

Modulare Smart-Grid-Automatisierungsarchitektur mit integriertem Konfigurationsprozess auf Basis der IEC 61850-6

Von der

Fakultät für Elektrotechnik und Informationstechnik
der
Technischen Universität Dortmund

genehmigte

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von

Dipl.-Ing. Björn Bauernschmitt
aus Hamm

Referent: Prof. Dr.-Ing. Christian Rehtanz, Technische Universität Dortmund

Korreferent: Prof. Dr.-Ing. Markus Zdrallek, Bergische Universität Wuppertal

Tag der mündlichen Prüfung: 12.10.2023

Danksagung

Die vorliegende Arbeit ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Energiesysteme, Energieeffizienz und Energiewirtschaft der Technischen Universität Dortmund entstanden. Viele Menschen haben zum Gelingen dieser Arbeit beigetragen. Ich danke ihnen allen. Bei einigen von ihnen möchte ich mich persönlich bedanken.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Christian Rehtanz, der mir die Gelegenheit zur wissenschaftlichen Arbeit am Institut gab. Er hat mich in dieser Zeit nicht nur fachlich, sondern auch persönlich hervorragend betreut und bei der Entstehung dieser Arbeit stets unterstützt. Er gab mir die Möglichkeit und den Freiraum, mich mit vielfältigen Themen und Fragestellungen der elektrischen Energietechnik zu befassen.

Des Weiteren bedanke ich mich bei Herrn Prof. Dr.-Ing. Markus Zdrallek vom Lehrstuhl für Elektrische Energieversorgungstechnik der Bergischen Universität Wuppertal für die freundliche Übernahme des Korreferats.

Allen Mitarbeiterinnen und Mitarbeitern, die mich während meiner Zeit am ie³ begleitet haben, danke ich für die freundschaftliche und kollegiale Zusammenarbeit, die angenehme Arbeitsatmosphäre und den produktiven Austausch. Bedanken möchte ich mich zuallererst bei Dominik Hilbrich. Über all die Jahre, in denen wir das Büro geteilt, gemeinsam an vielen interessanten Konferenzen teilgenommen und auch privat vieles unternommen haben, ist er nicht nur ein Kollege, sondern vielmehr ein toller Freund geworden. Ganz herzlich danken möchte ich auch allen weiteren, ehemaligen und wie aktuellen Personen meines Teams, namentlich Marvin Albrecht, Felix Goeke, Kay Görner, Svenja Joseph, Björn Martin Keune, Rajkumar Palaniappan, Sebastian Raczka, Thomas Schwierz und Michael Steglich. Für die Unterstützung im Institutsalltag möchte ich mich insbesondere auch bei Emily Elvermann, Nicole Funke, Nina Ganser, Ulf Häger, Willi Horenkamp, Dieter König, Tobias Loidl, Sven Seibt und Klaus-Dieter Tesch bedanken.

Bei meinen Studierenden, die ich auf dem Weg durch ihr Studium ein Stück weit begleiten durfte, möchte ich mich für ihre Unterstützung bei der Umsetzung der vielen Details in meiner Forschung bedanken.

Den Kollegen der KoCoS Messtechnik AG danke ich für ihre technische Unterstützung. Besonders dem Geschäftsführer, Thomas Becker, gilt mein Dank für den fachlichen Austausch in vielen Gesprächen und Treffen.

Bei meinem ehemaligen Kollegen Jan Arph von der H & S Hard- & Software Technologie GmbH & Co. KG möchte ich mich bedanken für den fachlichen Austausch über all die Jahre. Ihm wie auch Frank Richter von der Energie Waldeck-Frankenberg GmbH danke ich für das Korrekturlesen einzelner Kapitel dieser Arbeit.

Mein Dank gilt auch den Mitarbeiterinnen und Mitarbeitern der Unternehmen und Forschungseinrichtungen, mit denen ich in all den Forschungsprojekten zusammenarbeiten durfte.

Ganz besonders bedanke ich mich bei meiner wundervollen Frau Nadine für ihre Liebe, Geduld und motivierende Unterstützung. Auch hat sie mich im Homeoffice immer mit leckerem Essen und Getränken versorgt. Schließlich danke ich auch unserem Kater Felix, der mich am heimischen Schreibtisch regelmäßig daran erinnerte, auch mal eine Pause einzulegen.

Dortmund, im März 2023

Kurzfassung

Getrieben durch die Energiewende, befindet sich die elektrische Energieversorgung in einem steten Transformationsprozess. Insbesondere die Verteilnetze sehen sich mit der Integration einer großen Zahl neuer Netzteilnehmer konfrontiert. Um diese Herausforderungen zu meistern, müssen sich die Verteilnetze von bisherigen passiven Netzen hin zu intelligenten Smart Grids wandeln. Hierzu werden die Verteilnetze mit Systemen und Funktionen zur Messung, Überwachung, Steuerung, Automatisierung und zum Schutz ausgestattet. Durch die Anzahl und räumlichen Ausdehnung der Verteilnetze und die daraus resultierende Vielzahl an zu integrierenden Komponenten werden hierzu allerdings – verglichen mit den aus Netzen der Hoch- und Höchstspannung bekannten Systemen – gänzlich neue Lösungen benötigt.

In dieser Arbeit wird eine Smart-Grid-Automatisierungsarchitektur vorgestellt, die es ermöglicht, verschiedenste schutz- und leittechnische sowie neuartige Smart-Grid-Funktionen modular und flexibel auf einer einheitlichen Plattform abzubilden. Berücksichtigt wird vor allem auch eine klare Trennung von Software und Hardware. Eine Besonderheit stellt der integrierte Konfigurationsprozess auf Basis des Standards IEC 61850-6 dar. Dieser stellt sicher, dass auch eine große Anzahl an Systemen schnell und effizient projiziert und konfiguriert werden kann – ein deutlicher Vorteil gegenüber existierender Lösungen.

Es wird ein Softwareframework geschaffen, das es ermöglicht, Funktionen und Algorithmen auf einheitliche Art und Weise zu implementieren und parallel und wechselwirkungsfrei auf der Systemplattform auszuführen. Anschließend wird der zuvor genannte Konfigurationsprozess entworfen und in das Softwareframework integriert. Exemplarische Schutz- und Smart-Grid-Funktionen sowie Kommunikationsprotokolle werden umgesetzt. Im Rahmen sowohl von Laborprüfungen als auch innerhalb eines Feldtests erfolgt eine Verifikation der implementierten Funktionen selbst als auch des Konfigurationsprozesses.

Abstract

Driven by the energy turnaround, the electrical energy supply is in a constant transformation process. Distribution grids in particular are confronted with the integration of a large number of new grid participants. In order to master these challenges, distribution grids must transform themselves from passive grids to intelligent smart grids. For this purpose, distribution grids are being equipped with systems and functions for measurement, monitoring, control, automation and protection. However, due to the number and spatial expansion of the distribution grids and the resulting multitude of components to be integrated, completely new solutions are required – compared to the systems known from high-voltage and extra-high-voltage grids.

In this thesis, a smart grid automation architecture is presented that makes it possible to map a wide variety of protection and control functions as well as novel smart grid functions in a modular and flexible way on a uniform platform. Above all, a clear separation of software and hardware is taken into account. A special feature is the integrated configuration process based on the IEC 61850-6 standard. This ensures that even a large number of systems can be planned and configured quickly and efficiently – a clear advantage over existing solutions.

A software framework is created that enables functions and algorithms to be implemented in a uniform manner and executed in parallel and without interaction on the system platform. Subsequently, the aforementioned configuration process is designed and integrated into the software framework. Exemplary protection and smart grid functions as well as communication protocols are implemented. Verification of both the implemented functions themselves and the configuration process take place within the framework of both laboratory tests and a field test.

Inhaltsverzeichnis

| | |
|---|-----|
| Danksagung | III |
| Kurzfassung | V |
| Abstract | VI |
| 1 Einleitung..... | 1 |
| 1.1 Motivation | 1 |
| 1.2 Begriffsdefinitionen..... | 5 |
| 1.3 Einordnung und Ziele dieser Arbeit | 7 |
| 1.4 Forschungsfragen | 10 |
| 1.5 Gliederung | 11 |
| 2 Schutz-, Leit- und Automatisierungssysteme für Smart Grids..... | 13 |
| 2.1 Kategorisierung | 14 |
| 2.2 Anforderungen | 18 |
| 2.3 Stand von Forschung und Technik | 21 |
| 2.4 Zusammenfassung..... | 28 |
| 3 Engineering und Projektierung elektrischer Energieanlagen | 29 |
| 3.1 Bisherige Engineering- und Projektierungsprozesse | 31 |
| 3.1.1 Anwendungsbeispiel Transportnetzbetreiber | 32 |
| 3.1.2 Anwendungsbeispiel Verteilnetzbetreiber | 37 |
| 3.1.3 Zusammenfassung | 39 |
| 3.2 Die Normenreihe IEC 61850..... | 40 |
| 3.3 Datenmodelle nach IEC 61850-6..... | 43 |
| 3.3.1 System Configuration description Language..... | 43 |
| 3.3.2 Sprachelemente..... | 44 |
| 3.4 Der Engineeringprozess nach IEC 61850-6..... | 49 |
| 3.4.1 Verfügbare Engineeringwerkzeuge | 59 |
| 3.5 Zusammenfassung..... | 62 |
| 4 Realisierung einer Smart-Grid-Automatisierungsarchitektur..... | 63 |
| 4.1 Systemplattform..... | 64 |
| 4.2 Softwareframework | 68 |

| | | |
|-------|---|-----|
| 4.3 | Funktionen..... | 70 |
| 4.3.1 | Schutzfunktionen..... | 71 |
| 4.3.2 | Spannungsregelung | 76 |
| 4.3.3 | Netzzustandsschätzung | 77 |
| 4.3.4 | Weitere Funktionen und Algorithmen | 82 |
| 4.4 | Kommunikation und Datenhaltung..... | 83 |
| 4.4.1 | UDP | 83 |
| 4.4.2 | IEC 61850-8-1 (MMS, GOOSE) | 86 |
| 4.4.3 | TCP | 87 |
| 4.4.4 | Modbus..... | 88 |
| 4.4.5 | SQLite | 89 |
| 4.5 | Systemkonfiguration | 90 |
| 4.5.1 | Genutzte SCL-Sprachelemente..... | 90 |
| 4.5.2 | Automatische Gerätekonfiguration..... | 105 |
| 4.5.3 | Programmablaufsteuerung | 112 |
| 4.6 | Zusammenfassung..... | 116 |
| 5 | Verifikation der Smart-Grid-Automatisierungsarchitektur..... | 117 |
| 5.1 | Schutzfunktionen..... | 117 |
| 5.1.1 | Eigenschaften des implementierten Distanzschutzes..... | 117 |
| 5.1.2 | Prüfablauf | 119 |
| 5.1.3 | Ergebnisse der Distanzschutzprüfung | 120 |
| 5.2 | Netzzustandsschätzung | 125 |
| 5.2.1 | Netzmodellaufbereitung und Erstellung einer Prüfumgebung..... | 126 |
| 5.2.2 | Verifikation des Algorithmus der Netzzustandsschätzung | 132 |
| 5.2.3 | Darstellung der Gesamtsystemstruktur | 133 |
| 5.2.4 | Engineering und Konfiguration des Gesamtsystems | 135 |
| 5.2.5 | Vorprüfung der Komponenten im Labor..... | 144 |
| 5.2.6 | Einbringung der Komponenten in das Feld | 145 |
| 5.2.7 | Durchführung und Ergebnisse des Feldtests..... | 146 |
| 5.3 | Zusammenfassung..... | 153 |

| | |
|--|-----|
| 6 Zusammenfassung und Ausblick..... | 155 |
| 6.1 Zusammenfassung..... | 155 |
| 6.2 Bewertung | 157 |
| 6.3 Ausblick..... | 159 |
| Abkürzungsverzeichnis..... | 163 |
| Formelverzeichnis | 169 |
| Literaturverzeichnis | 171 |
| Wissenschaftlicher Tätigkeitsnachweis | 183 |
| Anhang A Strukturierung der Namensräume | 189 |
| Anhang B Protokolldetails der TCP-Kommunikation | 191 |
| Anhang C Modellierung der Modbus-Kommunikation | 193 |
| Anhang D Konfiguration der binären Ein- und Ausgänge | 195 |

1 Einleitung

1.1 Motivation

Getrieben durch die Energiewende als Abkehr von fossilen Energiequellen und dem Wechsel hin zu erneuerbaren Energiequellen erlebt die elektrische Energieversorgung, ebenso wie auch andere Energiesysteme, seit den letzten zwanzig Jahren, sowohl national als auch international, einen enormen Transformationsprozess. Auf der einen Seite steht der ökologisch, ökonomisch und sozial getriebene Rückbau konventioneller Großkraftwerke auf Basis von Braun-, Steinkohle und – zumindest in Deutschland – Kernenergie. Auf der anderen Seite stehen der Zubau und die Integration dezentraler, vorrangig auf regenerativen Energiequellen wie Photovoltaik (PV) und Windkraft (WK) basierender Erzeugungsanlagen, neuartiger Lasten wie Wärmepumpen und Elektromobilität sowie Energiespeicher. Abbildung 1.1 zeigt die Struktur der transformierten elektrischen Energieversorgung.

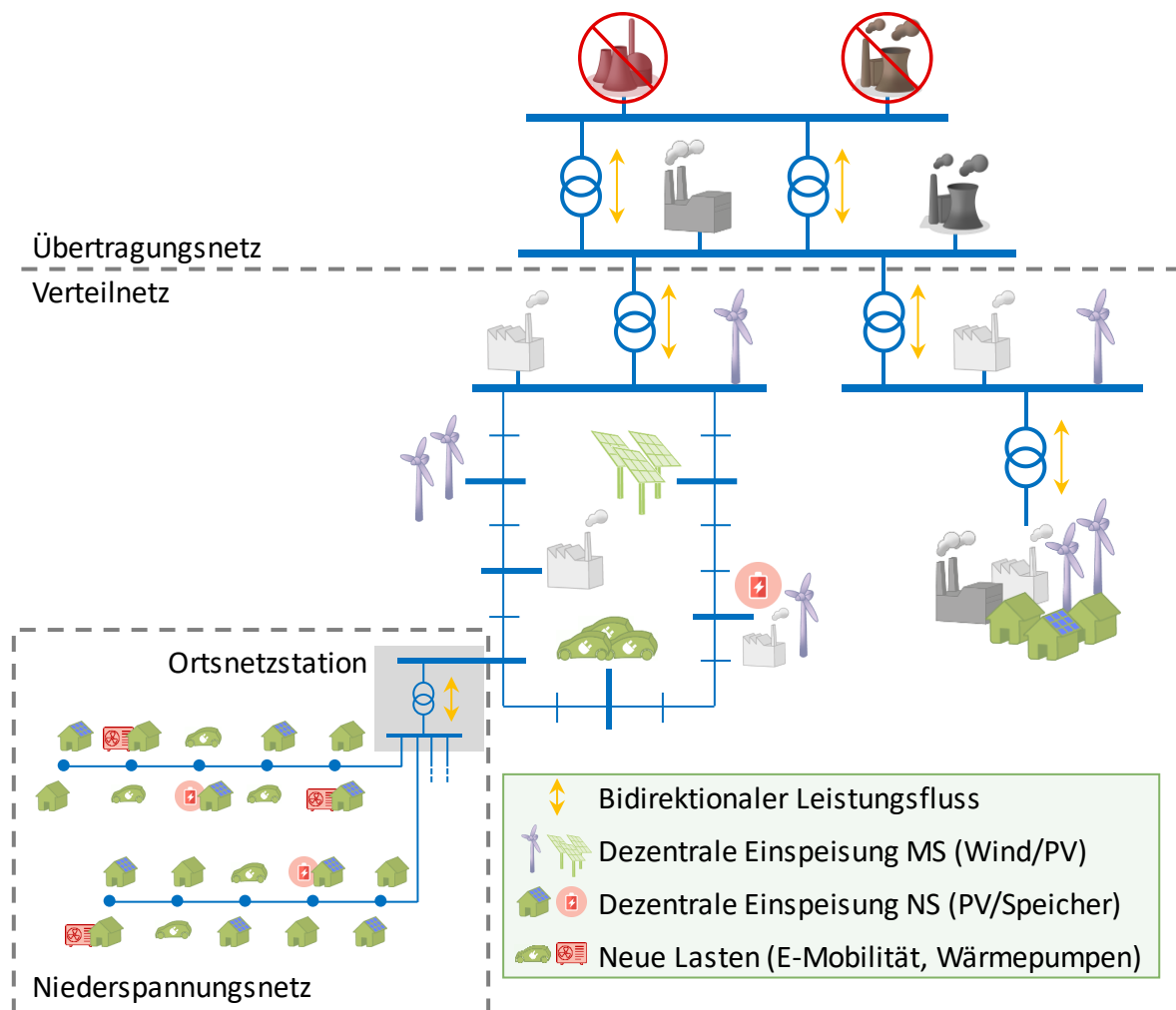


Abbildung 1.1: Struktur der transformierten elektrischen Energieversorgung

Besonders markant im Bereich der elektrischen Energieversorgung ist der Wandel der traditionell unidirektionalen Energieflüsse, von den oberen in die unteren Spannungsebenen, hin zu bidirektionalen Energieflüssen. Die Diversität der neuen Netzteilnehmer und das tendenziell stochastische Energiedargebot erneuerbarer Energiequellen führen zu einer hohen Gleichzeitigkeit und Volatilität der elektrischen Energieflüsse. Dadurch, dass diese Netzteilnehmer zu großen Teilen in die Verteilnetze der Hoch-, Mittel- und Niederspannung (HS, MS, NS) integriert werden – bei den erneuerbaren Energiequellen sind es auf die Leistung bezogen bereits 95 % [1] –, stehen insbesondere diese Netze neuen Herausforderungen gegenüber. Diese Tatsache charakterisierend, hat sich der Satz „Die Energiewende findet in den Verteilnetzen statt“ etabliert [2]. Hieraus wird die Bedeutung der Verteilnetze für die aktuelle und zukünftige elektrische Energieversorgung deutlich und die gesteigerte Verantwortung der Verteilnetzbetreiber für einen sicheren, zuverlässigen und effizienten Netzbetrieb.

Folge der bidirektionalen Energieflüsse sind etwa Probleme bei der Spannungshaltung: Verteilnetze, die ursprünglich für den Lastfall geplant wurden, erfahren an weiter entfernten, schwach angebundenen Ausläufern Überspannungen durch Einspeisung aus neu hinzugekommenen, dezentralen Energiequellen. Leistungsstarke Lasten wie Elektromobilität und Wärmepumpen, aber auch die Konzentration einer hohen Anzahl dezentraler Erzeugungsanlagen, können zu einer zeitlich ausgeprägten Leistungsspitze und ggf. zu thermischen Überlastungen von Netzbetriebsmitteln führen. Eine Lastspitze kann beispielsweise auftreten, wenn in einem Netzsegment mit hoher Durchdringung von Elektromobilität zu ähnlichen Zeiten, etwa nachmittags nach der Heimkehr von der Arbeitsstelle, eine große Anzahl an Fahrzeugen geladen wird. Eine Einspeisespitze kann dagegen bei hoher Einspeisung aus Photovoltaikanlagen bei gleichzeitig geringer Last, etwa zur Mittagszeit an einem sonnigen Sonntag, auftreten.

Den Problematiken der thermischen Überlastung und der Spannungshaltung kann auf konventionelle Art mit einem Ausbau der Netze begegnet werden. Die Betrachtung der Häufigkeit und Dauer von Netzengpässen zeigt jedoch, dass die Auslegung der Netze auf Spitzenleistungsflüsse nicht effizient ist. Darüber hinaus ist ein konventioneller Netzausbau durch die große räumliche Ausdehnung der Verteilnetze und die hohe Anzahl der Netzbetriebsmittel sehr kostspielig. Auch sind zukünftige Anpassungen an sich verändernde Netzanforderungen nur mit großem Aufwand umsetzbar.

Ebenso ist der Netzschutz in den Verteilnetzen nicht für die neuen Bedingungen ausgelegt: Als Fehlerkriterium wird in Niederspannungsnetzen ausschließlich und in Mittelspannungsnetzen überwiegend der Überstrom herangezogen. Die technische Realisierung des Netzschutzes erfolgt dabei in Niederspannungsnetzen durch

Sicherungen, in Mittelspannungsnetzen durch Sicherungen oder Schutzgeräte mit Überstromzeitauslösung. Zur Detektion eines Fehlers ist das Auftreten eines ausreichend großen Fehlerstroms erforderlich. Neben dem überlagerten Netz leisten jedoch auch dezentrale Energiequellen einen Beitrag zum Fehlerstrom. Infolgedessen kann der Fehlerstrombeitrag aus dem überlagerten Netz so gering werden, dass eine zuverlässige Auslösung des Schutzorgans nicht mehr gegeben ist. Mittelspannungsseitige Abgänge in Umspannanlagen sind teilweise mit Unterfrequenzrelais ausgestattet. Deren Aufgabe ist es, im Falle einer großräumigen Netzstörung mit großem Energiedefizit und als Folge einer Unterfrequenz Teile der unterlagerten Netze gestaffelt abzuschalten, um das Netz durch Reduzierung der Last zu stützen. Die Integration einer großen Anzahl dezentraler Energiequellen kann jedoch dazu führen, dass in einzelnen Netzsegmenten zeitweise ein Energieüberschuss auftritt und Energie in das überlagerte Netz zurückgespeist wird. Ein konventionelles Unterfrequenzrelais berücksichtigt jedoch nicht die Energieflussrichtung, sondern würde auch in diesem Fall eine, mit Blick auf die Netzstützung, kontraproduktive Abschaltung vornehmen.

Eine Alternative zum konventionellen Netzausbau liegt im Übergang vom bisher rein passiven hin zu einem aktiven, intelligent betriebenen Verteilnetz, welches auch als Smart Grid bezeichnet wird. Hierzu wird das Verteilnetz mit Systemen und Funktionen zur Messung, Überwachung, Steuerung, Automatisierung und zum Schutz ausgestattet. Dies ist in Abbildung 1.2 dargestellt, die Abbildung 1.1 um solche Systeme erweitert. Durch Messung und Überwachung des Netzes wird eine Beobachtbarkeit des Netzzustands erreicht, etwaige Netzengpässe lassen sich identifizieren. Mit diesem Wissen können Einspeiseanlagen, Speicher und flexible Lasten bedarfsgerecht, im Sinne einer optimalen Netzausnutzung, gesteuert und koordiniert und somit Netzengpässe aufgelöst werden. Gleichzeitig wird so die Integration weiterer Netzkunden und Erzeuger, insbesondere Erneuerbare-Energie-Anlagen, durch Vermeidung eines zeit- und kostenaufwendigen Netzausbaus vereinfacht und beschleunigt, was der Energiewende Vorschub leistet. Durch die Beobachtbarkeit und Fernsteuerbarkeit der Netze kann der Netzbetreiber darüber hinaus schneller auf Fehler und Störungen reagieren, die Ausfallzeiten verringern und somit die Versorgungszuverlässigkeit erhöhen. Mit dem Einsatz höherwertiger Schutzeinrichtungen kann den zuvor genannten Unzulänglichkeiten der bisherigen Netzschutzkonzepte in Verteilnetzen begegnet werden.

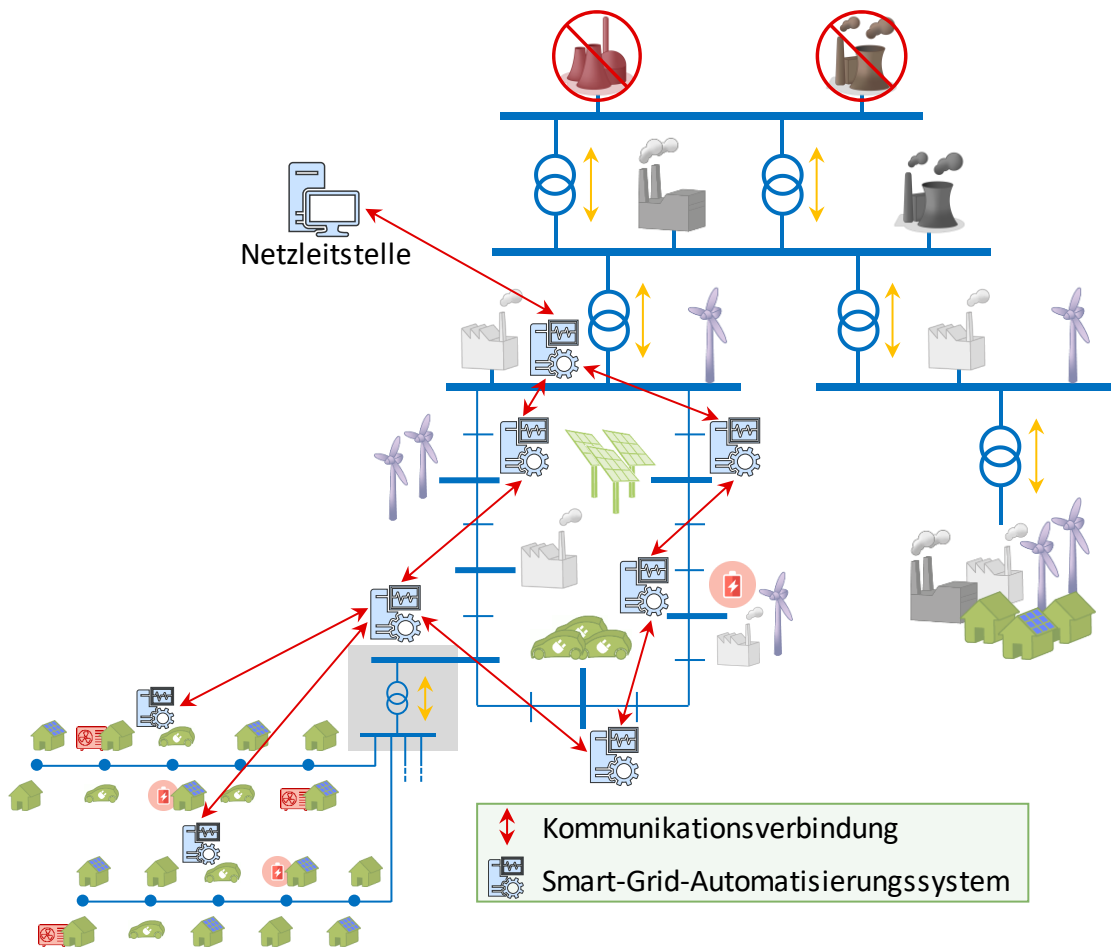


Abbildung 1.2: Erweiterung des Netzes um Smart-Grid-Automatisierungssysteme

Der grundsätzliche Bedarf für den Einsatz intelligenter Systeme und Funktionen in Verteilnetzen, vor allem zur Optimierung und Minimierung des notwendigen Netzausbaus, wurde in mehreren Studien gezeigt [3–6]. Innerhalb des nationalen Energieversorgungsnetzes existieren rund 600.000 Ortsnetzstationen (ONS), von denen mittelfristig 10-20 % mit Mess-, Steuer- und Regelungstechnik ausgestattet werden [7, 8]. Zu den Anlagen, die mit solchen Systemen nachgerüstet werden (Retrofit), kommen rund 2.000-4.000 Anlagen, die aufgrund steigender regenerativer Energieeinspeisung jedes Jahr zusätzlich neu gebaut werden. Allein in Deutschland besteht damit bei konservativer Schätzung ein jährlicher Bedarf von rund 5.000 solcher Systeme für die Verteilnetzautomatisierung innerhalb der nächsten zehn Jahre.

National wie international haben sich bereits zahlreiche Forschungsprojekte mit Fragestellungen rund um Smart Grids beschäftigt, verschiedenste Prototypen wurden in Netze integriert und erprobt. Ebenso existiert bereits eine Reihe kommerzieller Produkte für die Überwachung, Steuerung und Automatisierung von Verteilnetzen. Diese Projekte und Systeme lösen zwar einzelne Probleme und Herausforderungen, sind aber dennoch immer auch durch eine Reihe von Nachteilen gekennzeichnet. Die in dieser Arbeit vorgestellten Lösungen begegnen einigen dieser Nachteile.

1.2 Begriffsdefinitionen

In dieser Arbeit werden einige Kernbegriffe genutzt. Um ein gemeinsames Verständnis dieser Begriffe sicherzustellen, erfolgt an dieser Stelle zunächst eine Definition. Der Fokus liegt hierbei insbesondere auf den Arbeiten in dieser Dissertation und allgemein auf dem Bereich von Anlagen der elektrischen Energieversorgung, so dass sich teilweise eine spezifische Ausprägung der Begriffe ergibt.

Planung

Soll ein Projekt, beispielsweise der Neubau oder die Erweiterung einer elektrotechnischen Anlage, umgesetzt werden, so steht an erster Stelle die Planung. Die Planung betrachtet auf theoretischer Ebene den gesamten Prozess, der erforderlich ist, um vom Ausgangszustand in den Zielzustand überzugehen. Hierbei muss zunächst das Ziel klar definiert werden. Auf dieser Basis erfolgt die Kalkulation von Kosten, Zeit und Personal. Auch sind rechtliche und regulatorische Rahmenbedingungen ebenso wie Umweltschutzvorgaben zu berücksichtigen.

Projektierung

Auf die Planung folgt die Projektierung. Sie umfasst auf praktischer Ebene die Definition der konkreten, auszuführenden Arbeitsschritte, die erforderlich sind, um ein Projekt zielgerecht umzusetzen. Hierzu zählen etwa die Auswahl geeigneter technischer Komponenten, die Koordination der Arbeitsschritte und deren Ineinandergreifen, aber auch die Beantragung von Genehmigungen sowie die Beauftragung von Lieferanten und Dienstleister.

Engineering

Das Engineering kann als Teilgebiet der Projektierung mit Konzentration auf die technischen Aspekte betrachtet werden. Innerhalb des Engineerings erfolgt die Umsetzung der konkreten technischen Arbeitsschritte. Hierzu zählen etwa die Durchführung von Berechnungen, die Konstruktion von Anlagen und Komponenten, die Erstellung von Datenmodellen, Stromlaufplänen und Zeichnungen sowie die Dokumentation. Im Rahmen des Standards IEC 61850 und begleitender Dokumente und Veröffentlichungen wird die gesamte Projektierung einer Anlage der elektrischen Energieversorgung, vor allem der Sekundärtechnik (Schutz- und Leittechnik), als Engineeringprozess bezeichnet.

Konfiguration

Konfiguration bezeichnet die Arbeiten, die erforderlich sind, um Anlagen und Komponenten so vorzubereiten, dass sie die projektierten Aufgaben erfüllen. Hierzu gehören etwa Einrichtungs- und Einstellarbeiten (z. B. Parameter), das Erstellen von Programmabläufen und das Aufeinanderabstimmen mehrerer Komponenten, so dass eine reibungslose Funktion des Gesamtsystems möglich wird. Im Rahmen des Standards IEC 61850 wird insbesondere das Übernehmen von Informationen aus Datenmodellen, die während des Engineeringprozesses erstellt wurden, in die Geräte als Konfiguration bezeichnet.

Implementierung

In dieser Arbeit bezeichnet Implementierung die konkrete Erstellung eines Algorithmus oder einer Software durch Programmieren, so dass eine Ausführung auf einem Rechnersystem ermöglicht ist.

Systemplattform

Als Systemplattform wird in dieser Arbeit ein Hardwaresystem zur Ausführung von Funktionen und Algorithmen bezeichnet. Die Systemplattform besitzt mindestens ein Rechnersystem, etwa in Form eines eingebetteten Computers, optional aber auch eine Messhardware, binäre Ein- und Ausgänge und weitere Schnittstellen vor allem zur Anbindung an energietechnische Systeme. Neben der reinen Hardware schließt der Begriff Systemplattform jedoch auch grundlegende Softwarekomponenten wie ein Betriebssystem und Firmware mit ein.

Architektur

Als Architektur, Systemarchitektur oder – speziell hier – Smart-Grid-Automatisierungsarchitektur wird in dieser Arbeit eine Einheit aus Hard- und Software bezeichnet. Diese Einheit ermöglicht es, Funktionen und Algorithmen auszuführen, umfasst aber auch Verfahren zur Kommunikation und Konfiguration. Der Begriff Architektur beschreibt dabei sowohl den konzeptionellen Aufbau eines solchen Systems als auch dessen Realisierung in Form einer konkreten Hard- und Software.

Softwareframework

Softwareframework bezeichnet die in dieser Arbeit erstellte Implementierungsumgebung, die es ermöglicht, Funktionen und Algorithmen umzusetzen und auf der Systemplattform auszuführen. Hierzu bietet das Softwareframework wohldefinierte Schnittstellen, so dass sich Funktionen unabhängig voneinander implementieren lassen und gemeinsame Komponenten wie ein Online-Datenmodell zum Datenaustausch und zur Kommunikation nutzen können.

Verifikation

Verifikation bezeichnet den Nachweis, dass ein technisches System diejenigen Eigenschaften erfüllt, die in einer zuvor erstellten Spezifikation definiert wurden. Es handelt es sich also um einen entwicklungsinternen Prozess, der die ordnungsgemäße Umsetzung der Spezifikation in eine konkrete Lösung überprüft.

Validierung

Validierung bezeichnet hingegen den Nachweis, dass ein technisches System die Anforderungen erfüllt, die ein Einsatz in einer realen Betriebsumgebung stellt. Es wird überprüft, ob die Lösung den Anforderungen des Kunden gerecht wird und somit der Kundenbedarf ordnungsgemäß in den Entwicklungsprozess eingeflossen ist.

In-Feld-Bringung

In-Feld-Bringung bezeichnet die konkrete Installation von Komponenten in einer realen Anlage, so dass schließlich ein Betrieb unter realen Bedingungen stattfinden kann. Im Gegensatz zu einzelnen Umsetzungsmaßnahmen etwa im Rahmen von Pilot- und Forschungsprojekten wird mit *Rollout* die flächendeckende Installation gleichartiger, standardisierter Systeme bezeichnet.

1.3 Einordnung und Ziele dieser Arbeit

Diese Arbeit ist zum Teil im Rahmen des Forschungsprojekts *i-Automate „Modular konfigurier- und prüfbare Automatisierungsarchitektur für zukünftige aktive elektrische Energienetze“* [P1] entstanden. Daher sollen an dieser Stelle zunächst Inhalte und Ziele dieses Forschungsprojekts vorgestellt werden, um eine Einordnung dieser Arbeit in den Kontext des Forschungsprojekts und auch eine Abgrenzung davon zu ermöglichen.

Im Forschungsprojekt *i-Automate* wurde eine Systemarchitektur entworfen, die es ermöglicht, schutz- und leittechnische Funktionen als auch Smart-Grid-Automatisierungsfunktionen modular und flexibel auf einer standardisierten Plattform zu integrieren. Als exemplarische Hardwareplattform innerhalb des Forschungsprojekts diente ein kommerzielles, am Markt etabliertes System aus dem Bereich der Messwerterfassung, Netzqualitätsanalyse (Power-Quality-Analyse) und Fehleraufzeichnung (Störschrieb). Besonderes Augenmerk lag auf einer klaren Trennung von Software und Hardware, um Funktionen leicht auch auf andere Hardwareplattformen übertragen zu können. Abbildung 1.3 zeigt das Gesamtkonzept von *i-Automate*.

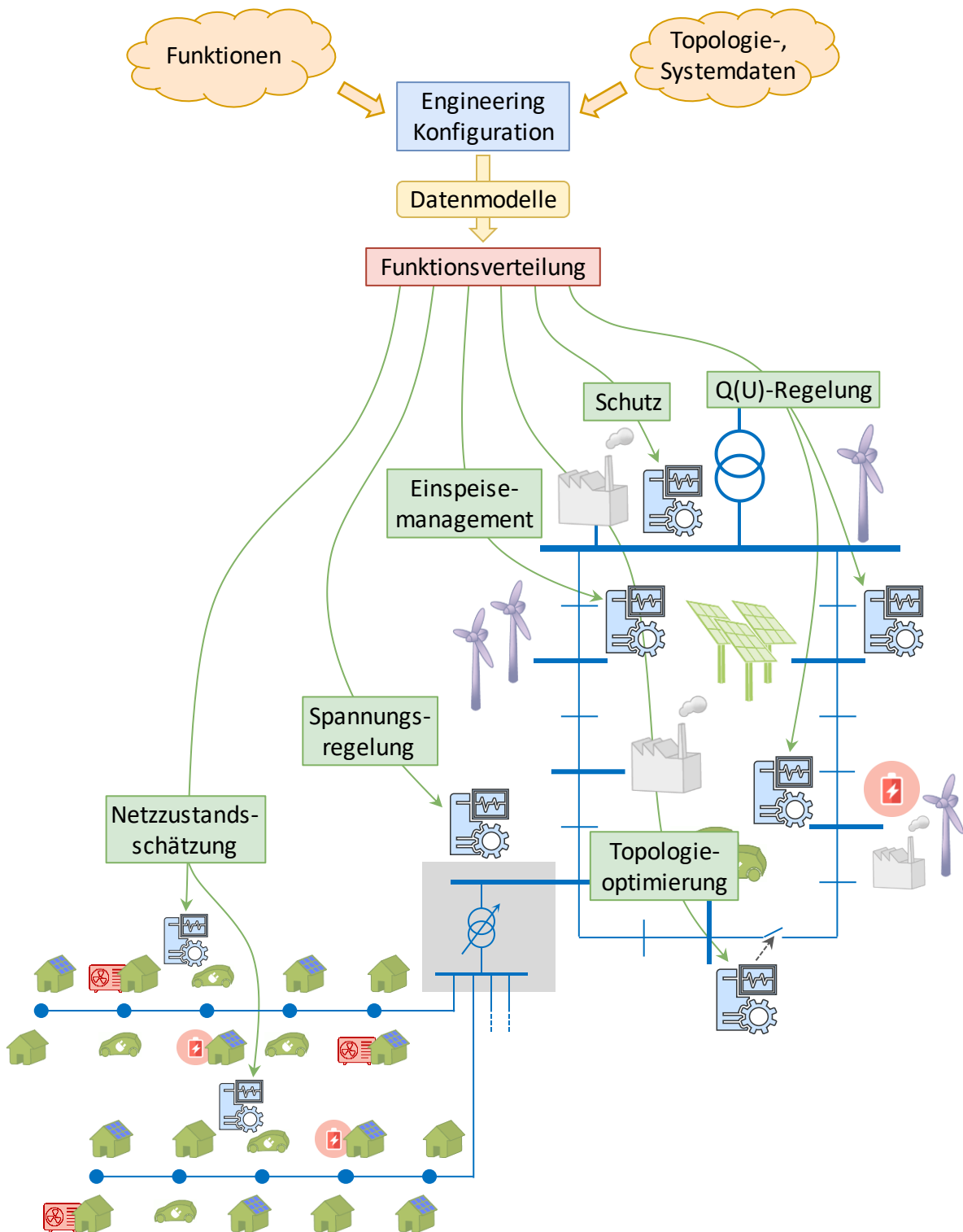


Abbildung 1.3: Gesamtkonzept von i-Automate

Zunächst wurde ein Softwareframework geschaffen, das es ermöglicht, verschiedenste Smart-Grid-Funktionen auf einheitliche Art und Weise zu implementieren und parallel und wechselwirkungsfrei auf der gewählten Hardwareplattform auszuführen. Anschließend wurden als Funktionen und Algorithmen lokale Schutzfunktionen, verschiedene Varianten einer Spannungsregelung, eine Netzzustandsschätzung (engl.: State Estimation, SE) sowie Leistungsflussoptimierungen (engl.: Optimal Power Flow,

OPF) implementiert. Ein besonderes Merkmal der Systemarchitektur besteht darin, dass die Modellierung und Konfiguration der auszuführenden Funktionen, des Datenaustauschs zwischen Funktionen und der Kommunikationsbeziehungen zwischen Systeminstanzen vollständig über Datenmodelle nach IEC 61850-6 [9] (engl.: System Configuration description Language, SCL) erfolgt. Dies stellt sicher, dass die Konfiguration des Systems nahtlos im Rahmen eines übergeordneten Engineeringprozesses nach ebendiesem Standard erfolgen kann. Um dies zu ermöglichen, wurde ein geeigneter Konfigurationsprozess entworfen und in das Softwareframework integriert. Zur Verifikation der implementierten Funktionen wurden entsprechende Prüfverfahren, überwiegend als Hardware-in-the-Loop-Prüfungen unter Ausführung simulationsfähiger Netzmodelle auf einem Echtzeitsimulator, genutzt. Diese wurden insbesondere im Rahmen des Promotionsvorhabens aus [10] entworfen und umgesetzt. Abschließend erfolgte ein mehrmonatiger Feldtest, um das Verhalten ausgewählter Funktionen unter den Bedingungen eines realen Netzbetriebs zu analysieren.

Diese Arbeit hat sich zwei der zuvor genannten Aufgaben angenommen: Eine Kernaufgabe war die Erstellung des Softwareframeworks als Implementierungs-umgebung für Smart-Grid-Funktionen und Algorithmen. Durch klar definierte Schnittstellen ermöglicht es dieses Softwareframework, dass Funktionen unabhängig voneinander implementiert und ausgeführt werden können. Zu berücksichtigen hierbei waren unter anderem die Anforderungen der einzelnen Funktionen und Algorithmen etwa bezüglich Rechendauer, Zeitdeterminismus und Kommunikationsbedarf. Die zweite Kernaufgabe lag im Entwurf eines geeigneten Konfigurationsprozesses, der eine Modellierung und Konfiguration der Systeme auf Basis von Datenmodellen nach IEC 61850-6 ermöglicht, und dessen Integration in das Softwareframework. Da die IEC 61850-6 ebenfalls die Modellierung und Konfiguration der Kommunikationsbeziehungen zwischen Systeminstanzen vorsieht, wurde auch das Thema Kommunikation intensiv berücksichtigt. Neben einer Kommunikation nach IEC 61850-8-1 [11], die durch Nutzung eines externen Kommunikationsstacks bereits gegeben war, wurden weitere Kommunikationsverfahren implementiert, um unter anderem einen Datenaustausch mit einem Echtzeitsimulator zu ermöglichen sowie Drittgeräte an die Systemplattform anzubinden. Die Implementierung der eigentlichen Smart-Grid-Funktionen und Algorithmen war nicht zentraler Teil dieser Arbeit, sondern erfolgte überwiegend im Rahmen paralleler Forschungs- und Laborarbeiten. Ebenso wenig war der Entwurf von Prüfverfahren Teil dieser Arbeit. Wo erforderlich, wurden diese Prüfverfahren lediglich angewandt.

1.4 Forschungsfragen

Vorhandene Systeme und Lösungsansätze zur Automatisierung und Digitalisierung von Verteilnetzen bieten meist nur einzelne oder eine begrenzte Zahl an Funktionalitäten. Zudem besteht oft eine feste Kopplung zwischen Software und Hardware. In der Folge müssen mehrere Einzelkomponenten kombiniert werden, um die gewünschte Gesamtfunktionalität zu erzielen. Dies führt zu einer hohen Komplexität und zu hohen Kosten des Gesamtsystems. Hieraus folgt:

Forschungsfrage 1: Wie lässt sich eine Smart-Grid-Automatisierungsarchitektur so gestalten, dass verschiedenste Funktionen zur Überwachung, Steuerung, Automatisierung und zum Schutz eines Smart Grids modular und flexibel auf einer Plattform umgesetzt und nebeneinander ausgeführt werden können?

Insbesondere die hohe Dynamik im Wandel der Verteilnetze durch die Integration zahlreicher neuer Netzteilnehmer erfordert eine hohe Flexibilität bei sekundärtechnischen Funktionen und Komponenten, um schnell, kostengünstig und ohne Gerätetausch auf sich verändernde Anforderungen reagieren zu können. Hieraus folgt:

Forschungsfrage 2: Wie kann ein Konfigurationsprozess aussehen, der eine solche Smart-Grid-Automatisierungsarchitektur einfach und durchgängig für die Ausführung verschiedener Smart-Grid-Funktionen konfiguriert?

Bereits am Markt verfügbare Systeme oder Lösungsansätze aus der Forschung lassen sich oftmals nur aufwendig und mit spezifischen Softwarewerkzeugen konfigurieren und in ihrer Funktion an den jeweiligen Anwendungsfall anpassen. Das steht einer Austauschbarkeit und Interoperabilität der Systeme entgegen. Der Standard IEC 61850 bietet, neben seiner Eigenschaft als Kommunikationsstandard für elektrische Energieanlagen, durch die in seinem Teil 6 definierten Datenmodelle und Engineeringprozesse die notwendigen Mittel, um die funktionalen primär- und sekundärtechnischen Objekte sowie die Kommunikation solcher Anlagen einheitlich, konsistent und vor allem geräte- und herstellerneutral zu modellieren. Der ursprüngliche Fokus dieses Standards lag auf der Schutz- und Leittechnik klassischer Umspannanlagen der Hoch- und Höchstspannung. Eine Erweiterung auf Anlagen des Verteilnetzes – insbesondere im Sinne eines Smart Grids – ist an dieser Stelle eine logische Folge. Hieraus folgt:

Forschungsfrage 3: Wie können die im Standard IEC 61850-6 definierten Mittel genutzt werden, um einen Konfigurationsprozess zu gestalten, der sich nahtlos in einen herstellerneutralen Engineeringprozess nach ebendiesem Standard einfügt?

1.5 Gliederung

Diese Arbeit ist wie folgt gegliedert:

In Kapitel 2 erfolgt eine grundsätzliche Betrachtung von Schutz-, Leit- und Automatisierungssystemen für Smart Grids. Es werden zunächst Kriterien definiert, anhand derer eine Kategorisierung von Schutz-, Leit- und Automatisierungsfunktionen vorgenommen wird. Hiervon ausgehend werden Anforderungen an die zugrundeliegenden Systeme wie etwa die Hardware definiert. Schließlich werden ausgewählte Systeme und Lösungen – sowohl kommerzielle Produkte als auch prototypische Umsetzungen – hinsichtlich ihrer Eigenschaften und ihres Funktionsumfangs betrachtet, kategorisiert und bewertet.

Kapitel 3 betrachtet das Engineering und die Projektierung elektrischer Energieanlagen. Zunächst wird eine grundsätzliche Übersicht gegeben. Es werden bisherige Engineering- und Projektierungsprozesse anhand zweier Anwendungsbeispiele aufgezeigt. Diese Prozesse sind durch einige Nachteile geprägt. Eine zukunftsweisende Alternative stellt die Normenreihe IEC 61850 dar. Neben ihrer Eigenschaft als Kommunikationsstandard bietet sie Richtlinien und Werkzeuge zur Modellierung schutz- und leittechnischer Systeme. Zunächst werden die dort definierten Datenmodelle erläutert. Die Norm selbst beschreibt das Engineering und die Konfiguration lediglich abstrakt. Eine Konkretisierung erfolgte durch einen DKE-Arbeitskreis. Dessen Ergebnisse in Form detaillierter Engineeringschritte und Engineeringwerkzeuge werden schließlich ausführlich dargestellt.

Kapitel 4 bildet den Kern dieser Arbeit. Nach einer einführenden Vorstellung der exemplarisch genutzten Systemplattform wird das im Rahmen dieser Arbeit erstellte Softwareframework zur Umsetzung von Smart-Grid-Funktionen und Algorithmen erläutert. Anschließend wird im Detail auf die Implementierung ausgewählter Funktionen eingegangen, außerdem auf realisierte Kommunikationsverfahren. Schließlich werden der Entwurf und die Implementierung des Konfigurationsprozesses nach IEC 61850-6 dargestellt. Hierbei werden zunächst die genutzten normativen Sprachelemente beschrieben. Dann wird das Verfahren der automatischen Gerätekonfiguration dargelegt. Schließlich wird erläutert, wie die einzelnen Funktionen auf der Systemplattform ausgeführt und koordiniert werden.

In Kapitel 5 erfolgt eine Verifikation der zuvor erfolgten Implementierungsarbeiten. Hierbei werden sowohl ausgewählte Funktionen selbst als auch der integrierte Konfigurationsprozess verifiziert.

Kapitel 6 liefert eine Zusammenfassung dieser Arbeit und gibt einen Ausblick.

2 Schutz-, Leit- und Automatisierungssysteme für Smart Grids

Die Netze der Höchst- und Hochspannung sind von grundlegender Bedeutung für das elektrische Verbundsystem. Prämisse der Netzbetreiber ist die Sicherstellung eines allzeit zuverlässigen, sicheren und wirtschaftlichen Betriebs. Hierzu zählen insbesondere auch der Schutz der teuren, langlebigen primärtechnischen Komponenten vor Schäden etwa durch elektrische Fehler und die optimale Ausnutzung der Systeme im Sinne größtmöglicher Wirtschaftlichkeit. Daher wurden diese Netze schon früh mit einer Vielzahl ausgefeilter Systeme und Funktionen zum Schutz, zur Fernsteuerung und Überwachung ausgestattet. Die Kosten dieser sekundärtechnischen Einrichtungen können im Vergleich zu denen der primärtechnischen Netzkomponenten tendenziell als eher gering angesehen werden [12, S. 41].

Im Gegensatz dazu verfügen Mittel- und Niederspannungsnetze bisher über keine oder nur wenige sekundärtechnische Komponenten. Der Bau dieser Netze erfolgt unter dem Aspekt der Deckung maximaler Lastszenarien, der Betrieb erfolgt überwiegend blind. In Mittelspannungsnetzen sind lediglich Abgänge aus Umspannstationen und wichtige Knotenpunkte mit Schutzgeräten ausgestattet. Außerdem werden Kurzschluss- und Erdschlussanzeiger eingesetzt, um die Fehlersuche und die Störungsbeseitigung zu unterstützen. Das Auslesen dieser Anzeiger erfordert jedoch oft noch ein personal- und zeitaufwendiges Anfahren der einzelnen Stationen, eine kommunikationstechnische Anbindung an die Netzleitstelle erfolgt selten. Niederspannungsnetze verfügen bisher weitestgehend über keine sekundärtechnischen Komponenten. Durch die Wandlungsprozesse gerade in den Mittel- und Niederspannungsnetzen können die bisherigen Planungs- und Betriebsweisen, zumindest mittel- und langfristig, nicht aufrechterhalten werden. Es werden zunehmend Mittel zur Netzüberwachung, -steuerung und -automatisierung auch in diesen Netzen benötigt. Eine direkte Übertragung sekundärtechnischer Komponenten aus den Netzen der Höchst- und Hochspannung erscheint nicht praktikabel, da die existierenden Systeme für einen zahlenmäßig großen Einsatz in den Mittel- und Niederspannungsnetzen tendenziell zu teuer und zu spezialisiert sind. Daher wird es erforderlich, die in den Mittel- und Niederspannungsnetzen benötigten Systeme und Funktionen zu identifizieren und detailliert zu spezifizieren, um zu technisch und ökonomisch optimalen Lösungen zu gelangen.

Im ersten Teil dieses Kapitels werden zunächst grundsätzliche Kriterien definiert, anhand derer sich Schutz-, Leit- und Automatisierungsfunktionen kategorisieren lassen. Es werden die charakteristischen Eigenschaften einer Reihe von Funktionen aufgezeigt und eine Kategorisierung anhand der zuvor definierten Kriterien vorgenommen. Hierbei liegt der Fokus insbesondere auf Funktionen, die für einen Einsatz in Mittel- und Niederspannungsverteilsnetzen von besonderem Interesse sind.

Ausgehend von den Kategorisierungskriterien und Funktionseigenschaften werden im zweiten Teil dieses Kapitels Anforderungen an die zugrundeliegenden Systeme wie an die Hardware und die Kommunikationsinfrastruktur definiert. Diese Anforderungen müssen erfüllt sein, um eine ordnungsgemäße Ausführung der jeweiligen Funktionen sicherzustellen.

Im dritten Teil dieses Kapitels werden schließlich ausgewählte schutz-, leit- und automatisierungstechnische Systeme und Lösungen hinsichtlich ihrer Eigenschaften und ihres Funktionsumfangs betrachtet, kategorisiert und bewertet. Bei diesen Systemen und Lösungen handelt es sich sowohl um kommerziell am Markt verfügbare Produkte als auch um prototypische Umsetzungen im Rahmen von Forschungs- und Entwicklungsarbeiten.

2.1 Kategorisierung

Sekundärtechnische Systeme und Funktionen können nach ihrer grundlegenden Art und Funktionsweise grob in folgende Kategorien eingeteilt werden:

- Passive Funktionen zum Messen, zum Monitoring und zur Überwachung
- Leittechnische Funktionen zur Fernsteuerung
- Schutzfunktionen zum Schutz der Netze und ihrer Komponenten gegen die Auswirkungen elektrischer Fehler und gegen Betriebszustände, die den sicheren Netzbetrieb gefährden
- Fehlerdetektion und Fehlerortung zur Eingrenzung aufgetretener Fehler, zur Unterstützung bei der Fehlerbeseitigung und der Wiederherstellung eines sicheren Netzbetriebs
- Regelungs- und Automatisierungsfunktionen zur Beeinflussung des Netzes hin zu einem sicheren und technisch sowie wirtschaftlich optimalen Betriebspunkt ohne oder nur mit zeitweisen manuellen Eingriffen

Abhängig von den Eigenschaften des Netzes wie Spannungsebene und Netzstruktur kommen diese Systeme und Funktionen in unterschiedlichem Maße und unterschiedlicher Ausprägung zum Einsatz. Gleichzeitig weisen sie gänzlich unterschiedliche Eigenschaften und Anforderungen auf. Daher ist es erforderlich, Kriterien zu definieren, über die diese Systeme und Funktionen charakterisiert und kategorisiert werden können. Hieraus lassen sich Anforderungen ableiten, die eine Systemumgebung erfüllen muss, um eine Ausführung der Systeme und Funktionen zu ermöglichen. Im Folgenden wird dementsprechend eine Reihe von Kategorisierungskriterien aufgestellt.

Ausführungsort

Funktionen und Systeme können danach kategorisiert werden, wo sie innerhalb des elektrischen Netzes verortet sind und wie weit ihre Sicht und Einflussnahme ist. Mögliche Ausführungsorte sind etwa Umspannanlagen (HS/MS), Verteilerstationen (MS), Ortsnetzstationen (MS/NS), Kabelverteilerschränke (NS) und Kundenanschlüsse (MS/NS). Klassische Schutzfunktionen wie der Überstromzeit- und Distanzschutz arbeiten lokal-autonom in Umspannanlagen und Verteilerstationen. Dabei erfolgt eine dedizierte Zuordnung des Schutzsystems zu einem zu schützenden Objekt, etwa zu einer Leitung oder einem Transformator. Um den Ausfall eines Schutzsystems zu kompensieren, können im Sinne einer Redundanz für ein zu schützendes Objekt auch mehrere unabhängige Schutzsysteme zuständig sein, etwa ein Distanzschutz, der durch einen Überstromzeitschutz als Rückfallebene ergänzt wird. Auch können Schutzsysteme, die jeweils ein Objekt schützen, sich im Verbund gegenseitig Reserve geben. Eine Anwendung wäre etwa eine Leitung, die in mehrere Abschnitte unterteilt ist und durch einen aus mehreren Schutzsystemen bestehenden Distanzschutz geschützt wird. Die Koordination der einzelnen Schutzsysteme erfolgt durch eine Impedanz- und Zeitstaffelung, eine Kommunikation untereinander wird nicht benötigt.

Eine Funktion wie die Netzzustandsschätzung wird dagegen klassischerweise zentral ausgeführt: Messwerte werden an im Netz verteilten Knoten erfasst und per Kommunikationsverbindung an einen zentralen Punkt wie die Netzleitstelle übertragen. Dort wird aus den Messwerten ein Gesamtnetzabbild berechnet.

Mit Blick auf die Automatisierung von Mittel- und Niederspannungsnetze kommen insbesondere dezentrale, verteilte Ansätze und Lösungen in den Fokus. Die zunehmende Integration leistungsstarker, stochastischer Einspeiser und Lasten führt zu einem Bedarf, auch diese Netze überwachen und steuernde Eingriffe vornehmen zu können. Durch die Notwendigkeit, die einzelnen Netzteilnehmer zu koordinieren und das Netz aktiv zu regeln, scheiden hierbei einerseits lokal-autonome Lösungen größtenteils aus. Aufgrund der großen elektrischen und räumlichen Netzausdehnung und der Vielzahl an Netzteilnehmern ist es andererseits auch nicht zielführend, alle Messwerte aus dem Netz erst in die Netzleitstelle zu übermitteln, von wo nach Ausführung entsprechender Algorithmen Steuerbefehle zurück in das Netz übermittelt werden. Ein solcher Ansatz hätte einen hohen Kommunikationsbedarf und große Reaktionszeiten zur Folge, außerdem würden Kommunikationsstörungen die Regelung des Netzes einschränken oder ganz ausfallen lassen. Dezentral verteilte Funktionen und Systeme stellen hier das Mittel der Wahl dar. Eine verteilte, koordinierte Spannungsregelung etwa, die auf einem im Netz installierten Automatisierungssystem abläuft, könnte Messwerte von Netzknoten und Netzteilnehmern in ihrem direkten

Einflussbereich erhalten und ebenso Steuerbefehle an diese Netzteilnehmer senden. Nur das Automatisierungssystem, nicht aber die unterlagerten Netzteilnehmer, kommuniziert mit der Netzleitstelle und übermittelt vorverdichtete, für die Netzleitstelle relevante Daten. Umgekehrt kann die Netzleitstelle Vorgaben für das betrachtete Netzsegment machen, die konkrete Umsetzung in Regeleinriffe erfolgt dezentral im Netzsegment. Solche dezentralen oder auch zentral-dezentralen Lösungen reduzieren den Kommunikationsbedarf und die Fehleranfälligkeit.

Abbildung 2.1 zeigt eine Kategorisierung von Funktionen und Systemen anhand ihres Ausführungsortes und die dazugehörigen Kommunikationsbeziehungen in Anlehnung an [13]. Beim rein lokalen Ansatz findet keine Kommunikation zwischen Systemen statt. So können die lokalen Regelungsaufgaben zwar mit geringster Reaktionszeit ausgeführt werden, allerdings wird mit Blick auf das Gesamtnetz nicht das beste Regelungsergebnis erzielt. Der zentrale Ansatz verfügt über einen einzelnen, zentralen Koordinator. Da diesem Koordinator die Zustandsgrößen sämtlicher lokaler Systeme vorliegen, kann er das optimale Regelungsergebnis bestimmen. Allerdings stellt dieser Ansatz hohe Ansprüche an die Kommunikation und versagt vollständig, sobald die Kommunikation ausfällt. Beim verteilten Ansatz koordinieren sich benachbarte Systeme untereinander und kommen so zu einem besseren Regelungsergebnis verglichen mit dem rein lokalen Ansatz. Beim dezentralen Ansatz schließlich wird das Netz in Zonen unterteilt. Lokale Koordinatoren können, ähnlich wie beim zentralen Ansatz, für die jeweilige Zone das optimale Regelungsergebnis bestimmen, gleichzeitig werden die Nachteile des zentralen Ansatzes vermieden.

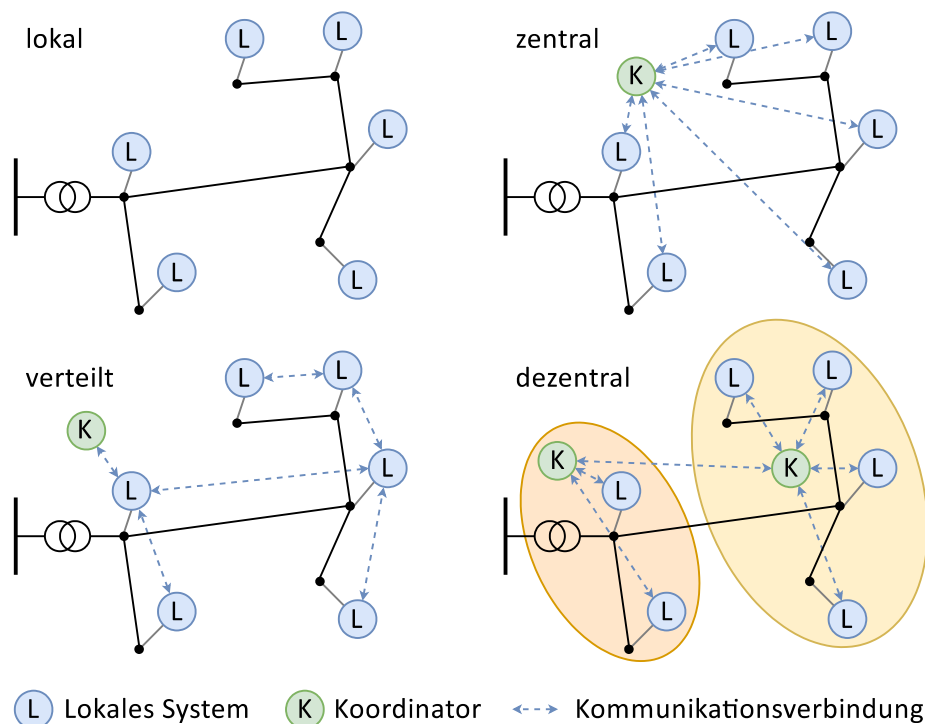


Abbildung 2.1: Kategorisierung anhand des Ausführungsortes (nach [13])

Zeitverhalten

Funktionen für den Einsatz in elektrischen Netzen können nach ihrem Zeitverhalten kategorisiert werden. Dabei kann mit Zeitverhalten einerseits die Anforderung beschrieben werden, wie schnell eine Funktion ablaufen muss, um eine rechtzeitige Reaktion auf die physikalischen Vorgänge im Netz sicherzustellen. Andererseits kann mit Zeitverhalten auch die Eigenschaft der Funktion beschrieben werden, wie lange diese für eine Ausführung benötigt.

An Schutzsysteme wird die Anforderung gestellt, innerhalb einer deterministischen Maximalzeit ein Fehlerereignis zu klären. Hierdurch wird sichergestellt, dass die Auswirkungen eines Fehlers zeitlich begrenzt bleiben und so große Schäden an Betriebsmitteln und Personenschäden vermieden werden. Die Gesamtzeit eines Schutzsystems ergibt sich als Summe unter anderem aus der Zeit, die eine Schutzfunktion zur Detektion des Fehlerereignisses benötigt, und der Zeit, die von der Auslösung des Leistungsschalters bis zum tatsächlichen Öffnen des primären Energiepfads verstreicht. Die maximal zulässige Fehlerklärungszeit hängt von der Spannungsebene und der Art der Schutzfunktion ab: Während für einen Distanzschutz in Hoch- und Höchstspannungsnetzen Zeiten von maximal 50-150 ms gefordert sind, dürfen die Zeiten in Mittelspannungsnetzen durchaus bei bis zu 1 s liegen [14, S. 33].

Für Regelungsfunktionen wie eine Spannungs- oder Blindleistungsregelung sind dagegen größere Reaktionszeiten zulässig. Ein Transformatorspannungsregler etwa soll gar nicht zu schnell auf jede kurzzeitige Spannungsabweichung reagieren. Jede Transformatorstufung ist mit mechanischen Belastungen und so auch mit Verschleiß verbunden, außerdem erzeugen zu schnelle Stufungsvorgänge eine „unruhige“ Spannung und beeinträchtigen damit die Netzqualität. Funktionen zur Transformatorspannungsregelung oder auch Blindleistungsregelung arbeiten daher eher mit Reaktionszeiten von einigen Sekunden bis hin zu Minuten. Dennoch müssen solche Funktionen deterministisch innerhalb der projizierten Zeiten reagieren.

Funktionen, die keine direkten Eingriffe auf das Netz oder seine Betriebsmittel vornehmen, erfordern oftmals kein streng deterministisches Zeitverhalten. Eine Monitoringfunktion wie die Netzzustandsschätzung dient dem Personal in der Netzleitstelle dazu, einen Überblick über die Verhältnisse im Netz zu erlangen. Hierbei sind zeitliche Abweichungen bei der Aktualisierung des dargestellten Netzabbilds zulässig. Auch bei Funktionen der Fehlerortung sind gewisse Verzögerungen zulässig. Während der elektrische Fehler durch eine Schutzfunktion innerhalb deterministischer Zeit geklärt wurde, dient die Fehlerortung dazu, den Fehler räumlich im elektrischen Netz zu lokalisieren. Da Netzumschaltungen infolge eines Fehlerereignisses oftmals noch manuell erfolgen, sind hier Verzögerungen im Minutenbereich zulässig.

Messwerte und Daten

Funktionen lassen sich nach ihrem Bedarf an Messwerten und Daten, die sie für ihre Ausführung benötigen, kategorisieren. Dabei ist einerseits relevant, welche Messwerte etwa in Form elektrischer Größen sie benötigen, aber auch, wie schnell und wie häufig diese Messwerte zur Verfügung stehen müssen. Eine Schutzfunktion wie der Überstromschutz benötigt in der einfachsten Ausführung lediglich den RMS-Wert des Stroms. Soll eine Richtungserkennung erfolgen, dann werden ebenso wie beim Distanzschutz die komplexen Ströme und Spannungen benötigt. Von der zeitlichen Auflösung können beide Schutzfunktionen mit Halb- oder Vollwellenwerten (10-ms- bzw. 20-ms-Werten) arbeiten. Um eine hohe Genauigkeit und Selektivität sicherzustellen, sollten die Messwerte mit einer hohen Anzahl an Quantisierungsstufen diskretisiert werden, z. B. mit 16 Bit. Im Gegensatz zu Regelungsfunktionen, die im fehlerfreien Betrieb des Netzes mit den Betriebsnennwerten arbeiten, ist es für Schutzfunktionen erforderlich, dass auch Messwerte im Bereich eines Vielfachen der Betriebsnennwerte exakt erfasst werden. Hierauf sind entsprechend unter anderem die Stromwandler und die Messwernerfassung auszulegen. Funktionen wie die Erdschlusserkennung nach dem Erdschlusswischerverfahren basieren auf transienten Vorgängen im Netz. Dementsprechend benötigen solche Funktionen die Netzgrößen als abgetastete Werte mit Abtastraten von z. B. 30 kHz. Im Gegensatz dazu genügen für Regelungsfunktionen wie eine Transformatorspannungsregelung Messwerte im Sekundenzyklus oder ggf. auch über mehrere Sekunden bis zu Minuten gemittelte Durchschnittswerte. Für Monitoringfunktionen wie eine Netzzustandsschätzung werden Messwerte vom Sekunden- bis in den Minutenbereich herangezogen. Eine klassische Offline-Netzzustandsschätzung kann mit 15-Minuten-Messwerten arbeiten. Damit die Schätzung aber auf demselben physikalischen Netzzustand basiert, ist eine Zeitsynchronisation oder zumindest Zeitstempelung der Messwerte erforderlich, so dass für einen Berechnungsdurchlauf Messwerte mit ähnlichem Erfassungszeitpunkt genutzt werden. Alternativ kann ein gleitender Mittelwert genutzt werden, um so insbesondere auch Ausreißer auszufiltern. Im Gegensatz dazu liefert eine Online-Netzzustandsschätzung den Netzzustand in höherer zeitlicher Rate, vom Minuten- bis in den Sekunden- oder gar Subsekundenbereich. Entsprechend zeitlich hochaufgelöst müssen die Messwerte erfasst und schnell kommuniziert werden.

2.2 Anforderungen

Aus den im vorherigen Kapitel vorgestellten Kategorisierungskriterien lassen sich Anforderungen ableiten, die eine Systemumgebung erfüllen muss, um eine Ausführung der Systeme und Funktionen zu ermöglichen.

Systemarchitektur

- Über welche Rechenleistung verfügt die Hardware der Systemarchitektur? Wie schnell können Funktionen ausgeführt werden? Wie viele Funktionen können gleichzeitig innerhalb der Systemarchitektur ausgeführt werden?
- Wie groß ist der Arbeitsspeicher der Hardware der Systemarchitektur? Dies ist insbesondere für Funktionen relevant, die auf umfangreichen Daten arbeiten wie z. B. eine Netzzustandsschätzung oder Optimierungsalgorithmen.
- Lässt die Systemarchitektur eine zeitdeterministische Funktionsausführung zu?
- Wie zuverlässig ist die Systemarchitektur? Gibt es Mechanismen der Redundanz?

Messwerterfassung

- Welche Messwerte (elektrische und ggf. auch nicht-elektrische) werden erfasst?
- Mit welcher Abtastrate und welcher Auflösung werden sie erfasst?
- Stehen nur RMS-Werte, Harmonische oder auch abgetastete Werte zur Verfügung?

Kommunikation

- Welche Art der Kommunikation ist möglich (horizontal zwischen gleichberechtigten Systemen und/oder vertikal zu über- bzw. untergeordneten Systemen)?
- Welche Kommunikationsmedien werden unterstützt? Beispiele: Mobilfunk, DSL (engl.: Digital Subscriber Line, Digitaler Teilnehmeranschluss), PLC (engl.: Powerline Communication, Datenkommunikation über das elektrische Energienetz), dedizierte Funklösungen wie CDMA450 [15]
- Welche Kommunikationsprotokolle werden unterstützt? Beispiele: IEC 60870-5-101/-104 [16], IEC 61850-8-1 MMS (engl.: Manufacturing Messaging Specification) und GOOSE (engl.: Generic Object Oriented Substation Events), IEC 61850-9-2 [17] SV (engl.: Sampled Values), Modbus TCP/RTU [18–20], MQTT [21], Smart-Meter-Protokolle
- Welche Datenraten sind erforderlich? Welche Datenmenge ist zu übertragen? Welche Latenz tritt auf? Wie zuverlässig ist das Kommunikationsmedium?

Implementierungs-, Engineering- und Konfigurationsprozesse

- Wie erfolgt die Umsetzung der Algorithmen? Ist eine freie Programmierbarkeit gegeben oder liefert die Systemarchitektur nur eine fest definierte Funktionalität?
- Existiert ein wohldefinierter Engineering- und Konfigurationsprozess, der Arbeitsschritte automatisiert und den Anwender leitet? Basiert dieser Prozess auf Standards oder auf proprietären Ansätzen und Werkzeugen?

Anhand der relevantesten Anforderungen erfolgt in Tabelle 2.1 eine Einordnung ausgewählter Funktionen.

Tabelle 2.1: Anforderungen ausgewählter Funktionen

| Funktion | Messwerte | Geschwindigkeit / Qualität Messwerte | Zeitverhalten | Kommunikation | Weiteres |
|----------------|---------------------------------|--|----------------------------------|--|---|
| Monitoring | Visualisierung | U, I, P, Q | 1-60 s | Leitstellen- anbindung | Zeitsynchronisation |
| | Power Quality | U, I | 25./50. Harmonische | | |
| | Online-SE | <u>U</u> , <u>I</u> | 100 ms-60 s | | |
| | Offline-SE | U, I, P, Q | 1-15 min | | |
| Topologie | Schalterstellung | 1 s--15 min | | | |
| Schutz | Überstromzeit- schutz | I oder <u>I</u> und <u>U</u> (Fehlerrichtung) | 10-20 ms | Deterministisch, Auslösezeit 300 ms | Leistungsschalter |
| | Distanzschutz | <u>I</u> und <u>U</u> | 10-20 ms | Deterministisch, Auslösezeit 100 ms | Leistungsschalter |
| | Differenzial- schutz | <u>I</u> oder i | 10-20 ms | Deterministisch, Auslösezeit 100 ms | Leistungsschalter, Zeitsynchronisation |
| | Frequenzschutz/ Lastabwurf | f | Genauigkeit 30 mHz | Deterministisch, Auslösezeit 150 ms | Leistungsschalter |
| | Konventionell | I oder <u>I</u> und <u>U</u> | 20 ms, ggf. 5./7. Harmonische | | |
| Fehleranzeiger | Ortung, multiple Messorte | <u>I</u> und <u>U</u> (Z-Bestimmung) | 10-20 ms | ggf. Fernauslesung | Zeitsynchronisation |
| | Erdschluss- wischer | i und u | 24 kHz | | |
| | Traforegelung | U | 1-10 s | | |
| Regelung | Q-/P-Regelung Flexibilitäten | U, I, P, Q | 100 ms-1 s | Anbindung an Flexibilität | |

2.3 Stand von Forschung und Technik

Netzbetreiber haben bereits damit begonnen, auf die Herausforderungen, denen sich die Verteilnetze gegenübergestellt sehen, sowohl mit konventionellen als auch mit neuartigen Maßnahmen zu reagieren. Zu den konventionellen Maßnahmen zählt etwa der Netzausbau, welcher jedoch sehr kostenintensiv und unflexibel ist. Innovative primärtechnische Maßnahmen bestehen im Einbau neuartiger Betriebsmittel wie etwa Längsreglern. Im Zusammenspiel mit sekundärtechnischen Maßnahmen bietet sich Potenzial, den sicheren und wirtschaftlichen Netzbetrieb auch zukünftig zu ermöglichen. Zu den sekundärtechnischen Maßnahmen zählen insbesondere die im vorherigen Kapitel dargestellten Funktionen. Diese können einerseits für sich allein betrachtet Aufgaben wie die Netzzustandsüberwachung oder die Fehlerortung erfüllen und so dem Netzbetreiber einen Mehrwert unabhängig von der Primärtechnik bieten. Andererseits können sie als Schutz-, Leit-, Regelungs- und Automatisierungsfunktionen aktiv primärtechnische Betriebsmittel beeinflussen, etwa als Spannungsregler eines Ortsnetztransformators (ONT). Die folgenden Betrachtungen beschränken sich auf sekundärtechnische Maßnahmen zur Umsetzung eines Smart Grids. Hierunter fallen prototypische Lösungen aus Forschungsprojekten sowie kommerzielle Produkte. Die Eigenschaften und Fähigkeiten der einzelnen Systeme werden beleuchtet und Vor- und Nachteile dargestellt. Grundlegende Recherchearbeiten hierzu fanden in [S12] statt.

DG DemoNet – Smart LV Grid

Im Forschungsprojekt *DG DemoNet – Smart LV Grid* [22, 23] (2011-2014, Österreich) wurde untersucht, wie sich die Integration von Photovoltaik und Elektromobilität in Niederspannungsnetze unter Vermeidung von Netzausbau und bei Einhaltung der zulässigen Spannungsgrenzen maximieren lässt. In vier Netzabschnitten fanden dabei Feldtests statt. Methodisch war das Forschungsprojekt dreigeteilt: Als *Smart Planning* wurde eine probabilistische Planungsmethode entwickelt, die auf Basis von im Netz erfassten Daten die Integrierbarkeit weiterer PV-Anlagen beurteilt. Als *Smart Monitoring* wurde betrachtet, wie unter Einsatz einer Smart-Meter-Infrastruktur Ortsnetze individuell überwacht und deren Nutzung optimiert werden können. Als *Smart Control* wurden Verfahren zur Spannungsregelung in Ortsnetzen unter Einsatz verschiedener Regelungskonzepte – von rein autonomen Spannungsreglern an Ortsnetztransformatoren bis hin zu koordinierten Regelungen mit Sollwertvorgabe an dezentrale Netzteilnehmer – evaluiert. Zur Kommunikation kamen PLC und Breitband-Internetanschlüsse zum Einsatz. Als besonders problematisch erwies sich der hohe manuelle Engineeringaufwand. Für zukünftige Automatisierungslösungen, besonders für Niederspannungsnetze, wurde daher die Notwendigkeit nach einfachen und automatisierten Engineeringprozessen betont.

