



Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften  
Department für Informatik

Masterstudiengänge Informatik & Wirtschaftsinformatik

---

# Abschlussbericht der Projektgruppe Blackout-Restart

---

**vorgelegt von:**

Hendrik Buhl  
Tim Dombrowski  
Erik Hogen  
Maximilian Kreutz  
Daniel Palm  
Sanja Stark  
Peer Stubbe  
Stephanie Warsch

**Gutachter:**

Prof. Dr. Michael Sonnenschein  
Prof. Dr. Sebastian Lehnhoff  
Dr. Jörg Bremer

Oldenburg, den 31.03.2017

# Zusammenfassung

Durch die zunehmende Verbreitung kleiner Energiewandlungsanlagen im Rahmen der durch die Bundesregierung ausgerufenen Energiewende, ist eine Veränderung in der Struktur der Stromerzeugung zu beobachten. Von einer zentralen Erzeugung ausgehend, entwickelt sich eine dezentrale, verteilte Stromproduktion. Diese bringt durch ihre erhöhte Anzahl an Entitäten eine deutlich komplexere Organisation des Stromnetzes mit sich.

Gleichzeitig erhalten Informations- und Kommunikationstechnik Einzug in Stromversorgung – das Smart Grid entwickelt sich. Zwischen dem hierfür benötigten Kommunikationsnetz und dem Stromnetz besteht eine wechselseitige Abhängigkeit. Diese ist dafür verantwortlich, dass das Wiederherstellen der Stromversorgung nach einem Blackout in einem von dezentraler Energieerzeugung geprägten Smart Grid eine große Herausforderung darstellt.

Im Rahmen der Projektgruppe wurde ein Algorithmus entwickelt, der den Wiederaufbau eines solchen Szenarios ermöglicht. Dieser verfolgt das zyklische Bilden von Inselnetzen und den Zusammenschluss dieser, um schnellstmöglich eine stabile Stromversorgung wiederherzustellen.

Der Algorithmus bietet die Möglichkeit beim Wiederaufbau des Stromnetzes verschiedene Zielvorgaben zu berücksichtigen und Netzknoten zu priorisieren. Dabei ist er in der Lage auch bei Einschränkung des Kommunikationsnetzes zu arbeiten.

Das entwickelte Verfahren wurde im Rahmen der Projektgruppe prototypisch implementiert und in einer einfachen Simulationsumgebung getestet. Anschließend wurde eine Validierung durchgeführt, die den sinnvollen Ansatz des Verfahrens unterstreicht.

# Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Tabellenverzeichnis	VIII
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Aufgabenstellung	1
1.2 Ziele und Vision der Projektgruppe	2
1.3 Aufbau des Berichts	3
<b>2 Projektmanagement</b>	<b>5</b>
2.1 Vorgehensmodell	5
2.2 Projektorganisation und -ablauf	5
2.2.1 Seminar- und Projektphase	6
2.2.2 Projektsitzungen	6
2.2.3 Zeitplanung	7
2.2.4 Rollenmodell	9
2.2.5 Qualitätsmanagement	12
2.3 Infrastruktur	12
<b>3 Spezifikation der Problemstellung</b>	<b>15</b>
3.1 Related Work	15
3.2 Szenariodefinition	17
3.3 Anforderungserhebung	24
3.3.1 Ziele	24
3.3.2 Anforderungen	25
3.3.2.1 Szenarioanforderungen	25
3.3.2.2 Funktionale Anforderungen bezüglich des System	26
3.3.2.3 Funktionale Anforderungen bezüglich des Verfahrens	26
3.3.2.4 Nicht funktionale Anforderungen	29
3.3.3 Rahmenbedingungen	29
3.4 Optimierungsziele	30
3.4.1 Mathematische Formulierung	30
3.4.2 Multikriterielle Zielfunktion	35
3.5 COHDA als Lösungsheuristik	37
<b>4 Algorithmische Umsetzung</b>	<b>39</b>
4.1 Gesamtüberblick	39
4.2 Zustandskonzept	40
4.2.1 Sprecher	40

---

4.2.2	Normale Agenten . . . . .	41
4.3	Verfahren . . . . .	43
4.3.1	Verbundbildung . . . . .	43
4.3.2	Erstellen des Overlay-Netzes . . . . .	45
4.3.3	Verhandlung mit COHDA . . . . .	47
4.3.3.1	Anpassungen an COHDA . . . . .	47
4.3.3.2	Ablauf im Algorithmus . . . . .	51
4.3.4	Verarbeitung des Optimierungsergebnisses . . . . .	54
4.3.5	Aufnahme von Agenten in eine Insel . . . . .	55
4.3.6	Vorbereitung einer Inselvereinigung . . . . .	56
4.3.7	Inselvereinigung . . . . .	57
4.3.8	Terminierung des Verfahrens . . . . .	58
<b>5</b>	<b>Implementierung</b>	<b>61</b>
5.1	Systemarchitektur . . . . .	61
5.1.1	Grunddesign und verwendete Frameworks . . . . .	61
5.1.1.1	Entwicklungsprinzipien . . . . .	61
5.1.1.2	Objekthierarchie . . . . .	62
5.1.2	Multiagentenframework . . . . .	63
5.1.3	Module . . . . .	64
5.1.3.1	CSV-Manager . . . . .	65
5.1.3.2	Printer . . . . .	66
5.1.3.3	Actions . . . . .	66
5.1.3.4	Behaviours . . . . .	67
5.1.3.5	Graphstream . . . . .	67
5.2	Pacman . . . . .	69
5.2.1	Das MiniMAS . . . . .	69
5.2.1.1	Multiagentenplattform . . . . .	69
5.2.1.2	Events . . . . .	70
5.2.1.3	Zeitschritte . . . . .	71
5.2.1.4	Termination des Multiagentensystems . . . . .	71
5.2.2	Agenten . . . . .	72
5.2.2.1	Default-Agent . . . . .	72
5.2.2.2	Pacman-Agent . . . . .	72
5.2.2.3	Schalter-Agent . . . . .	73
5.2.2.4	Konvergenz-Agent . . . . .	73
5.2.3	Actions . . . . .	74
5.2.4	Behaviours . . . . .	75
5.2.5	Nachrichtentypen . . . . .	75
5.3	Grenzen des Systems . . . . .	76
<b>6</b>	<b>Tests</b>	<b>78</b>
6.1	Vorgehen . . . . .	78
6.2	Vorbereitung . . . . .	79
6.2.1	Identifizieren der Eingabeparameter . . . . .	79
6.2.2	Bildung von Äquivalenzklassen . . . . .	83
6.2.3	Ableiten der Testfälle . . . . .	86
6.2.4	Nicht-lösbare Szenarien . . . . .	90
6.2.5	Erstellung der Testszenarien . . . . .	91

---

6.2.6	Mapping von Testfällen und Anforderungen . . . . .	93
6.3	Durchführung . . . . .	94
6.4	Auswertung . . . . .	95
6.4.1	Ergebnisse und Interpretation . . . . .	95
6.4.2	Fazit . . . . .	96
<b>7</b>	<b>Validierung</b> . . . . .	<b>97</b>
7.1	Aufbau . . . . .	97
7.1.1	Validierungskriterien . . . . .	97
7.1.1.1	Korrektheit . . . . .	98
7.1.1.2	Terminierung . . . . .	99
7.1.1.3	Selbststabilisierung . . . . .	101
7.1.1.4	Performanz . . . . .	101
7.1.1.5	Effizienz . . . . .	102
7.1.1.6	Skalierbarkeit . . . . .	102
7.1.1.7	Robustheit . . . . .	103
7.1.2	Eingabeparameter . . . . .	103
7.1.3	Konstruktionsvorschriften . . . . .	105
7.2	Ergebnisse und Interpretation . . . . .	107
7.2.1	Allgemeine Erkenntnisse . . . . .	108
7.2.2	Versuchsergebnisse . . . . .	109
7.2.2.1	Länge der Niederspannungsstränge . . . . .	109
7.2.2.2	Anzahl Agenten . . . . .	111
7.2.2.3	Anzahl Sprecher . . . . .	112
7.2.2.4	Anteil flexibler Erzeuger und Verbraucher . . . . .	114
7.2.2.5	Anzahl Shortcuts . . . . .	115
7.2.2.6	Gewichtung der Zielfunktion . . . . .	117
7.2.2.7	Priorisierung . . . . .	119
7.2.2.8	Kombination von Parametern . . . . .	121
7.3	Fazit . . . . .	125
<b>8</b>	<b>Fazit und mögliche Erweiterungen</b> . . . . .	<b>128</b>
	<b>Literaturverzeichnis</b> . . . . .	<b>130</b>
	<b>Abkürzungen</b> . . . . .	<b>132</b>
<b>A</b>	<b>Validierungsergebnisse</b> . . . . .	<b>134</b>
<b>B</b>	<b>Anwenderhandbuch</b> . . . . .	<b>143</b>
<b>C</b>	<b>Programmierkonventionen</b> . . . . .	<b>149</b>
<b>D</b>	<b>Abbildungen</b> . . . . .	<b>153</b>
<b>E</b>	<b>Seminarband</b> . . . . .	<b>157</b>

# Abbildungsverzeichnis

2.1	Zeitplanung des dritten Prototyps . . . . .	8
2.2	Rollenmodell des ersten Projektphase . . . . .	10
2.3	Rollenmodell der zweiten Projektphase . . . . .	10
3.1	Mögliches Netz mit verschiedenen Kanten und aktivierbaren Nachbarn	20
4.1	Zustandskonzept eines Sprechers . . . . .	41
4.2	Zustandskonzept eines Agenten . . . . .	43
4.3	Nachbarschaftsmöglichkeiten einer Insel . . . . .	44
4.4	Overlay-Netz für das Beispiel aus Abb. 4.3 . . . . .	46
4.5	Verarbeitung des Optimierungsergebnisses . . . . .	54
5.1	Klassendiagramm der in der Implementierung verwendeten Hierarchie der Agenten-Klasse . . . . .	63
5.2	Module des implementierten Systems . . . . .	65
5.3	Visualisierung eines Szenarios mittels Graphstream . . . . .	68
5.4	Aktivitätsdiagramm des MiniMAS . . . . .	70
5.5	Aufbau der Events . . . . .	71
6.1	Kombinationsmatrix der Basisnetze . . . . .	89
6.2	Graphische Darstellung von Testfall 10 . . . . .	91
6.3	Graphische Darstellung von Testfall 8 . . . . .	93
7.1	Standardisierter Mittelspannungsring . . . . .	107
7.2	Validierungsergebnisse bei Skalierung der Länge der Niederspannungs- stränge . . . . .	111
7.3	Validierungsergebnisse bei Skalierung der Anzahl der Agenten . . . . .	112
7.4	Validierungsergebnisse bei Skalierung der Anzahl der Sprecher . . . . .	114
7.5	Skalierung des Anteils flexibler Erzeuger und Verbraucher . . . . .	115

