“. In: *VDI Berichte 2032 – Automation 2008*. Juni 2008, S. 23–26.
- [258] M. Schlütter. *Über die Interpretation von Anlagenplanungsdaten – Ein Erfahrungsbericht*. Foliensatz. 18. Feb. 2008.
- [259] D. Großmann, A. Kaiser und H. Zipper. „Mit FDI Tools einfach und effizient FDI Packages entwickeln“. In: Hrsg. von A. Laubenstein. 12. März 2015.
- [260] H. Zipper. „FDI Quality Assurance“. In: Hrsg. von A. Laubenstein. 12. März 2015.

- [261] R. Drath, A. Fay und T. Schmidberger. „Computer-aided design and implementation of interlock control code“. In: *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. Okt. 2006, S. 2653–2658. DOI: 10.1109/CACSD-CCA-ISIC.2006.4777057.
- [262] D. Großmann, M. Bregulla, S. Banerjee, D. Schulz und R. Braun. „Auf dem Weg zum Internet of Portals – Anwendungen für OPC UA Server Aggregation“. In: *atp edition* (07/08 Aug. 2014).
- [263] FDI Cooperation. „FDI Status – TC4 Meeting“. 2. Juni 2008.
- [264] S. Bukva, U. Enste und F. Uecker. „Selbstkonfiguration und automatisiertes Änderungsmanagement von MES-Systemen“. In: *atp – Automatisierungstechnische Praxis* (08 Aug. 2009).
- [265] K. Büttner und A. Hennecke. „Profibus in der Prozessautomation – Ausgedieht oder zeitgemäß?“. In: *PI Konferenz 2015*. März 2015.
- [266] T. Hadlich, M. Muehlhause und C. Diedrich. „Discovery and integration of information in a heterogeneous environment“. In: *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*. Sep. 2010, S. 1–8. DOI: 10.1109/ETFA.2010.5641363.
- [267] NAMUR – Interessengemeinschaft Automatisierungstechnik in der Prozessindustrie. *FDI-User-Requirements 2011, in addition to NE 105*. 19. Okt. 2011. URL: [http://www.namur.de/nc/publikationen-und-news/news/news/article/fdi-user-requirements-2011-in-addition-to-ne-105/?tx\\_ttnews\[backPid\]=156](http://www.namur.de/nc/publikationen-und-news/news/news/article/fdi-user-requirements-2011-in-addition-to-ne-105/?tx_ttnews[backPid]=156) (besucht am 27.06.2014).
- [268] J. George, D. Großmann und M. Pelz. „Vorausschauende Instandhaltung durch standort-übergreifende Analyse von Feldgerätedaten“. In: *PI Konferenz 2015*. März 2015.
- [269] J. Aro. *Subscribing to a very large number of Nodes in a Prosys OPC UA Java SDK based Server*. 17. Juni 2013. URL: <http://www.prosysopc.com/blog/2013/06/17/jouni/subscribing-to-a-very-large-number-of-nodes-in-a-prosys-opc-ua-java-sdk-based-server/> (besucht am 27.09.2013).
- [270] M. Schleipen und M. Okon. „The CAEX tool suite - User assistance for the use of standardized plant engineering data exchange“. In: *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*. Sep. 2010, S. 1–7. DOI: 10.1109/ETFA.2010.5641284.
- [271] CodeSynthesis Tools CC. *XSD: XML Data Binding for C++*. 1. Apr. 2015. URL: <http://www.codesynthesis.com/products/xsd/>.
- [272] Die Dia Entwickler. *Dia - Ein Zeichenprogramm – Überblick*. 2014. URL: <http://dia-installer.de/index.html.de> (besucht am 29.04.2014).
- [273] World Wide Web Consortium. *XSL Transformations (XSLT) Version 2.0 – W3C Recommendation*. 23. Jan. 2007. URL: <http://www.w3.org/TR/xslt20/> (besucht am 29.04.2014).

- [274] A. Ohle. „CO<sub>2</sub>-Abtrennung aus Gasströmen durch Absorption in Poly(methylglykol)amin“. Diss. Technische Universität Dresden, 10. Aug. 2009.
- [275] PROFIBUS International, Hrsg. *PROFIBUS TC/WG Draft Guideline: EDD Interoperability Guideline*. März 2005.
- [276] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 81346-1: Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte – Strukturierungsprinzipien und Referenzkennzeichnung – Teil 1: Allgemeine Regeln (IEC 81346-1:2009); Deutsche Fassung EN 81346-1:2009*. Mai 2010.
- [277] FDI Cooperation, Hrsg. *Field Device Integration FDI Usability Style Guide*. 2013.
- [278] Deutsches Institut für Normung e. V., Hrsg. *DIN IEC 60025-341: Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik (IEC 1/2201/FDIS:2012)*. Juli 2013.
- [279] J.-M. Dautelle. *JScience – Java Tools and Libraries for the Advancement of Sciences*. 6. Jan. 2014. URL: <http://jscience.org/> (besucht am 06.01.2014).
- [280] W. Keil, Hrsg. *UOMu*. 2014. URL: <http://www.eclipse.org/uomo/> (besucht am 29.04.2014).
- [281] Deutsches Institut für Normung e. V., Hrsg. *DIN 1301-1: Einheiten – Teil 1: Einheitenamen, Einheitenzeichen*. Okt. 2010.
- [282] Deutsches Institut für Normung e. V., Hrsg. *DIN EN ISO 80000-1: Größen und Einheiten – Teil 1: Allgemeines (ISO 80000-1:2009 + Cor 1:2011); Deutsche Fassung EN ISO 80000-1:2013*. Aug. 2013.
- [283] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61360-1: Genormte Datenelementtypen mit Klassifikationsschema für elektrische Bauteile – Teil 1: Definitionen – Regeln und Methoden (IEC 61360-1:2002 + A1:2003); Deutsche Fassung EN 61360-1:2002 + A1:2004*. DIN EN 61360-1:2004-12. Dez. 2004.
- [284] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 60027-1: Formelzeichen für die Elektrotechnik – Teil 1: Allgemeines (IEC 60027-1:1995 (Reprint) + A1:1997 + A2:2005); Deutsche Fassung EN 60027-1:2006 + A2:2007*. Sep. 2007.
- [285] F. Asseg. *exp4j*. 2013. URL: <http://www.objecthunter.net/exp4j/> (besucht am 29.04.2014).
- [286] S. Hennig und A. Braune. „On the development of sustainable visualization solutions in industrial automation through Domain Specific Languages“. In: *Industrial Technology (ICIT), 2012 IEEE International Conference on*. März 2012, S. 72–77. DOI: 10.1109/ICIT.2012.6209916.
- [287] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 100: Nutzung von Merkmalleisten im PLT-Engineering-Workflow*. 30. Sep. 2010.
- [288] EDDL Cooperation Team, Hrsg. *Field Device Integration Project – EDDL Encoded File Format*. 1. Sep. 2010.
- [289] EDDL Cooperation Team, Hrsg. *Field Device Integration Project – Technical Specification – Profile PROFIBUS*. 8. Feb. 2012.

- [290] PROFIBUS International, Hrsg. *PROFIBUS Specification: Specification for PROFIBUS Device Description and Device Integration - Volume 2: EDDL*. Dez. 2005.
- [291] International Organization for Standardization, Hrsg. *ISO 15745-1: Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*. 1. März 2003.
- [292] J. Müller. „Automatisierung des Engineerings kommunikationsfähiger Anlagenkomponenten im Anlagennahen Assetmanagement in der Prozessindustrie“. Diss. Fakultät Georesourcen und Materialtechnik, RWTH Aachen, Okt. 2007.
- [293] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 62541-4: OPC Unified Architecture – Teil 4: Dienste (IEC 62541-4:2011); Englische Fassung EN 62541-4:2011*. Juni 2012.
- [294] M. Graube, J. Pfeffer, J. Ziegler und L. Urbas. „Linked Data as Integrating Technology for Industrial Data“. In: *Network-Based Information Systems (NBiS), 2011 14th International Conference on*. Sep. 2011, S. 162–167. DOI: 10.1109/NBiS.2011.33.
- [295] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61346-1: Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte; Strukturierungsprinzipien und Referenzkennzeichnung – Teil 1 Allgemeine Regeln (IED 1346-1:1996) – Deutsche Fassung EN 61346-1:1996*. Jan. 1997.
- [296] Deutsches Institut für Normung e. V., Hrsg. *DIN EN 61346-2: Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte; Strukturierungsprinzipien und Referenzkennzeichnung – Teil 2: Klassifizierung von Objekten und Kodierung von Klassen (IEC 61346-2:2000) Deutsche Fassung EN 61346-2:2000*. Dez. 2000.
- [297] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Hrsg. *NAMUR-Empfehlung 131: Anforderungen an Feldgeräte für Standardanwendungen*. Apr. 2009.
- [298] U. Epple, Hrsg. *CAEX – IEC 62424, ACPLT/OV-Bibliothek\_V2.15.1, Pumpwerk*. 12. Nov. 2009. URL: <http://www.plt.rwth-aachen.de/cms/PLT/Forschung/Projekte2/~ejwy/CAEX-IEC-62424/> (besucht am 28.04.2015).
- [299] T. Schmidt. „Sehen – Wissen – Handeln: Symbole erleichtern Plant Asset Management“. In: *PROCESS* (03 März 2012).
- [300] H. Nack und W. Setzwein. „Benchmark in der Automatisierungstechnik – Ein Erfahrungsbericht“. In: *atp – Automatisierungstechnische Praxis* (2 Feb. 2008), S. 82–85.
- [301] A. Strey. *Computer-Arithmetik – Vorlesungsfolien SS 2005, Kapitel 6: Intervall-Arithmetik*. 21. Juni 2005. URL: <http://www.informatik.uni-ulm.de/ni/Lehre/SS05/CompArith/ArithInterval2.pdf> (besucht am 24.11.2015).
- [302] OPC Foundation, Hrsg. *OPC Unified Architecture Specification Part 11: Historical Access*. Jan. 2007.
- [303] PROFIBUS International, Hrsg. *PROFIBUS Specification: Test Specification for PA Devices*. V 4.1. Nov. 2006.