Grid4EU

Im europäischen Verbundprojekt Grid4EU [24] (2011-2016) wurden in sechs Demonstrationsprojekten verschiedene Smart-Grid-Ansätze erprobt. Der Fokus des deutschen Demonstrationsprojekts [25–27] lag auf der Überwachung und Steuerung von Mittelspannungsnetzen mit dem Ziel, durch Topologieänderungen Netzengpässe zu kompensieren und Netzverluste zu reduzieren, die Wiederversorgungszeiten im Fehlerfall durch automatische Umschaltungen zu minimieren, die Integrierbarkeit dezentraler Einspeiser zu steigern und insgesamt die Beobachtbarkeit zu erhöhen. Für den Feldtest wurden in einem Netzbereich mit ca. 100 Stationen sieben Stationen mit Lasttrennschaltern samt Steuerung und Messung sowie elf Stationen mit Messung ausgestattet und außerdem Fehlerrichtungsanzeiger eingebunden. Als Steuerungs- bzw. Messmodule dienten RTUs (engl.: Remote Terminal Unit, Fernbedienungsterminal). Die im Netz erfassten Werte wurden über Mobilfunk an eine zentrale RTU kommuniziert, wo Netzoptimierungen in Form von Topologieänderungen berechnet und als Schaltfolge an die dezentralen RTUs übermittelt wurden. Bei Erkennung von Netzfehlern wurden ebenfalls Umschaltungen ausgelöst. Es konnte gezeigt werden, dass die Maßnahmen sowohl die Netzverluste als auch die Wiederversorgungszeiten reduzierten. Die eingesetzten Lasttrennschalter unterlagen allerdings einem hohen Verschleiß, da sie für die hohe Anzahl an Schalthandlungen nicht ausgelegt waren.

GridBox

GridBox ist eine Mess- und Steuerplattform für MS- und NS-Netze, die zunächst im gleichnamigen Forschungsprojekt [28–33] (2013-2016, Schweiz) entwickelt und dann durch die smart grid solutions AG vermarktet wurde. Nach Fusion mit der EVUtion AG [34] ist sie nun Teil der GRIDCONTROL-Plattform. Hard- und Software wurden von Grund auf entwickelt. Messung und Monitoring sollen Verteilnetze beobachtbar machen, die Ansteuerung von Flexibilitäten soll den Netzbetrieb optimieren. GridBox arbeitet hierarchisch: Das Basissystem erfasst Messwerte z. B. an ONT und Kabelverteilerschränken (KVS) und erfüllt PMU-Funktionalitäten (engl.: Phasor Measurement Unit, zeitsynchronisiertes Zeigermessgerät). Die Basissysteme übermitteln ihre Werte an ein übergeordnetes System, etwa in einer ONS. Diese hierarchische Struktur wiederholt sich im überlagerten Netz. Es sind Algorithmen zur Netztopologieerkennung und Netzparameterschätzung auf Basis der PMU-Messwerte verfügbar. Maßnahmen werden lokal ausgeführt, aber auch mit benachbarten und überlagerten Netzbereichen koordiniert. Das System setzt eine hohe Durchdringung des Netzes mit Mess- und Steuerhardware voraus. Als Kommunikationsmedium wurden Mobilfunk, Glasfaser und PLC evaluiert. Zur Kommunikation zwischen den Systemen dient ein eigenes Protokoll, eine direkte Anbindung von Drittkomponenten ist nicht möglich. Eine Praxiserprobung erfolgte im Rahmen zweier Pilotinstallationen.

GridCal

Bei GridCal handelt es sich um eine Lösung zum Monitoring von Ortsnetzstationen der PSInsight GmbH (Power System Insight) [35, 36], einer Ausgründung der Hochschule Düsseldorf [37]. Die Lösung verfolgt einen zellulären Ansatz, indem bereits einzelne ONS digitalisiert werden können, ohne direkt eine übergeordnete IT-Infrastruktur aufsetzen zu müssen. Als *GridCal Node* werden hierzu ONS mit Hardware ausgestattet, wobei die PSInsight GmbH am Markt verfügbare Standardkomponenten zu einem Gesamtsystem zusammenfügt. Zur Messung dient ein Power-Quality-Messgerät der PQ Plus GmbH, lokaler Controller ist eine SPS (Speicherprogrammierbare Steuerung) der WAGO GmbH & Co. KG. Zusätzliche Komponenten sind ein Mobilfunkmodul und Netzteil. Die Implementierung der Software und lokalen Logik erfolgt durch die PSInsight GmbH. Innerhalb der ONS werden kontinuierlich Messwerte erfasst, aufbereitet und archiviert. Durch lokale Logiken können einzelne Netzteilnehmer netzdienlich dezentral gesteuert werden. Relevante Messwerte werden bei Bedarf von den ONS abgerufen, um etwa für Netzberechnungen genutzt zu werden. Neben dem Netzbetrieb werden insbesondere auch Netzplanung und Asset-Management als Nutzer der Daten hervorgehoben. Die Softwarelösung *GridCal Operator* dient der zentralen Verwaltung und Überwachung aller digitalisierten ONS.

GridEye

GridEye ist ein Produkt des Schweizer Unternehmens DEPsys SA [38], das als DEPsys GmbH mittlerweile auch in Deutschland vertreten ist. GridEye [39] stellt eine Hard- und Softwarelösung zur Überwachung und Regelung von Niederspannungsnetzen dar. Das GridEye-Basismodul wird in ONS und KVS integriert und erfasst neben Effektivwerten auch Spannungs- und Stromphasoren [40] sowie Netzqualitätswerte. Da es nur eine einzige dreiphasige Strommessung besitzt, ist jedoch keine abgangsselektive Erfassung möglich. GridEye kann regelbare ONT und Flexibilitäten ansteuern. Das System besitzt Funktionen zur Netzzustandsermittlung, Netzregelung und Fehlererkennung. Die Netzzustandsermittlung nutzt Sensitivitätsanalysen, die die Spannungssensitivität einzelner Netzknoten in Abhängigkeit der Leistungen von Quellen und Lasten bestimmen. Die Algorithmen arbeiten dezentral auf den GridEye-Basismodulen. Algorithmen zur Fehlererkennung ermöglichen die Identifikation von Fehlerart und Fehlerrichtung. Über einen zentralen Server in der Netzleitstelle erfolgen die Visualisierung des Netzes sowie die Konfiguration der Basismodule. Das Basismodul besitzt ein integriertes Mobilfunkmodul zur Kommunikation mit anderen Basismodulen und dem zentralen Server sowie Ethernet- und RS485-Schnittstellen zur Anbindung dezentraler Flexibilitäten.

iNES, NiVeAu, Grid Commander

iNES (Intelligente Ortsnetzstation) [41, 42] startete zunächst als Forschungsprojekt (2011-2013) mit Fokus auf der Überwachung, Steuerung und Automatisierung von Niederspannungsnetzen. Diese wurden hierzu mit Komponenten ausgestattet [43]: Ein Steuerungssystem (sBox) in der Ortsnetzstation erfasste einerseits lokale Messwerte und sammelte andererseits Messwerte von im Netz verteilten Messsystemen (mBox) sowie an Netzteilnehmern installierten Aktoren (aBox) ein. Auf Basis dieser Messwerte führte die sBox eine Netzzustandsidentifikation aus, berücksichtigte hierbei durch u. a. eine phasenselektive Leistungsflussberechnung die besonderen Bedingungen im Niederspannungsnetz [44]. Zur Netzbeeinflussung konnte die sBox einen regelbaren ONT oder Längsregler ansteuern, aBoxen konnten auf steuerbare Netzteilnehmer einwirken. sBox und aBox basierten auf einer RTU der Bilfinger Mauell GmbH [45], bei der mBox handelte es sich um eine Messkarte. Die Kommunikation erfolgte über PLC, lokale Funklösungen, dedizierte Leitungen und Mobilfunk. Im Folgeprojekt NiVeAu [46] wurde das iNES-System auf die Mittelspannungsebene übertragen, im Projekt Grid Commander [47, 48] (2015-2018) schließlich wurden beide Lösungen im Sinne einer hierarchisch-koordinierten Überwachung und Regelung zusammengeführt. Das iNES-System wurde durch die SPIE SAG GmbH zwischenzeitlich weiterentwickelt und als Produkt vermarktet, ist mittlerweile jedoch eingestellt worden.

InGO

InGO (Intelligent Grid Operator, [49–51]) ist eine Plattform der PSI GridConnect GmbH zur Überwachung, Steuerung und Automatisierung von MS- und NS-Netzen. Je nach Netzebene und Anwendungsfall kommen unterschiedlich ausgeprägte, dezentral im Netz platzierte Geräte zum Einsatz (Mittelspannung: PSIngo/MV, Niederspannung: PSIngo/LV, Elektromobilität: PSIngo/EV, Microgrids: PSIngo/MG). Die Geräte basieren auf einer Weiterentwicklung der RTU-Serie *PGC-3000* (ehemals *Smart Telecontrol Unit*) [52]. Sie selbst besitzen keine direkte Messwerverfassung, sondern stützen sich auf kommunikationstechnisch angebundene Drittkomponenten. Verfügbare Funktionen sind unter anderem Netzzustandsschätzung und -prognose, Last-, Einspeise-, Speicher- und Lademanagement, Spannungs-, Weitbereichs- und Blindleistungsregelung sowie Topologieoptimierung. Jedes Gerät ist zunächst einmal für die autarke Überwachung und Steuerung des ihm zugeordneten Netzsegments bzw. -teilnehmers zuständig, mehrere Geräte können jedoch auch hierarchisch-koordiniert agieren. Das System kann assistierend, teilautonom oder vollautonom arbeiten. Weiterhin sind ein cloudbasiertes Monitoring- und Kontrollsystem (PSIngo/MC) sowie Management- und Patchsystem (PSIsiu) verfügbar. Die Kommunikation zwischen Geräten und zum Monitoring- und Kontrollsystem erfolgt über OPC-UA [53], Drittkomponenten können über Protokolle wie IEC 60870-5-104, IEC 61850 oder Modbus angebunden werden.

Quantum Edge device

Als Quantum Edge device (QEd) [54] wird ein Produkt des italienischen Unternehmens Gridspertise S.r.l., einem Tochterunternehmen der Enel Grids S.r.l. [55], beworben. QEd stellt eine multifunktionale Plattform für den Einsatz in Ortsnetzstationen dar. Durch eine rechenstarke Hardware und Funktionsvirtualisierung soll die dezentrale Ausführung (engl.: Edge Computing) verschiedenster Funktionen ermöglicht werden, so dass bisher notwendige Einzelkomponenten entfallen können. Das System dient der Überwachung, Steuerung und Automatisierung von Mittel- und Niederspannungsnetzen und dezentraler Netzkomponenten. Es ermöglicht die Messung der mittelspannungsseitigen Sammelschienenenspannung sowie der Ströme zweier Abgänge, jeweils über Kleinsignaleingänge. Als Schutzfunktionen für die Mittelspannung sind u. a. ein Überstromzeit-, Spannungs- und Frequenzschutz vorgesehen. Außerdem kann das System als Datenkonzentrator für Smart Meter arbeiten. Eine Kommunikation ist über Ethernet, Mobilfunk, WLAN und PLC möglich, zudem sind binäre Ein- und Ausgänge vorhanden. Weitergehende, detaillierte technische Spezifikationen sind gegenwärtig allerdings nicht verfügbar. Ebenso ist nicht eindeutig zu klären, ob QEd bereits als Produkt am Markt bezogen werden kann.

Smart Area Aachen / FIONA

Im Forschungsprojekt Smart Area Aachen [56] (2012-2016) wurden Aufbau und Betrieb von Smart Grids erforscht und erprobt. Es gliederte sich in die Teilprojekte *Intelligente Ortsnetzstationen (iONS)* [57], *Spannungsqualität*, *Kommunikationsinfrastruktur*, *Netzzustandsprognose*, *Stromnetzplanung* und *Instandhaltung*. iONS, auf das sich die nachfolgenden Betrachtungen beschränken, integrierte Funktionen zur Spannungsregelung und -beobachtung sowie Fehlererkennung und -lokalisierung in ONS. Als *FIONA* (Flexible intelligente Ortsnetzautomatisierung, [58]) wurde eine Hardware, bestehend u. a. aus einer Standard-RTU, Mobilfunkrouter und akkugepufferter Stromversorgung, entwickelt. Auf Basis niederspannungsseitiger Messwerte sowie einer Abschätzung der PV- bzw. Windkrafteinspeisung erfolgte die Ansteuerung eines Längsreglers oder die Stufung eines ONT. Ein Spannungsbeobachter konnte kritische Spannungswerte detektieren, ohne das Netz selbst mit Messtechnik ausstatten zu müssen. Mit den lokalen Messwerten in der ONS wurden hierzu über ein hinterlegtes, vereinfachtes Modell des Niederspannungsnetzes die dortigen Spannungen abgeschätzt. Dies ließ eine präzisere Spannungsregelung im Vergleich zu ausschließlich lokalen Messwerten zu. Weiter wurden eine Fehlererkennung und -lokalisierung integriert, allerdings über separate Kurzschluss- und Erdschlussrichtungsanzeiger. Deren Meldungen wurden in die Netzleitstelle übertragen, von wo umgekehrt über motorisch betätigte Mittelspannungsschaltgeräte Netzumschaltungen durchgeführt werden konnte, um so Unterbrechungszeiten zu verringern.

SMIGHT Grid2

SMIGHT Grid2 [59] ist ein Messsystem für die Niederspannung der SMIGHT GmbH, einer Ausgründung der EnBW AG [60]. Mit SMIGHT Grid2 können die Abgänge von Ortsnetzstationen und Kabelverteilerschränken ausgestattet werden. Zur Messung wurden besonders kompakte Stromsensoren entwickelt, die sich schnell und ohne Freischaltung installieren lassen. Jeder Sensor erfasst die Stromeffektivwerte sowie die Leistungsflussrichtungen aller vier Leiter eines Abgangs (L1-L3, N) und kommuniziert diese minütlich per Funk an ein in der ONS bzw. im KVS installiertes Gateway. Die Messwerterfassung geschieht dabei zeitsynchron an allen Messstellen im Netz. Ihre Hilfsenergie beziehen die Sensoren direkt aus den Messwandlern, so dass keine zusätzliche Verkabelung erforderlich ist. Das Gateway selbst misst einphasig die Sammelschienenspannung, speichert und aggregiert alle Messwerte und kommuniziert diese als 15-Minutenwerte über Mobilfunk an eine Cloud-Plattform. Von dort lassen sie sich über ein Web-Portal abrufen und visualisieren. Auch ein Fernupdate der Systeme ist vorgesehen. SMIGHT Grid2 versteht sich somit als Komplettpaket zur einfachen und schnellen messtechnischen Erschließung von Niederspannungsnetzen.

SynchroGuard

Bei SynchroGuard [61] handelt es sich um ein Echtzeitüberwachungssystem für elektrische Verteilnetze der Zaphiro Technologies SA [62], einem Spin-off der EPFL (École Polytechnique Fédérale de Lausanne, Schweiz). Das System basiert auf im Netz verteilten Micro-PMUs (μ PMU), die als SynchroSense vermarktet werden. Auf Basis der von den μ PMUs erfassten Phasoren wird auf einem zentralen System (SynchroHub) eine schnelle Netzzustandsschätzung (bis hinunter zu 20 ms) ausgeführt. Hierdurch können Ereignisse im Netz sehr schnell visualisiert werden. Weitere Algorithmen ermöglichen die Onlinedetektion und -lokalisierung von Fehlerereignissen, wobei verschiedene Netzformen abgedeckt und dauerhafte als auch intermittierende Fehler erkannt werden [63]. Neben der reinen Netzüberwachung ist eine Echtzeitregelung von dezentralen Einspeisern zwecks Spannungsregelung und Engpassmanagements verfügbar. SynchroSense [64] bietet neben der PMU-Funktionalität auch Power-Quality-Messungen sowie Störschreiberfunktionen. Ein SynchroSense-System hat bis zu fünfzehn Messkanäle, die mit hoher Auflösung und Abtastrate erfasst werden. Zur Kommunikation sind neben Ethernet optional eine optische Schnittstelle sowie ein integriertes Mobilfunkmodul verfügbar. Das SynchroGuard-System kann über Standardprotokolle in bestehende Systeme integriert werden. Die Systeme und Algorithmen wurden bereits vor der Firmengründung durch einen Feldtest auf dem EPFL-Campus erprobt [65]. Als Produkt ist das System in Pilotinstallationen in der Schweiz und in Hongkong im Einsatz.

Gesamtübersicht

Tabelle 2.2 stellt die zuvor betrachteten Smart-Grid-Systeme vergleichend gegenüber.

Tabelle 2.2: Vergleich der Smart-Grid-Systeme

| System | DG DemoNet | Grid4EU | GridBox | GridCal | GridEye | iNES | InGO | Quantum Edge device | Smart Area Aachen | SMIGHT Grid2 | SynchroGuard |
|--|------------|---------|---------|---------|---------|------|------|---------------------|-------------------|--------------|--------------|
| Eigenschaften | | | | | | | | | | | |
| Monitoring, Messung | + | + | +++ | +++ | +++ | ++ | ++ | ++ | ++ | ++ | +++ |
| Messhardware integriert | - | + | +++ | +++ | +++ | +++ | - | ++ | +++ | +++ | +++ |
| Power-Quality-Messung | - | - | +++ | +++ | +++ | - | - | ++ | - | - | +++ |
| PMU-Funktionalität | - | - | +++ | - | +++ | - | - | - | - | - | +++ |
| Netzzustandsbestimmung | - | - | ++ | - | +++ | ++ | +++ | - | - | - | +++ |
| Fernsteuerung | - | + | + | - | - | - | ++ | ++ | - | - | - |
| Regelung | ++ | ++ | ++ | ++ | +++ | +++ | +++ | + | ++ | - | + |
| Spannungsregelung | +++ | - | - | - | +++ | +++ | +++ | - | +++ | - | - |
| Netzoptimierung | - | +++ | +++ | + | ++ | +++ | +++ | ++ | - | - | + |
| Fehlerdetektion/-lokalisierung | - | ++ | - | - | + | - | - | - | ++ | - | +++ |
| Schutzfunktionen | - | - | - | - | - | - | - | ++ | - | - | - |
| Nicht-technische Funktionen¹ | - | - | - | ++ | - | - | - | - | - | - | - |
| Dezentrale/zentrale Algorithmen | D | DZ | DZ | D | D | D | DZ | D | D | Z | Z |
| Funktionsbündelung² | - | - | + | - | + | - | ++ | +++ | - | - | - |
| Spannungsebenenübergreifend | - | + | ++ | - | - | ++ | ++ | ++ | - | - | + |
| Kommunikation | + | + | + | + | ++ | + | +++ | +++ | + | + | + |
| Leitstellenanbindung | + | + | ++ | ++ | + | + | ++ | ++ | ++ | ++ | + |
| Engineering/Konfiguration | - | - | + | + | + | + | + | + | - | + | - |
| Validierung im Feld | +++ | + | ++ | +++ | + | ++ | +++ | - | + | +++ | + |
| Bewertung: - nicht erfüllt, + grundlegend erfüllt, ++ teilweise erfüllt, +++ vollständig erfüllt | | | | | | | | | | | |

¹ Z. B. Verwaltung betrieblicher Daten, Unterstützung von Netzplanung/Assetmanagement

² Ein modulares System integriert alle Funktionen.

2.4 Zusammenfassung

In diesem Kapitel wurde eine grundlegende Übersicht über Funktionen und Systeme zur Überwachung, Steuerung, Automatisierung und zum Schutz insbesondere von Verteilnetzen gegeben. Zunächst wurden Kriterien definiert, die eine Kategorisierung solcher Funktionen und Systeme ermöglichen. Die Kernkriterien waren hierbei der Ausführungsort, das Zeitverhalten und die benötigten Messwerte und Daten. Diese Kriterien wurden beispielhaft an Funktionen erläutert. Auf dieser Basis wurden Anforderungen definiert, anhand derer geprüft und bewertet werden kann, ob eine Ausführungsumgebung wie eine Automatisierungshardware oder Kommunikationsinfrastruktur geeignet ist, konkrete Funktionen ordnungsgemäß auszuführen. So muss eine Systemarchitektur genügend Rechenleistung und Datenspeicher bieten und die benötigten Messwerte in geeigneter Art und Geschwindigkeit zur Verfügung stellen. Bei der konkreten Umsetzung von Funktionen und Systemen in elektrischen Netzen ergibt sich die Frage, inwiefern Implementierungs-, Engineering- und Konfigurationsprozesse definiert und verfügbar sind und wie praktikabel deren Anwendung ist.

Die Betrachtung von am Markt verfügbaren Systemen und Lösungsansätzen aus der Forschung zeigte, dass zwar einzelne Probleme und Herausforderungen gelöst werden, aber dennoch immer auch eine Reihe von Nachteilen bestehen bleibt. So werden oftmals lediglich einzelne oder eine vordefinierte Anzahl und Art von Funktionen geboten. Das Hinzufügen fremder oder vom Anwender entworfener Funktionen oder Algorithmen ist nicht vorgesehen. Zur Erfüllung einer Gesamtaufgabe müssen so multiple Komponenten zusammengefügt werden, etwa in einer Ortsnetzstation mit getrennten Geräten für die Messwernerfassung, Fehlerrichtungserkennung, lokale Logiken und Kommunikation. Dies führt zu komplexen und teuren Gesamtsystemen. Die Dynamik im Zubau neuer Netzkomponenten erfordert dagegen zunehmend einen flexiblen Einsatz verschiedener Funktionen an unterschiedlichen Positionen im Netz. Die hierzu nötige Flexibilität, Funktionen frei auf Geräte verteilen und bei Bedarf auch erweitern und verschieben zu können, fehlt bestehenden Lösungen. Herkömmliche Engineering- und Konfigurationsprozesse sind für das flächendeckende Ausrollen einer großen Anzahl von Systemen, wie es die Transformation der Verteilnetze hin zu Smart Grids erfordert, nicht geeignet. Die bisherigen Prozesse umfassen viele einzelne, unabhängige und teils manuelle Arbeitsschritte mit oftmals herstellerspezifischen Werkzeugen, denen Schnittstellen zum Datenaustausch fehlen. Für eine effiziente Verteilnetzprojektierung müssen diese Schritte verzahnt und automatisiert werden.

Insgesamt besteht noch erheblicher Optimierungsbedarf auf Seiten der Smart-Grid-Automatisierungssysteme, um die Herausforderungen, mit denen sich die elektrischen Verteilnetze zunehmend konfrontiert sehen, erfolgreich meistern zu können.

3 Engineering und Projektierung elektrischer Energieanlagen

Bei der Erneuerung oder Erweiterung elektrischer Energieversorgungsnetze steht der Netzbetreiber vor der Aufgabe, mehrere, teils konkurrierende Teilziele in Einklang zu bringen: Das Zielnetz muss die Versorgungsaufgabe technisch und wirtschaftlich optimal erfüllen und gleichzeitig für zukünftige Entwicklungen gerüstet sein, um den hohen Investitionskosten und langen Lebensdauern der Netzkomponenten sowie den umfangreichen Planungsverfahren gerecht zu werden. Sicherheitsreserven garantieren die Zuverlässigkeit beispielsweise im Falle des Ausfalls einzelner Komponenten, erhöhen jedoch die Errichtungskosten. Eine Standardisierung der Netzstrukturen verringert den betrieblichen Aufwand und damit die Betriebskosten. Daneben müssen weitere Aspekte wie bestehende Netzstrukturen, gesetzliche und regulatorische Rahmenbedingungen, interne Standards, die Umweltverträglichkeit sowie Anforderungen besonderer Kunden berücksichtigt werden.

Im ersten Schritt fokussiert sich die Netzplanung auf die Primärtechnik. Es werden mögliche Planungsvarianten erstellt, diese auf ihre netztechnische Eignung hin geprüft und die Kosten für Errichtung und Betrieb kalkuliert. Zur Prüfung der netztechnischen Eignung werden in der Regel Modelle in Netzberechnungswerkzeugen erstellt und diese verschiedenen Simulationen wie Lastfluss- und Kurzschlussrechnungen unterzogen. Weiterhin müssen insbesondere jedoch auch planungsrechtliche und regulatorische Rahmenbedingungen berücksichtigt und entsprechende Antrags- und Genehmigungsprozesse eingeplant werden, da diese je nach Umfang der Baumaßnahme erheblichen Einfluss auf Aufwand und Kosten haben. Unter Berücksichtigung aller Eingangsgrößen werden die Planungsvarianten verglichen und es wird die technisch und wirtschaftlich bestmögliche Variante ausgewählt. Mit der Planungsvariante ist gleichfalls die Auswahl der notwendigen primärtechnischen Komponenten bezüglich Art und Dimensionierung getroffen. Aus dem Produktkatalog verschiedener Hersteller und Lieferanten müssen nun geeignete primärtechnische Komponenten gewählt werden. Weiterhin sind Art und Umfang weiterer Gewerke, etwa für Kabellegearbeiten und Anlagenerrichtung, zu definieren.

Mit den Vorgaben aus der Primärtechnik beginnt die Planung der Sekundärtechnik. Die Fachleute aus dem Bereich der Leit- und Nachrichtentechnik bestimmen den Signalumfang der Primär- und Sekundärtechnik, wählen entsprechend geeignete Leit- und Fernwirksysteme aus und planen notwendige Kommunikationsstrecken. Oftmals traditionell als eigener Bereich angesiedelt, analysieren die Fachleute der Schutztechnik das geplante Netz auf seinen Bedarf an schutztechnischen Komponenten. Netzstrukturen, Netzbetriebsphilosophien und Planungsgrundsätze geben entsprechende Randbedingungen vor. Zusammen mit den Fachleuten aus dem

Bereich der Leit- und Nachrichtentechnik werden Schnittstellen zwischen Schutz- und Leittechnik abgestimmt. Kommt beispielsweise physisch getrennte Schutz- und Leittechnik zum Einsatz, wie dies in Netzen der Höchst- und Hochspannung in der Regel der Fall ist, so handelt es sich bei den Schnittstellen um Kommunikationsschnittstellen. Bei kombinierter Schutz- und Leittechnik (Kombigeräte), wie sie häufig in Mittelspannungsnetzen genutzt wird, liegen die Schnittstellen auf Geräteebene, d. h. es müssen Systeme ausgewählt werden, die die Anforderungen sowohl der Leit- als auch der Schutztechnik erfüllen.

Abbildung 3.1 fasst die vorherigen Aussagen in einer grundsätzlichen Übersicht zusammen, konzentriert sich hierbei allerdings auf die technischen Aspekte.

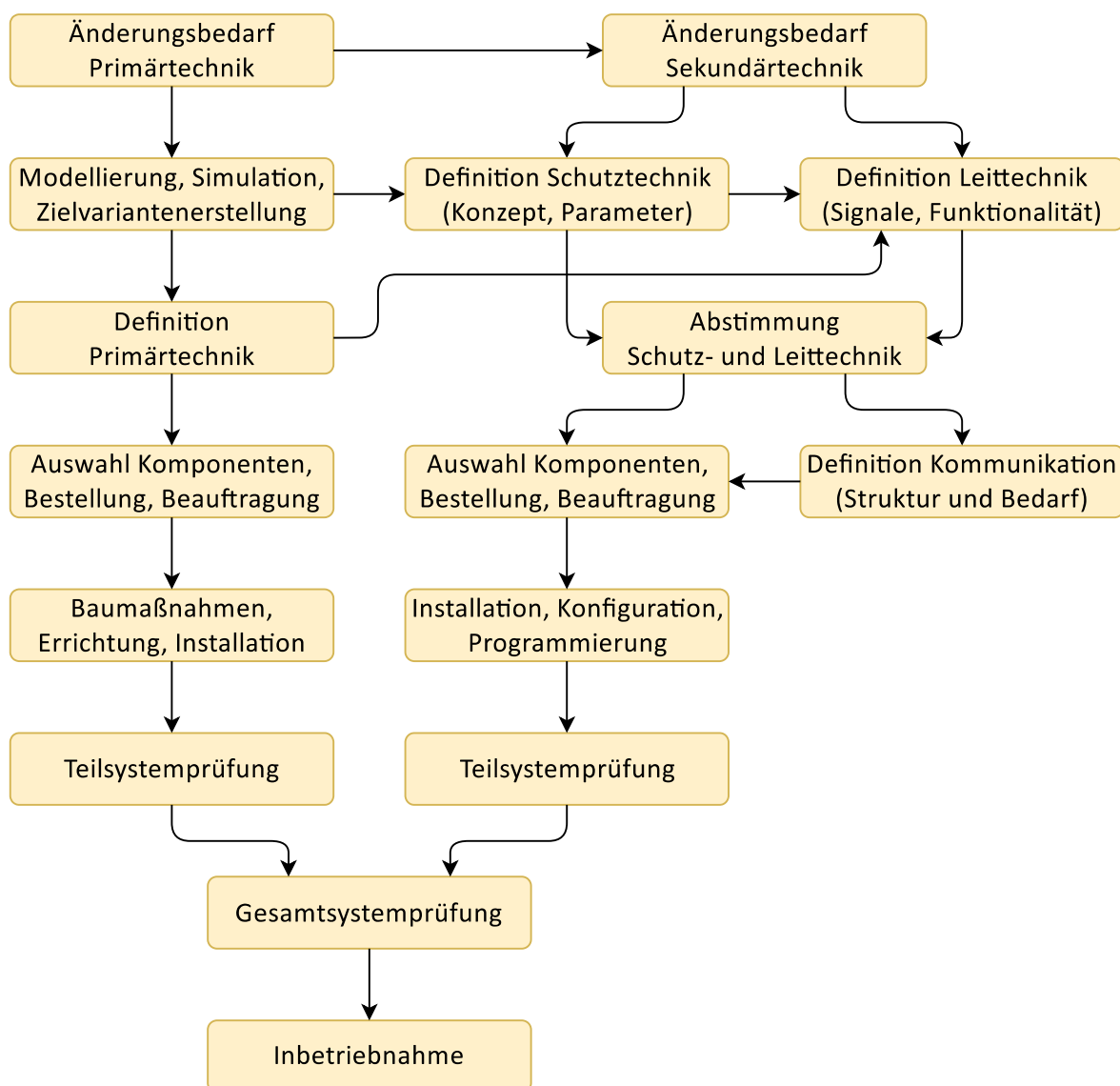


Abbildung 3.1: Engineering und Projektierung elektrischer Energieanlagen

In welcher Tiefe und mit welchem personellen Aufwand ein Netzbetreiber insbesondere die Planung der Sekundärtechnik durchführt, ist sicherlich maßgeblich von der Größe des Unternehmens abhängig. Auf der einen Seite stehen beispielsweise

kleine Stadtwerke, die die Planung dem Lieferanten der Schutz- und Leittechnik überlassen und schlüsselfertige Anlagen erhalten. Entsprechend groß muss das Vertrauen in den Lieferanten sein und ebenso groß ist die Abhängigkeit von diesem Lieferanten. Auf der anderen Seite stehen große Transport- und Verteilnetzbetreiber, die die Planung aller Fachbereiche in eigenen Händen halten möchten und dafür eine entsprechend große Personaldecke mit zahlreichen Spezialisten bereithalten müssen. So vielfältig Netzbetreiber sind, so vielfältig waren die in der Vergangenheit gelebten Engineering- und Projektierungsprozesse.

3.1 Bisherige Engineering- und Projektierungsprozesse

Die Umsetzung primär- und sekundärtechnischer Anpassungen in den Netzen erfordert immer eine Reihe organisatorischer, technischer und betrieblicher Schritte. Es müssen Netze analysiert, Anpassungsbedarfe lokalisiert und klassifiziert, Umbaumaßnahmen geplant, Anlagen und Geräte konfiguriert und die Änderungen dokumentiert werden. Auch wenn die grundlegenden Schritte hierzu von allgemeiner Natur sind, so haben sich bei Netzbetreibern aus der gelebten Historie heraus sehr heterogene und unternehmensspezifische Engineering- und Projektierungsprozesse herausgebildet, die darüber hinaus teils auch noch von der eingesetzten Gerätetechnik abhängig waren und damit eine Herstellerprägung aufwiesen. Dementsprechend kam eine Vielfalt verschiedener Planungs-, Dokumentations- und Konfigurationswerkzeuge zum Einsatz, zwischen denen keine Daten oder wenn, dann nur mit Medienbruch ausgetauscht wurden.

Insbesondere bei Übertragungsnetzbetreibern und großen Verteilnetzbetreibern kamen auch schon in der Vergangenheit ausgefeilte Engineering- und Projektierungsprozesse zum Einsatz. Durch die Unternehmensgröße bestand die Möglichkeit, das erforderliche Fachpersonal im Hause vorzuhalten. Außerdem erforderte es die Bedeutung des Netzes für die überregionale Versorgungssicherheit, dass ein solcher Netzbetreiber selbst über eine große Fachexpertise verfügte. Bei Betreibern von (kleinen) Mittel- und besonders auch von Niederspannungsnetzen fanden dagegen sehr unterschiedliche und fragmentierte Planungsverfahren Anwendung. Das grundsätzliche Problem, dass verschiedene Softwarewerkzeuge und eine Vielzahl von Medien³ eingesetzt werden, gibt es jedoch in allen Spannungsebenen. Daher existieren Bestrebungen, Prozesse zu vereinheitlichen, übergreifende Werkzeuge einzuführen und kompatible Schnittstellen zu schaffen. Insbesondere mit Blick auf die Digitalisierung von Mittel- und Niederspannungsnetzen

³ Medien bezeichnen hier verschiedene Dokumentenarten wie Tabellendokumente (z. B. Microsoft Excel), Textdokumente (z. B. Microsoft Word), PDF, proprietäre Dateiformate, Papierausdrucke etc.

im Sinne eines Smart Grids sind bisherige, manuelle und fragmentierte Engineering- und Projektierungsprozesse allein durch die Anzahl der Netze nicht mehr praktikabel.

Aus der Erfahrung des Autors sollen nachfolgend die bisherigen Engineering- und Projektierungsprozesse samt der daran beteiligten (Software-) Werkzeuge und Datenbestände sowohl eines Transportbetreibers als auch eines Verteilnetzbetreibers exemplarisch skizziert werden, um ein Verständnis für die Problematik zu schaffen. Um den Fokus dieser Dissertation zu wahren, konzentrieren sich die Betrachtungen dabei auf die Sekundärtechnik, lediglich bei direkten Berührungspunkten zur Primärtechnik wird auf diese in kurzer Form eingegangen.

3.1.1 Anwendungsbeispiel Transportnetzbetreiber

Der hier betrachtete Transportnetzbetreiber führt das Engineering und die Projektierung seiner Primär- und Sekundärtechnik im eigenen Haus mit großem Detaillierungsgrad aus und verfügt hierzu über zahlreiche Spezialisten in verschiedenen Fachabteilungen. Insbesondere findet bereits seit langer Zeit eine strikte Trennung zwischen Grundlagenplanung und projektspezifischer Planung statt: In der Fachabteilung der Grundlagenplanung werden Anlagen und Komponenten wie zum Beispiel Schaltfelder projektübergreifend standardisiert und liegen anschließend in einer Typbibliothek vor. Bei der konkreten Projektierung etwa einer Umspannanlage greift die Fachabteilung der Projektplanung auf diese Typbibliothek zu und erstellt unter Nutzung der vordefinierten Bausteine konkrete Anlagen. Als Vorteile dieses Vorgehens sind insbesondere die unternehmensweite und anlagenübergreifende Harmonisierung und die Reduzierung der Typenvielfalt zu nennen.

Zur Erstellung, Bearbeitung und Verwaltung der Engineering- und Projektierungsdaten kommen verschiedene, oftmals auf einzelne Aspekte beschränkte Softwarewerkzeuge zum Einsatz. Die Ursache hierfür liegt darin begründet, dass diese Werkzeuge und Datenbestände über Jahre und Jahrzehnte in einzelnen Fachabteilungen gewachsen sind und durch verschiedene Mitarbeiter unabhängig voneinander entwickelt und gepflegt wurden. In der Zusammenschau bilden diese Werkzeuge zwar bereits einen Großteil der erforderlichen Engineeringdaten ab. Allerdings weisen die Werkzeuge untereinander wenige bis keine Schnittstellen auf. Ein konsistenter und durchgängiger Datenfluss innerhalb des Engineerings und der Projektierung ist damit nicht gegeben. Vielmehr liegt es in der Verantwortung des Anwenders, Konsistenz und Aktualität der einzelnen Datenbestände sicherzustellen.

Im Nachfolgenden werden die Engineeringschritte, Werkzeuge und Datenbestände vorgestellt [66, 67]. Hierbei wird zwischen den eingangs erwähnten Rollen *Grundlagenplanung* und *projektspezifischer Planung* unterschieden.

Grundlagenplanung

Datenbanksystem (unternehmenseigene Entwicklung)

- **Schaltfeldtyp:** Innerhalb eines Datenbanksystems, bei dem es sich um eine unternehmenseigene Entwicklung handelt, erfolgt die Typisierung von Schaltfeldern. Hierbei werden unter anderem folgende Daten festgelegt:
 - Typ des Schaltfeldes z. B. Leitungsfeld, Transformatorfeld, Erdungsfeld
 - Anzahl und Art der Sammelschienen, für die das Schaltfeld ausgelegt ist
 - Anzahl und Art der Schaltgeräte z. B. Leistungsschalter, Trennschalter
 - Zuordnung möglicher einpoliger Ersatzschaltbilder
 - Art der Messung (Normierung der Spannungs- und Stromwandler)
 - Zuordnung leittechnischer Funktionen
 - Zuordnung schutztechnischer Funktionen

Zu beachten ist, dass die Schaltgeräte und schutz- und leittechnischen Funktionen eine Obermenge darstellen und somit teilweise optional sind. Erst im Rahmen der Projektierung einer Anlage erfolgt eine Auswahl, welche dieser optionalen Schaltgeräte und Funktionen konkret zum Einsatz kommen.

Grafik (Microsoft Excel)

- **Schaltfeldbezogene einpolige Ersatzschaltbilder:** Es wird die primärtechnische Struktur von Schaltfeldern definiert. Hierzu werden schaltfeldbezogene einpolige Ersatzschaltbilder (ESB) als Zeichnung innerhalb von Microsoft Excel erstellt. Ebenfalls definiert werden je ESB die eingesetzten Schaltgerätetypen. Zu beachten ist, dass die Anzahl der Schaltgeräte eine Maximalmenge darstellt. Im Rahmen der Projektierung können einzelne, in der konkreten Anlage nicht vorhandene Schaltgeräte abgewählt werden. Durch dieses Vorgehen kann die Variantenvielfalt an ESB reduziert werden. Zu beachten ist außerdem, dass für einen Schaltfeldtyp ein oder mehrere ESB zulässig sein können. Die Zuordnung eines konkreten ESB zu einem Schaltfeld erfolgt im Rahmen der Projektierung.

Tabellarische Daten (Microsoft Excel)

Verschiedene Informationen werden als tabellarische Daten in Microsoft Excel gepflegt:

- **Adressierung elektrischer Signale:** Es werden Umfang und Belegung elektrischer Signale der Sekundärtechnik festgelegt. Die Signale sind den in den Schaltfeldtypen definierten leittechnischen Funktionen zugeordnet und fließen direkt ein in den elektrischen Signalumfang sowohl des Feldleitgeräts (Anzahl und Art der Ein- und Ausgänge) als auch des feldbezogenen Schaltschranks (Verdrahtung).

- **Adressierung kommunikationstechnischer Signale:** Bei der bisher standardmäßig eingesetzten Kommunikation gemäß IEC 60870-5-101/104 handelt es sich um signalorientierte Protokolle. Der Standard definiert lediglich die Semantik von Telegrammtypen (z. B. Meldung, Befehl oder Messwert), nicht jedoch die Semantik von Telegrammadressen. Unternehmensspezifisch erfolgt daher an dieser Stelle die Festlegung der Semantik der Telegrammadressen projektunabhängig und geräte- sowie funktionsbezogen, d. h. in verschiedenen Anlagen haben gleiche Funktionen und Geräte (z. B. Leistungsschalter) identische Adressbestandteile. Umgekehrt kann somit aus einer gegebenen Telegrammadresse direkt auf die vom Signal beschriebene Funktion bzw. das beschriebene Gerät geschlossen werden.
- **Namen kommunikationstechnischer Signale:** Hier sind den einzelnen Signaladressen die zugehörigen Klartexte (Meldungs- und Befehlsnamen) zugeordnet, z. B. „Leistungsschalter geschlossen“.
- **Netzwerkconfiguration:** Auch das IP-Adressschema ist unternehmensspezifisch festgelegt. Jede Anlage stellt ein eigenes IP-Teilnetz mit festgelegtem Netzanteil der IP-Adresse dar. Innerhalb einer Anlage gibt es ebenfalls Vorgaben zur Zuordnung des Hostanteils der IP-Adresse zu Komponenten etwa anhand der Spannungsebene, des Feldes und der Funktion.

Textuelle Daten (Microsoft Word)

Weitere Informationen werden auch rein textuell in Microsoft Word beschrieben:

- **Verriegelungsbedingungen:** Ob eine Schalthandlung an einem Schaltgerät zulässig ist, hängt ab vom Schaltzustand weiterer Schaltgeräte innerhalb eines Schaltfeldes als auch in anderen Schaltfeldern. Beispielsweise darf ein Erdungstrennschalter nur zugeschaltet werden, wenn die Trennschalter Richtung Sammelschiene und Leitung geöffnet sind, das Feld also definitiv spannungsfrei ist. Die Verriegelungsbedingungen stellen umfangreiche boolesche Regeln dar, die durch die Feldleitgeräte berücksichtigt werden müssen und liegen in textuell beschreibender Form vor.

Technische Zeichnungen (CAD-Werkzeuge)

Konstruktions- und Fertigungsunterlagen, die der Leittechniklieferant zur Fertigung der entsprechenden Komponenten benötigt, wie etwa mechanische Aufbaupläne und Stromlaufpläne, werden in verschiedenen Werkzeugen erstellt und gepflegt.

Projektspezifische Planung

Datenbanksystem (unternehmenseigene Entwicklung)

Die projektspezifische Planung der Sekundärtechnik erfolgt größtenteils innerhalb des bereits genannten Datenbanksystems:

- Zunächst wird die primärtechnische Struktur der Schaltanlage, hierarchisch gegliedert in Spannungsebenen und untergeordnete Schaltfelder, modelliert.
- Die Typisierung eines Schaltfeldes erfolgt durch Auswahl eines in der Grundlagenplanung definierten Schaltfeldtyps.
- Die primärtechnische Struktur des Schaltfeldes wird durch die Auswahl eines in der Grundlagenplanung definierten einpoligen Ersatzschaltbildes festgelegt. Da die in einem ESB definierten Schaltgeräte eine Obermenge darstellen, können hier im konkreten Feld nicht vorhandene Schaltgeräte abgewählt werden.
- Ebenfalls erfolgt an dieser Stelle die feldspezifische Auswahl der optionalen schutz- und leittechnischen Funktionen des Schaltfeldtyps.
- Zusätzlich zu der feldbezogenen Konfiguration können darüber hinaus spannungsebenen- oder anlagenbezogene Komponenten und Parameter konfiguriert werden.

Zu beachten ist, dass das Resultat der projektspezifischen Planung einer Anlage bisher lediglich zum Zwecke der Ausschreibung der entsprechenden Komponenten genutzt wird. Hierzu werden die Daten in eine textuelle Darstellung exportiert und anschließend als PDF-Dokument (engl.: Portable Document Format, Dateiformat) oder Papierausdruck an Hersteller und Lieferanten weitergegeben. Eine direkte Nutzung der Daten zur Konfiguration und Parametrierung der Komponenten erfolgt nicht.

Zusammenfassung und Bewertung

Zwischen Transportnetzbetreiber und den Lieferanten sekundärtechnischer Komponenten und Anlagen erfolgt ein kontinuierlicher Austausch bzgl. der unternehmensinternen Standardisierung. Insbesondere wird von den Lieferanten gefordert, unternehmensspezifisch konfigurierte Systeme liefern zu können. Wird beispielsweise ein Schaltfeldtyp neu erstellt oder maßgeblich modifiziert, so werden immer auch die Lieferanten mit der Fertigung und Lieferung von Prototypen nach neuem Standard beauftragt. Im Hause des Transportnetzbetreibers werden die neuen Systeme anschließend im Laborumfeld erprobt und auf Einhaltung der Anforderungen hin untersucht.

Bei der Beauftragung von Komponenten ist es daher nicht mehr erforderlich, diese wiederholt im Detail zu spezifizieren. Stattdessen können Komponenten direkt gemäß einmal durchgeführter Spezifikation angefordert werden.

Der Engineering- und Projektierungsprozess weist folgende **Vorteile** auf:

- Strikte Trennung zwischen projektübergreifender Grundlagenplanung und projektbezogener Anlagenplanung
- Standardisierung und Typisierung von Anlagen und Komponenten
- Wiederverwendbarkeit und Mehrfachnutzung von Engineeringdaten und Komponententypen. Dies führt zu einer Reduzierung des Engineeringaufwands und damit zu Zeitersparnis. Gleichzeitig werden Fehlerquellen verringert, da in der Grundlagenplanung definierte Standards über viele Projekte hinweg verwendet, ausgiebig geprüft und ggf. fehlerbereinigt werden.
- Durch eine detaillierte Spezifikation auf Ebene der Kommunikationsprotokolle ist eine Austauschbarkeit von Geräten verschiedener Hersteller möglich.

Dem stehen folgende **Nachteile** gegenüber:

- Engineeringdaten werden in verschiedenen Datenständen vorgehalten.
- Das Resultat der projektspezifischen Planung dient nur zur Ausschreibung der entsprechenden Komponenten und wird lediglich in textueller Form genutzt.
- Beides widerstrebt dem Ideal einer Durchgängigkeit der Engineeringdaten während des gesamten Engineeringprozesses. Es kommt zu Medienbrüchen, Mehrfacheingaben sowie daraus resultierende Inkonsistenzen.
- Die Standardisierung und Typisierung von Anlagen und Komponenten erfolgt lediglich unternehmensintern, so dass eine detaillierte, bilaterale Abstimmung mit allen Komponentenlieferanten erforderlich wird.
- Die Konfiguration der schutz- und leittechnischen Komponenten wird auf Basis der vorliegenden Spezifikationen und der projektspezifischen Planungsdaten durch den Lieferanten über geräte- bzw. herstellerspezifische Werkzeuge projektbezogen vorgenommen. Eine direkte Übernahme von Daten aus dem Engineeringprozess erfolgt nicht, stattdessen sind zahlreiche Arbeitsschritte zum manuellen Übertragen der Daten erforderlich.

Übergang zum Engineering auf Basis der IEC 61850-6

Seit einigen Jahren unternimmt der Transportnetzbetreiber intensive Anstrengungen, ein Engineering auf Basis der IEC 61850-6 einzuführen und seine bestehenden Engineering- und Projektierungsprozesse entsprechend zu migrieren. Ein Teil dieser Anstrengungen besteht in der Entwicklung eines neuen Engineeringwerkzeugs, das vollständig auf der IEC 61850-6 basiert. Das bewährte Konzept der Standardisierung und Typisierung von Anlagen und Komponenten sowie der Trennung von Grundlagenplanung und projektspezifischer Planung werden auch hier konsequent verfolgt.

Ein Teil der bestehenden Engineeringinformationen kann direkt auf die Sprach-elemente der IEC 61850-6 abgebildet werden. Für weitere, insbesondere stark unternehmensspezifische Informationen, werden die in der IEC 61850-6 definierten privaten Elemente genutzt. Diese ermöglichen die Ablage nicht normativer Informationen innerhalb des normativen Datenmodells.

Das Resultat des Engineerings und der Projektierung einer Anlage ist eine SSD-Datei mit der System Specification Description. Diese enthält das vollständige Datenmodell der Anlage, insbesondere die Spezifikation der sekundärtechnischen Komponenten (IED-Datenmodelle) und deren Kommunikationsbeziehungen, und dient als zentrales Austauschformat zwischen Transportnetzbetreiber und Hersteller bzw. Lieferanten.

Durch die Nutzung der IEC 61850-6 wird ein Großteil der Engineeringinformationen in einem international genormten Format beschrieben. Allerdings bleibt es durch die Nutzung privater Elemente weiterhin erforderlich, deren Semantik mit den Herstellern bzw. Lieferanten abzustimmen. Außerdem werden einige der bisherigen Engineeringinformationen, etwa die Verriegelungsbedingungen, zurzeit noch nicht innerhalb des neuen Engineeringwerkzeugs modelliert.

3.1.2 Anwendungsbeispiel Verteilnetzbetreiber

Das hier betrachtete Unternehmen ist als Versorgungsdienstleister in einer überwiegend ländlich geprägten Region mit einem Versorgungsgebiet von rund 1.800 km² tätig und versorgt rund 165.000 Einwohner unter anderem mit Strom, Erdgas, Wärme und Wasser. Die Durchführung der einzelnen betrieblichen Aufgaben erfolgt dabei im eigenen Hause innerhalb verschiedener Fachabteilungen. Die Fachabteilung *Netzbetrieb Strom* ist dabei für das Mittel- und Niederspannungs-verteilnetz zuständig und untergliedert sich in mehrere Sachgebiete.

Bedarfe für den Ausbau und die Erneuerung von Netzen und Anlagen werden einerseits im Rahmen einer leitungsgebundenen Rehabilitationsstrategie, die Leitungstyp, Alter und Störanfälligkeit berücksichtigt, ermittelt. Bei technischen Problemen und Störungen kann durch die zuständigen Anlagenverantwortlichen zusätzlich sofortiger Bedarf angemeldet werden. Weiterhin kann durch andere Fachabteilungen über den Wirtschaftsplan Bedarf angemeldet werden. Auch Anschlussbegehren durch Netzkunden generieren Bedarfe. Zielnetzplanungen werden bedarfsorientiert in Abstimmung mit anderen Fachabteilungen durchgeführt, unternehmensinterne Planungsgrundsätze definieren und standardisieren hierbei mögliche Netzausbauvarianten. Die Dokumentation der Netzdaten wie Leitungsverläufe und Leitungstypen erfolgt innerhalb eines geografischen Informationssystems (GIS, hier: System der *VertiGIS GmbH* [68]). Betriebsmitteldaten werden in einem

Betriebsmittelinformationssystem (BIS, hier: System der *Lovion GmbH* [69]) verwaltet. Des Weiteren werden seit einigen Jahren die primärtechnischen Netze sukzessive in der Netzberechnungssoftware *DlgSILENT PowerFactory* [70] elektrisch modelliert.

Das Sachgebiet *Leittechnik* verantwortet die Auswahl und Spezifikation der zum Einsatz kommenden fernwirktechnischen Komponenten, wobei auch besondere Anforderungen der Netzleitstelle berücksichtigt werden. Die Ausprägung der Fernwirkgeräte in Bezug auf die technische Ausstattung (z. B. Anzahl der Ein- und Ausgänge) etwa zur Anbindung unterschiedlicher Schaltfeldtypen (z. B. Leitungsfeld, Transformatorfeld) ist über unternehmensinterne Standards definiert. Als Leittechnik kommen Geräte mehrerer Hersteller (hauptsächlich *SAE IT-systems GmbH & Co. KG* [71] und *VIVAVIS AG* [72] (ehemals *Erwin Peters Systemtechnik GmbH* [73]), selten auch *WAGO GmbH & Co. KG* [74]) zum Einsatz. Die Gerätekonfiguration erfolgt durch eigene Mitarbeiter.

Das Sachgebiet *Leittechnik* ist ebenfalls verantwortlich für die Auswahl und Konfiguration der Schutzgeräte, die hier immer als eigenständige Geräte ausgeführt sind (d. h. im Gegensatz zu kombinierten Schutz-/Leitgeräten, die auch am Markt verfügbar sind). Bei Neuinstallationen werden Geräte des Herstellers *Siemens* (*SIPROTEC*, Gerätegeneration 3 bis 5 [75]) verwendet, der konkrete Gerätetyp wird fallweise nach Anforderung ausgewählt; im Bestand sind jedoch auch noch Geräte verschiedener anderer Hersteller im Einsatz. Die Auswahl der Schutzkonzepte und Berechnung der Schutzparameter erfolgen durch das Sachgebiet *Prozesskoordination*. Hierzu werden Berechnungstabellen in Microsoft Excel genutzt, zukünftig ist jedoch eine Berechnung über *DlgSILENT PowerFactory* geplant. Die Konfiguration der Schutzgeräte erfolgt ebenfalls durch eigene Mitarbeiter unter Nutzung eines herstellerspezifischen Softwarewerkzeugs (*Siemens DIGSI* [76]).

Schließlich ist das Sachgebiet *Netzleitstelle* für die Netzführung zuständig. Darüber hinaus sind die Mitarbeiter dieser Fachabteilung aber auch für die Planung und Konfiguration des Netzleitsystems (hier: *PSIprins* [77]) verantwortlich. Aus dem Netzleitsystem heraus ergeben sich auch unternehmensintern standardisierte Vorgaben für Kommunikationsadressen (IP-Adressen und Telegrammadressen des Kommunikationsprotokolls IEC 60870-5-104) an das Sachgebiet *Leittechnik*.

Zum Engineering und zur Konfiguration der Sekundärtechnik dienen verschiedene Softwarewerkzeuge: Die elektrische Planung von Schaltschränken etwa im Sinne der Erstellung von Stromlaufplänen erfolgt in *EPLAN Electric P8* [78]. Die Parametrierung und Programmierung der Fernwirkgeräte erfolgt unter anderem in den Softwareprodukten *setIT* und *codeIT* der *SAE IT-systems GmbH & Co. KG*. Weitere Informationen wie Feldbezeichnungen, Meldetexte, Telegrammadressen und IP-Adressen werden in Standard-Officedokumenten (Microsoft Excel und Word) dokumentiert.

Der Engineering- und Projektierungsprozess weist folgende **Vorteile** auf:

- Durch die Definition unternehmensinterner Standards wird eine Einheitlichkeit von Netzstrukturen sowie der schutz- und leittechnischen Komponenten ermöglicht.
- Dadurch, dass alle betrieblichen Aufgaben im eigenen Hause erfolgen, ist eine enge Abstimmung der einzelnen Fachabteilungen und Sachgebiete gegeben, Schnittstellenproblematiken zu externen Dienstleistern entfallen.
- Die Nutzung von Komponenten verschiedener Hersteller erhöht die Diversität und vermeidet starke Herstellerabhängigkeiten.

Dem stehen folgende **Nachteile** gegenüber:

- Die Projektierung, das Engineering und die Konfiguration der Netze und Anlagen erfolgt in mehreren, voneinander unabhängigen Softwarewerkzeugen. Dies erzeugt hohen personellen und fachlichen Aufwand. Außerdem wird so eine zentrale, konsistente Datenhaltung verhindert.
- Einige Informationen werden lediglich in deskriptiver in Standard-Officedokumenten gepflegt. Normativ wohldefinierte Datenformate werden nicht genutzt.
- Die Verwendung von Komponenten verschiedener Hersteller macht zur Konfiguration und Parametrierung mehrere herstellereigenspezifische Softwarewerkzeuge erforderlich. Dementsprechend müssen diese Werkzeuge vorgehalten werden und die Mitarbeiter in ihrer Nutzung geschult sein.

3.1.3 Zusammenfassung

Die beiden dargestellten Anwendungsbeispiele zeigen, dass die bisher gelebten Engineering- und Projektierungsprozesse oftmals sehr heterogen und durch zahlreiche, voneinander unabhängige und manuelle Arbeitsschritte geprägt sind. Zum Einsatz kommen dabei diverse Softwarewerkzeuge, zwischen denen kein Datenaustausch möglich ist, es wird eine Vielzahl von Medien eingesetzt, eine zentrale Datenhaltung findet nicht statt. Dies führt zu Medienbrüchen und infolgedessen zu Eingabe- und Verarbeitungsfehlern, die aufwendig etwa bei der Inbetriebnahme von Anlagen identifiziert und beseitigt werden müssen. Folge ist ein hoher personeller und finanzieller Aufwand. Daher existieren Bestrebungen, Prozesse zu vereinheitlichen, übergreifende Werkzeuge einzuführen und kompatible Schnittstellen zu schaffen. Deren Fokus lag bisher eher bei den Betreibern von Hoch- und Höchstspannungsnetzen, da hier der initiale Aufwand zur Einführung neuer Prozesse verhältnismäßig war im Vergleich zu den Kosten der Netze selbst. Ein einzelner, kleiner Verteilnetzbetreiber hingegen kann diesen Aufwand für sich allein kaum stemmen. Umso wichtiger ist es, hier standardisierte Prozesse zu schaffen, die den Anforderungen möglichst vieler Netzbetreiber gerecht werden. Gerade auch für Betreiber von Mittel-

und Niederspannungsnetzen ist dies von enormer Bedeutung, um die Digitalisierung ihrer Netze im Sinne des Smart-Grid-Gedankens realisieren zu können. Ein Rückblick in Kapitel 2.3 zeigt, dass die dort vorgestellten Ansätze und Lösungen zur Digitalisierung der Netze das Thema Engineering und Projektierung gar nicht oder nur in Ansätzen mitdenken. Hier gibt es erheblichen Nachholbedarf.

3.2 Die Normenreihe IEC 61850

Die Normenreihe IEC 61850 stellt einen internationalen Kommunikationsstandard für Automatisierungssysteme in der elektrischen Energieversorgung dar. Während der Fokus anfangs auf Stationsautomatisierungssystemen (SAS, Schutz- und Leittechnik) in Höchst-, Hoch- und Mittelspannungsanlagen lag, findet der Standard zunehmend auch Anwendung in unteren Spannungsebenen und zur Anbindung dezentraler Anlagen.

Die Gründe für einen neuen Kommunikationsstandard in der elektrischen Energieversorgung sind vielfältig: Stationsautomatisierungssysteme nutzten in der Vergangenheit proprietäre und/oder signalorientierte Kommunikationsprotokolle. Proprietäre Kommunikationsprotokolle schränken den Anlagenbetreiber bei der Wahl der Gerätetechnik, bezogen auf eine Anlage, oftmals auf einen Hersteller ein und führen damit zu einer starken Abhängigkeit, eine Mehrlieferantenstrategie im Sinne technischer und wirtschaftlicher Vielfalt wird erschwert. Sollen Anlagen unterschiedlicher Hersteller und Technologien miteinander oder mit einer übergeordneten Leitstelle kommunizieren, so sind Komponenten zur Protokollumwandlung notwendig, die aufwendig und kostenintensiv projiziert werden müssen. Signalorientierte Kommunikationsprotokolle wie zum Beispiel IEC 60870-5-104 sind auf hohe Dateneffizienz ausgelegt, um der in der Vergangenheit begrenzten und teuren Kommunikationsbandbreite Rechnung zu tragen. Während eine hohe Kommunikationsbandbreite heute günstig verfügbar ist, sind mittlerweile vielmehr die Kosten für das Engineering ausschlaggebend. Ein modernes, objektorientiertes Engineering, das auf klar strukturierte, selbstbeschreibende Datenmodelle setzt, ist hierbei dem signalorientierten Engineering der Vergangenheit deutlich überlegen. Kommunikationstechnologien wie Ethernet und TCP/IP sind allgegenwärtig, so dass Hardware und Software hierfür in großer Vielfalt günstig am Markt verfügbar sind. Damit ist es nur logisch, auch im Bereich der elektrischen Energieversorgung diese Standardtechnologien zu nutzen und damit anwendungsspezifische Speziallösungen zu ersetzen.

Ein Ziel der IEC 61850 ist es, eine Interoperabilität von schutz- und leittechnischen Komponenten verschiedener Hersteller zu ermöglichen, so dass diese Komponenten mit nur geringem Engineeringaufwand Daten austauschen können. Gemeinsame Basis hierfür ist ein einheitliches Kommunikationsprotokoll, das die Aufgaben bisher unter-

schiedlicher, anwendungsspezifischer Kommunikationsprotokolle zusammenfasst und diese somit ersetzt. Die IEC 61850 definiert jedoch nicht nur die Kommunikation im Sinne von Kommunikationsprotokollen. Vielmehr ermöglicht sie auch die funktionale Beschreibung eines SAS und liefert hierzu Datenmodelle für eine Vielzahl von Funktionstypen. Dabei werden keine Vorgaben zur algorithmischen Umsetzung der Funktionen selbst gemacht, sondern lediglich die Daten, die zu einer Funktion gehören, definiert. Hersteller behalten damit Freiheiten bei der Implementierung und Unterscheidungsmöglichkeiten bei Funktionen und Produkten. Es wird somit nicht der Anspruch einer direkten Austauschbarkeit (engl.: Interchangeability) von Systemen verschiedener Hersteller unter Beibehaltung identischer Funktionalität erhoben.

Ein weiteres Ziel der IEC 61850 ist die logische Trennung der Funktionsmodelle von der Kommunikationsarchitektur. Dadurch wird sichergestellt, dass zukünftige Erweiterungen oder Weiterentwicklungen beider Aspekte unabhängig voneinander möglich sind. Dies ist gerade mit Blick auf die höheren Entwicklungs- und Innovationszyklen der Kommunikations- und Gerätetechnik gegenüber denen schutz- und leittechnischer Funktionen ein entscheidender Vorteil und stellt die Nachhaltigkeit von Funktionsmodellen im Sinne einer Wiederverwendbarkeit sicher. Zur Umsetzung dieser logischen Trennung definiert die IEC 61850 ein objektorientiertes Datenmodell (Objektmodell). In diesem Objektmodell wird eine Funktion als *logischer Knoten* (engl.: Logical Node, LN) gemäß IEC 61850-7-4 [79] beschrieben, dem die Daten der Funktion als *Datenobjekte* (engl.: Data Object, DO) und *Datenattribute* (engl.: Data Attribute, DA) gemäß IEC 61850-7-3 [80] untergeordnet sind. Die Daten des Datenmodells können zwischen Funktionen und Systemen ausgetauscht werden. Für diesen Datenaustausch definiert die IEC 61850 ein abstraktes Dienstmodell (engl.: Abstract Communication Service Interface, ACSI, [81]). Dienste sind etwa das Lesen oder Schreiben eines Wertes, das Ausführen eines Befehls oder das Abonnieren von Datenobjekten oder Datenattributen, so dass bei Wertänderungen automatische Benachrichtigungen erfolgen. Erst im zweiten Schritt wird das Dienstmodell auf ein konkretes Kommunikationsprotokoll abgebildet (engl.: Specific Communication Service Mapping, SCSM) wie zum Beispiel auf MMS (engl.: Manufacturing Messaging Specification) und GOOSE (engl.: Generic Object Oriented Substation Events) gemäß IEC 61850-8-1.

Mit dem Normenteil IEC 61850-7-4 beschränkten sich die logischen Knoten zunächst auf schutz- und leittechnische Funktionen für Schalt- und Umspannanlagen. Eine Erweiterung auf die funktionalen Anforderungen von Wasserkraftanlagen erfolgte mit der IEC 61850-7-410 [82], auf dezentrale Erzeugungsanlagen mit der IEC 61850-7-420 [83] und auf Windkraftanlagen mit der IEC 61400-25-2 [84]. Auch bei der Abbildung des ACSI auf weitere Kommunikationsprotokolle gab es Erweiterungen, so etwa mit der IEC 61400-25-4 [85] für Anwendungen im Bereich von Windkraftanlagen.

Ziel der IEC 61850 ist es außerdem, das Engineering und die Konfiguration von Schutz- und leittechnischen Systemen möglichst einfach, durchgängig und nachhaltig zu gestalten. Mit der System Configuration description Language (SCL) wird hierzu im Normenteil IEC 61850-6 eine geeignete Beschreibungssprache definiert. Eine vertiefende Betrachtung der SCL erfolgt in Kapitel 3.3. Auf den Engineeringprozess selbst, Engineeringrollen, d. h. Akteure, die am Engineeringprozess beteiligt sind, sowie Engineeringwerkzeuge, zwischen denen Daten auf Basis der SCL ausgetauscht werden, wird in Kapitel 3.4 eingegangen. Tabelle 3.1 zeigt die wichtigsten Normenteile.

Tabelle 3.1: Die wichtigsten Normenteile der IEC 61850

Systemaspekte

| |
|--|
| Teil 1: Einführung und Übersicht |
| Teil 2: Begriffsdefinitionen |
| Teil 3: Allgemeine Anforderungen |
| Teil 4: System- und Projektverwaltung |
| Teil 5: Kommunikationsanforderungen für Funktionen und Gerätemodelle |

Konfiguration

| |
|---|
| Teil 6: Konfigurationsbeschreibungssprache für die Kommunikation in Energieautomatisierungssystemen |
|---|

Grundlegende Kommunikationsstruktur

| |
|---|
| Teil 7-1: Grundsätze und Modelle |
| Teil 7-2: Abstrakte Schnittstelle für Kommunikationsdienste (ACSI) |
| Teil 7-3: Gemeinsame Datenklassen (CDC) |
| Teil 7-4: Kompatible Logische-Knotenklassen und Datenobjektklassen |
| Teil 7-410: Logische Knoten für Wasserkraftwerke |
| Teil 7-420: Logische Knoten für dezentrale Erzeugungsanlagen und Verteilnetzautomatisierung |

Abbildung auf spezifische Kommunikationsdienste (SCSM)

| |
|---|
| Teil 8-1: Abbildung auf MMS (ISO 9506-1 und ISO 9506-2) und GOOSE (ISO/IEC 8802-3 Ethernet) |
| Teil 8-2: Abbildung auf XMPP (RFC 6120, RFC 6121, RFC 6122) |
| Teil 9-2: Abgetastete Werte (Sampled Values) über ISO/IEC 8802-3 (Ethernet) |

Prüfungen

| |
|------------------------------|
| Teil 10: Konformitätsprüfung |
|------------------------------|

Weitere (Auswahl)

| |
|--|
| Teil 80-1: Leitfaden zum Austausch von Informationen aus einem CDC-basierten Datenmodell unter Verwendung von IEC 60870-5-101 oder IEC 60870-5-104 |
| Teil 90-1: Verwendung der IEC 61850 für die Kommunikation zwischen Unterstationen |
| Teil 90-2: Verwendung der IEC 61850 für die Kommunikation zwischen Unterstationen und Leitstellen |

3.3 Datenmodelle nach IEC 61850-6

3.3.1 System Configuration description Language

Mit der System Configuration description Language existiert eine Sprache zur Beschreibung der Spezifikation und Konfiguration eines Stationsautomatisierungssystems. Die SCL basiert dabei auf der erweiterbaren Auszeichnungssprache (engl.: eXtensible Markup Language, XML [86]). Ziel der SCL ist es, eine Möglichkeit zu schaffen, die gesamten Engineeringdaten in einem wohldefinierten, standardisierten Format vorzuhalten und einen kompatiblen Informationsaustausch zwischen verschiedenen Engineeringwerkzeugen zu ermöglichen. Durch die Vermeidung von Medienbrüchen und Mehrfacheingaben gewährleistet die SCL einen konsistenten und nahtlosen Datenfluss während des gesamten Engineeringprozesses von der Planung, der Gerätekonfiguration, der Systemvalidierung und -prüfung bis zur Inbetriebnahme und Dokumentation. Da die SCL herstellerneutral und geräteunabhängig sowie (mit Einschränkungen in Bezug auf Fehlerkorrekturen) abwärtskompatibel ist, stellt sie die Nachhaltigkeit und Dauerhaftigkeit sämtlicher Engineeringdaten über Geräte- und Systemgenerationen hinaus sicher. Die SCL existiert mittlerweile in vier Spracheditionen gemäß Tabelle 3.2.

Tabelle 3.2: Spracheditionen der SCL

| Edition | Standard ⁴ | Namespace ⁵ |
|-----------------|--------------------------------|------------------------|
| 1.0 | IEC 61850-6:2004 | IEC 61850-6:2003 |
| 2.0 | IEC 61850-6:2009 | IEC 61850-6:2007B |
| 2.0 Amendment 1 | IEC 61850-6:2009/AMD1:2018 | IEC 61850-6:2007B4 |
| 2.1 | IEC 61850-6:2009/AMD1:2018 CSV | IEC 61850-6:2007B4 |

Mit der SCL lassen sich folgende Aspekte einer energietechnischen Anlage modellieren:

- Die primärtechnischen Komponenten der Anlage wie z. B. Schaltgeräte und Transformatoren sowie deren elektrische Verbindungen (*Substation*),
- die sekundärtechnischen (schutz- und leittechnischen) Komponenten der Anlage (*Intelligent Electronic Devices, IEDs*),
- die Basisfunktionen in Form von logischen Knoten (*LN*s), die in der Anlage verwendet werden, sowie die zugehörigen Daten (*DO*s, *DA*s),
- die Zuordnung der Basisfunktionen (*LN*s) zu den primärtechnischen Komponenten, d. h. welche primärtechnische Komponente verfügt über welche Funktionalität,

⁴ AMD: Änderung (engl.: Amendment), CSV: Konsolidierte Fassung (engl.: Consolidated Version)

⁵ Namensraum (engl.: Namespace): Beispiel IEC 61850-6:2007B4 =: Version 2007, Revision B, Release 4

- die Zuordnung der Basisfunktionen (LNs) zu den IEDs, d. h. welches IED stellt welche Funktionalität bereit,
- die Kommunikation auf Anwendungsebene, d. h. Definition der Daten, die kommuniziert werden
- und die Kommunikation auf Netzwerkebene, d. h. welches IED ist an welches Kommunikationsnetzwerk angeschlossen inkl. der Kommunikationsparameter (*Communication*).

Die zuvor beschriebenen Aspekte beziehen sich zunächst einmal auf die Komponenten und Beziehungen innerhalb einer einzelnen Anlage. Dementsprechend wurde die SCL bisher primär dafür genutzt, jeweils genau eine Anlage zu beschreiben, d. h. bei der Beschreibung mehrerer Anlagen wurde für jede Anlage ein eigenes SCL-Modell erstellt. Gemäß SCL-Sprachdefinition spricht allerdings nichts dagegen, innerhalb eines SCL-Modells mehrere Anlagen (Substations) zu beschreiben und die IEDs aller Anlagen in diesem Modell abzulegen. Insbesondere wurde die SCL mit der Edition 2.1 um Sprachelemente (*Line* und *Process*) erweitert, mit denen die Modellierung primärtechnischer Komponenten außerhalb einer Schaltanlage wie zum Beispiel Leitungen möglich ist.

3.3.2 Sprachelemente

Jedes SCL-Modell besitzt genau ein *SCL*-Element als Wurzelement. Diesem Element sind gemäß Abbildung 3.2 alle weiteren Elemente untergeordnet.

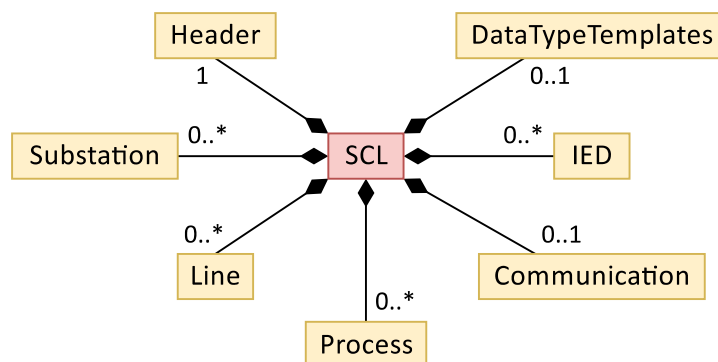


Abbildung 3.2: Grundstruktur eines SCL-Modells

Im *Header*-Element können Metainformationen des SCL-Dokuments wie die Version und Revision, der Name des Bearbeitungswerkzeugs sowie die Bearbeitungshistorie in textueller Form abgelegt werden. Das SCL-Dokument kann beliebig viele *Substation*-Elemente zur Beschreibung primärtechnischer Anlagen enthalten, ebenso beliebig viele *IED*-Elemente zur Beschreibung der sekundärtechnischen Komponenten. Innerhalb eines *Communication*-Elements wird die Konfiguration der Kommunikation auf Netzwerkebene beschrieben. Mit den Elementen *Line* und *Process* lassen sich primärtechnische Komponenten außerhalb der eigentlichen Anlage wie etwa

Leitungen modellieren. Das *DataTypeTemplates*-Element enthält Typvorlagen für Funktionen (*LNnodeType*) und die zugehörigen Daten (*DOType*, *DAType*, *EnumType*). Zur Identifikation der im Folgenden beschriebenen SCL-Elemente besitzen die meisten von ihnen ein Attribut *name*, einige ein Attribut *id*. Innerhalb einer Hierarchieebene muss dabei eine Eineindeutigkeit des entsprechenden Attributwerts sichergestellt werden.

Abbildung 3.3 zeigt das Objektmodell von Substation und IED.

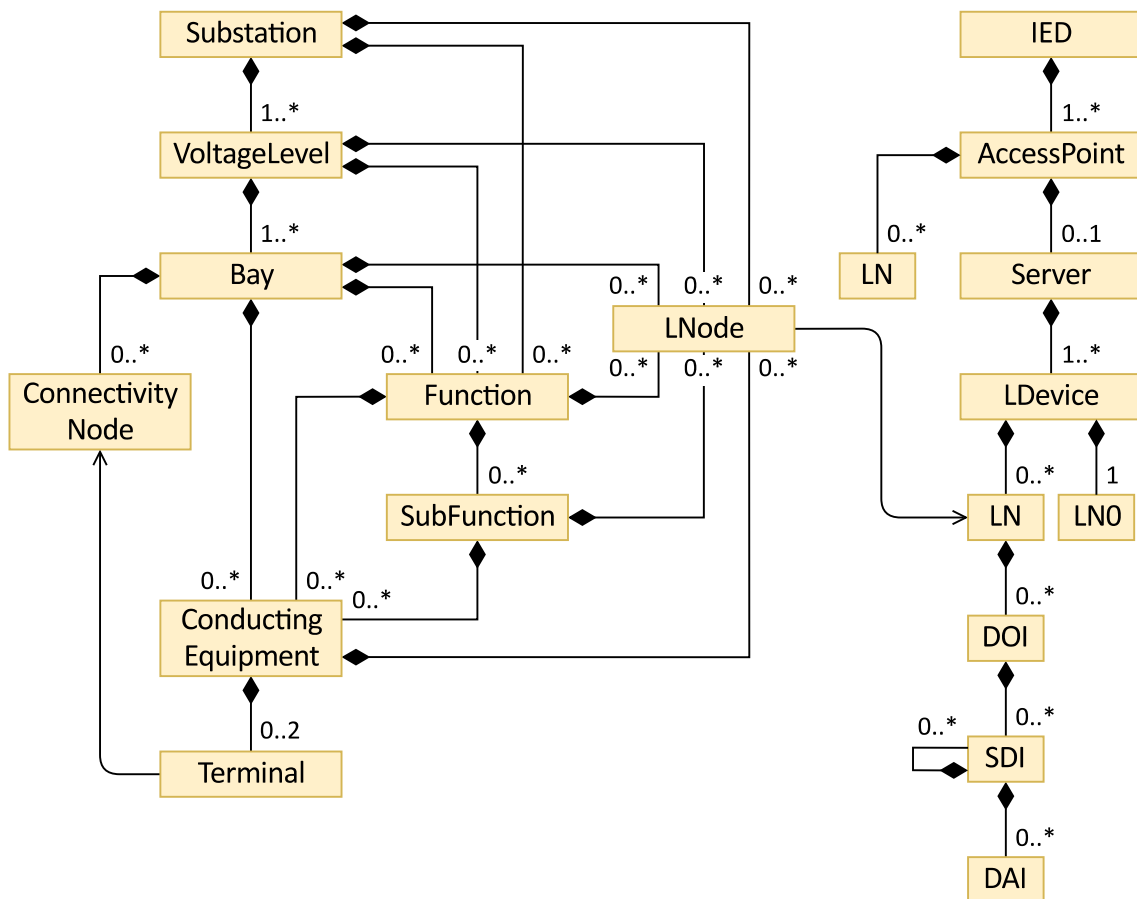


Abbildung 3.3: Objektmodell des Substation- und IED-Teils der SCL (reduziert)

Die Beschreibung von primärtechnischen Komponenten im Sinne von Hardwareprodukten liegt außerhalb des Anwendungsbereichs der SCL. Vielmehr beziehen sich die Beschreibungen im Substation-Teil der SCL auf deren funktionale Eigenschaften.