---

7.6	Validierungsergebnisse bei Skalierung der Anzahl der Shortcuts . . . .	117
7.7	Szenario zur Validierung der Gewichtung der Zielfunktion und der Priorisierung . . . . .	120
7.8	Kombination Sprecher und Länge NS-Stränge . . . . .	122
7.9	Kombination Sprecher und Anzahl MS-Ringe . . . . .	124
B.1	Ordnerstruktur für Netze . . . . .	143
B.2	Auswahl Netze . . . . .	144
B.3	Auswahl Sprecher . . . . .	145
B.4	Auswahl Ausgaben . . . . .	145
B.5	Fehler in einer der Dateien für die Kanten . . . . .	147
B.6	Eine Datei wurde falsch benannt . . . . .	148
B.7	Der Ordnerpfad wurde falsch benannt . . . . .	148
B.8	Der Planungshorizont ist falsch . . . . .	148
D.1	Zeitplan des ersten Semesters . . . . .	154
D.2	Meilensteine des ersten Semesters . . . . .	155
D.3	Testfälle . . . . .	156

# Tabellenverzeichnis

3.1	Übersicht über die Akteure im Szenario. . . . .	19
3.2	Abstraktionsniveau der verschiedenen Prototypen . . . . .	22
5.1	Übersicht über die verwendeten Nachrichtentypen. . . . .	77
6.1	Liste der Eingabeparameter des Systems und deren Beschreibung . . .	82
6.2	Auflistung der Äquivalenzklassen der Eingabeparameter des Systems	84
6.3	Definition der Testfälle nicht-lösbarer Szenarien . . . . .	90
7.1	Untersuchung der Gewichtung der Zielfunktion . . . . .	119
7.2	Step der Aufnahme der priorisierten Agenten . . . . .	121

# Referenten

Hendrik Buhl

[hendrik.buhl1@uni-oldenburg.de](mailto:hendrik.buhl1@uni-oldenburg.de)

Tim Dombrowski

[tim.dombrowski@uni-oldenburg.de](mailto:tim.dombrowski@uni-oldenburg.de)

Erik Hogen

[erik.julian.hogen@uni-oldenburg.de](mailto:erik.julian.hogen@uni-oldenburg.de)

Maximilian Kreutz

[maximilian.kreutz@uni-oldenburg.de](mailto:maximilian.kreutz@uni-oldenburg.de)

Daniel Palm

[daniel.palm@uni-oldenburg.de](mailto:daniel.palm@uni-oldenburg.de)

Sanja Stark

[sanja.stark@uni-oldenburg.de](mailto:sanja.stark@uni-oldenburg.de)

Peer Stubbe

[peer.vincent.stubbe@uni-oldenburg.de](mailto:peer.vincent.stubbe@uni-oldenburg.de)

Stephanie Warsch

[stephanie.alexandra.warsch@uni-oldenburg.de](mailto:stephanie.alexandra.warsch@uni-oldenburg.de)

# 1. Einleitung

Im Rahmen der an der Carl von Ossietzky Universität Oldenburg angebotenen Masterstudiengänge Wirtschaftsinformatik, Informatik, Eingebettete Systeme und Mikrorobotik wird im Studienverlauf eine Projektgruppe durchgeführt. Diese Projektgruppe, die sich über einen Zeitraum von zwei Semestern erstreckt und in der Regel aus sechs bis zwölf Studierenden besteht, setzt sich zum Ziel, eine komplexe Aufgabe aus dem Bereich der Informatik zu lösen. Zum erfolgreichen Bestehen dieser Studienleistung arbeiten die Studierenden gemeinsam ein Jahr lang als Team an dieser Aufgabenstellung, wobei sie zugleich berufstypische Arbeitsweisen kennenlernen und ihre Kompetenzen und Fähigkeiten ausbauen.

Der vorliegende Bericht entstammt der Projektgruppe Blackout – Restart. Die Projektgruppe wurde aus den Abteilungen Energieinformatik und Umweltinformatik ins Leben gerufen und wird von Prof. Dr. Sebastian Lehnhoff, Prof. Dr. Michael Sonnenschein, Dr. Christian Hinrichs, Dr. Jörg Bremer und Okko Nannen betreut und geleitet. Die Teilnehmer dieser Projektgruppe sind Studierende der Masterstudiengänge Informatik und Wirtschaftsinformatik.

## 1.1 Motivation und Aufgabenstellung

Der Aufbau der klassischen Stromversorgung ist geprägt durch eine verbrauchernahe Stromerzeugung durch fossile Großkraftwerke. Dieser Aufbau wandelt sich zunehmend zu einer räumlichen Entkoppelung von Erzeugung und Verbrauch. Hintergrund dieser Entwicklung ist die Energiewende, welche eine wachsende Integration erneuerbarer Energiequellen nach sich zieht. Intelligente Stromnetze – auch Smart Grids genannt – unterstützen dabei auf der einen Seite das Zusammenspiel von Energieangebot und -nachfrage und auf der anderen Seite die bereits genannte Integration erneuerbarer Energiequellen. Um diese Anforderungen zu realisieren, stellt die Informations- und Kommunikationstechnik einen wichtigen Bestandteil des Smart Grid dar. Die Stromversorgung der Zukunft besteht aus einem Strom- und einem Kommunikationsnetz, welche aufeinander angewiesen sind. Bezüglich des vorliegenden Berichts war die Projektgruppe mit der Aufgabenstellung konfrontiert, ein System zu entwickeln, das die Stromversorgung im Smart Grid im Falle eines Blackouts mit Hilfe eines Multiagentensystems so vollständig wiederherstellt.

## 1.2 Ziele und Vision der Projektgruppe

Nachdem die Aufgabenstellung in Abschnitt 1.1 beschrieben wurde, wird in diesem Abschnitt näher auf die Vision und die damit verbundenen Zielen eingegangen. Das Hauptziel der Projektgruppe, die Entwicklung eines Multiagentensystems zur Wiederherstellung der Stromversorgung im Smart Grid, unterliegt den Herausforderungen der Entwicklung eines Algorithmus, welcher eine verteilte Optimierung des Verbrauchs und der Erzeugung im Verbund ermöglicht, der prototypischen Implementierung des Ansatzes, der Realisierung der Kommunikation zwischen den Agenten und der Evaluation der Effizienz und Performanz des Ansatzes.

Im Rahmen der Projektorganisation und des weiteren Vorgehens wurde durch die Projektgruppe zu Beginn der Projektphase unter anderem eine gemeinsame Vision entwickelt. Diese soll ein erfolgreiches und zielführendes Zusammenarbeiten ermöglichen und stellt die Grundlage für die Ableitung von Zielen, Anforderungen und Rahmenbedingungen dar. Die Projektgruppe verfolgte dabei die folgende Vision, wobei das zu entwickelnde System den Namen Pacman (Power and Communication Management) trägt:

*Es wird ein Agentensystem entwickelt, das die Stromversorgung im Smart Grid im Falle eines Blackouts automatisch wiederherstellt.*

Ausgehend von dieser Vision lassen sich Ziele, die in Verbindung mit der Aufgabenstellung stehen, ableiten.

1. Es soll der Prototyp eines Verfahrens erstellt werden, das im Falle eines Blackouts die Stromversorgung wiederherstellt.
2. Der Wiederaufbau des Stromnetzes soll trotz eingeschränkter Kommunikationsfähigkeit funktionieren.
3. Das Verfahren soll dezentral realisiert werden.

Im Falle eines Blackouts bricht die komplette Stromversorgung zusammen. Um das Stromnetz wieder hochzufahren, benötigt man einerseits schwarzstartfähige Erzeuger, die den Aufbau initiieren können und muss andererseits sicherstellen, dass die Leistungsbilanz des Stromnetzes während des gesamten Vorgangs ausgeglichen bleibt. Das erste Ziel besagt, dass der entwickelte Prototyp, bzw. das Verfahren dazu in der Lage ist, das Stromnetz automatisch wieder hochzufahren und zusammenzuschließen, ohne dass es zu einer unausgeglichenen Leistungsbilanz kommt.

Da nach einem Blackout die Kommunikationsfähigkeit stark eingeschränkt ist, gibt das zweite Ziel vor, dass der Wiederaufbau des Stromnetzes auch trotz eingeschränkter Kommunikationsfähigkeit funktionieren sollte. Die eingeschränkte Kommunikationsfähigkeit bedeutete für die Projektgruppe zunächst, dass nicht alle Knoten im Netz miteinander kommunizieren können, sondern jeder Knoten nur eine eingeschränkte Kommunikationsreichweite hat.

Mit dem letzten Ziel ist gemeint, dass keine zentrale Einheit den Wiederaufbau koordiniert, sondern dieser durch den Einsatz von Agenten dezentral an verschiedenen Stellen des Stromnetzes initiiert und durchgeführt wird. Das bedeutet, dass Agenten von selbst an verschiedenen Stellen des Netzes aufwachen und dezentral mit dem wieder Hochfahren des Stromnetzes beginnen.

### 1.3 Aufbau des Berichts

Der vorliegende Bericht der Projektgruppe soll bezüglich der ihr gesetzten Aufgabenstellung, die in Abschnitt 1.1 vorgestellt wurde, vor allem aufzeigen, wie gearbeitet, was geleistet und wie es umgesetzt wurde. Um diesen Ansprüchen gerecht zu werden wird zunächst in Kapitel 2 auf das interne Projektmanagement eingegangen. Es zeigt insgesamt die organisatorischen Rahmenbedingungen der Projektgruppe auf und beinhaltet die Vorstellung des verwendeten Vorgehensmodells, Ausführungen über die Organisation des Projektablaufs und der Infrastrukturwerkzeuge, von denen Gebrauch gemacht worden ist.

Anschließend befasst sich Kapitel 3 mit der Spezifikation der Problemstellung, mit der die Projektgruppe konfrontiert wurde. Neben der Sichtung von Related Work liegt der Fokus des Kapitels auf der Szenariodefinition, welche zu Beginn der Projektphase vorgenommen wurde, der Anforderungserhebung sowie den verschiedenen Optimierungszielen, die insgesamt die Problematik näher spezifizieren. Darüber hinaus wird COHDA als verwendete Heuristik vorgestellt.

In Kapitel 4, Algorithmische Umsetzung, wird der Gesamtprozess des Verfahrens skizziert. Das von der Projektgruppe entwickelte Zustandskonzept wird hier beleuchtet und der Ablauf des Systems für die unterschiedlichen Agenten dargestellt. Der dritte und abschließende Teil des Kapitels zeigt das Verfahren hinsichtlich des Gesamtprozesses detailliert auf.

Kapitel 5 beschreibt die zugrundeliegende Systemarchitektur und geht auf die Implementierung des Prototyps näher ein. Die verwendeten Designentscheidungen und deren Alternativen werden unter Erläuterung der Architektur aufgezeigt. Dabei stehen ebenso die Auswahl eines geeigneten Multiagentensystems und die Modularisierung

des Systems im Vordergrund. Es folgt keine Beschreibung der verwendeten Algorithmen wie in Kapitel 4, sondern eine begründete Erklärung der Umsetzungsstruktur solcher.

Nachdem die Implementierung vorgestellt wurde, befasst sich Kapitel 6 mit dem Testen des Systems. Auf Basis der Recherche zur Theorie von Tests werden hier die Eingabeparameter des Systems identifiziert und zu Testfällen kombiniert. Diese von der Projektgruppe durchgeführten Tests werden in diesem Kapitel unter Zuhilfenahme von Äquivalenzklassen vorgestellt. Anschließend wird im Detail auf die Ergebnisse dieser, auch in Bezug zu den ermittelten Anforderungen, eingegangen.