- [304] S. Mätzler, R. Lehmann, S. Theurich, A. Dennert und M. Wollschlaeger. „Virtual industrial Ethernet devices for testing purposes“. In: *Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on*. Sep. 2012, S. 1–4. DOI: 10.1109/ETFA.2012.6489698.
- [305] MESA International, Hrsg. *B2MML V0600 – Type: Standards and Tools*. Mai 2013. URL: <https://services.mesa.org/ResourceLibrary/ShowResource/0f47758b-60f0-40c6-a71b-fa7b2363fb3a> (besucht am 03.12.2015).
- [306] OPC Foundation, Hrsg. *OPC Unified Architecture for ISA-95 Common Object Model – Companion Specification*. Release 1.00. 30. Okt. 2013.
- [307] MESA International, Hrsg. *Key Performance Indicator Markup Language (KPI-ML)*. 2015. URL: <http://www.mesa.org/en/KPIML.asp> (besucht am 03.12.2015).
- [308] N. Suchold, M. Mühlhause und C. Diedrich, Hrsg. *Automatisierungstechnisches Glossar 2.0*. Version Revision: 110. 2015. URL: <http://ai.ifak.eu/glossar2/> (besucht am 09.12.2015).





# A Veröffentlichte Vorarbeiten

In diesem Anhang sind für die Darstellungen wichtige, aber nicht erneut wiedergegebene Texte aus eigenen, auf Konferenzen veröffentlichten Vorarbeiten *wortgleich* abgedruckt. Nachträgliche Anmerkungen oder Korrekturen sind mit einem [...] gekennzeichnet.

## A.1 Prüfbare Konstrukte in CAEX-Dateien

Die Veröffentlichung [3] entstand mit den Co-Autoren Andreas Schüller, Martin Wollschlaeger und Ulrich Epple. Abschnitt 3 ist im Folgenden abgedruckt. Die Literaturstelle „[1]“ entspricht [17], „[6]“ entspricht [49].

### [3]: 3. Prüfbare Konstrukte

Eine CAEX-Datei kann mit generischen XML-Tools einer Basisprüfung unterzogen werden. Es kann festgestellt werden, ob eine Datei wohlgeformt ist. Ist dies nicht der Fall, sind alle weiteren Prüfungen unnötig. Als nächstes wird die Datei gegen das CAEX-Metamodell (XSD in [1]) validiert, welches als XML-Schema notiert ist. Diese Syntaxprüfungen können mit Standard-XML-Werkzeugen durchgeführt werden und sollten bei der Verarbeitung von XML vorausgesetzt werden können.

Alle weiteren Prüfungen basieren auf der Semantik der Konstrukte und sind mithilfe von Standardwerkzeugen nicht mehr zu bewerkstelligen. Hier ist ein Prüfwerkzeug erforderlich, welches die nicht-formalen, normativen Vorgaben aus dem CAEX-Standard überprüft. In den folgenden Abschnitten werden die von einem inhaltsunabhängigen Werkzeug überprüfbaren Konstrukte beschrieben.

### [3]: 3.1. Namen und UUIDS

Es können die Einzigartigkeit von Namen und UUIDs in der Datei und in eingebundenen Bibliotheken dateiübergreifend getestet werden. Dies betrifft das vorgeschriebene XML-Attribut „Name“ und das optionale XML-Attribut „ID“ im ComplexType „CAEXObject“. Dabei sind folgende Entscheidungen zu treffen:

1. Dürfen Namen in unterschiedlichen Hierarchieebenen wiederverwendet werden?
2. Dürfen Namen in unterschiedlichen Teilbäumen wiederverwendet werden?
3. Dürfen Namen in unterschiedlichen Dateien wiederverwendet werden?

Es gibt keine Festlegung, die erfordert, dass der Name eines Elementes über die Datei einzigartig sein muss. Dennoch ist dies meist eine Vorgabe aus dem Engineering, dass Elemente eindeutig zu identifizieren sind. Demzufolge ist es sinnvoll, auch in Engineeringdaten-Exporten nach Dopplungen zu suchen. Da Namen von Attributen und Schnittstellen

durch Rollenklassen vorgegeben werden können, können diese nicht auf Einzigartigkeit getestet werden. Die vorliegende Implementierung prüft nur auf eindeutige Objektpfade. Namen können wiederverwendet werden, solange der absolute Pfadname in der Hierarchie einer Datei eindeutig ist (dies ist die Minimalanforderung).

Die UUID eines Elements dient der Wiedererkennung nach Umbenennung. Sie wird vom AutomationML-Editor [6] in allen Objekten in der Hierarchie (ohne Schnittstellen und Attributen) gefordert. Hier muss immer über alle Dateien nach Duplikaten gesucht werden. Selbst in unterschiedlichen Dateien (eines zusammengehörenden Arbeitsstandes) sind doppelte UUIDs auf jeden Fall ein Fehler. Es besteht auch die Möglichkeit nach bestimmten Vorgaben UUIDs zu erzeugen, wenn sie vom Anwender gefordert werden [6].

### **[3]: 3.2. Referenzen**

In verschiedenen CAEXObject-Typen sind Referenzen erlaubt. Diese können Referenzen auf Oberklassen, realisierte oder unterstützte Rollenklassen sein. Diese Referenzen sind absolute Pfadangaben. Weiterhin sind in Instanzhierarchien relative Verweise in die unterlagerte Hierarchie (InternalLink) möglich. Hier können die Existenz eines Objektes mit dem Namen, die korrekte Maskierung von Sonderzeichen, und der richtige Typ des Zielobjektes geprüft werden. Daran anschließende Prüfungen werden in Abschnitt 3.5 und 3.6 behandelt.

### **[3]: 3.3. Attributwerte / Constraints**

In allen CAEXObject-Typen können Attribute angegeben werden. Diese können Datentyp und Maßeinheit definieren. Zusätzlich können Constraints den Raum der möglichen belegbaren Werte einschränken. Die Prüfung der Einhaltung eines direkt neben einem angegebenen Wert (Value) definierten Constraint mutet zunächst trivial und in der Notation redundant an. Die Stärke dieses Konstrukts zeigt sich dann, wenn die Vererbungen und Verlinkungen aus Abschnitt 3.2 zum Einsatz kommen. Dann können viele allgemein gültige Beschränkungen einmal definiert und häufig angewendet werden (siehe Abschnitte 3.5 und 3.6).

### **[3]: 3.4 Vollständigkeit der Definition von Objekten**

In Schnittstellenklassen und Rollenklassen werden Sätze von Attributen definiert und Randbedingungen für ihre Werte festgelegt. Hier ist aber nicht notwendigerweise ein Wert für das Attribut festzulegen. Datentyp und Maßeinheit können jedoch vorgegeben werden. Hier stellen sich drei grundsätzliche Fragen:

1. Ist es ein Fehler, wenn keine Maßeinheit angegeben wird? Wenn Bibliotheken sehr generisch sind, kann sie unter Umständen diese Aussage für bestimmte Attribute noch nicht treffen, was eine Fehlerausgabe zur Folge hätte.
2. Dürfen Datentypen oder Maßeinheiten überschrieben werden? Das Überschreiben mit kompatiblen Maßeinheiten scheint akzeptabel, könnte aber die Fehlerquote bei der Verwendung erhöhen. Datentypen sollten generell nicht geändert werden.

3. Muss festgelegt werden, ob ein Default-Wert als gesetzter Wert eines Attributs zählt oder nur als Hinweis auf eine übliche Ausprägung dient?