Eine Anlage (*Substation*) enthält hierarchisch Spannungsebenen (*VoltageLevel*), Schaltfelder (*Bay*) und Geräte (*Equipment*). Bei einem Gerät kann es sich beispielsweise um einen Leistungsschalter, Trennschalter, Spannungs- oder Stromwandler (*ConductingEquipment*), aber auch um eine einzelne Wicklung (*TransformerWinding*) eines Leistungstransformators (*PowerTransformer*) handeln. Mit *SubEquipment*-Elementen können *Equipment*-Elemente weiter untergliedert werden, etwa in phasenbezogene Einheiten innerhalb eines Dreiphasensystems. Ein Transformator als Gesamteinheit kann einer Anlage, einer Spannungsebene oder auch nur einem

Schaltfeld zugeordnet werden. Mit den Elementen *Function* und *SubFunction* können Funktionen, die nicht direkt primärtechnischer Art sind, logisch definiert werden, etwa Schutzgeräte innerhalb eines Schaltfeldes, aber beispielsweise auch eine Feuermeldeeinrichtung innerhalb der Gesamtanlage.

Auch die elektrischen Verbindungen der primärtechnischen Komponenten lassen sich im Sinne eines einpoligen Ersatzschaltbildes logisch modellieren. Hierzu dienen die Elemente *Terminal* und *ConnectivityNode*. Ein *ConductingEquipment* kann als Unterobjekte bis zu zwei *Terminals* als Anschlusspunkte enthalten. Ein Leistungsschalter hat (im einpoligen Ersatzschaltbild) zwei Anschlusspunkte, während ein Spannungswandler nur über einen solchen verfügt. Jedes *Terminal* hält eine Referenz auf einen *ConnectivityNode*. Dieser stellt einen elektrischen Verknüpfungspunkt dar und kann durch eine beliebige Anzahl von *Terminals* referenziert werden, etwa im Falle einer Sammelschiene. Ein *ConnectivityNode* befindet sich als Unterobjekt unter einem *Bay*.

Mit abstrakten⁶ logischen Knoten (*LNode*-Elemente) kann schließlich der erforderliche Funktionsumfang der Anlage spezifiziert werden, d. h. welche der zuvor genannten Elemente über welche Basisfunktionen verfügen. Dabei stellen *LNodes* im Substation-Teil der SCL Referenzen auf logische Knoten (*LN*s) in IEDs dar. Das Erstellen einer solchen Referenz ist ein expliziter Schritt innerhalb des Engineeringprozesses, bei dem definiert wird, durch welchen logischen Knoten (*LN*) innerhalb eines IEDs welche Basisfunktion (*LNode*) realisiert wird.

Die Modellierung der sekundärtechnischen Komponenten stellt den zweiten zentralen Aspekt der SCL dar. Jede derartige Komponente, etwa ein Schutz- oder Leitgerät, wird durch ein IED repräsentiert. Mit einem oder mehreren untergeordneten *AccessPoint*-Elementen wird die Anbindung eines IEDs an ein oder mehrere Kommunikationsnetzwerke beschrieben. Die eigentlichen datenhaltenden Kommunikationseinheiten werden durch *Server*-Elemente modelliert. Hierbei kann einem *AccessPoint* genau ein *Server* untergeordnet sein. Aus den Kommunikationsnetzwerken heraus ist über den jeweiligen *AccessPoint* ein Zugriff auf die in einem *Server* enthaltenen Daten möglich. Ein Sonderfall ist ein *AccessPoint* ohne *Server*-Element, aber mit untergeordneten logischen Knoten, wodurch eine reine Client-Funktionalität beschrieben wird.

Eine funktionale Untergliederung innerhalb eines Servers erfolgt durch logische Geräte (*Logical Device, LD*). Innerhalb der *LD*s sind logische Knoten zur Beschreibung der Basisfunktionen enthalten. Zur Beschreibung der eigentlichen Daten enthalten die *LN*s ihrerseits Datenobjekte (*DOI*), ggf. in hierarchischer Schachtelung mit Unterdaten-

⁶ Abstrakt soll an dieser Stelle ausdrücken, dass diese logischen Knoten nicht als Online-Datenknoten in einem physischen Gerät existieren („leben“), sondern lediglich ein Modellierungselement der SCL darstellen.

objekten (*SDI*), und schließlich Datenattribute (*DAI*) als Blattelemente. Die LNs werden während des Engineeringprozesses durch Zuordnung zu einem *LNodeType* aus den *DataTypeTemplates* typisiert. Zwei Besonderheiten stellen die logischen Knoten *LPHD* (*Physical Device Information*) und *LNO* (*Logical Node Zero*) dar: Der *LPHD* enthält allgemeine, das physische Gerät betreffende Informationen wie den Systemzustand und muss pro Server genau einmal vorhanden sein. Der *LNO* übernimmt eine ähnliche Funktion für logische Geräte und muss entsprechend pro LD einmal vorhanden sein.

Innerhalb der LNs erfolgt weiterhin die Konfiguration der Kommunikation auf Anwendungsebene. In einem LN, standardmäßig dem *LNO*, wird dazu mit *DataSet*-Elementen beschrieben, welche Daten (*DOs*, *DAs*) kommuniziert werden, und mit *ReportControl*- (*Report Control Block*) oder *GSEControl*-Elementen (*GOOSE Control Block*), mit welchem Kommunikationsverfahren die Daten kommuniziert werden.

Innerhalb des Communication-Teils der SCL gemäß Abbildung 3.4 wird die Kommunikation auf Netzwerkebene modelliert. Einzelne Netzwerksegmente werden durch *SubNetwork*-Elemente beschrieben, über *ConnectedAP*-Elemente (dt.: verbundener *AccessPoint*) erfolgt die Verknüpfung von *IEDs* zu einem *SubNetwork*. Hierzu hält jeder *ConnectedAP* Referenzen auf die Namen des *IEDs* und *AccessPoints*. Weiterhin erfolgt die Konfiguration der Kommunikationsadressen. IP-Adressbestandteile werden als *P*-Elemente in einem *Address*-Element direkt unterhalb eines *ConnectedAP* beschrieben, *MAC*-Adressen für *GOOSE* und *Sampled Values* eine Hierarchieebene tiefer unterhalb *GSE*- bzw. *SMV*-Elementen.

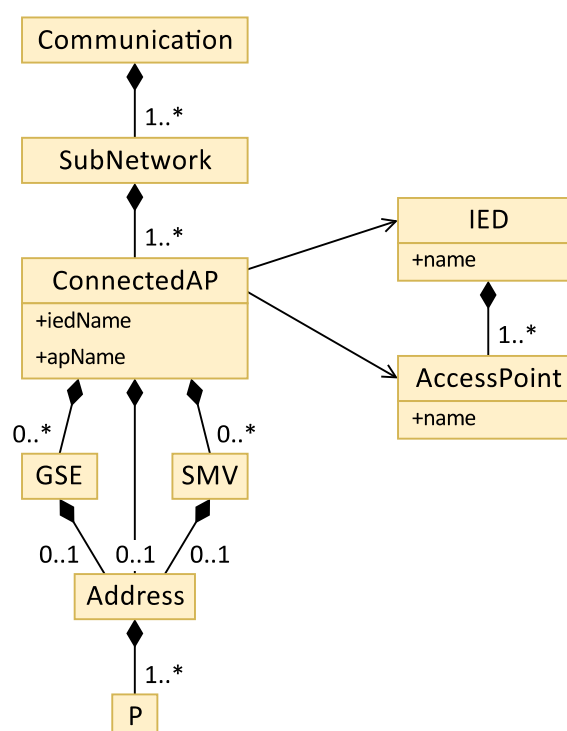


Abbildung 3.4: Objektmodell des Communication-Teils der SCL

Die Abbildungen 3.5 und 3.6 zeigen die SCL-Elemente *Line* bzw. *Process*. Da sowohl innerhalb des Substation- als auch des Line-Teils *ConductingEquipment*-Elemente modelliert werden können, lassen sich beide SCL-Teile über die dortigen Terminals verknüpfen. Ebenso wie ein Bay kann auch eine Line ConnectivityNodes enthalten, so dass ein Terminal aus dem einen Teil einen ConnectivityNode im jeweils anderen Teil referenzieren kann. Der Process-Teil ermöglicht darüber hinaus eine weitergehende, hierarchische Strukturierung primärtechnischer und funktionaler Komponenten.

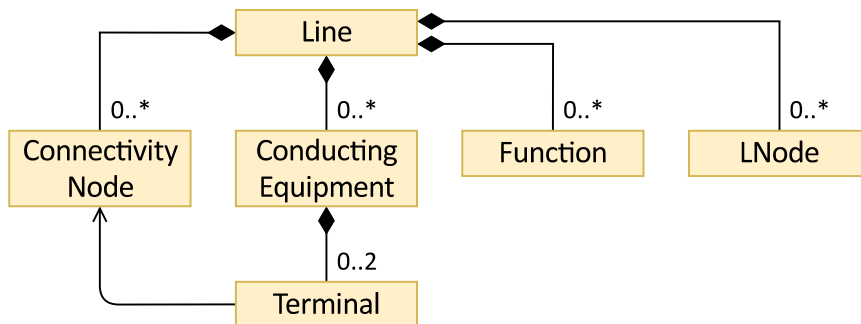


Abbildung 3.5: Objektmodell des Line-Teils der SCL

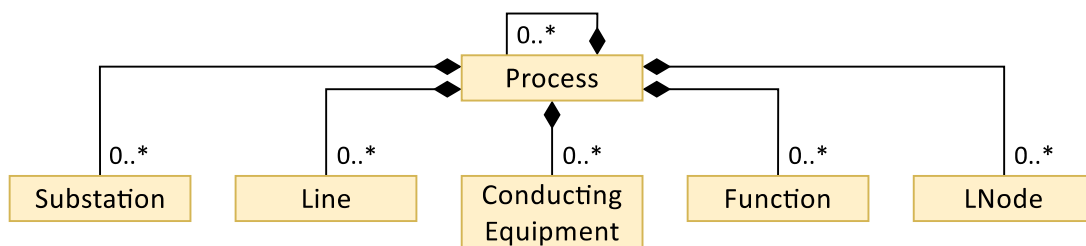


Abbildung 3.6: Objektmodell des Process-Teils der SCL (reduziert)

Über die DataTypeTemplates gemäß Abbildung 3.7 lassen sich schließlich Typvorlagen für logische Knoten (*LNodeType*) sowie die entsprechenden untergeordneten Elemente (*DOType*, *DAType*, *EnumType*) definieren. Dabei verweist ein unterlagertes Element wie z. B. ein DO über sein Attribut *type* auf das Attribut *id* des jeweiligen Typs.

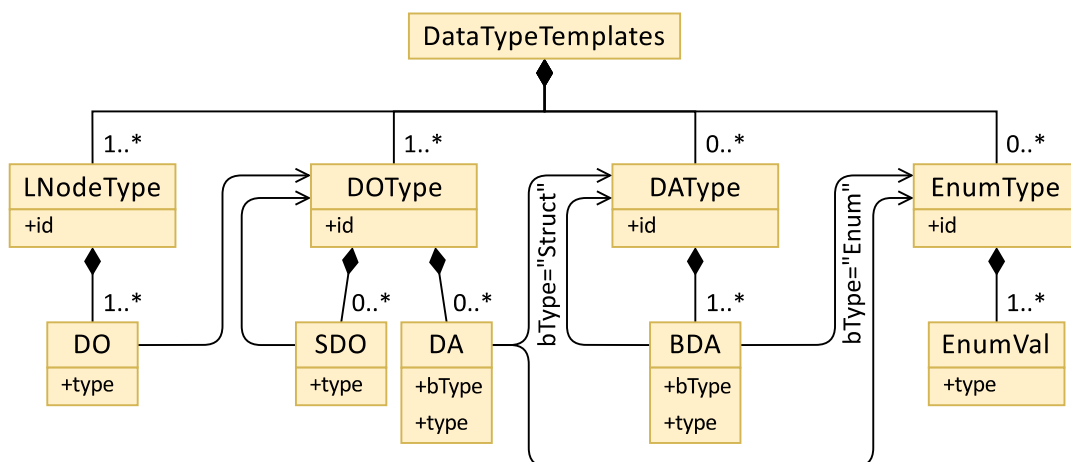


Abbildung 3.7: Objektmodell des DataTypeTemplates-Teils der SCL

Um über die Norm hinausgehende, z. B. anwender- oder werkzeugspezifische Informationen in der SCL ablegen zu können, besteht die Möglichkeit, den meisten in der SCL definierten Elemente untergeordnete *Private*-Elemente hinzuzufügen. Ein solches Element kann beliebige Daten aufnehmen. Über sein Attribut *type* kann informativ auf seinen Inhalt verwiesen werden, etwa auf das Softwarewerkzeug oder den Hersteller, der das private Element eingeführt hat und verwendet. Private Elemente dürfen von fremden Softwarewerkzeugen nicht verändert werden, sondern müssen bei der Verarbeitung von SCL-Daten eingelesen und unverändert wieder gespeichert werden.

Weitergehende Details zu einzelnen Sprachelementen werden, wo erforderlich, in Kapitel 4.5 gegeben.

3.4 Der Engineeringprozess nach IEC 61850-6

Die IEC 61850-6 beschreibt ausführlich, wie sich Engineeringdaten unter Nutzung der SCL abbilden lassen. Den Engineeringprozess selbst, Engineeringrollen, d. h. Akteure, die am Engineeringprozess beteiligt sind, sowie Engineeringwerkzeuge, zwischen denen Daten auf Basis der SCL ausgetauscht werden, werden von der Norm allerdings lediglich abstrakt skizziert. Weitergehende Details, die für eine Umsetzung in konkrete Engineeringprozesse und Engineeringwerkzeuge erforderlich wären, werden nicht geliefert. An dieser Stelle setzt der DKE-Arbeitskreis 952.0.1 unter anderem mit seinen Dokumenten *Beschreibung des Engineeringprozesses* [87] und *Anforderungen an IEC 61850 Engineeringwerkzeuge* [88] an, indem er die einzelnen Engineeringsschritte und Engineeringwerkzeuge näher beleuchtet und Engineeringprozesse in verschiedenen Varianten konkretisiert.

Auf die Engineeringrollen soll an dieser Stelle nicht weiter eingegangen werden, da zur grundsätzlichen Betrachtung der Engineeringprozesse nicht relevant ist, welcher Akteur (z. B. Anlagenbetreiber, Hersteller, Dienstleister) konkret welchen Engineeringsschritt ausführt.

Die IEC 61850-6 definiert als Engineeringwerkzeuge den *Systemspezifikator*, den *Systemkonfigurator* sowie den *Geräte- oder IED-Konfigurator*. Aus dieser Definition lassen sich als Kernprozesse die *Systemspezifikation*, die *Systemkonfiguration* sowie die *Geräte- bzw. IED-Konfiguration* ableiten. Als Hilfswerkzeuge werden darüber noch Werkzeuge zum Test, zur Diagnose und zur Dokumentation genannt. Die nachfolgenden Betrachtungen beschränken sich allerdings auf die drei genannten Kernprozesse. Tabelle 3.3 gibt eine Übersicht über diese Werkzeuge und zeigt deren Aufgaben.

Tabelle 3.3: Werkzeuge gemäß IEC 61850-6 [87, S. 5], [88, S. 6-10]

| Werkzeug zur/zum | Aufgabe |
|---|--|
| Systemspezifikation | <ul style="list-style-type: none"> • Definition der primärtechnischen Struktur der Anlage • Definition der Anforderungen an die Prozessfunktionalität • Definition der IEDs |
| Systemkonfiguration | <ul style="list-style-type: none"> • Zusammenführung von Systemspezifikation und IED-Datenmodellen • Erstellung der Systemkonfiguration • Konfiguration der Kommunikation auf Anwendungs- und Netzwerkebene |
| Gerätekonfiguration (IED-Konfiguration) | <ul style="list-style-type: none"> • Bereitstellung von produktspezifischen IED-Datenmodellen • Verknüpfung von IED-Datenmodell und IED-Applikation • Konfiguration und Parametrierung der IEDs auf Basis der Daten der Systemkonfiguration |
| Test | <ul style="list-style-type: none"> • Test der IEDs und des Gesamtsystems durch Simulation von Prozess- und Kommunikationssignalen • Verifikation der IED- und Systemfunktionen |
| Diagnose | <ul style="list-style-type: none"> • Abgleich zwischen spezifiziertem Datenmodell und in den IEDs umgesetzten Datenmodellen • Diagnose der Netzwerkkommunikation |
| Dokumentation | <ul style="list-style-type: none"> • Erstellung und Ablage von Informationen, die über das SCL-Datenmodell hinausgehen und mit den Elementen der Norm nicht abgebildet werden können, z. B. Signallisten, Schaltpläne, Prüfberichte, gerätespezifische Parameter- und Konfigurationsdaten |

Da Werkzeuge zur Gerätekonfiguration auf die konkrete Gerätetechnik eines Herstellers ausgelegt sind, sind diese immer hersteller- und ggf. sogar gerätespezifisch. Dagegen können und sollten Werkzeuge zur Systemspezifikation und Systemkonfiguration herstellerneutral sein. Einzelne Werkzeuge können darüber hinaus durchaus in einem Gesamtwerkzeug zusammengefasst sein, etwa als kombiniertes Werkzeug zur Systemspezifikation und -konfiguration.

Zwischen den einzelnen Engineeringwerkzeugen werden Daten- und Applikationsmodelle in Form von SCL-Datenmodellen ausgetauscht, die je nach Stadium im Engineeringprozess unterschiedliche Daten enthalten. Um die jeweilige Bedeutung der SCL-Daten kenntlich zu machen, werden sie nach IEC 61850-6 gemäß Tabelle 3.4 bezeichnet. Dabei werden Dateien, die entsprechende SCL-Datenmodelle enthalten, mit den dargestellten Dateibezeichnern („Dateiendung“) versehen. Die Tabelle zeigt außerdem, welche Teilaspekte der SCL (Substation, IED, Communication, DataTypeTemplates) in den verschiedenen Daten enthalten sind bzw. sein können.

Tabelle 3.4: SCL-Daten im Engineeringprozess

| Bezeichnung | Dateibezeichner | Substation | IED | Communication | DataTemplates |
|----------------------------------|-----------------|------------|------|---------------|---------------|
| System Specification Description | SSD | 1..* | 0..* | 0..1 | 0..1 |
| System Configuration Description | SCD | 0..* | 1..* | 1 | 1 |
| IED Capability Description | ICD | 0..1 | 1 | 0..1 | 1 |
| Instantiated IED Description | IID | 0..1 | 1 | 0..1 | 1 |
| Configured IED Description | CID | 0..1 | 1 | 1 | 1 |

Die drei genannten Kernprozessen (*Systemspezifikation*, *Systemkonfiguration*, *Gerätekonfiguration*) sind nicht als monolithische Vorgänge zu verstehen, die nacheinander in strikter Reihenfolge ausgeführt werden. Jeder Prozess kann selbst in weitere Teilprozesse untergliedert werden. Je nach grundsätzlicher und anwender-spezifischer Ausgestaltung des Engineeringprozesses kann durchaus nur eine Auswahl der Teilprozesse ausgeführt werden, außerdem sind Verschränkungen von Teilprozessen verschiedener Kernprozesse während des gesamten Engineerings möglich.

An dieser Stelle sind zwei Engineering-Paradigmen zu unterscheiden, die im Allgemeinen als *Top-Down*- und *Bottom-Up-Engineering* bezeichnet werden [89, S. 18-22]. Beim *Top-Down-Engineering* handelt es sich um einen modellgetriebenen Ansatz: Der Anwender beginnt mit der Spezifikation seiner Anlage und deren funktionalen Anforderungen, herstellernerneutral und losgelöst von einer konkreten Gerätetechnik. Nach der Konfiguration des Gesamtsystems (wozu unter anderem die Kommunikationsbeziehungen zählen) werden schließlich die Geräte (IEDs) mit dem erstellten Datenmodell konfiguriert. Beim *Bottom-Up-Engineering* handelt es sich dagegen um einen produktgetriebenen Ansatz: Startpunkt hierbei sind Datenmodelle konkreter Geräte, die dem Anwender etwa durch den Hersteller bereitgestellt werden. Auf Basis dieser Modelle erstellt der Anwender die Konfiguration des Gesamtsystems, eine Spezifikation findet gar nicht oder nur in begrenztem Umfang statt.

Vorteilhaft beim *Top-Down-Engineering* ist die Flexibilität des Anwenders in der Gestaltung seiner Anlage und der Umsetzung seiner Anforderungen. Durch die Herstellerneutralität ermöglicht dieser Ansatz eine Austauschbarkeit der konkreten Gerätetechnik, ohne dass Änderungen an den Datenmodellen erforderlich werden, und schützt den Anwender so etwa vor Abhängigkeiten von einzelnen Herstellern.

Allerdings muss der Anwender zur Umsetzung dieses Ansatzes über eine tiefgreifende Fachexpertise verfügen und umfangreiche Engineeringarbeiten ausführen. Die Gerätetechnik auf der anderen Seite muss die Fähigkeit besitzen, mit extern vorgegebenen Datenmodellen zu arbeiten.

Beim Bottom-Up-Engineering hingegen kommt der Anwender durch die Nutzung vordefinierter Gerätedatenmodelle mit geringerem Aufwand zu einem lauffähigen Gesamtsystem. Allerdings sind Anpassungen etwa an unternehmensinterne Richtlinien kaum oder gar nicht möglich. Da jeder Hersteller seine Geräte unterschiedlich ausstattet, ist auch eine Austauschbarkeit von Geräten oder der gemeinsame Betrieb von Geräten unterschiedlicher Hersteller nicht oder nur eingeschränkt möglich. Für den Hersteller bietet der Bottom-Up-Ansatz hingegen Vorteile, da er seine Geräte mit fester Funktionalität und vordefiniertem Datenmodell ausstatten kann und so Entwicklungsaufwand und -kosten einspart. Daher waren insbesondere die ersten Gerätegenerationen, die IEC 61850 unterstützten, auf die Nutzung des Bottom-Up-Ansatzes beschränkt.

Abbildung 3.8 gibt in Anlehnung an [88, S. 15] eine Übersicht der einzelnen Engineering Schritte und Engineeringwerkzeuge sowie der beteiligten Daten. Der gesamte Engineeringprozess mit den drei zuvor genannten Kernprozessen wird hierbei weiter in die folgenden zehn Teilschritte untergliedert:

1. Spezifikation Anlagendatenmodell
2. Spezifikation IED-Datenmodell
3. Verknüpfung von Anlagendatenmodell und spezifiziertem IED-Datenmodell
4. Bereitstellung von produktspezifischen IED-Datenmodellen
5. Vergleich von spezifiziertem und produktspezifischem IED-Datenmodell
6. Übernahme produktspezifischer IED-Datenmodelle in die Systemkonfiguration
7. Kommunikationskonfiguration
8. Netzwerkkonfiguration
9. Verknüpfung von IED-Datenmodell und IED-Applikation
10. Laden der Konfiguration in die IEDs

Die Teilschritte sind in der Abbildung als (1) bis (10) gekennzeichnet. Welche Teilschritte konkret ausgeführt werden, hängt insbesondere von der grundsätzlichen Ausgestaltung des Engineeringprozesses ab, d. h. von der Frage, ob es sich eher um ein Top-Down- oder ein Bottom-Up-Engineering handelt.

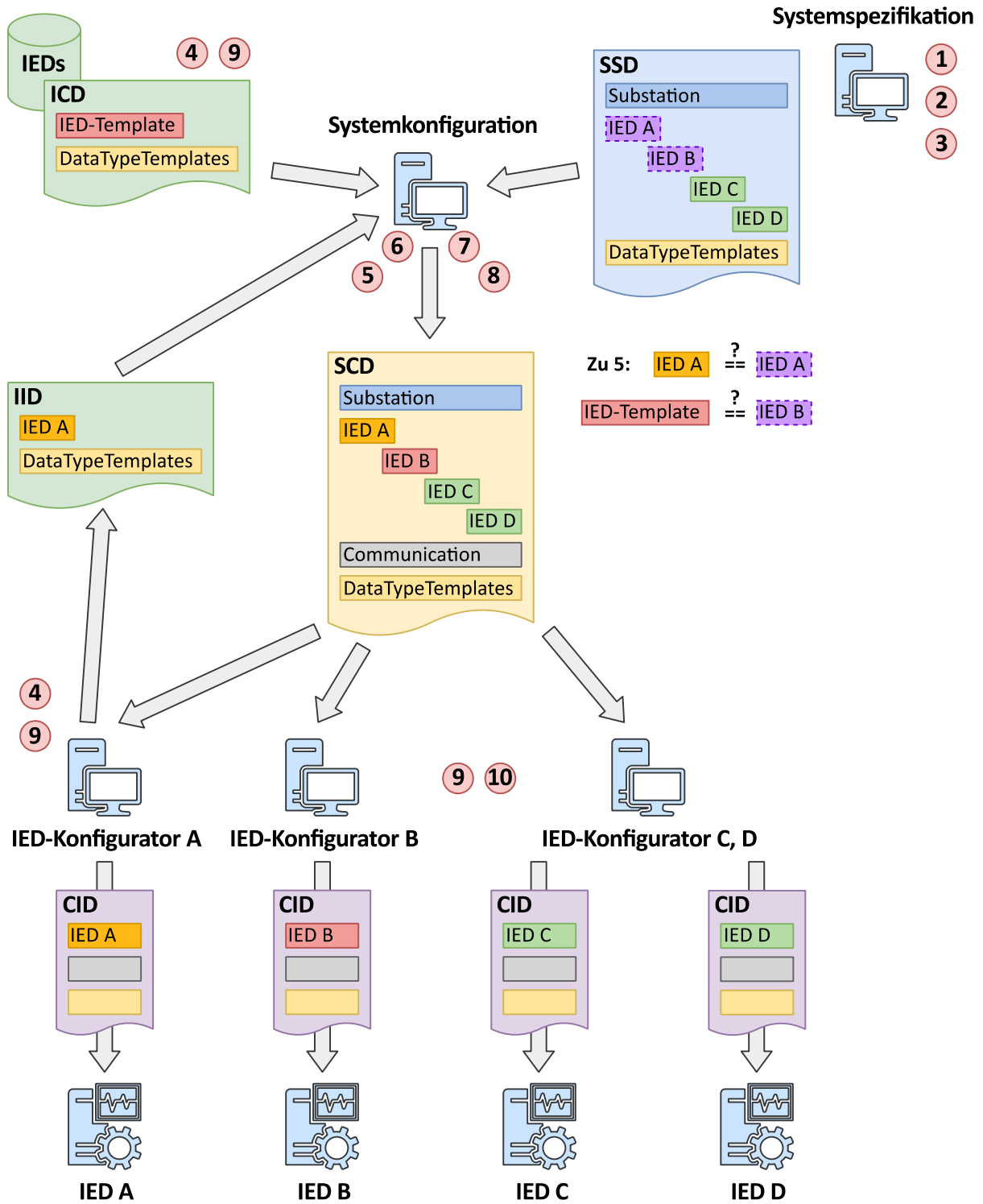


Abbildung 3.8: Engineeringschritte und Engineeringwerkzeuge sowie beteiligte Daten (nach [88, S. 15])

Das DKE-AK-Dokument *Anforderungen an IEC 61850 Engineeringwerkzeuge* definiert fünf verschiedene Engineeringansätze A-E, die entweder mehr einem Bottom-Up- oder einem Top-Down-Engineering zugeordnet werden können. Ansatz A entspricht dabei am ehesten einem vollständigen Bottom-Up-Engineering, Ansatz E einem vollständigen Top-Down-Engineering, die übrigen Ansätze stellen Mischformen dar. Tabelle 3.5 gibt eine Übersicht der Engineeringansätze und zeigt, ob eine Systemspezifikation stattfindet, welche SCL-Daten beteiligt sind und welche Teilschritte ausgeführt werden.

Tabelle 3.5: Engineeringansätze [88, S. 16]

| Engineering-ansatz | Systemspezifikation | | Gerätedatenmodell | | | Systemkonfiguration | Ausgeführte Teilschritte |
|--------------------|--------------------------|-------------------------|-------------------|-----|-----|---------------------|--------------------------|
| | Mit Anlagenspezifikation | Mit Gerätespezifikation | ICD | IID | SCD | | |
| | SSD (Substation) | SSD (IED) | | | | SCD | |
| A | | | X | X | | X | 4, 7-8, 10 |
| B | X | | X | X | | X | 1, 4, 6-8, 10 |
| C | X | X | X | | | X | 1-8, 10 |
| D | X | X | | X | | X | 1-10 |
| E | X | X | | | X | X | 1-3, 7-10 |

Um die Charakteristika der beiden Engineering-Paradigmen Top-Down- und Bottom-Up-Engineering zu veranschaulichen, beschränken sich die nachfolgenden Erläuterungen auf die Ansätze A und E.

Top-Down-Engineering (Ansatz E)

Abbildung 3.9 zeigt das Top-Down-Engineering gemäß Ansatz E.

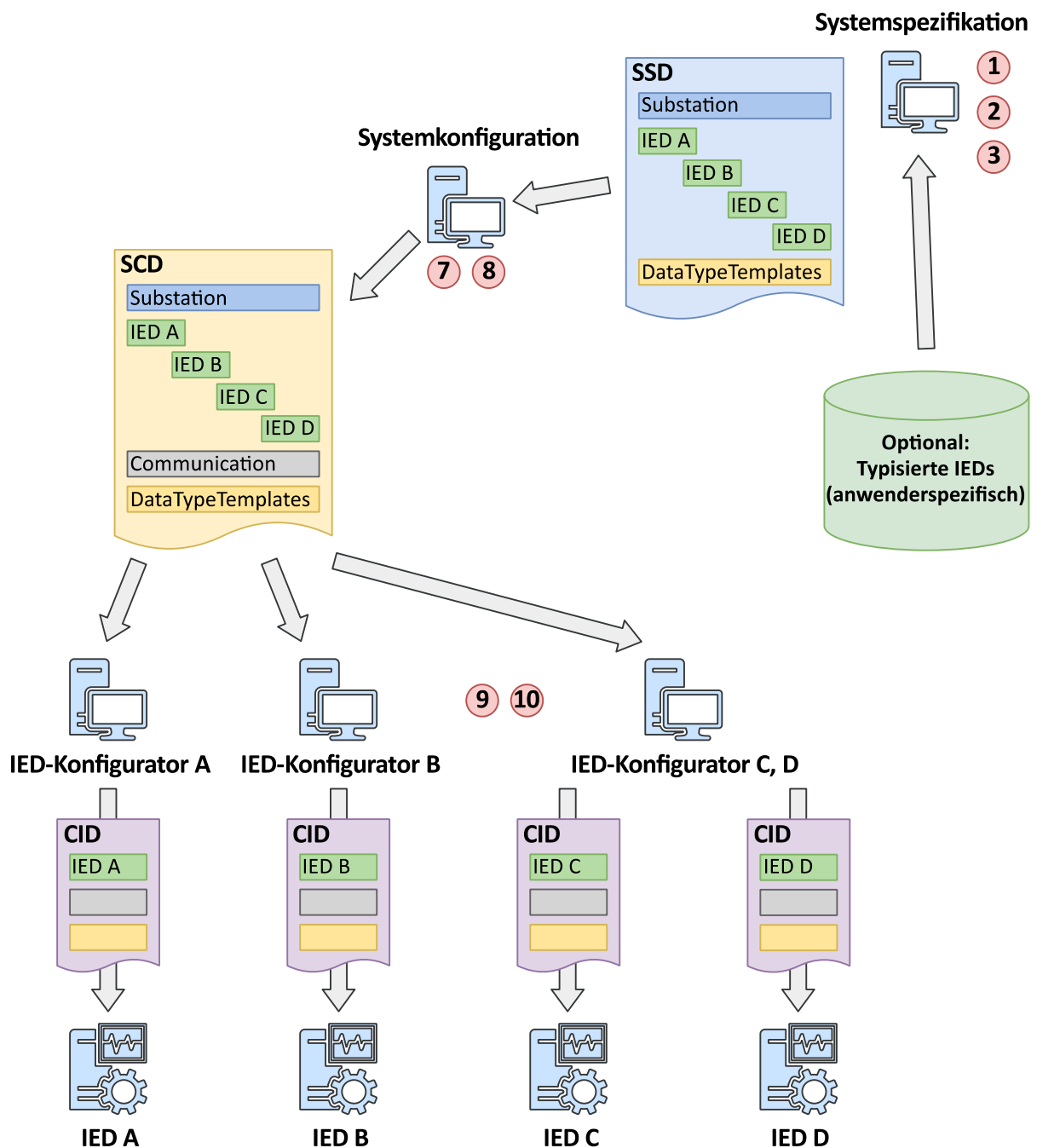


Abbildung 3.9: Top-Down-Engineering gemäß Ansatz E (nach [88, S. 39])

Systemspezifikation

Im ersten Schritt erfolgt die Spezifikation des Anlagendatenmodells (1). Hierbei wird zunächst die primärtechnische Struktur der Anlage modelliert. Neben der logischen Strukturierung der Anlage mit Unterelementen wie Spannungsebenen, Schaltfeldern, Schaltgeräten und Transformatoren zählt hierzu auch die topologische Strukturierung, also die Festlegung, wie die einzelnen primärtechnischen Komponenten elektrisch

miteinander verbunden sind im Sinne eines einpoligen Ersatzschaltbildes. Die Anlagentopologie wird innerhalb der SCL über Referenzen modelliert (Elemente *Terminal* und *ConnectivityNode*, siehe Kapitel 3.3.2), eine grafische Repräsentation ist durch die Norm nicht definiert. Nichtsdestotrotz ist es dem Werkzeug zur Systemspezifikation freigestellt, über die Norm hinaus auch eine Möglichkeit zur grafischen Eingabe der Topologie anzubieten und so dem Anwender diesen Spezifikationsschritt zu vereinfachen. Als Nächstes erfolgt die Festlegung der Anforderungen an die Prozessfunktionalität der Anlage über die Spezifikation von Basisfunktionen in Form von logischen Knoten (*LNodes*), die den einzelnen Elementen der Anlage zugeordnet werden. Als Ergebnis der Anlagenspezifikation enthält das resultierende SCL-Datenmodell die Elemente *Substation* und *DataTypeTemplates*.

Im zweiten Schritt erfolgt die Spezifikation der IED-Datenmodelle (2). Hierzu werden die in der Anlage einzusetzenden IEDs mit den untergeordneten Elementen wie logischen Geräten (*LDevices*) und logischen Knoten (*LN*s) modelliert. Darüber hinaus kann die Kommunikation auf Anwendungsebene (*DataSets*, *Control*-Blöcke) spezifiziert werden. Um eine anwenderspezifische Standardisierung von IEDs umzusetzen und sich wiederholende Aufwände bei der Projektierung mehrerer Anlagen zu reduzieren, kann der Anwender jedoch auch im Vorhinein einmalig IED-Datenmodelle für häufige Anwendungsfälle typisieren. Diese IED-Datenmodelle stehen anschließend als Typvorlagen zur Verfügung und können bei der konkreten Projektierung einer Anlage in die Spezifikation importiert werden. Als Ergebnis der IED-Spezifikation enthält das resultierende SCL-Datenmodell die Elemente *IED* und *DataTypeTemplates*.

Da sowohl das Anlagendatenmodell als auch die IED-Datenmodelle spezifiziert wurden, kann nun eine Verknüpfung beider Modelle (3) vorgenommen werden. Hierzu werden die (abstrakten) logischen Knoten des Anlagendatenmodells mit logischen Knoten in den IED-Datenmodellen verknüpft. Dadurch wird festgelegt, welches IED welche Prozessfunktionalität innerhalb der Anlage ausführt.

Systemkonfiguration

Im ersten Schritt der Systemkonfiguration erfolgt die Konfiguration der Kommunikation auf Anwendungsebene (7). Hierzu wird mit *DataSets* beschrieben, welche Daten kommuniziert werden, und mit *ReportControl*- bzw. *GOOSE-Control*-Blöcken, mit welchem Kommunikationsverfahren die Daten kommuniziert werden. Im zweiten Schritt wird die Konfiguration der Kommunikation auf Netzwerkebene (8) durchgeführt. Hierbei erfolgen die Zuordnung der einzelnen IEDs zu Subnetzen und die Vergabe der Kommunikationsadressen.

Gerätekonfiguration

Im Rahmen der Gerätekonfiguration werden zunächst die IED-Datenmodelle mit den jeweiligen Applikationen (Funktionen) innerhalb der IEDs verknüpft (9). Abschließend werden die IED-Datenmodelle in die jeweiligen IED geladen.

Bottom-Up-Engineering (Ansatz A)

Abbildung 3.10 zeigt das Bottom-Up-Engineering gemäß Ansatz A.

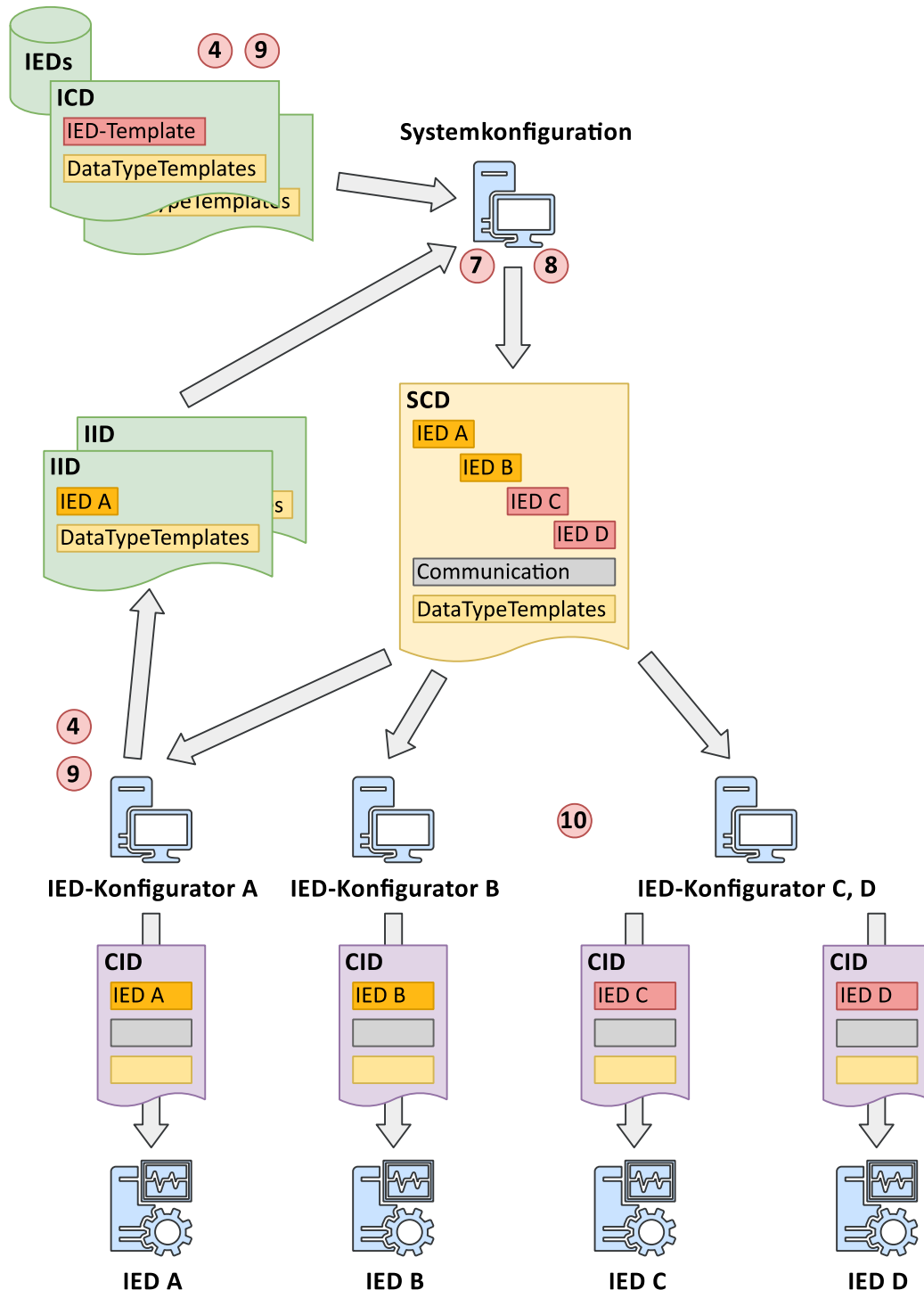


Abbildung 3.10: Bottom-Up-Engineering gemäß Ansatz A (nach [88, S. 17])

Systemspezifikation

Eine Systemspezifikation findet nicht statt.

Systemkonfiguration

Den Startpunkt bilden produktspezifische IED-Datenmodelle (4), die entweder durch den Gerätehersteller als Vorlagen (ICD) bereitgestellt oder mit Hilfe des IED-Konfigurators erzeugt werden (IID). Anschließend erfolgt wie auch bei allen anderen Engineeringansätzen die Konfiguration der Kommunikation auf Anwendungs- (7) und Netzwerkebene (8).

Gerätekonfiguration

Durch die Nutzung produktspezifischer IED-Datenmodelle ist deren Zuordnung zu den IED-Applikationen implizit vorgegeben, so dass keine Verknüpfung stattfindet. Die IED-Datenmodelle werden lediglich wie auch bei allen anderen Engineeringansätzen in die jeweiligen IED geladen (10).

Ansätze B bis D

Bei den übrigen Engineeringansätzen B bis D findet neben der Spezifikation des Anlagendatenmodells teils auch eine Spezifikation der IED-Datenmodelle statt, außerdem werden produktspezifische IED-Datenmodelle (ICD, IID) eingebunden. Die Herausforderung besteht darin, IEDs mit entsprechenden Datenmodellen zu finden, die die spezifizierte Prozessfunktionalität erfüllen können. In diesem Fall können die produktspezifischen IED-Datenmodelle in die Systemkonfiguration übernommen werden (Engineeringansatz B) bzw. ersetzen dort die zuvor spezifizierten IED-Datenmodelle (Engineeringansätze C und D).

3.4.1 Verfügbare Engineeringwerkzeuge

Zum Engineering nach IEC 61850-6 existieren verschiedene Softwarelösungen auf dem Markt, teils als herstellernerneutrale, eigenständige Werkzeuge, teils in hersteller- bzw. gerätespezifische Werkzeuge integriert. Im Nachfolgenden wird ein Überblick über verfügbare Werkzeuge gegeben, ohne jedoch Anspruch auf Vollständigkeit zu erheben.

Bei den folgenden Lösungen handelt es sich um herstellernerneutrale, eigenständige Engineeringwerkzeuge:

ASE61850 SCL Manager

Der *ASE61850 SCL Manager* [90] der Applied Systems Engineering Inc. ist ein Werkzeug zur Systemspezifikation und Systemkonfiguration gemäß IEC 61850-6. Es ermöglicht die Erstellung eines einpoligen Ersatzschaltbildes und die Modellierung der primärtechnischen Anlagenstruktur. Ersatzschaltbilder können dabei auch vorab als Templates in einer Datenbank abgelegt und somit für verschiedene Anlagen immer wieder genutzt werden. Das Werkzeug ermöglicht den Import vorgegebener IED-Datenmodelle und die Verknüpfung der dort definierten logischen Knoten mit denen, die in der primärtechnischen Anlagenstruktur spezifiziert sind. Außerdem ist der Vergleich von IED-Datenmodelle in Bezug auf ihren Funktionsumfang möglich. Die Bearbeitung erfolgt in grafischen Editoren, die Ablage der Datenmodelle erfolgt in den normkonformen Dateiformaten.

Helinks STS

Helinks STS [91] der Helinks LLC ist ein Werkzeug zur Systemspezifikation und Systemkonfiguration gemäß IEC 61850-6. Während der Systemspezifikation können ein einpoliges Ersatzschaltbild erstellt, Funktionen sowie logische Knoten auf Anlagenebene modelliert und diese mit IED-Platzhaltern verknüpft werden. In der Systemkonfiguration können IED-Templates (ICD) eingelesen werden und ersetzen dabei die zuvor angelegten IED-Platzhalter, wobei vorhandene Verknüpfungen erhalten bleiben. Anschließend lassen sich die Kommunikations- und Netzwerkkonfiguration ausführen. Die Arbeitsschritte erfolgen über grafische Editoren, die zugehörigen SCL-Datenmodelle werden automatisch im Hintergrund erstellt. Die Datenmodelle können dabei einerseits in den normkonformen Dateiformaten abgelegt werden. Da IED-Konfiguratoren verschiedener Hersteller zurzeit noch nicht in der Lage sind, streng normative Datenmodelle einzulesen, sondern zusätzliche, herstellerspezifische Informationen benötigen, erlaubt es das Werkzeug, SCL-Datenmodelle zu exportieren, die um diese spezifischen Informationen angereichert sind. Für Dokumentationszwecke ermöglicht das Werkzeug außerdem einen Export in eine menschenlesbare Darstellung.

IEC 61850 Systemkonfigurator

Der *IEC 61850 Systemkonfigurator* [92] der Siemens AG ist ein Werkzeug ausschließlich zur Systemkonfiguration. Obwohl es sich bei der Siemens AG auch um einen Gerätehersteller handelt, ist das Werkzeug selbst herstellernerutral und unterstützt Geräte und Modelle von Fremdherstellern [93]. Das Werkzeug ermöglicht die Arbeitsschritte der Systemkonfiguration, d. h. den Import und das Zusammenfügen einzelner IED-Datenmodelle zum Gesamtsystemdatenmodell sowie die Kommunikations- und Netzwerkkonfiguration. Über Softwareschnittstellen ist eine Anbindung an die eigenen, hersteller- und gerätespezifischen IED-Konfigurationswerkzeuge gegeben.

H & S Substation Configuration Tool

Das *Substation Configuration Tool (SCT)* [94] der H & S Hard- & Software Technologie GmbH & Co. KG ist ein Werkzeug zur normkonformen Systemspezifikation und Systemkonfiguration. Es ermöglicht die Erstellung von primärtechnischen Anlagendatenmodellen inkl. einpoligem Ersatzschaltbild und IED-Datenmodellen, die Definition von sekundärtechnischen Funktionen und logischen Knoten auf Anlagenebene, die Verknüpfung von Anlagendatenmodell und IED-Datenmodell sowie die Kommunikations- und Netzwerkkonfiguration. Außerdem ermöglicht es die Erstellung von Typvorlagen für logische Knoten, Datenobjekte und Datenattribute. Die Bearbeitung der Datenmodelle kann dabei sowohl über grafische Editoren als auch direkt in der hierarchischen Baumstruktur der SCL erfolgen. Die Ablage der Datenmodelle erfolgt in den normkonformen Dateiformaten. Außerdem ermöglicht das Werkzeug es, erstellte oder importierte Datenmodelle auf Normkonformität hin zu validieren.

Bei den folgenden Lösungen handelt es sich um hersteller- bzw. gerätespezifische Engineeringwerkzeuge:

DIGSI

DIGSI der Siemens AG ist ein Werkzeug zur gerätespezifischen Projektierung, Konfiguration, Parametrierung und Bedienung der eigenen Schutzgerätereihe *SIPROTEC*. Das Werkzeug ermöglicht den Import von SCL-Datenmodellen mit Systemspezifikationen (SSD) und Systemkonfigurationen (SCD) sowie den Im- und Export von SCL-Datenmodellen mit Gerätekonfigurationen (ICD, IID, CID). Weiterhin ist auch ein Datenaustausch mit dem zuvor beschriebenen *IEC 61850 Systemkonfigurator* möglich.

SPRECON-E Engineering Center

Beim *SPRECON-E Engineering Center* [95, 96] der Sprecher Automation GmbH handelt es sich um ein Werkzeug zur Projektierung, Konfiguration und Parametrierung der eigenen Schutz- und Leitechnikgeräte. Es ermöglicht die Erstellung von Gerätetemplates, sowohl für die eigene Geräteserie als auch, durch Import entsprechender SCL-Datenmodelle (ICD), für Geräte von Fremdherstellern. Unter Nutzung der in einer Datenbank abgelegten Templates können anschließend konkrete Anlagen projektiert werden. Dabei lassen sich projektspezifische Anpassungen vornehmen. Anschließend können die gerätespezifischen Konfigurationsdateien exportiert oder direkt in das jeweilige Gerät geladen werden. Der Import von Systemspezifikationen oder Systemkonfigurationen wird zurzeit nicht unterstützt, es ist lediglich der Export der gesamten Systemspezifikation (SCD) am Ende des Projektierungsprozesses zu Dokumentationszwecken vorgesehen.

CET850 Configuration Software

Die *CET850 Configuration Software* [97, 98] der Schneider Electric SE ist ein Werkzeug zur Erstellung und Bearbeitung erweiterter IEC-61850-Konfigurationen unter anderem für Schutzgeräten der Serie *Sepam*. Es ermöglicht, vorhandene Gerätedatenmodelle (ICD), die als Datei oder in einer vom Hersteller bereitgestellten Template-Datenbank vorliegen, zu importieren. So können einzelne IED-Datenmodelle zu einem Gesamtsystemdatenmodell zusammengefügt werden. Außerdem lassen sich die Kommunikations- und Netzwerkkonfigurationen bearbeiten. Gerätedatenmodelle lassen sich in einer anwenderspezifischen Datenbank ablegen (ICD), als instanziiertes Datenmodell (IID) zum Austausch mit anderen Werkzeugen oder als konfiguriertes Datenmodell (CID) für eine Übertragung in ein Gerät exportieren. Auch der Im- und Export von Systemkonfigurationen (SCD) zum Austausch mit anderen Werkzeugen ist möglich. Den eigentlichen IED-Konfigurator stellt die Software *SFT2841* dar. Hiermit können Datenmodelle in Geräte hinein bzw. aus Geräten heraus geladen werden. Der Datenaustausch zwischen *CET850* und *SFT2841* erfolgt über Systemkonfigurationsdatenmodelle (SCD) oder konfigurierte Gerätedatenmodelle (CID).

Zusammenfassung

Im Rahmen dieser Arbeit wird ein normkonformes Werkzeug zur Systemspezifikation und Systemkonfiguration benötigt. Ein explizites Werkzeug zur Gerätekonfiguration ist nicht erforderlich, da dieser Vorgang inhärent durch die auf einer Systemplattform ablaufenden Applikation erfolgt. Als konkretes Werkzeug kommt das H & S Substation Configuration Tool zum Einsatz, da es alle erforderlichen Arbeitsschritte unterstützt, normkonform und herstellerneutral arbeitet und bereits aus Forschungsprojekten Erfahrungen mit diesem Werkzeug bestehen.

3.5 Zusammenfassung

Die Betrachtung bisheriger Engineering- und Projektierungsprozesse zeigte deren hohe Komplexität auf. Die Nutzung zahlreicher Softwarewerkzeuge, Datenformate und Medien führt zu einer Vielzahl an Schnittstellen, die einer Durchgängigkeit der Engineeringdaten entgegensteht. Als Folge werden manuelle und damit fehleranfällige Arbeitsschritte und Datenübertragungen notwendig. Bei der Inbetriebnahme von Anlagen werden so umfangreiche Prüf- und Korrekturmaßnahmen mit entsprechend hohem personellem und finanziellem Aufwand erforderlich.

An dieser Stelle setzt die Normenreihe IEC 61850 an. Mit den dort definierten Datenmodellen und Engineeringprozessen wird eine einheitliche und konsistente Modellierung der funktionalen primärtechnischen und insbesondere sekundärtechnischen Objekte sowie der Kommunikation elektrischer Energieanlagen ermöglicht. Dabei deckt der Engineering- und Projektierungsprozess auf durchgängige und nahtlose Weise die Phasen der Spezifikation und der Konfiguration des Gesamtsystems sowie der einzelnen schutz- und leittechnischen Geräte ab.

Für einen Anwender entfaltet sich das volle Potenzial der Normenreihe IEC 61850, wenn er vollständig die dort definierten Datenmodelle nutzt und Engineeringprozesse umsetzt. In vielen Unternehmen existieren jedoch bereits unternehmensspezifische Prozesse, so dass eine Migration geeignet geplant werden muss. Es ist zu untersuchen, wie sich die bestehenden Prozesse, Daten und Medien mit den Möglichkeiten der Norm abbilden lassen. Oftmals ist keine vollständige Überdeckung gegeben, so dass entweder bestehende Prozesse angepasst oder parallel zu den Mitteln der Norm weitere Datenstände gepflegt werden müssen. Ein Lösungsweg besteht in der Nutzung der privaten Elemente. Als Folge ist eine enge Abstimmung mit den Gerätelieferanten erforderlich, um ein einheitliches Verständnis der spezifischen Daten sicherzustellen.

Eine weitere Problematik stellt die teils noch fehlende oder mangelnde Unterstützung durch Gerätehersteller dar. Eine vollständige Unterstützung der Norm fordert von den Herstellern eine große Flexibilität, etwa in der Konfigurierbarkeit ihrer Produkte. Der damit verbundene Entwicklungsaufwand schreckt manchen Hersteller davon ab, solch eine maximale Kompatibilität umzusetzen. Infolgedessen wird dem Anwender nur ein Teil der normativen Möglichkeiten geboten, außerdem sind weiterhin herstellereigene Prozesse durchzuführen. Dies steht einer nahtlosen Austauschbarkeit von Systemen unterschiedlicher Hersteller entgegen. Hier sind auch die Anwender gefordert, einen entsprechenden Erwartungsdruck auf die Hersteller auszuüben, da sie es sind, die letztendlich von einer vielfältigen Gerätelandschaft und einem Innovationswettbewerb zwischen Anbietern profitieren.

4 Realisierung einer Smart-Grid-Automatisierungsarchitektur

Dieses Kapitel beschreibt die Implementierungsarbeiten, die den Kern dieser Arbeit darstellen und die in folgende Teilarbeiten untergliedert werden können: Im ersten Arbeitsschritt erfolgte die Erstellung eines Softwareframeworks, das als Implementierungsumgebung die Basis für die Umsetzung aller netztechnischen Funktionen und Algorithmen bildet. Das Softwareframework ermöglicht es, alle Funktionen auf einheitliche Art und Weise mit wohldefinierten Schnittstellen zu implementieren und anschließend, einzeln oder kombiniert, auf der Systemplattform auszuführen. Hierbei werden insbesondere auch die Anforderungen der einzelnen Funktionen und Algorithmen etwa bezüglich Rechendauer, Zeitdeterminismus und Kommunikationsbedarf berücksichtigt.

Im zweiten Arbeitsschritt wurde ein geeigneter Konfigurationsprozess, der eine Modellierung und Konfiguration der Systeme auf Basis von Datenmodellen nach IEC 61850-6 ermöglicht, entworfen und in das Softwareframework integriert. Dies ermöglicht es, die auf einem System auszuführenden Funktionen einzig und allein über eine IEC 61850-6-Datenmodelldatei zu konfigurieren, diese beliebig miteinander zu kombinieren und die zwischen ihnen auszutauschen Daten zu beschreiben.

Für eine Vielzahl von Anwendungen ist darüber hinaus der Austausch von Daten wie Messwerten oder Steuerbefehlen auch zwischen Systeminstanzen erforderlich. Die in der IEC 61850-6 definierten Sprachelemente ermöglichen hierzu eine Modellierung und Konfiguration der Kommunikationsbeziehungen zwischen Systeminstanzen, beziehen sich grundsätzlich aber zunächst einmal auf die in der IEC 61850 definierten Kommunikationsverfahren wie MMS und GOOSE. Durch Einbindung eines externen Kommunikationsstacks ist die Nutzung dieser Kommunikationsverfahren direkt gegeben. Darüber hinaus besteht jedoch der Bedarf, weitere Kommunikationsverfahren einzusetzen, um unter anderem einen Datenaustausch mit einem Echtzeitsimulator zu ermöglichen sowie Drittgeräte an die Systemplattform anzubinden. Daher wurden in einem dritten Arbeitsschritt weitere Kommunikationsverfahren umgesetzt. Die Implementierung der eigentlichen Smart-Grid-Funktionen und Algorithmen erfolgte größtenteils im Rahmen paralleler Forschungs- und Laborarbeiten. Im Rahmen dieser Arbeit wurden jedoch einzelne Schutzfunktionen auf die Systemplattform übertragen, um die Eignung der Systemplattform für die Ausführung dieser anspruchsvollen Funktionen zu verifizieren. Außerdem wurden vorhandene Funktionen dahingehend angepasst, dass eine Einbindung in das Softwareframework ermöglicht wird. Um die Funktionsfähigkeit einzelner Funktionen und des Gesamtsystems zu verifizieren, wurden vorhandene Prüfverfahren angewandt, jedoch nicht selbst entworfen.

Aus den genannten Arbeitsschritten ergibt sich folgende Kapitelstruktur: Kapitel 4.1 beschreibt die exemplarisch genutzte Systemplattform, die sich aus der Messhardware und einer entsprechenden Basissoftware in Form eines Betriebssystems und Firmware zusammensetzt, und stellt die Architektur sowie die technischen Kennwerte der Systemplattform dar. Kapitel 4.2 erläutert das erstellte Softwareframework. Es wird aufgezeigt, wie Smart-Grid-Funktionen und Algorithmen innerhalb des Softwareframeworks umgesetzt werden können. Auf die Implementierung ausgewählter Funktionen, insbesondere auf Schutzfunktionen, wird in Kapitel 4.3 im Detail eingegangen. In Kapitel 4.4 werden die realisierten Kommunikationsverfahren dargestellt, zudem eine Datenbanklösung, die eine Langzeitaufzeichnung von Daten wie Messwerten und Berechnungsergebnissen ermöglicht. Kapitel 4.5 schließlich stellt den Entwurf und die Implementierung des Konfigurationsprozesses dar. Im ersten Teil dieses Kapitels werden zunächst die SCL-Sprachelemente beschrieben, die zur Modellierung und Konfiguration der Smart-Grid-Funktionen und Kommunikationsverfahren aus den vorherigen Kapiteln genutzt werden. Im zweiten Teil wird das Verfahren der automatischen Gerätekonfiguration auf Basis eines so erstellten SCL-Datenmodells dargelegt. Der dritte Teil erläutert, wie die einzelnen Funktionen schließlich auf der Systemplattform ausgeführt und koordiniert werden.

4.1 Systemplattform

Ein besonderes Ziel bei der Umsetzung der Gesamtlösung liegt in der größtmöglichen Trennung von Software und Hardware. Dadurch wird sichergestellt, dass sich Funktionen und Algorithmen mit möglichst geringem Aufwand auf verschiedene Hardwareplattformen übertragen lassen. Als exemplarische Systemplattform zur Funktionsimplementierung, Laborverifikation und Felderprobung dient im Rahmen dieser Arbeit ein bewährtes und robustes Hard- und Softwaresystem aus dem Bereich der Messwerterfassung, der Netzqualitätsanalyse und der Störereignisaufzeichnung der KoCoS Messtechnik AG. In der Grundkonfiguration als Gerätemodell *EPPE CX* verfügt das System über jeweils vier Eingänge zur direkten Messung von Spannungen bis 600 V sowie zur direkten Messung von Strömen bis 10 A. Die Messeingänge ermöglichen einen direkten Anschluss an Niederspannungsnetze oder an die Sekundärseite von Spannungs- und Stromwandlern in Mittel- und Hochspannungsstationen. Die erweiterte Konfiguration als Gerätemodell *SHERLOG CRX* bietet bis zu 32 Messkanäle, die neben einer Direktmessung auch als Sensoreingänge zur indirekten Messung von Strömen über Stromsensoren konfiguriert werden können. Dies ermöglicht die parallele Messung mehrerer Abgänge innerhalb einer Unterstation oder eines Netzanschlusspunkts. Da das System für die Netzqualitätsanalyse gemäß IEC 61000-4-30 Klasse A [99] ausgelegt ist, werden die Messeingänge mit einer hohen Rate von 200 kHz abgetastet und mit 16 Bit aufgelöst. Weiterhin verfügt das System

über Eingänge zur Zeitsynchronisation u. a. mittels GPS, IRIG-B und DCF77, Schnittstellen zur Kommunikation über Ethernet und RS 232/RS 485, ein Netzteil mit Kurzzeitpufferung und binäre Ein- und Ausgänge. Die Abbildungen 4.1 und 4.2 zeigen beide Gerätemodelle.

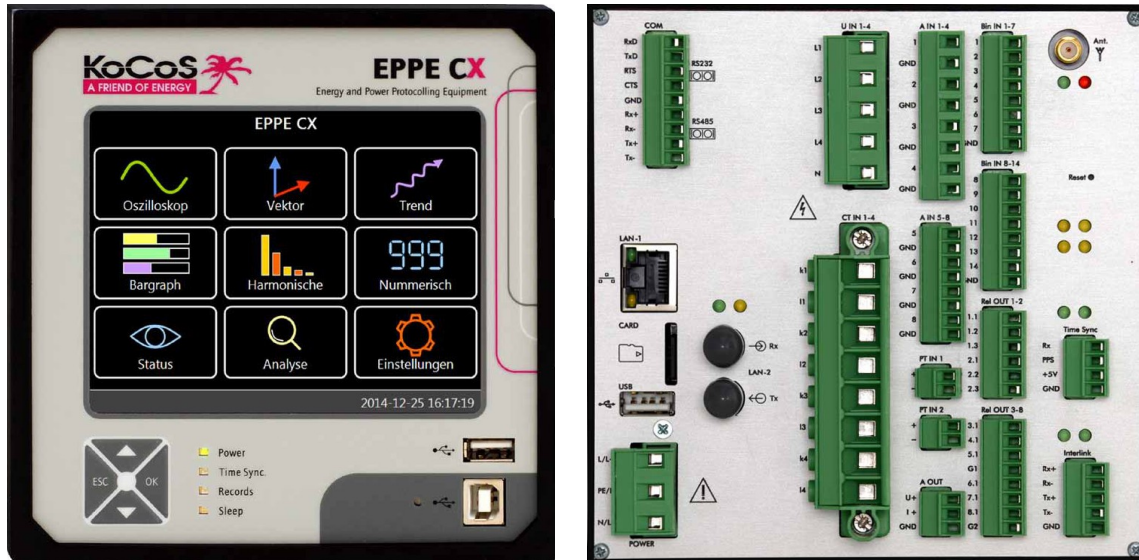


Abbildung 4.1: Systemplattform KoCoS EPPE CX (Vorder- und Rückseite) [100]

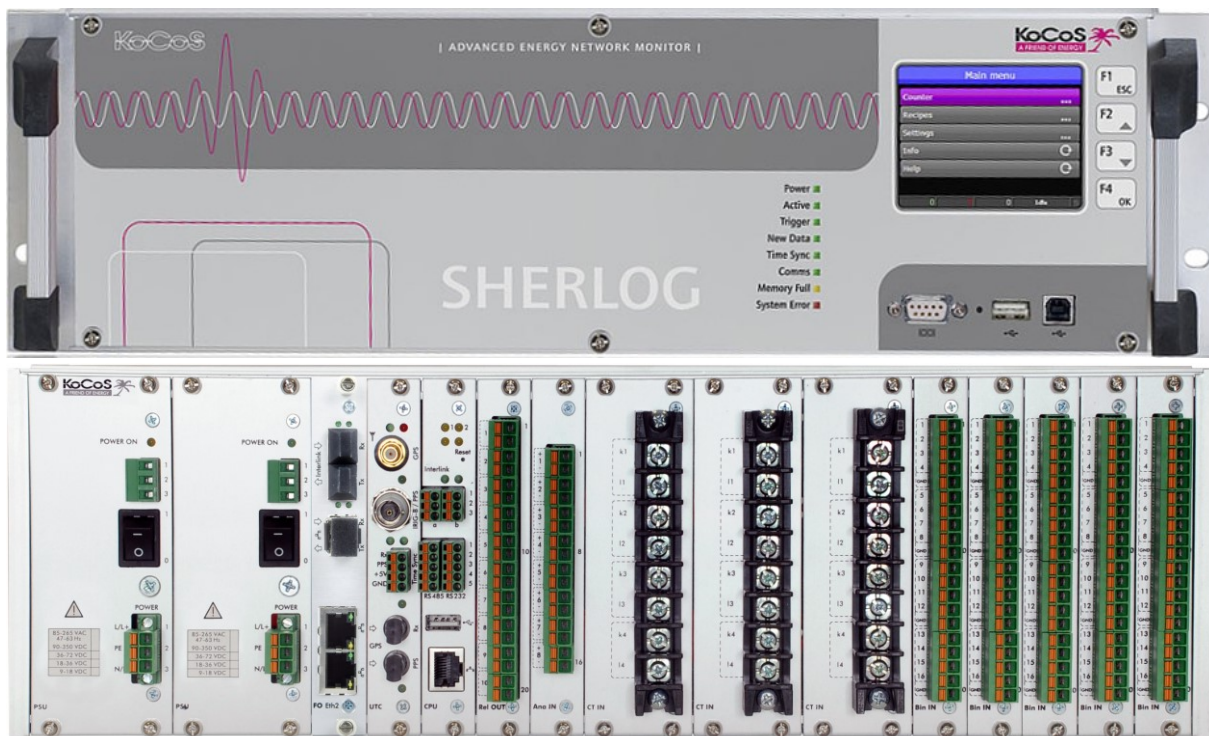


Abbildung 4.2: Systemplattform KoCoS SHERLOG CRX (Vorder- und Rückseite) [100]

Die Systemarchitektur ist, wie in Abbildung 4.3 dargestellt, in zwei Schichten unterteilt.

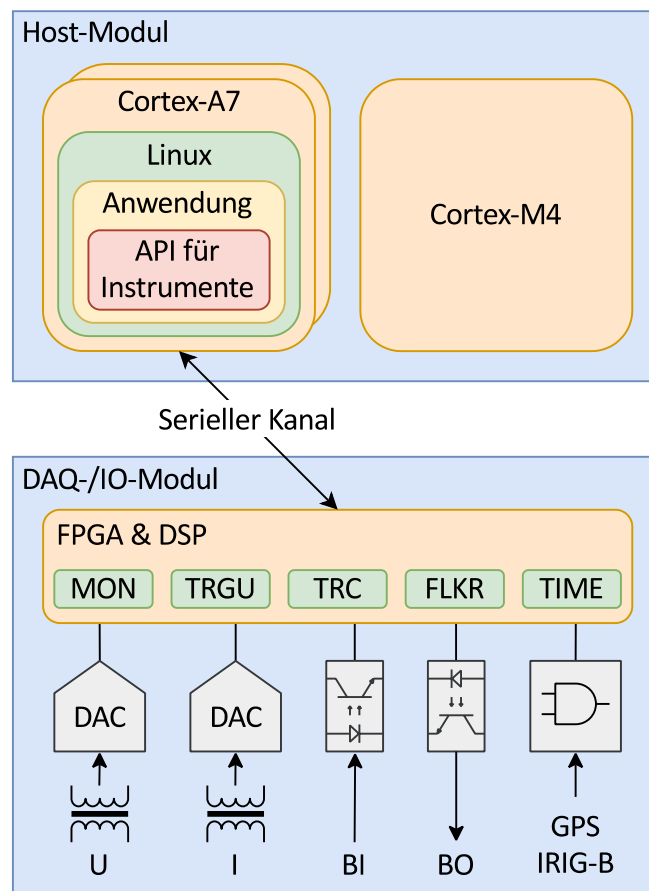


Abbildung 4.3: Systemarchitektur

Die untere Schicht als DAQ-/IO-Modul (engl.: Data Acquisition, Datenerfassung; engl.: Input/Output, Eingang/Ausgang) übernimmt die Messwerterfassung (U , I), Filterung, Signalverarbeitung und Berechnung abgeleiteter Messwerte. Die Signalverarbeitung und Messwertberechnung erfolgt über einen FPGA (engl.: Field Programmable Gate Array, im Feld programmierbare Logikgatter-Anordnung) und DSP (Digitaler Signalprozessor), so dass ein streng definiertes Zeitverhalten garantiert ist. Hierzu sind verschiedene Instrumente in Software implementiert, etwa zur Berechnung von 10-ms-RMS-Werten und Harmonischer (MON, Monitoring), zur Erfassung von Triggerbedingungen (TRGU, Trigger Unit), zur Aufzeichnung hoch abgetasteter Samples (TRC, Transient Recorder) oder zur Berechnung von Flickermesswerten (FLKR). Auch die Zeitsynchronisation (TIME) aus verschiedenen Zeitquellen (GPS, IRIG-B) ist auf dieser Schicht untergebracht und ermöglicht damit eine exakte Zeitstempelung der Messwerte. Schließlich erfolgt auch die Anbindung der binären Ein- und Ausgänge (engl.: Binary Input, BI ; engl.: Binary Output, BO) auf dieser Schicht, wodurch etwa eine ereignisgesteuerte Triggerung von Aufzeichnungen möglich wird.

Die obere Schicht als Host-Modul ist für die Kommunikation mit externen Systemen, die Speicherung der Messwerte und das Benutzerinterface zuständig. Der Daten-

austausch mit dem DAQ-/IO-Modul erfolgt über einen seriellen Kommunikationskanal mit einer Datenrate von 921,6 kBaud. Das Host-Modul basiert auf einem System-on-Module (SoM, dt.: Einplatinencomputer) vom Typ CL-SOM-iMX7 [101] des Herstellers CompuLab Ltd. [102]. Dieses SoM verfügt serienmäßig über einen ARM-Cortex-A7-Dualcore-Prozessor (engl.: Advanced RISC (Reduced Instruction Set Computer) Machine) mit 1 GHz Taktfrequenz, einen ARM-Cortex-M4-Co-Prozessor mit 200 MHz Taktfrequenz, 256 MiB Arbeitsspeicher sowie 128 MiB nichtflüchtigen Speicher (SLC NAND-Flash). Als Betriebssystem wird auf dem ARM Cortex-A7 ein mit dem Softwarepaket Buildroot [103] spezifisch erstelltes Linux-System ausgeführt, der ARM Cortex-M4 ist in der serienmäßigen Softwareausstattung der KoCoS Messtechnik AG ungenutzt. Das SoM ist auch in abweichenden Konfigurationen erhältlich, insbesondere mit größerem Arbeits- und nichtflüchtigem Speicher, so dass bei Bedarf zukünftig eine Aufrüstung des Systems möglich ist.

Die Implementierung der Smart-Grid-Funktionen und Algorithmen erfolgt als eigenständige Anwendung auf Betriebssystemebene des Host-Moduls. Diese Anwendung ersetzt dabei die serienmäßige Anwendung des Herstellers. Über eine Anwendungsprogrammierschnittstelle (engl.: Application Programming Interface, API) können das DAQ-/IO-Modul instruiert und Messwerte von dort abgerufen werden. Durch das fehlende zeitdeterministische Verhalten auf Betriebssystemebene lässt sich hier zwar die Ausführung schneller und zeitkritischer Anwendungen wie Schutzfunktionen für einen produktiven Einsatz des Systems nicht realisieren. Das Testen der allgemeinen Funktionalität der Implementierung, etwa hinsichtlich der Messgenauigkeit und der Funktionskoordination, ist dennoch möglich.

Der bisher ungenutzte Cortex-M4-Co-Prozessor bietet durch den Verzicht auf ein Betriebssystem bzw. die Nutzung eines Echtzeitbetriebssystems (engl.: Real-Time Operating System, RTOS) potenziell die Möglichkeit, Funktionen mit zeitdeterministischem Verhalten auszuführen. In Kapitel 4.3.1 wird die exemplarische Implementierung einer Schutzfunktion auf diesem Controller aufgezeigt. Anhand dieser Schutzfunktion wird schließlich in Kapitel 5.1 die grundsätzliche Fähigkeit des Systems zur zeitdeterministischen Ausführung von Funktionen verifiziert.

4.2 Softwareframework

Das Softwareframework als Implementierungsumgebung ermöglicht es, Smart-Grid-Funktionen umzusetzen, unter Nutzung von Datenmodellen nach IEC 61850-6 zu konfigurieren und schließlich auf der Systemplattform auszuführen.

Um eine maximale Portabilität des Programmcodes zu erreichen, erfolgt die Softwareimplementierung in der Sprache C++ in der Version 2011 (C++11, ISO/IEC 14882:2011 [104]). Als integrierte Entwicklungsumgebung (engl.: Integrated Development Environment, IDE) dient Qt Creator [105].

Jede Schutz-, Überwachungs-, Leit- und Automatisierungsfunktion ist modular als separate Klasse mit ggf. weiteren Hilfsklassen mit wohldefinierten Schnittstellen implementiert. Die Speicherung und der Austausch von Messwerten, Parametern, Auslösesignalen usw. erfolgt ausschließlich über ein Online-Datenmodell, das aus logischen Knoten, Datenobjekten und Datenattributen gemäß IEC 61850-7-3 und IEC 61850-7-4 besteht und durch das algorithmisch navigiert werden kann. Zur Umsetzung wurde ein IEC-61850-Kommunikationsstack der H & S Hard- & Software Technologie GmbH & Co. KG implementiert. Dieser Stack ist zweigeteilt: Das *Abstract Communication Service Interface* (ACSI) dient der Erzeugung und Bereitstellung eines Online-Datenmodells als abstraktes Dienstmodell; das *Specific Communication Service Mapping* (SCSM) bildet das Dienstmodell auf konkrete Kommunikationsprotokolle (MMS und GOOSE gemäß IEC 61850-8-1), ab, wodurch eine Netzwerkkommunikation zwischen Systeminstanzen ermöglicht wird. Abbildung 4.4 zeigt die Implementierungsstruktur für das Beispiel der zweistufigen Spannungsregelung.

Zurzeit sind sämtliche Funktionen in einer einzigen Projektmappe implementiert, zur Ausführung auf der Systemplattform wird eine monolithische Konsolenanwendung erstellt (kompiliert und gelinkt). Diese Anwendung kann anschließend auf Ebene des Linux-Betriebssystems ausgeführt werden. Eine automatische und flexible Konfiguration des Systems auf Basis eines SCL-Datenmodells wird durch die Implementierung der Programmkomponente *Autoconfig* ermöglicht. Diese verarbeitet bei Programmstart eine SCL-Datenmodelldatei, instanziiert auf Basis eines darin enthaltenen IED-Datenmodells ein Online-Datenmodell (ACSI) und erzeugt Instanzen für alle Funktionen, die im Substation-Datenmodell spezifiziert sind. Die Systemkonfiguration sowie die Programmkomponente *Autoconfig* werden in Kapitel 4.5 detailliert beschrieben.