Im Kapitel 7 wird die durchgeführte Validierung beschrieben. Dazu wird zunächst der Versuchsaufbau erläutert, wobei hierbei die zu untersuchenden Validierungskriterien die Basis darstellen. Darauf aufbauend wird auf die Durchführung der Validierungsphase näher eingegangen. Analog zu Kapitel 6 werden die Ergebnisse der Validierung vorgestellt und ein entsprechendes Fazit gezogen.

Anschließend wird in Kapitel 8 ein Fazit der gesamten Projektgruppe gezogen. Dies beinhaltet eine Zusammenfassung des Projektverlaufs, einen Rückblick der Projektgruppe sowie eine kritische Beurteilung des erstellten Produkts. In der zweiten Hälfte wird ein Ausblick auf mögliche Erweiterungen des Produkts gegeben.

## 2. Projektmanagement

In diesem Kapitel wird beschrieben, wie Projektmanagement innerhalb der Projektgruppe organisiert und gelebt wurde. Dazu werden im Folgenden das Vorgehensmodell, die Projektorganisation und der Projektablauf, wozu die Projektsitzungen, die Zeitplanung, das Rollenmodell, die Arbeitsgruppen und das Qualitätsmanagement gehören, geschildert. Im Anschluss werden die verwendeten Softwarewerkzeuge, die die Projektgruppe als Mittel zur Infrastrukturbildung im Projektalltag benutzt hat, vorgestellt.

### 2.1 Vorgehensmodell

Eine der ersten Herausforderungen zum Projektbeginn war die Wahl eines geeigneten Vorgehensmodells, das die Projektgruppe durch den Projektalltag führt. Da es die grundlegende Idee der Projektgruppe war, zunächst einen Prototyp zu entwickeln, der in mehreren Iterationsschritten ausgebaut und verfeinert wird, fiel die Wahl des Vorgehensmodells auf Prototyping. Dieses wird als gängige Methode in der Softwareentwicklung genutzt und beinhaltet die kontinuierliche Entwicklung und Verbesserung einer lauffähigen Software in mehreren Stufen.

Der Grund für die Wahl des Prototypings als Vorgehensmodell ist die Tatsache, dass die Projektgruppe eine bessere Gestaltung der arbeitsspezifischen Organisation, unter anderem durch kontinuierliches Einbinden der Betreuer im Sinne einer stetigen Präzisierung und Verifizierung des Produktes, anvisiert. Somit wird versucht, Fehlentwicklungen bei der Softwareentwicklung entgegenzuwirken. Ausführungen über die konkrete Umsetzung des Vorgehensmodells werden in Abschnitt [2.2.3](#) vorgestellt.

### 2.2 Projektorganisation und -ablauf

Der folgende Abschnitt geht auf die Organisation und den Ablauf der einjährigen Projektarbeit ein und zeigt somit auf, wie sich die Projektgruppe in dem Zeitraum vom Projektbeginn bis zur Vorstellung des finalen Prototyps und Abgabe des vorliegenden Berichts organisiert hat.

### 2.2.1 Seminar- und Projektphase

Das erste gemeinsame Treffen und gegenseitige Kennenlernen der Projektmitglieder fiel auf den 7. April 2016 im Rahmen der Einführungsveranstaltung zur Projektgruppe „Blackout – Restart“, an dem wichtige Informationen zur Durchführung ausgetauscht und Inhalte des Projekts vorgestellt worden sind. Gleichzeitig leitete dieses Treffen die Seminarphase ein, welche als Bestandteil des Projektes für jedes Projektmitglied und als Maßnahme zur Grundlagenbildung und Einarbeitung in projektspezifische Themenbereiche durchgeführt wurde. Inhalt der Seminarphase spiegelten Ausarbeitungsthemen wieder, die in Höhe der Anzahl der Projektmitglieder von den Betreuern vergeben wurden. Nach vier Wochen der Einarbeitung in die von den Betreuern vergebenen Themen wurden gegen Ende der Seminarphase am 17. Mai 2016 in einer halbtägigen Seminarblockveranstaltung die Präsentationen der Projektmitglieder gehalten. Die Ausarbeitung wurde mittels  $\text{\LaTeX}$  verfasst und beinhaltete einen Rahmen von maximal zehn Seiten. Die Ergebnisse der Seminarphase können in Anhang E eingesehen werden. In dieser Zeit fanden kurz vorher bereits Betreuergespräche zur Besprechung der Ausarbeitungen statt und der Beginn der Projektphase wurde am 17. Mai offiziell eingeleitet.

In der Projektphase lag der Fokus auf die Selbstorganisation der Projektgruppe. Ab dieser Phase galt es ihr allein, sich zu organisieren, um die Aufgabenstellung zu bewältigen. Da hierbei die eigentliche Projektarbeit startete und diese Phase den größten Zeitraum von ca. 10 Monaten in Anspruch nahm, bezieht sich der vorliegende Bericht auf diese. Die nachfolgenden Abschnitte zeigen im Detail, wie die Projektphase im internen Projektmanagement organisiert und gelebt wurde.

### 2.2.2 Projektsitzungen

Im Mittelpunkt des Projektablaufs und der Projektorganisation standen die regelmäßigen Sitzungen, die in interne Treffen und Betreuertreffen untergliedert wurden. Das Betreuertreffen fand während der Projektphase wöchentlich statt und beinhaltete die Vorstellung des aktuellen Arbeitsstandes der Projektgruppe. Verbindlich für das Betreuertreffen waren die Organisation in Protokollant und Sitzungsleiter, Anfertigung einer Agenda und Abgabe des Protokolls der letzten Sitzung. Sitzungsleiter und Protokollant wurden in der ersten Sitzung festgelegt und dann jede Woche von einem anderen Projektmitglied alphabetisch abgelöst. Der Protokollant der aktuellen Woche wurde dabei automatisch Sitzungsleiter der vorausgehenden Woche und übernahm die Verantwortung der Verteilung der Agenda und des Protokolls über den von den Betreuern eingerichteten Mailverteiler, was spätestens am Abend zwei Tage vor dem Betreuertreffen erledigt wurde.

Das Betreuertreffen folgte einem üblichen Ablauf, nach dem der Sitzungsleiter zunächst die Anwesenheit überprüft und den Protokollanten der Sitzung nennt. An-

schließlich wurden Anmerkungen zum Protokoll des letzten Betreuertreffens aufgenommen und dieses nachfolgend verabschiedet. Nachdem im weiteren Verlauf eventuell anstehende Ergänzungen zur Tagesordnung besprochen wurden, starteten üblicherweise die Berichte der vorherigen Woche. Hierbei stellten anfangs die jeweiligen Arbeitsgruppen und später einzelne Projektmitglieder in einem Kurzbericht ihren aktuellen Arbeitsstand bzw. die Berichte der letzten Woche vor. Einzelne Themen, die konzeptionell neu ausgearbeitet worden sind, wurden als Unterpunkt der Berichte der letzten Woche in gesonderten Punkten aufgelistet und vorgestellt. Nachdem alle für die Projektmitglieder relevanten Aspekte für den Wochenbericht vorgestellt und das Feedback der Betreuer eingeholt worden ist, wurden anstehende Arbeitspakete für die nächste Woche vorgestellt.

Die internen Treffen stellen alle Treffen, die abseits des Betreuertreffens innerhalb der Projektphase von der Projektgruppe durchgeführt worden sind, dar. In der ersten Hälfte der Projektphase fanden interne Treffen geplant, regelmäßig und im gesamten Team statt. Im Zeitraum um die Abgabe des Zwischenberichts im September fanden arbeitsgruppenorientierte Treffen statt, die keiner regelmäßigen Planung unterlagen und auch nicht im gesamten Team gehalten worden sind.

An den regelmäßigen Sitzungen in der ersten Hälfte der Projektphase traf sich die Projektgruppe zwecks Planung des weiteren Projektverlaufs. Für die internen Treffen wurde von verschiedenen Mitgliedern ein bis zwei Tage vorher eine Agenda erstellt, die ähnlich dem Betreuertreffen aufgebaut war. Sitzungsleiter war hierbei ein beliebiges Projektmitglied.

In der zweiten Hälfte wurden interne Treffen anderweitig organisiert. Diese bestanden aus dem Koordinatorentreffen, das der weiteren Koordination der Projektplanung diente und an dem sich lediglich die jeweiligen Hauptverantwortlichen der anstehenden Themengebiete trafen, und der normalen internen Sitzung, an der jedes Projektmitglied teilnahm. Einberufen wurden beide Sitzungen vom Projektkoordinator, welcher in Abschnitt 2.2.4 vorgestellt wird und durch die jeweiligen Sitzungen führte. Inhaltlich fielen überwiegend Themen an, die dem weiteren Voranschreiten des Projektes dienten und es wurde über anfallende bzw. geleistete Arbeitspakete diskutiert.

### 2.2.3 Zeitplanung

Das in Abschnitt 2.1 vorgestellte Vorgehensmodell unterlag einer der Projektgruppe ausgearbeiteten Zeit- und Meilensteinplanung, die in die unterschiedlichen Prototypen aufgeteilt wurde. Im späteren Verlauf wurden zwecks einer agileren Arbeitsweise Sprintplanungen in den Projektalltag eingeführt. Der Zeitplan des ersten Prototyps kann exemplarisch in Abbildung D.1 eingesehen werden.

In ihm spiegelt sich das Vorgehensmodell des Prototypings wieder. Der endgültige Plan sah den ersten Zyklus von Anforderungs- und Entwurfsphase vor, um einen ersten Prototypen bis Ende September zu entwickeln. Der Projektplan in Abbildung D.1 zeigt strukturiert auf, wann die Übergänge in verschiedene Projektphasen zu realisieren waren. Demnach bildete die Anforderungsphase, die von Anfang Mai bis Mitte Juli ausgelegt worden ist, neben der Entwurfsphase, welche im Zeitraum von Mitte Juni bis Ende August behandelt wurde, den längsten Zeitraum. Die Entwicklung des ersten Prototypen fiel auf Mitte August bis Ende September und konnte terminlich eingehalten werden. Allgegenwärtig wurde die Phase der Dokumentation behandelt: Sie begleitete das Projektgeschehen kontinuierlich und sollte sicherstellen, dass Zwischenergebnisse in den Bericht aufgenommen worden sind. Ein weiteres Element der Zeitplanung war das Setzen von Meilensteinen. Der Übersicht halber wurde hierfür ein entsprechendes Dokument erstellt, welches in Abbildung D.2 für den ersten Prototypen exemplarisch zu sehen ist.

Die zwei nachfolgenden Prototypen unterlagen analog zum ersten Prototypen einer Zeit- und Meilensteinplanung. Ein Ausblick auf die Zeitplanung des dritten Prototyps ist in Abbildung und 2.1 zu sehen. Im Vergleich zur Zeitplanung des ersten Semesters in Abbildung D.1 wurde hier noch detaillierter in die einzelnen Themen geplant und entsprechende Meilensteine gesetzt.