In der beschriebenen Umsetzung werden `AttributeDataType` und `Unit` in allen Attributen angemahnt (siehe XML-Beispiel in Abschnitt 2.3). Wenn sie fehlen, wird eine Warnung ausgegeben. Der Attributwert muss nur in konkreten Instanzen (in `SystemUnitClass` und `InternalElements`) angegeben sein. Ein Default-Wert wird nicht als gesetzter Wert anerkannt.

### [3]: 3.5 Vererbung und Rollenumsetzung

Sowohl die Vererbung in den Interface-, Role- und `SystemUnitClasses` (Family-Typen) als auch die Anwendung von Rollen (`RoleRequirements` und `SupportedRoleClasses`) können bei der Prüfung der Konstrukte ähnlich behandelt werden. Es werden in jedem Fall Definitionen miteinander kombiniert. Gleichnamige Definitionen werden als selbiges (nicht gleiches) Objekt angenommen und aus den Unterdefinitionen kombiniert (überschrieben / overridden im Sprachgebrauch[ $\emptyset$ ]\* der Objektorientierung). Bei den Rollen müssen davor noch die Namensmappings beachtet werden. Diese Generalisierung spart erheblichen Implementierungsaufwand, hat aber auch den Nachteil, dass alle Fehler auf Schnittstellen- und Attributebene gemeldet werden. Der Kontext, ob der Fehler durch eine Vererbung oder beim Erfüllen einer Rollenabhängigkeit entstanden ist, geht verloren.

Während dieser Kombination können eine große Menge von Modellinkonsistenzen gefunden werden. Attribut-Datentypen oder Maßeinheiten könnten geändert werden; Werte von Attributen könnten mit den Constraints in ihrer Basisdefinition inkompatibel sein oder ganz fehlen. Sich ergänzende Constraints könnten sich widersprechen, beispielsweise wenn Constraints in Spezialisierungen einen größeren Wertebereich oder andere Werte erlauben als die Constraints in der zugehörigen Basisdefinition.

Im Beispiel aus Abschnitt 2.3 würde das Attribut „Attr“ im `ExternalInterface` „ExtIF“ der Rolle „Rolle“ aus allen in der Vererbungshierarchie des verwendeten Schnittstellentyps existierenden Vorkommen zusammengestellt. Es besitzt demzufolge einen Datentyp und eine Maßeinheit aus der Basisdefinition, ein Constraint aus deren Überdefinition und einen Wert aus dem `ExternalInterface`. Die Definition ist somit vollständig.

Diese Sammlung von Eigenschaften aus unterschiedlichen Definitionen wird immer durchgeführt wenn:

1. Schnittstellen-, Rollen- oder `SystemUnit`-Klassen Attribute erben
2. `InternalElements` von `SystemUnit`-Klassen realisiert werden
3. Rollen- oder `SystemUnit`-Klassen `ExternalInterfaces` erben
4. `SystemUnit`-Klassen oder `InternalElements` [ändern Namen (`AttributeNameMapping`, `InterfaceNameMapping`)]\*
5. `SystemUnit`-Klassen Rollenklassen unterstützen (`SupportedRoleClass`)
6. `InternalElements` Rollenklassen fordern (`RoleRequirements`)

### [3]: 3.6 InternalLink

InternalLinks verbinden Teile von Instanzhierarchien oder von SystemUnit-Klassen miteinander. Dies geschieht mittels der ExternalInterfaces der InternalElements (in PandIX „HLINK“). Ebenfalls können sie InternalElements mit den Schnittstellen des umgebenden Elements verbinden (in PandIX „VLINK“).

Die Definition der ExternalInterfaces ist, wie in Abschnitt 3.5 beschrieben, aus Basisdefinitionen und Erweiterungen zusammengestellt. Die Attribute werden auf Kompatibilität der Definitionen, Gleichheit der Werte und gegenseitig auf Vollständigkeit und Kompatibilität der Constraints überprüft. Beispielsweise muss bei ProductConnectionPoints (Stoffverbindungen) der Nenndurchmesser (Attribut „DN“) auf beiden Seiten identisch sein. Zusätzlich können die geerbten Typinformationen der ExternalInterfaces verglichen werden. Man kann fordern, dass sie von derselben Schnittstellenklasse abgeleitet sind, oder, bei weniger strikter Prüfung, sie zumindest eine gemeinsame Basisschnittstelle in der Vererbungshierarchie haben müssen.

## A.2 Benutzungsoberfläche des Package-Werkzeugs

Die Veröffentlichung entstand mit dem Co-Autor Martin Wollschlaeger. Referenzen auf Abbildungen wurden nicht adaptiert und Verweisen auf die Bilder im Text. Der referenzierte Abschnitt 2 beschreibt die Werkzeugkette, siehe Abschnitt 3.3 ab Seite 42. Die Literaturstelle „[IEC62424]“ entspricht [17].

### [2]: 4 Das Package-Designer-Werkzeug

Das Package-Designer-Werkzeug ist ein Prototyp, der zum Erzeugen von beispielhaften Apparatebeschreibungsdateien entwickelt wurde. Einige Konzepte, die zur Vereinfachung der Erstellung sinnvoll sind, sind darin dennoch umgesetzt und werden an passen der Stelle erläutert. Wie in Abschnitt 2 erläutert, müssen, um der Apparatedefinition eine feste Semantik zuordnen zu können, standardisierte CAEX-Bibliotheken verwendet werden. Die dort erwähnten PandIX-Bibliotheken sind derzeit die einzigen im Tool verfügbaren, es können aber auf einfache Weise weitere Bibliotheken integriert werden, sodass weitere Rollenklassen und Schnittstellenklassen in den Menüs verfügbar sind.

Im Werkzeug wird das Modell eines Apparates mit dessen Prozessanschlüssen modelliert (Abb. 4). Zu allen Anschlüssen (am Anschluss und dem Apparat selbst (darunter) können beliebig viele Parameter und Funktionen definiert werden, die das Modell des Apparates beschreiben. Zu allen Modellelementen müssen bestimmte Informationen angegeben werden, zusätzlich können beliebige Eigenschaften hinzu definiert werden.

Die Darstellung des Apparates (Abb. 5) lehnt sich an den Darstellungen in R&I Fließbildern an. Die Messstellen sitzen jedoch auf der Achse der Leitung. Die Bezeichnung der Messstelle folgt keinem Anlagen-Benennungsschema, sondern sollte anhand der Rolle oder Funktion der Messstelle für die zu implementierenden Berechnungsmodelle gewählt werden. Für die Berechnungen benötigte Messstellen sind als R&I-Oval dargestellt. Parameter sind in einem Rechteck mit einem „P“ dargestellt, Berechnungsfunktionen mit einem Achteck mit einem „f“.