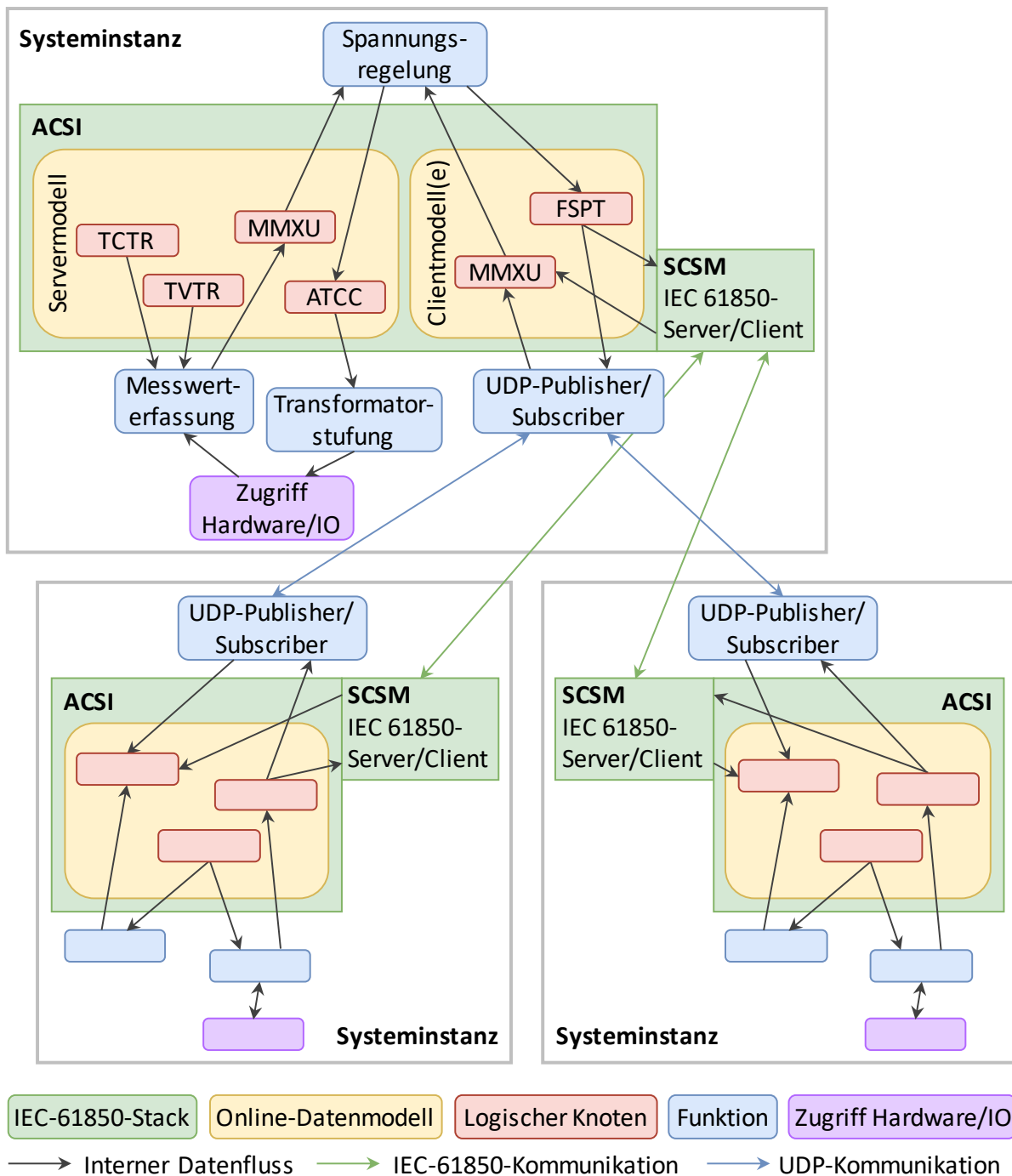


Abbildung 4.4: Implementierungsstruktur (Beispiel zweistufige Spannungsregelung)

Innerhalb der Projektmappe sind die einzelnen Klassen gemäß ihrer Funktion in Namensräume und Verzeichnisse strukturiert (siehe Tabelle A.1 in Anhang A).

4.3 Funktionen

Tabelle 4.1 zeigt die bisher implementierten Funktionen.

Tabelle 4.1: Implementierte Funktionen

| Funktionsschlüsselwort | Beschreibung | Benötigte LNs ⁷ |
|-------------------------------------|--|---|
| Messwerterfassung und -verarbeitung | | |
| <i>MeasurementSinglePhase</i> | Einphasige Messung | n*MMXU, 1*TCTR, 1*TVTR |
| <i>MeasurementThreePhase</i> | Dreiphasige Messung | |
| <i>PrintMeasurements</i> | Messwerte auf der Konsole ausgeben | n*MMXU |
| <i>LogMeasurements</i> | Messwerte in SQLite-Datenbank loggen | n*MMXU |
| Schutz | | |
| <i>PTOC_Function</i> | Überstromzeitschutz (engl.: Time-Overcurrent Protection) | 1*PTOC, 1*MMXU |
| <i>PDIS_Function</i> | Distanzschutz (engl.: Distance Protection) | 1*PDIS, 1*MMXU |
| <i>PTRC_Function</i> | Auslösekoordination (engl.: Trip Conditioning) | 1*PTRC, n*PTOC/PDIS/... |
| Regelung, Monitoring, Optimierung | | |
| <i>VoltageControl_Master</i> | Hauptfunktionalität der Spannungsregelung | MMXU, FSPT, ATCC, CSWI |
| <i>VoltageControl_Slave</i> | Dezentrale Funktionalität der Spannungsregelung | MMXU, FSPT |
| <i>VoltageControl_Centralised</i> | Zentralisierte Spannungsregelung | n*MMXU ⁸ , 1*ATCC, 1*AVCO |
| <i>ATCC_Function</i> | Transformatorstufensteuerung (engl.: Automatic Tap Changer Controller) | ATCC |
| <i>StateEstimation</i> | Netzzustandsschätzung | n*MMXU ⁸ , 0..1*GAPC |
| <i>OptimalPowerFlow</i> | Leistungsflussoptimierung | n*MMXU ⁸ , n*ATCC |
| Weitere | | |
| <i>CSWI_Function</i> | Schaltgerätesteuerung (engl.: Switch Controller) | CSWI |
| <i>FPID_Function</i> | PID-Regler (genutzt von der Spannungsregelung) | AVCO, FPID, MMXU, FSPT |

⁷ Die vorgegebene Reihenfolge der LNs ist einzuhalten.

⁸ Die Angabe dieser logischen Knoten erfolgt über separate Konfigurationsdateien.

Der Datenaustausch zwischen Funktionen erfolgt ausschließlich über logische Knoten mit ihren Datenobjekten und Datenattributen. Hierzu wird für jede Funktion definiert, auf welche notwendigen und optionalen logischen Knoten sie zugreift.

In den nachfolgenden Unterkapiteln wird auf einzelne implementierte Funktionen, namentlich Schutzfunktionen, Spannungsregelungen sowie eine Netzzustands-schätzung, im Detail eingegangen.

4.3.1 Schutzfunktionen

Systeme zur Steuerung und Automatisierung von Mittel- und Niederspannungsnetzen verfügen größtenteils nicht über Schutzfunktionen. Die Integration von Schutzfunktionen in solche Systeme bietet den Vorteil, das Netz im Vergleich zu klassischen Sicherungen oder einfachen Schutzeinrichtungen wie dem ungerichteten Überstromzeitschutz mit höherwertigen Schutzsystemen auszustatten und dabei gleichzeitig dedizierte Schutzgeräte zu vermeiden. Daher werden an dieser Stelle explizit Schutzfunktionen auf der Systemplattform umgesetzt. Da Schutzfunktionen relativ hohe Anforderungen an das System etwa bezüglich deterministischen Zeitverhaltens und Messgenauigkeit stellen, eignen sie sich insbesondere auch zum Prüfen der Randbereiche des Systemverhaltens. Bei vielen Schutzfunktionen, wie zum Beispiel dem Überstromzeit- und dem Distanzschutz, handelt es sich um lokal-autonome Funktionen, so dass die sonstige Peripherie, etwa die Kommunikation, außer Acht gelassen werden kann.

Die Implementierung grundlegender Schutzfunktionen auf einer SPS-basierten Automatisierungsplattform wurde im Rahmen des Forschungsprojekts *i-Protect* [P2] durchgeführt. Auf Basis dieser Ergebnisse werden der gerichtete Überstromzeit- sowie der Distanzschutz auf die Systemplattform portiert und dort weiterentwickelt. Im ersten Schritt erfolgt die Implementierung dieser Funktionen aus Gründen des Entwicklungsablaufs auf Ebene des Linux-Betriebssystems. Mit einer derartigen Umsetzung kann zwar die grundsätzliche Funktionalität der implementierten Schutzfunktionen hinsichtlich Genauigkeit und Koordination gezeigt werden. Allerdings lassen sich aufgrund des nicht gegebenen Zeitdeterminismus auf Betriebssystemebene keine quantitativen Aussagen zum Zeitverhalten treffen. Abbildung 4.5 zeigt den grundsätzlichen Datenfluss im Zusammenhang mit einer Schutzfunktion. Die von der Messhardware erfassten Messwerte werden, nach Skalierung über hinterlegte Wandlerverhältnisse, in einem logischen Knoten der Klasse MMXU abgelegt und von dort weiter der Schutzfunktion zugeführt. Eine Distanzschutzfunktion liest beispielsweise aus einem logischen Knoten der Klasse PDIS die

hinterlegten Schutzparameter zum Aufspannen des Distanzschutzpolygons⁹. Anhand des Distanzschutzpolygons wird geprüft, ob für die gegebenen Messwerte ein Fehlerereignis vorliegt. Entsprechende Anrege- und Auslösesignale werden in den logischen Knoten PDIS zurückgeschrieben. Eine Funktion *Schutzkoordination* liest die Anrege- und Auslösesignale ggf. mehrerer vorhandener Schutzfunktionen und ermittelt daraus ein Gesamtsignal, das in einem logischen Knoten PTRC gespeichert und schließlich über einen binären Ausgang der Hardware ausgegeben wird.

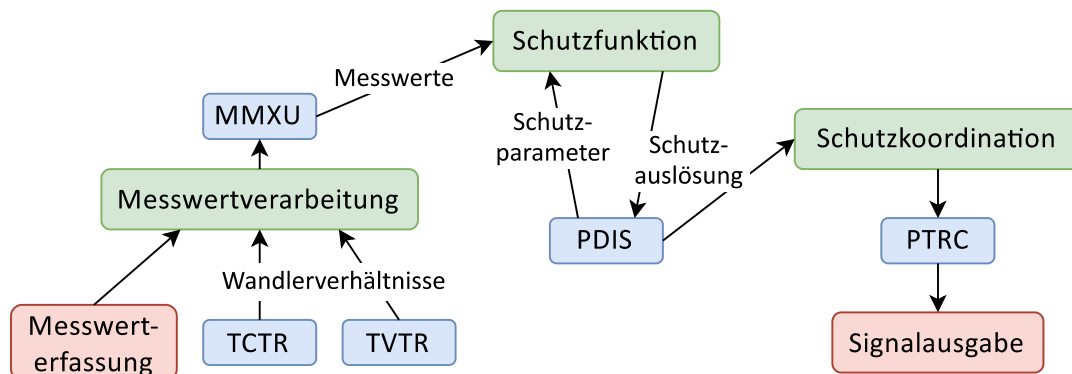


Abbildung 4.5: Datenfluss im Zusammenhang mit einer Schutzfunktion

Da auf Betriebssystemebene platzierte Schutzfunktionen für einen produktiven Einsatz ungeeignet sind, wird hier auf weitergehende Erläuterungen zu deren Implementierung verzichtet.

Wie in Kapitel 4.1 erwähnt, verfügt das SoM der Systemplattform über einen bisher ungenutzten Cortex-M4-Co-Prozessor. Bei Verzicht auf ein Betriebssystem bzw. die Nutzung eines Echtzeitbetriebssystems können hier Algorithmen zeitdeterministisch ausgeführt werden. Daher erfolgt im zweiten Schritt die Implementierung von Schutzfunktionen auf dieser Ebene. Grundlegende Arbeiten hierzu erfolgten in [S6].

Voraussetzung für eine Schutzfunktion ist, dass ihr sowohl die Messwerte in deterministischer Zeit zur Verfügung stehen, als auch, dass ihre Reaktionen etwa in Form von Auslösesignalen in deterministischer Zeit ausgegeben werden. Wie in Abbildung 4.3 dargestellt, ist der serielle Kanal des DAQ-/IO-Modul bisher direkt an das System des Cortex-A7-Prozessors angebunden. Würden die Messwerte von diesem System lediglich an den Co-Prozessor weitergeleitet, wäre kein Zeitdeterminismus gegeben. Vielmehr ist eine direkte Anbindung des DAQ-/IO-Modul an den Co-Prozessor erforderlich.

An dieser Stelle kommt der sogenannte *Device Tree* (DT, [106, 107]) ins Spiel: Hierbei handelt es sich um eine Datenstruktur, mit der die Hardware eines Rechners in

⁹ Der hier umgesetzte Distanzschutz basiert auf dem Prinzip der Impedanzpolygon-Charakteristik mit Lastaussparung. Details hierzu werden im Rahmen der Verifikation in Kapitel 5.1 erläutert.

wohldefinierter Form beschrieben werden kann. Das Konzept des Device Trees wird vor allem bei eingebetteten Systemen verwendet, die beispielsweise auf der ARM-Architektur basieren. Die internen Kommunikationsbusse dieser Systeme unterstützen häufig keine Bus-Enumeration, d. h. sie verfügen über keine Möglichkeit, die an den Bus angeschlossenen Komponenten zu identifizieren. Anstatt die Hardwarekonfiguration (Speichergröße und -layout, Schnittstellen etc.) fest im Betriebssystemkern (*Kernel*) zu definieren, wird diese dem Kernel flexibel als Datenstruktur übergeben. Dies ermöglicht die Unterstützung einer größeren Anzahl an Hardwarevarianten durch ein und dasselbe Kernelabbild. PC-basierte Systeme hingegen nutzen keinen Device Tree. Die hier eingesetzten Busse wie PCI (engl.: Peripheral Component Interconnect) unterstützen eine Bus-Enumeration, so dass die Systemfirmware (engl.: Basic Input/Output System, BIOS oder engl.: Unified Extensible Firmware Interface, UEFI) die an den Bus angeschlossenen Komponenten identifizieren und diese Information an das Betriebssystem weiterreichen kann.

Im hier dargestellten konkreten Anwendungsfall wird die Hardwarekonfiguration des SoM über den Device Tree konfiguriert und an den Kernel des Linux-Betriebssystems bzw. an das ggf. auf dem Cortex-M4-Co-Prozessor ausgeführte Echtzeitbetriebssystem FreeRTOS [108] übergeben. Über den Device Tree wird auch konfiguriert, welche IO-Pins (engl.: General Purpose Input/Output, GPIO) des SoM mit welchen Funktionen belegt sind und welchem Prozessor sie zur Verfügung stehen. So kann die serielle Schnittstelle (engl.: Universal Synchronous/Asynchronous Receiver Transmitter, USART), an die das DAQ-/IO-Modul angeschlossen ist, auf einen der Prozessoren gemappt werden. Während das DAQ-/IO-Modul bisher auf den Cortex-A7-Prozessor gemappt war, muss es durch eine entsprechende Anpassung am Device Tree nun auf den Cortex-M4-Co-Prozessor gemappt werden. Dadurch kann eine auf dem Cortex-M4-Co-Prozessor ausgeführte Applikation nun direkt auf das DAQ-/IO-Modul zugreifen und so direkt die erfassten Messwerte erhalten.

In einem zweiten Schritt muss eine Kommunikation zwischen beiden Prozessorsystemen hergestellt werden. Diese ermöglicht, dass die Applikation auf dem Hauptsystem (Cortex-A7-Prozessor) die Applikation auf dem Co-System (Cortex-M4-Co-Prozessor) instruieren kann, indem sie etwa Parameter für eine dort ausgeführte Schutzfunktion übermittelt, und dass umgekehrt die vom DAQ-/IO-Modul erfassten Messwerte vom Co- an das Hauptsystem weitergereicht werden können, so dass diese dort auch weiterhin für nicht zeitdeterministische Funktionen zur Verfügung stehen.

Das eingesetzte SoM ist ein Beispiel für ein heterogenes Multiprozessorsystem, d. h. ein System mit mehreren, jedoch typverschiedenen Prozessoren. Eine verbreitete Methode zur Kommunikation zwischen den Prozessoren eines solchen Systems

besteht in der Nutzung eines gemeinsamen, geteilten Speicherbereichs. Ein Verfahren, mit dem eine solche Kommunikation implementiert werden kann, ist Remote Processor Messaging (RPMsg [109, 110]). Das RPMsg-Protokoll ist in die in Abbildung 4.6 dargestellten drei Schichten *Transport Layer*, *Media Access Layer (MAC)* und *Physical Layer* unterteilt.

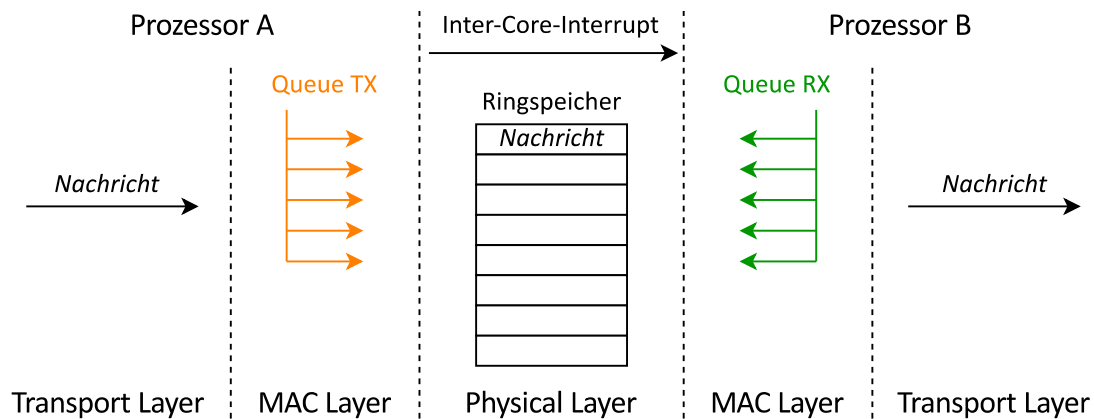


Abbildung 4.6: Schichten des Remote Processor Messaging

Über einen entsprechenden Treiber wird eine zu übermittelnde Nachricht dem Transport Layer übergeben. Da beliebig viele RPMsg-Kanäle existieren können (beispielsweise zur Kommunikation mit mehreren Prozessorkernen), wird hierbei u. a. auch eine Zieladresse angegeben, die den RPMsg-Kanal identifiziert. Der Transport Layer leitet die Nachricht an den MAC-Layer weiter. Letzterer setzt sie so um, dass sie in einen Ringspeicher abgelegt werden kann. Durch Nutzung der sog. „single-writer single-reader“-Technik wird sichergestellt, dass niemals gleichzeitig von mehreren Prozessorkernen auf dieselben Blöcke des Ringspeichers geschrieben wird. Der MAC-Layer legt die Nachricht schließlich im Ringspeicher, der als geteilter Speicher den Physical Layer darstellt, ab. Abschließend wird der entfernte Prozessorkern über einen Interrupt darüber informiert, dass eine neue Nachricht vorliegt. Der MAC-Layer des entfernten Prozessorkerns entnimmt die Nachricht aus dem Ringspeicher, leitet sie an den dortigen Transport Layer weiter, von wo sie der Applikation übergeben wird.

Abbildung 4.6 zeigt eine unidirektionale Datenübertragung zwischen zwei Prozessorkernen. Für eine bidirektionale Datenübertragung müssen dementsprechend zwei RPMsg-Kanäle mit jeweils eigenem Ringspeicher im geteilten Speicher eingerichtet werden.

Abbildung 4.7 zeigt die gegenüber Abbildung 4.3 so modifizierte Systemarchitektur, dass eine zeitdeterministische Ausführung von Funktion ermöglicht wird.

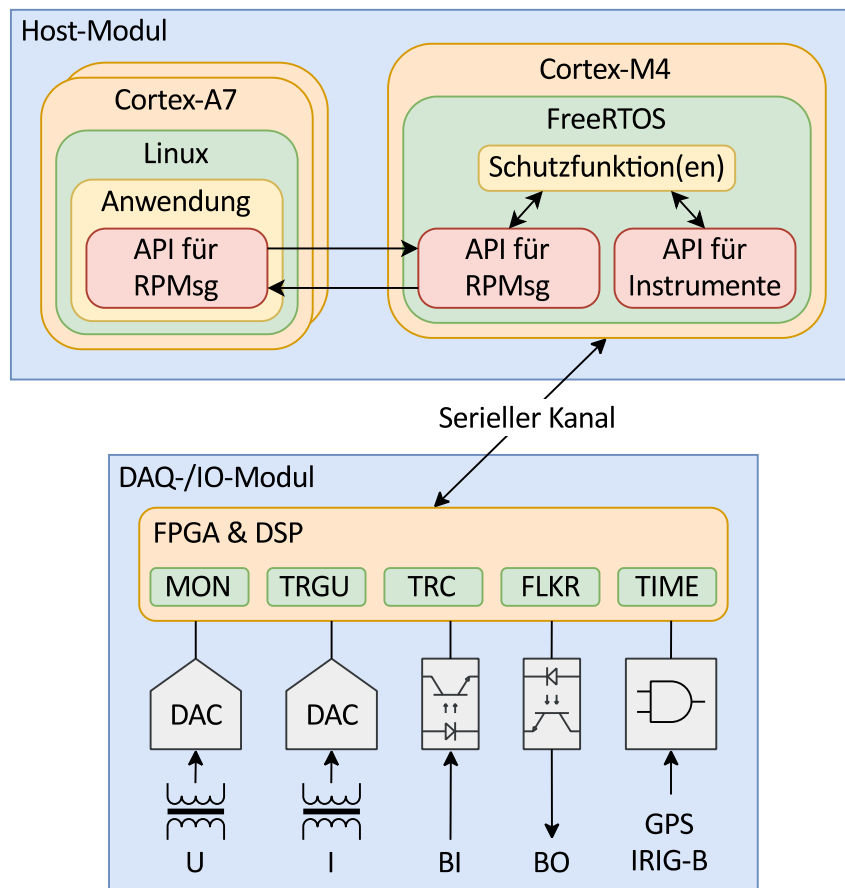


Abbildung 4.7: Modifizierte Systemarchitektur

Zur Datenübertragung vom Haupt- zum Co-Prozessorsystem wird ein Protokoll gemäß Tabelle 4.2 definiert. Hierüber können die von einer Schutzfunktion (bisher Überstromzeit- und Distanzschutz) benötigten Parameter sowie Befehle zum Starten und Beenden des Schutzalgorithmus übertragen werden. Der eigentliche Schutzalgorithmus ist gegenüber der früheren Implementierung zur Ausführung auf dem Hauptsystem unverändert, lediglich die Anbindung an die Eingangsmesswerte und die Ausgabe der Schutzreaktionen wurden entsprechend angepasst.

Tabelle 4.2: Befehle zwischen Haupt- und Co-Prozessorsystem

| Befehl | Funktion | Parameter |
|--------|------------|---|
| start | PDIS, PTOC | Funktionsinstanz |
| stop | PDIS, PTOC | Funktionsinstanz |
| config | PDIS | Funktionsinstanz, Schutzparameter (AngDir, AngLod, AngNegRes, LinAng, RisLod, RisPhRch, X1, OpDITmms, RsDITmms) |
| | PTOC | Funktionsinstanz, Schutzparameter (StrVal, OpDITmms, RsDITmms) |

In Kapitel 5.1 wird im Rahmen einer Schutzprüfung der auf Ebene des Co-Prozessors implementierte Distanzschutz verifiziert.

Neben Schutzfunktionen gibt es ein breites Anwendungsgebiet für Überwachungs-, Steuerungs- und Automatisierungsfunktionen. Aus der Vielzahl möglicher Smart-Grid-Funktionen werden unter anderem verschiedene Varianten der Spannungsregelung und die Netzzustandsschätzung auf der Systemplattform implementiert. Im Gegensatz zu lokalen Schutzfunktionen erfordern diese beiden Funktionen die Kommunikation von Messwerten und Steuerbefehlen zwischen verteilten Geräten. Bei der Implementierung dieser Funktionen müssen darüber hinaus auch Aspekte der Robustheit und Belastbarkeit berücksichtigt werden.

4.3.2 Spannungsregelung

Die Untersuchung und Implementierung verschiedener Verfahren und Algorithmen zur Spannungsregelung in Verteilnetzen erfolgte im Rahmen von [111].

Ein Konzept zur zentralen Spannungsregelung wurde in [B6] und [B19] eingeführt und in [B16] weiterentwickelt und verifiziert. Gegenüber einer Spannungsregelung, die lediglich lokale Messwerte berücksichtigt, bietet eine Spannungsregelung mit dezentralen Messungen im Sinne einer Weitbereichsregelung den Vorteil, dass sich auch Spannungsvarianzen an den Enden von Netzzweigen und in Netzausläufern berücksichtigt lassen. Im Rahmen der Untersuchungen wurde ein Niederspannungsnetz mit zentralem, stufbaren Ortsnetztransformator und zwei Segmenten mit Lasten und dezentralen Einspeisern betrachtet. In jedem Netzsegment werden die Spannungsmesswerte erfasst und an ein zentrales System übermittelt. Weiterhin können die beiden Netzsegmente an ihren Enden über einen fernsteuerbaren Schalter gekoppelt werden.

Auf Basis der dezentral erfassten Messwerte führt der auf dem zentralen System ablaufende Algorithmus definierte Maßnahmen durch. Hierzu zählen Stufung des Transformators, Blindleistungssteuerung der dezentralen Erzeugungsanlagen sowie Netzrekonfiguration. Sobald die Spannungen beider Netzsegmente die in [112] definierte zulässige Abweichung von $\pm 10\%$ in dieselbe Richtung überschreiten, wird der Transformator gestuft. Überschreiten die Spannungen beider Netzsegmente die zulässige Abweichung in gegensätzliche Richtungen, wird eine Netzrekonfiguration durchgeführt und beide Netzsegmente werden durch den Schalter gekoppelt. Überschreitet nur die Spannung eines Netzsegments die zulässige Abweichung, führen die vorgenannten Maßnahmen zu keinem befriedigenden Ergebnis. In diesem Fall wird in demjenigen Netzsegment, in dem die Spannungsbandverletzung vorliegt, eine Blindleistungsregelung der dezentralen Erzeugungsanlagen durchgeführt.

Dieses Konzept ist prinzipiell auf Netze mit mehr als zwei Segmenten erweiterbar, allerdings führt dies zu einer höheren Komplexität der Maßnahmenmatrix. Auch sind

bei der Durchführung von Netzrekonfigurationen in realen Netzen Aspekte wie die Kurzschlussleistung und der Netzschutz zu berücksichtigen.

Die zentrale Spannungsregelung stellt hohe Ansprüche an die Zuverlässigkeit des zentralen Systems und der Kommunikation. Wenn etwa die Kommunikation zu einem dezentralen Messsystem ausfällt, geht das gesamte System in einen kritischen Betriebszustand über. Daher wurde in [B6] das Konzept einer hierarchisch-koordinierten Spannungsregelung eingeführt und in [B15] weiterentwickelt und verifiziert. Bei diesem Ansatz wird in jedes Netzsegment ein Zwischensystem eingefügt, das als lokaler Koordinator für die Überwachung und Regelung seines Netzsegments zuständig ist. Das Zwischensystem kommuniziert sowohl mit den dezentralen Systemen seines Netzsegments als auch mit dem überlagerten zentralen System. Regelungsaufgaben, die lediglich ein Netzsegment betreffen, werden durch das jeweilige Zwischensystem koordiniert und von den dortigen dezentralen Systemen gelöst. Überlagerte Regelungsaufgaben, die mehrere Netzsegmente betreffen, werden hingegen durch das zentrale System koordiniert, indem dieses entsprechende Maßnahmen an die einzelnen Zwischensysteme befiehlt.

In [B14] werden die Systeme der hierarchisch-koordinierten Spannungsregelung um Funktionen zur Detektion von Netzfehlern in den jeweils unterlagerten Netzsegmenten erweitert. Sobald ein Netzsegment fehlerbedingt abgeschaltet wird, werden die Messwerte der dortigen Systeme sowie die steuerbaren Flexibilitäten bei der Ausführung der überlagerten Spannungsregelung ausgespart.

In [113, 114] wird ein Spannungsregelungsalgorithmus implementiert, der sich bei der Bestimmung der auszuführenden Maßnahmen auf die Ergebnisse einer Berechnung der Leistungsflussoptimierung stützt. Während der OPF hierbei noch zentral abläuft, erfolgt in [115] eine Erweiterung hin zu einer hierarchischen Lösung, um Problemen, die durch einen Ausfall von Kommunikationsverbindungen oder einzelner Messstellen entstehen, entgegenzutreten.

4.3.3 Netzzustandsschätzung

Eine Voraussetzung für den sicheren und zuverlässigen Betrieb zukünftiger Verteilnetze ist die Fähigkeit, die Systemzustände des Netzes, die sich durch dynamische Ereignisse im Netz ständig ändern, kontinuierlich abschätzen zu können. Ein hierfür geeignetes Verfahren stellt die Online-Netzzustandsschätzung dar. Grundlage der Netzzustandsschätzung ist die messtechnische Erfassung der betragsmäßigen Knotenspannungen sowie der komplexen Knoten- und Strangströme bzw. -leistungen an verteilten Knoten im Netz. Mittels Netzgleichungen werden Gleichungssysteme aufgestellt und gelöst, um den Netzzustand in Form komplexer

Knotenspannungen zu ermitteln. Während Netzzustandsschätzungen schon lange Zeit in den Übertragungsnetzen eingesetzt werden, stand dem Einsatz in den Verteilnetzen bisher die dort fehlende messtechnische Ausstattung entgegen. Mit einer Zunahme an Mess-, Steuerungs- und Automatisierungstechnik in den Verteilnetzen ergibt sich nun jedoch zunehmend die Möglichkeit, auch dort Verfahren der Netzzustandsschätzung – mit zumindest teilbeobachtbaren Netzen – umzusetzen.

Im Gegensatz zu einer einfachen Leistungsflussberechnung werden bei der Netzzustandsschätzung zusätzliche Verfahren angewandt, um den Einfluss stochastischer, normalverteilter Messfehler auf den ermittelten Netzzustand zu verringern und somit die Schätzgüte zu verbessern. Dies ist allerdings nur in überbestimmten Netzen, also Netzen, in denen mehr als die minimale Anzahl an unabhängigen Messungen vorhanden ist, möglich. Der am weitesten verbreitete Algorithmus zur Optimierung der Schätzgüte ist das Verfahren der gewichteten kleinsten Quadrate (engl.: Weighted Least Squares, WLS). Dementsprechend basiert die an dieser Stelle durchgeführte Implementierung eines Algorithmus der Netzzustandsschätzung ebenfalls auf diesem Verfahren.

Der Algorithmus der Netzzustandsschätzung benötigt dezentral erfasste Messwerte für die Berechnung des Netzzustands. Diese Messwerte werden von im Feld verteilten Messstellen an ein zentrales System übermittelt und dort verarbeitet. Die Messdatenerfassung erfordert keine hochpräzise, aber dennoch eine ausreichende Zeitsynchronisation, da der Netzzustand immer nur eine Momentaufnahme darstellt. Sind Messwerte zu asynchron, dann weichen die Netzzustände, die den erfassten Messwerten zugrunde liegen, unter Umständen zu weit voneinander ab. Im Rahmen der Implementierung und Verifikation im Rahmen dieser Arbeit werden Abweichungen von bis zu einer Minute toleriert.

Mathematische Grundlagen

Die Zustandsberechnung unter Verwendung des WLS-Ansatzes löst die in Gleichung (4.1) angegebene Zielfunktion $J(x)$. Darin stellt z den Messvektor dar, der die erfassten Messgrößen enthält. x ist der Zustandsvektor und somit der Netzzustand des Systems, der in die Messfunktion $h(x)$ eingesetzt wird. In der Messfunktion ist die Abhängigkeit zwischen Zustandsvektor und Messvektor beschrieben. Demnach ist die Abweichung zwischen z und $h(x)$ zu minimieren. Dabei können einzelne Messwerte mit der Gewichtungsmatrix W unterschiedlich stark gewichtet werden, um beispielsweise unterschiedliche Genauigkeiten bei der Messwerterfassung zu berücksichtigen.

$$J(x) = [z - h(x)]^T \cdot W \cdot [z - h(x)] \quad (4.1)$$

Die Minimierung der Funktion $J(x)$ führt zu einer höheren Schätzungsgüte der Netzzustandsschätzung. Normalverteilte Messfehler können durch die WLS-basierte Netzzustandsschätzung kompensiert werden. Systematische Fehler sind aber auch mit dieser Form der Netzzustandsschätzung problematisch.

Die Normalgleichung aus Gleichung (4.2) der Netzzustandsschätzung ergibt sich durch weitere Umformungen und unter Berücksichtigung der Bedingungen für einen Extremwert der Funktionswerte. Darin werden in der Jakobi-Matrix $H(x)$ die partiellen Ableitungen der Messfunktion $h(x)$ beschrieben. In einem iterativen Prozess wird die Normalgleichung nach Δx^{k+1} bis zur Konvergenz gelöst. Im Falle der Konvergenz ist der im letzten Iterationsschritt bestimmte Zustandsvektor x der optimierte Netzzustand des Gesamtsystems.

$$H^T(x^k) \cdot W \cdot H(x^k) \cdot \Delta x^{k+1} = H^T(x^k) \cdot W \cdot [z - h(x^k)] \quad (4.2)$$

Optimale Messpunktplatzierung

Eine Zustandsschätzung basiert auf Messungen von Spannungen, Strömen und Leistungen an verschiedenen Knoten im Netz. Mittels dieser Messwerte werden die Werte der nicht gemessenen Knoten durch Anwendung des oben beschriebenen mathematischen Verfahrens berechnet. Für eine möglichst gute Zustandsschätzung wäre es wünschenswert, Messungen von möglichst vielen Knoten zur Verfügung zu haben. Aus praktischen und wirtschaftlichen Gründen ist es jedoch nicht möglich, an jedem Knoten Geräte zur Messwerterfassung zu installieren. Jedoch ist eine gewisse Anzahl an Messungen und eine geschickte Messstellenplatzierung notwendig, um eine Beobachtbarkeit des Netzes zu erreichen. Beobachtbar ist ein Netz dann, wenn die vorhandenen, unabhängigen Messungen ausreichen, um für alle Knoten die komplexen Knotenspannungen bestimmen zu können. Zur Bewertung der Beobachtbarkeit und zur Identifikation der für die Beobachtbarkeit kritischen Netzknoten kommt ein im Rahmen paralleler und studentischer Arbeiten entwickeltes Softwarewerkzeug zum Einsatz. Exemplarisch wurde dieses Werkzeug auf das modifizierte Cigré-Niederspannungs-Benchmark-Netz in Abbildung 4.8 angewandt, um diejenigen Knoten zu identifizieren, die für eine vollständige Beobachtbarkeit des Netzes messtechnisch erfasst werden müssen. Die hierzu als erforderlich identifizierten Knoten sind in der Abbildung orange markiert.

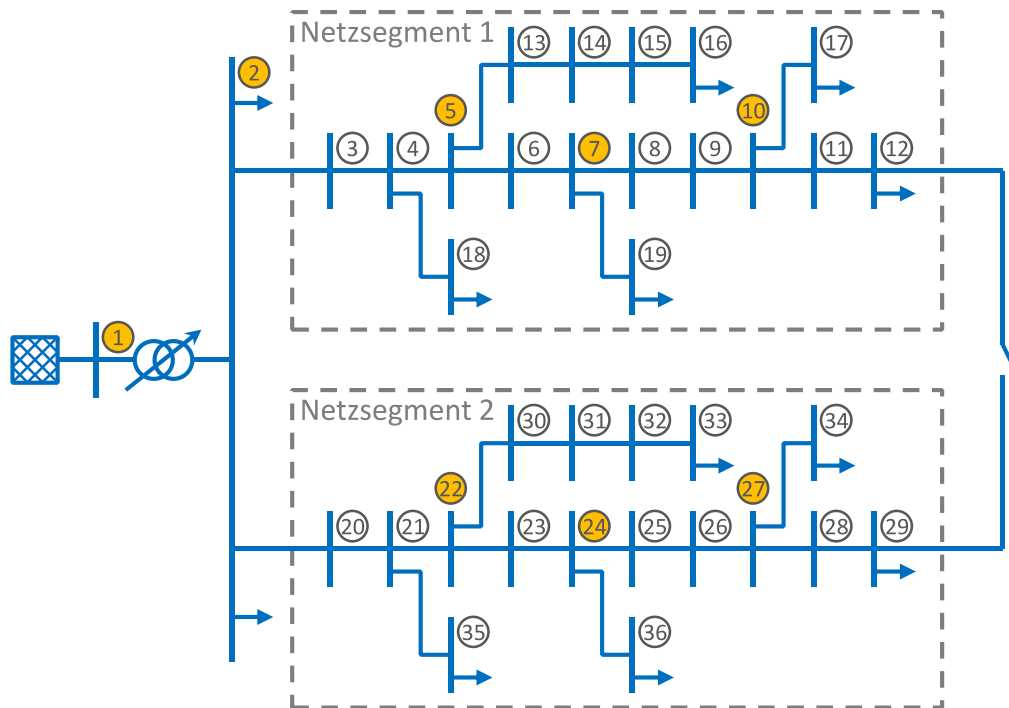


Abbildung 4.8: Optimale Platzierung der Messungen

Bei der Beobachtbarkeitsanalyse sind sogenannte Nullknoten von großer Bedeutung. Nullknoten heißt in diesem Zusammenhang, dass die Knotenleistung an diesem Knoten Null beträgt. Diese Information kann in die Netzzustandsschätzung als virtuelle Messung einfließen. Insgesamt kann festgehalten werden, dass insbesondere Verzweigungen, die nicht Nullknoten sind, für eine Ausstattung mit Messtechnik von großer Bedeutung sind. Weiterhin sind Strangenden für die Netzzustandsschätzung weniger relevant, sofern der vorletzte Strangknoten messtechnisch erschlossen ist. Nichtsdestotrotz ist eine hohe Durchdringung mit Messtechnik in den Netzen zumindest unter dem Aspekt der Schätzgüte wünschenswert. Weitergehende Betrachtungen und Verfahren zur möglichst optimalen Platzierung von Messtechnik und allgemeiner von Smart-Grid-Automatisierungssystemen in Verteilnetzen werden in [116] vorgestellt.

Implementierung

Die Implementierung einer statischen Zustandsschätzung auf Basis des WLS-Verfahrens ist bereits aus [117] als Softwarewerkzeug in Matlab, bestehend aus Kernalgorithmus und grafischer Bedienoberfläche, verfügbar. Diese Implementierung ist darauf ausgelegt, Messwerte, die in einer Netzsimulation in der Software DlgSILENT PowerFactory berechnet wurden, sowie die zugehörigen Topologieinformationen zu verarbeiten. Die entsprechenden Daten werden als Testdateien aus DlgSILENT PowerFactory exportiert und anschließend über das Softwarewerkzeug eingelesen. Über die grafische Bedienoberfläche können darüber hinaus Parameter wie die

Konvergenzschranke oder die maximale Anzahl an Iterationen konfiguriert und einzelne Softwarefunktionen ausgeführt werden.

Für tiefergehende Analysen kann der Anwender aus der Gesamtzahl aller Messwerte diejenigen Messwerte spezifizieren, die bei der Zustandsschätzung berücksichtigt werden sollen. Vor der eigentlichen Zustandsschätzung wird zunächst eine Beobachtbarkeitsanalyse durchgeführt. Diese prüft, ob mit den zuvor gewählten Messwerten eine vollständige Beobachtbarkeit des Netzes gegeben ist. Nur in diesem Fall kann die anschließende Zustandsschätzung sinnvolle Ergebnisse liefern. Auf diese Weise lässt sich der Einfluss der Messwerte auf die Beobachtbarkeit gezielt untersuchen. Außerdem können die Messwerte mit künstlichen Messfehlern überlagert werden. Da im Rahmen der Zustandsschätzung eine Detektion und Lokalisierung großer Messfehler durchgeführt wird, kann so der Einfluss von Messfehlern auf das Detektions- und Lokalisierungsergebnis untersucht werden.

Um eine Ausführung der Netzzustandsschätzung auf der Systemplattform zu ermöglichen, wurde der bestehende Algorithmus im Rahmen studentischer Tätigkeiten vollständig neu in C++ implementiert. Wie bei der bisherigen Matlab-Implementierung war zunächst jedoch lediglich die Verarbeitung statischer Messwerte auf Basis von Dateien möglich.

Im Rahmen dieser Arbeit erfolgt die vollständige Einbettung des Algorithmus der Netzzustandsschätzung in die Implementierungsumgebung. Hierbei sind insbesondere Anpassungen an den Schnittstellen des Algorithmus zu den Eingangsdaten und den Berechnungsergebnissen erforderlich. Durch entsprechende Erweiterungen ist es nun möglich, dass die Netzzustandsschätzung ihre Eingangsdaten alternativ zu statischen Dateien dynamisch aus logischen Knoten eines IEC-61850-Online-Datenmodells einlesen kann. Somit kann die Netzzustandsschätzung kontinuierlich für Messwerte ausgeführt werden, die von in einem elektrischen Netz verteilten Messgeräten erfasst und über ein entsprechendes Kommunikationsnetzwerk übertragen werden. Umgekehrt können die Berechnungsergebnisse ebenso in logische Knoten eines IEC-61850-Online-Datenmodells zurückgeschrieben werden. Damit ist es möglich, dass weitere Smart-Grid-Funktionen wie etwa Regelungsfunktionen die Ergebnisse einer Netzzustandsschätzung für ihre Zwecke nutzen. Durch die Möglichkeit der Kommunikation per IEC 61850 ist es dabei unerheblich, ob diese Funktionen lokal auf dem System, das die Netzzustandsschätzung durchführt, platziert sind, oder auf kommunikationstechnisch angebundene entfernte Systemen. Der Vollständigkeit halber soll erwähnt werden, dass darüber hinaus auch eine Schnittstelle zu SQLite-Datenbanken umgesetzt ist, so dass auch hieraus Messwerte eingelesen und Berechnungsergebnisse dorthin ausgegeben werden können.

Für die Berechnung des Netzzustands ist die Vorgabe von Topologieinformationen essenziell, daher müssen diese Informationen als Eingangsdaten vorliegen. Im ersten Implementierungsschritt wurden diese Daten statisch vorgegeben. In der Realität kann sich die Topologie jedoch sehr plötzlich ändern, etwa aufgrund eines fehlerbedingten Ausfalls einer Leitung oder auch aus betrieblichen Gründen. Nicht immer ist solch eine Änderung mit den vorhandenen Algorithmen für die Fehlererkennung detektierbar, da es sich hierbei nicht um einen normalverteilten Messfehler, sondern vielmehr um einen systematischen Fehler handelt. Daher müssen Topologieänderungen nachgepflegt werden. Um diesem Umstand Rechnung zu tragen, wird der Algorithmus in einem zweiten Implementierungsschritt so erweitert, dass er vor jeder Berechnung nach Bedarf die Topologiedaten dynamisch neu einlesen kann. Allerdings werden zukünftig auch Algorithmen benötigt, die eine Änderung in der Topologie selbstständig erkennen und, wenn möglich, auch direkt bestimmen können, welches die neue Topologie ist. In [118] wird ein solcher, auf Messwerten basierender Algorithmus vorgestellt. Für den Fall, dass keine Topologie bestimmt werden kann, müssen zumindest die Regelungsfunktionen, die sich auf die Berechnungsergebnisse der Netzzustandsschätzung stützen, unterbrochen werden.

4.3.4 Weitere Funktionen und Algorithmen

Neben den in den vorangegangenen Kapiteln beschriebenen Schutz-, Regelungs- und Überwachungsfunktionen wurden im Rahmen paralleler Arbeiten weitere Funktionen und Algorithmen zur Ausführung auf der Systemplattform implementiert. Im Folgenden soll ein kurzer Überblick hierüber gegeben werden.

In [B21] wurde ein Algorithmus zur modellprädiktiven Regelung (engl.: Model Predictive Control, MPC) eingeführt und in [119] weiterentwickelt. Als exemplarischer Anwendungsfall wurde der netzdienliche Einsatz der Ladevorgänge von Elektrofahrzeugen im Sinne einer Behebung bzw. Vermeidung von Netzengpässen betrachtet. Hierzu bestimmt der Algorithmus auf Basis eines Prädiktions- und Optimierungsverfahrens die optimalen Stellgrößen der Ladevorgänge. Mittels simulativer Untersuchungen konnte die Funktionsfähigkeit des Algorithmus gezeigt werden.

In weiteren Arbeiten wurden Verfahren des maschinellen Lernens (engl.: Machine Learning, ML) betrachtet und angewendet. In [120] dient im Normalfall ein OPF dazu, im Netz verteilte dezentrale Energieeinspeiser koordiniert zu regeln. Im Störfall, etwa bei einer Kommunikationsunterbrechung, übernimmt eine lokale Regelung als Rückfallebene die Ansteuerung einzelner dezentraler Energieeinspeiser. Diese lokale Regelung wurde zuvor offline mittels Verfahren des maschinellen Lernens für alle Betriebspunkte der einzelnen Einspeiser trainiert. Die dabei ermittelten Beziehungen

zwischen Netzspannung und Sollwerten der Einspeiser werden in einer Look-Up-Tabelle abgelegt und von der lokalen Regelung im Störfall genutzt. Der Algorithmus wurde im Rahmen von Hardware-in-the-Loop-Simulationen verifiziert.

In [121] wird ein Verfahren zur Teilnetzerkennung und Inselnetzbildung in Mittelspannungsnetzen vorgestellt, welches auf einer Reihe von Algorithmen beruht: Ein auf maschinellem Lernen basierender Algorithmus dient zur Erkennung einer ungesteuerten Teilnetzbildung und nutzt hierzu Messwerte von im Netz verteilten zeitsynchronisierten Zeigermessgeräten. Ein weiterer Algorithmus dient dazu, auf Basis von RMS-Messwerten die aktuelle Netztopologie zu bestimmen. Sobald der erste Algorithmus eine Teilnetzbildung erkennt, bestimmt ein weiterer Algorithmus, ausgehend von der aktuellen Netztopologie, die Größe des gebildeten Teilnetzes und die beteiligten Netzbetriebsmittel. Außerdem wird ermittelt, ob für das gebildete Teilnetz ein stabiler Inselnetzbetrieb möglich ist. In diesem Fall überführt ein vierter Algorithmus, nach Freigabe durch den Netzbetreiber, das Teilnetz in einen Inselbetrieb. Hierbei werden entsprechende Voraussetzungen geschaffen, etwa die Bestimmung eines Führungskraftwerks. Die einzelnen Algorithmen werden mittels Docker [122] als Container virtualisiert und laufen parallel auf einer physischen Hardware ab. Zur Verteilung und Verwaltung der einzelnen Container dient Kubernetes [123]. Die Algorithmen wurden im Rahmen von Hardware-in-the-Loop-Simulationen verifiziert.

4.4 Kommunikation und Datenhaltung

Dieses Kapitel beschreibt die auf der Systemplattform realisierten Kommunikationsverfahren sowie eine umgesetzte Datenbanklösung zur Langzeitaufzeichnung von beispielsweise Messwerten und Berechnungsergebnissen.

4.4.1 UDP

Die in Kapitel 4.3.2 beschriebene zweistufige Spannungsregelung war eine der ersten auf der Systemplattform implementierten Funktionen. Hierdurch ergab sich frühzeitig die Notwendigkeit, eine Kommunikation zwischen mehreren Systeminstanzen zu realisieren. Allerdings erfolgten die ersten Implementierungsarbeiten noch auf einer Vorgängergeneration der in Kapitel 4.1 beschriebenen Systemplattform. Diese Generation der Systemplattform nutzte als Betriebssystem Windows CE in der Version 6. Hierbei handelt es sich um ein Betriebssystem, das im Jahre 2006 veröffentlicht wurde und daher bereits veraltet war. Zwar ließ sich das ACSI des IEC-61850-Kommunikationsstacks für dieses Betriebssystem erstellen. Damit war es den implementierten Funktionen möglich, Online-Datenmodelle gemäß IEC 61850-7-2 zu nutzen. Um auch eine Kommunikation gemäß IEC 61850-8-1 zu ermöglichen, wäre es

jedoch zusätzlich erforderlich gewesen, das SCSM für dieses Betriebssystem zu erstellen, was allerdings umfangreiche Anpassungen am Quellcode des SCSM notwendig gemacht hätte. Da zu diesem Zeitpunkt bereits mittelfristig ein Wechsel auf die aktuelle Systemplattform absehbar war, wurde auf die Umsetzung des SCSM für die erste Generation der Systemplattform verzichtet.

Um dennoch kurzfristig eine Kommunikation zwischen Systeminstanzen zu ermöglichen, wurde eine eigene Kommunikation auf Basis von UDP (engl.: User Datagram Protocol [124]) implementiert. Bei UDP handelt es sich um ein verbindungsloses Netzwerkprotokoll. Der Vorteil ist, dass kein Verbindungsaufbau und -abbau zwischen den Teilnehmern durchgeführt werden muss, wodurch die Programmlogik vereinfacht wird. Nachteilig ist, dass es sich um ein nicht-zuverlässiges und ungesichertes Netzwerkprotokoll handelt. Weder ist garantiert, dass gesendete Pakete vollständig beim Empfänger ankommen, noch, dass Pakete in der Reihenfolge ankommen, in der sie gesendet wurden, noch, dass Pakete nicht mehrfach beim Empfänger ankommen. Beim Einsatz in einem lokalen Netzwerk wie dem des ie³-Labors kann jedoch davon ausgegangen werden, dass die genannten nachteiligen Effekte, wenn überhaupt, nur sehr selten auftreten. Wenn erforderlich, werden zusätzliche Schutzmaßnahmen zur Überprüfung der Datenübertragung, wie die Prüfung eines Zeitstempels, auf Anwendungsebene implementiert.

Die Kommunikation basiert auf einem Publisher-Subscriber-Modell: Ein System, das Daten versenden möchte, ist der Publisher und sendet die Daten als Broadcast in das Netzwerk. Ein beliebiges System im Netzwerk, das Daten empfangen möchte, ist der Subscriber und abonniert die Daten. Die Struktur der übertragenen Nutzdaten orientiert sich dabei an den logischen Knoten gemäß IEC 61850-7-4, d. h. es werden pro Datenpaket all diejenigen Werte übertragen, die zu den Datenattributen eines logischen Knotens im Online-Datenmodell gehören. Für den logischen Knoten MMXU sind dies beispielsweise sämtliche Messwerte (Spannungen und Ströme mit Betrag und Phase, Wirk-, Blind- und Scheinleistung etc.) inkl. Zeitstempel und Qualitätsattributen. Zusätzlich zu den Werten werden als Header die Logische-Knoten-Klasse sowie die volle Logische-Knoten-Referenz übertragen. Hiermit ist es dem Subscriber möglich, die Herkunft eines Datenpakets zu bestimmen und damit nur Datenpakete, die von Interesse sind, zu verarbeiten. Auf Seite des Subscribers werden die empfangenen Werte in das lokale Online-Datenmodell übernommen, so dass Smart-Grid-Funktionen darauf zugreifen können, ohne einen Unterschied zu sehen, ob es sich um lokale Werte handelt oder um Werte, die per Netzwerk kommuniziert werden. Tabelle 4.3 zeigt die Struktur eines Datenpakets beispielhaft für einen logischen Knoten der Klasse *MMXU*.

Tabelle 4.3: Datenpaketstruktur für einen logischen Knoten der Klasse MMXU

| Offset [Byte] | Länge [Byte] | Beschreibung | Beispiel (hexadezimal) | Wert (Beispiel) |
|--|-----------------|---|--|--|
| 0 | 4 | Logische-Knoten-Klasse (Enumeration, Int32) | 02 00 00 00 | 2 \triangleq MMXU |
| 4 | 64 | Logische-Knoten-Referenz (nullterminierte Zeichenkette) | 49 45 44 31 4C 44 65 76 69 63 65 31 2F 4D 4D 58 55 31 00 | IED1LDevice1/ MMXU1 |
| Spannung Phase 1 | | | | |
| 68 | 4 | Betrag (Float) | EB EB 6D 43 | 237,92 Volt |
| 72 | 4 | Winkel (Float) | 00 00 00 00 | 0° |
| 76 | 2 | Datenqualität (Quality Word) | 00 00 | Good |
| 78 | 4 | Zeitstempel in Sekunden (Unix Time) | 2B C0 29 5E | 1579794475 \triangleq 2020-01-23 15:47:55 |
| 82 | 4 | Bruchteil Time/Quality (derzeit nicht genutzt) | 00 00 00 00 | |
| 86 | 18 | Spannung Phase 2 (Aufbau wie Spannung Phase 1) | | |
| 104 | 18 | Spannung Phase 3 (Aufbau wie Spannung Phase 1) | | |
| Strom Phase 1 | | | | |
| 122 | 4 | Betrag (Float) | 0E 6F 1B 41 | 9,71 Ampere |
| 126 | 4 | Winkel (Float) | 00 00 00 00 | 0° |
| 130 | 2 | Datenqualität (Quality Word) | 00 00 | Good |
| 132 | 4 | Zeitstempel in Sekunden (Unix Time) | 2B C0 29 5E | 1579794475 \triangleq 2020-01-23 15:47:55 |
| 136 | 4 | Bruchteil Time/Quality (derzeit nicht genutzt) | 00 00 00 00 | |
| 140 | 18 | Strom Phase 2 (Aufbau wie Strom Phase 1) | | |
| 158 | 18 | Strom Phase 3 (Aufbau wie Strom Phase 1) | | |
| <i>Es folgen Wirk-, Blind-, Scheinleistung, Leistungsfaktor und Impedanz (jeweils dreiphasig, analog zu Spannungen und Strömen) sowie Frequenz (einzelner Messwert).</i> | | | | |

Im Rahmen von System-in-the-Loop-Prüfungen wurden im ie³-Labor bislang Messwerte von einem Echtzeitsimulator als analoge elektrische Signale ausgegeben, über Strom- und Spannungsverstärker verstärkt und dann von Geräten der Systemplattform über analoge Messkanäle eingelesen. Dies bedeutet einen erheblichen Verkabelungsaufwand, außerdem ist die Anzahl der übertragbaren Strom- und Spannungsmesswerte durch die Anzahl der vorhandenen Strom- und Spannungsverstärker begrenzt. An dieser Stelle erwies sich die implementierte Kommunikation per UDP zusätzlich als besonders vorteilhaft, da sich das Protokoll mit relativ geringem Aufwand auch auf dem Echtzeitsimulator umsetzen ließ. Dadurch ist es möglich, eine Netzwerkkommunikation zwischen Echtzeitsimulator und Geräten der Systemplattform aufzubauen und nahezu beliebig viele Messwerte auszutauschen.

4.4.2 IEC 61850-8-1 (MMS, GOOSE)

Mit Verfügbarkeit der in Kapitel 4.1 beschriebenen Systemplattform ist eine vollständige Nutzbarkeit des IEC-61850-Kommunikationsstacks gegeben. Da die bis zum Wechsel der Systemplattform implementierten Funktionen bereits das Online-Datenmodell (ACSI) zum Datenaustausch nutzten, waren an den Funktionen selbst keine Anpassungen notwendig. Lediglich die Programmkomponente *Autoconfig* musste so erweitert werden, dass sie das SCSM gemäß der im SCL-Datenmodell modellierten Kommunikationsbeziehungen konfiguriert.

Der IEC-61850-Kommunikationsstack besitzt sowohl die Fähigkeit zur Client-Server-Kommunikation per MMS als auch zur Publisher-Subscriber-Kommunikation per GOOSE. Die MMS-Kommunikation basiert auf dem zuverlässigen TCP (engl.: Transmission Control Protocol [125]), wodurch es sich auch bei MMS um ein zuverlässiges Kommunikationsprotokoll handelt. Dies ist insbesondere wichtig bei der Nutzung unzuverlässiger oder störanfälliger Kommunikationskanäle wie etwa der Weitbereichskommunikation per DSL oder Mobilfunk. Klassischerweise wird MMS zur Kommunikation zwischen Umspannanlage und Leitstelle eingesetzt. Die GOOSE-Kommunikation arbeitet auf OSI-Ebene 2 (Ethernet) und versendet Daten im Multicast oder Broadcast. Dadurch, dass GOOSE direkt auf OSI-Ebene 2 aufsetzt, ist eine schnelle Datenübertragung gegeben. Eine zeitlich gestaffelte Wiederholung von Paketen gleicht einen etwaigen Verlust einzelner Pakete aus. GOOSE wird klassischerweise zur Kommunikation innerhalb einer Umspannanlage eingesetzt, um Ereignisse wie z. B. Schutzauslösungen schnell zu kommunizieren. Durch die fehlende Routingfähigkeit ist GOOSE allerdings auf lokale Netzwerke begrenzt.

4.4.3 TCP

Im Rahmen des Forschungsprojekts *EOSG* [126] ergab sich die Notwendigkeit, eine Kommunikation zwischen einem Gerät der Systemplattform und einem externen Gerät zu realisieren. Die Kommunikation sollte einerseits zuverlässiger sein als UDP, andererseits war die Implementierung eines IEC-61850-Kommunikationsstacks auf dem externen Gerät nicht realisierbar. Als alternative Lösung wurde ein TCP-basiertes Kommunikationsprotokoll umgesetzt. Die auf der Systemplattform laufende Anwendung implementiert hierzu einen TCP-Server, der auf Port 44227 auf Verbindungsanfragen wartet und mit dem sich ein externer TCP-Client verbinden kann. Jede TCP-Nachricht besteht aus einem obligatorischen Header (siehe Tabelle B.1 in Anhang B) und optionalen Nutzdaten.

Das Protokoll arbeitet auf Basis von Anfragen, die der Client an den Server sendet und die Letzterer beantwortet. Hiermit kann der Client die Werte der Datenattribute, die zu einem logischen Knoten gehören, aus dem Online-Datenmodell des Servers auslesen bzw. dorthin schreiben. Dies ist vergleichbar mit den Read-/Write-Diensten von MMS. Zusätzlich kann der Client einzelne logische Knoten abonnieren: Sobald sich im serverseitigen Online-Datenmodell der Wert eines Datenattributs, das zu dem abonnierten logischen Knoten gehört, ändert, sendet der Server spontan die aktualisierten Daten an den Client. Dies ist vergleichbar mit dem Reporting-Dienst von MMS. Die Struktur der übertragenen Nutzdaten ist identisch zu der des UDP-basierten Kommunikationsprotokolls (siehe auch Beispiel in Tabelle 4.3). Die definierten Nachrichten-Typen können Tabelle B.2 entnommen werden.

Logische Knoten in einem Server-IED enthalten einerseits Datenobjekte und -attribute, die von einem Client nur gelesen werden sollen, zum Beispiel Messwerte oder Schalterstellungen. Datenobjekte und -attribute andererseits, die von einem Client in einen Server geschrieben werden sollen, sind beispielsweise Schaltbefehle oder Sollwertbefehle. Zur Unterscheidung, welche Datenobjekte und -attribute bei einem Lese- bzw. Schreibvorgang übertragen werden, wird jeder Nachricht der Logische-Knoten-Typ mitgegeben (siehe Tabelle B.3).

Möchte der TCP-Client zum Beispiel die Messwerte eines logischen Knotens der Klasse *MMXU* lesen, sendet er eine Nachricht mit *type=GET_LN_COMMAND*, *InType=STATUS*, *InClass=MMXU* und z. B. *reference=IED1LDevice1/MMXU1* an den TCP-Server. Der Server antwortet mit *type=GET_LN_RESPONSE_POSITIVE*, lässt die anderen Parameter unverändert und hängt die Werte des logischen Knotens als Nutzdaten an. Wenn die vom Client gesendete Referenz ungültig ist, weil etwa der angeforderte logische Knoten nicht existiert, antwortet der Server mit *type=GET_LN_RESPONSE_NEGATIVE*.

4.4.4 Modbus

Als weitere Kommunikationsverfahren sind Modbus RTU in der Master-Funktionalität zur Kommunikation über serielle Netzwerke (RS232/RS485) und Modbus TCP in der Client- und Server-Funktionalität zur Kommunikation über IP-Netzwerke implementiert. Der erste Anwendungsfall der Modbus-RTU-Kommunikation liegt hierbei in der Anbindung von NH-Sicherungslastschaltleisten vom Typ *PLPlano* der *Jean Müller GmbH* [127, 128] an die Systemplattform. Diese Sicherungslastschaltleisten verfügen über ein integriertes Messsystem und ermöglichen mit wenig Aufwand eine messtechnische Ertüchtigung von Niederspannungsabgängen in Ortsnetzstationen und Kabelverteilerschränken.

Zur Umsetzung der Modbus-Kommunikation wird die Open-Source-Bibliothek *libmodbus* [129] genutzt. Die Kommunikationspartner, die Daten bereitstellen, werden im Umfeld von Modbus als Slaves (seriell) bzw. Server (TCP) bezeichnet, diejenigen, die Daten abfragen, als Master (seriell) bzw. Client (TCP). Zur Vereinfachung wird im Folgenden ausschließlich von Server und Client gesprochen. Die Konfiguration der Modbus-Kommunikation (IP-, Slave-Adressen, Registeradressen usw.) erfolgt ausschließlich über das SCL-Datenmodell und ist in Kapitel 4.5.1 beschrieben.

Die Modbus-Lese- und Schreibbefehle werden automatisch mit Datenattributen innerhalb des Online-Datenmodells verknüpft. Daten, die von einem Modbus-Server gelesen werden sollen, werden periodisch abgefragt (*gepollt*) und der gelesene Wert wird in das entsprechende Datenattribut geschrieben. Daten, die in einen Modbus-Server geschrieben werden sollen, werden aus dem entsprechenden Datenattribut gelesen und nur bei Wertänderung in den Modbus-Server geschrieben.

Grundsätzlich sind Datenattribute innerhalb eines IEC-61850-Online-Datenmodells entweder zum Lesen durch die Applikation (Status- und Messwerte) oder zum Schreiben (Befehle und Sollwerte) vorgesehen. Diese Eigenschaft eines Datenattributs wird durch den sogenannten *Functional Constraint* (FC, funktionale Einschränkung) beschrieben. Für Status- und Messwerte ist der FC *ST* bzw. *MX*, für Befehle und Sollwerte *CO*. Abhängig vom FC wird bestimmt, ob der Wert eines Datenattributs von einem Modbus-Server gelesen oder in diesen geschrieben werden soll.

- Das Datenattribut *MMXU1/PhV/phsA/cVal/mag/f* repräsentiert einen Messwert (FC MX). Das Datenattribut kann von der Applikationen nur gelesen werden und wird dementsprechend vom Modbus-Server gelesen.
- Das Datenattribut *FSPT1/SptVal/Oper/ctlVal/f* repräsentiert einen Sollwert (FC CO). Dieses Datenattribut hingegen kann von der Applikation nur geschrieben werden und wird dementsprechend in den Modbus-Server geschrieben.

Das Listing 4.8 auf Seite 100 zeigt Beispiele für beides. Außerdem ist für jedes Datenattribut der Datentyp definiert. Hierüber wird bestimmt, welchen Wert das Datenattribut aufnehmen kann, beispielsweise einen ganzzahligen numerischen Wert (*Integer*), eine Gleitkommazahl (*Float*) oder einen logischen Zustand (*Bool*).

Der Zugriff auf die Werte eines Modbus-Servers erfolgt über adressierbare Speicherbereiche. Modbus definiert mehrere Datentypen mit unterschiedlicher Datenbreite und unterschiedlichen Zugriffsrechten gemäß Tabelle 4.4:

Tabelle 4.4: Modbus-Zugriffsverfahren

| Speicherblock | Datentyp | Zugriffsrechte |
|------------------|------------------------|-----------------|
| Coil | Boolean (1 Bit) | Lesen/Schreiben |
| Discrete Input | Boolean (1 Bit) | Lesen |
| Holding Register | Unsigned Word (16 Bit) | Lesen/Schreiben |
| Input Register | Unsigned Word (16 Bit) | Lesen |

Die Konfiguration, wie zwischen den Werten eines IEC-61850-Datenattributs und denen eines Modbus-Speicherblocks konvertiert ist, erfolgt über Parameter im SCL-Datenmodell. Beispielsweise kann der aus einem Modbus-Register (16 Bit) gelesene Wert direkt als Ganzzahl interpretiert und als Float-Wert in einem Datenattribut gespeichert werden. Alternativ können aber auch zwei benachbarte Modbus-Register (32 Bit) auf Bit-Ebene (d. h. ohne numerische Interpretation) direkt in einen Float-Wert (ebenfalls 32 Bit) überführt werden.

4.4.5 SQLite

Neben der reinen Kommunikation von Daten wie etwa Messwerten zwischen verteilten Systemen ergibt sich häufig der Bedarf, diese Daten auch zu archivieren, um so beispielsweise eine Analyse im Nachhinein durchführen zu können. Dies macht die Umsetzung einer geeigneten Datenhaltung erforderlich.

Als Datenbanksystem wird SQLite [130] implementiert. Bei SQLite handelt es sich um ein relationales SQL-Datenbanksystem, das einen Großteil der SQL-Sprachbefehle unterstützt und als gemeinfreie Programmbibliothek im C-Quellcode zur Verfügung steht. Im Gegensatz zu anderen Datenbanksystemen wird SQLite direkt in die Anwendung integriert und benötigt keinen separaten Serverprozess, sondern arbeitet dateibasiert. SQLite ist als kompakte und schnelle Datenbanklösung insbesondere für die Nutzung in eingebetteten Systemen ausgelegt und findet Verwendung in zahlreichen Anwendungsprogrammen als auch beispielsweise auf Mobiltelefonen.

Die SQLite-Bibliothek ist derart in das Softwareframework integriert, dass eine direkte Nutzung durch Smart-Grid-Funktionen möglich ist. So kann beispielsweise die Funktion der Netzzustandsschätzung die als Eingangsdaten benötigten Messwerte alternativ zum Online-Datenmodell auch aus einer SQLite-Datenbank einlesen und umgekehrt ihre Berechnungsergebnisse in eine SQLite-Datenbank schreiben.

4.5 Systemkonfiguration

Ein besonderes Merkmal des realisierten Systems besteht in seiner Konfigurierbarkeit über SCL-Datenmodelle. Ohne Anpassungen am Quellcode oder Neuerstellung der Anwendung kann allein durch Anpassung und Austausch des SCL-Datenmodells eine andere Funktionalität des Systems erreicht werden. Ermöglicht wird dies durch die Programmkomponente *Autoconfig*. Diese verarbeitet bei Programmstart eine gegebene Konfigurationsdatei mit einem SCL-Datenmodell und erzeugt Instanzen für alle konfigurierten Smart-Grid-Funktionen. Anschließend werden die Funktionen zyklisch auf dem System ausgeführt. Innerhalb eines SCL-Datenmodells können Datenmodelle für beliebig viele Systeminstanzen und darüber hinaus auch die Kommunikationsbeziehungen zwischen den Systeminstanzen modelliert werden. Durch die Bündelung in einem einzigen SCL-Datenmodell wird insbesondere die Konsistenz der Engineeringdaten sichergestellt.

Die grundlegenden Engineering- und Konfigurationsschritte, -rollen und -werkzeuge gemäß IEC 61850-6 wurden bereits in Kapitel 3.4 allgemein eingeführt. Im nachfolgenden Unterkapitel 4.5.1 werden zunächst die zur Konfiguration der Systeme genutzten SCL-Sprachelemente dargestellt. Anschließend wird in Unterkapitel 4.5.2 im Detail erläutert, wie die Programmkomponente *Autoconfig* auf Basis eines SCL-Datenmodells die Systemkonfiguration durchführt. Unterkapitel 4.5.3 beschreibt schließlich, wie die Programmablaufsteuerung die Ausführung der Smart-Grid-Funktionen koordiniert. Der Engineeringprozess zur Erstellung von SCL-Datenmodellen wird in Kapitel 5.2 exemplarisch für das Anwendungsszenarium der Netzzustandsschätzung ausgeführt.

4.5.1 Genutzte SCL-Sprachelemente

Der Standard IEC 61850 definiert in seinem Teil 6 umfangreiche Sprachelemente, die es ermöglichen, die Konfiguration eines Stationsautomatisierungssystems inklusive der Zuordnung zur Schaltanlagenprimärtechnik zu beschreiben. Beim Design eines Engineering- und Konfigurationsprozesses für das hier realisierte System ist es daher erforderlich, im ersten Schritt die zu modellierenden Elemente und Beziehungen zu definieren. Im zweiten Schritt werden Sprachelemente aus der IEC 61850-6 identifiziert, mit denen eine solche Modellierung durchgeführt werden kann.

Zu modellierende Elemente und Beziehungen sind unter anderem:

- Logische Knoten, die als Datencontainer dienen
- Smart-Grid-Funktionen, ggf. mit zugehörigen Daten und Parametern
- Kommunikationsbeziehungen und Kommunikationsverfahren
- Hardwarespezifische Konfigurationen wie beispielsweise die Definition von analogen und binären Ein- und Ausgängen

Logische Knoten

Jede Systeminstanz stellt ein Intelligent Electronic Device (IED) dar und wird durch ein eigenes IED-Datenmodell innerhalb des Gesamt-SCL-Datenmodells repräsentiert. Das IED-Datenmodell enthält insbesondere die logischen Knoten, die innerhalb einer laufenden Applikation einen Datenspeicher darstellen und dazu dienen, zwischen einzelnen Funktionen Informationen wie Messwerte, Statuswerte oder Parameter auszutauschen. Weiterhin ist es möglich, über einen Kommunikationskanal auf die logischen Knoten zuzugreifen, um deren Daten zu anderen Systeminstanzen (horizontal) oder zu überlagerten Systemen wie einer Netzleitstelle (vertikal) zu kommunizieren. Es müssen logische Knoten in ausreichender Anzahl und von entsprechenden Klassen vorhanden sein, so dass ein Datenaustausch zwischen den Funktionen möglich ist und etwaige weitere Daten gespeichert werden können.

Im SCL-Datenmodell werden innerhalb eines *IED*-Elements als physischem Gerät hierarchisch *AccessPoint*-Elemente, ein *Server*-Element und zur weiteren Strukturierung *LDevice*-Elemente als logische Geräte angelegt. Auf die Elemente *AccessPoint* und *Server* soll an dieser Stelle nicht weiter eingegangen werden. Im einfachsten Fall ist genau ein *LDevice* vorhanden, das schließlich die logischen Knoten als *LN*-Elemente enthält. Den logischen Knoten wird eine Klasse gemäß IEC 61850-7-4 zugewiesen (*InClass*). Außerdem werden sie durch Auswahl eines zuvor definierten Logischen-Knoten-Typs (*LNNodeType*) aus den *DataTemplates* typisiert (*InType*) und damit in ihrer exakten Ausprägung, d. h. über welche Datenobjekte und Datenattribute sie verfügen, spezifiziert. Während einige Datenobjekte verpflichtend sind (*mandatory*), sind einige optional (*optional*) oder an Bedingungen geknüpft (*conditional*). Daher ist darauf zu achten, dass der gewählte Logischen-Knoten-Typ alle für eine Funktion erforderlichen Datenobjekte enthält. Für das Szenario der Spannungsregelung zeigt das Listing 4.1 beispielhaft die Modellierung im IED-Teil der SCL.

Listing 4.1: Beispielhafte Modellierung im IED-Teil der SCL

```

<IED desc="Central voltage controller" name="Master">
  <AccessPoint name="AccessPoint1">
    <Server desc="Server">
      <LDevice inst="LDevice1">
        <LN0 InType="iA_LN0" InClass="LLN0" inst="" />
        <LN InType="iA_LPHD" InClass="LPHD" inst="1" />
        <LN desc="Automatic Tap Changer Controller" InType="iA_ATCC" InClass="ATCC"
          inst="1">
          <DOI name="HiTapPos">
            <DAI name="stVal">
              <Val>13</Val>
            </DAI>
          </DOI>
          <DOI name="LoTapPos">
            <DAI name="stVal">
              <Val>-13</Val>
            </DAI>
          </DOI>
        </LN>
        <LN desc="Switch Controller" InType="iA_CSWI" InClass="CSWI" inst="1" />
      </LDevice>
    </Server>
  </AccessPoint>
</IED>

```

Neben der Art und Anzahl der Datenobjekte und Datenattribute legt ein Logischer-Knoten-Typ auch Standardwerte für die Werte von Datenattributen fest. Bei Datenattributen mit numerischen Werten ist dies in der Regel Null. Sollen einzelne Logische-Knoten-Instanzen jedoch mit abweichenden Werten initialisiert werden, so kann dies direkt im IED-Datenmodell beschrieben werden. Für den logischen Knoten *ATCC* in Listing 4.1 wird beispielsweise der Wert des Datenattributs *HiTapPos/stVal* abweichend mit *13* initialisiert, der des Datenattributs *LoTapPos/stVal* mit *-13*. Alle übrigen Objekte werden gemäß den Vorgaben des Logischen-Knoten-Typs instanziiert und brauchen daher an dieser Stelle nicht explizit beschrieben werden.

Funktionen

Die im Teil 4 des Standards IEC 61850 definierten Logische-Knoten-Klassen sind auf konkrete Funktionalitäten bezogen. Beispielsweise kann der Funktionalität des Überstromzeitschutzes direkt die Logische-Knoten-Klasse *PTOC* zugeordnet werden, der Funktionalität der dreiphasigen Messung die Logische-Knoten-Klasse *MMXU*. Eine direkte Zuordnung zwischen exakt einer Funktion und exakt einer Logische-Knoten-Klasse liegt daher zwar nahe, ist dennoch aus folgenden Gründen nicht allgemeingültig möglich:

1. Einerseits kann eine Funktion Zugriff auf mehr als einen logischen Knoten benötigen. Die implementierte Funktion des Überstromzeitschutzes (*PTOC_Function*) liest zwar ihre Schutzparameter aus einem logischen Knoten der Klasse *PTOC* und schreibt ebenso ihre Anrege- und Auslöseinformation dorthin zurück. Dennoch benötigt die Funktion Zugriff auf einen logischen Knoten der Klasse *MMXU*, um die erforderlichen Strom- und Spannungsmesswerte zu erhalten.
2. Zu manchen, insbesondere komplexen Funktionen existiert keine direkte Entsprechung als Logische-Knoten-Klasse. Das trifft beispielsweise auf die Spannungsregelung und die Netzzustandsschätzung zu. Die Spannungsregelung mit der Funktion *VoltageControl_Master* greift in der aktuellen Implementierung auf insgesamt sechs logische Knoten zu: Aus zwei logischen Knoten der Klasse *MMXU* werden Messwerte eingelesen, über zwei logische Knoten der Klasse *FSPT* werden Sollwerte für die Blindleistungsregelung dezentraler Anlagen ausgegeben, über einen logischen Knoten der Klasse *ATCC* erfolgt eine Transformatorstufung und über einen weiteren logischen Knoten der Klasse *CSWI* die Steuerung eines Schaltgeräts. Die Netzzustandsschätzung benötigt Knoten- und Zweigmesswerte. Diese liest sie, für jeden Knoten und jeden Zweig separat, über logische Knoten der Klasse *MMXU* ein. Ebenso werden die estimierten komplexen Spannungen in der aktuellen Implementierung in logische Knoten der Klasse *MMXU* gespeichert.
3. Weiterhin können, je nach Anwendungsfall, unterschiedliche Varianten einer Funktion benötigt werden. Die Messwernerfassung ist sowohl ein- als auch dreiphasig implementiert, mit den Funktionen *MeasurementSinglePhase* und *MeasurementThreePhase*. Obwohl beide Funktionen auf eine gleiche Anzahl logischer Knoten zugreifen, unterscheiden sie sich in ihren Algorithmen.

Die Beispiele zeigen die Notwendigkeit, Funktionen unabhängig von logischen Knoten zu modellieren. Eine Lösung hierzu findet sich im Substation-Teil eines SCL-Datenmodells. Der Standard IEC 61850-6 sieht unterhalb der Elemente *Substation*, *VoltageLevel* und *Bay* sogenannte *Function*-Elemente vor, um Funktionen abstrakt und unabhängig von primärtechnischen Geräten zu modellieren. Ein *Function*-Element selbst stellt einen Container für abstrakte logische Knoten (*LNnodes*) dar. Mit einem *Function*-Element ist es somit möglich, logische Knoten funktional zu gruppieren. Damit erweist sich das *Function*-Element als geeignet zur Modellierung und Referenzierung der implementierten Smart-Grid-Funktionen.