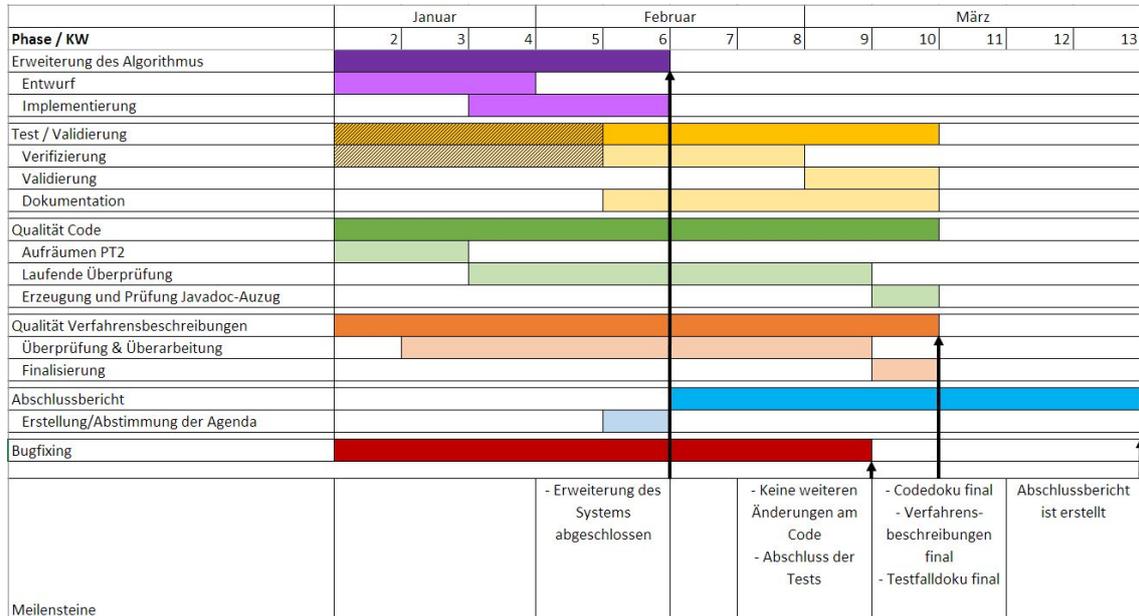


Abbildung 2.1: Zeitplanung des dritten Prototyps

Des Weiteren wurden Sprintplanungen ab der zweiten Hälfte der Projektphase in den Projektalltag integriert. So wurde gemeinsam ab der Arbeit an dem zweiten Prototyp wöchentlich eine Sprintplanung ausgearbeitet. Das weitere Vorgehen wurde zunächst im Koordinatorentreffen, welches in Abschnitt 2.2.2 vorgestellt wurde,

besprochen und gleichzeitig Arbeitspakete geschnürt. In der darauffolgenden Sitzung, dem gemeinsamen internen Treffen, wurden diese Arbeitspakete, die für diese Woche zu erledigen sind, an die Projektmitglieder verteilt. Dabei wurde darauf geachtet, dass jeder in etwa gleiche Ressourcen für seine Aufgabenpakete einzusetzen hat. Der Fortschritt der Umsetzung der Sprintplanung der einzelnen Projektmitglieder handhabte die Projektgruppe mittels Statusmails, die von jedem einzelnen geschrieben und in denen über den jeweiligen Arbeitsstand der jeweiligen Person berichtet wurde. So konnte zum einen gewährleistet werden, dass sich die Projektkoordination mit der weiteren Planung der nächsten Sprintplanung befassen konnte, zum anderen, dass jedes Projektmitglied den allgemeinen Arbeitsstand kennt. Diese agile Arbeitsweise mittels Sprintplanungen wurde bis zum Ende der Projektgruppe beibehalten.

Ein weiterer Punkt der Zeitplanung war die Urlaubsplanung. Projektmitgliedern stand es frei, innerhalb des Projektzeitraums Urlaub von insgesamt sechs Wochen zu nehmen. Weitere Überlegungen, wie die Berücksichtigung der Klausurenphase, flossen als Ergebnis in die Projektplanung hinein und beeinflussten diese.

#### 2.2.4 Rollenmodell

Ein Rollenmodell wurde im Verlauf der Projektdurchführung von der Projektkoordination ausgearbeitet, nachdem sich bereits einige Rollen und Spezialisierungen innerhalb der Projektgruppe herauskristallisiert hatten. Das Rollenmodell stellt somit auf der einen Seite die Grundlage für Koordination und Steuerung innerhalb der Projektgruppe und auf der anderen Seite die Grundlage für eine agile Arbeitsweise, die für die Projektphase angestrebt wurde, dar. Darüber hinaus sollte das Rollenmodell eine enge Verzahnung der Zuständigkeiten gewährleisten und ein Mittel zur Transparenzbildung bezüglich der Aufgabenbereiche verkörpern. Da von Betreuerseite aus der Wunsch des Austausches der Rollen der Projektmitglieder fiel, unterlag das Rollenmodell für das letzte Drittel der Projektphase insofern eine Neuverteilung der Rollen, als dass einige Rollen von den Projektmitgliedern gewechselt worden sind. Die jeweiligen Rollenmodelle sind in [Abbildung 2.2](#) und [2.3](#) dargestellt.

Für die jeweiligen Rollen wurden zeitliche Ressourcen in Prozentsätzen abgeschätzt, die als Tendenz für die jeweilige Rolle zu verstehen sind. Beispielsweise bedeutet eine Zuweisung einer Rolle mit 30 Prozent Zeitressourcen, dass vom wöchentlichen Arbeitsaufwand ca. 30 Prozent der zeitlichen Ressourcen in diese Rolle fließen sollten und die restlichen 70 Prozent beispielsweise in die Produktentwicklung. Bei einem wöchentlichen Arbeitspensum von 16 Stunden pro Woche, die sich aus den 24 Kreditpunkten des Moduls ergeben, sollten also nicht mehr als in etwa fünf Stunden für diese Rolle investiert werden.

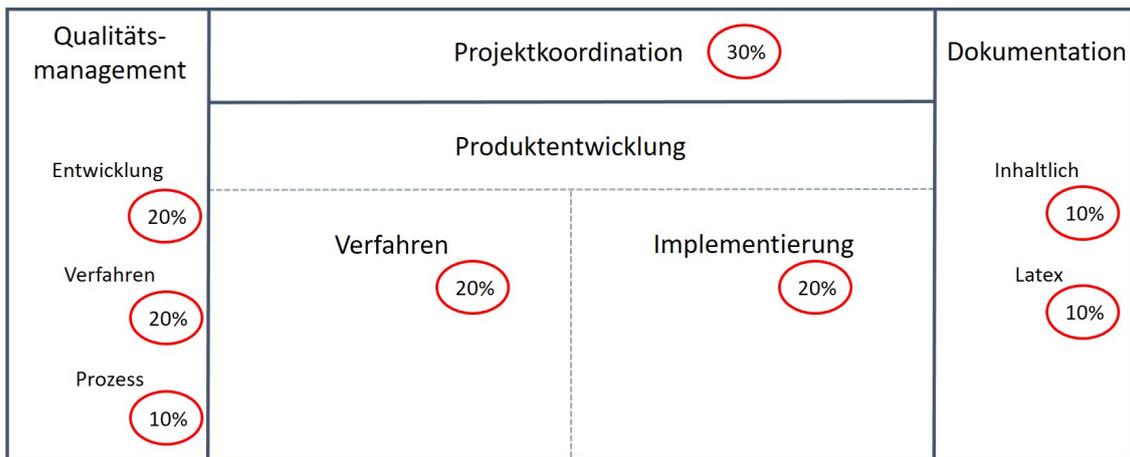


Abbildung 2.2: Rollenmodell des ersten Projektphase

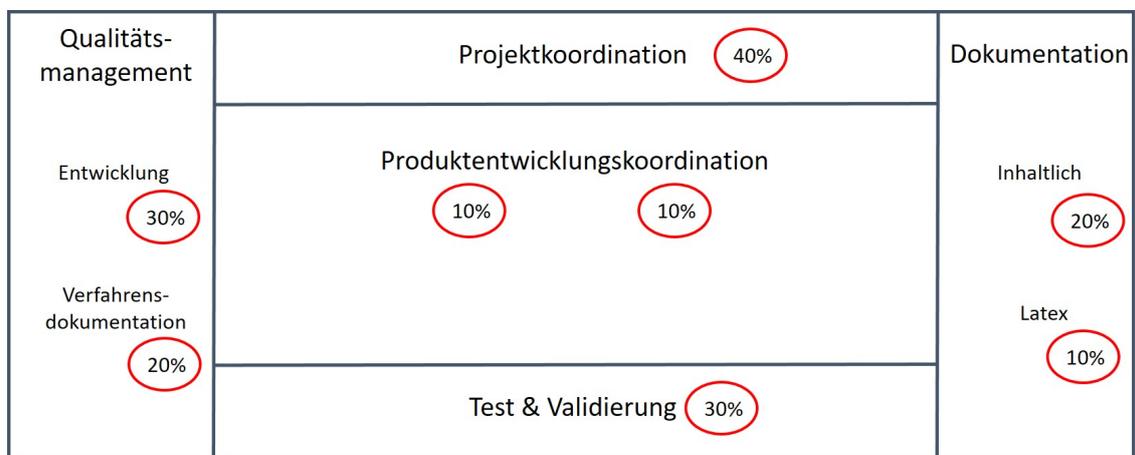


Abbildung 2.3: Rollenmodell der zweiten Projektphase

Des Weiteren fällt in den Rollenmodellen in [Abbildung 2.2](#) und [2.3](#) auf, dass nicht allen Projektmitgliedern eine Rolle zugeordnet worden ist. Dies hat zwei Gründe: Zum einen wollte die Projektgruppe nicht Rollen in Höhe der Anzahl der Projektmitglieder festmachen, zum anderen gilt das Hauptaugenmerk der Produktentwicklung, da diese für eine entsprechende Notenvergabe maßgeblich relevant ist und die vorliegenden Rollen nur als Nebenarbeit anzusehen sind. Somit ist jedes Projektmitglied aktiv an der Produktentwicklung beteiligt und muss nicht zwangsweise einer Rolle zugeordnet sein. Im Folgenden werden die Rollen der Modelle kurz beschrieben.

### Projektkoordination

Die Projektkoordination steht als übergeordnetes und übergreifendes Instrument im Rollenmodell. Der zuständige Projektkoordinator verkörpert typische Managementfunktionen wie das Planen, Steuern und Überwachen der projektspezifischen Organisation. Als Aufgaben trägt er das Ausarbeiten einer Zeit- und Ressourcenplanung, gewährleistet und koordiniert die Zusammenarbeit innerhalb der Projektgruppe, steuert und plant Teamsitzungen, führt Statustracking über spezielle Themengebiete.

te und stellt insgesamt dadurch sicher, dass die Projektgruppe mehr Kapazitäten in die Produktentwicklung stecken kann.

### **Produktentwicklungskoordinatoren (Verfahren und Entwicklung)**

Produktentwicklungskoordinatoren in den Bereichen Verfahren und Implementierung hatten zur Aufgabe, den Überblick über aktuelle und offene Themen ihres Bereiches zu behalten. Leitfragen sind hier „Was wird / wurde umgesetzt?“ und „Welche offenen Arbeitspakete gibt es?“. Darüber hinaus waren sie für die Planung und das Tracking der Erweiterungen der Prototypen verantwortlich und verglichen Anforderungen mit geplanten Arbeitspaketen. Das entsprechende Feedback wurde zur weiteren Planung in dem Koordinationstreffen (s. Abschnitt 2.2.3) gegeben.