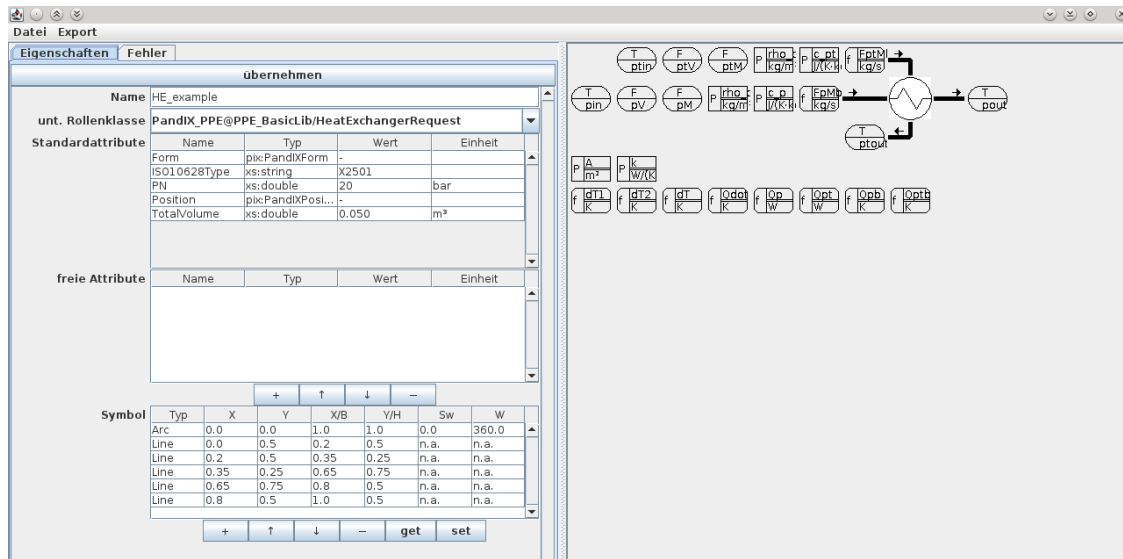


Abbildung A.1: [2] Abb. 4: Gesamtansicht Package-Designer-Werkzeug

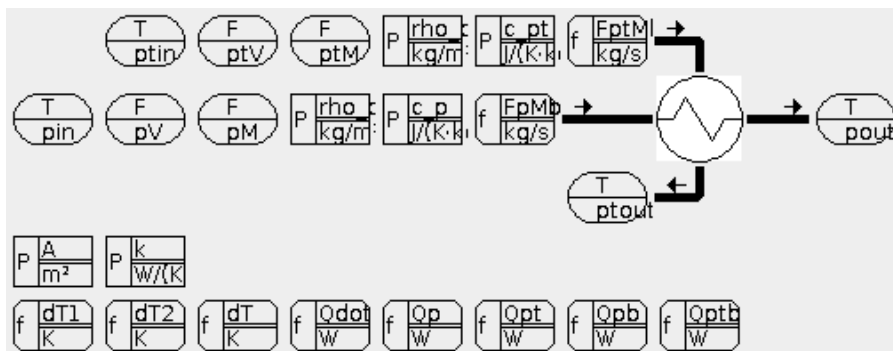


Abbildung A.2: [2] Abb. 5: Darstellung des Apparates und der Modellanforderungen

Für den Apparat kann eine unterstützte Rollenklasse (supported role class) festgelegt werden (Abb. 6). Diese meist standardisierte Rollenklasse definiert eine Menge von Schnittstellen und Attributen, die sie von der konkreten Umsetzung (hier den Apparat) fordert. Die Werte können entsprechend eingegeben werden. Weiterhin können hersteller- oder umsetzungsspezifische Attribute definiert werden.

Ein Prozessanschluss des Apparates (Abb. 7) kann entweder eine durch die Rollenklasse vorgegebene Schnittstelle sein oder er kann frei definiert werden. In ersterem Fall sind Name und Schnittstellenklasse (interface class) von der Rollenklasse des Apparates vorgegeben und – wie im Beispiel – nicht änderbar. Auch hier sind Werte für Standardattribute anzugeben und es können zusätzlich freie Attribute definiert werden. Die Hauptflussrichtung der Anschlüsse wird hier festgelegt.

Messstellenanforderungen (Abb. 8) können nur an Prozessanschlüssen platziert werden. Sensoren am Apparat (z. B. Behältern) kommen an einen dedizierten Messgeräteanschluss. Nach der Auswahl der PCE-Kategorie (Tabelle 2 aus [IEC62424]) kann die physikalische Größe, die die Messstelle liefern soll, festgelegt werden. Dahinter ist die *SI*-Standard-Maßeinheit der Größe hinterlegt. Geräte mancher PCE-Kategorien erfordern weitere

The screenshot shows a dialog box titled 'übernehmen' with two tabs: 'Eigenschaften' and 'Fehler'. The 'Eigenschaften' tab is active. The 'Name' field contains 'HE\_example'. Below it, 'unt. Rollenklasse' is set to 'PandIX\_PPE@PPE\_BasicLib/HeatExchangerRequest'. Under 'Standardattribute', there is a table with the following data:

Name	Typ	Wert	Einheit
Form	pik:PandIXForm	-	
ISO10628Type	xs:string	X2501	
PN	xs:double	20	bar
Position	pik:PandIXPosi...	-	
TotalVolume	xs:double	0.050	m³

Below this table is an empty table for 'freie Attribute' with the same headers. At the bottom, there are four buttons: '+', '↑', '↓', and '-'.

Abbildung A.3: [2] Abb. 6: Einstellungen für den Apparat

The screenshot shows a dialog box titled 'übernehmen' with two tabs: 'Eigenschaften' and 'Fehler'. The 'Eigenschaften' tab is active. The 'Name' field contains 'Pln'. Below it, 'realisierte Rollenklassenschnittstelle' is set to 'Pln' and 'Schnittstellenklasse' is set to 'PandIX\_ICL@PandIX\_ICL/ProductConnectionPoint'. Under 'Standardattribute', there is a table with the following data:

Name	Typ	Wert	Einheit
DN	xs:double	30	mm
Position	xs:string	-	

Below this table is an empty table for 'freie Attribute' with the same headers. At the bottom, there is a 'Flussrichtung' dropdown menu set to 'nur Eingang'. At the very bottom, there are four buttons: '+', '↑', '↓', and '-'.

Abbildung A.4: [2] Abb. 7: Einstellungen für einen Prozessanschluss am Apparat

Angaben, um die Messgröße am Gerät in die tatsächliche Größe am Apparateanschluss umrechnen zu können. Dies ist z. B. für einen Drucksensor offensichtlich. Hier müssten als Parameter Höhendifferenz und Dichte des Mediums vom Anwender angegeben werden und der Messwert in einer Funktion berechnet werden.

Parameter (Abb. 9) können sowohl für Anschlüsse als auch für den ganzen Apparat angegeben werden. Auch hier ist die Angabe der Maßeinheit erforderlich. Wie in Abb. 9 und Abb. 10 ersichtlich wurde die Maßeinheit zur Vereinfachung implizit über die physikalische Größe angegeben.

Für eine Berechnungsfunktion (Abb. 10) wird die Maßeinheit des Rückgabewertes und die Berechnungsvorschrift angegeben. Variablennamen (voll qualifizierte Namen der Apparatmodellelemente) und EDD-Funktionen (Built-Ins) können mittels einer Liste, die anhand begonnener Eingaben gefiltert wird, vervollständigt werden. Alle Eingaben werden fortwährend auf Korrektheit und Konsistenz geprüft und Fehler entsprechend

The screenshot shows a dialog box titled 'übernehmen' with two tabs: 'Eigenschaften' and 'Fehler'. The 'Eigenschaften' tab is active. The 'Name' field contains 'FpV'. Below it, 'PCE Kategorie' is set to 'F (Fluss)' and 'Physikalische Größe' is set to 'Volumenstrom'. At the bottom, there are four buttons: '+', '↑', '↓', and '-'.

Abbildung A.5: [2] Abb. 8: Einstellungen für eine benötigte Messstelle



Abbildung A.6: [2] Abb. 9: Einstellungen für einen benötigten Parameter

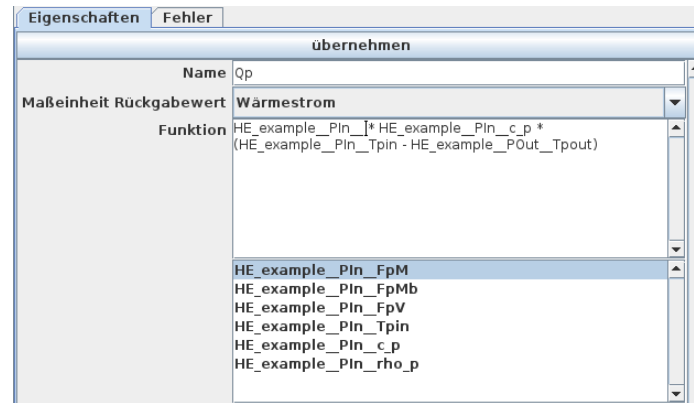


Abbildung A.7: [2] Abb. 10: Einstellungen für eine zu berechnende Funktion

angezeigt. Zeitreihen können in diesem Prototyp nicht zugegriffen werden, sind aber in der Datenquelle (FDI Server) verfügbar. Für komplexere Modelle können diese bei entsprechender Werkzeugunterstützung ebenfalls verwendet werden.





# B Quelltexte

## B.1 Dia2CAEX

### Quelltext B.1: Dia2CAEX XSLT-Template für einen Wärmeübertrager

```
<xsl:template match="dia:object[@type='chemeng - hx' or @type='chemeng - hxv']"
  mode="intelem">
  <!-- Simple Heat Exchanger -->
  <InternalElement>
    <xsl:attribute name="RefBaseSystemUnitPath">
      <xsl:text>stth_SUC@PPE_SystemUnitLib2/TubeShellHeatExchanger</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="Name">
      <xsl:apply-templates select="." mode="intelemname" />
    </xsl:attribute>
    <xsl:attribute name="ID">
      <xsl:value-of select="uuid:randomUUID()" />
    </xsl:attribute>
    <RoleRequirements
      RefBaseRoleClassPath="PandIX_PPE@PPE_BasicLib/HeatExchangerRequest" />
  </InternalElement>
</xsl:template>

<xsl:template match="dia:object[@type='chemeng - hx' or @type='chemeng - hxv']"
  mode="intelemname">
  <!-- Simple Heat Exchanger -->
  <xsl:text>HE</xsl:text>
  <xsl:value-of select="@id" />
</xsl:template>

<xsl:template match="dia:object[@type='chemeng - hx']" mode="extifname">
  <!-- Simple Heat Exchanger -->
  <xsl:param name="connection" />
  <xsl:choose>
    <!-- TO-DO which is which -->
    <xsl:when test="$connection='0'">
      <xsl:text>PIn</xsl:text>
    </xsl:when>
    <xsl:when test="$connection='2'">
      <xsl:text>POut</xsl:text>
    </xsl:when>
    <xsl:when test="$connection='3'">
      <xsl:text>PTempIn</xsl:text>
    </xsl:when>
    <xsl:when test="$connection='1'">
      <xsl:text>PTempOut</xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select="." mode="unsupported_connector">
        <xsl:with-param name="connection" select="$connection" />
      </xsl:apply-templates>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