Für jede gewünschte Funktion wird unterhalb eines *Bays* eine *Function* erstellt. Eine *Function* besitzt unter anderem die Attribute *type* und *name* (Datentyp *String*). Das *type*-Attribut wird auf das Funktionschlüsselwort gemäß Tabelle 4.1 (Spalte 1) gesetzt und identifiziert damit die jeweilige Smart-Grid-Funktion. Das *name*-Attribut kann frei belegt werden, um beispielsweise mehrere Instanzen eines Funktionstyps zu

benennen. Unterhalb der *Function* werden die von der Funktion benötigten logischen Knoten aus dem IED-Datenmodell referenziert. Da Anzahl und Reihenfolge der logischen Knoten durch den Algorithmus der jeweiligen Funktion festgelegt sind, muss die in Tabelle 4.1 (Spalte 3) dargestellte, vorgegebene Reihenfolge eingehalten werden. Für das Szenario der Spannungsregelung zeigt Listing 4.2 beispielhaft die Modellierung von Funktionen in einem *Bay* im Substation-Teil der SCL.

Listing 4.2: Beispielhafte Modellierung von Funktionen in einem Bay

```
<Bay name="Master">
  <Function name="VoltageControl_Master1" type="VoltageControl_Master">
    <LNode iedName="Slave1" IdInst="LDevice1" InClass="MMXU" InInst="1"
      InType="iA_MMXU" />
    <LNode iedName="Slave1" IdInst="LDevice1" InClass="FSPT" InInst="1"
      InType="iA_FSPT" />
    <LNode iedName="Slave2" IdInst="LDevice1" InClass="MMXU" InInst="1"
      InType="iA_MMXU" />
    <LNode iedName="Slave2" IdInst="LDevice1" InClass="FSPT" InInst="1"
      InType="iA_FSPT" />
    <LNode iedName="Master" IdInst="LDevice1" InClass="ATCC" InInst="1"
      InType="iA_ATCC" />
    <LNode iedName="Master" IdInst="LDevice1" InClass="CSWI" InInst="1"
      InType="iA_CSWI" />
  </Function>
  <Function name="ATCC_Function1" type="ATCC_Function">
    <LNode iedName="Master" IdInst="LDevice1" InClass="ATCC" InInst="1"
      InType="iA_ATCC" />
  </Function>
  <Function name="CSWI_Function1" type="CSWI_Function">
    <LNode iedName="Master" IdInst="LDevice1" InClass="CSWI" InInst="1"
      InType="iA_CSWI" />
  </Function>
</Bay>
```

Einige Smart-Grid-Funktionen benötigen Informationen, die sich mit den normativen Elementen der IEC 61850-6 nicht direkt abbilden lassen. So benötigt die Netzzustandsschätzung etwa Topologieinformationen des zugrunde liegenden Netzes sowie weitere Parameter wie die maximale Anzahl an Iterationen, bis zu der eine Konvergenz der Berechnung eintreten muss. Diese Informationen werden in der aktuellen Implementierung in einem Satz weiterer Konfigurationsdateien vorgehalten. Der Smart-Grid-Funktion *StateEstimation* wird über das SCL-Datenmodell dabei lediglich der relative Pfad zu diesen Dateien auf der Systeminstanz mitgegeben. Hierzu wird ein *private Element* wie in Listing 4.3 genutzt.

Listing 4.3: Nutzung eines privaten Elements

```

<Function name="StateEstimation1" type="StateEstimation">
  <Private type="ie3.SmartGridTechnologies.SCL.PrivateContents.
    tStateEstimationConfiguration">
    <StateEstimationConfiguration path="StateEstimation/128-System" xmlns=
      "http://www.ie3.e-technik.tu-dortmund.de/ie3.SmartGridTechnologies.SCL.
        PrivateContents" />
  </Private>
</Function>

```

Die IEC 61850 definiert keinen Bezug zwischen einem Bay und einem IED. Ein Bay stellt ein primärtechnisches Schaltfeld dar, ein IED ein sekundärtechnisches Gerät. Prinzipiell können einem Bay mehrere IEDs zugeordnet werden, beispielsweise ein Leitgerät und mehrere Schutzgeräte. Ebenso kann ein IED nicht nur feldbezogene, sondern auch übergeordnete oder feldübergreifende Aufgaben erfüllen.

Im Rahmen der hier vorgestellten Implementierung werden jedoch Funktionen, die zu einer Systeminstanz und damit zu einem IED gehören, innerhalb eines Bays modelliert. Um vom IED auf die zugehörigen Funktionen schließen zu können, wird daher folgende Konvention vereinbart:

1. Jedem IED wird genau ein Bay zugeordnet.
2. IED und Bay tragen denselben Namen (Attribut *name*).

Kommunikationsbeziehungen und Kommunikationsverfahren

Der Standard IEC 61870 beschreibt in seinem Teil 7-1 Verfahren der Client-Server- und Publisher-Subscriber-Kommunikation und spezifiziert diese Verfahren im Teil 8-1 in Form von MMS und GOOSE. Der Teil 6 ermöglicht entsprechend eine Modellierung dieser Kommunikationsbeziehungen und Kommunikationsverfahren.

Im Rahmen der hier umgesetzten Implementierung wird als Kommunikationsverfahren nach IEC 61850 die Client-Server-Kommunikation MMS eingesetzt. Daher wird nachfolgend zunächst die hierzu erforderliche Modellierung im SCL-Datenmodell beschrieben. Anschließend wird auf die Modellierung weiterer implementierter Kommunikationsverfahren eingegangen.

IEC 61850-8-1 (MMS)

Soll ein IEC-61850-Client Daten von einem IEC-61850-Server abrufen, so ist dies auf zwei Varianten möglich: Der Client kann den Server zyklisch pollen, also in regelmäßigen Abständen die Daten vom Server lesen. Bei dieser Variante überwiegen aber die Nachteile. Einerseits sind Client und Server durch kontinuierlichen Rechenaufwand belastet. Weiterhin wird ein kontinuierlicher Datenverkehr erzeugt.

Beides geschieht unabhängig davon, ob sich die angefragten Daten überhaupt zwischenzeitlich geändert haben. Schließlich erscheinen Aktualisierungen der Daten nur in fest definierten Abfrageintervallen, so dass zwischenzeitliche Änderungen an den Daten unter Umständen unnötig verspätet eintreffen. Die bessere Variante besteht darin, dass der Client sich auf bestimmte Daten am Server registriert und letzterer diese bei Änderungen automatisch an den Client überträgt. Im Rahmen von IEC 61850 wird dieses Verfahren als *Reporting* bezeichnet.

Zur IEC-61850-konformen Kommunikation zwischen Systeminstanzen wird im Rahmen der Implementierung daher Reporting genutzt. Reporting kann innerhalb der IED-Datenmodelle konfiguriert werden. Im ersten Schritt sind die Daten zu definieren, die kommuniziert werden sollen. Dies geschieht in Form von *DataSets*-Elementen. Ein DataSet stellt eine Liste von Referenzen auf Datenobjekte und/oder Datenattribute dar, deren Werte kommuniziert werden sollen. Durch ein *ReportControl*-Element schließlich wird eine Kommunikationsbeziehung per Reporting modelliert. Ein ReportControl seinerseits referenziert ein DataSet. Weiterhin kann im ReportControl das Ziel der Kommunikation festgelegt werden. Listing 4.4 zeigt beispielhaft die Modellierung eines DataSets und eines ReportControls.

Listing 4.4: Modellierung eines DataSets und ReportControls

```
<IED desc="Decentralised measurement" name="Slave1">
  <AccessPoint name="AccessPoint1">
    <Server desc="Server">
      <LDevice inst="LDevice1">
        <LNO InType="iA_LNO" InClass="LLNO" inst="">
          <DataSet name="DataSet1">
            <FCDA IdInst="LDevice1" InClass="MMXU" InInst="1" doName="PhV" fc="MX" />
          </DataSet>
          <ReportControl name="ReportControl1" datSet="DataSet1" rptID="Slave1Report1"
            confRev="1">
            <TrgOps dchg="true" />
            <OptFields />
            <RptEnabled>
              <ClientLN iedName="Master" IdInst="None" InClass="ITCI" InInst="1" />
            </RptEnabled>
          </ReportControl>
        </LNO>
        <LN InType="iA_LPHD" InClass="LPHD" inst="1" />
        <LN desc="Measurement" InType="iA_MMXU" InClass="MMXU" inst="1" />
        [...]
      </LDevice>
    </Server>
  </AccessPoint>
</IED>
```

Die Modellierung von ReportControl und DataSets erfolgt i. d. R. im logischen Knoten *LN0*, der standardmäßig in einem *LDevice* vorhanden sein muss. Das DataSet enthält eine Liste von *FCDA*-Elementen, die auf Datenobjekte oder Datenattribute verweisen, im dargestellten Beispiel auf das Datenobjekt *LDevice1/MMXU1/PhV*. Der ReportControl seinerseits verweist auf das DataSet. Über sogenannte Trigger-Options *TrgOps* kann konfiguriert werden, wann ein Report versandt werden soll. Im gezeigten Beispiel geschieht dies bei Änderungen an den Daten (*dchg*: data change). Alternativ wäre auch eine zyklische Übertragung konfigurierbar. Innerhalb des Elements *RptEnabled* können Clients definiert werden, an die der Report versandt werden soll. Es wird jedoch nicht direkt das Client-IED referenziert, sondern ein spezieller logischer Knoten, der als Kommunikationsinterface dient (*ITCI*: Telecontrol Interface). Dieser logische Knoten muss sich direkt in einem separaten AccessPoint (*ClientAP*) innerhalb des Client-IEDs befinden. Listing 4.5 zeigt dies.

Listing 4.5: Modellierung des Clients mit einem weiteren AccessPoint *ClientAP*

```
<IED desc="Central voltage controller" name="Master">
  <AccessPoint name="AccessPoint1">
    [...]
  </AccessPoint>
  <AccessPoint name="ClientAP">
    <LN desc="Telecontrol Interface" lnType="IA_ITCI" lnClass="ITCI" inst="1" />
  </AccessPoint>
</IED>
```

Neben der Modellierung der zu kommunizierenden Daten und der Kommunikationsverfahren sind die Kommunikationspartner in Form der IEDs und die Kommunikationsnetzwerke zu modellieren. Dies geschieht im *Communication*-Teil des SCL-Datenmodells. Über *SubNetwork*-Elemente können in sich geschlossene Netzwerksegmente definiert werden. IEDs, die zu einem SubNetwork gehören, können untereinander kommunizieren, nicht jedoch mit IEDs in anderen SubNetworks. Die Referenzierung von IEDs in einem SubNetwork erfolgt über *ConnectedAP*-Elemente. Es wird nicht das IED an sich in ein SubNetwork eingebunden, sondern vielmehr ein AccessPoint innerhalb eines IEDs. Durch Modellierung von mehr als einem AccessPoint in einem IED ist es dabei auch möglich, ein IED mehreren SubNetworks zuzuordnen. Innerhalb eines ConnectedAPs werden darüber hinaus die Kommunikationsadressen und -parameter als *P*-Elemente beschrieben. Im Fall von MMS sind dies IP-Adressinformationen (*IP*, *Subnetz* und *Gateway*) und Parameter für die OSI-Schichten *Presentation*, *Session* und *Transport*. Listing 4.6 zeigt beispielhaft eine Modellierung.

Listing 4.6: Modellierung der Kommunikationspartner und Kommunikationsnetzwerke

```

<Communication>
  <SubNetwork name="SubNetwork1">
    <ConnectedAP iedName="Master" apName="AccessPoint1">
      <Address>
        <P xsi:type="tP_IP" type="IP">129.217.202.47</P>
        <P xsi:type="tP_IP-SUBNET" type="IP-SUBNET">255.255.255.192</P>
        <P xsi:type="tP_IP-GATEWAY" type="IP-GATEWAY">129.217.202.62</P>
        <P xsi:type="tP_OSI-PSEL" type="OSI-PSEL">00000001</P>
        <P xsi:type="tP_OSI-SSEL" type="OSI-SSEL">0001</P>
        <P xsi:type="tP_OSI-TSEL" type="OSI-TSEL">0001</P>
      </Address>
    </ConnectedAP>
    <ConnectedAP iedName="Slave1" apName="AccessPoint1">
      [...]
    </ConnectedAP>
    <ConnectedAP iedName="Slave2" apName="AccessPoint1">
      [...]
    </ConnectedAP>
  </SubNetwork>
</Communication>

```

UDP

Zur Konfiguration der UDP-Kommunikation werden direkt die als Reporting gemäß IEC 61850-6 modellierten Kommunikationsbeziehungen in den IED-Datenmodellen genutzt. Sind in dem IED-Datenmodell, das die lokale Systeminstanz beschreibt, Reports und DataSets definiert, so wird der UDP-Publisher dahingehend konfiguriert, dass er die Werte derjenigen logischen Knoten versendet, deren Datenattribute und Datenobjekte in den DataSets referenziert werden. Enthält das IED-Datenmodell, das die lokale Systeminstanz beschreibt, einen AccessPoint *ClientAP* mit einem logischen Knoten *ITCI* und referenzieren andere IED-Datenmodelle diesen logischen Knoten über einen ihrer ReportControl-Elemente, so wird der UDP-Subscriber entsprechend für den Empfang dieser Reports konfiguriert.

TCP

Zur Nutzung des eigens implementierten TCP-Kommunikationsprotokolls sind keine Anpassungen am SCL-Datenmodell erforderlich. Die Applikation der TCP-Kommunikation greift direkt auf das Online-Datenmodell zu.

Modbus

Die lokale Systeminstanz, die als Modbus-Client Daten von einem Modbus-Server lesen bzw. dorthin schreiben soll, wird unverändert als IED modelliert und kann nach Bedarf beliebige logische Knoten enthalten, die z. B. von Funktionen benötigt werden. Wie bei Reporting, muss auch dieses IED-Datenmodell einen zusätzlichen AccessPoint mit einem logischen Knoten *ITCI* als Kommunikationsinterface enthalten, siehe Listing 4.7. Das Server-IED-Datenmodell wird diesen logischen Knoten referenzieren, um die Kommunikationsbeziehung zwischen beiden Geräten zu beschreiben.

Listing 4.7: Client-IED-Datenmodell

```
<IED name="Client">
  <AccessPoint name="AccessPoint1">
    <Server desc="Server">
      <LDevice inst="LDevice1">
        <LN0 InType="iA_LN0" InClass="LLN0" inst="" />
        <LN InType="iA_LPHD" InClass="LPHD" inst="1" />
      </LDevice>
    </Server>
  </AccessPoint>
  <AccessPoint name="ClientAP">
    <LN desc="Telecontrol Interface" InType="iA_ITCI" InClass="ITCI" inst="1" />
  </AccessPoint>
</IED>
```

Auch wenn es sich bei einem externen Modbus-Server, mit dem kommuniziert werden soll, nicht um ein System nach IEC 61850 handelt, so erfolgt an dieser Stelle die Modellierung dennoch konsequent als IED gemäß IEC 61850. Für jeden Modbus-Server, der über eine separate IP-Adresse (Modbus TCP) bzw. einen eigenen seriellen Port (Modbus RTU) ansprechbar ist, wird entsprechend ein IED-Datenmodell angelegt.

Das Server-IED-Datenmodell muss logische Knoten enthalten, die geeignet sind, alle vom Modbus-Server gelesenen bzw. in ihn geschriebenen Daten zu speichern. Für jedes zu lesende bzw. zu schreibende Modbus-Register muss innerhalb des IED-Datenmodells ein geeignetes Datenattribut instanziiert werden, wie dies in den umrahmten Elementen in Listing 4.8 gezeigt ist.

Listing 4.8: Server-IED-Datenmodell

```

<IED name="Server">
  <AccessPoint name="AccessPoint1">
    <Server desc="Server">
      <LDevice inst="LDevice1">
        <LNO InType="iA_LNO" InClass="LLNO" inst="">
          <DataSet name="DataSet1">
            <FCDA IdInst="LDevice1" InClass="MMXU" InInst="1" doName="PhV.phsA" fc="MX"
              />
          </DataSet>
          <ReportControl name="ReportControl1" datSet="DataSet1" intgPd="1000"
            rptID="rptServer" confRev="1">
            <TrgOps period="true" />
            <RptEnabled>
              <ClientLN iedName="Client" IdInst="None" InClass="ITCI" InInst="1" />
            </RptEnabled>
          </ReportControl>
        </LNO>
        <LN InType="iA_LPHD" InClass="LPHD" inst="1" />
        <LN desc="Measurement" InType="iA_MMXU" InClass="MMXU" inst="1">
          <DOI name="PhV">
            <SDI name="phsA">
              <SDI name="cVal">
                <SDI name="mag">
                  <DAI name="f" sAddr="Modbus(InputRegister=5;DataType=float)" />
                </SDI>
              </SDI>
            </SDI>
          </DOI>
        </LN>
        <LN desc="Set-point control function" InType="iA_FSPT" InClass="FSPT" inst="1">
          <DOI name="SptVal">
            <SDI name="Oper">
              <SDI name="ctlVal">
                <DAI name="f" sAddr="Modbus(HoldingRegister=7;DataType=float)">
                  <Val>123</Val>
                </DAI>
              </SDI>
            </SDI>
          </DOI>
        </LN>
        <LN desc="Measurement" InType="iA_MMXU" InClass="MMXU" inst="1">
          <DOI name="PhV">
            <SDI name="phsA">
              <SDI name="cVal">
                <SDI name="mag">
                  <DAI name="f" sAddr="Modbus(HoldingRegister=7;DataType=float)" />
                </SDI>
              </SDI>
            </SDI>
          </DOI>
        </LN>
      </LDevice>
    </Server>
  </AccessPoint>
</IED>

```

Die Modellierung der Kommunikationsbeziehungen erfolgt analog zur MMS-Kommunikation über ein ReportControl-Element, der das Kommunikationsinterface *ITCI* des Clients referenziert.

Zur Beschreibung der Datenattribut-bezogenen Modbus-Konfiguration wird, wie bereits in Listing 4.8 zu sehen, das normative Attribut *sAddr* genutzt, über das jedes Datenattribut verfügt und das zur Ablage herstellerspezifischer Informationen genutzt werden kann. Bei der Datenattribut-bezogenen Modbus-Konfiguration handelt es sich im einfachsten Fall um die Adresse eines Modbus-Registers, dessen Wert dem Datenattribut zugeordnet werden soll.

Grundsätzlich kann das *sAddr*-Attribut eine beliebige Zeichenkette aufnehmen¹⁰. Zur Modbus-Konfiguration wird die Konvention eingeführt, dass der Inhalt dem Format *Modbus(<Parameterliste>)* entsprechen muss, wobei *<Parameterliste>* eine durch Semikola getrennte Liste von Schlüsseln bzw. durch Gleichheitszeichen getrennte Schlüssel-Wert-Paare ist. Gültige Schlüssel sind in Tabelle C.1 (Anhang C) aufgeführt.

Das native Datenformat der Input- und Holding-Register ist eine vorzeichenlose 16-Bit-Ganzzahl (unsigned integer 16 bit). Mit Hilfe des Parameters *DataType* kann eine Konvertierung und Neuinterpretation in andere Datentypen modelliert werden. Für Datentypen, die breiter als 16 Bit sind, werden benachbarte Modbus-Register kombiniert, wobei die angegebene Adresse die des ersten Modbus-Registers ist. Tabelle C.2 zeigt gültige Werte für den Parameter *DataType*.

Die Reihenfolge der Bytes und Worte (engl.: Endianness) in den Modbus-Daten kann von Anwendung zu Anwendung und System zu System variieren. Während die Endianness von Modbus standardmäßig Big Endian ist, kann mit den Parametern *SwapByte* und *SwapWord* eine andere Endianness konfiguriert werden, wie in Tabelle C.3 gezeigt.

Eine gültige Modbus-Konfiguration könnte wie in Listing 4.9 dargestellt aussehen.

Listing 4.9: Beispiel für eine gültige Modbus-Konfiguration (Wert eines *sAddr*-Attributs)

Modbus(Address=1;HoldingRegister=7;DataType=float;SwapByte;SwapWord;Scaling=100)

Die Modellierung der Kommunikationspartner und Kommunikationsnetzwerke erfolgt bei Modbus TCP analog zu anderen Kommunikationsverfahren wie MMS oder UDP im Communication-Teil des SCL-Datenmodells. Listing 4.10 zeigt dies beispielhaft.

¹⁰ Das SCL-Schema definiert keine Höchstlänge der Zeichenkette, jedoch kann implementierungsabhängig eine Längenbegrenzung erfolgen. So erlaubt der hier genutzte IEC-61850-Kommunikationsstack für ein *sAddr*-Attribut maximal 64 Zeichen.

Listing 4.10: Modellierung der Kommunikationsbeziehungen für Modbus TCP

```

<Communication>
  <SubNetwork name="SubNetwork1">
    <ConnectedAP iedName="Client" apName="AccessPoint1">
      <Address>
        <P xsi:type="tP_IP" type="IP">129.217.202.47</P>
        <P xsi:type="tP_IP-SUBNET" type="IP-SUBNET">255.255.255.192</P>
        <P xsi:type="tP_IP-GATEWAY" type="IP-GATEWAY">129.217.202.62</P>
        <P xsi:type="tP_OSI-PSEL" type="OSI-PSEL">00000001</P>
        <P xsi:type="tP_OSI-SSEL" type="OSI-SSEL">0001</P>
        <P xsi:type="tP_OSI-TSEL" type="OSI-TSEL">0001</P>
      </Address>
    </ConnectedAP>
    <ConnectedAP iedName="Server" apName="AccessPoint1">
      <Address>
        <P xsi:type="tP_IP" type="IP">129.217.202.46</P>
        <P xsi:type="tP_IP-SUBNET" type="IP-SUBNET">255.255.255.192</P>
        <P xsi:type="tP_IP-GATEWAY" type="IP-GATEWAY">129.217.202.62</P>
        <P xsi:type="tP_OSI-PSEL" type="OSI-PSEL">00000001</P>
        <P xsi:type="tP_OSI-SSEL" type="OSI-SSEL">0001</P>
        <P xsi:type="tP_OSI-TSEL" type="OSI-TSEL">0001</P>
      </Address>
    </ConnectedAP>
  </SubNetwork>
</Communication>

```

Auch bei Modbus RTU nutzt die Applikation auf der lokalen Systeminstanz, die als Modbus-Client fungiert, unverändert das IEC-61850-Online-Datenmodell und den IEC-61850-Stack. Daher ist für dieses IED die Modellierung im Communication-Teil des SCL-Datenmodells unverändert.

Zur Modellierung der Kommunikationsschnittstelle, über die ein entfernter, serieller Modbus-RTU-Server angesprochen wird, werden hingegen eigene *P*-Adresselemente definiert (siehe Tabelle C.4).

Listing 4.11 zeigt beispielhaft die Modellierung der Kommunikationsbeziehungen für Modbus RTU.

Listing 4.11: Modellierung der Kommunikationsbeziehungen für Modbus RTU

```

<Communication>
  <SubNetwork name="SubNetwork1">
    <ConnectedAP iedName="Client" apName="AccessPoint1">
      <Address>
        <P xsi:type="tP_IP" type="IP">129.217.202.47</P>
        <P xsi:type="tP_IP-SUBNET" type="IP-SUBNET">255.255.255.192</P>
        <P xsi:type="tP_IP-GATEWAY" type="IP-GATEWAY">129.217.202.62</P>
        <P xsi:type="tP_OSI-PSEL" type="OSI-PSEL">00000001</P>
        <P xsi:type="tP_OSI-SSEL" type="OSI-SSEL">0001</P>
        <P xsi:type="tP_OSI-TSEL" type="OSI-TSEL">0001</P>
      </Address>
    </ConnectedAP>
    <ConnectedAP iedName="Server" apName="AccessPoint1">
      <Address>
        <P type="SERIAL-PORT">/dev/ttymx2</P>
        <P type="SERIAL-BAUDRATE">19200</P>
        <P type="SERIAL-MODE">RS485</P>
      </Address>
    </ConnectedAP>
  </SubNetwork>
</Communication>

```

Hardwarespezifische Konfigurationen

Der Zugriff auf analoge und binäre Ein- und Ausgänge des Systems ist hardwarespezifisch und damit von der konkreten Gerätetechnik abhängig. Ziel der Implementierung ist es dennoch, die hierzu erforderlichen Konfigurationsparameter ebenfalls im SCL-Datenmodell abzulegen. Es sind die Konfiguration der Messhardware der Systemplattform (untere Schicht gemäß Kapitel 4.1), das Einlesen analoger Messwerte sowie das Einlesen bzw. Setzen binärer Ein- und Ausgänge implementiert.

Konfiguration der Messhardware und Lesen analoger Messwerte

Die Konfiguration der Messhardware und das Lesen analoger Messwerte werden durch die beiden Funktionen *MeasurementSinglePhase* bzw. *MeasurementThreePhase* abstrahiert. Daher sind diese beiden Funktionen hardwarespezifisch und nicht ohne Anpassung an ihrem Algorithmus auf einer anderen Gerätetechnik einsetzbar. Aufgabe dieser Funktionen ist es, die gerätespezifisch abgerufenen Messwerte in normative logische Knoten (*MMXU*) abzulegen. Bis auf eine Auswahl dieser Funktionen im *Substation*-Teil des SCL-Datenmodells ist keine weitere Konfiguration erforderlich.

Einlesen bzw. Setzen binärer Ein- und Ausgänge

Das Einlesen bzw. Setzen binärer Ein- und Ausgänge ist mit Datenattributen gemäß IEC 61850 verknüpft. Die binären Eingänge werden zyklisch durch die Hardware-Abstraktions-Schicht abgefragt und deren Werte in Datenattribute geschrieben. Die Applikation kann somit direkt die Werte aus den Datenattributen des Online-Datenmodells auslesen. Zum Setzen binärer Ausgänge wiederum braucht die Applikation die Werte lediglich in die entsprechenden Datenattribute zu schreiben. Die Hardware-Abstraktions-Schicht übermittelt diese an die binären Ausgänge. Die Information, ob ein Datenattribut einen Eingang oder einen Ausgang repräsentiert, ergibt sich aus der Funktionseinschränkung (FC). Ein Datenattribut mit FC *Status* ist (aus Sicht der Leittechnik) ein Eingang, ein Datenattribut mit FC *Command* (engl. Befehl) ist entsprechend ein Ausgang.

Zur Konfiguration, welcher binäre Ein- bzw. Ausgang mit einem Datenattribut verknüpft ist, wird das Attribut *sAddr* eines Datenattributs genutzt. Hierbei wird die Konvention eingeführt, dass der Inhalt dem Format *IO(<Parameterliste>)* entsprechen muss, wobei *<Parameterliste>* eine durch Semikola getrennte Liste von Parametern ist. Listing 4.12 zeigt beispielhaft die Modellierung der Verknüpfung eines binären Eingangs mit einem Datenattribut.

Listing 4.12: Modellierung der Verknüpfung eines binären Eingangs mit einem Datenattribut

```
<LN desc="Generic Process I/O" InType="iA_GGIO" InClass="GGIO" inst="1">  
  <DOI name="SPCSO1">  
    <DAI desc="Binary Input" name="stVal" sAddr="IO(4)" />  
  </DOI>  
</LN>
```

Das Beispiel zeigt, dass der Zustand des binären Eingangs 4 der Applikation über das Datenattribut *GGIO1/SPCSO1/stVal* zur Verfügung gestellt wird. Dieses Datenattribut ist vom Basisdatentyp *BOOLEAN*. Auch jeweils ein einzelner binärer Ausgang kann einem Datenattribut vom Basisdatentyp *BOOLEAN* zugeordnet werden. Darüber hinaus definiert die IEC 61850-7-2 eine Vielzahl weiterer Basisdatentypen. Der Datentyp des Datenattributs zur Transformatorstufung *Tcmd* (engl.: Transformer Command) ist beispielsweise eine Enumeration mit den Werten *stop*, *lower* und *higher* (entsprechend binär *00*, *01* und *10*). Ein solches Datenattribut wird mit zwei binären Ausgängen, getrennt zum Herunter- und Heraufstufen, verknüpft. Die in der aktuellen Implementierung unterstützten Konfigurationsparameter können Tabelle D.1 (Anhang D) entnommen werden.

4.5.2 Automatische Gerätekonfiguration

Im vorherigen Kapitel wurden die zur Konfiguration der Systeme genutzten SCL-Sprachelemente dargestellt. In diesem Kapitel wird nun im Detail erläutert, wie die Programmkomponente *Autoconfig* auf Basis eines solchen SCL-Datenmodells eine automatische Gerätekonfiguration vornimmt. Die auf einem System ausgeführte Applikation enthält eine Reihe von Komponenten, auf die im Weiteren eingegangen wird. Zum besseren Verständnis ist in Abbildung 4.9 zunächst einmal das gesamte Objektmodell der Applikation dargestellt.

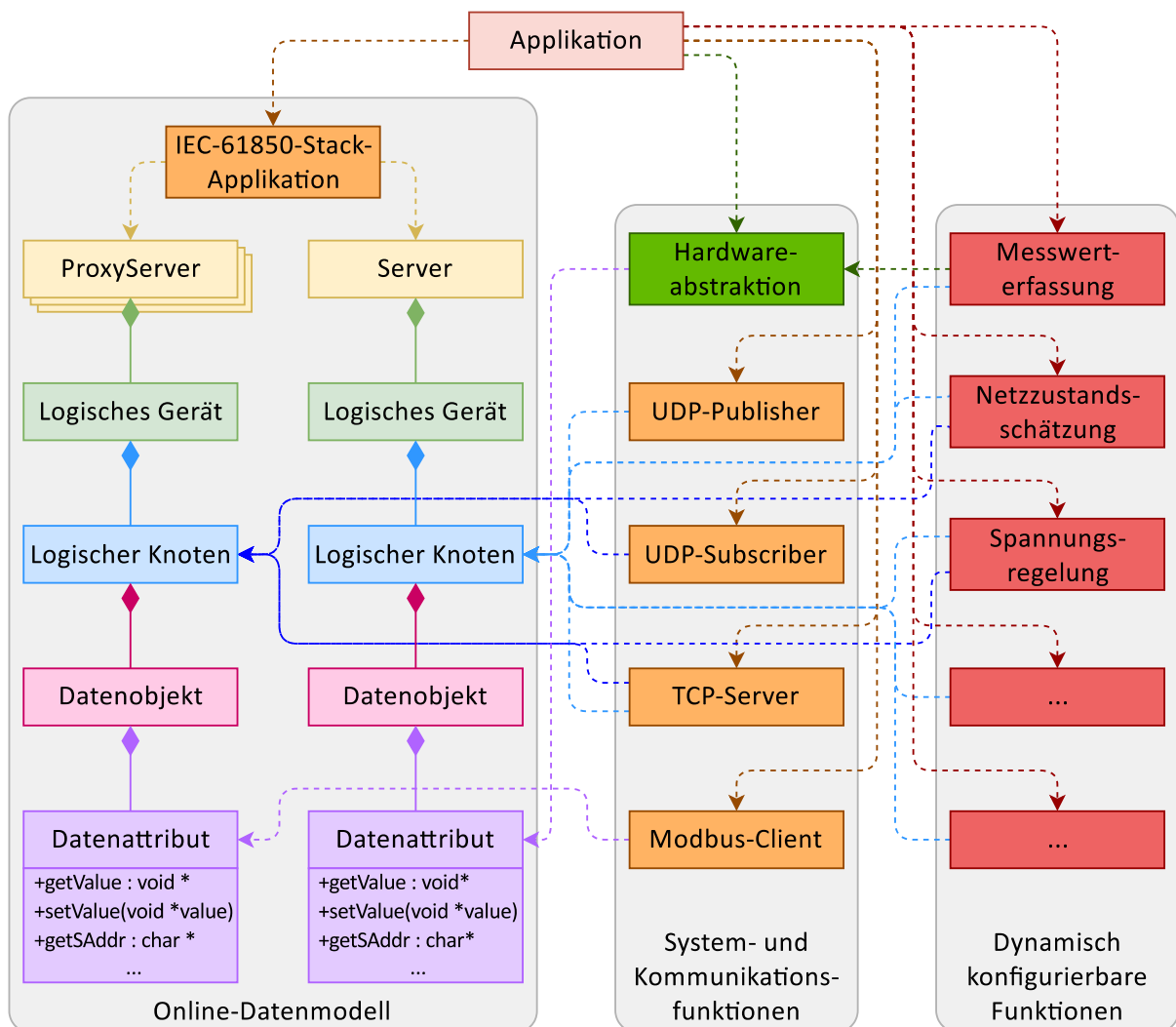


Abbildung 4.9: Objektmodell der Applikation

Im Zuge der automatischen Gerätekonfiguration ist es erforderlich, die SCL-Datenmodelldatei als auch weitere XML-basierte Dateien einzulesen, zu modifizieren und zu speichern. Hierzu wird als generischer XML-Parser die Open-Source-Bibliothek *XML* [131] eingesetzt. Diese Bibliothek prüft bzw. stellt sicher, dass die eingelesenen bzw. geschriebenen Dateien gültiges XML darstellen. Eine weitergehende Validierung gegen ein SCL-Schema erfolgt jedoch nicht. Vielmehr wird vorausgesetzt, dass die aus

dem vorhergehenden Engineeringprozess stammende Datei valide SCL-Daten enthält. Hiervon kann ausgegangen werden, da das für den Engineeringprozess genutzte Werkzeug *SCT* bereits eine automatische Validierung durchführt. Durch den Verzicht auf eine zusätzliche Validierung kann das Programm selbst schlanker gehalten werden.

Die automatische Gerätekonfiguration lässt sich in die in Abbildung 4.10 dargestellten Hauptschritte unterteilen. Diese werden im Folgenden im Detail erläutert.

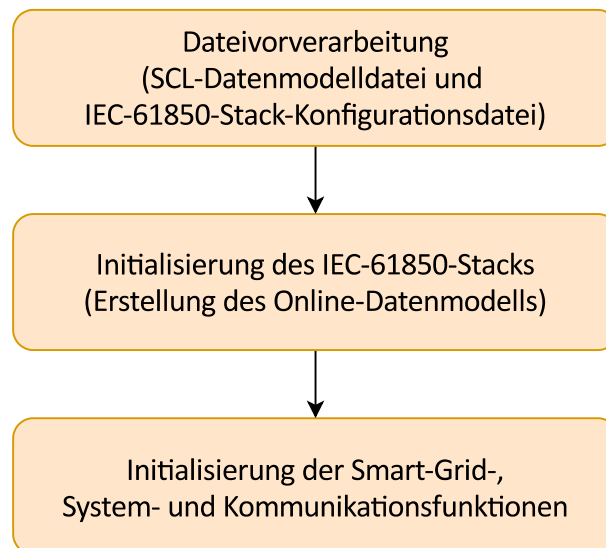


Abbildung 4.10: Hauptschritte der automatischen Gerätekonfiguration

Dateivorverarbeitung

Die Dateivorverarbeitung gliedert sich in folgende Schritte:

1. Die SCL-Datenmodelldatei (*autoconfig.scd*) wird eingelesen.
2. Ein SCL-Datenmodell kann die IED-Datenmodelle einer beliebigen Anzahl an Systeminstanzen enthalten. Daher muss das für die jeweilige Systeminstanz relevante IED-Datenmodell ermittelt werden. In der aktuellen Implementierung erfolgt die Identifizierung anhand der IP-Adresse(n) der konkreten Systeminstanz. Daher werden zunächst die IP-Adressen aller Netzwerkadapter der Systeminstanz ermittelt.
3. Im Communication-Teil des SCL-Datenmodells wird nach einem *ConnectedAP* gesucht, bei dem die IP-Adresse bzw. eine der IP-Adressen der Systeminstanz konfiguriert ist. Wird kein solcher *ConnectedAP* gefunden, erfolgt ein Programmabbruch.
4. Der *ConnectedAP* referenziert mit den Attributen *iedName* und *apName* ein IED und einen darin enthaltenen *AccessPoint*. Anhand dieser beiden Kriterien wird im IED-Teil des SCL-Datenmodells ein IED mit einem entsprechenden *AccessPoint* gesucht. Wird ein IED gefunden, so handelt es sich genau um jenes, welches die Systeminstanz repräsentiert, andernfalls erfolgt ein Programmabbruch.

5. Es werden alle entfernten IEDs ermittelt, die Daten wie Status- oder Messwerte an das lokale IED kommunizieren sollen (per MMS bzw. GOOSE). In den Datenmodellen dieser IEDs ist das lokale IED entsprechend in einem RptEnabled-Element innerhalb eines ReportControls referenziert, siehe auch Listing 4.4. Der IEC-61850-Stack erstellt später sowohl für das lokale IED ein Online-Datenmodell als auch für alle entfernten IEDs Proxy-Online-Datenmodelle.
6. Ausgehend von einer Template-Datei *autoconfig.config* wird auf Basis der ermittelten Informationen eine gerätespezifische Konfigurationsdatei *autoconfig_processed.config* für den IEC-61850-Stack erstellt. Diese enthält unter anderem Information über das lokale IED sowie die entfernten IEDs. Listing 4.13 zeigt einen Ausschnitt aus einer solchen Datei.
7. Zur Modellierung der Konfigurationsparameter des seriellen Modbus-RTU-Servers wurden, wie in Tabelle C.4 dargestellt, eigene *P*-Adresselemente definiert. Dies ist gemäß Standard (IEC 61850-6) zulässig, allerdings verweigert der IEC-61850-Stack seine Ausführung, sobald diese Adresselemente vorhanden sind. Daher werden diese *P*-Adresselemente, nachdem eine Kopie des entsprechenden Abschnitts des SCL-Datenmodells zur späteren Konfiguration der Modbus-Kommunikation erstellt wurde, aus dem SCL-Datenmodell entfernt.
8. Das so modifizierte SCL-Datenmodell wird als *autoconfig_processed.scd* gespeichert.

Listing 4.13: Konfigurationsdatei für den IEC-61850-Stack

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration>
  <ProductProperties>
    <Server enabled="true" />
    <Client enabled="true" />
    <GSE enabled="false" />
    <SMV enabled="false" />
    <SNTP enabled="false" />
  </ProductProperties>
  <SCL file="/home/kocos/autoconfig_processed.scd" />
  <Server>
    <IED name="Master" accessPoint="AccessPoint1" />
    <Settings>[...]</Settings>
  </Server>
  <Client>
    <IED name="Master" accessPoint="ClientAP" />
    <ProxyIED name="Slave1" accessPoint="AccessPoint1" instantiation="complete" />
    <ProxyIED name="Slave2" accessPoint="AccessPoint1" instantiation="complete" />
  </Client>
</Configuration>
```

Initialisierung des IEC-61850-Stacks

Im nächsten Schritt wird der IEC-61850-Stack initialisiert. Hierbei wird die zuvor geschriebene Konfigurationsdatei *autoconfig_processed.config* als Parameter übergeben. Diese verweist auf die SCL-Datenmodelldatei *autoconfig_processed.scd*. Beiden Dateien werden vom IEC-61850-Stack eingelesen. Aus dem SCL-Datenmodell erstellt der IEC-61850-Stack ein Online-Datenmodell (ACSI), das sowohl das lokale IED als auch alle Proxy-IEDs enthält. Die IEDs erscheinen im Online-Datenmodell als Server bzw. Proxy-Server. Die Anwendung kann anschließend durch das Online-Datenmodell navigieren, um etwa auf logische Knoten, Datenobjekte und Datenattribute zuzugreifen.

Initialisierung der Smart-Grid-, System- und Kommunikationsfunktionen

In diesem Schritt werden zunächst die Smart-Grid-Funktionen, die im Rahmen des Engineeringprozesses über das SCL-Datenmodell beschrieben wurden, instanziiert und initialisiert. Der detaillierte Ablauf sieht wie folgt aus:

1. Die Smart-Grid-Funktionen sind, wie bereits in Kapitel 4.5.1 in Listing 4.2 beschrieben, über *Function*-Elemente innerhalb eines *Bays* referenziert. Zu ihrer Identifizierung muss daher zunächst das dem lokalen IED zugeordnete Bay gefunden werden. Gemäß zuvor aufgestellter Konvention müssen IED und Bay denselben Namen (Attribut *name*) haben. Daher wird im Substation-Teil des SCL-Datenmodells nach einem Bay gesucht, das denselben Namen trägt wie das lokale IED. Wird kein solches Bay gefunden, erfolgt ein Programmabbruch.
2. In dem ermittelten Bay wird über alle untergeordneten *Functions* iteriert.
3. Die *Autoconfig* besitzt ein Verzeichnis aller implementierten Smart-Grid-Funktionen. Die Identifikation der konkreten Smart-Grid-Funktionen erfolgt über das Funktionsschlüsselwort, welches im *type*-Attribut einer *Function* gespeichert ist. Anhand dieses Attributs sucht die *Autoconfig* für jede *Function* in ihrem Funktionsverzeichnis nach einem entsprechenden Eintrag. Bei negativem Resultat erfolgt ein Programmabbruch.
4. Bei positivem Resultat wird eine neue Instanz der Smart-Grid-Funktionsklasse erstellt. Hierzu besitzt jede Smart-Grid-Funktionsklasse eine instanzunabhängige (statische) *Create*-Methode.
5. Eine *Function* kann *LNodes* als abstrakte logische Knoten enthalten, die auf LNs in IEDs verweisen. Anhand der vollständigen LN-Referenz (zusammengesetzt aus den Attributen gemäß Listing 4.2) wird im Server und in den Proxy-Servern des Online-Datenmodells nach entsprechenden LN-Instanzen gesucht. Sollte eine LN-Instanz nicht gefunden werden (beispielsweise aufgrund eines fehlerhaften *LNodes*), so führt dies zum Programmabbruch.

6. Einige Smart-Grid-Funktionen benötigen über die normativen Elemente der IEC 61850-6 hinausgehende Informationen, die mit einem privaten Element innerhalb einer *Function* beschrieben sind.
7. Abschließend wird an der Smart-Grid-Funktionsinstanz eine *Init*-Methode aufgerufen. Als Parameter wird jeweils eine Liste aller Server (lokaler Server sowie Proxy-Server), aller Referenzen auf zugehörige LN-Instanzen sowie aller privaten Elemente übergeben. Die Signatur der *Init*-Methode ist in Listing 4.14 dargestellt.

Listing 4.14: Signatur der *Init*-Methode einer Smart-Grid-Funktion

```
bool Init(const vector<GenServerClass*> &Servers, const vector<GenLogicalNodeClass*>
        &LNs, const vector<XMLElement*> &Privates)
```

Anschließend erfolgt die Initialisierung der System- und Kommunikationsfunktionen. Diese sind fest in die Applikation integriert, d. h. sie werden immer instanziiert und ausgeführt, können aber über das SCL-Datenmodell konfiguriert werden.

Zunächst wird als Systemfunktionen das *HardwareAbstractionLayer* initialisiert. Über diese Funktion wird eine Verbindung zum DAQ-/IO-Modul hergestellt, so dass Messwerte abgerufen sowie analoge und binäre Ausgänge gesetzt werden können. Hierbei wird außerdem der *BinaryOutputManager* konfiguriert. Dieser ist dafür zuständig, bei Wertänderungen von Datenattributen im Online-Datenmodell automatisch entsprechend verknüpfte binäre Ausgänge zu setzen. Die Konfiguration des *BinaryOutputManagers* erfolgt im Detail wie folgt:

1. Im Online-Datenmodell des lokalen Servers wird über alle Datenattribute iteriert.
2. Enthält das Attribut *sAddr* eines Datenattributs einen Inhalt, der dem in Kapitel 4.5.1 definierten Format *IO(<Parameterliste>)* entspricht, so wird die Parameterliste geparkt.
3. Abhängig vom Datentyp des Datenattributs und dem Inhalt der Parameterliste (siehe Tabelle D.1) wird eine passende *Add*-Methode an der *BinaryOutputManager*-Instanz aufgerufen. Die Methodensignaturen sind in Listing 4.15 dargestellt.

Listing 4.15: Signaturen der *Add*-Methoden des *BinaryOutputManagers*

```
bool BinaryOutputManager::Add(GenConstructedAttributeClass *DA, unsigned int channel)
bool BinaryOutputManager::Add(GenConstructedAttributeClass *DA, unsigned int
    channelLower, unsigned int channelHigher, unsigned int pulseLength)
```

Abschließend werden die einzelnen Kommunikationsfunktionen wie nachfolgend beschrieben konfiguriert.

UDP-Publisher

Zur Konfiguration des UDP-Publishers werden die als Reporting gemäß IEC 61850-6 modellierten Kommunikationsbeziehungen in den IED-Datenmodellen genutzt.

1. Im SCL-Datenmodell des lokalen IEDs wird über alle *ReportControls* iteriert.
2. Jeder *ReportControl* referenziert ein *DataSet*.
3. Jedes *DataSet* referenziert eine beliebige Anzahl an *Datenobjekten* und *Datenattributen*.
4. Es werden alle *LN*s gesammelt, die den referenzierten Datenobjekten und Datenattributen übergeordnet sind.
5. Zu den gesammelten *LN*s aus dem SCL-Datenmodell werden die Entsprechungen im Online-Datenmodell des lokalen Servers gesucht. Die Werte dieser *LN*s werden durch den UDP-Publisher versendet.
6. Im *ReportControl* ist als weitere Information enthalten, wie der Versand der Werte erfolgen soll, d. h. ob bei Wertänderung oder zyklisch unter Angabe des Intervalls.
7. An der UDP-Publisher-Instanz wird für jeden zu versendenden *LN* die *Add*-Methode aufgerufen, dessen Signatur in Listing 4.16 dargestellt ist. Als IP-Adresse wird die Broadcast-Adresse übergeben, d. h. alle UDP-Pakete werden als Broadcast in das Netzwerk gesendet, als Port wird 44227 genutzt. Die letzten beiden Parameter sind die unter 6. genannten Informationen zur Art des Versands.

Listing 4.16: Signatur der Add-Methode des UDP-Publishers

```
bool UdpPublisher::Add(GenLogicalNodeClass *LN, const char *ip, unsigned short port, bool  
    sendOnDataChange, unsigned int sendPeriodInterval)
```

UDP-Subscriber

Auch zur Konfiguration des UDP-Subscribers werden die als Reporting modellierten Kommunikationsbeziehungen in den IED-Datenmodellen genutzt.

1. Im SCL-Datenmodell aller entfernter IEDs wird über alle *ReportControls* iteriert.
2. Wird innerhalb eines *ReportControls* (untergeordnet über ein Element *RptEnabled/ClientLN*) das lokale IED referenziert, so soll das lokale IED die Daten dieses Reports empfangen.
3. Analog zum UDP-Publisher Punkt 2.-5. werden auch hier die *LN*s gesammelt, allerdings in den entfernten IEDs bzw. den Proxy-Servern. Die Werte dieser *LN*s werden durch den UDP-Subscriber empfangen.
4. An der UDP-Subscriber-Instanz wird für jeden zu empfangenen *LN* die *Add*-Methode aufgerufen, dessen Signatur in Listing 4.17 dargestellt ist.

Listing 4.17: Signatur der Add-Methode des UDP-Subscribers

```
bool UdpSubscriber::Add(GenLogicalNodeClass *LN)
```


TCP-Server

Der TCP-Server wird mit den LNs aller Server des Online-Datenmodells (lokaler Server und Proxy-Server) konfiguriert.

1. Im Online-Datenmodell wird über alle Server iteriert.
2. Für jeden Server wird über alle untergeordneten LNs iteriert.
3. An der TCP-Server-Instanz wird für jeden LN die *Add*-Methode aufgerufen, dessen Signatur in Listing 4.18 dargestellt ist.

Listing 4.18: Signatur der Add-Methode des TCP-Servers

```
bool TcpServer::Add(GenLogicalNodeClass *LN)
```

Modbus-Client

Im Engineeringprozess werden Modbus-Register, wie in Kapitel 4.4.4 beschrieben, direkt mit Datenattributen innerhalb des SCL-Datenmodells verknüpft. Zu lesende Daten werden periodisch vom Modbus-Server abgefragt und in die entsprechenden Datenattribute geschrieben. Zu schreibende Daten werden aus den entsprechenden Datenattributen gelesen und nur bei Wertänderung in den Modbus-Server geschrieben. Die Kommunikationsparameter wie Name und Baudrate der seriellen Schnittstelle bzw. IP-Adresse sind im Communication-Teil des SCL-Datenmodells zu finden. Die Konfiguration des Modbus-Clients erfolgt im Detail wie folgt:

1. Im Online-Datenmodell wird über alle Proxy-Server iteriert.
2. Für jeden Proxy-Server werden die Namen des zugrunde liegenden IEDs und AccessPoints abgerufen.
3. Im Communication-Teil des SCL-Datenmodells wird mit dem IED- und AccessPoint-Namen nach einem entsprechenden ConnectedAP gesucht.
4. Aus dem ConnectedAP werden die Kommunikationsparameter entnommen. Ist hier eine serielle Schnittstelle beschrieben (mit den Parametern gemäß Tabelle C.4 und Tabelle C.5), so repräsentiert der Proxy-Server einen Modbus-RTU-Server. Ist hingegen eine IP-Adresse beschrieben, so repräsentiert der Proxy-Server einen Modbus-TCP-Server.
5. Für jeden Proxy-Server wird über alle Datenattribute iteriert.
6. Enthält das Attribut *sAddr* eines Datenattributs einen Inhalt, der dem in Kapitel 4.5.1 definierten Format *Modbus(<Parameterliste>)* entspricht, so wird die Parameterliste geparkt (siehe Tabelle C.1, Tabelle C.2 und Tabelle C.3).
7. Abhängig von der Rolle des Proxy-Servers gemäß Punkt 4. wird eine passende *Add*-Methode an der Modbus-Client-Instanz aufgerufen. Der Methode werden die Parameter aus der Parameterliste sowie die Kommunikationsparameter übergeben. Die Methodensignaturen sind in Listing 4.19 dargestellt.

Listing 4.19: Signaturen der Add-Methoden des Modbus-Clients

```
bool ModbusClient::AddRTU(GenConstructedAttributeClass *DA, uint8_t slaveAddress,  
    E_REGISTER_TYPE registerType, uint16_t registerAddress, E_DATA_TYPE dataType,  
    bool swapByte, bool swapWord, const string &port, unsigned int baudRate,  
    bool modelsRS485)
```

```
bool ModbusClient::AddTCP(GenConstructedAttributeClass *DA, uint8_t slaveAddress,  
    E_REGISTER_TYPE registerType, uint16_t registerAddress, E_DATA_TYPE dataType,  
    bool swapByte, bool swapWord, const string &ip, unsigned short port)
```

4.5.3 Programmablaufsteuerung

Nach der Konfiguration und Initialisierung aller Funktionen durch die *Autoconfig* werden als Nächstes die System- und Kommunikationsfunktionen gestartet. Beispielsweise öffnet das *HardwareAbstractionLayer* als Systemfunktionen hierbei den seriellen Kommunikationskanal zum DAQ-/IO-Modul und versetzt letzteres in den Betriebsmodus. Die Kommunikationsfunktionen starten ihre Netzwerkfunktionalität, wobei etwa lokale Ports geöffnet und gebunden (Server) oder Verbindungen zu entfernten Kommunikationspartnern aufgebaut werden (Client). Anschließend wird ein neuer Thread erzeugt, in welchem die zyklische Ausführung der einzelnen Smart-Grid-, System- und Kommunikationsfunktionen stattfindet. Hierzu implementiert jede Funktion eine *Run*-Methode, innerhalb der ihr jeweiliger Kernalgorithmus ausgeführt wird. Die Signatur dieser Methode, die in der Basisklasse *FunctionBase*, von der jede Funktionsklasse abgeleitet ist, deklariert ist, ist in Listing 4.20 dargestellt.

Listing 4.20: Signatur der Run-Methode