### **Qualitätsmanagement (QM) – Entwicklung**

Der Qualitätsmanager im Bereich Entwicklung trägt die Verantwortlichkeit über die Code-Qualität. Er stellt sicher, dass der Code den Entwicklungsrichtlinien entspricht, die in Anhang C vorgestellt werden. Darüber hinaus ist er verantwortlich für die Übersicht über die Umsetzung des Algorithmus in der Implementierung. Die Leitfrage lautet hier „Wie wird etwas umgesetzt?“

### **QM – Verfahren**

Der Qualitätsmanager im Bereich Verfahren sichert die Qualität und die Vollständigkeit der Verfahrensbeschreibungen. Dazu zählt unter anderen die Pflege von Aktivitäts- und Klassendiagrammen und Feinspezifikationen. Des Weiteren trägt er die Verantwortung über die Korrektheit von UML-Diagrammen.

### **QM – Prozess**

Der Qualitätsmanager im Bereich Prozess ist dafür zuständig, das Leben der Projektorganisation sicherzustellen. Seine Aufgabe ist es, das Einhalten der definierten Prozesse, wie beispielsweise der Koordinatorentreffen, der wöchentlichen Sprintplanung, Statusmails oder das Mergen, zu überwachen. Außerdem hat der QM-Prozessbeauftragte ein Blick auf das Einhalten der Zeitplanung und der Verantwortlichkeiten.

### **Dokumentation – Inhalt**

Für inhaltliche Belange des Zwischen- und Abschlussbericht ist der Dokumentationsbeauftragte mit der Spezialisierung „Inhalt“ verantwortlich. Dieser hat zur Aufgabe eine Struktur und Gliederung für den Bericht zu erstellen, Aufgabenpakete zu schnüren und zu verteilen und ein Lektorat durchzuführen.

## Dokumentation – L<sup>A</sup>T<sub>E</sub>X

Für technische Belange bezüglich des Berichts ist der Dokumentationsbeauftragte mit der Spezialisierung „L<sup>A</sup>T<sub>E</sub>X“ verantwortlich. Er trägt die Erstellung und Pflege des L<sup>A</sup>T<sub>E</sub>X -Templates zur Hauptaufgabe und ist darüber hinaus als Ansprechpartner für Probleme und Schwierigkeiten bezüglich L<sup>A</sup>T<sub>E</sub>X zu sehen.

## Test & Validierung

Die im zweiten Rollenmodell neu eingeführte Rolle des Test- und Validierungsbeauftragten beinhaltet die Verantwortung für die Themen Test und Validierung im Gesamten. Er trägt sowohl Verantwortung für die Planung und Durchführung der Tests im Rahmen der Validierung, als auch für die Dokumentation solcher. Der Test- und Validierungsbeauftragte soll sich bei der Validierung an vorher bestimmten Validierungskriterien orientieren.

### 2.2.5 Qualitätsmanagement

Bestandteil eines jeden Projektes ist ein Qualitätsmanagement, das auch die Projektgruppe in ihre Organisation einbettete. Es wurden demnach Qualitätsstandards in Verbindung mit einem internen Qualitätsmanagement definiert. Mit ihnen konnte sichergestellt werden, dass entsprechend definierte Rollen ihre Aufgabe wahrnehmen können und dass die erforderliche Qualität bei der Entwicklung sichergestellt wird. Zu den geschaffenen Qualitätsstandards zählen Programmierkonventionen, die von dem Qualitätsbeauftragten der Entwicklung erstellt worden sind, und Qualitätsrichtlinien im Bereich QM-Verfahren und QM-Prozess.

## 2.3 Infrastruktur

Für den erfolgreichen Projektverlauf benötigt es innerhalb des internen Projektmanagements entsprechende Infrastruktur-Einrichtungen, die das tägliche Arbeiten im Team ermöglichen und fördern. Nachfolgend werden die Werkzeuge, welche hauptsächlich von der Projektgruppe benutzt wurden, im Allgemeinen erläutert und die Beweggründe, die zur Wahl des Hilfsmittels führten, erörtert.

### Git

Als sinnvolle Tools zur Unterstützung von Softwareentwicklung haben sich Versionsverwaltungssysteme etabliert. Git ist ein freies, dezentrales Verwaltungssystem, das sich von typischen Versionskontrollsystemen, wie SVN, unterscheidet. Der Hauptunterschied liegt bei Git darin, dass das Repository nicht nur auf einem zentralen Server liegt, sondern jeder Nutzer eine lokale Kopie des Hauptrepositorys (master)

auf dem lokalen Rechner besitzt. Darüber hinaus bestehen im Vergleich zu üblichen Verwaltungssystemen die Funktionen `branching` und `merging`, mit denen der Nutzer beliebige Entwicklungszweige erstellen bzw. ineinander verschmelzen kann. Dazu besteht die übliche Funktion der Versionsgeschichte (`history`), mit dessen Hilfe verschiedene Versionen abgerufen werden können.

Die Projektgruppe hat sich im Rahmen des Programmierens und der Erstellung des Berichts auf Git geeinigt, da bereits einige Projektmitglieder Erfahrung mit Git sammeln konnten und Git die Möglichkeit bietet, eigene lokale Repositories anlegen zu können. Der Vorteil besteht für die Projektgruppe darin, dass jedes Mitglied beim Programmieren eigene Repositories anlegt und damit von überall aus lokal arbeiten kann. Zudem erweisen sich im Projektalltag das Branching und Mergen beim Programmieren als hilfreiche und nützliche Werkzeuge, die von der Projektgruppe nicht gemisst werden wollen.

### **L<sup>A</sup>T<sub>E</sub>X**

L<sup>A</sup>T<sub>E</sub>X ist ein Textsatzsystem, das eine Erweiterung zu dem Textsatzsystem TeX darstellt und eine Möglichkeit schafft, wissenschaftlich-technische Texte in Buchdruckqualität zu erzeugen. Das Programmpaket L<sup>A</sup>T<sub>E</sub>X schafft im Vergleich zu TeX eine benutzerfreundlichere Zwischenebene zwischen Benutzer und Programm und basiert auf das „What You See Is What You Mean“-Prinzip. Mit Hilfe von L<sup>A</sup>T<sub>E</sub>X wird ein Benutzer ohne spezielle Vorkenntnisse in die Lage versetzt, die Möglichkeiten von TeX zu nutzen und bereits nach kurzer Zeit anspruchsvolle Textausgaben in Buchdruckqualität zu erzeugen. Vor allem im Bereich der Erzeugung von komplexen Tabellen und mathematischen Formeln hat sich L<sup>A</sup>T<sub>E</sub>X bewährt.

Die Verwendung von L<sup>A</sup>T<sub>E</sub>X wurde zu Projektbeginn an von den Betreuern als Abgabekriterium festgelegt. Es wurde somit als Werkzeug für die Anfertigung des Zwischen- bzw. Berichts von der Projektgruppe benutzt.

### **Jira**

Als Projektmanagement-, Problembehandlungs- und Fehlerverwaltungstool bietet die Webanwendung JIRA die Möglichkeit, Tickets zu erstellen, die das Planen, Steuern und Verwalten von Sprints ermöglichen. Zudem lassen sich alle Projektvorgänge übersichtlich in Boards darstellen, womit man einen groben Gesamtüberblick der Arbeitspaketverteilung und dem Projekt als solches erhält.

Zur Auswahl eines geeigneten Projektmanagementtools standen der Projektgruppe sowohl Redmine als auch Jira zur Verfügung. Die Wahl fiel hierbei auf Jira, da die Vorteile der Übersichtlichkeit, vielfältigen Optionen und Bedienbarkeit des Tools die Vorteile von Redmine überwiegen.

Im Verlauf des Projektes hat sich ergeben innerhalb der Projektgruppe ergeben, dass die Sprintplanungen außerhalb von Jira vorgenommen worden sind. Jira wurde somit in der zweiten Hälfte der Projektphase nicht mehr genutzt.

## SVN

Apache Subversion, auch SVN genannt, stellt eine kostenfreie Software zur zentralen Versionsverwaltung von Dateien und Verzeichnissen dar. Mittels SVN-Client, wie beispielsweise TortoiseSVN, bietet es die Möglichkeit, im Kontextmenü des Windows Explorer beliebige Ordner für ein SVN Repository festzulegen. Das Arbeiten mittels SVN erfolgt dann ausschließlich über das Kontextmenü des Explorers, sodass keine zusätzlichen Programme gestartet werden müssen. Grundsätzliche Methoden zur Versionsverwaltung stellt SVN ebenso zur Verfügung, sodass Versionskonflikte ebenfalls mit Hilfe der Lizenz von GPL unterliegenden, freien Software lösbar sind.

Zu Projektbeginn wurde sich seitens der Betreuer und Projektmitglieder darauf geeinigt, die Agenden und Protokolle der wöchentlichen Meetings den Betreuern mittels SVN zur Verfügung zu stellen. Als formales Abgabewerkzeug entwickelte sich das Repository schnell zum Sammelpunkt von allumfassenden Informationen, die den Betreuern zur Verfügung gestellt werden sollten. So fanden beispielsweise Meetingpräsentationen, Dokumente zur Projektplanung, Requirements Engineering und der Seminarphase ihren Platz in dem Repository, das mit Hilfe von SVN verwaltet wurde. Im weiteren Projektverlauf wurde aufgrund der einfachen Bedienbarkeit der vorliegende Bericht mit Hilfe von SVN verwaltet.

# 3. Spezifikation der Problemstellung

Dieses Kapitel soll mit seinen Abschnitten die Projektgruppe und ihr Thema in den Kontext aktueller technischer Entwicklungen und verwandter Themengebiete einordnen und die Problemstellung der Projektgruppe darstellen. Hierzu wird zunächst im folgenden Abschnitt beleuchtet mit welchen verwandten Themengebiete sich die Projektgruppe während der Bearbeitungszeit beschäftigt hat. Im weiteren Verlauf des Kapitels wird in Abschnitt 3.2 die Szenariodefinition betrachtet, sowie die Erhebung der Anforderungen in 3.3. In 3.3.2 werden die Anforderungen definiert.

## 3.1 Related Work

Dieses Kapitel beschäftigt sich mit Themen und Forschungsgebieten, welche direkt oder indirekt im Zusammenhang mit dem Thema dieser Projektgruppe stehen. Dazu zählen vor Allem Smart Grids und Multiagentensysteme aber auch Simulationsumgebungen. Zum Beginn der Projektgruppe wurde über diese Themen ein Seminarband E erstellt, welcher diesem Abschlussbericht anhängt. Dieser Abschnitt soll das Thema der Projektgruppe in Kontext zu den bearbeiteten Themenfeldern aus der Seminarphase setzen und helfen, das Thema der Projektgruppe im gesamthematischen Kontext einzuordnen. Dabei greift dieser Abschnitt exemplarisch einige der zentralen Themenschwerpunkte der Projektgruppe auf und geht zum Teil auch über die Themen der Seminarphase hinaus.

Die allgemeine Entwicklung in der Energieerzeugung weg von großen zentralen Kraftwerken, hin zu vermehrt kleinen dezentralen Erzeugungseinheiten im Stromnetz ist Ausgangspunkt für die Entwicklung von Smart Grids. Als Smart Grid bezeichnet man den Einsatz neuer Technologien im Betrieb des Energienetzes. Diese neuen Technologien zeigen sich insbesondere in Form von vernetzten Mess- und Steuerelementen, wie etwa Smart Meter also Stromzähler, deren Messdaten aus der Ferne über das Kommunikationsnetz abgerufen werden können. In Kombination mit der Entwicklung hin zu dezentralen Erzeugungsanlagen und der fortschreitenden Entwicklung im Bereich der Multiagentensysteme, liegt ein kombinierter Ansatz zur Nutzung dieser Technologien nahe.