```
<xsl:template match="dia:object[@type='chemeng - hxv']" mode="extifname">
  <!-- Simple Heat Exchanger -->
  <xsl:param name="connection" />
  <xsl:choose>
    <!-- TO-DO which is which -->
    <xsl:when test="$connection='4'">
      <xsl:text>PIn</xsl:text>
    </xsl:when>
    <xsl:when test="$connection='4'">
      <xsl:text>POut</xsl:text>
    </xsl:when>
    <xsl:when test="$connection='4'">
      <xsl:text>PTempIn</xsl:text>
    </xsl:when>
    <xsl:when test="$connection='4'">
      <xsl:text>PTempOut</xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select="." mode="unsupported_connector">
        <xsl:with-param name="connection" select="$connection" />
      </xsl:apply-templates>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

### Quelltext B.2: Dia2CAEX Eingabe für einen Wärmeübertrager

```
<dia:object type="chemeng - hx" version="1" id="0107">
  <dia:attribute name="obj_pos">
    <dia:point val="82,42"/>
  </dia:attribute>
  <dia:attribute name="obj_bb">
    <dia:rectangle val="81.95,41.95;84.05,44.05"/>
  </dia:attribute>
  <dia:attribute name="meta">
    <dia:composite type="dict"/>
  </dia:attribute>
  <dia:attribute name="elem_corner">
    <dia:point val="82,42"/>
  </dia:attribute>
  <dia:attribute name="elem_width">
    <dia:real val="2"/>
  </dia:attribute>
  <dia:attribute name="elem_height">
    <dia:real val="2"/>
  </dia:attribute>
  <dia:attribute name="line_width">
    <dia:real val="0.10000000000000001"/>
  </dia:attribute>
  <dia:attribute name="line_colour">
    <dia:color val="#000000"/>
  </dia:attribute>
  <dia:attribute name="fill_colour">
    <dia:color val="#ffffff"/>
  </dia:attribute>
  <dia:attribute name="show_background">
    <dia:boolean val="true"/>
  </dia:attribute>
  <dia:attribute name="line_style">
    <dia:enum val="0"/>
    <dia:real val="1"/>
  </dia:attribute>
  <dia:attribute name="flip_horizontal">
    <dia:boolean val="false"/>
  </dia:attribute>
```

```

<dia:attribute name="flip_vertical">
  <dia:boolean val="false"/>
</dia:attribute>
<dia:attribute name="subscale">
  <dia:real val="1"/>
</dia:attribute>
</dia:object>

```

### Quelltext B.3: Dia2CAEX Ausgabe für einen Wärmeübertrager

```

<InternalElement xmlns:fn="http://www.w3.org/2005/xpath-functions"
  xmlns:dia="http://www.lysator.liu.se/~alla/dia/"
  RefBaseSystemUnitPath="stth_SUC@PPE_SystemUnitLib2/TubeShellHeatExchanger"
  Name="HE0107"
  ID="96e04ec7-3a0e-4e76-b78e-08e5bc39622c">
  <RoleRequirements
    RefBaseRoleClassPath="PandIX_PPE@PPE_BasicLib/HeatExchangerRequest"/>
</InternalElement>

```

### Quelltext B.4: Dia2CAEX Eingabe für eine Messstelle

```

<dia:object type="IEC62424 - measurement" version="1" id="021">
  <dia:attribute name="obj_pos">
    <dia:point val="-2,10"/>
  </dia:attribute>
  <dia:attribute name="obj_bb">
    <dia:rectangle val="-2.005,9.995;3.0005,14"/>
  </dia:attribute>
  <dia:attribute name="meta">
    <dia:composite type="dict"/>
  </dia:attribute>
  <dia:attribute name="elem_corner">
    <dia:point val="-2,10"/>
  </dia:attribute>
  <dia:attribute name="elem_width">
    <dia:real val="5"/>
  </dia:attribute>
  <dia:attribute name="elem_height">
    <dia:real val="4"/>
  </dia:attribute>
  <dia:attribute name="line_width">
    <dia:real val="0.10000000000000001"/>
  </dia:attribute>
  <dia:attribute name="line_colour">
    <dia:color val="#000000"/>
  </dia:attribute>
  <dia:attribute name="fill_colour">
    <dia:color val="#ffffff"/>
  </dia:attribute>
  <dia:attribute name="show_background">
    <dia:boolean val="true"/>
  </dia:attribute>
  <dia:attribute name="line_style">
    <dia:enum val="0"/>
    <dia:real val="1"/>
  </dia:attribute>
  <dia:attribute name="padding">
    <dia:real val="0.10000000000000001"/>
  </dia:attribute>
  <dia:attribute name="text">
    <dia:composite type="text">
      <dia:attribute name="string">
        <dia:string>#FIC
1.1#</dia:string>

```

```

</dia:attribute>
<dia:attribute name="font">
  <dia:font family="sans" style="0" name="Helvetica"/>
</dia:attribute>
<dia:attribute name="height">
  <dia:real val="0.80000000000000004"/>
</dia:attribute>
<dia:attribute name="pos">
  <dia:point val="0,10.8"/>
</dia:attribute>
<dia:attribute name="color">
  <dia:color val="#000000"/>
</dia:attribute>
<dia:attribute name="alignment">
  <dia:enum val="1"/>
</dia:attribute>
</dia:composite>
</dia:attribute>
<dia:attribute name="flip_horizontal">
  <dia:boolean val="false"/>
</dia:attribute>
<dia:attribute name="flip_vertical">
  <dia:boolean val="false"/>
</dia:attribute>
<dia:attribute name="subscale">
  <dia:real val="1"/>
</dia:attribute>
</dia:object>

```

**Quelltext B.5:** Dia2CAEX Ausgabe für eine Messstelle

```

<InternalElement Name="FIC1.1" ID="b3f8c1d0-755b-4271-9bf2-5cbf1103115b">
  <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">
    <Value>...</Value>
  </Attribute>
  <Attribute Name="ISO10628Type" Unit="-" AttributeDataType="xs:string">
    <Value>C2054</Value>
  </Attribute>
  <Attribute Name="RemoteType" Unit="-" AttributeDataType="xs:string">
    <Value>Central</Value>
  </Attribute>
  <Attribute Name="SignalCode" Unit="-" AttributeDataType="xs:string">
    <Value>I</Value>
  </Attribute>
  <Attribute Name="FunctionCode" Unit="-" AttributeDataType="xs:string">
    <Value>F</Value>
  </Attribute>
  <Attribute Name="ReferenceCode" Unit="-" AttributeDataType="xs:string">
    <Value>IC</Value>
  </Attribute>
  <ExternalInterface Name="P"
    RefBaseClassPath="PandIX_ICL@PandIX_ICL/SensorProcessInterface">
    <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">
      <Value>...</Value>
    </Attribute>
  </ExternalInterface>
  <ExternalInterface Name="Out"
    RefBaseClassPath="PandIX_ICL@PandIX_ICL/SignalSource">
    <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">
      <Value>...</Value>
    </Attribute>
  </ExternalInterface>
  <ExternalInterface Name="L"
    RefBaseClassPath="PandIX_ICL@PandIX_ICL/SignalSource">
    <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">

```

```

    <Value>...</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="H"
  RefBaseClassPath="PandIX_ICL@PandIX_ICL/SignalSource">
  <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">
    <Value>...</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="LL"
  RefBaseClassPath="PandIX_ICL@PandIX_ICL/SignalSource">
  <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">
    <Value>...</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="HH"
  RefBaseClassPath="PandIX_ICL@PandIX_ICL/SignalSource">
  <Attribute Name="Position" Unit="-" AttributeDataType="PandIX-Position">
    <Value>...</Value>
  </Attribute>
</ExternalInterface>
<RoleRequirements RefBaseRoleClassPath="PandIX_PCE@PandIX_PCE/SensorRequest"/>
</InternalElement>

```

## B.2 SystemUnitClass für einen Apparat

Quelltext B.6 zeigt die vollständige Datei aus Quelltext 4.2 auf Seite 68.