```
bool Run()
```

Abbildung 4.11 zeigt den gesamten Programmablauf. Die zyklische Funktionsabarbeitung (Run-Thread) gliedert sich dabei in folgende Schritte:

- Zunächst werden die Funktionen der Netzwerkkommunikation (IEC 61850-8-1, UDP, TCP, Modbus) aufgerufen. Dadurch werden von entfernten Systeminstanzen eingetroffene Daten verarbeitet, d. h. beispielsweise Werte im Online-Datenmodell aktualisiert und im vorherigen Programmdurchlauf veränderte lokale Daten an entfernte Systeminstanzen kommuniziert.
- Die lokalen Messwerte werden von der Messhardware eingelesen.
- Die Smart-Grid-Funktionen werden nacheinander aufgerufen, wobei jede Funktion die jeweils relevanten Messwerte und Parameter aus dem Online-Datenmodell einliest und die erforderliche Verarbeitung durchführt, etwa den Vergleich eines Messwerts mit einem Schwellwert oder die Ausführung eines Regelalgorithmus. Ergebnisse werden in das Online-Datenmodell zurückgeschrieben.

- Schließlich werden die Ausgaben der Smart-Grid-Funktionen wie Auslösebefehle, Transformatorstufungsbefehle oder Sollwerte über das DAQ-/IO-Modul auf Ausgänge des Systems zurückgeschrieben.

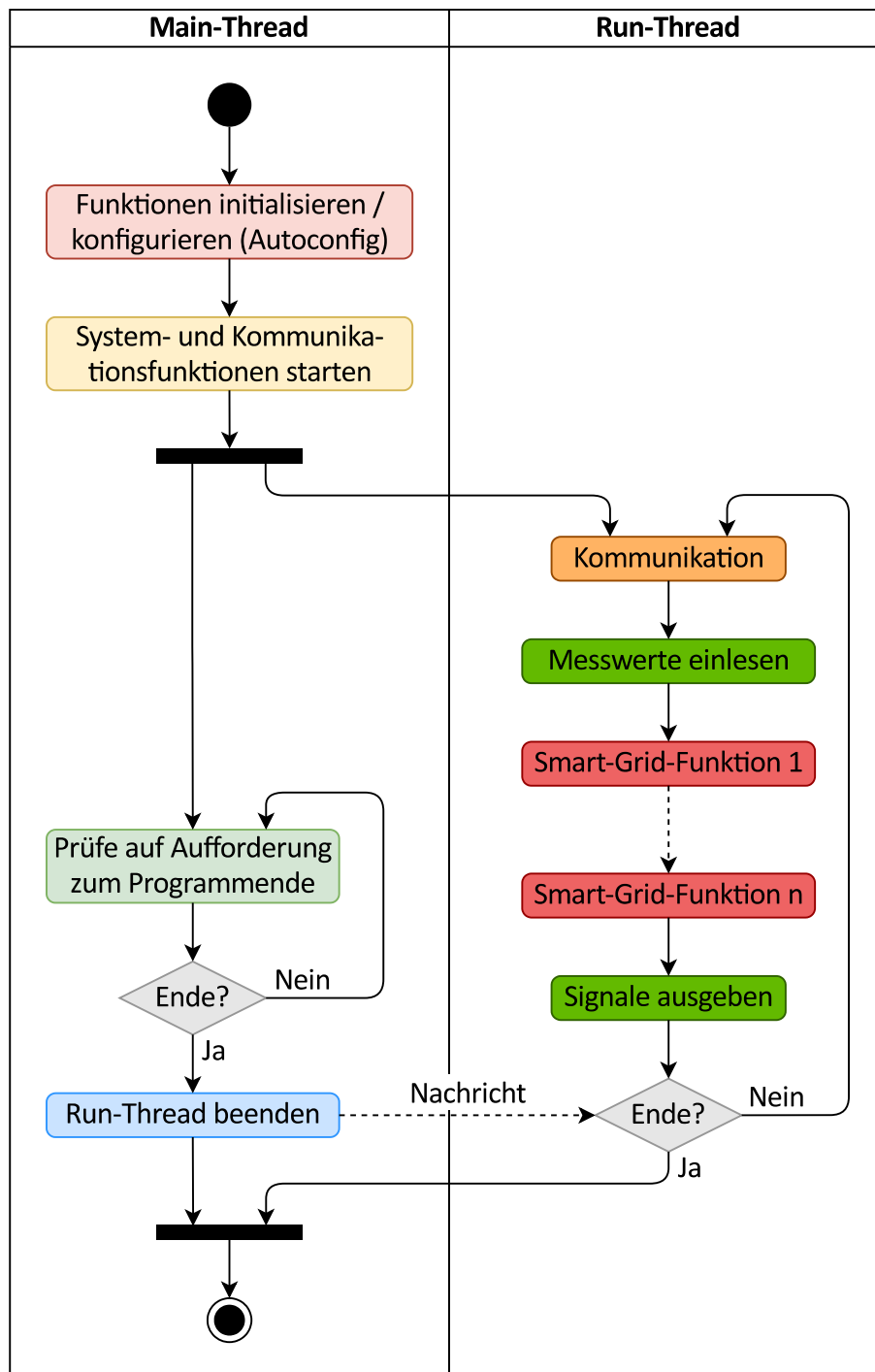


Abbildung 4.11: Programmablauf

Im Main-Thread wird kontinuierlich geprüft, ob eine Aufforderung zum Programmende (Signal *SIGINT* oder *SIGTERM*) vorliegt. In diesem Fall wird der Run-Thread darüber informiert, so dass er sich am Ende des Durchlaufs beendet.

Für das Beispiel der Spannungsregelung aus Abbildung 4.4 ergibt sich der zyklische Programmablauf für das Mastergerät (Systeminstanz obere Bildhälfte) wie folgt:

1. Die Funktionen der Netzwerkkommunikation werden aufgerufen. Damit werden beispielsweise von entfernten Systeminstanzen eingetroffene Messwerte (logische Knoten MMXU) verarbeitet sowie im vorherigen Programmdurchlauf gesetzte Sollwerte (logische Knoten FSPT) an entfernte Systeminstanzen kommuniziert.
2. Die Messwerterfassungsfunktion *MeasurementThreePhase* wird aufgerufen:
 - a. Über das in Abbildung 4.3 dargestellte API, das der Funktion implizit bekannt ist, greift diese auf die entsprechenden Instrumente des DAQ-/IO-Moduls zu und ruft die Rohmesswerte (Spannungen, Ströme, Leistungen, Impedanzen etc.) ab.
 - b. Aus den logischen Knoten TCTR und TVTR werden etwaige Wandlerverhältnisse als Parameter eingelesen.
 - c. Die Wandlerverhältnisse werden auf die Rohmesswerte angewandt und die so skalierten Messwerte in den logischen Knoten MMXU geschrieben.
3. Die Master-Funktion der Spannungsregelung *VoltageControl_Master* wird aufgerufen:
 - a. Es werden die lokalen Messwerte (aus dem logischen Knoten MMXU), aber auch Messwerte entfernter Systeminstanzen (über weitere logische Knoten MMXU, die als Proxy fungieren und die Werte über einen Kommunikationskanal (IEC 61850-8-1, UDP) enthalten haben) eingelesen.
 - b. Auf Basis der Messwerte berechnet der Algorithmus der Spannungsregelung etwaige Maßnahmen wie die Stufung des lokalen Transformators oder neue Sollwertvorgaben für dezentrale Anlagen (Blindleistungsregelung), die an entfernten Systeminstanzen angeschlossen sind.
 - c. Soll der Transformator gestuft werden, so wird ein entsprechender Befehl in den logischen Knoten ATCC geschrieben.
 - d. Wurde ein neuer Sollwert für eine dezentrale Anlage berechnet, so wird dieser als Sollwertbefehl in den logischen Knoten FSPT geschrieben. Die Kommunikation an die entfernte Systeminstanz erfolgt zu Beginn des nächsten Programmdurchlaufs.
4. Die Funktion der Transformatorstufungssteuerung *ATCC_Function* wird aufgerufen:
 - a. Die Funktion liest den logischen Knoten ATCC ein.
 - b. Liegt ein Befehl zur Stufung des Transformators vor, so wird durch einen API-Zugriff auf das entsprechende Instrument des DAQ-/IO-Moduls ein binärer Ausgang der lokalen Systeminstanz gesetzt.
 - c. Anschließend wird der Befehl im logischen Knoten ATCC zurückgesetzt.

Für Funktionen, die eine längere Rechenzeit erfordern, wie etwa die Netzzustands-schätzung, wird vom *Run-Thread* aus jeweils ein weiterer Thread gestartet, der über mehrere Iterationen des *Run-Threads* hinweg nebenläufig ausgeführt wird. Abbildung 4.12 zeigt dies.

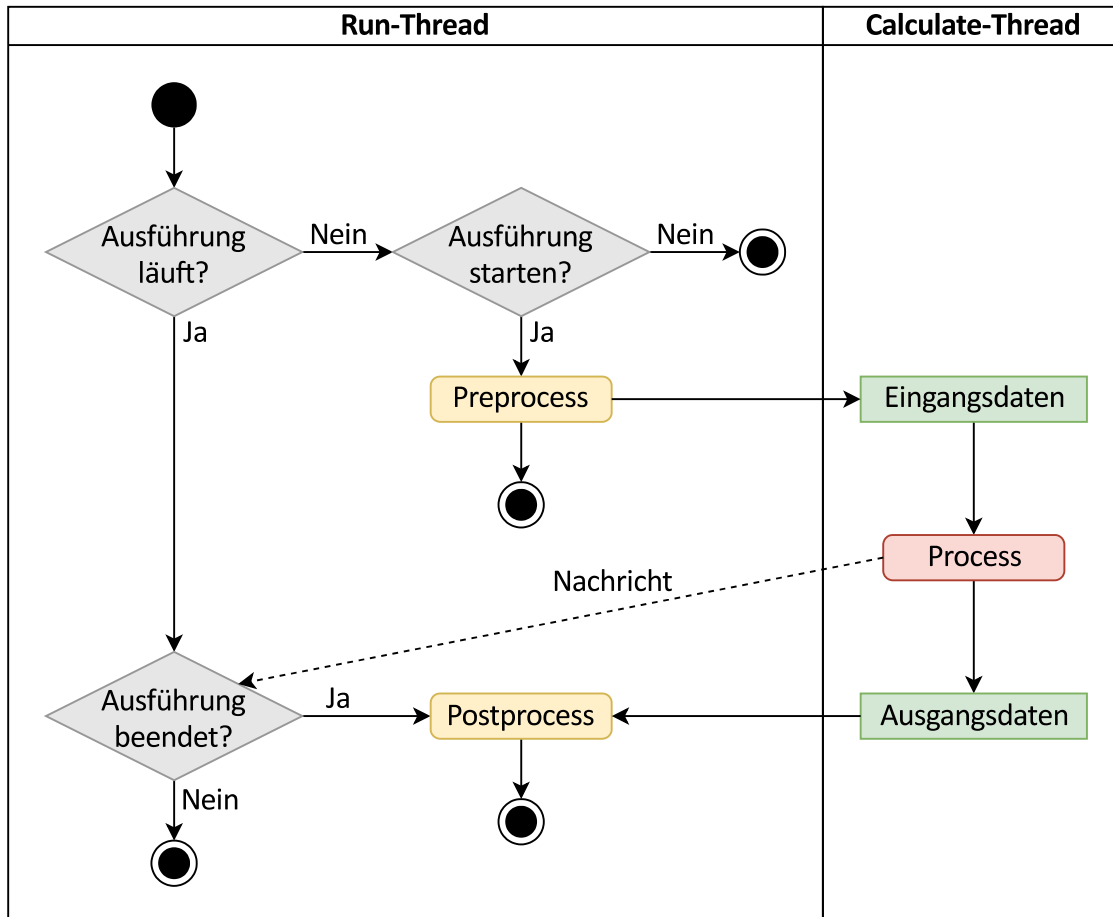


Abbildung 4.12: Programmablauf einer rechenzeitintensiven Funktion

Die Ausführung einer rechenzeitintensiven Funktion kann etwa zyklisch in definierten Intervallen oder befehls-gesteuert erfolgen. Soll eine neue Ausführung gestartet werden, so wird im *Run-Thread* zunächst die Methode *Preprocess* ausgeführt. Diese Methode bereitet die von der rechenzeitintensiven Funktion benötigten Eingangsdaten wie etwa Messwerte vor, indem diese aus dem Online-Datenmodell in eine separate Speicherstruktur kopiert werden. Anschließend wird ein neuer Thread (*Calculate-Thread*) erzeugt, der die Methode *Process* ausführt. Innerhalb dieser Methode wird der rechenzeitintensive Algorithmus ausgeführt, der hierbei ausschließlich auf die in der Speicherstruktur gespeicherten Eingangsdaten zugreift. Dies löst das Problem des kritischen Abschnitts, das andernfalls auftreten würde, sobald mehrere Threads (*Run-Thread* und *Calculate-Thread*) zeitgleich auf eine Ressource wie beispielsweise das Online-Datenmodell zugreifen würden.

Sobald der Algorithmus im *Calculate*-Thread abgeschlossen ist, informiert die Methode *Process* den *Run*-Thread mittels einer entsprechenden Nachricht. Daraufhin verarbeitet die Methode *Postprocess* die Ausgangsdaten der rechenzeitintensiven Funktion wie Berechnungsergebnisse und kopiert diese aus einer separaten Speicherstruktur in das Online-Datenmodell.

Die Ausführungszeit der Methoden *Preprocess* und *Postprocess* ist verhältnismäßig kurz, so dass andere Funktionen, die im *Run*-Thread ausgeführt werden, nicht unverhältnismäßig stark verzögert werden.

4.6 Zusammenfassung

In diesem Kapitel wurde dargelegt, wie die einzelnen Bestandteile des Gesamtsystems umgesetzt wurden. Zunächst wurde die im Rahmen dieser Arbeit genutzte Systemplattform vorgestellt und ein Überblick über deren Architektur und technischen Kennwerte gegeben. Es soll an dieser Stelle noch einmal betont werden, dass es sich um eine exemplarische Plattform handelt. Diesem Umstand wurde insbesondere durch klar getrennte Schnittstellen zur Software Rechnung getragen. Eine Übertragung der Softwarefunktionen auf eine andere Hardwareplattform ist explizit vorgesehen und wird in parallelen Forschungsarbeiten bereits untersucht. Anschließend wurde das erstellte Softwareframework vorgestellt und es wurde aufgezeigt, wie sich Smart-Grid-Funktionen und Algorithmen innerhalb dieses Softwareframeworks umsetzen lassen. Die Implementierung ausgewählter Funktionen, hier des Distanzschutzes und der Netzzustandsschätzung, wurde als Nächstes vorgestellt. Außerdem wurde ein Überblick über weitere Funktionen, die in parallelen Forschungsarbeiten implementiert wurden bzw. werden, gegeben. Des Weiteren wurde die Umsetzung verschiedener Kommunikationsverfahren vorgestellt. Diese Verfahren ermöglichen es einerseits, dass mehrere Instanzen der Systemplattform untereinander kommunizieren, und andererseits, dass Dritt- und Fremdkomponenten an die Systemplattform angebunden werden können. Im letzten Teil dieses Kapitels wurde schließlich der Konfigurationsprozess, ein besonderes Merkmal des Systems, im Entwurf und in der Implementierung vorgestellt. Es wurde aufgezeigt, wie unter Nutzung der SCL-Sprachelemente die auf dem System auszuführenden Funktionen, die auszutauschenden Daten und die Kommunikationsbeziehungen – auch für Kommunikationsverfahren außerhalb der Normenreihe IEC 61850 – modelliert werden können. Weiterhin wurden die automatische Gerätekonfiguration unter Nutzung dieser Datenmodelle sowie schließlich die Ausführung und Koordination von Funktionen im Sinne der Programmablaufsteuerung im Detail dargelegt.

5 Verifikation der Smart-Grid-Automatisierungsarchitektur

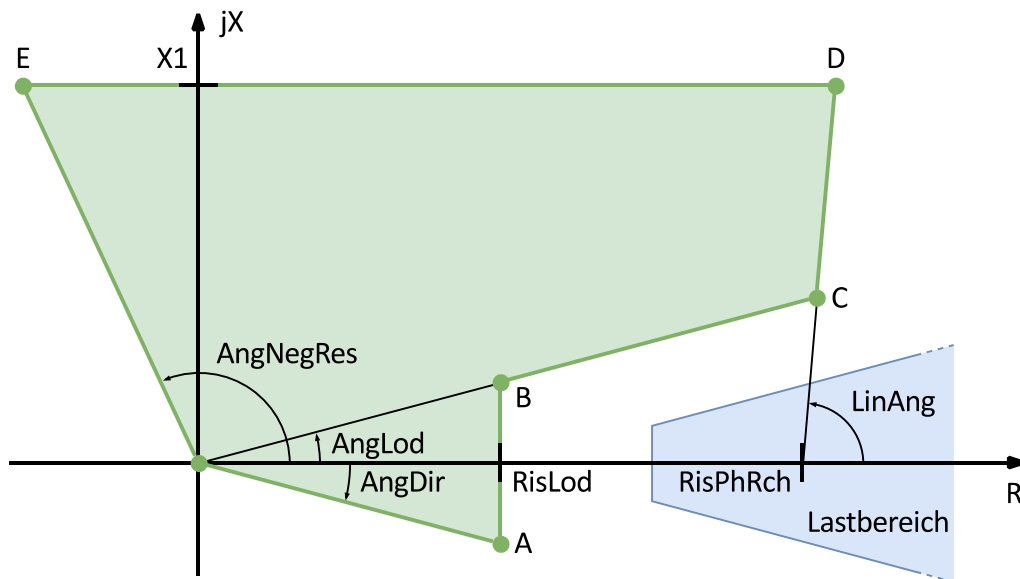
In diesem Kapitel wird das ordnungsgemäße Verhalten ausgewählter Funktionen und des Gesamtsystems selbst verifiziert. Im ersten Teil wird hierzu der Distanzschutz als eine der realisierten Schutzfunktionen betrachtet und in Bezug auf Messgenauigkeit und Auslösezeiten untersucht. Der zweite Teil zeigt exemplarisch die vollständige Durchführung des Engineering- und Konfigurationsprozesses am Beispiel der vorbereitenden Arbeiten für einen Feldtest auf. Als Smart-Grid-Funktion kommt hierbei die Netzzustandsschätzung zum Einsatz. Es werden das Verhalten dieser Funktion sowie ihre Ergebnisse unter realen Betriebsbedingungen während des Feldtests dargestellt. Abschließend folgt eine Zusammenfassung dieses Kapitels.

5.1 Schutzfunktionen

Als Schutzfunktionen wurden sowohl ein Überstromzeitschutz als auch ein Distanzschutz implementiert. Die erste Umsetzung auf Ebene des Linux-Betriebssystems ermöglichte durch den dort fehlenden Zeitdeterminismus keinen produktiven Einsatz und ließ auch keine quantitativen Aussagen zum Zeitverhalten zu. Daher folgte im zweiten Schritt eine Umsetzung auf dem Co-Prozessor, der durch Nutzung eines Echtzeitbetriebssystems eine zeitdeterministische Funktionsausführung ermöglicht. Der Distanzschutz weist im Vergleich zum Überstromzeitschutz einen komplexeren Algorithmus auf, erfordert umfangreichere Berechnungen und stellt höhere Anforderungen an die Messwerte, insbesondere an die Winkelinformationen. Daher beschränkt sich die nachfolgende Verifikation auf den Distanzschutz. Primär wird die Umsetzung auf Ebene des Co-Prozessors betrachtet, da dies einerseits quantitative Aussagen zum Zeitverhalten ermöglicht und sich andererseits eine größere Aussagekraft bezüglich der Messgenauigkeit des Gesamtsystems ergibt. Den hierbei erlangten Ergebnissen werden zum Vergleich die Ergebnisse bei Ausführung des Distanzschutzes auf Ebene des Linux-Betriebssystems gegenübergestellt.

5.1.1 Eigenschaften des implementierten Distanzschutzes

Der Distanzschutz beruht auf einer Impedanzmessung in der komplexen Ebene, die Fehlererkennung basiert dabei auf einer Unterimpedanzauslösung. Bei Fehlereintritt verringert sich der aus den Spannungs- und Strommesswerten ermittelte Impedanzbetrag, außerdem ändert sich der Impedanzwinkel. Der hier umgesetzte Distanzschutz basiert auf dem Prinzip der Impedanzpolygon-Charakteristik mit Lastausparung (im Gegensatz zu einer MHO- bzw. Admittanz-Charakteristik), siehe auch [132, S. 24 ff., 41 ff.]. Der Fehlerzustand eines Betriebsmittels, etwa einer Leitung, wird durch eine polygonale Ebene beschrieben, welche als Auslösekriterium dient. Abbildung 5.1 zeigt ein solches Polygon mit den charakterisierenden Parametern.


Abbildung 5.1: Impedanzpolygon des Distanzschutzes

Im Normalbetrieb liegt die Impedanz im Lastbereich mit einem verhältnismäßig größeren resistiven (ohmschen) Anteil. Durch die Winkeländerung der Impedanz und die Impedanzreduktion dreht sich der Impedanzzeiger im Fehlerfall in die Polygonfläche. Dies wird vom Distanzschutzalgorithmus detektiert und führt zur Auslösung. Die Bedeutung der charakterisierenden Parameter wird in Tabelle 5.1 erläutert. Die Spalte *Wert* zeigt überdies die für die im weiteren Verlauf durchgeführte Verifikation konkret gewählten Parameter.

Tabelle 5.1: Parameter des Impedanzpolygons

| Parameter | Beschreibung | Wert |
|-----------|---|-------------|
| RisLod | Ohmsche Reichweite des Lastbereichs (engl.: Resistive Reach for Load Area) | 40 Ω |
| RisPhRch | Ohmsche Phasenreichweite (engl.: Resistive Phase Reach) | 80 Ω |
| AngLod | Winkel des Lastbereichs (engl.: Angle for Load Area) | 15° |
| LinAng | Leitungswinkel (engl.: Line Angle) | 85° |
| X1 | Mitsystem-Reichweite der Leitungsreaktanz (engl.: Positive Sequence Line (Reach) Reactance) | 50 Ω |
| AngNegRes | Negativer Winkel des gerichteten Bereichs (engl.: Negative Angle for Directional Area) | 115° |
| AngDir | Winkel des gerichteten Bereichs (engl.: Angle for Directional Area) | -15° |

Tabelle 5.2 zeigt die Vorschriften zur Berechnung der Polygon-Eckpunkte anhand der charakterisierenden Parameter.

Tabelle 5.2: Vorschriften zur Berechnung der Polygon-Eckpunkte

| Eckpunkt | Berechnung der Koordinaten (R, X) |
|----------|--|
| A | $(RisLod, \tan(AngDir) \cdot RisLod)$ |
| B | $(RisLod, \tan(AngLod) \cdot RisLod)$ |
| C | Schnittpunkt der Schenkel der Winkel $AngLod$ und $LinAng$ |
| D | $(RisPhRch + X1 / \tan(LinAng), X1)$ |
| E | $(-X1 \cdot \tan(AngNegRes - 90^\circ), X1)$ |

5.1.2 Prüfablauf

Zur Prüfung des Distanzschutzes werden zehn Prüfpunkte ausgewählt, die sich auf der Polygonlinie im Bereich eines Impedanzwinkels von 0° bis 90° befinden. Für jeden Prüfpunkt werden wiederum 21 Prüfschritte definiert, die sich auf einer Geraden senkrecht zur jeweiligen Polygonlinie befinden. Jeweils zehn Schritte liegen dabei außerhalb und innerhalb des Polygons, der mittlere Prüfschritt liegt auf der Polygonlinie. Der äußerste bzw. innerste Prüfschritt befindet sich +10 % bzw. -10 % vom mittleren Prüfschritt entfernt (bei den Prüfschritten 1-4 bezogen auf R, bei den Prüfschritten 5-10 bezogen auf X).

Im Rahmen der Untersuchungen wird ein einstufiger Distanzschutz betrachtet. Abbildung 5.2 zeigt das gewählte Impedanzpolygon mit Prüf- und Pausenimpedanzen.

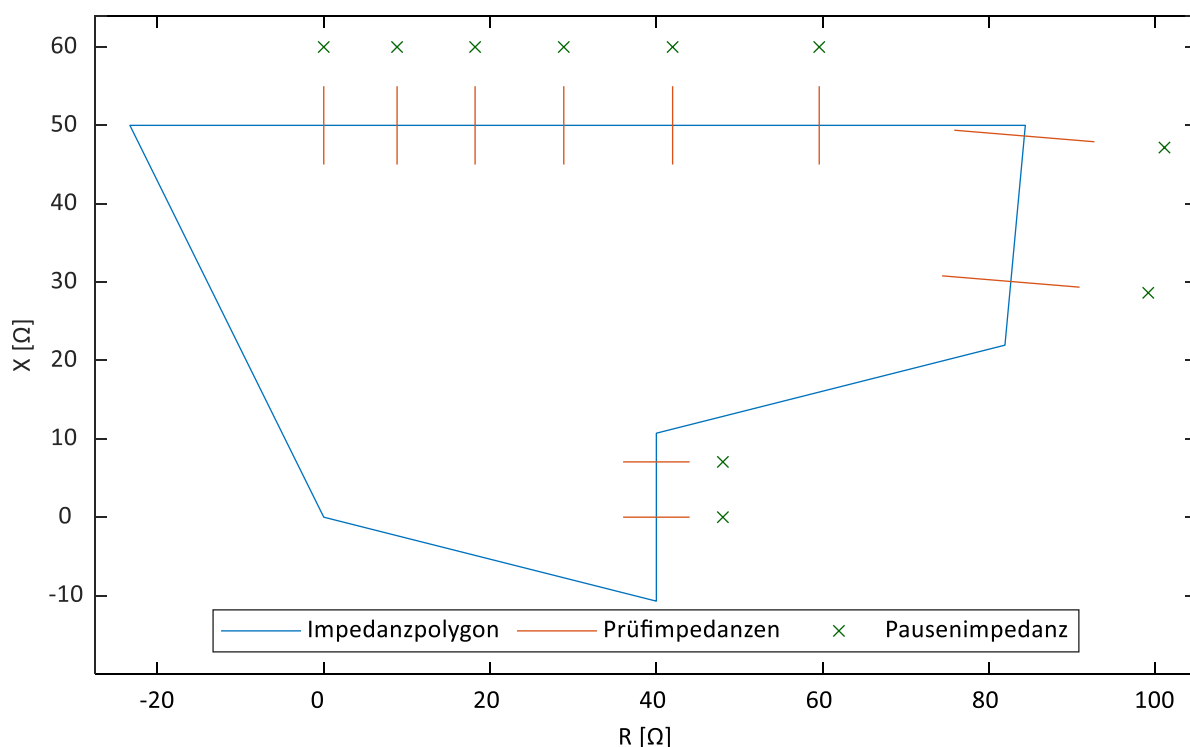


Abbildung 5.2: Impedanzpolygon mit Prüf- und Pausenimpedanzen

Zur Generierung der Spannungs- und Stromsignale sowie zur automatisierten Steuerung des Prüfablaufs wird ein Echtzeitsimulator des Herstellers *OPAL-RT Technologies* [133] genutzt. Hierbei ist das zu prüfende System in geschlossener Schleife (engl.: Closed loop) an den Echtzeitsimulator angekoppelt, so dass Schutzanregungen und -auslösungen Rückfluss in die Simulation finden. Dadurch ist eine Messung der Reaktionszeiten möglich. Da der Echtzeitsimulator lediglich Kleinsignale im Bereich ± 10 V liefert, das zu prüfende System jedoch mit Standardgrößen von 100 V bzw. 1 A arbeitet, sind zwischen Echtzeitsimulator und zu prüfendem System Spannungs- bzw. Stromverstärker geschaltet. Abbildung 5.3 zeigt den gesamten Aufbau.

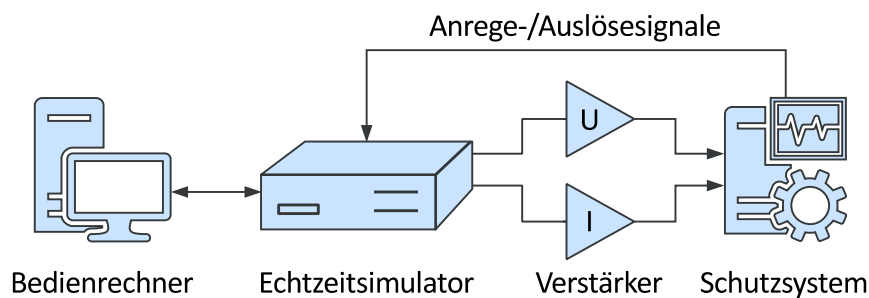


Abbildung 5.3: Aufbau zur Schutzprüfung

Zu Beginn der Prüfung werden anhand der charakterisierenden Parameter mittels eines MATLAB-Skripts die den Prüfpunkten und -schritten entsprechenden komplexen Spannungs- und Stromwerte berechnet. Die eigentliche Prüfablaufsteuerung erfolgt anschließend über die Ausführung eines Simulink-Modells. Während der Prüfung selbst wird der Spannungswert konstant gehalten und der Stromwert variiert. Vor jedem Prüfschritt wird durch Wahl eines geringen Stromwerts eine hohe, außerhalb des Polygons liegende Impedanz eingestellt. Diese Pausenimpedanz wurde hier zu +20 % gewählt. Zu Beginn jedes Prüfschritts wird schließlich der Stromwert, der der gewünschten Prüfimpedanz entspricht, eingestellt. Dieser Wert wird gehalten, bis entweder eine Anregung des zu prüfenden Systems detektiert wird oder eine Zeit von fünf Sekunden verstrichen ist. Die jeweils verstrichene Zeit wird anschließend aufgezeichnet und wieder eine Impedanz außerhalb des Polygons angesprungen, bevor mit dem nächsten Prüfschritt fortgefahren wird. Die Prüfung erfolgt mit dreiphasigen, symmetrischen Signalen, so dass das Verhalten des Distanzschutzes bei einem dreiphasigen, symmetrischen Fehlerereignis untersucht wird. Einphasige und zweiphasige Fehler sowie Fehler mit Erdberührung werden im Rahmen der hier durchgeführten Prüfungen nicht betrachtet.

5.1.3 Ergebnisse der Distanzschutzprüfung

Im Folgenden werden die Prüfergebnisse bei Ausführung des Distanzschutzes sowohl auf dem Co-Prozessor als auch auf dem Linux-Betriebssystem gezeigt und einander gegenübergestellt.

Abbildung 5.4 zeigt die Anreizeiten eines vollständigen Prüfdurchlaufs für das oben angegebene Distanzschutzpolygon bei Ausführung der Schutzfunktion auf dem Co-Prozessor.

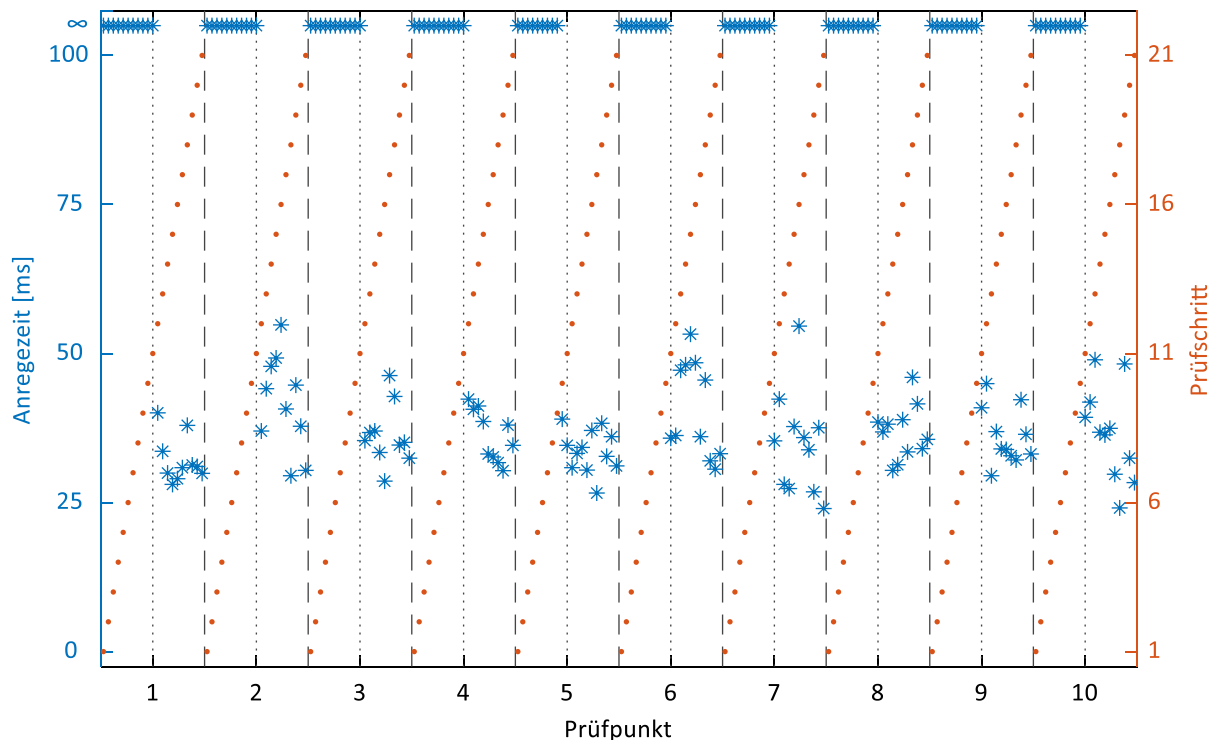


Abbildung 5.4: Anreizeiten des Distanzschutzes bei Ausführung auf dem Co-Prozessor

Das Distanzschutzpolygon wird an den in Abbildung 5.2 dargestellten 10 Punkten, entsprechend einem Impedanzwinkel von 0 bis 90° in 10°-Schritten, mit jeweils 21 Schritten geprüft. Die gestrichelten Linien in Abbildung 5.4 dienen zur optischen Abgrenzung der einzelnen Prüfpunkte, der jeweils mittlere Prüfschritt (Prüfschritt 11) ist als gepunktete Linie dargestellt. In der Abbildung ist zu sehen, dass – bis auf eine Ausnahme nahe der Polygongrenze (Prüfpunkt 5, Prüfschritt 10) – außerhalb des Polygons keine Anregung stattfindet, sondern frühestens bei einer Prüfimpedanz, die auf der Polygonlinie liegt (mittlerer Prüfschritt). Der eine Fall etwas zu früher Anregung ist auf Toleranzen der Prüfkette (Analogausgabe des Echtzeitsimulators und Verstärker) und der Messwerterfassung (Systemplattform) zurückzuführen. Die größte Anreizeit des Systems liegt bei ca. 55 ms, die kleinste bei ca. 25 ms.

Die Abbildungen 5.5 und 5.6 zeigen die Verläufe von Spannung, Strom, Anregung und Auslösung für einen beispielhaften Prüfschritt als Oszilloskop-Aufzeichnung. Während die Spannung konstant gehalten wird, wird zum Eintritt des Prüfereignisses in Form einer reduzierten Impedanz entsprechend der Strom erhöht. Mit einer Eigenzeit, die das System gemäß Abbildung 5.4 zur Detektion des Fehlerereignisses benötigt, erfolgt zunächst die Ausgabe des Anregesignals. Nach Ablauf der Auslöseverzögerung erfolgt schließlich die Ausgabe des Auslösesignals.

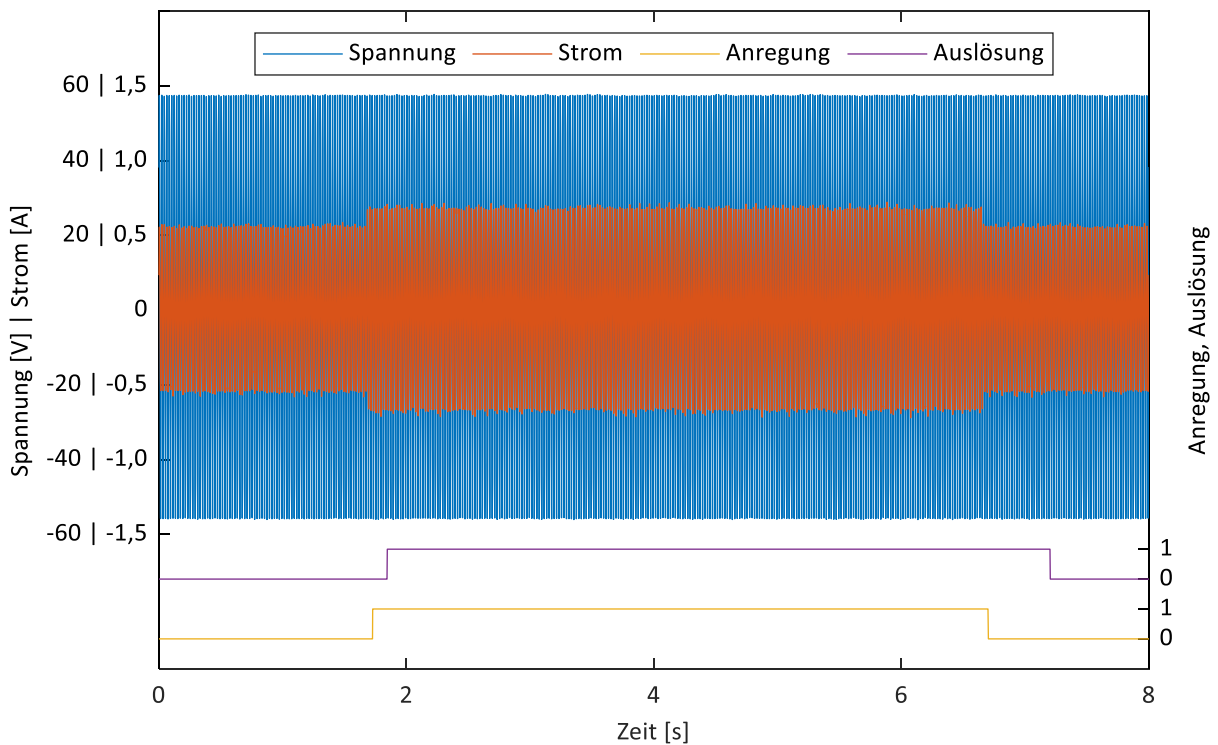


Abbildung 5.5: Beispielhafte Verläufe von Spannung, Strom, Anregung und Auslösung

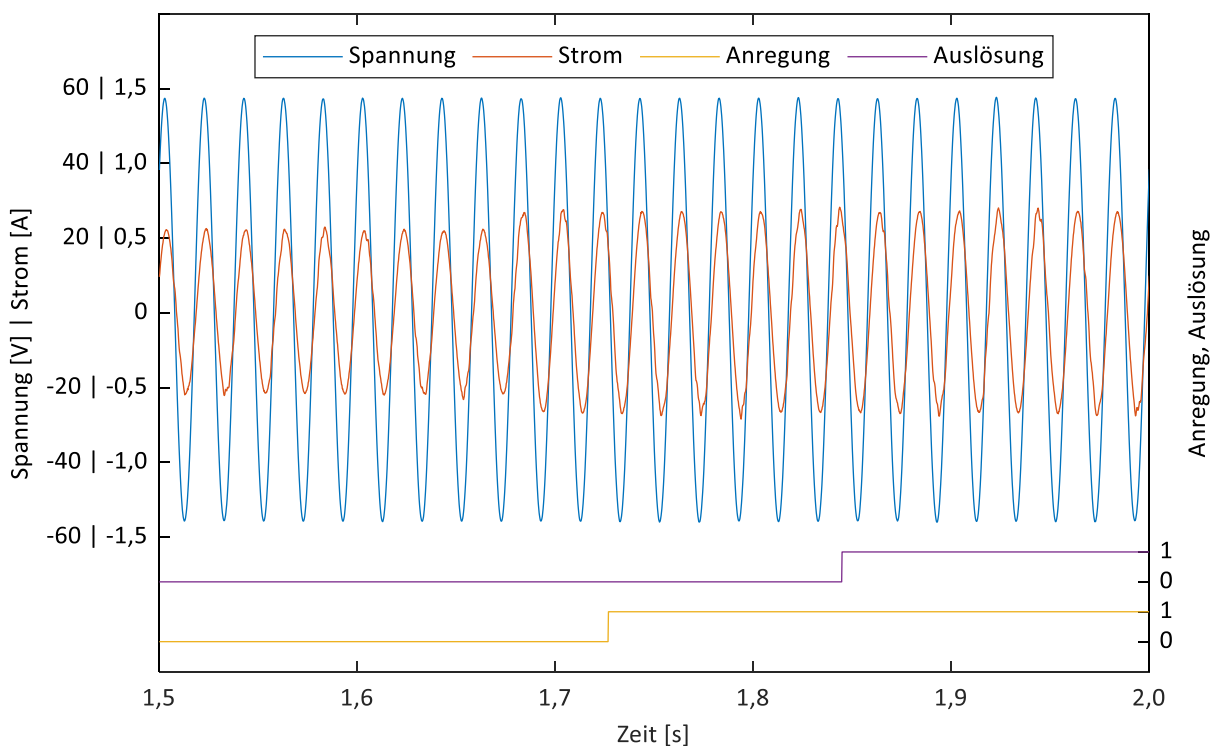


Abbildung 5.6: Detaildarstellung zu Abbildung 5.5

Als Referenz sind in den Abbildungen 5.7 und 5.8 die Anreizeiten eines vollständigen Prüfdurchlaufs bei Ausführung der Schutzfunktion auf dem Linux-Betriebssystem dargestellt. Abbildung 5.7 zeigt dabei die Anreizeiten, wenn innerhalb der

Anwendung nur die Distanzschutzfunktion ausgeführt wird, in Abbildung 5.8 wird zusätzlich kontinuierlich eine Netzzustandsschätzung parallel ausgeführt.

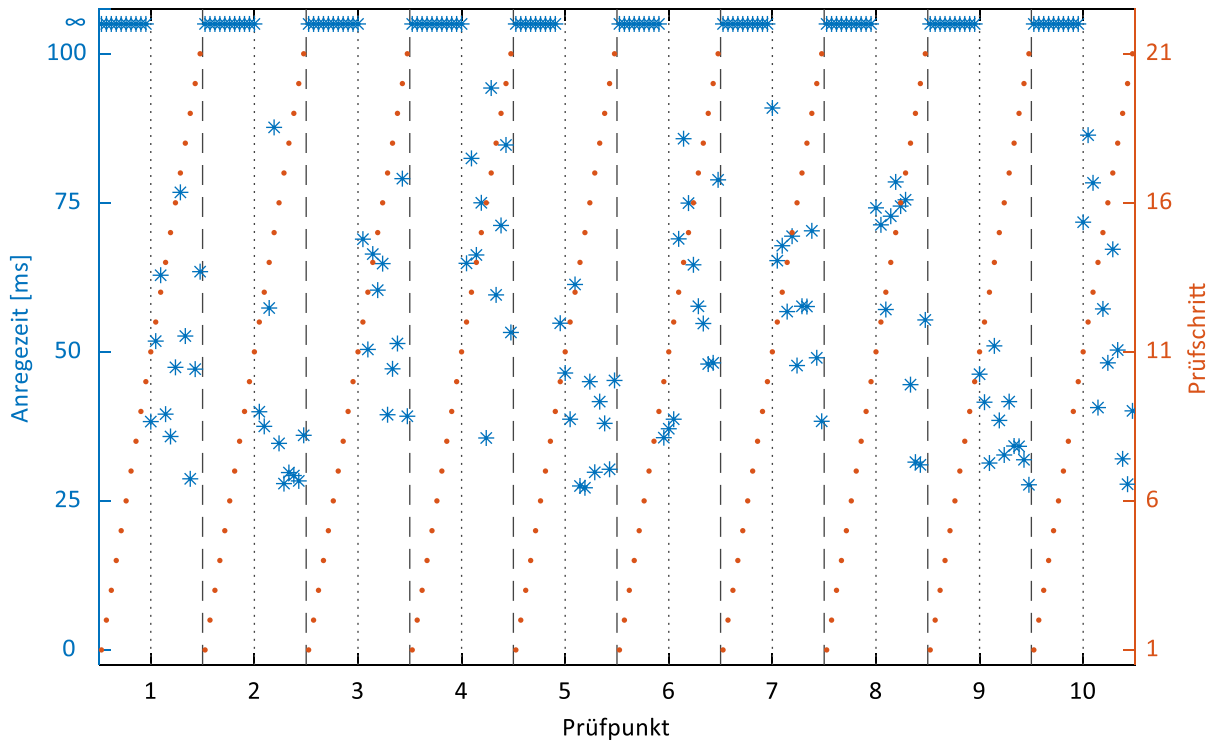


Abbildung 5.7: Anreizeiten des Distanzschutzes bei Ausführung auf dem Linux-Betriebssystem

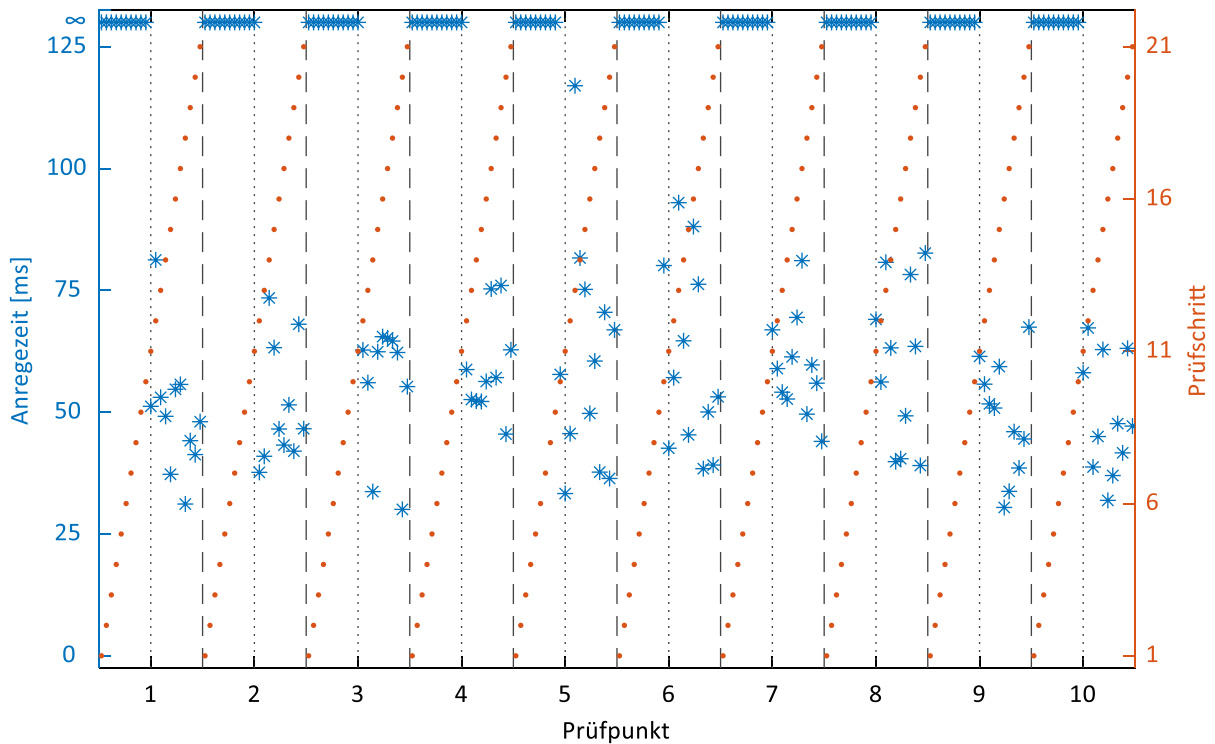


Abbildung 5.8: Anreizeiten des Distanzschutzes bei Ausführung auf dem Linux-Betriebssystem mit parallel ausgeführter Netzzustandsschätzung

In Abbildung 5.9 ist die Verteilung der Anregezeiten für alle drei Prüfzenarien dargestellt. Hierbei wurden für jedes Szenario jeweils drei vollständige Prüfdurchläufe ausgeführt.

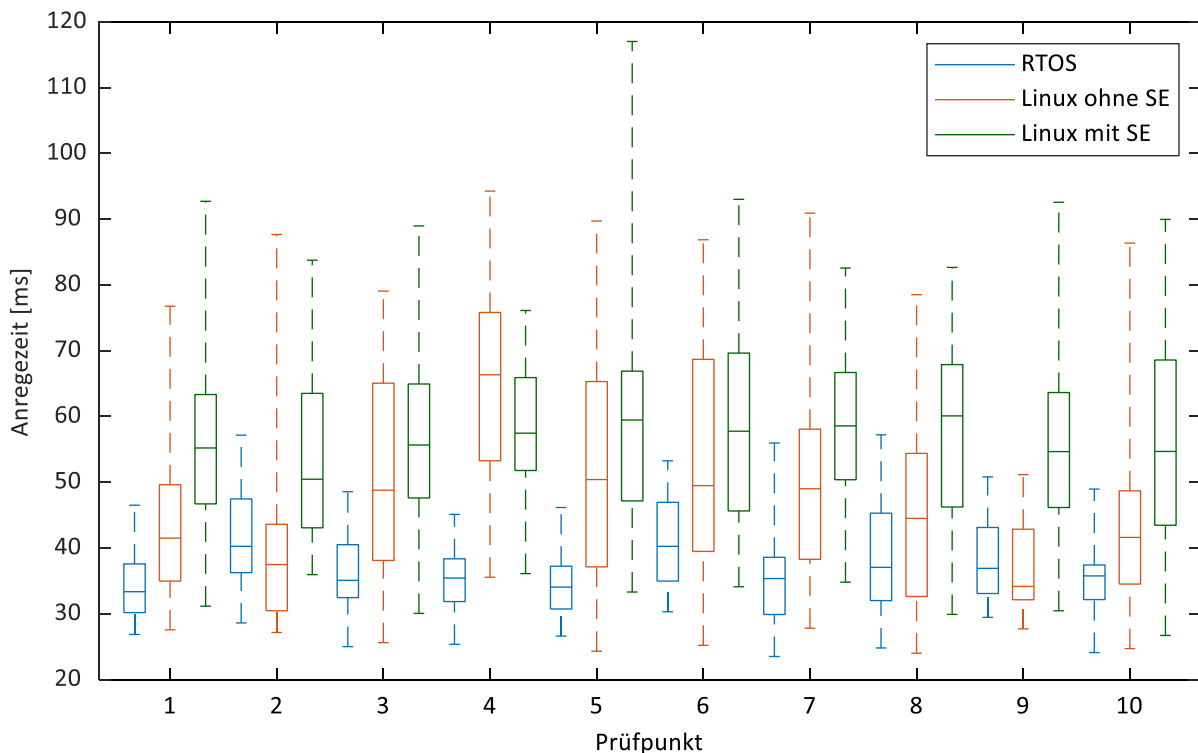


Abbildung 5.9: Verteilung der Anregezeiten für alle drei Prüfzenarien

Deutlich zu erkennen ist, dass der Distanzschutz bei Ausführung auf dem Echtzeitbetriebssystem die geringsten Anregezeiten aufweist. Bei Ausführung auf dem Linux-Betriebssystem sind die mittleren Anregezeiten höher, außerdem sind die Extrema größer. Insbesondere bei paralleler Ausführung einer rechenintensiven Funktion erhöhen sich die mittleren Anregezeiten weiter und vereinzelte große Ausreißer treten auf.

Während der Ausführung der Schutzfunktion auf dem Echtzeitbetriebssystem läuft gleichzeitig auch eine Funktion auf dem Linux-Betriebssystem ab. Diese dient dazu, der Schutzfunktion initial die erforderlichen Schutzparameter zu übermitteln. Nach dieser initialen Phase läuft die Schutzfunktion unabhängig von der Funktion auf dem Linux-Betriebssystem ab, so dass Laufzeitschwankungen letzterer keinen Einfluss auf die eigentliche Schutzfunktion haben. Daher sind Betrachtungen bezüglich der Anregezeiten der Schutzfunktion in Abhängigkeit von der Rechenbelastung auf dem Linux-Betriebssystem nicht erforderlich.

Insgesamt kann gesagt werden, dass der auf dem Echtzeitbetriebssystem implementierte Distanzschutz schnelles Zeitverhalten zeigt und auch die Messgenauigkeit zufriedenstellend ist. Zum Nachweis, dass der Algorithmus auch

herausfordernden Fehlerereignissen gewachsen ist, wie etwa unsymmetrischen Fehlern, Fehlern mit Erdberührung und Fehlern in Netzen mit verschiedenen Arten einer Sternpunktbehandlung, müssen weitergehende Prüfungen unter Nutzung entsprechender Simulationsmodelle genutzt werden. Auch das Verhalten unter realen Fehlersignalen, entweder aus aufgezeichneten Störschrieben oder durch Simulation ganzer Netze im Sinne einer anwendungsorientierten Schutzprüfung, ist zu untersuchen. Dies würde allerdings den Bereich dieser Arbeit sprengen, so dass diese sich auf die vorgestellten Prüfungen beschränkt.

5.2 Netzzustandsschätzung

Die Grundlagen der Netzzustandsschätzung wurden bereits in Kapitel 4.3.3 eingeführt. Im Nachfolgenden werden nun die vorbereitenden Arbeiten, die zur Umsetzung des Feldversuchs erforderlich waren, sowie die im Feldversuch selbst erlangten Ergebnisse vorgestellt. Teile der nachfolgenden Beschreibungen wurden bereits im Rahmen des Schlussberichts des Forschungsprojekts *i-Automate* [P1] veröffentlicht. Um das Gesamtverständnis, insbesondere auch im Zusammenhang mit dem durchgeführten Engineering- und Konfigurationsprozess, zu ermöglichen, wird an dieser Stelle dennoch an einer entsprechenden Beschreibung festgehalten.

Die Durchführung des Feldversuchs erfolgte in dem in Abbildung 5.10 dargestellten Netz:

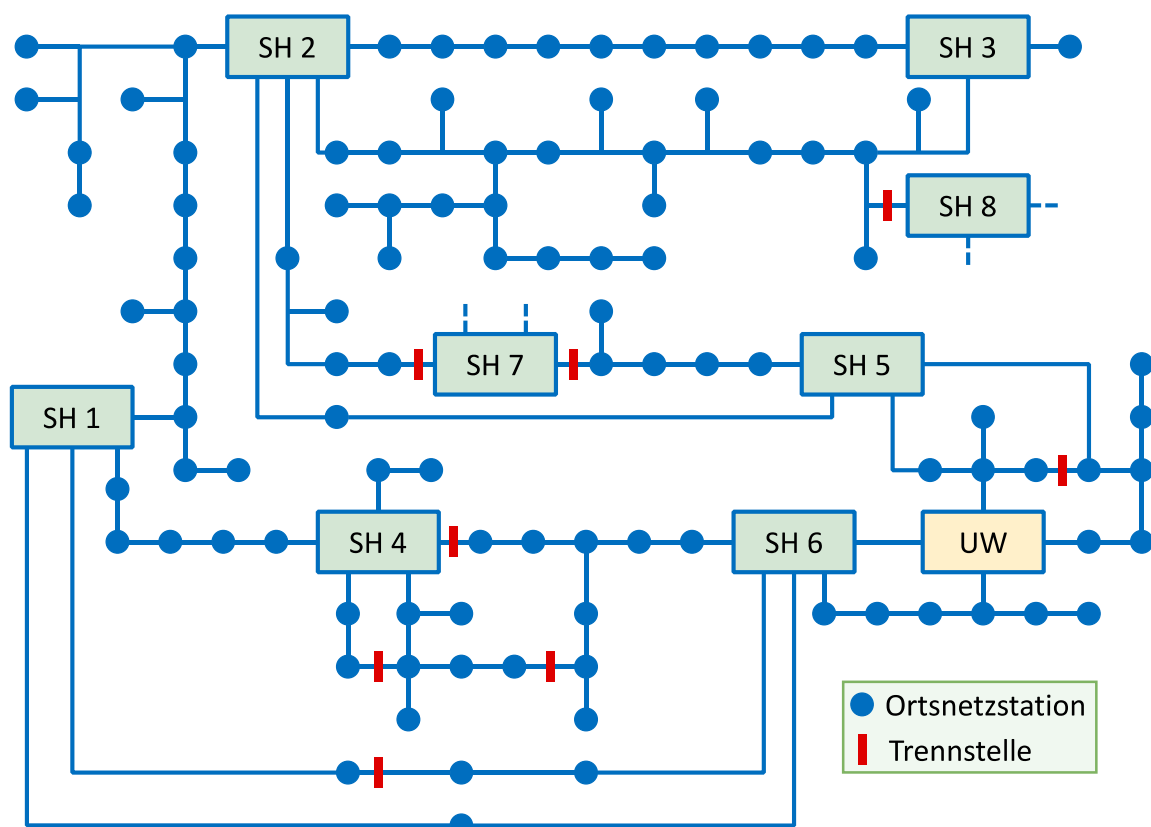


Abbildung 5.10: Gesamtnetz

Hierbei handelt es sich um ein ländlich geprägtes 20-kV-Mittelspannungsnetz, das über ein Umspannwerk (UW) aus dem überlagerten 110-kV-Netz gespeist wird. Das Netz umfasst rund 100 Ortsnetzstationen sowie acht Knotenstationen, die im Folgenden als Schalthäuser (SH) bezeichnet werden. Die Schalthäuser sind fernwirktechnisch an die Leitstelle des Netzbetreibers angebunden und ermöglichen eine Fernsteuerung der dort angeschlossenen Netzsegmente. Außerdem sind die einzelnen Netzabgänge zum Teil mit digitaler Schutztechnik ausgestattet. Weiterhin enthalten die Schalthäuser selbst je einen Ortsnetztransformator zur Speisung der unterlagerten Niederspannungsnetze. In der Abbildung sind die im Normalbetrieb geöffneten Trennstellen eingezeichnet. Die Schalthäuser *SH 7* und *SH 8* sind in diesem Fall nicht mit dem betrachteten Netz, sondern mit benachbarten Netzen verbunden. Daher können diese beiden Schalthäuser im Folgenden unberücksichtigt bleiben.

Die Realisierung eines bestimmten oder sogar überbestimmten Systems ist in Verteilnetzen aus heutiger Sicht kaum praktikabel, da hierzu eine engmaschige Ausstattung der Verteilnetze mit Messtechnik erforderlich wäre. Auch im Rahmen des Forschungsprojekts waren die Möglichkeiten zur Integration von Messtechnik begrenzt, so dass eine wie im folgenden Kapitel beschriebene Netzreduktion durchgeführt wurde.

5.2.1 Netzmodellaufbereitung und Erstellung einer Prüfumgebung

Vor der eigentlichen Erprobung im Feldversuch stand die Analyse und Simulation des Netzes im Laborumfeld. Insbesondere war sicherzustellen, dass die Netzreduktion nicht zu übermäßigen Abweichungen in den Netzzustandsberechnungen führte. Die Simulation des Netzes erfolgte innerhalb von *Matlab/Simulink*, die Ausführung von anwendungsorientierten Prüfungen in Form von Hardware-in-the-Loop-Simulationen erfolgte auf einem Echtzeitsimulator. Die Analyse- und Simulationsarbeiten erfolgten überwiegend im Rahmen studentischer Arbeiten.

Die Netzdaten des zuvor dargestellten Versuchsnetzes wurden durch den Netzbetreiber als Datenmodell im Format der Netzberechnungssoftware *DigSILENT PowerFactory* zur Verfügung gestellt. Mit der Exportfunktion dieser Software wurde das Netzmodell in das standardisierte CIM-Datenmodell (engl.: Common Information Model, IEC 61970-301 [134]) konvertiert, so dass es sich schließlich durch ein am Institut *ie³* für diesen Zweck entwickeltes Programm in ein Simulink-Modell überführen ließ. Die Übertragung des Netzmodells in ein Simulink-Modell ist notwendig, um eine Ausführung auf dem verwendeten Echtzeitsimulator zu ermöglichen. Die an diesem Simulator durchgeführten Echtzeitsimulationen basieren auf der Simulationssoftware *Simulink Simscape Electrical*. Die bereitgestellten Netze werden vollautomatisch konvertiert und stehen anschließend als lauffähige Netzmodelle zur Verfügung.

Vor der Konvertierung waren jedoch noch vorbereitende Arbeiten und Analysen notwendig. Dabei galt es zum einen, die Verteilung der dezentralen Erzeugungsanlagen innerhalb des Netzes aufzudecken, und zum anderen, die Schlechtpunkte des Netzes zu identifizieren. In Abbildung 5.11 ist dargestellt, an welchen Netzknoten welche Art von Erzeugungsanlagen angeschlossen ist. Außerdem ist das Spannungsprofil für einen mittleren Leistungsflusszustand visualisiert.

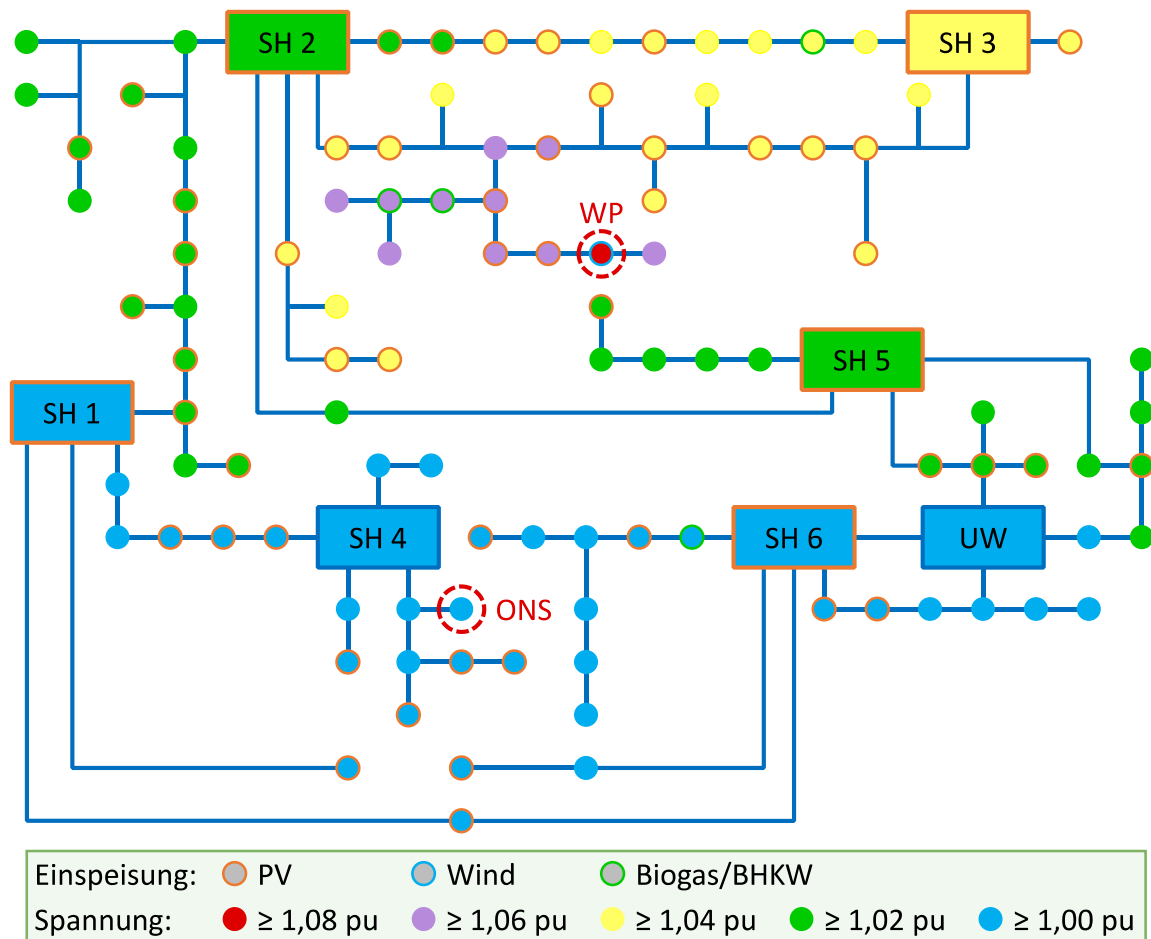


Abbildung 5.11: Angeschlossene dezentrale Erzeugungsanlagen und Spannungsprofil¹¹

Neben zahlreichen kleinen und mittleren Erzeugungsanlagen zeigt sich eine lokale Konzentration von dezentralen Erzeugungsanlagen an drei Netzknoten. Dabei handelt es sich um einen Windpark (WP), eine Biogasanlage sowie ein Blockheizkraftwerk. Zur Untersuchung der allgemeinen Leistungsflüsse im Netz wurden Starklast- und Rückspeiseszenarien simuliert und innerhalb des Netzes die Knoten mit Spannungsmaximum und Spannungsminimum identifiziert. Als Knoten mit Maximalspannung erwies sich der Windpark WP, als Knoten mit Minimalspannung die Ortsnetzstation ONS. Die Ergebnisse der Analyse sind in Tabelle 5.3 zusammengefasst.

¹¹ Für die Spannung gilt hier wie auch im weiteren Verlauf des Kapitels 5.2: $1 pu \stackrel{\text{def}}{=} 20 kV$

Tabelle 5.3: Maximal- und Minimalspannungen im betrachteten Netz

| Szenario | Windpark | Ortsnetzstation |
|-------------------------------|----------|-----------------|
| Volllast - Volleinspeisung | 1,07 pu | 1,01 pu |
| Schwachlast - Volleinspeisung | 1,09 pu | 1,03 pu |
| Volllast - Schwacheinspeisung | 1,00 pu | 0,99 pu |

Obwohl Extremszenarien betrachtet wurden, muss festgehalten werden, dass es keine unzulässigen Spannungsabweichungen oder Überlastungen innerhalb des Netzgebiets gibt. In Rücksprache mit dem Netzbetreiber wurde festgestellt, dass im Netzgebiet auch zukünftig keine unzulässigen Grenzwertverletzungen erwartet werden. Aufgrund der übrigen, günstigen Rahmenbedingungen wurde dennoch an dem Netzgebiet für den Feldversuch festgehalten.

Für die Positionierung der Messwerterfassung sind aus strategischer und auch rein praktischer Sicht insbesondere die Schalthäuser geeignet, da hier einerseits jeweils mehrere Netzsegmente aufeinandertreffen und andererseits eine bauliche Integration am einfachsten bzw. überhaupt nur möglich ist. Daher wurden die Schalthäuser auch im Rahmen des Feldversuchs als Messorte ausgewählt. Bei der hohen Anzahl an Ortsnetzstationen und einer begrenzten Anzahl an Messorten ist eine vollständige Beobachtbarkeit des Netzes nicht gegeben. Zur Durchführung der Netzzustandschätzung ist daher eine Reduktion der Netzkomplexität erforderlich. Dadurch geht zwar die vollständige Beobachtbarkeit verloren, jedoch bleibt zumindest eine Aussage über den mittleren Netzzustand möglich.

Auf Basis einer spezifischen, vollständigen Leistungsflussrechnung innerhalb eines Leistungsflussberechnungsprogramms wird die Netzreduktion selbst durch die Einführung von Ersatzknoten (EK) realisiert. Dabei werden die Ersatzknoten derart positioniert und leistungsmäßig skaliert, dass sich die Leistungsflüsse an den Messknoten geringstmöglich ändern. Dadurch wird erreicht, dass der Netzzustand des Ersatzknotens einen *mittleren* Netzzustand repräsentiert. Wie bereits erwähnt, geht dabei die vollständige Beobachtbarkeit verloren und Engpässe innerhalb eines so reduzierten Netzbereichs werden möglicherweise nicht entdeckt. Wie in Abbildung 5.12 zu sehen, können für die Platzierung eines Ersatzknotens sowohl die Ersatzknotenleistung \underline{S}_E als auch die Ersatzknotenposition mittels der Leitungslängen l_{1E} und l_{2E} variiert werden, um die Leistungsflüsse des Ursprungsnetzes möglichst gut zu treffen.

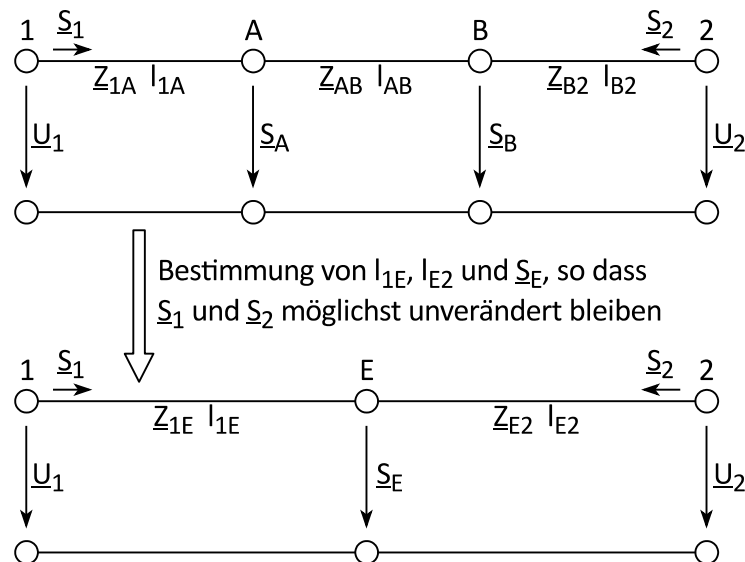


Abbildung 5.12: Netzreduktion: Konzentration der Leistungen A und B zur Ersatzleistung E

Für die Parameter der Leitung zum Ersatzknoten werden die Leitungsparameter des dominanten Leitungstyps im Ursprungsnetz verwendet. Mit Hilfe der beiden Freiheitsgrade *Leistung* und *Ort* lassen sich die Leistungsflüsse an den Messknoten beeinflussen. Durch eine Optimierung kann diejenige Position und Leistung des Ersatzknotens ermittelt werden, bei der sich die geringsten Abweichungen vom ursprünglichen Leistungsfluss ergeben. Im Rahmen der Arbeiten wurde lediglich eine iterative Berechnung durchgeführt und die Variante mit dem geringsten Fehler als Netztopologie für die Netzzustandsschätzung ausgewählt. In [135, 136] wird ein hierüber hinausgehendes Verfahren zur optimierten Netzreduktion in unterbestimmten Verteilnetzen vorgestellt. Weicht weiterhin der tatsächliche Lastfluss stark von demjenigen ab, der bei der Netzreduktion zugrunde gelegt wurde, so spiegeln die Ergebnisse der Netzzustandsschätzung nicht mehr den mittleren Netzzustand wider. Auch hier besteht Optimierungspotenzial, etwa durch Nutzung einer adaptiven Auswahl einer reduzierten Netztopologie auf Basis der gemessenen Leistungsflüsse oder in der Anwendung zusätzlicher Verfahren wie Netzpartitionierung.

Für das Versuchsnetz wurde die Netzreduktion nach dem iterativen Verfahren vollständig durchgeführt. Dafür wurden zunächst die Leistungszentren auf den Strängen bestimmt und die festgelegten Messstellen berücksichtigt. Das Ergebnis zeigt Abbildung 5.13.

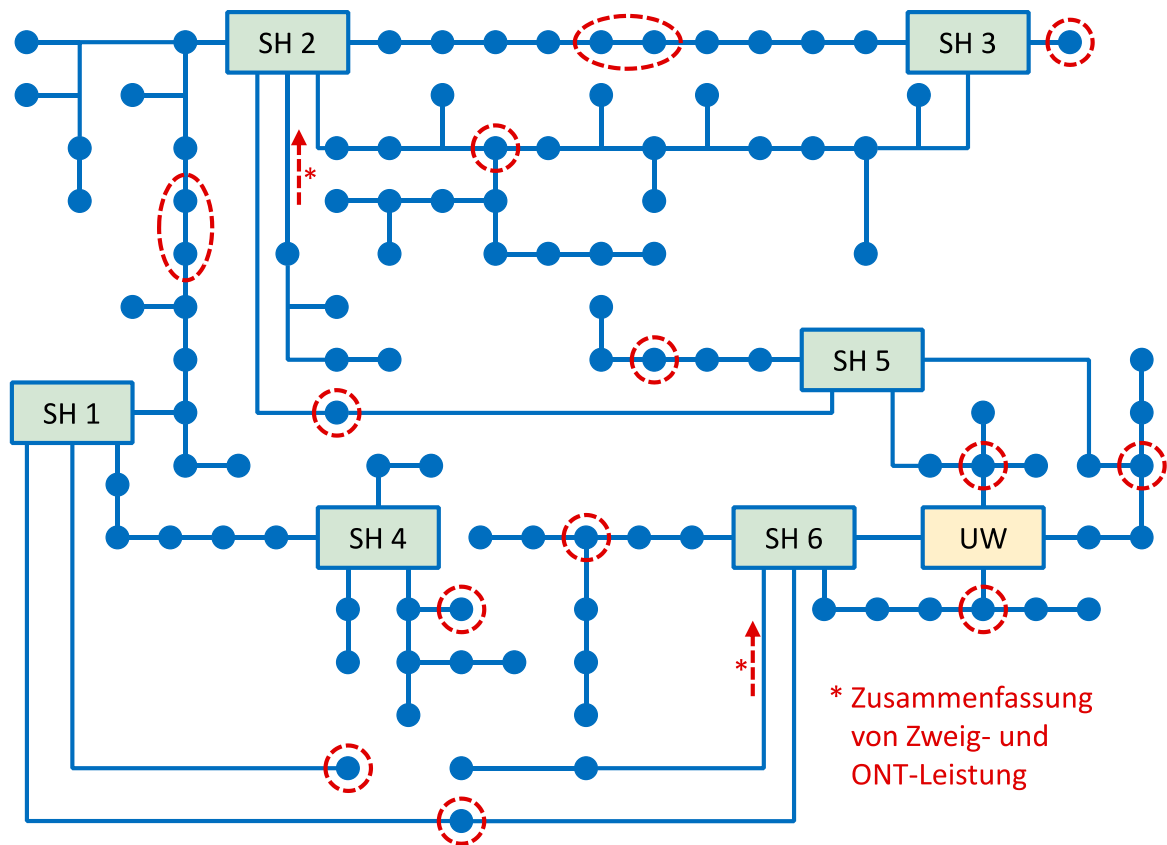


Abbildung 5.13: Gesamtnetz, Definition von Konzentrationsknoten

Für das Versuchsnetz ergibt sich eine entsprechende Vereinfachung des Netzes. Als finale Topologie für die Netzzustandsschätzung verbleibt das Netz aus Abbildung 5.14.

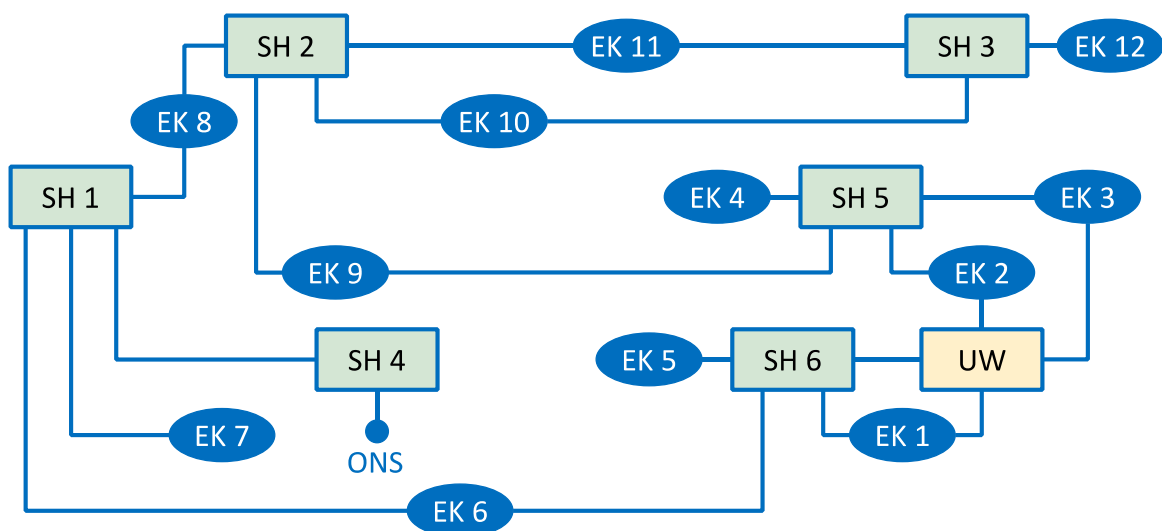


Abbildung 5.14: Reduziertes Netz mit Ersatzknoten

Neben dem Umspannwerk und den sechs Schalthäusern wurde auch die Ortsnetzstation *ONS*, die sich als Extrempunkt mit Minimalspannung herausgestellt hatte, für die Integration von Messtechnik ausgewählt. Der Anschlusspunkt des Windparks, der als Extrempunkt mit Maximalspannung ebenfalls für eine Messung prädestiniert gewesen wäre, ließ allerdings keine Integration von Messtechnik zu, da

es sich um eine Kundenstation handelt und keine entsprechende Vereinbarung mit dem Betreiber erzielt werden konnte. Daher ging dieser Anschlusspunkt mit in den Ersatzknoten EK 10 auf. Neben den gemessenen Stationen verbleiben schließlich zwölf Knoten als Ersatzknoten. Mit den geplanten Messpunkten liegt schließlich für das reduzierte Netz ein genau bestimmtes System vor, so dass sich mittels einer Netzzustandsschätzung der Netzzustand bestimmen lässt.

In Abbildung 5.15 sind die Ergebnisse einer Leistungsflussrechnung in DigSILENT PowerFactory in Form der Beträge und Winkel der Knotenspannungen für das originale und das reduzierte Netzmodell gegenübergestellt. Hierbei zeigen sich Abweichungen in den Spannungsbeträgen von bis zu 1 % und in den Spannungswinkeln von bis zu 0,2 %. Diese sind aufgrund der Netzreduktion nicht zu vermeiden und auf das iterative, teils manuelle Vorgehen zurückzuführen, das nicht zwingend zu optimalen Ergebnissen führt. Dies hat insbesondere Auswirkungen auf die Schätzgüte der Netzzustandsschätzung. Zukünftig sollten daher für die Netzreduktion Optimierungsverfahren eingesetzt werden, um die Ergebnisse zu verbessern.

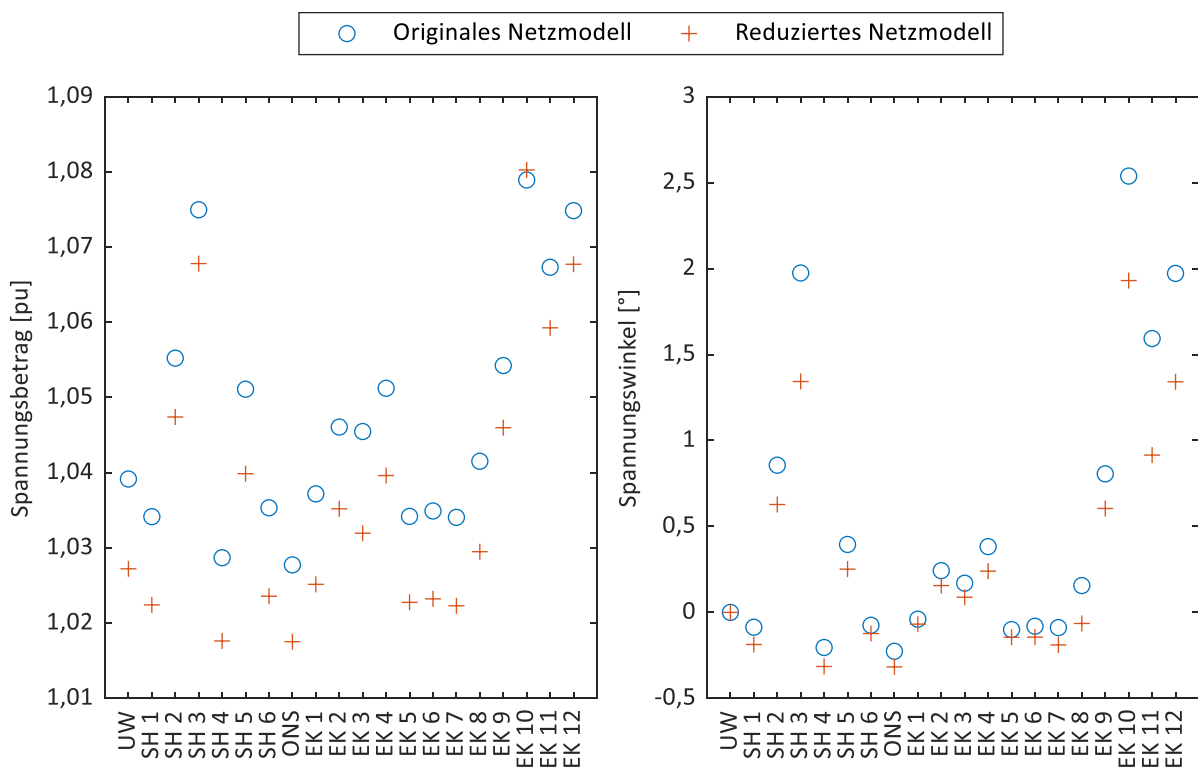


Abbildung 5.15: Vergleich zwischen originalem und reduziertem Netzmodell

Im Anschluss wurde das reduzierte Netzmodell aus DigSILENT PowerFactory in das CIM-Datenmodell exportiert und in ein Simulink-Modell überführt. Es wurden geringfügige Anpassungen umgesetzt, um das Netzmodell für die Ausführung auf dem Echtzeitsimulator vorzubereiten. Hierzu zählen vor allem die Definition von Ausgangsgrößen und die Partitionierung des Netzmodells für eine parallelisierte Simulationsexecution. In Abbildung 5.16 sind die Beträge und Winkel der Knotenspannungen aus

der Leistungsflussrechnung in DigSILENT PowerFactory für das reduzierte Netzmodell denen aus einer Netzsimulation auf dem Echtzeitsimulator gegenübergestellt. Vereinzelt Abweichungen sind äußerst gering und liegen in etwaigen Rundungen bei der Datenübertragung und unterschiedlichen numerischen Verfahren der zugrunde liegenden Berechnungen begründet.

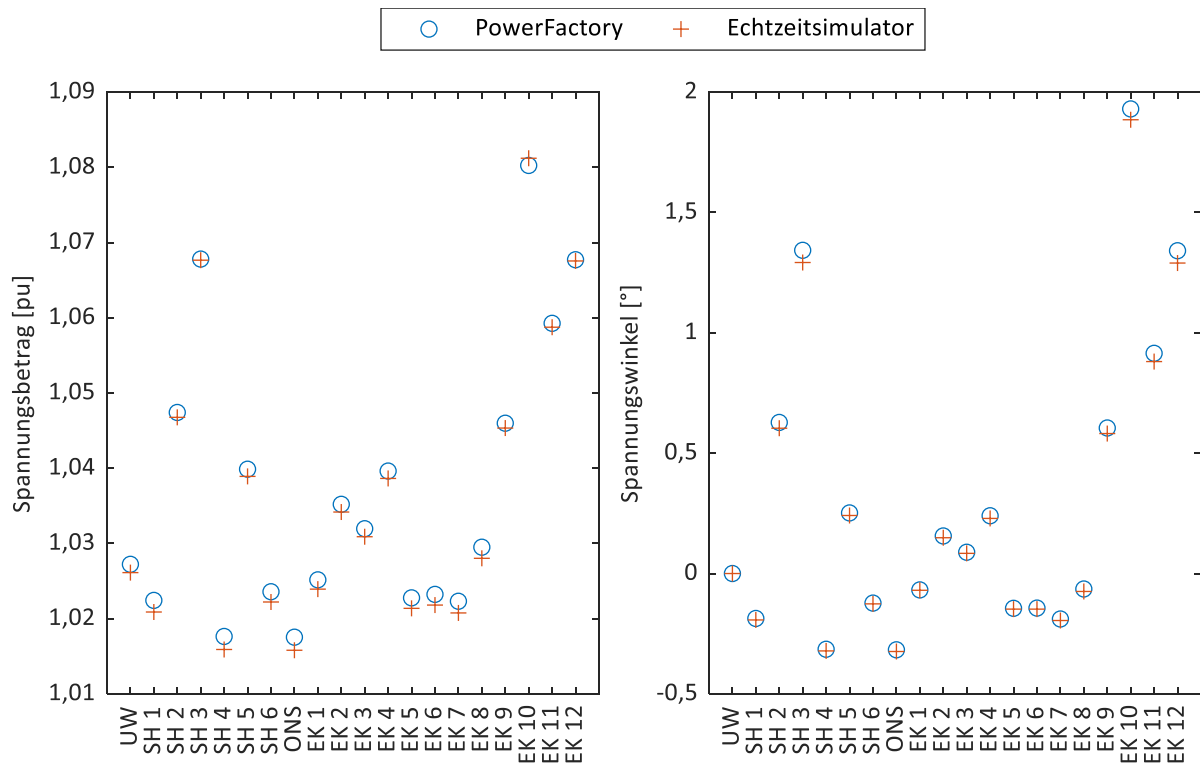


Abbildung 5.16: Vergleich zwischen DigSILENT PowerFactory und Echtzeitsimulator

5.2.2 Verifikation des Algorithmus der Netzzustandsschätzung

Im letzten Schritt galt es, den auf der Systemplattform ausgeführten Algorithmus der Netzzustandsschätzung im Rahmen einer Hardware-in-the-Loop-Prüfung zu verifizieren. Hierzu wurde das Zentralgerät über ein Ethernetnetzwerk mit dem Echtzeitsimulator verbunden. Auf dem Zentralgerät wurde lediglich die Funktion der Netzzustandsschätzung ausgeführt, die Messwerte aller Geräte wurden hingegen durch den Echtzeitsimulator vorgegeben. Das in Simulink vorliegende und bereits zuvor genutzte Modell des reduzierten Netzes wurde um Blöcke zur Messwerterfassung und zur Netzwerkkommunikation erweitert. Im Gegensatz zum finalen Feldaufbau, bei dem eine Kommunikation per IEC 61850-8-1 MMS erfolgte, wurde hier die selbst implementierte Kommunikation per UDP genutzt, da der Echtzeitsimulator keine Unterstützung für IEC 61850-8-1 MMS bot.

Bei den vom Echtzeitsimulator an das Zentralgerät übermittelten Messwerten handelte es sich um exakt diejenigen, die auch im finalen Feldansatz an die

Netzzustandsschätzung übermittelt werden, im Detail die Knotenspannungsbeträge sowie die Zweigleistungen aller gemessenen Stationen. Hieraus bestimmte der Algorithmus die komplexen Knotenspannungen aller Knoten, so auch die der Ersatzknoten. Abbildung 5.17 zeigt die Beträge und Winkel der Knotenspannungen aus der Netzsimulation auf dem Echtzeitsimulator im Vergleich zu denen aus der Netzzustandsschätzung. Es zeigen sich nur sehr geringe Abweichungen. Die Genauigkeit der Netzzustandsschätzung für das gegebene Netz ist demnach sehr hoch.

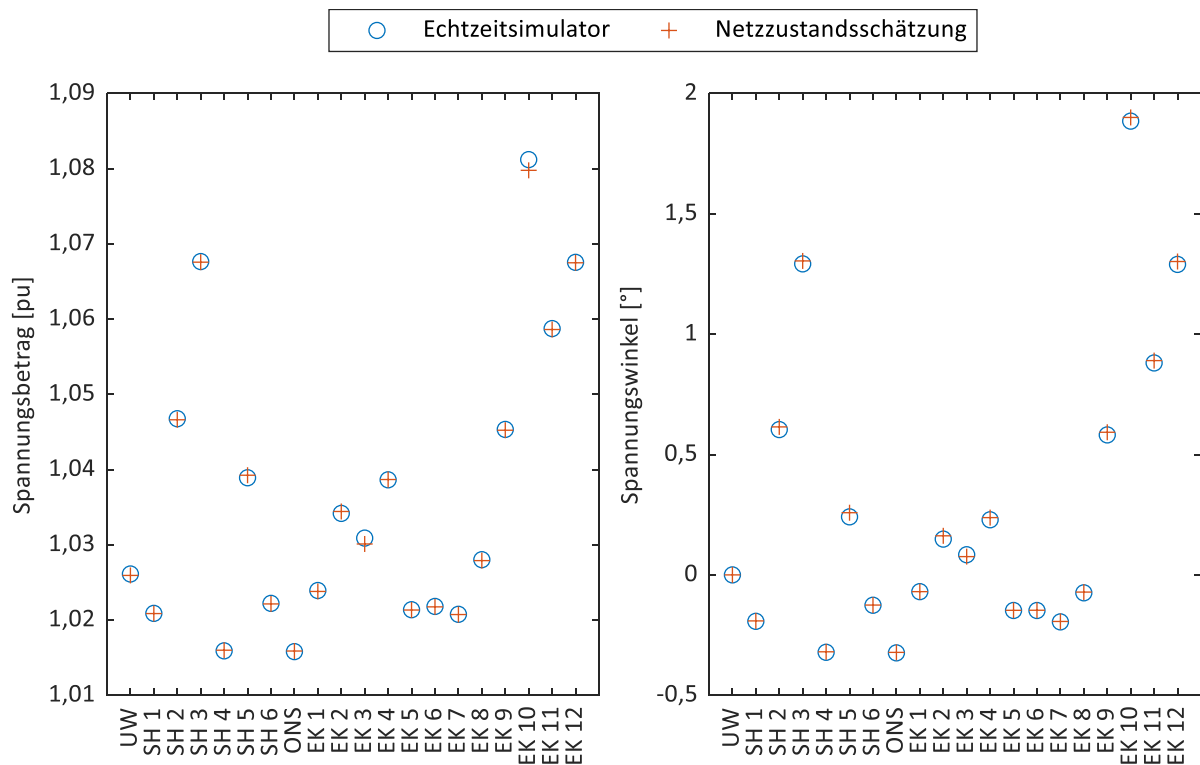


Abbildung 5.17: Vergleich zwischen Echtzeitsimulator und Netzzustandsschätzung

5.2.3 Darstellung der Gesamtsystemstruktur

Unter Berücksichtigung technischer Randbedingungen, einer begrenzten Anzahl zur Verfügung stehender Messgeräte sowie der Ergebnisse der Netzreduktion wurden schließlich acht Netzknoten zur Ausstattung mit Messtechnik ausgewählt. Neben den bereits erwähnten sechs Schalthäusern und der Ortsnetzstation war dies insbesondere auch das Umspannwerk selbst. Da sowohl das Umspannwerk als auch die Schalthäuser *SH 1*, *SH 2*, *SH 3*, *SH 5* und *SH 6* mittelspannungsseitig mit Schutzgeräten ausgestattet sind, sind in den einzelnen Abgängen bzw. an der Sammelschiene Strom- und Spannungswandler vorhanden. Dies ermöglichte eine einfache Einschleifung der Messgeräte in die Wandlerstromkreise. Im Schalthaus *SH 4* sind dagegen keine Schutzgeräte und damit auch keine Messwandler vorhanden. Daher erfolgte hier lediglich eine Spannungsmessung auf Niederspannungsseite, wobei eine Umrechnung auf die mittelspannungsseitigen Größen anhand des Transformatorverhältnisses

vorgenommen wurde. In der Ortsnetzstation *ONS* wurden ebenfalls niederspannungsseitig die Spannung sowie der Transformatorstrom erfasst und entsprechend auf die Mittelspannungsgrößen umgerechnet.

Im Umspannwerk kam ein Messgerät vom Typ SHERLOG CRX zum Einsatz, wodurch es möglich war, sämtliche vier zum Netzausschnitt zugehörigen Abgänge dreiphasig zu messen. Auf diesem Gerät wurde außerdem die Berechnung der Netzzustandschätzung ausgeführt. In den Schalthäusern und der Ortsnetzstation kamen dagegen aus Kostengründen Messgeräte von Typ EPPE CX zum Einsatz. Als problematisch erwies es sich, dass diese Geräte nur über vier Strommesskanäle verfügen. Bei einer dreiphasigen Messung hätte somit nur ein Abgang pro Station erfasst werden können, was ungenügend gewesen wäre. Letztendlich wurde daher entschieden, einphasige Messungen durchzuführen, um mit einem Messgerät die Ströme von bis zu vier Abgängen erfassen zu können. Damit musste jedoch von der Annahme eines nahezu symmetrischen Mittelspannungsnetzes ausgegangen werden, um so vom Messwert einer Phase auf die dreiphasigen Messwerte schließen zu können. Im Rahmen des Feldtests konnte die Gültigkeit dieser Annahme mangels tiefergehender Messmöglichkeiten jedoch nicht weiter überprüft werden.

Für die Datenkommunikation war ein DSL-Anschluss mit entsprechendem Router im Umspannwerk vorgesehen, in allen anderen Stationen kamen Mobilfunkrouter zum Einsatz. Die Kommunikation wurde durch Aufbau eines VPNs (engl.: Virtual Private Network, Virtuelles privates Netzwerk) entsprechend gesichert. Zur Sicherung kam IPsec (engl.: Internet Protocol Security, Sammlung mehrerer Standards zur gesicherten Kommunikation in IP-Netzwerken, [137]) mit X.509-Zertifikaten [138] zum Einsatz, der DSL-Router diente als Zertifikate-Server. Abbildung 5.18 zeigt die kommunikationstechnische Struktur. Neben den insgesamt acht Stationen (Umspannwerk, Schalthaus 1 bis 6 und eine Ortsnetzstation) ist in der Abbildung ein Fernwartungszugang (*Service-PC*) eingezeichnet. Dieser dient zum Übertragen von Softwareaktualisierungen und zum Auslesen aufgezeichneter Daten. Die Verbindung erfolgt hierbei von einem PC-Arbeitsplatz aus über das Internet per VPN-Software. Darüber hinaus war ursprünglich die Umsetzung einer Leitstellenfunktionalität auf Seiten des Netzbetreibers vorgesehen. Dieses Vorhaben wurde jedoch letztendlich nicht realisiert.

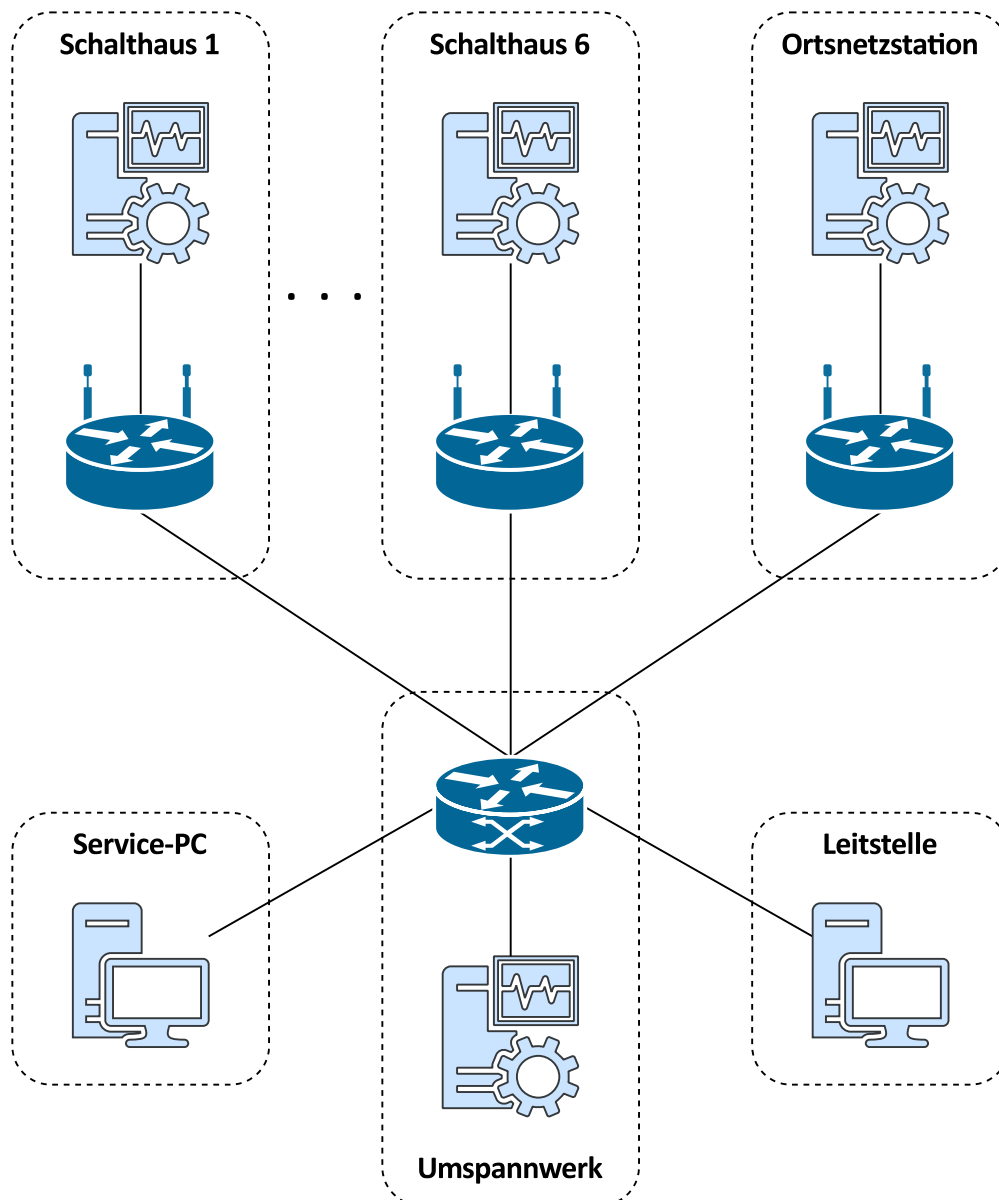


Abbildung 5.18: Kommunikationstechnische Struktur für den Feldtest

5.2.4 Engineering und Konfiguration des Gesamtsystems

Aus den vorangegangenen Beschreibungen ergeben sich direkt die funktionalen und kommunikationstechnischen Anforderungen an das Gesamtsystem und damit auch die Engineeringaufgabe, deren Durchführung im Folgenden beschrieben wird. Die Kommunikation zwischen verteilten Messpunkten und zentralem Gerät erfolgte gemäß IEC 61850-8-1 (MMS). Eine lokale SQLite-Datenbank auf dem zentralen Gerät diente zur Archivierung der Messwerte und der Berechnungsergebnisse der Netz-zustandsschätzung. Das Engineering erfolgt gemäß dem in Kapitel 3.4 beschriebenen Top-Down-Engineeringprozess und nutzt dabei die in Kapitel 4.5.1 beschriebenen Modellierungselemente. Im Folgenden werden die einzelnen Schritte des Erstellungsprozesses erläutert. Hierbei zeigen die Abbildungen 5.19 bis 5.26 die jeweiligen

Benutzereingaben des SCT, das an dieser Stelle als Softwarewerkzeug zur Systemspezifikation und Systemkonfiguration genutzt wird.

Als Erstes erfolgt die Systemspezifikation. Im Grundgedanken der Norm dient die SCL zur Beschreibung einzelner Unterstationen mit einer beliebigen Anzahl an Schaltfeldern und IEDs. Die hier vorliegende Engineeringaufgabe hingegen umfasst mehrere räumlich verteilte Netzknoten, die jeweils mit einem Messgerät ausgestattet sind. Um die Mittel der SCL ohne Anpassungen anwenden zu können, wurde das gesamte Netzgebiet des Feldtests als eine Unterstation mit einer Spannungsebene betrachtet und jeder messtechnisch ausgestattete Netzknoten durch ein Schaltfeld modelliert. Abbildung 5.19 zeigt das entsprechende Substation-Datenmodell.

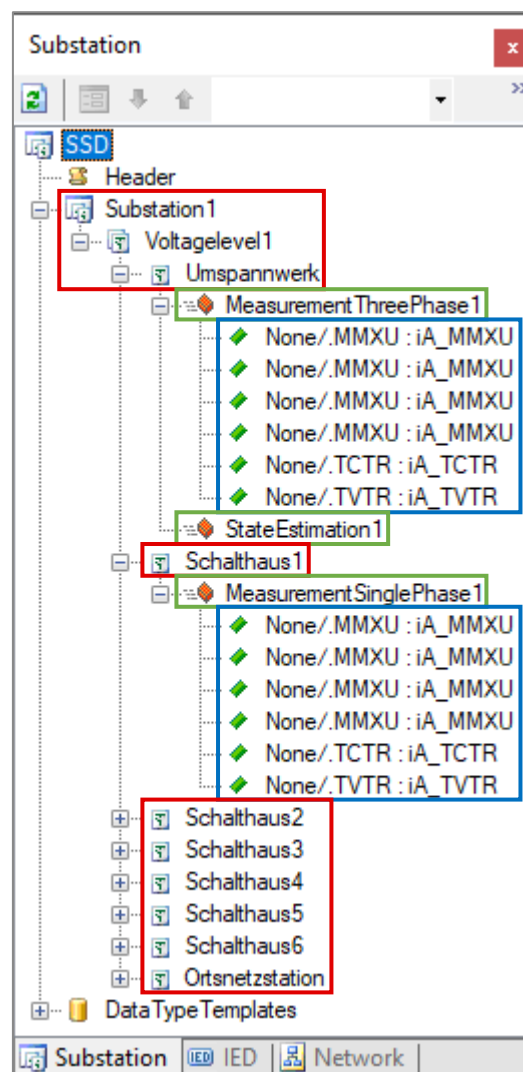


Abbildung 5.19: Datenmodell der Unterstation mit Spannungsebene und Schaltfeldern (rot), Funktionen (grün) und *abstrakten* logischen Knoten (blau)

Innerhalb der einen Unterstation und Spannungsebene wird jeder Netzknoten als ein Schaltfeld repräsentiert. Die einem Netzknoten zugeordnete Funktionalität wird dabei durch untergeordnete *Function*-Elemente modelliert. Um die Verknüpfung zwischen

normativen *Function*-Elementen und den spezifisch auf der Systemplattform implementierten Smart-Grid-Funktionen zu ermöglichen, wird das Attribut *type* eines *Function*-Elements genutzt. Der Wert dieses Attributs kann anwenderspezifisch festgelegt werden und muss in der hier vorliegenden Implementierung auf das entsprechende Funktionsschlüsselwort gemäß Tabelle 4.1 (Spalte 1) gesetzt werden. In diesem Anwendungsbeispiel werden die Funktionen *MeasurementSinglePhase* und *MeasurementThreePhase* zur ein- bzw. dreiphasigen Messung und *StateEstimation* zur Netzzustandsschätzung genutzt. Das Attribut *name* kann frei belegt werden, um beispielsweise mehrere Instanzen eines Funktionstyps zu benennen, wird aber nicht weiter ausgewertet. Abbildung 5.20 zeigt die Eigenschaften eines *Function*-Elements.

| Properties | |
|---------------------|--------------------------|
| tFunction | |
| SCL 1.4 | |
| Any | |
| AnyAttr | |
| desc | |
| GeneralEquipment | |
| LNode | tLNode[] Array |
| name | Measurement ThreePhase 1 |
| Private | |
| SubFunction | |
| Text | |
| SCL 3.0 | |
| ConductingEquipment | |
| type | Measurement ThreePhase |
| SCL Annex C | |
| SCL 1.4 | |

Abbildung 5.20: Eigenschaften eines Function-Elements

Darüber hinaus benötigen Funktionen i. d. R. logische Knoten, um z. B. Messwerte und Parameter einlesen oder berechnete Werte zurückschreiben zu können. Dies wird durch abstrakte logische Knoten, die den Funktionen untergeordnet sind, modelliert (Abbildung 5.19). Die einphasige Messung benötigt etwa je einen logischen Knoten der Klasse *TVTR* und *TCTR*, aus der die Wandlerverhältnisse des Spannungs- bzw. Stromwandlers eingelesen werden. Weiterhin werden vier logische Knoten der Klasse *MMXU* benötigt, in welche die abgangsbezogenen Messwerte geschrieben werden. Bei der Funktion der Netzzustandsschätzung erfolgt die Verknüpfung mit den logischen Knoten bisher noch nicht über die SCL, sondern über separate Konfigurationsdateien. Denn neben den logischen Knoten selbst sind weitere Informationen erforderlich, etwa über die Zuordnung zwischen primärtechnischem Netzknoten, an dem eine Messung erfolgt, und logischem Knoten, der diese Messung sekundärtechnisch repräsentiert. Eine Übersicht über diese Konfigurationsdateien wird später in Tabelle 5.4 gegeben.

An dieser Stelle ist der erste Engineeringsschritt, die Spezifikation des Anlagendatenmodells (vgl. Abbildung 3.9), abgeschlossen.

Im zweiten Engineeringsschritt werden nun die IED-Datenmodelle spezifiziert. Gemäß der in Kapitel 4.5.1 aufgestellten Konvention wird jedem Schaltfeld (hier: Netzknoten) genau ein IED (hier: Messgerät) zugeordnet, wobei Schaltfeld und IED denselben Namen erhalten. Abbildung 5.21 zeigt das Datenmodell eines Mess-IEDs.

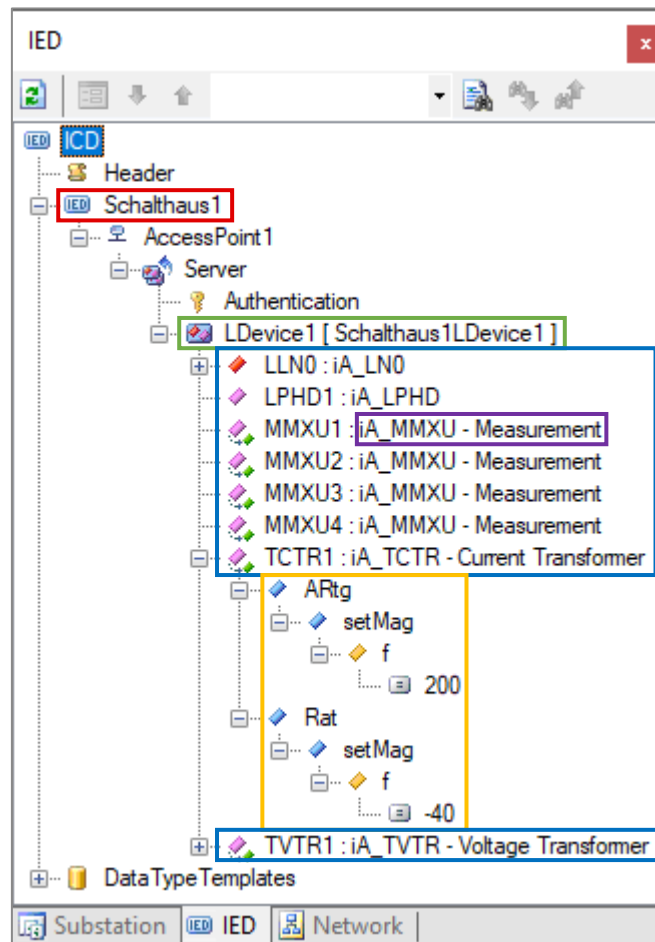


Abbildung 5.21: Datenmodell eines Mess-IEDs (rot) mit logischem Gerät (grün), logischen Knoten (blau), LN-Typ (violett) und Instanzen von Datenobjekten und Datenattributen (gelb)

Innerhalb des IEDs wird ein logisches Gerät erstellt, in welchem wiederum die erforderlichen logischen Knoten platziert werden. Dabei müssen mindestens Instanzen für diejenigen logischen Knoten, die als abstrakte logische Knoten im Substation-Datenmodell spezifiziert wurden, vorgesehen werden. Jeder logische Knoten wird mit einem in den *DataTypeTemplates* definierten Typ typisiert. Auch Standardwerte für Datenattribute können hier definiert werden, indem Instanzen von Datenobjekten und Datenattributen, die den logischen Knoten untergeordnet sind, erzeugt werden. In diesem Beispiel sind dies der Nennstrom des Stromwandlers (ARTg, 200 A) sowie das Wandlerverhältnis (Rat, 200 A zu 5 A, d. h. ein Wandlerverhältnis von 40, das negative Vorzeichen dient dazu, die Leistungsflussrichtung softwareseitig zu invertieren).

Neben den verteilten Messgeräten existiert ein zentrales Gerät, auf dem die Netzzustandsschätzung abläuft. Abbildung 5.22 zeigt das Datenmodell dieses IEDs.

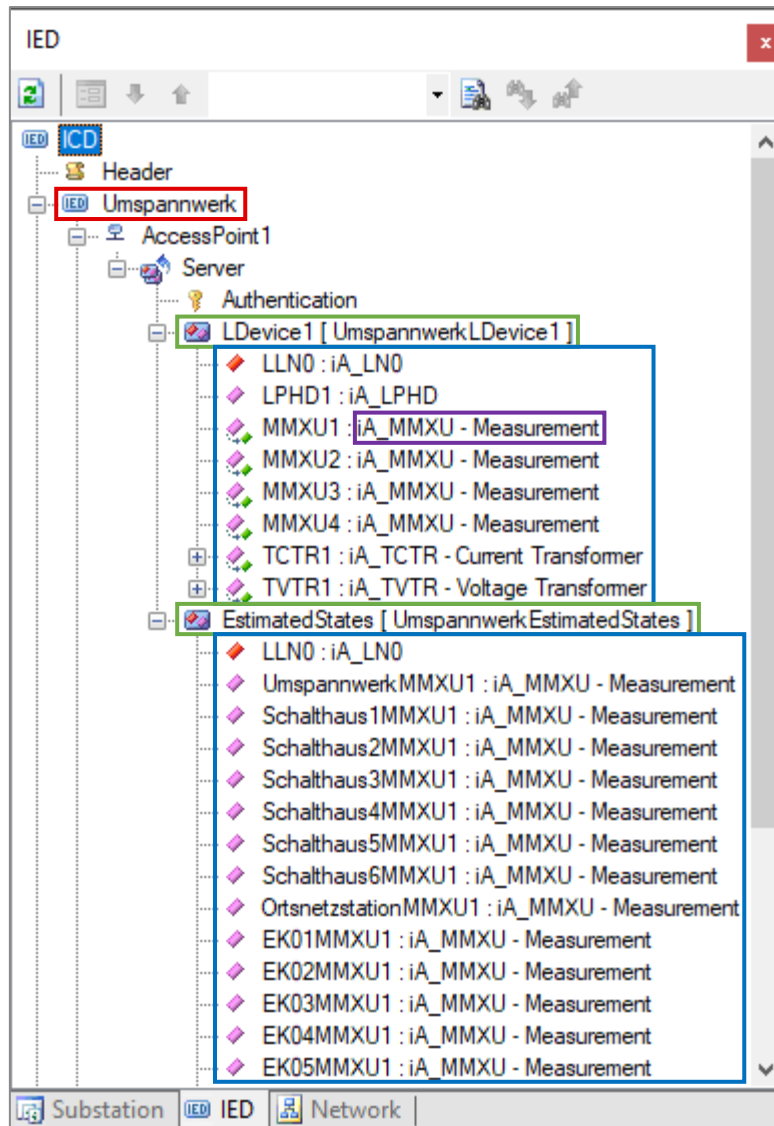


Abbildung 5.22: Datenmodell des zentralen IEDs (rot) mit logischen Geräten (grün), logischen Knoten (blau) und LN-Typ (violett)

Neben dem logischen Gerät *LDevice1* mit den für die lokale Messfunktion erforderlichen logischen Knoten enthält dieses IED ein weiteres logisches Gerät *EstimatedStates* mit logischen Knoten, in die die Ergebnisse der Netzzustandsschätzung (komplexe Spannungen) gespeichert werden. Da der Standard keine expliziten Logische-Knoten-Klassen für eine Netzzustandsschätzung vorsieht, wird hier auf logische Knoten der Klasse *MMXU* zurückgegriffen.

Im dritten Engineeringschritt wird das Anlagendatenmodell mit den IED-Datenmodellen verknüpft. Wie in Abbildung 5.23 dargestellt, werden hierbei Referenzen zwischen den abstrakten logischen Knoten im Anlagendatenmodell und den logischen Knoten in den IED-Datenmodellen erstellt. Während in Abbildung 5.19 beispielsweise

noch *None/.MMXU* stand, ist in Abbildung 5.23 nun eine konkrete Referenz wie *Umspannwerk/LDevice1.MMXU1* eingetragen.

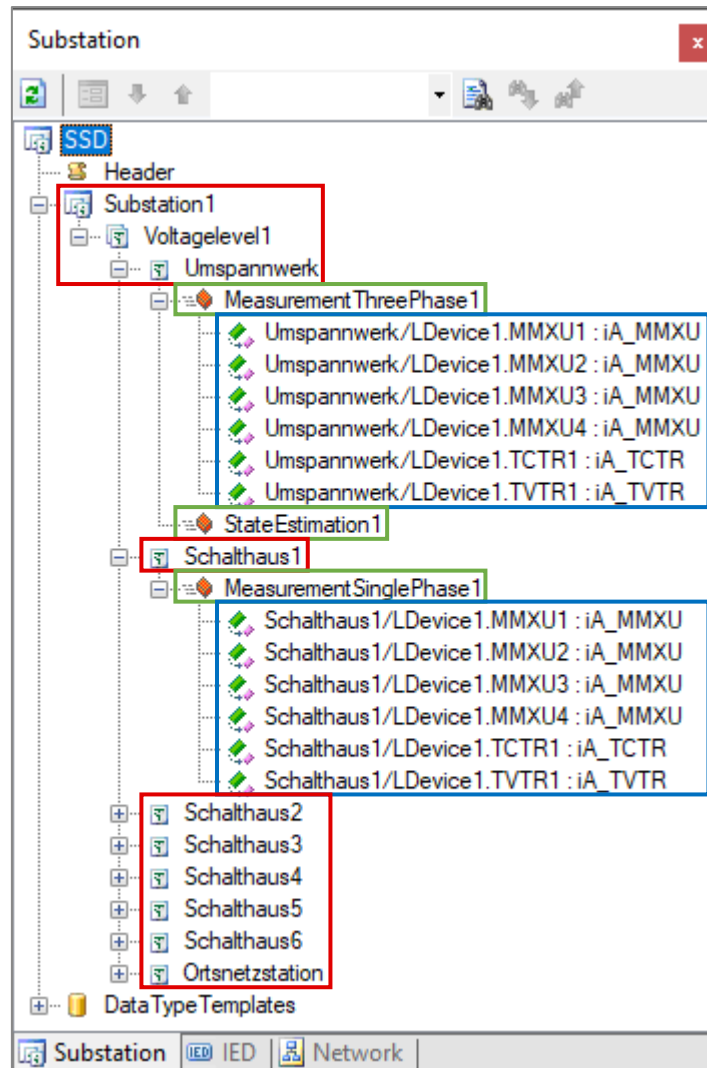


Abbildung 5.23: Datenmodell der Unterstation mit Spannungsebene und Schaltfeldern (rot), Funktionen (grün) und verknüpften logischen Knoten (blau)

Nach der Systemspezifikation wird als Nächstes die Systemkonfiguration ausgeführt. Hierbei erfolgt zunächst die Konfiguration der Kommunikation auf Anwendungsebene (siebter Engineeringsschritt). Hierzu wird mit DataSets beschrieben, *welche Daten* kommuniziert werden, und mit ReportControl- bzw. GOOSE-Control-Blöcken, *mit welchem Kommunikationsverfahren* die Daten kommuniziert werden.

Bei einem Mess-IED (Messgerät in einem Schalthaus, Abbildung 5.24) wird hierzu ein DataSet *DataSet1* angelegt. Innerhalb dieses DataSets werden Datenobjekte bzw. Datenattribute mit den zu kommunizierenden Daten referenziert. Die in dem Beispiel angegebenen Datenobjekte umfassen jeweils dreiphasige Messwerte. Konkret handelt es sich um die Sammelschienenspannung *PhV* (diese wird nur einmal übertragen, MMXU1) sowie die abgangsbezogenen Wirk- und Blindleistungen *W* bzw. *VAr* (MMXU1

bis MMXU4). Kommuniziert werden die Daten hier per Reporting. Dementsprechend wird jeweils ein Report-Control-Block *ReportControl1* angelegt. Dieser referenziert einerseits das DataSet, andererseits das Ziel der Kommunikation in Form des speziellen logischen Knotens der Klasse *ITCI* im zentralen IED (*Umspannwerk/None/ITCI1*) unterhalb des Elements *RptEnabled*.

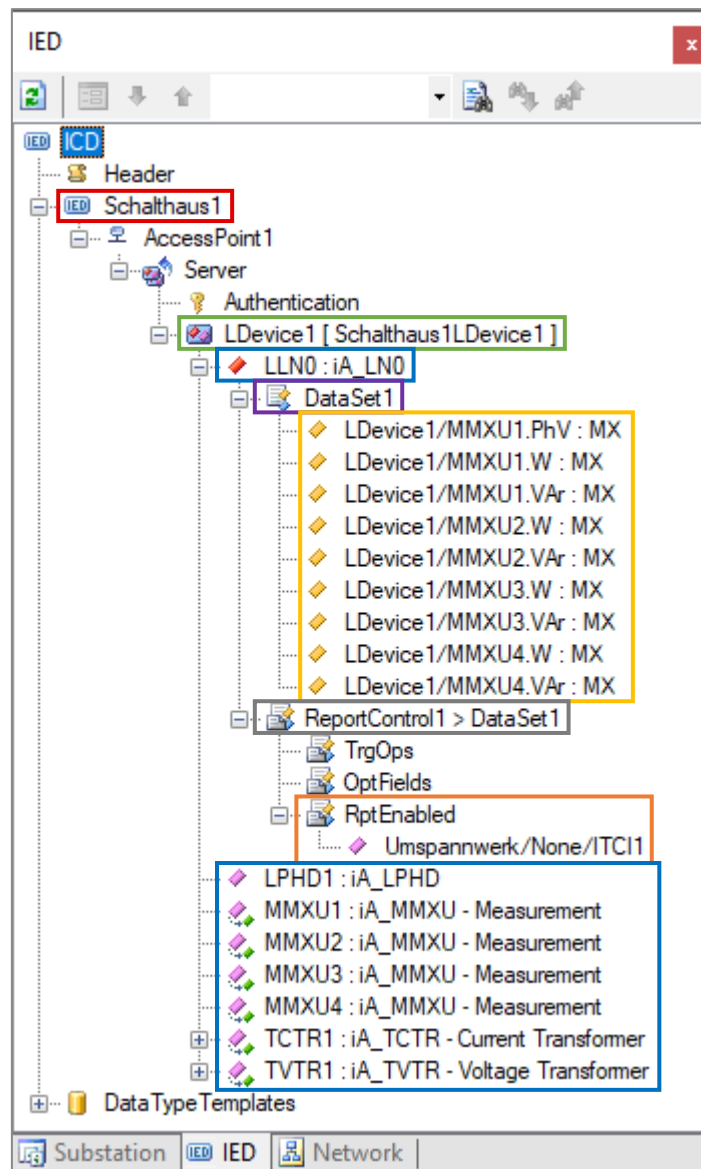


Abbildung 5.24: Kommunikationskonfiguration eines Mess-IEDs (rot) mit logischem Gerät (grün), logischen Knoten (blau), DataSet (violett) mit referenzierten Datenobjekten (gelb) sowie Report-Control-Block mit Referenz auf das DataSet (grau) und auf das Ziel der Kommunikation (orange)

Wie in Abbildung 5.25 ersichtlich, wird dieser logische Knoten *ITCI1* im zentralen IED innerhalb eines weiteren AccessPoints *ClientAP*, der ansonsten keine weiteren Elemente enthält, angelegt.

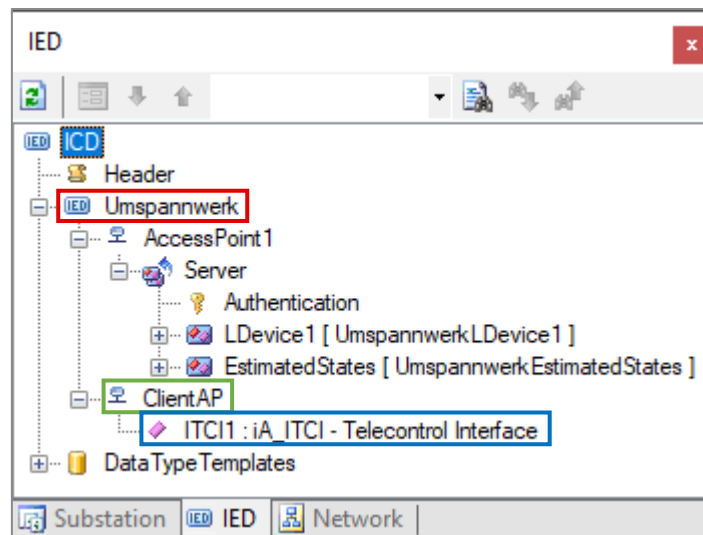


Abbildung 5.25: Kommunikationskonfiguration des zentralen IEDs (rot) mit weiterem AccessPoints *ClientAP* (grün) und speziellem logischem Knoten der Klasse *ITCI* (blau)