Dieser Ansatz findet durch die agentenbasierte Netzplanung Ausdruck, bei der Erzeuger und Verbraucher im Netz durch Softwareagenten repräsentiert werden. Diese Softwareagenten übernehmen dann die Steuerung der Energieanlagen etwa durch Auswahl eines bestimmten Fahrplans für eine dezentrale Erzeugungseinheit [RSLK15]. Zur Koordination der Softwareagenten in verteilten Systemen gibt es verschiedene Verfahren wie den in der Projektgruppe verwendeten COHDA-Algorithmus [Himr14]. COHDA ist eine Heuristik zum Lösen kombinatorischer Optimierungsprobleme und wird in der Projektgruppe genutzt, um eine ausgeglichene Leistungsbilanz im Stromnetz zu gewährleisten. Die Ausgeglichenheit der Leistungsbilanz ist unerlässlich, da Energienetze nur innerhalb ihrer Belastungsparameter ordnungsgemäß funktionieren [Grü14].

Die Versorgung mit Energie und das Bereitstellen von Kommunikationsmöglichkeiten, sind Teil der kritischen Infrastrukturen, deren Ausfall die öffentliche Sicherheit gefährden oder andere schwerwiegende Folgen haben kann [BSBB17]. Eine automatisierte Wiederherstellung mit einem dezentral organisierten Multiagentensystem, wie die Projektgruppe es entwickelt, soll im Falle eines Blackouts die Dauer dieses Ausfalls minimieren.

Die Projektgruppe hat sich aber auch mit verschiedenen Szenarien für den Regelbetrieb von Stromnetzen beschäftigt, wie sie auch in [Umwe10, 6] im Szenario "Lokal-Autark" beschrieben werden, um daraus Konzepte für den Wiederaufbau abzuleiten. Im genannten Szenario geht es um den stabilen Betrieb von sich selbst versorgenden Inselnetzen, also gekapselte Netzabschnitte ohne Verbindungen in andere Stromnetze. Ein solcher Betriebsfall ist auch in zweierlei Hinsicht für den Wiederaufbau nach einem Blackout relevant. Im ersten Fall bilden sich durch den dezentralen Wiederaufbau zunächst autarke Inselnetze, welche für sich stabil funktionieren müssen, bevor sie sich zu einem großen Netz verbinden. Der zweite Fall setzt an genau dieser Stelle an, nämlich wenn das Zusammenschalten zu einem Gesamtnetz nicht möglich ist. In diesem Fall muss das Inselnetz weiter autark bestehen können, bis das Hindernis für den Zusammenschluss beseitigt ist.

Für den automatisierten Aufbau eines Smart Grids mit einem Multiagentensystem wurden bereits Ansätze simuliert und evaluiert, wie in diesem Paper [GRST+10] dargestellt wird. Bei der betrachteten Umsetzung wurde verschiedene Agentenklassen für die Gerätetypen im Stromnetz gewählt, wie etwa "Device-Agents", welche bestimmte Geräte im Stromnetz, wie Leistungsschalter, darstellen. Für die Implementierung wurde im betrachteten Anwendungsbeispiel das Multiagentenframework JADE genutzt da dieses den Standard der FIPA nutzt [GRST+10]. FIPA steht für The Foundation for Intelligent Physical Agents und ist eine non-profit Organisation, welche einen Standard entwickelt hat, der die Zusammenarbeit und den Datenaus-

tausch und somit die Interoperabilität verschiedener Multiagentensysteme ermöglichen soll. Aus diesen Bemühungen ist das FIPA Agent Interaction Protocol Suite Model entstanden, welches die Nachrichten zwischen den Agenten auf semantischer Ebene betrachtet [Posl07].

Die Erprobung solcher Multiagentensysteme findet zunächst in Simulationsumgebungen statt, welche entsprechend auf die Anforderungen von Smart Grids abgestimmt sind. Da es sowohl für Kommunikations- als auch Stromnetze bereits erprobte und bewährte Simulationsumgebungen gibt und verschiedene Multiagentenframeworks keine einheitlichen Schnittstellen für die Anbindung von Simulationssoftware haben, sind Entwicklungen von Co-Simulationsframeworks zu beobachten. Solche Co-Simulationsframeworks ermöglichen es, die Simulationsschritte verschiedener Simulationsumgebungen zu synchronisieren und die errechneten Informationen in das Multiagentensystem zu übergeben [MeOD14]. Ein Beispiel für ein solches Co-Simulationsframework ist MOSAIK, welches eine Austauschplattform für die generierten Informationen aus den Simulationsumgebungen und sonstigen angeschlossenen Programmen darstellt. Im konkreten Anwendungsfall würde die Stromnetzsimulation die für eine Entscheidung im Multiagentensystem benötigten simulierten Messdaten liefern [OFFI16]. In Verbindung mit Simulationen auf realen Netzkomponenten lassen sich so möglichst realistische Szenarien gestalten und simulieren, ohne dabei kritische Infrastrukturen zu gefährden [MeOD14].

Die hier genannten Arbeiten bilden dabei nicht den gesamten Teil der in der Literaturrecherche der Projektgruppe betrachteten Forschungsgebiete und Themenfelder ab, sondern sollen vielmehr darstellen, wie die zentralen Themen der Projektgruppe in Verbindung miteinander stehen. Im oben bereits erwähnten Seminarband finden sich detaillierte Ausarbeitungen zu einigen der in diesem Abschnitt genannten Themen, aber auch zu Forschungsbereichen, die hier nicht dargestellt wurden wie zum Beispiel zu Algorithmen aus der Graphentheorie.

## 3.2 Szenariodefinition

Die Szenariodefinition war ein wesentliches Mittel der Projektgruppe, um den Beginn der eigentlichen Arbeit vorzubereiten. Durch sie wurde die komplexe Realität soweit abstrahiert, dass sie in einem übersichtlichen Modell zusammengefasst werden konnte. Um das Szenario zu definieren, wurde in der Anfangsphase der Projektgruppe ein Workshop durchgeführt.

Die Notwendigkeit einer Szenariodefinition ist dadurch gegeben, dass es nicht möglich ist, beziehungsweise nur unter unangemessenem Aufwand, die komplette Realität, in die eine Menge Faktoren einfließen, in einem System abzubilden. Abgesehen

davon ist es für die betrachtete Problemstellung nicht notwendig, die komplette Realität abzubilden. Da das Ziel ist, wie in Abschnitt 1.2 beschrieben, ein Stromnetz nach einem Blackout wieder hochzufahren, ist es nicht Gegenstand der Untersuchung, wie zum Beispiel Nachrichten übermittelt werden. Wichtig ist lediglich, dass die Nachrichten übermittelt werden können. Es ist ebenfalls nicht von Interesse, um was für einen Akteur, wie zum Beispiel Windenergieanlage oder Photovoltaikanlage, es sich im Stromnetz explizit handelt, wichtig ist nur, ob es ein Verbraucher oder Erzeuger ist, und welche Eigenschaften er mit sich bringt. Deshalb wurden in der Szenariodefinition verschiedene Bausteine geschaffen, mit denen die Projektgruppe gearbeitet hat. Grundstein des definierten Szenarios bildet ein Graph in Form einer Small-World-Topologie. Bei einem Graphen handelt es sich um eine Struktur, die verschiedene Objekte, in diesem Fall die Akteure, mit Verbindungen zwischen den Objekten darstellt. Die Verbindungen heißen Kanten, bei den Objekten handelt es sich um Knoten. Eine Small-World-Topologie ist eine bestimmte Art von Graph, bei der ein nicht benachbarter Knoten durch eine kleine Anzahl an Schritten über andere Knoten erreicht werden kann [NCBI17]. Das liegt daran, dass bei einer Small-World-Topologie eine starke Vermaschung vorliegt. Das bedeutet, dass die Knoten nicht nur zu einem oder zwei benachbarten Knoten Kanten haben, sondern auch Kanten zu nicht direkt benachbarten Knoten bestehen können. Diese Topologie kommt häufig in Stromnetzen vor. Durch die Knoten werden Erzeuger oder Verbraucher dargestellt, durch die Kanten Strom-, beziehungsweise Kommunikationswege sowie die sich auf den Stromkanten befindlichen Leistungsschalter, welche in Tabelle 3.1 erklärt werden. Bei den Erzeugern, Verbrauchern und Leistungsschaltern handelt es sich um Akteure. Dieser Begriff wird im Folgenden synonym verwendet.

Akteur	Erklärung
Leistungsschalter	Befindet sich zwischen zwei Erzeugern oder Verbrauchern und wird benötigt, um Erzeuger oder Verbraucher in eine Insel aufzunehmen. Er befindet sich immer auf einer Stromkante. Um einen Zusammenschluss von Akteuren vorzunehmen, muss ein Leistungsschalter geschlossen werden.
Flexibler Erzeuger	Hat verschiedene positive Powerwerte zur Auswahl und kann frei entscheiden, welchen Wert er in eine Verhandlung mit einbringt.
Dynamischer Erzeuger	Hat immer einen Powerwert, den er in eine Verhandlung einbringen kann. Dieser ändert sich aber über die verschiedenen Zeitschritte. Ein dynamischer Erzeuger kann in einigen Zeitschritten auch ein Verbraucher darstellen.
Statischer Erzeuger	Hat lediglich einen positiven Powerwert zur Auswahl, dieser ändert sich auch über die verschiedenen Zeitschritte nicht.
Flexibler Verbraucher	Hat verschiedene negative Powerwerte zur Auswahl und kann frei entscheiden, welchen Wert er in eine Verhandlung mit einbringt.
Dynamischer Verbraucher	Hat immer einen Powerwert, den er in eine Verhandlung einbringen kann. Dieser ändert sich aber über die verschiedenen Zeitschritte. Ein dynamischer Verbraucher kann in einigen Zeitschritten auch ein Erzeuger darstellen.
Statischer Verbraucher	Hat lediglich einen negativen Powerwert zur Auswahl, dieser ändert sich auch über die verschiedenen Zeitschritte nicht.

Tabelle 3.1: Übersicht über die Akteure im Szenario.

In einem Smart-Grid besteht im Falle eines Blackouts eine wechselseitige Abhängigkeit zwischen der Strom- und der Kommunikationsversorgung. Auf der einen Seite wird Strom benötigt, um die Kommunikation zu gewährleisten, auf der anderen Seite müssen Erzeuger und Verbraucher kommunizieren können, um die Stromversorgung wiederherzustellen. Der oben beschriebene Graph, der das Szenario abbildet, hat aus diesem Grund zwei Arten von Kanten. Dabei handelt es sich um Stromkanten und Kommunikationskanten. Auf der Abbildung 3.1 ist ein solcher Graph zu sehen, inklusive den aktivierbaren Nachbarn.