**Quelltext B.6:** Beispiel für eine exportierte SystemUnitClass (neuere Version des Bespiels aus [2])

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<CAEXFile FileName="SystemUnitLib_HE_example.xml" SchemaVersion="2.15"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd">
  <Version>passend zu PandIX V6.00 + stth Erweiterung</Version>
  <Copyright>YOUR COPYRIGHT</Copyright>
  <AdditionalInformation/>
  <Version>passend zu PandIX V6.00 + stth Erweiterung</Version>
  <Copyright>TODO: YOUR COPYRIGHT</Copyright>
  <AdditionalInformation/>
  <ExternalReference Alias="PandIX_ICL" Path="PandIX_ICL_V600.xml"/>
  <ExternalReference Alias="PandIX_PCE" Path="PandIX_PCE_V600.xml"/>
  <ExternalReference Alias="PandIX_PPE" Path="PandIX_PPE-BasicLib_V600.xml"/>
  <ExternalReference Alias="stth_PPE" Path="stth_PPE-BasicLib_V600.xml"/>
  <SystemUnitClassLib ID="2111c4a1-b2f4-41f3-a231-c1bd7679e4c1"
    Name="SystemUnitLib_HE_example">
    <SystemUnitClass ID="3606689b-2e2e-4a20-b0d0-b9f4b2af780a"
      Name="HE_example">
      <Attribute Name="Form" AttributeDataType="pix:PandIXForm" >
        <Value>-</Value>
      </Attribute>
      <Attribute Name="ISO10628Type" AttributeDataType="xs:string" >
        <Value>X2501</Value>
      </Attribute>
      <Attribute Name="PN" AttributeDataType="xs:double" Unit="bar">
        <Value>20</Value>
      </Attribute>
      <Attribute Name="Position" AttributeDataType="pix:PandIXPosition" >
        <Value>-</Value>
      </Attribute>
      <Attribute Name="TotalVolume" AttributeDataType="xs:double" Unit="m^3">
        <Value>0.050</Value>
      </Attribute>
      <ExternalInterface Name="PTempIn"

```

```

    RefBaseClassPath="PandIX_ICL@PandIX_ICL/ProductConnectionPoint">
  <Attribute Name="Position" AttributeDataType="xs:string" >
    <Value>-</Value>
  </Attribute>
  <Attribute Name="DN" AttributeDataType="xs:double" Unit="mm">
    <Value>32</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="PTempOut"
  RefBaseClassPath="PandIX_ICL@PandIX_ICL/ProductConnectionPoint">
  <Attribute Name="DN" AttributeDataType="xs:double" Unit="mm">
    <Value>32</Value>
  </Attribute>
  <Attribute Name="Position" AttributeDataType="xs:string" >
    <Value>-</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="PIn"
  RefBaseClassPath="PandIX_ICL@PandIX_ICL/ProductConnectionPoint">
  <Attribute Name="DN" AttributeDataType="xs:double" Unit="mm">
    <Value>32</Value>
  </Attribute>
  <Attribute Name="Position" AttributeDataType="xs:string" >
    <Value>-</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="POut"
  RefBaseClassPath="PandIX_ICL@PandIX_ICL/ProductConnectionPoint">
  <Attribute Name="DN" AttributeDataType="xs:double" Unit="mm">
    <Value>32</Value>
  </Attribute>
  <Attribute Name="Position" AttributeDataType="xs:string" >
    <Value>-</Value>
  </Attribute>
</ExternalInterface>
<SupportedRoleClass
  RefRoleClassPath="PandIX_PPE@PPE_BasicLib/HeatExchangerRequest"/>
</SystemUnitClass>
</SystemUnitClassLib>
</CAEXFile>

```

## B.3 Schema für die Konfigurationsdatei des Equipment Instance Handlers

Quelltext B.7: Konfigurationsschema für den Equipment Instance Handler

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.inf.tu-dresden.de/pk/stth/eih-config"
  xmlns:tns="http://www.inf.tu-dresden.de/pk/stth/eih-config"
  elementFormDefault="qualified">

  <xs:complexType name="Variable">
    <xs:sequence>
      <xs:element name="Slot" type="xs:byte" minOccurs="1" maxOccurs="1" />
      <xs:element name="Index" type="xs:byte" minOccurs="1" maxOccurs="1" />
      <xs:element name="DeviceBrowsePath" type="xs:string" minOccurs="1"
        maxOccurs="1" />
      <xs:element name="ValueBrowsePath" type="xs:string" minOccurs="1"