Als zweites erfolgt die Konfiguration der Kommunikation auf Netzwerkebene (achter Engineeringsschritt). Wie Abbildung 5.26 zeigt, wird für jeden kommunizierenden AccessPoint eines IEDs ein *ConnectedAP* angelegt und einem *SubNetwork* zugeordnet. Innerhalb der *ConnectedAPs* werden anschließend die entsprechenden Kommunikationsadressen gesetzt.

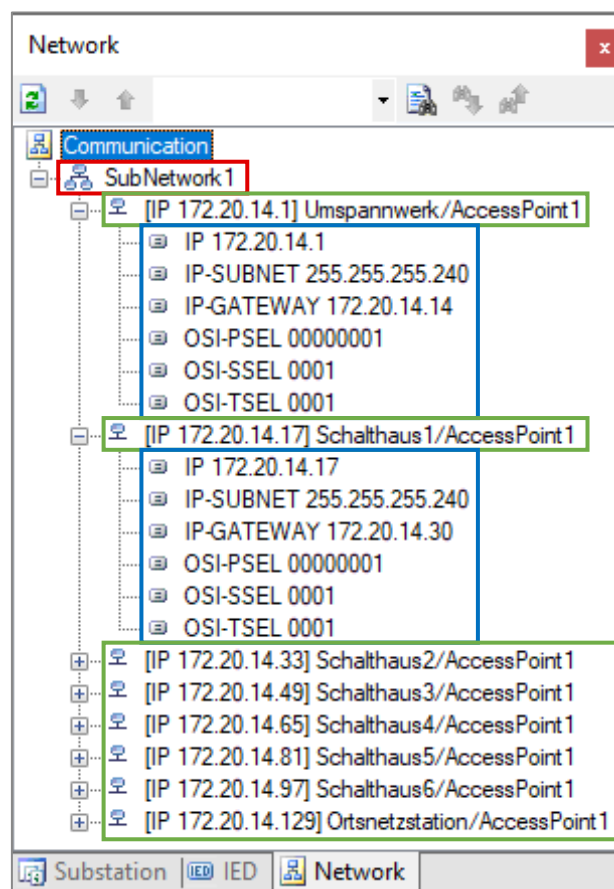


Abbildung 5.26: Netzwerkkonfiguration mit SubNetwork (rot), ConnectedAPs (grün) und Kommunikationsadressen (blau)

Abschließend wird die Gerätekonfiguration ausgeführt. In der hier vorliegenden Implementierung erfolgt die Modellierung der von einem Knoten bzw. IED auszuführenden Funktionalitäten bereits über *Function*-Elemente im Substation-Teil des SCL-Datenmodells. Diese Information wird durch die implementierte Programmkomponente *Autoconfig* bei Programmstart ausgewertet und das System entsprechend konfiguriert. Daher ist keine explizite Verknüpfung von IED-Datenmodell und IED-Applikation (neunter Engineeringschritt) erforderlich. Im Rahmen der Gerätekonfiguration muss lediglich die Datei mit dem SCL-Datenmodell auf die entsprechenden IEDs übertragen werden (zehnter Engineeringschritt).

Zusätzliche Konfigurationsdateien für die Netzzustandsschätzung

Wie bereits zuvor erwähnt, erfolgt die Konfiguration der Funktion der Netzzustandsschätzung in der aktuellen Implementierung noch nicht vollständig über das SCL-Datenmodell, sondern erfordert einige zusätzliche Konfigurationsdateien, die neben der SCL-Datenmodelldatei auf dem entsprechenden IED abgelegt sein müssen. In Tabelle 5.4 ist eine Übersicht dieser Dateien mit Beschreibung der jeweiligen Aufgabe gegeben. Es würde an dieser Stelle allerdings zu weit führen, auf die konkrete innere Struktur dieser Dateien einzugehen, so dass hierauf verzichtet wird.

Tabelle 5.4: Zusätzliche Konfigurationsdateien für die Netzzustandsschätzung

| Datei | Beschreibung |
|------------------------|--|
| StateEstimation.config | Parameter der Netzzustandsschätzung z. B. maximale Anzahl der Berechnungsiterationen, Bezugsgrößen für Spannung und Leistung zur Skalierung der physikalischen Größen auf pu-Größen, Referenzen zu den nachfolgenden Dateien |
| startingValues.csv | Definition der Knotennamen und Startwerte der Knotenspannungen (alle Knoten mit 1 pu, engl.: flat start) |
| topology.csv | Netztopologie (Knoten-Kanten-Modell mit den Leitungsparametern) |
| measurementType.csv | Zuordnung zwischen Netztopologie und Messwerten (U_i , P_i , Q_i , P_{ij} , Q_{ij} mit Referenzen zu logischen Knoten (MMXU) gemäß IEC 61850 bzw. Spaltennamen einer SQLite-Datenbanktabelle) |
| resultType.csv | Zuordnung zwischen Netztopologie und Ergebniswerten (\underline{U} mit Referenzen zu logischen Knoten (MMXU) gemäß IEC 61850 bzw. Spaltennamen einer SQLite-Datenbanktabelle) |

5.2.5 Vorprüfung der Komponenten im Labor

Vor der Einbringung der Komponenten in das Feld wurde ein Laboraufbau erstellt, um sämtliche Komponenten vorab konfigurieren und das ordnungsgemäße Zusammenspiel prüfen zu können. Hierbei wurden die Messgeräte mit den entsprechenden Programmen und Konfigurationen bespielt und die Router konfiguriert. Abbildung 5.27 zeigt die Komponenten für den Feldtest im Laboraufbau.



Abbildung 5.27: Komponenten für den Feldtest im Laboraufbau (oben: Mobilfunkrouter mit Antennen, Mitte: EPPE CX, unten: SHERLOG CRX, der DSL-Router ist nicht sichtbar)

Unter zum Feldtest identischen Bedingungen wurde abschließend das Zusammenspiel der Komponenten geprüft.

5.2.6 Einbringung der Komponenten in das Feld

Die Installation der Komponenten in den einzelnen Stationen erfolgte durch Personal des Netzbetreibers. Nach Bedarf wurden dazu seitens des Netzbetreibers zuvor notwendige Aufnahmen und Gehäuse zur mechanischen und elektrischen Montage der Komponenten vorbereitet. Nach Abschluss der Installationsarbeiten fand ein gemeinsamer Termin zur Inbetriebnahme mit Teilnehmern des Netzbetreibers und des ie³ im Netzgebiet statt. Ziel war es, die ordnungsgemäße Messwerterfassung der Messgeräte sowie die Qualität der Mobilfunkkommunikation zu überprüfen. Es wurden sämtliche, mit Messgeräten ausgestattete Stationen angefahren. Vor Ort wurde eine Netzwerkverbindung zwischen Messgerät und Service-Laptop hergestellt, um die aktuellen Messwerte aus den Messgeräten auszulesen. Diese Messwerte wurden mit Messwerten verglichen, die an den lokal vorhandenen Schutzgeräten angezeigt wurden. Neben der Größe der Messwerte wurde dabei insbesondere die Leistungsflussrichtung überprüft, da in den Stationen aus historischen Gründen die Leistungsflussrichtung der Messwandler unterschiedlich definiert wurde: Bei einigen Stationen bedeutete eine Leistung mit positivem Vorzeichen einen Leistungsfluss in die Station hinein, bei einigen Stationen hingegen einen Leistungsfluss aus der Station heraus. Die Leistungsflussrichtungen mussten in der Konfiguration der Netzzustandsschätzung berücksichtigt werden, um richtige, dem zugrunde liegenden Modell entsprechende Ergebnisse zu erhalten.

Die Abbildungen 5.28 und 5.29 zeigen beispielhaft die Installation in der Ortsnetzstation *ONS*.



Abbildung 5.28: Ortsnetzstation *ONS*



Abbildung 5.29: Ortsnetzstation ONS (Detail)

5.2.7 Durchführung und Ergebnisse des Feldtests

Sobald sämtliche Komponenten im Feld verbaut waren, wurden abschließende Konfigurationsarbeiten per Fernzugriff durchgeführt. Nach Abschluss dieser Arbeiten begann der eigentliche Zeitraum des Feldtests.

Jedes in den Schalthäusern bzw. Ortsnetzstationen installierte Gerät kommunizierte seine Messwerte zyklisch im 1-Sekunden-Zyklus zu dem Gerät im Umspannwerk Willingen. Auf Basis dieser Messwerte wurde dort im 60-Sekunden-Zyklus eine Netzzustandsberechnung ausgeführt. Deren Ergebnisse wurden zusammen mit den zugehörigen Messwerten in eine Datenbank abgelegt. Regelmäßig wurden per Fernzugriff Kopien dieser Datenbank heruntergeladen, um frühzeitig mit einer Auswertung der Daten beginnen zu können.

Im Folgenden soll auf die aufgezeichneten Messwerte und die Berechnungsergebnisse der online ausgeführten Netzzustandsschätzung eingegangen werden. Abbildung 5.30 zeigt den Verlauf der gemessenen Spannungen in allen acht Stationen über drei Tage, Abbildung 5.31 eine Detaildarstellung für einen Tag.

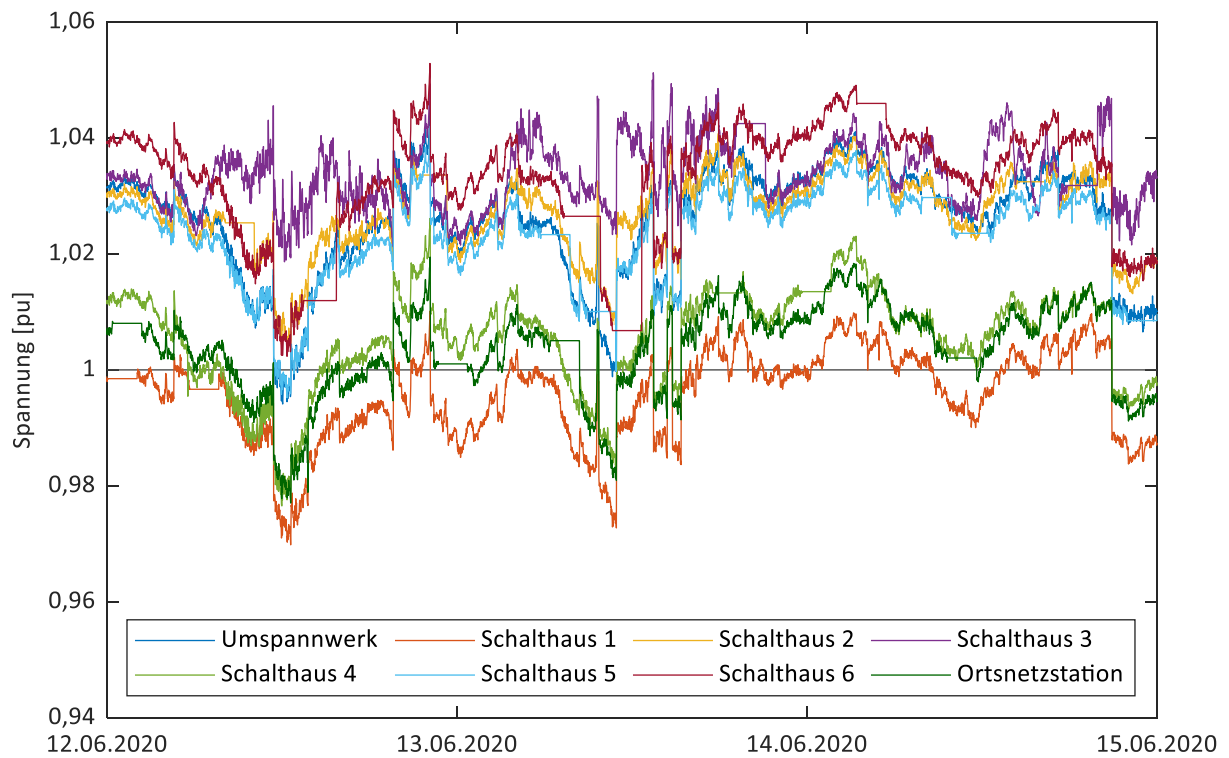


Abbildung 5.30: Gemessene Spannungen

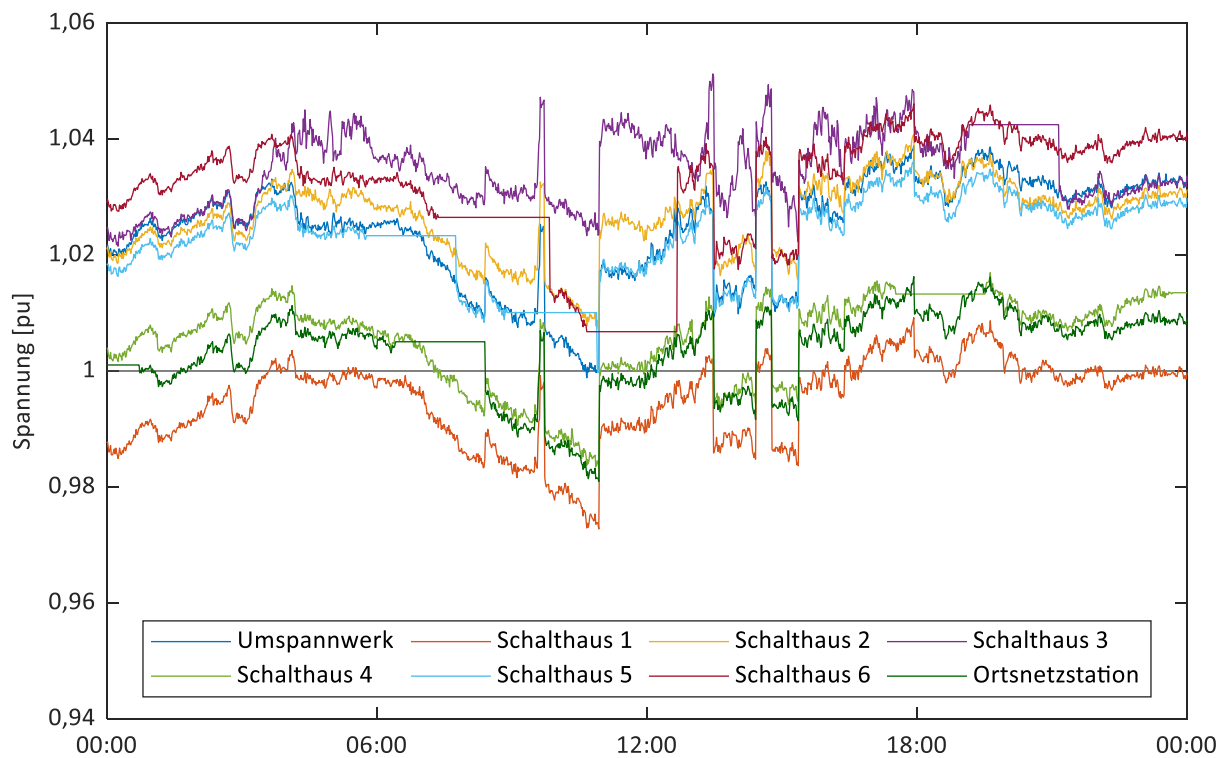


Abbildung 5.31: Gemessene Spannungen (Detail 13.06.2020)

In den Messwerten zeigen sich einerseits Kommunikationsunterbrechungen dergestalt, dass Messwerte über einen gewissen Zeitraum nicht aktualisiert werden und damit auf dem zuletzt empfangenen Wert verharren, etwa in der Ortsnetzstation von ca. 6:00 bis 8:00 in Abbildung 5.31. Ursachen liegen einerseits in gelegentlichen Verbindungsstörungen der Mobilfunkkommunikation selbst. Auffällig sind jedoch sich wiederholende, länger andauernde Unterbrechungen im Stundenbereich. Als konkrete Ursache wurde ein systematisches Problem bei der regelmäßigen Schlüsselerneuerung des genutzten VPN-Sicherungsverfahrens IPsec vermutet. Durch die hohe Komplexität von IPsec konnte jedoch trotz umfangreicher Analyse im Zeitrahmen des Feldtests keine Lösung gefunden werden.

Weiterhin werden in Spannungssprüngen die Stufungen des zentralen HS-/MS-Transformators im Umspannwerk sichtbar: Beispielsweise gegen 11:00 zeigt sich ein Hochstufen des Transformators, gegen 13:30 wird dieser wieder heruntergestuft, um gegen 14:30 erneut hochgestuft zu werden. Die Höhe der Spannung im Umspannwerk liegt planmäßig etwas über der Nennspannung von 1 pu, um Spannungsabfälle im Netz in Richtung der Kunden auszugleichen. Entsprechend fallen die Spannungen zu den Schalthäusern 1 und 4 sowie zur Ortsnetzstation ab. Die erhöhten Spannungen insbesondere in den Schalthäusern 2, 3 und 6 lassen sich durch starke dezentrale Einspeisung unter anderem aus einem Windpark und einem Blockheizkraftwerk erklären, wie bereits zuvor bei der Netzmodellauflösung in Kapitel 5.2.1 diskutiert und in Abbildung 5.11 dargestellt.

Es zeigten sich von Station zu Station teilweise deutlich unterschiedliche Abweichungen zwischen gemessener und estimerter Spannung. In den Abbildungen 5.32 und 5.33 sind jeweils beide Spannungsverläufe für Schalthaus 1 und 2 dargestellt. Während im Schalthaus 1 die estimierte Spannung durchgehend ca. 2 % höher als die gemessene Spannung ist, sind in Schalthaus 2 beide Verläufe nahezu deckungsgleich. Abbildung 5.34 zeigt zusammenfassend die prozentuale Abweichung der estimeren von den gemessenen Spannungen für alle Stationen.

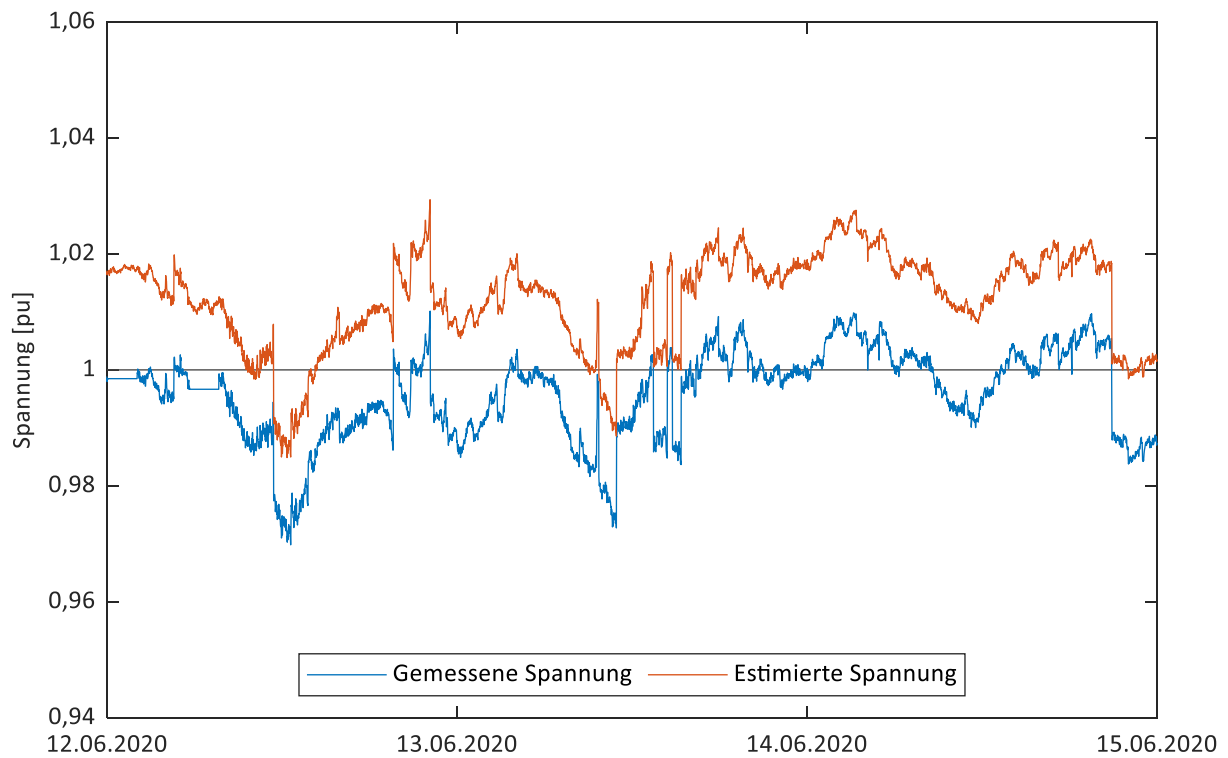


Abbildung 5.32: Gemessene und estimierte Spannung im Schalthaus 1

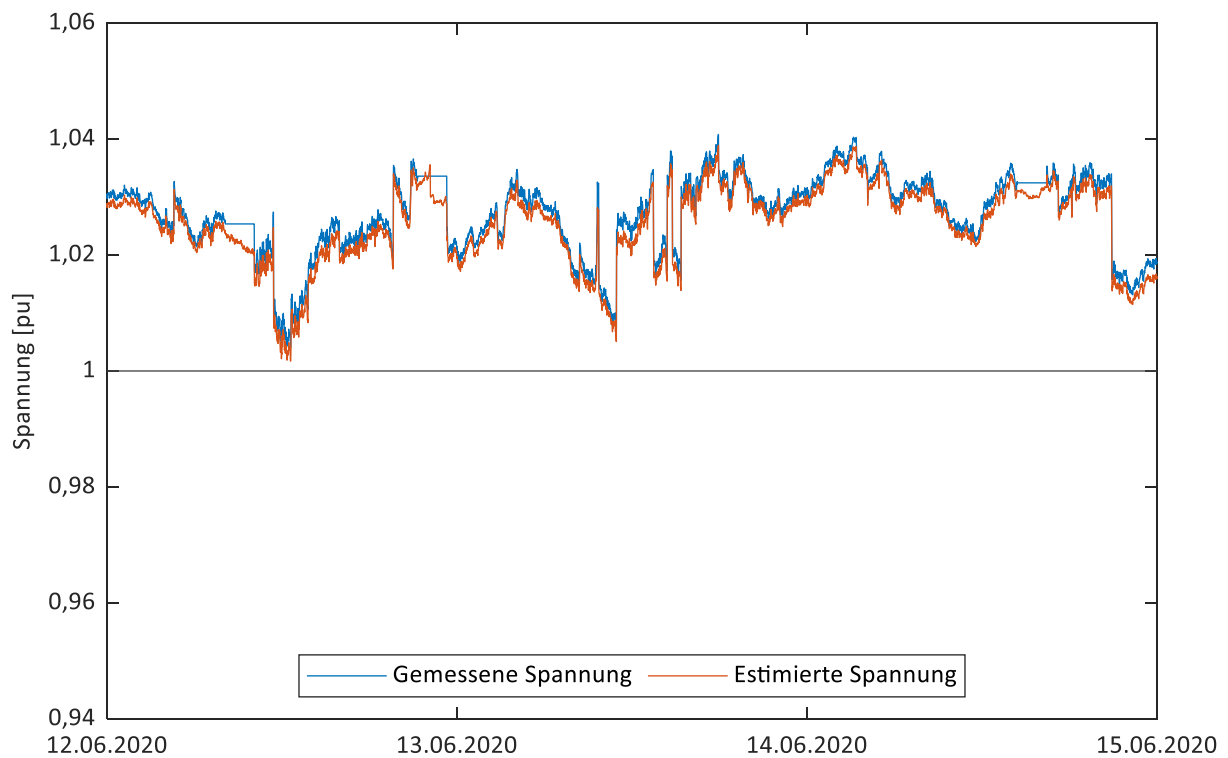


Abbildung 5.33: Gemessene und estimierte Spannung im Schalthaus 2

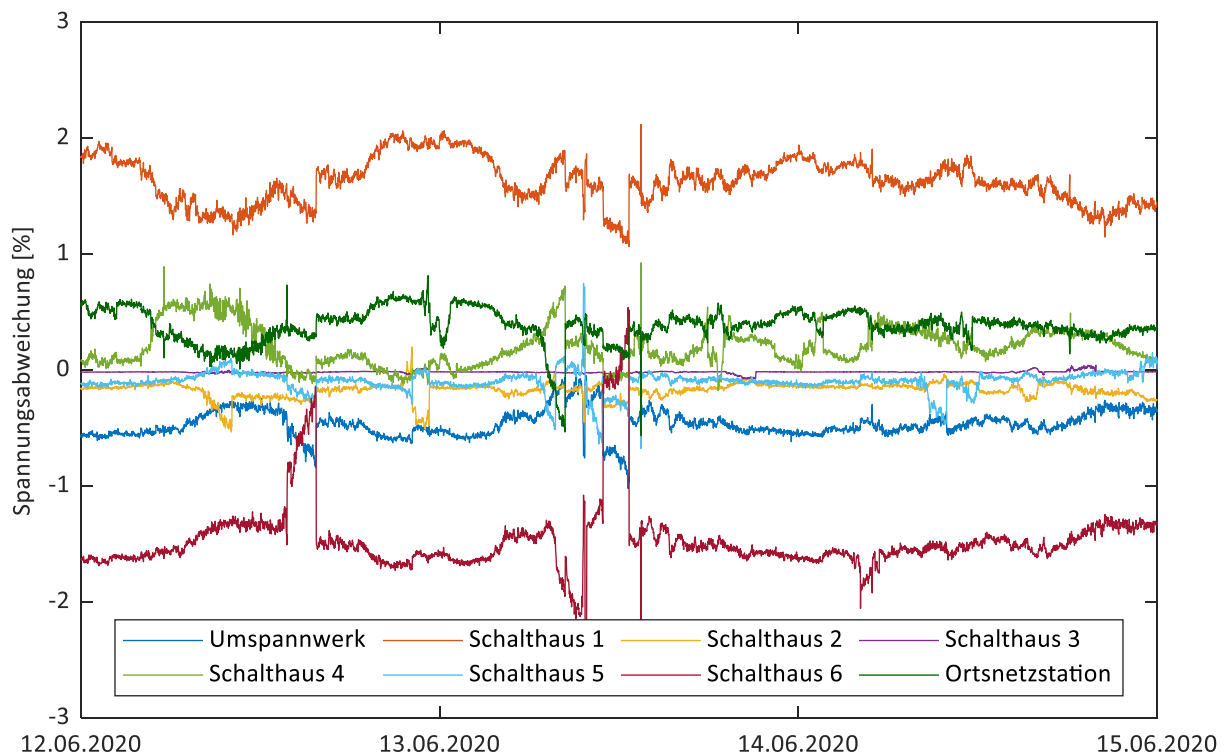


Abbildung 5.34: Abweichungen zwischen estimierten und gemessenen Spannungen

Ähnliche Abweichungen zeigten sich bereits in Abbildung 5.15 in Kapitel 5.2.1, wo die Beträge und Winkel der Knotenspannungen aus einer Simulation des nicht reduzierten Netzes denen des reduzierten Netzes gegenübergestellt wurden. Die im Feld gemessenen Spannungen sind vergleichbar denen des nicht reduzierten Netzes, während die estimierten Spannungen auf Basis des reduzierten Netzes berechnet werden. Als weiterer starker Einflussfaktor muss jedoch auch das dynamische Verhalten des Netzes genannt werden. Da das Netzmodell, das der Netzreduktion zugrunde liegt, nur einen statischen Lastflussfall abbildet, ist auch die Netzreduktion nur für diesen einen Fall als optimal anzusehen. Das reale Netz des Feldtests nimmt jedoch nach Last- und Einspeisesituation beliebige und dynamisch wechselnde Lastflüsse an, etwa im Unterschied von Tag und Nacht, von Starklast und Schwachlast oder von hoher und geringer Einspeisung. Dadurch ist das zugrunde liegende Berechnungsmodell mal mehr, mal weniger geeignet zur Berechnung der Netzzustände. Ein weiterer, jedoch vermutlich geringerer Einflussfaktor mag in Messungenauigkeiten liegen, beispielsweise durch Übersetzungsfehler der in den Stationen verbauten Spannungswandler.

Ziel der Netzzustandsschätzung im Feldtest war es, insbesondere auch die komplexen Netzspannungen an ausgewählten, nicht gemessenen Knoten im Netz, den Ersatzlasten, abzuschätzen. Während der Netzreduktion wurden die Ersatzlasten so positioniert und skaliert, dass die Abweichungen der Leistungsflüsse an den gemessenen Knoten gegenüber dem nicht reduzierten Netzmodell möglichst gering

bleiben. Der Netzzustand der Ersatzlast repräsentiert damit einen „mittleren“ Netzzustand.

Exemplarisch soll der Strang zwischen den Schalthäusern 2 und 3 betrachtet werden, da an beiden Stationen die Abweichungen zwischen gemessener und estimerter Spannung gering waren. Die Ortsnetzstationen zwischen den beiden Schalthäusern wurden im Rahmen der Netzreduktion in den Ersatzknoten EK 11 zusammengefasst (siehe reduziertes Netz in Abbildung 5.14).

Abbildung 5.35 zeigt die estimierte Spannungen aller drei Knoten. Ersichtlich ist, dass die Spannung in Schalthaus 3 nahezu durchgehend größer ist als die in Schalthaus 2 und die estimierte Spannung des Ersatzknotens zwischen den Spannungen der beiden Stationen liegt. Aus dem Spannungsprofil lässt sich schließen, dass der Leistungsfluss überwiegend vom Schalthaus 3 über den Ersatzknoten zum Schalthaus 2 fließt.

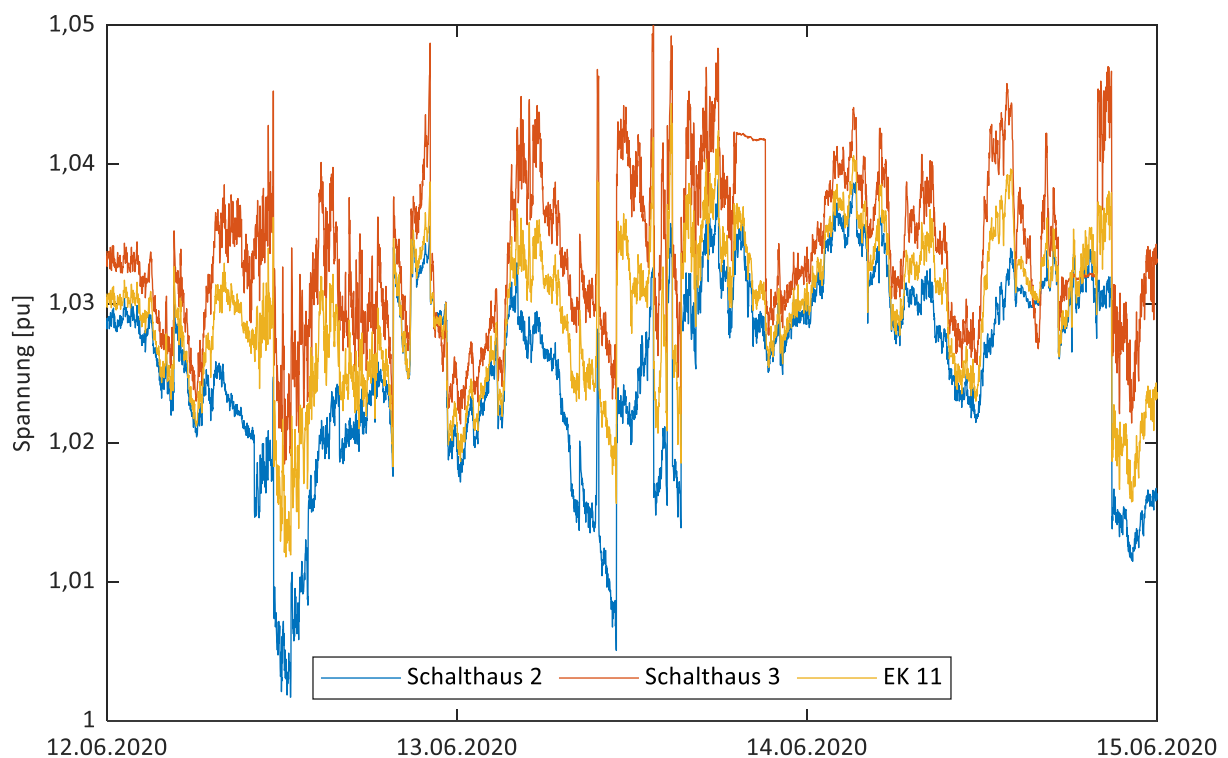


Abbildung 5.35: Estimierte Spannungen der Schalthäuser 2 und 3 und des Ersatzknotens EK 11

Dies soll anhand Abbildung 5.36, die die in den Schalthäusern 2 und 3 gemessenen Wirkleistungsflüsse in Richtung Strang bzw. Ersatzknoten zeigen, bestätigt werden. Es zeigt sich, dass insbesondere tagsüber Wirkleistung aus dem Schalthaus 3 in den Strang fließt und gleichzeitig Wirkleistung aus dem Strang in das Schalthaus 2 fließt. Damit ergibt sich ein Leistungsfluss vom Schalthaus 3 über den Strang bzw. den Ersatzknoten in das Schalthaus 2, was dem zuvor beobachteten Spannungsprofil entspricht. Zu sehen ist darüber hinaus, dass teilweise mehr Wirkleistung aus dem Strang in das

Schaltheis 2 fließt als aus dem Schaltheis 3 in den Strang. Dies ist durch stärkere Einspeisung aus Photovoltaikanlagen im Bereich des Stranges zu erklären und deckt sich auch mit der Beobachtung, dass die hohen Leistungsüberschüsse tagsüber auftreten und ihr Maximum zum Mittag hin aufweisen.

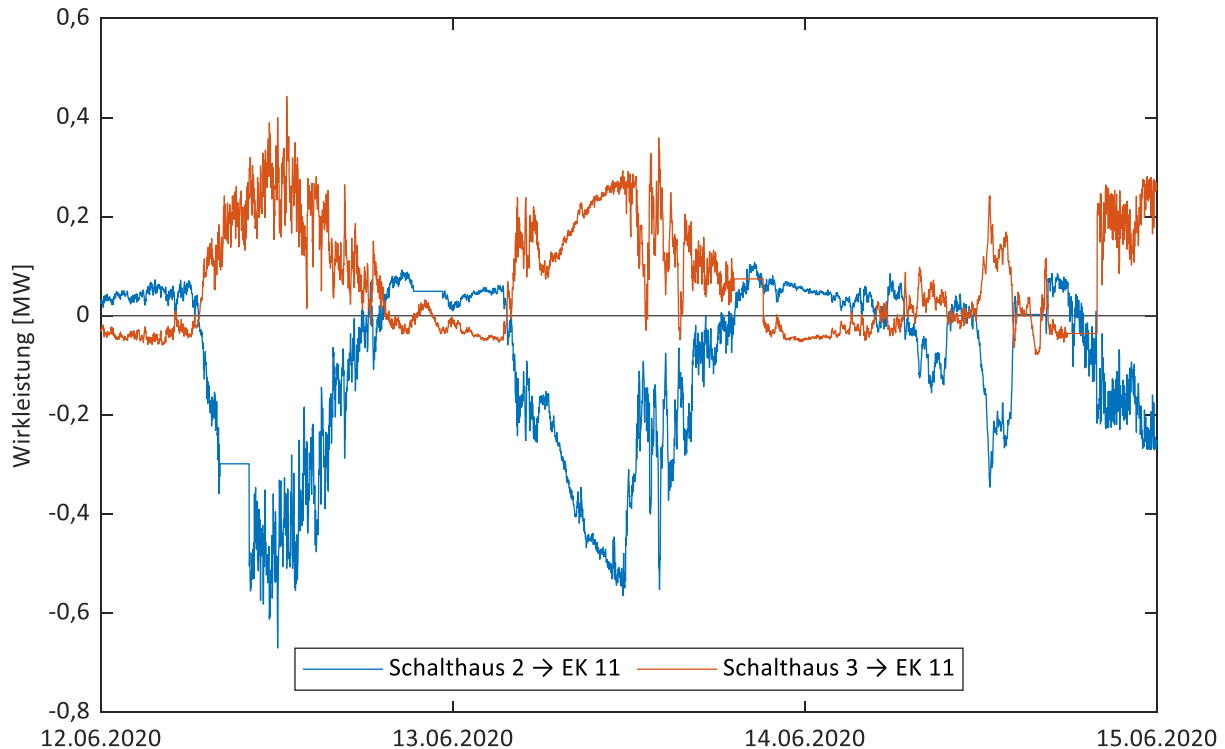


Abbildung 5.36: Wirkleistungsflüsse der Schaltheiser 2 und 3 zum Ersatzknoten EK 11

Durch die im Rahmen des Feldtests installierten Systeme war es grundsätzlich möglich, anhand der erfassten Messwerte ein Online-Abbild der Verhältnisse im Netz zu erlangen und beispielsweise die Einhaltung der Spannungsgrenzwerte zu überprüfen. Als wichtige Funktion innerhalb des Feldtests ist jedoch die Netzzustandsschätzung zu sehen, um einerseits die Messwerte zu verifizieren und andererseits die Netzspannungen an ausgewählten, nicht gemessenen Knoten abzuschätzen. Beim Vergleich zwischen gemessener und estimerter Spannung zeigten sich dabei teilweise Abweichungen von bis zu 2 %. Eine Hauptursache ist in der starken Vereinfachung des Netzmodells im Rahmen der Netzreduktion zu suchen. Exemplarisch wurden an einem Strang die Spannungen der beiden gemessenen Netzknoten und des estimierten Ersatzknotens den gemessenen Leistungsflüssen gegenübergestellt. Spannungsprofil und Leistungsflüsse erwiesen sich dabei als physikalisch korrekt. Zu beachten ist, dass bei der Netzreduktion ein Ersatzknoten zum Teil zahlreiche Ortsnetzstationen zusammenfasst. Starke Leistungsflüsse zwischen einzelnen Ortsnetzstationen, etwa durch dezentrale Einspeisungen, werden auf diese Weise jedoch nicht ersichtlich, so dass etwa Spannungsbandverletzungen an einzelnen Ortsnetzstationen oder Überlastungen von Leitungen unerkannt bleiben können. Weiterhin handelte es sich

bei der Netzzustandsschätzung um ein exakt bestimmtes System. Eine Identifikation und Kompensation normalverteilter Messfehler ist auf diese Weise nicht möglich, hierzu wären das Hinzufügen weiterer Messorte und damit die Überführung des Systems in ein überbestimmtes System erforderlich.

Als besonders problematisch erwiesen sich häufige Kommunikationsstörungen. Eine Ursache schien im genutzten VPN-Sicherungsverfahren zu liegen. Trotz umfangreicher Analyse und Variation von Parametern konnte hier keine Lösung gefunden werden, ein alternatives VPN-Sicherungsverfahren wurde seitens der bereitgestellten Router nicht unterstützt. Es zeigte sich aber auch, dass die Mobilfunkabdeckung selbst ortsweise unzuverlässig war, so dass es zu Abbrüchen kam und nur geringe Datenraten erzielt wurden. Bei zukünftigen Anwendungen ist es daher empfehlenswert, auch alternative Kommunikationsmethoden auf eine Eignung hin zu testen, insbesondere mit Blick auf heterogene Kommunikationsstrukturen im Sinne einer Nutzung des jeweils vor Ort am besten geeigneten Verfahrens.

5.3 Zusammenfassung

In diesem Kapitel wurde das ordnungsgemäße Verhalten ausgewählter Funktionen und des Gesamtsystems selbst verifiziert. Im ersten Teil wurde exemplarisch der Distanzschutz als eine der realisierten Schutzfunktionen betrachtet. Es wurde das Verhalten des Distanzschutzes in Bezug auf Messgenauigkeit und Auslösezeiten untersucht. Primär wurde der auf dem Echtzeitbetriebssystem implementierte Distanzschutz betrachtet, da nur hier ein deterministisches Zeitverhalten möglich ist. Dieser Distanzschutz zeigte ein schnelles Zeitverhalten und eine zufriedenstellende Messgenauigkeit. Es erfolgte ein Vergleich mit dem auf dem Linux-Betriebssystem ausgeführten Distanzschutz, der erwartungsgemäß ein deutlich schlechteres Zeitverhalten und insbesondere größere Ausreißer aufwies. Die Betrachtungen beschränkten sich in allen Fällen auf den dreiphasigen, symmetrischen Fehler. In weitergehenden Untersuchungen werden daher auch andere Fehlerereignisse, wie etwa unsymmetrische Fehler, betrachtet werden müssen.

Im zweiten Teil wurde zunächst der vollständige Engineeringprozess dargestellt, der erforderlich ist, um einen Netzbereich mit intelligenten Smart-Grid-Funktionen auszustatten. Exemplarisch wurde hierzu die Netzzustandsschätzung ausgewählt. Es wurden die vorbereitenden Analysearbeiten aufgezeigt, zu denen unter anderem die Reduzierung des gegebenen Netzmodells und die Verifikation im Rahmen von Simulationen gehörten. Ergebnis dieser Arbeiten waren schließlich eine Aussage über die Platzierung der Messsysteme innerhalb des Netzbereichs und eine hierauf angepasste reduzierte Netztopologie. Es erfolgten die Modellierung der sich daraus ergebenden applikations- und kommunikationstechnischen Struktur und die Konfigu-

ration der Systeme. Im Rahmen einer Vorprüfung im Labor wurden die ordnungsgemäße Durchführung des Engineeringprozesses und die fehlerfreie Konfiguration der Systeme verifiziert.

Es folgte die Installation der Systeme im Feld. Während des Feldtests wurde das Verhalten des Systems über einen längeren Zeitraum untersucht. Die hierbei erlangten Ergebnisse eines exemplarischen Zeitraums wurden aufgezeigt. Neben der Verifikation der Messwerte an gemessenen Stationen ermöglichte die Netzzustandsschätzung vor allem die Abschätzung der Netzspannungen an nicht gemessenen Netzknoten. Der Vergleich zwischen gemessener und estimierter Spannung zeigte allerdings teils beträchtliche Abweichungen, die jedoch auf die starke Vereinfachung des Netzmodells zurückzuführen ist. Auch wurde die Netzreduktion durch iterative, teils manuelle Verfahren vorgenommen. Hier sind durch Anwendung neuer Verfahren, die auf Optimierungsansätzen beruhen, zukünftig bessere Ergebnisse zu erwarten. Die genutzte Mobilfunkkommunikation mit VPN-Sicherungsverfahren erwies sich als störanfällig, was sich durch häufige Kommunikationsunterbrechungen zeigte. Die Nutzung heterogener oder auch dedizierter Kommunikationsstrukturen sollte als Alternative zur gewöhnlichen Mobilfunkkommunikation betrachtet werden.

Der Nachweis, dass die Durchführung des Engineering- und Konfigurationsprozesses zu einem fehlerfrei eingerichteten Gesamtsystem führt, und dass die Netzzustandsschätzung wie auch die Kommunikationsverfahren grundsätzlich funktionieren, konnte trotz der genannten Detailprobleme erbracht werden.

6 Zusammenfassung und Ausblick

Mit dem Wegfall von Großkraftwerken, der Integration einer Vielzahl dezentraler und leistungsstarker Einspeiser und Lasten und dem daraus folgenden Wandel der Leistungsflüsse stellt die Energiewende besonders die elektrischen Verteilnetze vor große Herausforderungen. Mit klassischen Netzausbaumaßnahmen allein ließe sich diesen Herausforderungen nur aufwendig und kostspielig begegnen. Eine Alternative oder Ergänzung besteht in der Integration von Systemen und Funktionen zur Messung, Überwachung, Steuerung, Automatisierung und zum Schutz und damit in der Überführung der bisher passiven Verteilnetze in intelligente Smart Grids. Es existieren bereits verschiedene Lösungen, sowohl als kommerzielle Produkte als auch als Gegenstand und Ergebnis der Forschung. Allerdings weisen diese immer auch noch eine Reihe von Nachteilen auf. Die Systeme bieten oftmals nur einen begrenzten und fest definierten Funktionsumfang, so dass mehrere Systeme kombiniert werden müssen, um die erforderliche Gesamtfunktionalität zu erreichen. Außerdem erfordern sie umfangreiche, manuelle und fehlerträchtige Projektierungs- und Konfigurationsarbeiten – für einen flächendeckenden Einsatz in den zahlreichen und weiträumigen Verteilnetzen ein erheblicher Hinderungsgrund.

In dieser Arbeit wurde eine Smart-Grid-Automatisierungsarchitektur vorgestellt, die einigen dieser Herausforderungen begegnet. Sie ermöglicht es, verschiedenste Funktionen und Algorithmen, wie sie für den Aufbau eines Smart Grids von Bedeutung sind, modular und flexibel zu implementieren und nebeneinander auf einer Hardwareplattform auszuführen. Berücksichtigt wurde vor allem auch eine klare Trennung von Software und Hardware. Eine weitere Besonderheit stellt der integrierte Konfigurationsprozess auf Basis der IEC 61850-6 dar. Mit seiner Hilfe kann das System im Rahmen eines übergeordneten Engineeringprozesses nach ebendiesem Standard schnell und effizient projiziert und konfiguriert werden.

6.1 Zusammenfassung

Basierend auf einer grundlegenden Betrachtung von Schutz-, Leit- und Automatisierungssystemen für Smart Grids wurden Kriterien definiert, anhand derer eine Kategorisierung von Smart-Grid-Funktionen und -Systemen vorgenommen werden kann, und diese an beispielhaften Funktionen erläutert. Hieraus wurden Anforderungen an eine Ausführungsumgebung, wie etwa eine Automatisierungshardware oder Kommunikationsinfrastruktur, abgeleitet. Diese können genutzt werden, um zu prüfen und zu bewerten, ob eine ordnungsgemäße Ausführung konkreter Smart-Grid-Funktionen auf einer gegebenen Ausführungsumgebung möglich ist. Schließlich wurden bereits am Markt als Produkt verfügbare Systeme und Lösungsansätze aus der Forschung

analysiert. Hierbei zeigte sich, dass diese zwar einzelne Probleme und Herausforderungen lösen, aber dennoch immer auch durch eine Reihe von Nachteilen gekennzeichnet sind. Insbesondere fehlt ihnen die für den Aufbau von Smart Grids erforderliche Flexibilität und Modularität. Auch sind geeignete Engineering- und Konfigurationsprozesse als Voraussetzung für ein flächendeckendes Ausrollen dieser Systeme allerhöchstens im Ansatz vorzufinden. Auf Seiten der Smart-Grid-Automatisierungssysteme besteht dementsprechend noch erheblicher Optimierungsbedarf.

Zur Betrachtung von Prozessen der Projektierung und des Engineerings elektrischer Energieanlagen wurden zunächst bestehende Prozesse beleuchtet und anhand zweier Anwendungsbeispiele erläutert. Hierbei zeigte sich die hohe Komplexität dieser Prozesse, etwa durch Nutzung verschiedener unabhängiger Softwarewerkzeuge, Datenformate und Medien. In der Folge werden, durch zahlreiche Schnittstellen und die fehlende Durchgängigkeit der Engineeringdaten, umfangreiche manuelle und fehleranfällige Arbeitsschritte notwendig, was zu hohem personellem und finanziellem Aufwand führt. Als vielversprechende Alternative hierzu wurde die Normenreihe IEC 61850 vorgestellt. Es wurde ein Überblick über die in ihrem Teil 6 definierten Datenmodelle gegeben. Diese ermöglichen eine einheitliche und konsistente Modellierung der funktionalen primärtechnischen und insbesondere sekundärtechnischen Objekte sowie der Kommunikation elektrischer Energieanlagen. Die ebenfalls in diesem Normenteil skizzierten Engineeringprozesse decken die Spezifikations- und Konfigurationsphasen bei der Erstellung eines schutz-, leit- und automatisierungstechnischen Gesamtsystems ab. Beides zusammen kann gewinnbringend genutzt werden, um die Nachteile der zuvor genannten bestehenden Prozesse aufzulösen.

Die Realisierung der Smart-Grid-Automatisierungsarchitektur bildete den Kern dieser Arbeit. Zunächst wurde ein Überblick über die Architektur und technischen Kenndaten der exemplarisch genutzten Hardwareplattform gegeben. Das Softwareframework, dessen Erstellung die erste Hauptaufgabe dieser Arbeit war, wurde vorgestellt und aufgezeigt, wie sich innerhalb dieses Softwareframeworks Smart-Grid-Funktionen und Algorithmen umsetzen lassen. Anschließend wurden exemplarisch die Implementierung ausgewählter Funktionen erläutert und ein Ausblick auf weitere, in parallelen Arbeiten realisierte Funktionen gegeben. Ergänzend wurde die Umsetzung verschiedener Kommunikationsverfahren dargestellt. Schließlich erfolgte die ausführliche Darstellung des Entwurfs und der Implementierung eines Konfigurationsprozesses auf Basis des Standards IEC 61850-6, die zweite Hauptaufgabe dieser Arbeit. Es wurde beschrieben, welche in diesem Standard definierten Sprachelemente ausgewählt wurden, um die Funktionalität des Automatisierungssystems zu modellieren und wie auf Basis eines solchen Datenmodells die automatische

Gerätekonfiguration erfolgt. Schließlich wurde aufgezeigt, wie die Programmablaufsteuerung im Detail die parallele und wechselwirkungsfreie Ausführung und Koordination mehreren Smart-Grid-Funktionen auf einem Automatisierungssystem sicherstellt.

Schließlich erfolgte die Verifikation der zuvor erstellten Smart-Grid-Automatisierungsarchitektur. Um insbesondere die Leistungsfähigkeit der Plattform nachzuweisen, wurde im Laborumfeld zunächst der Distanzschutz als eine der implementierten Funktionen mit hohen Anforderungen betrachtet und hinsichtlich Auslösegenauigkeit und -geschwindigkeit untersucht. Bei Ausführung auf dem echtzeitfähigen Co-Prozessor zeigte er ein schnelles Zeitverhalten und eine zufriedenstellende Messgenauigkeit. Im zweiten Schritt lag der Fokus auf der Verifikation des Prozesses zur Gerätekonfiguration und des überlagerten Engineerings sowie des gesamten Zusammenspiels von Funktionen, die auf mehreren Geräten ausgeführt werden und miteinander kommunizieren. Neben Vorprüfungen im Labor fand hier insbesondere auch ein Feldtest statt. Es konnte nachgewiesen werden, dass die Durchführung des Engineering- und Konfigurationsprozesses zu einem fehlerfrei eingerichteten Gesamtsystem führte, und dass die Netzzustandsschätzung als exemplarische Smart-Grid-Funktion sowie die Kommunikationsverfahren funktionierten.

6.2 Bewertung

Zu Anfang dieser Arbeit wurden drei Forschungsfragen aufgestellt. Diese Forschungsfragen sollen an dieser Stelle noch einmal aufgegriffen werden, um zu betrachten, wie diese Arbeit zu ihrer Beantwortung beigetragen hat.

Forschungsfrage 1: Wie lässt sich eine Smart-Grid-Automatisierungsarchitektur so gestalten, dass verschiedenste Funktionen zur Überwachung, Steuerung, Automatisierung und zum Schutz eines Smart Grids modular und flexibel auf einer Plattform umgesetzt und nebeneinander ausgeführt werden können?

Es konnte gezeigt werden, dass die Smart-Grid-Automatisierungsarchitektur mit dem im Rahmen dieser Arbeit realisierten Softwareframework erfolgreich zur Umsetzung verschiedenster Funktionen genutzt werden konnte. Die exemplarische Betrachtung konkreter Funktionsimplementierungen beschränkte sich in dieser Arbeit zwar auf den Distanzschutz als eine Schutzfunktion sowie auf die Netzzustandsschätzung als eine Smart-Grid-Monitoringfunktion. Allerdings wurde und wird das Softwareframework darüber hinaus umfangreich in weiteren Forschungsarbeiten des ie³ eingesetzt, um weitere Funktionen und Algorithmen zu implementieren und sowohl im Labor zu verifizieren als auch in Feldversuchen unter realen Netzbedingungen zu erproben.

Beispiele für solche Funktionen sind verschiedene Varianten einer Spannungsregelung, Leistungsflussoptimierungen, die modellprädiktive Regelung dezentraler Netzteilnehmer, Algorithmen zur Teilnetzerkennung und Inselnetzbildung sowie zur Fehlerrichtungserkennung. Durch die klar definierten Softwareschnittstellen des Softwareframeworks konnten und können diese Funktionen unabhängig voneinander und ohne störende Wechselwirkungen implementiert und einzeln oder in beliebiger Kombination auf dem Smart-Grid-Automatisierungssystem ausgeführt werden.

Forschungsfrage 2: Wie kann ein Konfigurationsprozess aussehen, der eine solche Smart-Grid-Automatisierungsarchitektur einfach und durchgängig für die Ausführung verschiedener Smart-Grid-Funktionen konfiguriert?

Das Ziel bei der Schaffung einer flexibel und modular nutzbaren Smart-Grid-Automatisierungsarchitektur liegt darin, eine saubere Trennung zwischen der Umsetzung von Funktionen und Algorithmen auf der einen Seite und deren Einsatz in konkreten Anwendungsfällen auf der anderen Seite zu schaffen. Hierdurch soll insbesondere verhindert werden, dass vom Grunde her gleichartige Funktionen und Algorithmen für verschiedene Anwendungsfälle jeweils erneut implementiert werden. Denn gerade die Generalität und Wiederverwendbarkeit von Funktionen zeichnet eine effiziente Entwicklung aus. Diesem Credo folgend dient das Softwareframework auf der einen Seite dazu, Funktionen umzusetzen, so dass sie anschließend innerhalb eines Funktionspools für beliebige Anwendungen zur Verfügung stehen. Auf der anderen Seite wurde ein Konfigurationsprozess entworfen, mit dem beschrieben werden kann, welche Funktionen für eine konkrete Anwendung zum Einsatz kommen und wie diese Funktionen interagieren, der aber dennoch losgelöst ist von Implementierungsdetails der Funktionen selbst. Somit können beide Seiten, die Implementierung von Smart-Grid-Funktionen und die Konfiguration für eine konkrete Anwendung, unabhängig voneinander bearbeitet werden.

Forschungsfrage 3: Wie können die im Standard IEC 61850-6 definierten Mittel genutzt werden, um einen Konfigurationsprozess zu gestalten, der sich nahtlos in einen herstellerneutralen Engineeringprozess nach ebendiesem Standard einfügt?

Der Standard IEC 61850 definiert in seinem Teil 6 Datenmodelle und abstrakte Engineeringprozesse, die es ermöglichen, die funktionalen primär- und sekundärtechnischen Objekte sowie die Kommunikation elektrischer Energieanlagen einheitlich, konsistent und vor allem geräte- und herstellerneutral zu modellieren. Auch wenn der Standard sich ursprünglich auf die Schutz- und Leittechnik klassischer Umspannanlagen der Hoch- und Höchstspannung bezog, so ist eine Erweiterung auf

Anlagen des Verteilnetzes naheliegend. In dieser Arbeit wurden die in diesem Standard definierten Modellierungsmittel analysiert und auf ihre Nutzbarkeit für die hier realisierte Smart-Grid-Automatisierungsarchitektur überprüft. Dabei zeigte sich, dass eine funktionale und kommunikationstechnische Modellierung der Smart-Grid-Automatisierungsarchitektur und ihrer Funktionen mit den Mitteln des Standards möglich ist. Dementsprechend wurde der hier realisierte Konfigurationsprozess basierend auf den Mitteln und Datenmodellen des Standards IEC 61850-6 aufgesetzt. Auch wurden die Modellierungsmöglichkeiten, die der Standard bietet, etwa für eine Kommunikation nach IEC 61850-8-1 adaptiert, um weitere, ursprünglich nicht vorgesehene Kommunikationsverfahren zu beschreiben. Durch Nutzung benutzerdefinierter Sprachbestandteile des SCL-Datenmodells und privater Elemente ließen sich darüber hinaus auch alle übrigen für das Smart-Grid-Automatisierungssystem relevanten Konfigurationsdaten modellieren. Insgesamt ergibt sich so der Vorteil, dass sich die Konfiguration des Smart-Grid-Automatisierungssystems nahtlos in den Engineering-prozess gemäß der IEC 61850-6 einfügt. Dies kann erheblich dazu beitragen, die Aufwände für die Konfiguration und das Ausrollen solcher Systeme in zukünftige Smart Grids zu beschleunigen und hebt das System aus bestehenden Lösungen heraus. Ist außerdem der Engineeringprozess nach IEC 61850-6 bereits bei einem Anwender etabliert oder aus höheren Spannungsebenen bekannt, so erhöht das zusätzlich die Akzeptanz der Lösung.

6.3 Ausblick

Die Smart-Grid-Automatisierungsarchitektur ermöglicht es, verschiedene Funktionen nebeneinander auszuführen. Allerdings werden die Funktionen bisher, mit Ausnahme rechenintensiver Funktionen wie der Netzzustandsschätzung, die bereits Mechanismen des softwareseitigen Multithreadings nutzt, zyklisch innerhalb einer Schleife ausgeführt. Mit zunehmender Komplexität und Anzahl an Funktionen erhöht sich so allerdings die Durchlaufzeit der Schleife. Dies ist zu berücksichtigen, sobald eine wachsende Anzahl an Funktionen auf einem System ausgeführt werden soll oder die Laufzeit einzelner Funktionen, etwa einer Regelungsfunktion, besonders zeitkritisch ist. Des Weiteren kann es zukünftig auch gewünscht sein, einzelne Smart-Grid-Funktionen beenden, aktualisieren (neu ausrollen) oder neu starten zu können, ohne andere Funktionen zu beeinflussen. So soll beispielsweise eine Schutzfunktion oder Spannungsregelungsfunktion weiterlaufen, während eine Netzzustandsschätzung mit einer geänderten Topologie neu initialisiert wird. Auch kann es in zukünftigen, komplexen Einsatzszenarien notwendig werden, Funktionen dynamisch zwischen verschiedenen Ausführungsumgebungen zu verschieben. Ein Lösungsansatz hierfür liegt in der Containerisierung der Funktionen, d. h. jede Funktion wird im Sinne einer Abstraktion auf Betriebssystemebene als eigenständige Applikation in einer eigenen

Laufzeitumgebung ausgeführt. Dies seinerseits führt zu neuen Herausforderungen, die berücksichtigt werden müssen, etwa der Notwendigkeit nach einem synchronisierten Datenaustausch zwischen Funktionen oder auch zu Laufzeiteinbußen durch die Abstraktion selbst.

Der realisierte Konfigurationsprozess ermöglicht es, unter Nutzung der normativen Elemente der IEC 61850-6 zu beschreiben, welche Funktionen auf einem Smart-Grid-Automatisierungssystem ablaufen, welche Daten zwischen Funktionen und Systemen ausgetauscht und welche Kommunikationsverfahren hierzu genutzt werden. Im aktuellen Implementierungsstand allerdings benötigen einzelne Funktionen darüber hinausgehende Daten, die separat bereitgestellt werden müssen, etwa Topologieinformationen für die Netzzustandsschätzung. Hier sollten Wege gefunden werden, eine weitere Vereinheitlichung und Zusammenführung voranzutreiben, um Schnittstellen und Medienbrüche zu reduzieren oder idealerweise ganz zu vermeiden. Während sich SCL-Datenmodelle vorwiegend auf die Beschreibung von Sekundärtechnik und Kommunikation konzentrieren, bieten CIM-Datenmodelle Mittel, um Netztopologie und Primärtechnik zu beschreiben. SCL und CIM weisen prinzipiell große Gemeinsamkeiten auf, allerdings sind diese Datenmodelle bisher kaum harmonisiert, so dass klar definierte Schnittstellen fehlen. Auch wurden mit den neusten Editionen des Standards IEC 61850-6 die SCL-Datenmodelle um Sprachmittel zur Modellierung der Primärtechnik ergänzt. Somit stehen zwei Wege zur Verfügung, um die weitergehenden, notwendigen Informationen in standardisierten, wohldefinierten Formaten zu beschreiben.

Der hier vorgestellte Konfigurationsprozess stellt allerdings nur einen Schritt im übergeordneten Engineeringprozess dar. Denn Projektierung und Engineering sind deutlich weiter gefasst. Hierzu gehören etwa auch vorbereitende Analysen der Netze, welche mit intelligenter Technik ausgestattet werden sollen. Grundsätzlich muss erst einmal definiert werden, welche Automatisierungssysteme mit welchen Funktionen an welchen Netzknoten eingesetzt werden sollen. Während im Rahmen dieser Arbeit die Analysen überwiegend manuell erfolgten, werden für optimale Ergebnisse automatisierte Verfahren notwendig. Erste Betrachtungen hierzu erfolgten bereits in [116, 135, 136].

Zur flächendeckenden Umsetzung von Smart Grids muss allein aufgrund der Anzahl und räumlichen Ausdehnung der Verteilnetze eine Vielzahl an Komponenten in die Netze verbracht werden. Um diese Aufgabe meistern zu können, wird eine Standardisierung der Ausroll-, Wartung-, Update- und Betriebsprozesse unerlässlich sein. Das Forschungsprojekt *i-Autonomous „Standardisierung und Integration modular-*

autonomer Automatisierungskomponenten in neuartige, intelligente Ortsnetzstationen“ [B20] befasst sich intensiv mit diesen Themen.

Da Smart-Grid-Automatisierungssysteme ihren Einsatz in Verteilnetzen und damit in der kritischen Infrastruktur finden, müssen sie besonderen Anforderungen genügen. So ist aus funktionaler Sicht die Robustheit und Resilienz aller Systeme und Funktionen beim Betrieb unter realen Feldbedingungen zu gewährleisten. Schutzfunktionen müssen unter allen möglichen Fehlerbedingungen ordnungsgemäß arbeiten. Regelungsfunktionen, die etwa mit entfernten Messwerten arbeiten, benötigen Backupstrategien, um Kommunikationsstörungen und -unterbrechungen zu widerstehen. Für Komponenten, deren Ausfall oder Störung den sicheren Netzbetrieb gefährdet, müssen gegebenenfalls Redundanzen vorgesehen werden. Da die zukünftigen Systeme nicht mehr nur lokal arbeiten, sondern miteinander vernetzt sind und interagieren, kommt Aspekten der Cyber-Security und der Informationssicherheit (engl.: Information Security Management System, ISMS) eine entscheidende Bedeutung zu.

Um sicherzustellen, dass alle Systeme die gewünschte Funktionalität aufweisen und ordnungsgemäß in der Zielanwendung arbeiten, sind für alle Phasen, von der Entwicklung einzelner Komponenten und Funktionen, über das Engineering und die Konfiguration ganzer Systeme bis hin zur Installation und Inbetriebnahme, iterative Prüfverfahren zu entwickeln und auszuführen. Während beispielsweise in dieser Arbeit noch konventionelle Schutzprüfungen zur Verifikation des implementierten Distanzschutzes zum Einsatz kamen, können zukünftig anwendungsorientierte Schutzprüfungen helfen, durch eine realitätsnahe Simulation von Fehlerereignissen nachzuweisen, dass Schutzalgorithmen auch solch herausfordernden Szenarien gewachsen sind. Im Rahmen verschiedener Feldtests zeigte sich bereits, wie schwierig es ist, die ordnungsgemäße Installation und Konfiguration solcher Systeme sicherzustellen. So sind auf den ersten Blick triviale Fehler wie Verdrahtungsfehler bei der Messwerterfassung oder Konfigurationsfehler in der Zuordnung von Datenpunkten im finalen Betrieb nur schwer aufzudecken oder bleiben gar unentdeckt. Umso wichtiger ist es, das Ausrollen und die Inbetriebnahme der Systeme durch ausgeklügelte und mehrstufige Prüfprozesse lückenlos zu begleiten.

Abkürzungsverzeichnis

| | |
|---------------|--|
| ACSI | Abstrakte Schnittstelle für Kommunikationsdienste (engl.: Abstract Communication Service Interface) |
| AK | Arbeitskreis |
| AMD | Änderung (engl.: Amendment, Begriff in der Normung) |
| AngDir | Winkel des gerichteten Bereichs (engl.: Angle for Directional Area, Distanzschutzparameter) |
| AngLod | Winkel des Lastbereichs (engl.: Angle for Load Area, Distanzschutzparameter) |
| AngNegRes | Negativer Winkel des gerichteten Bereichs (engl.: Negative Angle for Directional Area, Distanzschutzparameter) |
| API | Anwendungsprogrammierschnittstelle (engl.: Application Programming Interface) |
| ARM | Advanced RISC (Reduced Instruction Set Computer) Machine (Prozessorarchitektur) |
| ATCC | Transformatorstufensteuerung (engl.: Automatic Tap Changer Controller, Logische-Knoten-Klasse) |
| AVCO | Spannungsregelung (engl.: Voltage Control, Logische-Knoten-Klasse) |
| Baud | Symbolrate |
| BI | Binärer Eingang (engl.: Binary Input) |
| BIOS | Basic Input/Output System (Systemfirmware eines Computers) |
| BIS | Betriebsmittelinformationssystem |
| BL | Allgemeine Anwendungslogik (engl.: Business Logic, Ebene der Softwarearchitektur) |
| BO | Binärer Ausgang (engl.: Binary Output) |
| Bool, Boolean | Logischer Zustand (Datentyp) |
| CAD | Rechnerunterstütztes Konstruieren (engl.: Computer-Aided Design) |
| CDC | Gemeinsame Datenklassen (engl.: Common Data Classes) |
| CID | Beschreibung eines konfigurierten IEDs (engl.: Configured IED Description, SCL-Datenaspekt) |
| CIM | Common Information Model (Datenmodell) |
| CO | Befehl (engl.: Command) |
| CSV | Konsolidierte Fassung (engl.: Consolidated Version, Begriff in der Normung) |

| | |
|-------|---|
| CSWI | Schaltgerätesteuerung (engl.: Switch Controller, Logische-Knoten-Klasse) |
| DA | Datenattribut (engl.: Data Attribute) |
| DAI | Instanziiertes Datenattribut (engl.: Data Attribute Instance) |
| DAQ | Datenerfassung (engl.: Data Acquisition) |
| DCF77 | Zeitzeichensender |
| DIN | Deutsches Institut für Normung e. V. |
| DKE | Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE |
| DO | Datenobjekt (engl.: Data Object) |
| DOI | Instanziiertes Datenobjekt (engl.: Data Object Instance) |
| DSL | Digitaler Teilnehmeranschluss (engl.: Digital Subscriber Line) |
| DSP | Digitaler Signalprozessor |
| DT | Device Tree (Datenstruktur) |
| EK | Ersatzknoten |
| ESB | Ersatzschaltbild |
| FC | Funktionale Einschränkung (engl.: Functional Constraint) |
| FLKR | Flicker (Baugruppe) |
| Float | Gleitkommazahl (Datentyp) |
| FPGA | Im Feld programmierbare Logikgatter-Anordnung (engl.: Field Programmable Gate Array) |
| FPID | PID-Regler (engl.: Function Proportional Integral Derivative, Logische-Knoten-Klasse) |
| FSPT | Funktion zum Setzen eines Sollwerts (engl.: Set-Point Control Function, Logische-Knoten-Klasse) |
| GAPC | Generische automatische Prozesssteuerung (engl.: Generic Automatic Process Control, Logische-Knoten-Klasse) |
| GIS | Geografisches Informationssystem |
| GOOSE | Generic Object Oriented Substation Events (Kommunikationsprotokoll) |
| GPIO | Universeller Eingang/Ausgang (engl.: General Purpose Input/Output) |
| GPS | Globales Positionsbestimmungssystem (engl.: Global Positioning System, Globales Navigationssatellitensystem) |

| | |
|-----------------|---|
| HAL | Abstraktion von der Systemplattform (engl.: Hardware Abstraction Layer, Ebene der Softwarearchitektur) |
| HöS | Höchstspannung |
| HS | Hochspannung |
| ICD | Beschreibung der IED-Fähigkeiten (engl.: IED Capability Description, SCL-Datenaspekt) |
| IDE | Integrierte Entwicklungsumgebung (engl.: Integrated Development Environment) |
| ie ³ | Institut für Energiesysteme, Energieeffizienz und Energiewirtschaft |
| IEC | Internationale elektrotechnische Kommission (engl.: International Electrotechnical Commission) |
| IED | Intelligentes elektronisches Gerät (engl.: Intelligent Electronic Device) |
| IID | Beschreibung eines instanziierten IEDs (engl.: Instantiated IED Description, SCL-Datenaspekt) |
| Integer | Ganzzahl (Datentyp) |
| IO | Eingang/Ausgang (engl.: Input/Output) |
| IP | Internetprotokoll (engl.: Internet Protocol) |
| IPsec | Sammlung von Standards zur gesicherten Kommunikation in IP-Netzwerken (engl.: Internet Protocol Security) |
| IRIG-B | Inter Range Instrumentation Group Type B (Zeitcode) |
| ISMS | Managementsystem für Informationssicherheit (engl.: Information Security Management System) |
| ISO | Internationale Organisation für Normung (engl.: International Organization for Standardization) |
| ITCI | Fernsteuerschnittstelle (engl.: Telecontrol Interface, Logische-Knoten-Klasse) |
| KVS | Kabelverteilerschrank |
| L1, L2, L3 | Außenleiter |
| LD | Logisches Gerät (engl.: Logical Device) |
| LinAng | Leitungswinkel (engl.: Line Angle, Distanzschutzparameter) |
| LN | Logischer Knoten (engl.: Logical Node) |
| LN0 | Logical Node Zero (Logische-Knoten-Klasse) |
| LNnode | Abstrakter logischer Knoten (engl.: Abstract Logical Node) |
| LPHD | Physical Device Information (Logische-Knoten-Klasse) |

| | |
|--------------|--|
| MAC | Media Access Layer (Kommunikationsschicht) |
| MHO | Veraltete Bezeichnung für die Maßeinheit des elektrischen Leitwerts (Siemens) |
| MiB | Mebibyte ($1.024 \cdot 1.024 = 1.048.576$ Bytes) |
| ML | Maschinelles Lernen (engl.: Machine Learning) |
| MMS | Manufacturing Messaging Specification (Kommunikationsprotokoll) |
| MMXU | Dreiphasige Messung (engl.: Three-Phase Measurement, Logische-Knoten-Klasse) |
| MON | Monitoring (Baugruppe) |
| MPC | Modellprädiktive Regelung (engl.: Model Predictive Control) |
| MS | Mittelspannung |
| MX | Messwert (engl.: Measurement) |
| N | Neutralleiter |
| NH-Sicherung | Niederspannungs-Hochleistungs-Sicherung |
| NS | Niederspannung |
| ONS | Ortsnetzstation |
| ONT | Ortsnetztransformator |
| OpDITmms | Auslöseverzögerung (engl.: Operate Delay Time, Schutzparameter) |
| OPF | Leistungsflussoptimierung (engl.: Optimal Power Flow) |
| OSI | Open Systems Interconnection (Referenzmodell für Netzwerkprotokolle) |
| PCI | Peripheral Component Interconnect (Computerbus) |
| PDF | Portable Document Format (Dateiformat) |
| PDIS | Distanzschutz (engl.: Distance Protection, Logische-Knoten-Klasse) |
| PLC | Datenkommunikation über das elektrische Energienetz (engl.: Powerline Communication) |
| PMU | Zeitsynchronisiertes Zeigermessgeräte (engl.: Phasor Measurement Unit) |
| PTOC | Überstromzeitschutz (engl.: Time-Overcurrent Protection, Logische-Knoten-Klasse) |
| PTRC | Auslösekoordination (engl.: Trip Conditioning, Logische-Knoten-Klasse) |
| pu | Per-Unit-System |
| PV | Photovoltaik |

| | |
|----------------|--|
| RFC | Bitte um Kommentare (engl.: Requests for Comments) |
| RisLod | Ohmsche Reichweite des Lastbereichs (engl.: Resistive Reach for Load Area, Distanzschutzparameter) |
| RisPhRch | Ohmsche Phasenreichweite (engl.: Resistive Phase Reach, Distanzschutzparameter) |
| RMS | Effektivwert (engl.: Root Mean Square) |
| RPMsg | Remote Processor Messaging (Kommunikationsverfahren) |
| RS | Empfohlener Standard (engl.: Recommended Standard) |
| RsDITmms | Rücksetzverzögerung (engl.: Reset Delay Time, Schutzparameter) |
| RTOS | Echtzeitbetriebssystem (engl.: Real-Time Operating System) |
| RTU | Fernbedienungsterminal (engl.: Remote Terminal Unit) |
| SAS | Stationsautomatisierungssystem |
| SCD | Beschreibung der Systemkonfiguration (engl.: System Configuration Description, SCL-Datenaspekt) |
| SCL | System Configuration description Language (Datenmodell) |
| SCSM | Abbildung auf spezifische Kommunikationsdienste (engl.: Specific Communication Service Mapping) |
| SCT | Substation Configuration Tool (Softwarewerkzeug) |
| SDI | Instanziertes, untergeordnetes Datenobjekt (engl.: Sub Data object Instance) |
| SDO | Untergeordnetes Datenobjekt (engl.: Sub Data Object) |
| SE | Netzzustandsschätzung (engl.: State Estimation) |
| SH | Schaltheis |
| SIGINT | Unterbrechungsanforderung (engl.: Signal Interrupt, Systemnachricht eines Betriebssystems) |
| SIGTERM | Beendigungsanfrage (engl.: Termination Request, Systemnachricht eines Betriebssystems) |
| SLC NAND-Flash | Typ eines Datenspeichers |
| SoM | Einplatinencomputer (engl.: System-on-Module) |
| SPS | Speicherprogrammierbare Steuerung |
| SQL | Datenbanksprache |
| SSD | Beschreibung der Systemspezifikation (engl.: System Specification Description, SCL-Datenaspekt) |
| ST | Zustand (engl.: Status) |

| | |
|--------|---|
| StrVal | Anregewert (engl.: Start Value, Überstromzeitschutzparameter) |
| SV | Abgetastete Werte (engl.: Sampled Values, Kommunikationsprotokoll) |
| Tcmd | Transformatorbefehl (engl.: Transformer Command) |
| TCP | Transmission Control Protocol (Netzwerkprotokoll) |
| TCTR | Stromwandler (engl.: Current Transformer, Logische-Knoten-Klasse) |
| TIME | Zeitsynchronisation (Baugruppe) |
| TRC | Transient Recorder (Baugruppe) |
| TRGU | Trigger Unit (Baugruppe) |
| TVTR | Spannungswandler (engl.: Voltage Transformer, Logische-Knoten-Klasse) |
| UDP | User Datagram Protocol (Netzwerkprotokoll) |
| UEFI | Unified Extensible Firmware Interface (Systemfirmware eines Computers) |
| USART | Serielle Schnittstelle (engl.: Universal Synchronous/Asynchronous Receiver Transmitter) |
| UW | Umspannwerk |
| VDE | Verband der Elektrotechnik Elektronik Informationstechnik e. V. |
| VPN | Virtuelles privates Netzwerk (engl.: Virtual Private Network) |
| WK | Windkraft |
| WLS | Gewichtete kleinste Quadrate (engl.: Weighted Least Squares, Regressionsverfahren) |
| WP | Windpark |
| X1 | Mitsystem-Reichweite der Leitungsreaktanz (engl.: Positive Sequence Line (Reach) Reactance, Distanzschutzparameter) |