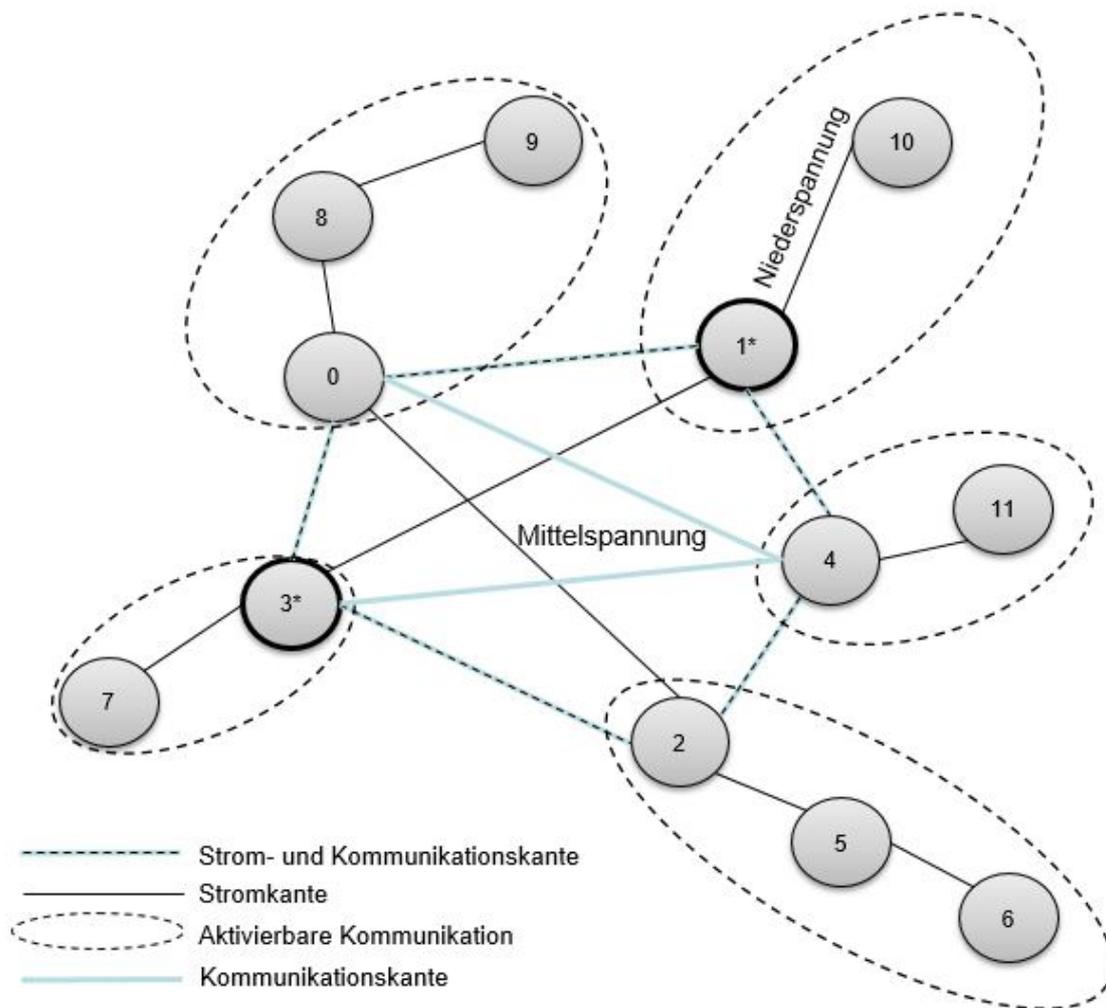


Abbildung 3.1: Mögliches Netz mit verschiedenen Kanten und aktivierbaren Nachbarn

Da das Stromnetz, wie in Abschnitt 1.1 bereits beschrieben, zu Beginn durch einen Blackout heruntergefahren ist, ist lediglich eine eingeschränkte Kommunikation möglich und es sind nicht alle Knoten über Kommunikation- oder Stromkanten erreichbar. Die sich im Netz befindenden Leistungsschalter sind zu Beginn alle geöffnet haben. Während das Netz wieder hochgefahren wird, muss die Frequenz des Netzes dauerhaft stabil bleiben, denn wenn im Stromnetz die Frequenz durch zu hohen

Verbrauch sinkt, müssten Verbraucher vom Netz genommen werden. Wenn auf der anderen Seite die Erzeugung zu hoch, oder der Verbrauch zu gering ist, steigt die Frequenz und die Erzeugung müsste reduziert werden oder der Verbrauch erhöht. In der Realität gibt es bei der Netzfrequenz minimale Toleranzen. Wenn diese auf unter 47,5 Hertz sinkt, können Generatoren beschädigt werden, wenn sie zu stark ansteigt, können elektrische Geräte beschädigt werden nach [Ampr17]. Diese Toleranzen wurden bei der Szenariodefinition allerdings abstrahiert und werden nicht betrachtet. Das bedeutet, dass die Bilanz zwischen Erzeugung und Verbrauch immer bei 0 liegen muss. Außerdem werden bei der Stromerzeugung keine Einheiten verwendet. Die Akteure haben ganzzahlige Powerwerte, die sie erzeugen oder verbrauchen können und somit keinen realistischen Leistungswerten entsprechen.

Beim Hochfahren des Stromnetzes sollen zu Beginn sogenannte Inseln gebildet werden, wie in 3.1 bereits beschrieben. Diese Strominseln müssen auch zu jeder Zeit eine ausgeglichene Energiebilanz haben. Innerhalb der Inseln herrscht vollständige Kommunikation, da durch gängige Kommunikationsprotokolle davon ausgegangen werden kann, dass Nachrichten über das Kommunikationsnetz weitergeleitet werden können. Bei der kleinstmöglichen Insel handelt es sich um einen Erzeuger, der sich mit einem Verbraucher zusammengeslossen hat. Im weiteren werden schwarzstartfähige Erzeuger, welche den Netzaufbau initiieren auch als Insel bezeichnet, obwohl sie genau genommen keine Insel sind.

Um Verbraucher oder Erzeuger in eine Insel mit aufzunehmen muss zuvor eine Verhandlung stattfinden. Bei einer Verhandlung kommunizieren die verschiedenen Akteure miteinander und bieten ihre Powerwerte an. Wenn eine Verhandlung erfolgreich abgeschlossen, die Energiebilanz ausgeglichen ist, können sich die Akteure zusammenschließen.

Eine wichtige Abstraktion der Realität ist die Zeit. Diese schreitet in der Realität kontinuierlich voran, wurde in der Projektgruppe aber abstrahiert. Statt einer gleichmäßig fortlaufenden Zeit wurden Zeitschritte eingeführt. Von diesen Zeitschritten kann es in einem zu handelnden Szenario beliebig viele geben. Die Zeitschritte vergehen im Gegensatz zur richtigen Zeit nicht gleichmäßig, sondern orientieren sich an sogenannten Steps, die durch das Multiagentensystem vorgegeben und in 5.2.1.1 erklärt werden. Wenn der letzte Zeitschritt erreicht wurde, wird wieder beim ersten Zeitschritt angefangen, die Zeitschritte der Akteure wiederholen sich also iterativ. Die Steps hingegen laufen durch das Multiagentensystem kontinuierlich fort. Des Weiteren wird auch davon ausgegangen, dass die Zukunft, was Erzeugung und Verbrauch betrifft, jederzeit bekannt ist.

Im Laufe der Projektgruppe hat die Komplexität der Kommunikationsrepräsentation zugenommen. Um einen ersten Prototypen erstellen zu können, war das Kommunika-

tionsnetz identisch mit dem Stromnetz, das Abstraktionsniveau war also sehr hoch. Dies war notwendig, um überhaupt einen ersten Arbeitsstand zu schaffen, auf dem die Projektgruppe aufsetzen konnte. Im zweiten Schritt wurde das Abstraktionsniveau etwas verringert und das Kommunikationsnetz vom Stromnetz losgelöst. So ist es möglich, dass zwischen zwei Akteuren zwar eine Kommunikationsverbindung besteht, aber keine potentielle Stromverbindung und umgekehrt. Im dritten Schritt wurde das Abstraktionsniveau noch weiter reduziert und es wurden die sogenannten aktivierbaren Nachbarn hinzugefügt. Auf Ebene der Prototypen wurden die Schritte zwei und drei im selben Prototypen gemacht. Die verschiedenen Abstraktionsniveaus der einzelnen Prototypen werden in Tabelle 3.2 dargestellt. Hierbei muss beachtet werden, dass die Prototypen nicht genau mit den oben beschriebenen Schritten übereinstimmen. Abstrahiert wurde auch das Übermitteln von Nachrichten. Diese werden im Szenario versandt und kommen garantiert auch beim Empfänger an. In der Realität kann es natürlich auch zu Verlusten von Nachrichten kommen. Da es aber nicht Gegenstand der Projektgruppe ist, die zuverlässige Übermittlung von Nachrichten zu gewährleisten, beziehungsweise ein nach Abschnitt 7.1.1.7 robustes System zu entwickeln, wird davon ausgegangen, dass die Nachrichten wie vorgesehen ankommen.

Prototyp	Abstraktionsniveau
Prototyp 1	Keine Zeit/Zeitschritte, also auch keine dynamischen Verbraucher/Erzeuger. Kommunikationsnetz ist gleich dem Stromnetz. Noch keine Leistungsschalter vorhanden. Keine unterschiedlichen Priorisierungen.
Prototyp 2	Zeitschritte und dynamische Verbraucher/Erzeuger kommen hinzu. Leistungsschalter kommen hinzu.
Prototyp 3	Trennung von Kommunikations- und Stromnetz. Aktivierbare Nachbarn kommen hinzu. Sichtbarkeit im Kommunikationsnetz wird berücksichtigt. Kritische Infrastruktur wird durch Knotenprioritäten berücksichtigt.

Tabelle 3.2: Abstraktionsniveau der verschiedenen Prototypen

Im definierten Szenario teilt sich das Stromnetz in zwei Bereiche auf. Zum einen gibt es Mittelspannungsringe, zum anderen gibt es Niederspannungsstränge. Die Niederspannungsstränge enthalten Erzeuger beziehungsweise Verbraucher, wie zum Beispiel Haushalte oder Blockheizkraftwerke. Die Niederspannungsstränge zweigen sich jeweils von Mittelspannungsringen ab und sind über diese auch indirekt verbunden. In den Mittelspannungsringen gibt es ebenfalls Erzeuger und Verbraucher. In der Realität gibt es neben der Mittelspannungs- und der Niederspannungsebene auch

noch die Hochspannungsebene sowie die Höchstspannungsebene. Da die Berücksichtigung der Hoch- und Höchstspannungsebene den Rahmen der Übersichtlichkeit stark strapazieren würde, wurden sie aus Komplexitätsgründen vernachlässigt.

Die Verbraucher und Erzeuger der Mittelspannungsebene sind teilweise vermascht. Das bedeutet, dass jeder Knoten Kontakt zu mindestens drei anderen Knoten in der Mittelspannungsebene vorweisen kann und nicht nur zu seinen direkten Nachbarn. Der Vorteil der Vermaschung liegt darin, dass, sollte ein Knoten ausfallen, das Netz um ihn herum aufgebaut werden kann. Die einzelnen Knoten sind also durch Strom- sowie Kommunikationskanten miteinander verbunden. Diese Ebene stellt die zuvor erwähnte Small-World-Topologie dar. Die Knoten der Niederspannungsstränge sind zwar auch durch Stromkanten miteinander verbunden, im Gegensatz zu den Mittelspannungsringen sind hier aber keine direkten Kommunikationskanten vorhanden. Die Kommunikation wird durch aktivierbare Nachbarn aus dem Mittelspannungsring hergestellt. Dies kann sowohl durch Erzeugern als auch Verbraucher aus dem Mittelspannungsring geschehen. Wenn ein solcher Erzeuger mit in eine Insel mit aufgenommen wurde, wird allen seinen aktivierbaren Nachbarn eine vollständige Kommunikation untereinander ermöglicht, dargestellt durch einen Hypergraphen.