```

### B.3 Schema für die Konfigurationsdatei des Equipment Instance Handlers

```
        maxOccurs="1" />
<xs:element name="StatusBrowsePath" type="xs:string" minOccurs="1"
  maxOccurs="1" />
<xs:choice minOccurs="1" maxOccurs="1">
  <xs:element name="ReadUnitBrowsePath" type="xs:string" minOccurs="1"
    maxOccurs="1" />
  <xs:element name="ReadUnitName" type="xs:string" minOccurs="1"
    maxOccurs="1" />
</xs:choice>
<xs:element name="ProvideUnitName" type="xs:string" minOccurs="1"
  maxOccurs="1" />
</xs:sequence>
<xs:attribute name="Name" type="xs:string" />
</xs:complexType>

<simpleType name="DDEntry">
  <restriction base="hexBinary">
    <!-- first is 6 uint16_t -->
    <minLength value="24"/>
    <!-- all other are 4 times (2 uint8_t + uint16_t) -->
    <maxLength value="32"/>
  </restriction>
</simpleType>

<xs:complexType name="VApparat">
  <xs:sequence>
    <xs:element name="DeviceDirectory" minOccurs="1" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="DateiTypNodeset" type="xs:string" minOccurs="1"
            maxOccurs="14" />
          <xs:element name="DateiInstanzNodeset" type="xs:string" minOccurs="1"
            maxOccurs="14" />
          <xs:element name="DDSslot" type="tns:DDEntry" minOccurs="1"
            maxOccurs="14" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="variable" type="tns:Variable" minOccurs="1"
      maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="Address" type="xs:byte" use="required" />
</xs:complexType>

<xs:element name="EIHKonfig">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Netzwerk" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VApparat" type="tns:VApparat" minOccurs="1"
              maxOccurs="unbounded" />
          </xs:sequence>
          <xs:attribute name="Name" type="xs:string" use="required" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