| XML | Erweiterbare Auszeichnungssprache (engl.: eXtensible Markup Language, Datenmodell) |

Formelverzeichnis

Allgemeine Notation

| | |
|-----------------|----------------|
| U | Effektivwert |
| \underline{U} | Komplexer Wert |
| u | Momentanwert |

Symbole

| Symbol | Beschreibung | Einheit |
|--------|-----------------------------|----------|
| f | Frequenz | Hz |
| $h(x)$ | Messfunktion | |
| $H(x)$ | Jakobi-Matrix | |
| I | Strom | A |
| i, j | Knotenvariable | |
| $J(x)$ | Zielfunktion | |
| l | Länge | m |
| P | Wirkleistung | W |
| Q | Blindleistung | var |
| R | Wirkwiderstand (Resistanz) | Ω |
| S | Scheinleistung | VA |
| t | Zeit | s |
| U | Spannung | V |
| W | Gewichtungsmatrix | |
| x | Zustandsvektor | |
| X | Blindwiderstand (Reaktanz) | Ω |
| z | Messvektor | |
| Z | Scheinwiderstand (Impedanz) | Ω |

Literaturverzeichnis

- [1] Ahle, Lothar; Maaßen, Katharina; Niehaus, Milan; Niehaus, Rainer; Steinert, Sarah: *Designnetz: Lösungen. Strategien. Impulse. Für das Energiesystem der Zukunft*. Band 1.
- [2] Jendernalik, Lars: *Moderne Ansätze der Netzplanung bei Westnetz*. Smart Energy 2014, Dortmund, 27.11.2014. <https://www.smart-systems-conference.de/wp-content/uploads/2014/S2/Jendernalik.pdf>
- [3] Deutsche Energie-Agentur GmbH (Hrsg.): *Ausbau- und Innovationsbedarf der Stromverteilnetze in Deutschland bis 2030 (kurz: dena-Verteilnetzstudie)*. Berlin, 11.12.2012. <https://www.dena.de/themen-projekte/projekte/energiesysteme/dena-verteilstudie>
- [4] Deutsche Energie-Agentur GmbH (Hrsg.): *dena-Studie Systemdienstleistungen 2030: Voraussetzungen für eine sichere und zuverlässige Stromversorgung mit hohem Anteil erneuerbarer Energien*. Berlin, 11.02.2014. <https://www.dena.de/themen-projekte/projekte/energiesysteme/dena-studie-systemdienstleistungen-2030>
- [5] E-Bridge Consulting GmbH (Hrsg.): *Moderne Verteilernetze für Deutschland (Verteilernetzstudie). Forschungsprojekt Nr. 44/12*. Studie im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi). Bonn, 12.09.2014. <https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/verteilstudie.html>
- [6] Ministerium für Wirtschaft Innovation, Digitalisierung und Energie des Landes Nordrhein-Westfalen (Hrsg.): *Verteilnetzstudie NRW. Gutachten zur Weiterentwicklung der Strom-Verteilnetze in Nordrhein-Westfalen auf Grund einer fortschreitenden Sektorenkopplung und neuer Verbraucher*. Dortmund, 09.06.2021. <https://www.wirtschaft.nrw/pressemitteilung/verteilstudie-nrw>
- [7] Nolte, Frank: *Intelligente Ortsnetzstationen stabilisieren die Energieversorgung*. *etz elektrotechnik & automation*, Jg. 133, S2, S. 23–25, 2012
- [8] Neuhoff, Andrea: *Energieflüsse dezentral steuern*. *Energy 2.0*, Nr. 6, S. 38–40, 2016. http://www.ipi-online.de/wp-content/uploads/2013/03/Energy20_IPI_Pfeffer_Interview.pdf
- [9] *Communication networks and systems for power utility automation. Configuration description language for communication in electrical substations related to IEDs*. IEC 61850-6. International Electrotechnical Commission, 2009
- [10] Hilbrich, Dominik: *Universelles Prüfsystem für innovative Schutz- und Automatisierungssysteme in Smart Grids. Dissertation*. 1. Auflage, Düren: Shaker Verlag, 2021

- [11] *Communication networks and systems for power utility automation. Specific communication service mapping (SCSM). Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3.* IEC 61850-8-1. International Electrotechnical Commission, 2011
- [12] Verband der Netzbetreiber (VDN) e. V. beim VDEW (Hrsg.): *IEC 61850 – Anforderungen aus Anwendersicht.* 07.2004
- [13] Antoniadou-Plytaria, Kyriaki E.; Kouveliotis-Lysikatos, Iasonas N.; Georgilakis, Pavlos S.; Hatziargyriou, Nikos D.: *Distributed and Decentralized Voltage Control of Smart Distribution Networks: Models, Methods, and Future Research.* IEEE Transactions on Smart Grid, Jg. 8, Nr. 6, S. 2999–3008, 2017, doi: 10.1109/TSG.2017.2679238
- [14] Schossig, Walter; Schossig, Thomas: *Netzschutztechnik.* 4. Auflage, Berlin: VDE Verlag GmbH, 2013
- [15] 450connect GmbH: *450connect: Ihre Lösung für sichere Kommunikation.* <https://www.450connect.de>
- [16] *Telecontrol equipment and systems. Transmission protocols. Network access for IEC 60870-5-101 using standard transport profiles.* IEC 60870-5-104. International Electrotechnical Commission, 2006
- [17] *Communication networks and systems for power utility automation. Specific communication service mapping (SCSM). Sampled values over ISO/IEC 8802-3.* IEC 61850-9-2. International Electrotechnical Commission, 2011
- [18] *The Modbus Organization.* <https://modbus.org>
- [19] *Industrial communication networks. Fieldbus specifications. Overview and guidance for the IEC 61158 and IEC 61784 series.* IEC 61158-1. International Electrotechnical Commission, 2019
- [20] *Industrial communication networks. Profiles. Fieldbus profiles. CPF15 Modbus, CPF15/1 Modbus/TCP.* IEC 61784-1. International Electrotechnical Commission, 2019
- [21] *MQTT: The Standard for IoT Messaging.* <https://mqtt.org>
- [22] Brunner, Helfried: *DG DemoNet – Smart LV Grid: Erkenntnisse für die Integration dezentraler Erzeugung.* 2015. https://www.ait.ac.at/fileadmin/mc/energy/downloads/Smart_Grids/Projects/DG_Demonet_Smart_Grid.pdf
- [23] Brunner, Helfried: *DG DemoNet – Smart LV Grid: Control concepts for active low voltage network operation with a high share of distributed energy resources – Endbericht.* Wien, 30.04.2015
- [24] *Grid4EU.* <https://ses.jrc.ec.europa.eu/grid4eu>
- [25] Jendernalik, Lars; Wiedemann, Thomas; Noglik, Peter; Shapovalov, Anton: *Grid4EU: Erfahrungen mit einem autonom schaltenden System in der Mittelspannung.* TU Graz, Institut für Elektrizitätswirtschaft und

- Energieinnovation (Hrsg.): 14. Symposium Energieinnovation, Graz, Österreich, 10.-12.02.2016
- [26] Kreusel, Jochen: *GRID4EU optimises the medium voltage grid*. Renewable Energy Focus, Jg. 17, Nr. 5, S. 194–197, 2016, doi: 10.1016/j.ref.2016.08.011
- [27] Jendernalik, Lars; Wiedemann, Thomas; Noglik, Peter; Shapovalov, Anton: *The German large scale demonstration project inside GRID4EU: Challenges of an autonomous Medium Voltage control system*. Institution of Engineering and Technology (Hrsg.): 23rd International Conference on Electricity Distribution, Lyon, Frankreich, 15.-18.06.2015
- [28] Brenzikofer, Alain; Eisenreich, Marc; Mangani, Marco; Bacher, Rainer: *GridBox Pilotnetz - Potential eines Echtzeit-Management-Systems für das Verteilnetz. Schlussbericht*. Zürich, 31.03.2016
- [29] Moser, Stephan: *Monitoring- und Management-Plattform für das Verteilnetz. Zwei Demonstrations-Regionen seit Juni 2015 in Betrieb*. Bulletin SEV/VSE, Nr. 9, S. 21–24, 2015
- [30] Brenzikofer, Alain: *GridBox. Smart-Grid-Pilotnetz ist in Betrieb*. ENERGY.NOW!, S. 26–27, 2015
- [31] Mangani, Marco; Kienzle, Florian; Eisenreich, Marc; Farhat Quinones, Yamshid; Bacher, Rainer; Brenzikofer, Alain: *GridBox – An open platform for monitoring and active control of distribution grids*. Institution of Engineering and Technology (Hrsg.): 23rd International Conference on Electricity Distribution, Lyon, Frankreich, 15.-18.06.2015
- [32] Moser, Stephan: *Gridbox-Technologie. Intelligente Monitoring- und Managementplattform für das Verteilungsnetz*. ew Spezial, Nr. 2, S. 12–15, 2016
- [33] Brenzikofer, Alain; Müller, Florian; Ketsetzis, Alexandros; Kienzle, Florian; Mangani, Marco; Eisenreich, Marc; Farhat, Yamshid; Bacher, Rainer: *GridBox pilot project. A platform for monitoring and active control of distribution grids*. Comput Sci Res Dev, Jg. 32, 1-2, S. 79–92, 2017, doi: 10.1007/s00450-016-0308-5
- [34] EVUtion: *GRIDCONTROL: Echtzeit-Plattform für das Netzmanagement*. <https://evulation.com/produkte/gridcontrol>
- [35] PSInsight GmbH: *GridCal*. <https://www.gridcal.com>
- [36] PSInsight GmbH: *Verteilnetz digital. Der digitale mitwachsende Maßanzug für das Verteilnetz*. https://www.gridcal.com/wp-content/uploads/2021/09/psinsight_gridcal_folder_web.pdf
- [37] Heinrichs, Martin: *Digitales Verteilnetz: Effizienz ist das entscheidende Stellglied für Netzbetreiber*. <https://www.energie.de/ew/news-detailansicht/nsctrl/detail/News/digitales-verteilnetz-effizienz-ist-das-entscheidende-stellglied-fuer-netz-betreiber>

- [38] DEPSys SA: *Smart grid solutions for simpler energy transitions*. <https://www.depsys.com>
- [39] DEPSys SA (Hrsg.): *GridEye: Für ein sicheres, geregeltes und effizientes Stromnetz. Produktbroschüre*. 29.03.2018
- [40] DEPSys SA (Hrsg.): *GridEye: Network Supervision - Use Case*. 27.09.2017
- [41] *iNES - intelligentes Verteilnetzmanagement-System*. <https://www.evt.uni-wuppertal.de/de/forschung/forschungsgruppe-intelligente-netze-und-systeme/ines-intelligentes-verteilnetzmanagement>
- [42] Uhlig, Roman; Neusel-Lange, Nils; Zdrallek, Markus: *Smart distribution grids for Germany's Energiewende*. J. Drápela (Hrsg.): 2014 15th International Scientific Conference on Electric Power Engineering (EPE), Brno-Bystrc, Czech Republic, 2014, S. 121–124, doi: 10.1109/EPE.2014.6839543
- [43] Bilfinger Mauell GmbH (Hrsg.): *Systemlösung „Intelligente Ortsnetzstation“ iNES*. 18.03.2014
- [44] Neusel-Lange, Nils; Oerter, Christian; Zdrallek, Markus: *State identification and automatic control of smart low voltage grids*: 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), Berlin, Germany, 2012, S. 1–6, doi: 10.1109/ISGTEurope.2012.6465838
- [45] *Mauell GmbH*. <https://www.mauell.de>
- [46] *NiVeAu - Netzintelligenz für die Verteilnetzautomatisierung*. <https://www.evt.uni-wuppertal.de/de/forschung/forschungsgruppe-intelligente-netze-und-systeme/niveau-netzintelligenz-fuer-die-verteilnetzautomatisierung>
- [47] *Grid Commander - Vernetzte, teilautarke Steuerung von Mittel- und Niederspannungsnetzen*. <https://www.evt.uni-wuppertal.de/de/forschung/forschungsgruppe-intelligente-netze-und-systeme/grid-commander-spannungsebenenuebergreifende-verteilnetzautomatisierung>
- [48] Korotkiewicz, Kamil; Zdrallek, Markus; Vormelker, Jens; Friedrich, Wolfgang; Plötz, Sigrid; Hetzel, Steffen: *Grid Commander - Vernetzte, teilautarke Steuerung von Mittel- und Niederspannungsnetzen : gemeinsamer Schlussbericht*. 26.03.2019
- [49] PSI GridConnect GmbH: *Intelligent Grid Operator - InGO. Stromnetze digitalisieren und sicher zukunftsfähig betreiben*. https://www.psigridconnect.com/fileadmin/files/downloads/PSI_GridConnect/marketing/Intelligent_Grid_Operator-InGO_de.pdf
- [50] Kubis, Andreas; Boller, Markus; Kemper, Julian; Uhlig, Roman; Stiegler, Martin; Stötzel, Marcus: *Enhancing Operational Awareness of Distribution System Operators with a Semi-Autonomous Intelligent Grid Control Suite*. Slovenian Committee of Electric Power Engineers CIGRE-CIRED (Hrsg.): 25th International

- Conference on Electricity Distribution, Madrid, Spanien, 03.-06.06.2019, doi: 10.34890/834
- [51] Uhlig, Roman; Stoetzel, Marcus; Stiegler, Martin; Lamberth, Sven; Kubis, Andreas: *Hybrid Cascaded Operation of Distribution Grids*. VDE ETG (Hrsg.): ETG Congress 2019, Esslingen am Neckar, 08.-09.05.2019
- [52] PSI GridConnect GmbH: *PGC-3000*. <https://www.psigridconnect.com/de/plattformen/pgc-3000-hutschiene>
- [53] OPC Foundation: *Unified Architecture*. <https://opcfoundation.org/about/opc-technologies/opc-ua>
- [54] Gridspertise: *Quantum Edge device*. <https://www.gridspertise.com/solutions/qed>
- [55] *Enel unveils Gridspertise, the company dedicated to the digital transformation of power grids*. <https://www.enel.com/media/explore/search-press-releases/press/2021/09/enel-unveils-gridspertise-the-company-dedicated-to-the-digital-transformation-of-power-grids->
- [56] *Smart Area Aachen. Netze für die Stromversorgung der Zukunft: Aufbau eines intelligenten Stromnetzes im Stadtgebiet Aachen durch Komponenten der Energie-, Informations- und Kommunikationstechnik*. <https://web.archive.org/web/20221007061949/http://smartarea.de>
- [57] Keune, Björn; Rehtanz, Christian; Aschenbroich, Ulrich: *Verbundprojekt: Intelligente Ortsnetzstation - Feldtest und Praxiserprobung von i3S : Abschlussbericht zum Verbundvorhaben Smart Area Aachen*. Dortmund, 07.04.2017
- [58] ABB AG Calor Emag Mittelspannungsprodukte, Oberhausener Straße 33, 40472 Ratingen, Deutschland: *FIONA Flexible intelligente Ortsnetzautomatisierung. Die Lösung für bestehende und neue Ortsnetzstationen*. https://library.e.abb.com/public/da8bbf44b0d1b339c1257b4f003e7361/2494%20FIONA_22032013.pdf
- [59] *SMIGHT Grid2: Smarte Sensorik für Ihr Verteilnetz*. <https://smight.com/smight-grid2>
- [60] *EnBW gründet Start-up für Smart Grid-Lösungen*. <https://www.enbw.com/unternehmen/presse/daten-sorgen-fuer-schlaue-netze.html>
- [61] Romano, Paolo: *SynchroGuard. Unlock your grid potential*. Produktpräsentation. 10.02.2020
- [62] Zaphiro Technologies SA: *Smart Grid Solutions*. <https://zaphiro.ch>
- [63] Pignati, Marco; Zanni, Lorenzo; Romano, Paolo; Cherkaoui, Rachid; Paolone, Mario: *Fault Detection and Faulted Line Identification in Active Distribution Networks Using Synchrophasors-Based Real-Time State Estimation*. IEEE Transactions on Power Delivery, Jg. 32, Nr. 1, S. 381–392, 2017, doi: 10.1109/TPWRD.2016.2545923

- [64] Zaphiro Technologies SA: *SynchroSense specifications*. 13.02.2020
- [65] Pignati, Marco; Popovic, Miroslav; Barreto, Sergio; Cherkaoui, Rachid; Dario Flores, German; Le Boudec, Jean-Yves; Mohiuddin, Maaz; Paolone, Mario; Romano, Paolo; Sarri, Styliani; Tesfay, Teklemariam; Tomozei, Dan-Cristian; Zanni, Lorenzo: *Real-time state estimation of the EPFL-campus medium-voltage grid by using PMUs*: 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2015, S. 1–5, doi: 10.1109/ISGT.2015.7131877
- [66] H & S Hard- & Software Technologie GmbH & Co. KG (Hrsg.): *Studie zur Integration des herstellerneutralen Engineeringansatzes gemäß IEC 61850-6 in den bestehenden Engineeringprozess der Sekundärtechnik von Höchstspannungsschaltanlagen der Amprion GmbH*. Studie im Auftrag der Amprion GmbH, Dortmund [Internes Dokument]. Dortmund, 17.11.2010
- [67] H & S Hard- & Software Technologie GmbH & Co. KG (Hrsg.): *Lastenheft für ein Engineeringwerkzeug für Sekundärtechnik in Hochspannungsschaltanlagen*. Lastenheft im Auftrag der Amprion GmbH, Dortmund [Internes Dokument]. Dortmund, 11.07.2011
- [68] VertiGIS GmbH. <https://www.aed-sicad.de>
- [69] Lovion GmbH. <https://lovion.de>
- [70] DlgSILENT GmbH: *PowerFactory*. <https://www.digsilent.de/de/powerfactory.html>
- [71] SAE IT-systems GmbH & Co. KG. <https://www.sae-it.de>
- [72] VIVAVIS AG. <https://www.vivavis.com>
- [73] VIVAVIS AG: *VIVAVIS bündelt Aktivitäten für Energieversorger*. 27.06.2022. <https://www.vivavis.com/die-unternehmensgruppe-vivavis-buendelt-ihre-aktivitaeten-fuer-energieversorger-in-der-neuen-vivavis-ag>
- [74] WAGO GmbH & Co. KG. <https://www.wago.com/de>
- [75] Siemens AG: *SIPROTEC 5*. <https://www.siemens.com/de/de/produkte/energie/energieautomatisierung-und-smart-grid/schutztechnik/siprotec-5.html>
- [76] Siemens AG: *DIGSI 5*. <https://www.siemens.com/de/de/produkte/energie/energieautomatisierung-und-smart-grid/schutztechnik/engineering-tools-fuer-schutztechnik/engineering-software-digsi-5.html>
- [77] PSI Software AG: *Netzleittechnik*. <https://www.psienergy.de/de/loesungen/netzleittechnik>
- [78] EPLAN GmbH & Co. KG: *EPLAN Electric P8*. <https://www.eplan.de/loesungen/eplan-electric-p8>
- [79] *Communication networks and systems for power utility automation. Basic communication structure. Compatible logical node classes and data object classes*. IEC 61850-7-4. International Electrotechnical Commission, 2010

- [80] *Communication networks and systems for power utility automation. Basic communication structure. Common data classes.* IEC 61850-7-3. International Electrotechnical Commission, 2010
- [81] *Communication networks and systems for power utility automation. Basic information and communication structure. Abstract communication service interface (ACSI).* IEC 61850-7-2. International Electrotechnical Commission, 2010
- [82] *Communication networks and systems for power utility automation. Basic communication structure. Hydroelectric power plants. Communication for monitoring and control.* IEC 61850-7-410. International Electrotechnical Commission, 2012
- [83] *Communication networks and systems for power utility automation. Basic communication structure. Distributed energy resources logical nodes.* IEC 61850-7-420. International Electrotechnical Commission, 2009
- [84] *Wind turbines. Communications for monitoring and control of wind power plants. Information models.* IEC 61400-25-2. International Electrotechnical Commission, 2006
- [85] *Wind turbines. Communications for monitoring and control of wind power plants. Mapping to communication profile.* IEC 61400-25-4. International Electrotechnical Commission, 2008
- [86] World Wide Web Consortium: *Extensible Markup Language (XML)*. <https://www.w3.org/XML>
- [87] DKE/AK 952.0.1: *Beschreibung des Engineeringprozesses.* 29.05.2017. <https://www.dke.de/resource/blob/1622842/707315d29d442841faa067f28182ae53/ak952-0-1-engineering-prozess-v10-data.pdf>
- [88] DKE/AK 952.0.1 Teilgruppe 3: *Anforderungen an IEC 61850 Engineeringwerkzeuge.* 29.05.2017. <https://www.dke.de/resource/blob/1622834/762050db66c0406098a48b08913bba2e/ak952-0-1-tg3-engineeringwerkzeuge-1-0-data.pdf>
- [89] Forum Netztechnik / Netzbetrieb im VDE (Hrsg.): *IEC 61850 aus Anwendersicht. Anwenderempfehlung.* 12.2012
- [90] Applied Systems Engineering Inc. (ASE), a Kalkitech Company: *ASE61850 SCL Manager.* <https://ase-systems.com/products/ase61850-relay-test-set/ase61850-scl-manager>
- [91] Helinks LLC: *System Tools For IEC 61850.* <https://www.helinks.com>
- [92] Siemens AG: *IEC 61850 Systemkonfigurator.* <https://www.siemens.com/de/de/produkte/energie/energieautomatisierung-und-smart-grid/schutztechnik/engineering-tools-fuer-schutztechnik/engineering-software-fuer-ic-61850-systeme-iec-61850-system-konfigurator.html>

- [93] Siemens AG: *12. ETG-/FNN-Tutorial Schutz- und Leittechnik, Online-Preview.* Diskussion, 15.02.2022
- [94] H & S Hard- & Software Technologie GmbH & Co. KG: *Substation Configuration Tool (SCT).* https://hs.beimdick-gruppe.de/systemtechnik/iec_61850
- [95] Sprecher Automation GmbH: *Stationsleittechnik.* <https://www.sprecher-automation.com/energie/produkte/stationsleittechnik>
- [96] Sprecher Automation GmbH: *IEC 61850 - Easy Engineering.* 01.09.2020
- [97] Schneider Electric SE: *CET850 IEC 61850 configuration software. Online help / user's manual.* <https://www.se.com/us/en/download/document/SEPED306025> EN
- [98] Schneider Electric SE: *CET850 IEC 61850 configuration tool.* <https://www.se.com/de/de/download/document/CET850+IEC850+configuration+tool>
- [99] *Elektromagnetische Verträglichkeit (EMV). Prüf- und Messverfahren. Verfahren zur Messung der Spannungsqualität.* DIN EN 61000-4-30 VDE 0847-4-30. DIN Deutsches Institut für Normung e. V. und VDE Verband der Elektrotechnik Elektronik Informationstechnik e. V., 2016
- [100] KoCoS Messtechnik AG: *KoCoS Messtechnik AG.* <https://www.kocos.com>
- [101] CompuLab Ltd.: *CL-SOM-iMX7.* <https://www.compulab.com/de/products/computer-on-modules/cl-som-imx7-freescale-i-mx-7-system-on-module>
- [102] *CompuLab Ltd.* <https://www.compulab.com>
- [103] *Buildroot - Making Embedded Linux Easy.* <https://buildroot.org>
- [104] *Information technology. Programming languages. C++.* ISO/IEC 14882. International Organization for Standardization, 2011. <https://www.iso.org/standard/50372.html>
- [105] *Qt Creator. A Cross-platform IDE for Application Development.* <https://www.qt.io/product/development-tools>
- [106] Grant Likely: *Linux and the Devicetree.* <https://www.kernel.org/doc/html/latest/devicetree/usage-model.html>
- [107] Linaro: *The Devicetree Specification.* <https://www.devicetree.org/specifications>
- [108] *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions.* <https://www.freertos.org>
- [109] PetrLukas: *RPMsg Messaging Protocol.* <https://github.com/OpenAMP/open-amp/wiki/RPMsg-Messaging-Protocol>
- [110] The Linux Kernel Organization: *Remote Processor Messaging (rpmsg) Framework.* <https://www.kernel.org/doc/html/latest/staging/rpmsg.html>
- [111] Palaniappan, Rajkumar: *Coordinated voltage control in active distribution grids using distributed field measurements. Dissertation.* 1. Auflage, Düren: Shaker Verlag, 2022

- [112] *Merkmale der Spannung in öffentlichen Elektrizitätsversorgungsnetzen*. DIN EN 50160. DIN Deutsches Institut für Normung e. V., 2011
- [113] Palaniappan, Rajkumar; Irshad, Muhammad Sohaib; Rehtanz, Christian: *Determination of Optimal Flexibility Potential for an Electrical Distribution Network*. TU Graz, Institut für Elektrizitätswirtschaft und Energieinnovation (Hrsg.): 16. Symposium Energieinnovation (EnInnov 2020), Graz, Österreich, 12.-14.02.2020
- [114] Palaniappan, Rajkumar; Molodchyk, Oleksii; Rehtanz, Christian: *Hardware implementation of an optimal power flow algorithm in a distribution network with decentralised measurements*. CIREC - Open Access Proceedings Journal, Jg. 2020, Nr. 1, S. 580–583, 2021, doi: 10.1049/oap-cired.2021.0122
- [115] Palaniappan, Rajkumar; Molodchyk, Oleksii; Rehtanz, Christian: *A hierarchical Optimal Power Flow method for Smart Grids with Field Measurements*. Institution of Engineering and Technology (Hrsg.): CIREC 2021 - The 26th International Conference and Exhibition on Electricity Distribution, Virtuell, 20.-23.09.2021, S. 1341–1345, doi: 10.1049/icp.2021.2171
- [116] Raczka, Sebastian; Hilbrich, Dominik; Rehtanz, Christian: *Identification of relevant local grid stations for a smart grid automation system*. VDE ETG (Hrsg.): ETG Congress 2021, Virtuell, 18.-19.05.2021
- [117] Görner, Kay: *Analyse und Anwendung der zeitsynchronisierten Zeigermessung*. Dissertation. 1. Auflage, Göttingen: Sierke Verlag, 2017
- [118] Schwierz, Thomas; Raczka, Sebastian; Hilbrich, Dominik; Rehtanz, Christian: *Development of a measurement-based algorithm for topology detection in distribution grids*. CIREC (Hrsg.): CIREC workshop on e-mobility and power distribution systems, Porto, Portugal, 02.-03.06.2022, doi: 10.1049/icp.2022.0712
- [119] Raczka, Sebastian; Hilbrich, Dominik; Brüggemann, Annika; Rehtanz, Christian: *A Model Predictive Control Algorithm for large-scale Integration of Electromobility*. Institute of Electrical and Electronics Engineers, Power & Energy Society (Hrsg.): Innovative Smart Grid Technologies Conference Europe (ISGT Europe), Virtuell, 26.-28.10.2020, doi: 10.1109/ISGT-Europe47291.2020.9248960
- [120] Palaniappan, Rajkumar; Ceschlaw, Jan-Niklas; Karagiannopoulos, Stavros; Rehtanz, Christian: *Implementation of Machine Learning-based DER Local Control Schemes on Measurement Devices for Counteracting Communication Failures*. International Institute for Research and Education in Power Systems (Hrsg.): 11th Bulk Power Systems Dynamics and Control Symposium, Banff, Kanada, 25.-30.07.2022, doi: 10.48550/arXiv.2207.08732
- [121] Puhe, Frederik; Schwierz, Thomas; Rehtanz, Christian: *Subgrid Identification and Islanding in Future Medium Voltage Smart Grids based on Container*

- Virtualisation and Machine Learning*. Institute of Electrical and Electronics Engineers, Germany Section (Hrsg.): NEIS Conference on Sustainable Energy Supply and Energy Storage Systems, Hamburg, 26.-27.09.2022
- [122] *Docker: Accelerated, Containerized Application Development*. <https://www.docker.com>
- [123] *Kubernetes: Produktionsreife Container-Orchestrierung*. <https://kubernetes.io>
- [124] *User Datagram Protocol*. RFC 768. Information Sciences Institute, University of Southern California, Jonathan Bruce Postel (RFC Editor), 1980. <https://datatracker.ietf.org/doc/rfc768>
- [125] *Transmission Control Protocol*. RFC 793. Information Sciences Institute, University of Southern California, 1981. <https://datatracker.ietf.org/doc/rfc793/>
- [126] Rehtanz, Christian; Hilbrich, Dominik; Schulz, Günter; Claus, Jonas: *Schlussbericht zum Forschungsprojekt EOSG: Energieflussoptimierung im Smart Grid mittels intelligenter Netzkomponenten*. Dortmund, 2021
- [127] Jean Müller GmbH: *NH-Sicherungslastschaltleisten - SL mit Energie-Monitoring-Modul PLPlano - ModBus-RTU*. https://shop.jeanmueller.de/web/de_DE/L_Plano_ModBus%20RTU/ModBus-RTU/index.xhtml
- [128] Jean Müller GmbH: *PLPlano - Das kleinste, intelligente Energie-Monitoring-Modul für Lastschaltleisten*. https://media.jeanmueller.de/download/Kataloge/Elektronik/Produktflyer_PLPlano.pdf
- [129] *libmodbus. A Modbus library for Linux, Mac OS X, FreeBSD, QNX and Win32*. <https://libmodbus.org>
- [130] *SQLite*. <https://www.sqlite.org>
- [131] Chourdakis, Michael: *XML: Include a Flexible Parser in Your C++ Applications*. <https://www.codeproject.com/Articles/18732/XML-Include-a-Flexible-Parser-in-Your-C-Applicatio>
- [132] Ziegler, Gerhard: *Numerical distance protection. Principles and applications*. 4. Auflage, Erlangen: Publicis Publishing, 2011
- [133] *OPAL-RT Technologies*. <https://www.opal-rt.com>
- [134] *Energy management system application program interface (EMS-API). Common Information Model (CIM) base*. DIN IEC 61970-301. International Electrotechnical Commission, 2013
- [135] Hilbrich, Dominik; Raczka, Sebastian; Rehtanz, Christian: *Optimized grid reduction for state estimation calculations in under-determined distribution grids*. Institute of Electrical and Electronics Engineers, Power & Energy Society (Hrsg.): Power Systems Computation Conference (PSCC), Porto, Portugal, 27.06.-01.07.2022
- [136] Hilbrich, Dominik; Raczka, Sebastian; Rehtanz, Christian: *Optimized grid reduction for state estimation algorithms in under-determined distribution grids*.

Institute of Electrical and Electronics Engineers, Germany Section (Hrsg.): NEIS Conference on Sustainable Energy Supply and Energy Storage Systems, Hamburg, 26.-27.09.2022

- [137] *Security Architecture for the Internet Protocol*. RFC 4301. Internet Engineering Task Force, 2005. <https://datatracker.ietf.org/doc/rfc4301>
- [138] *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. Internet Engineering Task Force, 2008. <https://datatracker.ietf.org/doc/rfc5280>

Alle Internetverweise wurden zuletzt am 17.03.2023 auf Gültigkeit geprüft.

Wissenschaftlicher Tätigkeitsnachweis

Während meiner Tätigkeit als wissenschaftlicher Angestellter sind im Rahmen dieser Arbeit nachfolgende Bachelor-, Master- und Projektarbeiten sowie wissenschaftliche Veröffentlichungen und Vorträge entstanden. Den beteiligten Studierenden gilt mein besonderer Dank.

Bachelor-, Master- und Projektarbeiten

- [S1] Ahlmann, Raphael; Albert, Jannis; Gatto, Denny; Ressel, Benjamin; Schmitz, Lucas; Weischenberg, Marwin; Mohsen, Yazdi; Zheng, Hongyu: *Weiterentwicklung einer Forschungs- und Entwicklungsplattform für Netzschutzanwendungen*. Projektgruppe, 2017
- [S2] Ahmadi, Iadh; Basic, Ejub; Doerr, Hendrik; Egner, Tim; Khan, Irteza; Haj Satouf, Homam; Spies, Laura; Splittgerber, Luca: *Modelling of Flexibilities and Integration into a Smart-Grid-System*. Projektgruppe, 2021
- [S3] Berger, Andre; Bernhardt, Vladimir; Bertram, Christopher; Büßen, Jan-Ole; Demming, Michael; Ngongang, Moline; Tophinke, Michael; Wittemeier, Jonathan: *Entwurf einer Forschungs- und Entwicklungsplattform für Netzschutzanwendungen*. Projektgruppe, 2015
- [S4] Ceschlaw, Jan-Niklas: *Implementation and validation of a state estimation algorithm for the real-time simulator*. Bachelorarbeit, 2017
- [S5] Funke, Alexander: *Enhancement and Validation of a Voltage Regulation Algorithm for Measurement Devices with a Hardware-in-the-Loop Simulator*. Bachelorarbeit, 2017
- [S6] Kolls, Felix: *Implementierung parallelisierter Schutzfunktionen auf einem Coprozessor innerhalb eines Hardwaresystems für kombinierte Schutz- und Automatisierungsfunktionen*. Masterarbeit, 2019
- [S7] König, Julia: *Analyse und Modellierung der statischen und dynamischen Spannungsabhängigkeit eines exemplarischen Verteilnetzes mit hoher Durchdringung Erneuerbarer Energien*. Masterarbeit, 2017
- [S8] Llumigusin Sarzosa, Alexandra Jackeline: *Validation of smart grid functions of an automation system using an analogue grid model*. Masterarbeit, 2018
- [S9] Obermayer, Carsten: *Beschreibung und prototypische Umsetzung eines durchgängigen Top-Down-Engineeringprozesses für IEC-61850-Stationsprojekte eines Verteilnetzbetreibers mit herstellereigenen Geräte-Datenmodellen*. Masterarbeit, 2020
- [S10] Palaniappan, Rajkumar: *Implementation and validation of decentralised smart grid functions using distributed measurement acquisition devices*. Masterarbeit, 2016

- [S11] Rohde, Lena: *Erweiterung und Validierung eines Konfigurators zur automatischen Schutzgeräte-Konfiguration aus Datenmodellen gemäß IEC 61850-6*. Bachelorarbeit, 2016
- [S12] Schmitz, Marven: *Analyse und Bewertung existierender Smart-Grid-Lösungen hinsichtlich technischer und wirtschaftlicher Eigenschaften*. Bachelorarbeit, 2020
- [S13] Sohaib, Syed Muhammad: *Extension of a configurator for the automatic configuration of automation systems based on data models according to IEC 61850-6*. Masterarbeit, 2018

Veröffentlichungen

- [B1] Albrecht, Marvin; Horenkamp, Willi; Palaniappan, Rajkumar; Hilbrich, Dominik; Bauernschmitt, Björn; Keune, Björn Martin; Rehtanz, Christian: *Decentralized Frequency Stabilization in Grids with a High Share of Renewable Energies*. VDE ETG (Hrsg.): International ETG Congress, Bonn, 28.-29.11.2017
- [B2] Bauernschmitt, Björn; Hilbrich, Dominik; Keune, Björn Martin; Rehtanz, Christian: *Application of IEC 61850-6 and IEC 61970 Data Models for Automated Protection Test Specification*. Institution of Engineering and Technology (Hrsg.): 13th International Conference on Developments in Power System Protection (DPSP), Edinburgh, Schottland, 07.-10.03.2016, doi: 10.1049/cp.2016.0035
- [B3] Bauernschmitt, Björn; Hilbrich, Dominik; Rehtanz, Christian: *Anwendung von Datenmodellen gemäß IEC 61850-6 und IEC 61970 zur Spezifikation automatisierter Schutzprüfungen*. VDE ETG/FNN (Hrsg.): 9. ETG-/FNN-Tutorial Schutz- und Leittechnik, Berlin, 23.-24.02.2016
- [B4] Bauernschmitt, Björn; Kaliwoda, Michael; Keune, Björn Martin; Hilbrich, Dominik; Rehtanz, Christian: *An integral Engineering Process for centralised Protection & Control Systems according to IEC 61850-6*. Institute of Electrical and Electronics Engineers (Hrsg.): 5th International Conference on Modern Electric Power Systems (MEPS), Wrocław, Polen, 06.-09.06.2015, doi: 10.1109/MEPS.2015.7477158
- [B5] Bauernschmitt, Björn; Palaniappan, Rajkumar; Hilbrich, Dominik; Keune, Björn Martin; Rehtanz, Christian: *Modular konfigurier- und prüfbare Automatisierungsarchitektur für zukünftige aktive elektrische Energienetze*. VDE ETG/FNN (Hrsg.): 10. ETG-/FNN-Tutorial Schutz- und Leittechnik, Berlin, 20.-21.02.2018
- [B6] Bauernschmitt, Björn; Palaniappan, Rajkumar; Hilbrich, Dominik; Rehtanz, Christian: *Modular Configurable and Testable Automation Architecture for future Active Electrical Energy Grids*. Institute of Electrical and Electronics Engineers (Hrsg.): 53rd International Universities Power Engineering Conference (UPEC), Glasgow, Schottland, 04.-07.09.2018, doi: 10.1109/UPEC.2018.8541947

- [B7] Böhm, Ralf; Steglich, Michael; Bauernschmitt, Björn; Rehtanz, Christian; Franke, Jörg: *A hybrid compensation system for provision of ancillary services in medium voltage grids*. Institute of Electrical and Electronics Engineers (Hrsg.): International Conference on Diagnostics in Electrical Engineering (Diagnostika), Pilsen, Tschechien, 04.-07.09.2018, doi: 10.1109/DIAGNOSTIKA.2018.8526100
- [B8] Ceschlaw, Jan-Niklas; Palaniappan, Rajkumar; Bauernschmitt, Björn; Brüggemann, Annika; Rehtanz, Christian: *Application of a Dynamic State Estimation Method for a Real-time Simulated Distribution System*. Institute of Electrical and Electronics Engineers, Power & Energy Society (Hrsg.): Power and Energy Student Summit (PESS), Magdeburg, 09.-11.07.2019
- [B9] Hilbrich, Dominik; Bauernschmitt, Björn; Keune, Björn Martin; Rehtanz, Christian; Lehnhoff, Sebastian: *Design of a Hybrid Platform for Online Application Testing with IEC 61970*. Institution of Engineering and Technology (Hrsg.): 13th International Conference on Developments in Power System Protection (DPSP), Edinburgh, Schottland, 07.-10.03.2016, doi: 10.1049/cp.2016.0053
- [B10] Hilbrich, Dominik; Bauernschmitt, Björn; Rehtanz, Christian: *Entwicklung eines hybriden Prüfsystems für anwendungsorientierte Prüfungen von Schutzsystemen*. VDE ETG/FNN (Hrsg.): 9. ETG-/FNN-Tutorial Schutz- und Leittechnik, Berlin, 23.-24.02.2016
- [B11] Hilbrich, Dominik; Keune, Björn Martin; Bauernschmitt, Björn; Palaniappan, Rajkumar; Rehtanz, Christian: *Hardware-in-the-Loop-Prüfung von kombinierten Schutz- und Leittechnikfunktionen für zukünftige aktive Verteilnetze*. VDE ETG/FNN (Hrsg.): 10. ETG-/FNN-Tutorial Schutz- und Leittechnik, Berlin, 20.-21.02.2018
- [B12] Keune, Björn Martin; Bauernschmitt, Björn; Rehtanz, Christian; Lehnhoff, Sebastian: *Standardization of Automated Disturbance Management for Urban Distribution Power Systems*. Institution of Engineering and Technology (Hrsg.): 13th International Conference on Developments in Power System Protection (DPSP), Edinburgh, Schottland, 07.-10.03.2016, doi: 10.1049/cp.2016.0060
- [B13] Krüger, Carsten; Kamsamrong, Jirapa; Lehnhoff, Sebastian; Raczka, Sebastian; Bauernschmitt, Björn; Hilbrich, Dominik; Palaniappan, Rajkumar; Rehtanz, Christian; Puhe, Frederik; Hermanns, Carsten; Keune, Björn M.; Finke, Sven; Schmidt, Marius; Ebeling, Marc; Obermayer, Carsten; Arph, Jan; Hein, Aleksej; Becker, Thomas; Richter, Frank: *Engineeringprozess für schutz- und leittechnische Applikationen in Ortsnetzstationen mittels Datenmodellierung nach IEC 61850 und Container-basierter Virtualisierung*. VDE ETG/FNN (Hrsg.): 12. ETG-/FNN-Tutorial Schutz- und Leittechnik, Online-Preview, 15.-16.02.2022
- [B14] Palaniappan, Rajkumar; Bauernschmitt, Björn; Hilbrich, Dominik; Rehtanz, Christian: *An Intelligent Measurement and Control Device for Active Distribution*

- Grids*. Institute of Electrical and Electronics Engineers, Power & Energy Society (Hrsg.): Innovative Smart Grid Technologies Conference Europe (ISGT Europe), Virtuell, 26.-28.10.2020, doi: 10.1109/ISGT-Europe47291.2020.9248862
- [B15] Palaniappan, Rajkumar; Funke, Alexander; Hilbrich, Dominik; Bauernschmitt, Björn; Rehtanz, Christian: *A robust and resilient voltage control strategy for smart grids using distributed measurements*. Institute of Electrical and Electronics Engineers (Hrsg.): Innovative Smart Grid Technologies Asia (ISGT Asia), Chengdu, China, 21.-24.05.2019, S. 2411–2415, doi: 10.1109/ISGT-Asia.2019.8881679
- [B16] Palaniappan, Rajkumar; Hilbrich, Dominik; Bauernschmitt, Björn; Rehtanz, Christian: *Co-ordinated voltage regulation using distributed measurement acquisition devices with a real-time model of the Cigré low-voltage benchmark grid*. IET Generation, Transmission and Distribution, Volume 13, Issue 5, S. 710–716, 2019, doi: 10.1049/iet-gtd.2018.5300
- [B17] Palaniappan, Rajkumar; Raczka, Sebastian; Bauernschmitt, Björn; Hilbrich, Dominik; Rehtanz, Christian: *i-Autonomous: Standardization and integration of modular and autonomous components in intelligent local substations*. IEEE Power & Energy Society (Hrsg.): General Meeting, Virtuell, 26.-29.07.2021
- [B18] Palaniappan, Rajkumar; Richter, Frank; Bauernschmitt, Björn; Hilbrich, Dominik: *Implementation of decentralized Smart Grid functions using Distributed Measurement Acquisition Devices*. Institute of Electrical and Electronics Engineers, Power & Energy Society (Hrsg.): Power and Energy Student Summit (PESS), Aachen, 19.-20.01.2016
- [B19] Palaniappan, Rajkumar; Richter, Frank; Bauernschmitt, Björn; Hilbrich, Dominik; Rehtanz, Christian: *Implementation and Validation of Decentralized Smart Grid Functions using Distributed Measurement Acquisition Devices*. Institute of Electrical and Electronics Engineers (Hrsg.): 6th International Conference on Power Systems (ICPS), Neu-Delhi, Indien, 04.-06.03.2016, doi: 10.1109/ICPES.2016.7584184
- [B20] Raczka, Sebastian; Bauernschmitt, Björn; Hilbrich, Dominik; Palaniappan, Rajkumar; Rehtanz, Christian; Krüger, Carsten; Kamsamrong, Jirapa; Lehnhoff, Sebastian; Puhe, Frederik; Keune, Björn M.; Obermayer, Carsten; Hermanns, Carsten; Stelmaszyk, Henriette; Finke, Sven; Schmidt, Marius; Ebeling, Marc; Arph, Jan; Hein, Aleksej; Becker, Thomas; Richter, Frank: *A novel software applications rollout and monitoring strategy for enabling the transition to electromobility in future smart grids*. CIREN (Hrsg.): CIREN workshop on e-mobility and power distribution systems, Porto, Portugal, 02.-03.06.2022, doi: 10.1049/icp.2022.0760
- [B21] Raczka, Sebastian; Hilbrich, Dominik; Bauernschmitt, Björn; Palaniappan, Rajkumar; Rehtanz, Christian: *Modellprädiktiver Regelungsalgorithmus als*

Smart-Grid-Automatisierungsfunktion für zukünftige aktive elektrische Energienetze. VDE ETG/FNN (Hrsg.): 11. ETG-/FNN-Tutorial Schutz- und Leittechnik, Berlin, 18.-19.02.2020

- [B22] Steglich, Michael; Löwe, Christopher; Bauernschmitt, Björn; Rehtanz, Christian; Böhm, Ralf; Franke, Jörg: *A novel method to reduce the harmonic currents in the residual earth fault current during a single phase to ground fault in compensated grids.* Institute of Electrical and Electronics Engineers, Power & Energy Society (Hrsg.): Innovative Smart Grid Technologies Europe (ISGT Europe), Bukarest, Rumänien, 29.09.-02.10.2019, doi: 10.1109/ISGTEurope.2019.8905599
- [B23] Velásquez, Jorge; Krüger, Carsten; Babazadeh, Davood; Lehnhoff, Sebastian; Palaniappan, Rajkumar; Bauernschmitt, Björn; Hilbrich, Dominik; Rehtanz, Christian: *Flexible and reconfigurable automation architecture for electrical power systems.* Institute of Electrical and Electronics Engineers (Hrsg.): 13th IEEE PowerTech, Mailand, Italien, 23.-27.06.2019, doi: 10.1109/PTC.2019.8810822

Projektberichte

- [P1] Rehtanz, Christian; Bauernschmitt, Björn; Hilbrich, Dominik; Palaniappan, Rajkumar: *Schlussbericht zum Forschungsprojekt i-Automate: Modular konfigurier- und prüfbare Automatisierungsarchitektur für zukünftige aktive elektrische Energienetze; Teilvorhaben: Automatisierungsfunktionen und Algorithmen für Smart Grids, automatisierte Systemvalidierung.* Dortmund, 2020
- [P2] Rehtanz, Christian; Kaliwoda, Michael; Keune, Björn Martin; Bauernschmitt, Björn; Hilbrich, Dominik: *Schlussbericht zum Forschungsprojekt i-Protect: Intelligente Schutz- und Leittechnik für elektrische Energienetze basierend auf zukünftigen IKT- und Automatisierungsarchitekturen; Teilvorhaben: Entwicklung der Mess- und Schutzfunktionen und Durchführung der Systemtests.* Dortmund, 2015

Anhang A Strukturierung der Namensräume

Tabelle A.1: Strukturierung der Namensräume

| Namensraum Beschreibung |
|---|
| ie3.SmartGridTechnologies.BL Allgemeine Anwendungslogik (engl.: Business Logic): Basisklasse für Smart-Grid-Funktionsklassen und Hilfsklassen |
| ie3.SmartGridTechnologies.BL.DeviceConfig Klassen der Programmkomponente <i>Autoconfig</i> zur Konfiguration des Systems auf Basis eines SCL-Datenmodells und zur Programmablaufsteuerung |
| ie3.SmartGridTechnologies.COMM Klassen zur Kommunikation per UDP, TCP und Modbus |
| ie3.SmartGridTechnologies.DATA Hilfsklassen zur Nutzung des IEC-61850-Kommunikationsstacks |
| ie3.SmartGridTechnologies.HAL.KoCoS Abstraktion von der Systemplattform (engl.: Hardware Abstraction Layer, hier: KoCoS) |
| ie3.SmartGridTechnologies.BL.Protection Klassen von Schutzfunktionen |
| ie3.SmartGridTechnologies.BL.VoltageControl Klassen von Spannungsregelungsfunktionen |
| ie3.SmartGridTechnologies.BL.MachineLearning Klassen von Algorithmen des maschinellen Lernens |
| ie3.SmartGridTechnologies.BL.ModelPredictiveControl Klassen von Algorithmen der modellprädiktiven Regelung |
| ie3.SmartGridTechnologies.BL.OptimalPowerFlow Klassen der Leistungsflussoptimierung |
| ie3.SmartGridTechnologies.BL.StateEstimation Klassen der Netzzustandsschätzung |
| ie3.SmartGridTechnologies.BL.ClassBase Hilfsklassen, gemeinsam von den Algorithmen des ML, MPC, OPF und SE genutzt |
| ie3.SmartGridTechnologies.BL.Solvers Klassen zur Lösung mathematischer Probleme, u. a. genutzt von MPC und OPF |
| HuS Deklarationen (Header) und Bibliothek des IEC-61850-Kommunikationsstacks |
| KoCoS Hardwarespezifische Klassen zur Konfiguration der und Kommunikation mit der KoCoS-Systemplattform, wird durch <code>ie3.SmartGridTechnologies.HAL.KoCoS</code> genutzt |
| SQLite3 Klassen zum Zugriff auf SQLite-Datenbanken |

Anhang B Protokolldetails der TCP-Kommunikation

Tabelle B.1: TCP-Header

| Datentyp | Name | Beschreibung |
|--------------------------------|-----------|--|
| uint32_t | length | Gesamtlänge der Nachricht (Header und Nutzdaten) |
| uint32_t | id | Nachrichten-ID, kann durch den TCP-Client genutzt werden, um eine Antwort einer Anfrage zuzuordnen, wird durch den TCP-Server unverändert gespiegelt |
| E_TCP_MESSAGE_TYPE : uint8_t | type | Nachrichtentyp, siehe Tabelle B.2 |
| E_TCP_MESSAGE_LNTYPE : uint8_t | lnType | Logischer-Knoten-Typ, siehe Tabelle B.3 |
| uint8_t | unused0 | |
| uint8_t | unused1 | |
| E_LNCLASS : uint32_t | lnClass | Logische-Knoten-Klasse |
| char[64] | reference | Logische-Knoten-Referenz |

Tabelle B.2: Nachrichten-Typen

| Name | Wert | Beschreibung | Quelle | Nutzdaten |
|----------------------------------|------|--|--------|-----------|
| GET_LN_COMMAND | 0x01 | Logischen Knoten lesen | Client | Nein |
| SET_LN_COMMAND | 0x02 | Logischen Knoten schreiben | Client | Ja |
| SUBSCRIBE_LN_COMMAND | 0x03 | Logischen Knoten abonnieren | Client | Nein |
| UNSUBSCRIBE_LN_COMMAND | 0x04 | Logischen Knoten nicht mehr abonnieren | Client | Nein |
| REPORT_LN | 0x09 | Report, der bei Wertänderung in einem Datenattribut des abonnierten logischen Knotens automatisch versendet wird | Server | Ja |
| GET_LN_RESPONSE_POSITIVE | 0x41 | Positive Antwort auf GET_LN_COMMAND, enthält die Werte des logischen Knotens | Server | Ja |
| GET_LN_RESPONSE_NEGATIVE | 0x81 | Negative Antwort auf GET_LN_COMMAND z. B. wenn ein ungültiger logischer Knoten angefragt wurde | Server | Nein |
| SET_LN_RESPONSE_POSITIVE | 0x42 | Positive Antwort auf SET_LN_COMMAND | Server | Nein |
| SET_LN_RESPONSE_NEGATIVE | 0x82 | Negative Antwort auf SET_LN_COMMAND | Server | Nein |
| SUBSCRIBE_LN_RESPONSE_POSITIVE | 0x43 | Positive Antwort auf SUBSCRIBE_LN_COMMAND | Server | Nein |
| SUBSCRIBE_LN_RESPONSE_NEGATIVE | 0x83 | Negative Antwort auf SUBSCRIBE_LN_COMMAND | Server | Nein |
| UNSUBSCRIBE_LN_RESPONSE_POSITIVE | 0x44 | Positive Antwort auf UNSUBSCRIBE_LN_COMMAND | Server | Nein |
| UNSUBSCRIBE_LN_RESPONSE_NEGATIVE | 0x84 | Negative Antwort auf UNSUBSCRIBE_LN_COMMAND | Server | Nein |

Tabelle B.3: Logischer-Knoten-Typ

| Name | Wert | Beschreibung |
|---------|------|--|
| STATUS | 0x00 | Nur Status-Datenattribute verändern, üblicherweise beim Lesen von Status- und Messwerten z. B. <i>stVal</i> , <i>cVal/mag/f</i> (Server zu Client) |
| COMMAND | 0x01 | Nur Befehls-Datenattribute verändern, üblicherweise beim Schreiben von Befehlswerten z. B. <i>ct/Val</i> (Client zu Server) |

Anhang C Modellierung der Modbus-Kommunikation

Tabelle C.1: Gültige Schlüssel für Modbus-Parameter

| Schlüssel | Beschreibung | Optional/Bedingt ¹² |
|-----------------------------|--|--------------------------------|
| Address | Server- bzw. Slave-Adresse, Standardwert=1 | Optional |
| InputBit DiscreteInput | Adresse eines Modbus-Input-Bits (Discrete Input) | Bedingung #1 |
| Bit Coil | Adresse eines Modbus-Bits (Coil) | Bedingung #1 |
| InputRegister | Adresse eines Modbus-Input-Registers | Bedingung #1 |
| Register HoldingRegister | Adresse eines Modbus-Registers (Holding Register) | Bedingung #1 |
| DataType | Typ der in dem/den Modbus-Register(n) gespeicherten Daten, siehe Tabelle C.2 | Optional |
| SwapByte | Siehe Tabelle C.3 | Optional |
| SwapWord | Siehe Tabelle C.3 | Optional |
| Endian | ABCD, BADC, CDAB, DCBA | Optional |
| Offset | Offset (zu dem Wert eines Datenattributs des Online-Datenmodells wird der Offset addiert, wenn er in den Wert eines Modbus-Registers umgewandelt wird bzw. von dem Wert eines Modbus-Registers wird der Offset subtrahiert, wenn er in den Wert eines Datenattributs des Online-Datenmodells umgewandelt wird) | Optional |
| Scaling | Skalierungsfaktor (der Wert eines Datenattributs des Online-Datenmodells wird mit dem Skalierungsfaktor multipliziert, wenn er in den Wert eines Modbus-Registers umgewandelt wird bzw. der Wert eines Modbus-Registers wird durch den Skalierungsfaktor dividiert, wenn er in den Wert eines Datenattributs des Online-Datenmodells umgewandelt wird) | Optional |

¹² Nur ein Schlüssel der Bedingung #n ist gleichzeitig erlaubt.

Tabelle C.2: Gültige Werte für den Parameter *DataType*

| Wert | Registeranzahl | Beschreibung |
|--------|----------------|--|
| bool | 1 | Register wird als bool interpretiert |
| uint8 | 1 | Register wird als uint8 interpretiert |
| int8 | 1 | Register wird als int8 interpretiert |
| uint16 | 1 | Register wird als uint16 interpretiert (natives Format) |
| int16 | 1 | Register wird als int16 interpretiert |
| uint32 | 2 | Zwei benachbarte Registers werden als uint32 interpretiert |
| int32 | 2 | Zwei benachbarte Registers werden als int32 interpretiert |
| float | 2 | Zwei benachbarte Registers werden als float interpretiert |

Tabelle C.3: Endianness

| Endianness | Bytereihenfolge | Parameter |
|---------------|-----------------|---------------------|
| Big Endian | ABCD | |
| | BADC | SwapByte |
| | CDAB | SwapWord |
| Little Endian | DCBA | SwapByte & SwapWord |

Tabelle C.4: P-Adresselemente zur Modellierung eines Modbus-RTU-Servers

| Typ | Beschreibung | Optional |
|-----------------|----------------------------------|--------------------------------|
| SERIAL-PORT | Name der seriellen Schnittstelle | |
| SERIAL-BAUDRATE | Baudrate | Optional, standardmäßig 115200 |
| SERIAL-MODE | Siehe Tabelle C.5 | Optional, standardmäßig RS485 |

Tabelle C.5: Gültige Werte des P-Adresselements *SERIAL-MODE*

| Wert | Beschreibung |
|-------|--|
| RS232 | Bei dem seriellen Netzwerk handelt es sich um RS232. |
| RS485 | Bei dem seriellen Netzwerk handelt es sich um RS485 (4-Draht). |

Anhang D Konfiguration der binären Ein- und Ausgänge

Tabelle D.1: Parameter zur Konfiguration der binären Ein- und Ausgänge

| Datentyp des Datenattributs | Funktions-einschränkung | Beispiel | Parameter Nr. | Beschreibung |
|----------------------------------|-------------------------|------------------------------|---------------|-----------------------|
| BOOLEAN | Status | GGIO1/SPCSO1/stVal | 1 | Eingang |
| BOOLEAN | Command | CSW11/Pos/Oper/ctIVal | 1 | Ausgang |
| Tcmd (Transformer Command) | Command | ATCC1/TapChg/Oper/ ctIVal | 1 | Ausgang <i>lower</i> |
| | | | 2 | Ausgang <i>higher</i> |
| | | | 3 | Impulsdauer |