**Quelltext B.8:** Schema für die NodeSet-Extension des Informationsmodells des FDI Communication Servers im Equipment Instance Handler

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.inf.tu-dresden.de/pk/stth/nodeset-ext"
  xmlns:tns="http://www.inf.tu-dresden.de/pk/stth/nodeset-ext"
  elementFormDefault="qualified">

  <xs:element name="VariableReference">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="NetworkName" type="xs:string" minOccurs="1"
          maxOccurs="1" />
        <xs:element name="InstanceName" type="xs:string" minOccurs="1"
          maxOccurs="1" />
        <xs:element name="VariableName" type="xs:string" minOccurs="1"
          maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```



# C Weitere Abbildungen

## C.1 Beispielanlage

Die Abbildung des R&I-Fließbildes der Beispielanlage ist auf die Abbildungen C.1 bis C.4 aufgeteilt und überlappend geschnitten. Ein Übersichtsbild ist in Abbildung 4.5 auf Seite 63 abgedruckt. Das vereinfachte Original-R&I-Fließbild aus [274] ist in Abbildung C.5 auf Seite 190 abgebildet. Hieraus können die Bezeichnungen für die Apparate entnommen werden. Das originale Bezeichnungsschema der Messstellen wurde wegen der Gliederung in Teilanlagen nicht übernommen.

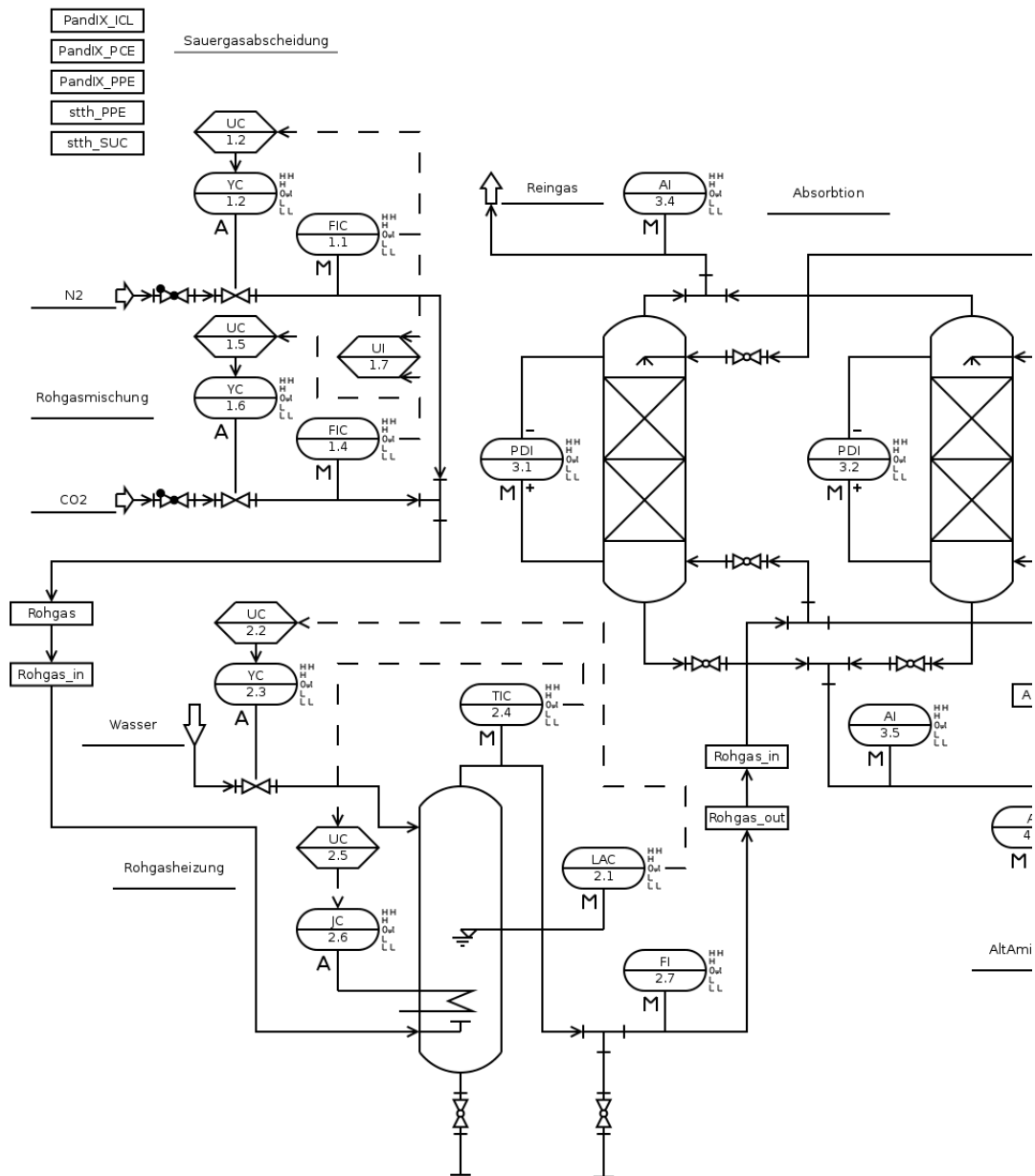


Abbildung C.1: In *Dia* modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 1

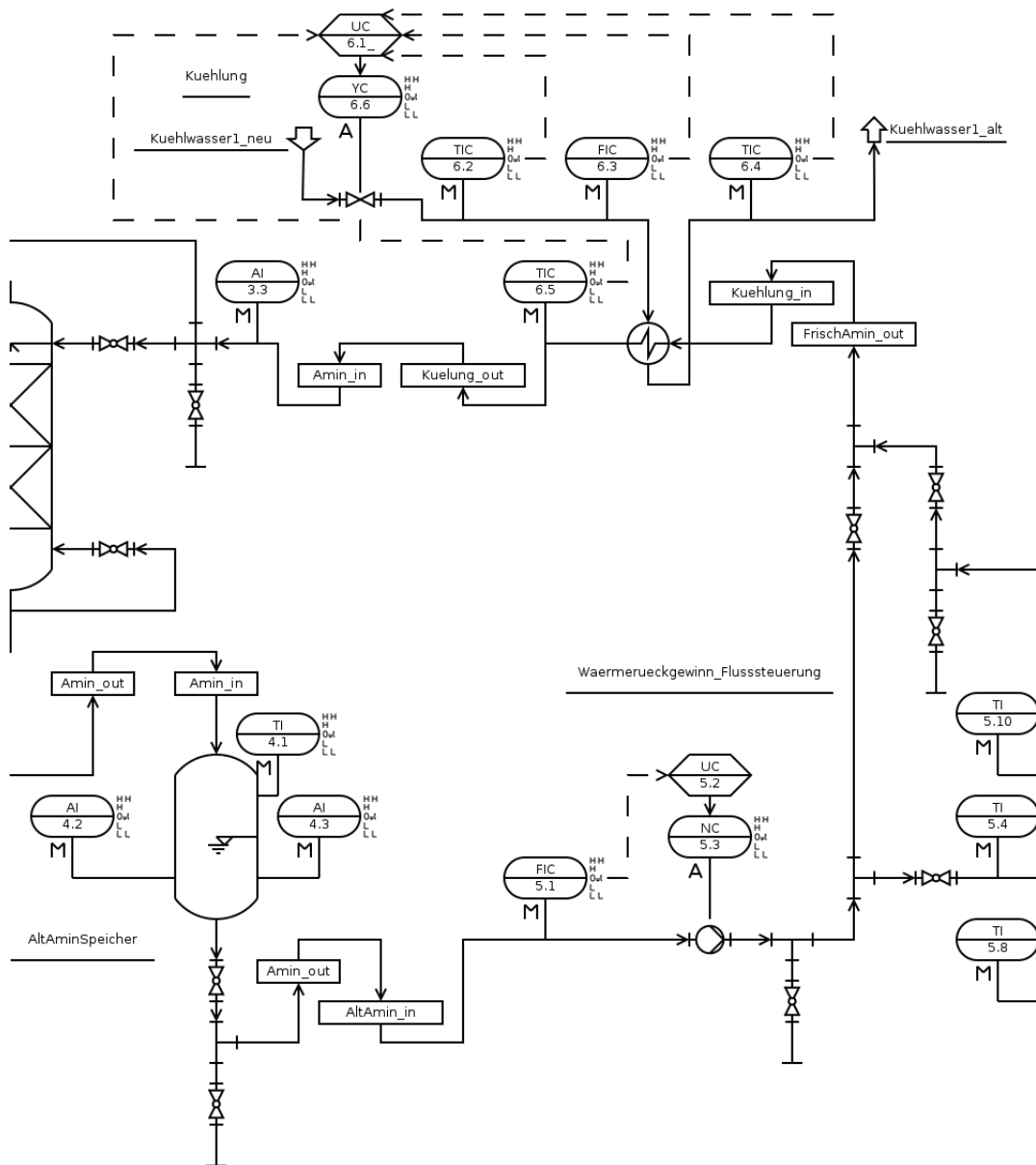


Abbildung C.2: In Dia modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 2

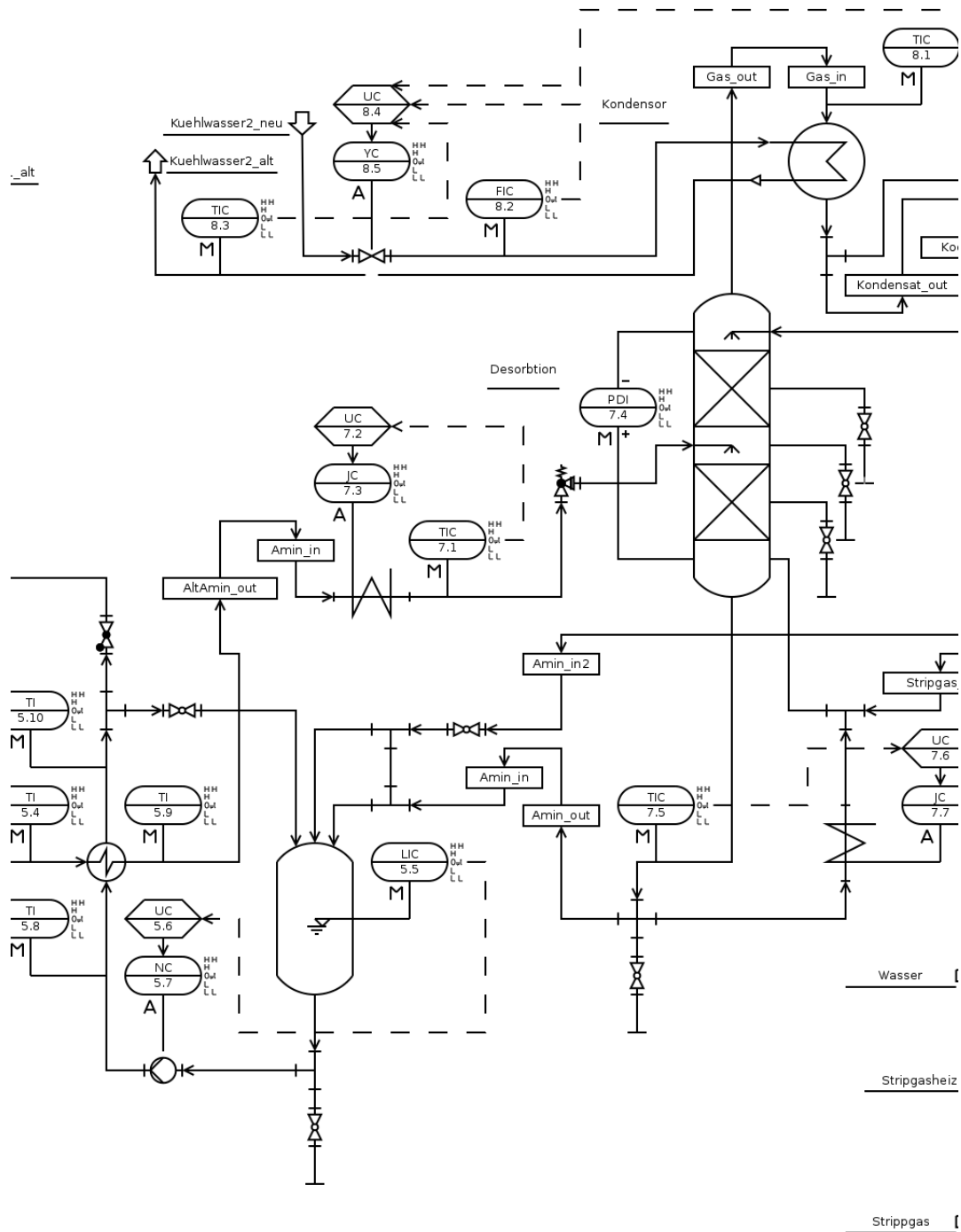


Abbildung C.3: In *Dia* modellierte Beispielanlage (aus [4] auf Basis von [274]), Teil 3

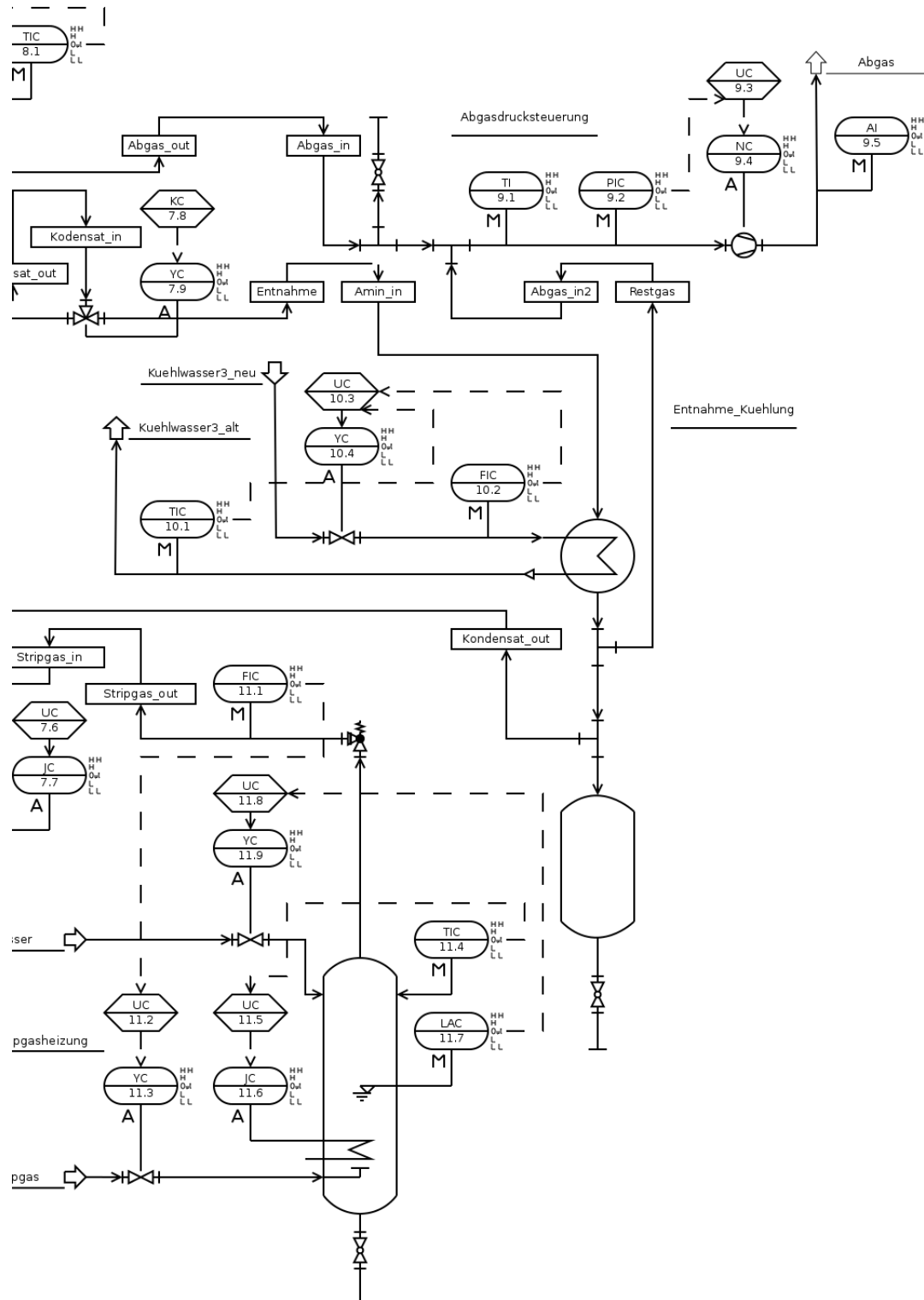


Abbildung C.4: In *Dia* modellierte Beispieranlage (aus [4] auf Basis von [274]), Teil 4

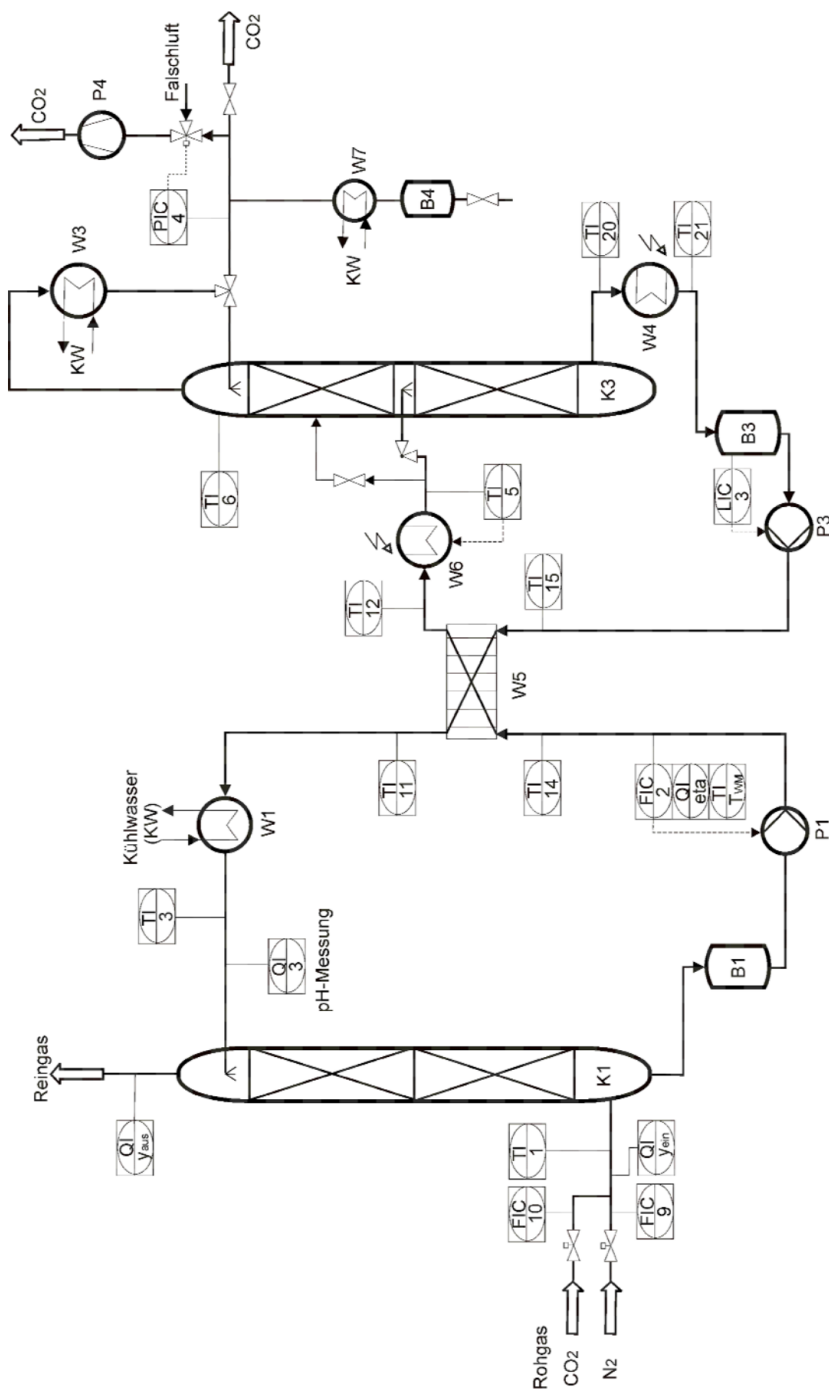


Abbildung C.5: Vereinfachtes RI-Fließschema der Beispielanlage (aus [274, Bild 4.2, S. 54])