Damit ein Stromnetz nach einem Blackout wieder hochgefahren werden kann, werden sogenannte schwarzstartfähige Erzeuger benötigt. Einen schwarzstartfähigen Erzeuger zeichnet aus, dass dieser, wenn er abgeschaltet ist, sich unabhängig von einer externen Stromversorgung selbst wieder hochfahren kann. Bei einem Teil der im Szenario vorhandenen flexiblen Erzeuger handelt es sich daher um schwarzstartfähige Erzeuger. Diese haben gleichzeitig auch die Funktion eines Sprechers, erklärt in Abschnitt 4.2.1 inne, befinden sich in der Regel im Mittelspannungsring und starten eine Verhandlung.

Da es aufgrund der eingeschränkten Kommunikation nicht möglich ist, alle Erzeuger und Verbraucher auf einmal zusammenzuschließen, werden zunächst Inseln gebildet. Wenn eine Verhandlung erfolgreich geführt wurde, müssen die hinzugewonnenen Erzeuger und Verbraucher mit in die Insel aufgenommen werden. Dies geschieht über die Leistungsschalter, die sich auf Befehl der Erzeuger und Verbraucher öffnen und schließen. Da es in einem Stromnetz mehrere schwarzstartfähige Erzeuger an verschiedenen Stellen gibt, bilden sich mit der Zeit konsequenterweise auch verschiedene Inseln, die sich durch die sukzessive Vergrößerung näher kommen. Die Stromversorgung wird also dezentral und nicht durch eine zentrale Instanz wiederhergestellt. Wenn zwei Inseln benachbart sind, werden diese ebenfalls über das Schließen der Leistungsschalter miteinander vereinigt.

Im Szenario, wie auch in der Realität, ist die Priorität bei den Verbrauchern, als bald wieder mit Strom versorgt zu werden, verschieden hoch. Der Wiederaufbau des

Netzes sollte selbstverständlich so schnell vonstatten gehen, wie nur möglich, aus politischer Sicht unterscheiden sich die Prioritäten. Deshalb besteht im Szenario die Möglichkeit, Verbrauchern und Erzeugern eine bestimmte Priorität zuzuweisen, mit der sie in das Netz aufgenommen werden. So könnte es zum Beispiel wichtiger sein, kritische Infrastrukturen von hoher Wichtigkeit mit in das Netz aufzunehmen, als einen klassischen Haushalt.

Die Prioritäten von Verbrauchern und Erzeugern gehört zu den in Abschnitt 3.4 behandelten Optimierungszielen. Da je nach Szenario die verschiedenen Ziele unterschiedlich wichtig sind, wurde eine Gewichtung eingeführt, mit der die Optimierungsziele bezüglich ihrer Relevanz unterschieden werden können.

### 3.3 Anforderungserhebung

In diesem Abschnitt werden die verschiedenen Schritte der Anforderungserhebung beschrieben, beginnend bei den Zielen, welche die Projektgruppe zu Beginn definierte. Um die vereinbarten Ziele zu erreichen, müssen aus diesen Anforderungen abgeleitet werden, welche die wesentlichen Grundbausteine der Projektgruppe sind. Um die Ziele zu erreichen, müssen daher die Anforderungen erfüllt werden. Dabei sollen die noch folgenden Rahmenbedingungen eingehalten werden. Aus dieser Kausalität lässt sich die folgende Gliederung ableiten. Zunächst werden die Ziele der Projektgruppe definiert. Darauf folgt die Erörterung der einzelnen Anforderungen, zum Schluss folgen die Rahmenbedingungen. Dieses Anforderungsdokument stellt die Leistungsvereinbarung zwischen Betreuern und Projektgruppe dar.

#### 3.3.1 Ziele

In Abschnitt 1.2 werden bereits die Vision und die Ziele der Projektgruppe genannt. Der Vollständigkeit halber wird der Weg von eben jener Vision zu den Zielen an dieser Stelle noch einmal aufgezeigt. Das Hauptziel (Z-01) ist die Entwicklung eines Multiagentensystems, welches in den unter Abschnitt 3.2 definierten Szenarien performant und effizient ist. Diese Begriffe werden in Abschnitt 7.1.1.7 noch genauer definiert.

Das zweite Ziel (Z-02) beschränkt die Trivialität des Systems und geht von einer ebenfalls in den Szenarien definierten eingeschränkten Kommunikationsfähigkeit aus. Das dritte Ziel (Z-03) ist die Dezentralität des Systems. Das zu entwickelnde System muss die Dezentralität der Akteure berücksichtigen. Das letzte Ziel (Z-04) ist die Festlegung der Aufgabenerfüllung. In diesem Fall soll die Erfüllung der Ziele durch Entwurf und Implementierung eines Prototypen erfolgen

Z-01 Das entwickelte Verfahren funktioniert in einem gegebenen Szenario performant und effizient

Z-02 Der Wiederaufbau des Stromnetzes funktioniert trotz eingeschränkter Kommunikationsfähigkeit

Z-03 Das Verfahren arbeitet dezentral

Z-04 Die Erfüllung der Ziele Z-01, Z-02 und Z-03 wird in Form eines Prototyps umgesetzt

### 3.3.2 Anforderungen

Die oben genannten Ziele führen, wie beschrieben, unweigerlich zu Anforderungen. Im Folgenden werden dafür vier unterschiedliche Kategorien von Anforderungen aufgeführt. Dabei wird an dieser Stelle auch auf die Erfüllung der einzelnen Anforderungen verwiesen.

#### 3.3.2.1 Szenarioanforderungen

Es werden, wie in Abschnitt 3.2 definiert, bei der Entwicklung des Prototypen von der Realität abstrahierte Szenarien verwendet. Dazu werden von der Projektgruppe und den Betreuern folgende Anforderungen festgelegt.

SZ-01 Das im Szenario verwendete Stromnetz muss durch einen ungerichteten Graphen repräsentiert werden (erfüllt, siehe 3.2)

SZ-02 Das im Szenario verwendete Kommunikationsnetz muss durch einen ungerichteten Graphen repräsentiert werden (erfüllt, siehe 3.2)

SZ-03 Der Graph des Stromnetzes muss sich aus vermaschten Teilnetzen in der Mitte und strahlenförmigen Teilnetzen am Rand zusammensetzen (erfüllt, siehe 3.2)

SZ-04 Die Knotenagenten müssen Erzeuger und Verbraucher des Stromnetzes darstellen (erfüllt, siehe 3.2)

SZ-05 Auf allen Kanten des Stromnetzes müssen sich Leistungsschalter befinden (erfüllt, siehe 3.2)

SZ-06 Das Szenario muss statische Verbraucher beinhalten. Statische Verbraucher haben nur einen möglichen Leistungswert (erfüllt, siehe 3.2)

SZ-07 Das Szenario muss dynamische Erzeuger und Verbraucher beinhalten. Dynamische Erzeuger/Verbraucher haben eine Zeitreihe mit Leistungswerten (erfüllt, siehe 3.2)

SZ-08 Das Szenario muss flexible Erzeuger und Verbraucher beinhalten. Flexible Erzeuger/Verbraucher müssen eine Menge von Leistungswerten und können eine Menge von Zeitreihen besitzen (erfüllt, siehe 3.2)

SZ-09 Es werden keine Leistungsspeicher berücksichtigt

SZ-10 Das Szenario muss beim Kommunikationssystem eine Sichtbarkeit um „Insel + 1“ gewährleisten. Dies soll im ersten Prototyp umgesetzt werden (erfüllt mit erstem Prototyp, siehe 3.2)

SZ-11 Das Kommunikationsnetz muss vom Stromnetz losgelöst sein. Beide Netze besitzen unterschiedliche Topologien, die sinnvoll von der Realwelt abstrahiert werden. Die Umsetzung erfolgt im dritten Prototyp (erfüllt mit dem dritten Prototyp, siehe 3.2)

Die Szenarien aus Abschnitt 3.2 werden exakt nach diesen Anforderungen modelliert, um somit die Erfüllung der Anforderungen zu garantieren.

### 3.3.2.2 Funktionale Anforderungen bezüglich des System

Die Anforderungen an das System beschreiben, welche Funktionalitäten das System aufweisen muss, um das Verfahren und die Szenarien zu unterstützen. Bezogen auf das Pacman-System bedeutet dies, dass das System Akteure und Infrastruktur der definierten Szenarien in einem Multiagentensystem abbilden muss (S-01). Außerdem kann das Pacman-System eine grafische Darstellung des Prozesses bieten (S-03) und soll den Ablauf des Verfahrens beim Wiederaufbau der Stromversorgung protokollieren (S-02).

S-01 Das System muss die Akteure und die Infrastruktur des definierten Szenarios in einem Multiagentensystem abbilden (erfüllt, siehe Abschnitt 5.1 )

S-02 Das System soll den Ablauf des Verfahrens beim Wiederaufbau der Stromversorgung protokollieren (erfüllt, siehe Abschnitt 5.1.3.2)

S-03 Das System kann eine grafische Darstellung des Prozesses bieten (erfüllt, s. Abschnitt 5.1.3.5)

### 3.3.2.3 Funktionale Anforderungen bezüglich des Verfahrens

Der Prototyp soll mittels eines Verfahrens innerhalb eines Szenarios ein Stromnetz nach einem Blackout wiederaufbauen. Das bedeutet, dass das Verfahren die unter 3.2 definierten Szenarien beherrschen muss. Um dies zu bewerkstelligen, sind folgende Anforderungen zu erfüllen.

V-01 Das Verfahren muss die Wiederherstellung der Stromversorgung bei einem Blackout im Rahmen der Szenarien ermöglichen (erfüllt, siehe 4.3.8 und 6.4.1)

- V-02 Das Verfahren muss kompatibel zu allen in den Szenarien definierten, durch Softwareagenten vertretenen Akteuren und Infrastrukturkomponenten sein (erfüllt, siehe [6.4.1](#))
- V-03 Das Verfahren soll den Prozess des Wiederaufbaus durch die iterative Bildung und Erweiterung von Inseln steuern (erfüllt, siehe [4.3.1](#))
- V-04 Alle Agenten, die über das Kommunikationsnetz erreichbar sind, müssen am Verfahren teilnehmen (erfüllt, siehe [4.3.2](#))
- V-05 Durch Leistungsschalter verbundene Akteure müssen sich bei erreichter gemeinsamer Leistungsneutralität zusammenschalten können (erfüllt, siehe [4.3.5](#) und [4.3.7](#))
- V-06 Der Zusammenschluss von Akteuren und Inseln erfolgt mittels Schalten der Leistungsschalter im Stromnetz (erfüllt, siehe [4.3.5](#) und [4.3.7](#))
- V-07 Inseln müssen zu jedem Zeitpunkt bilanzneutral sein (erfüllt, siehe [3.4.2](#))
- V-08 Ein Optimierungsziel jedes Knotenagenten muss die Vergrößerung seiner Insel sein (erfüllt, siehe [3.4.2](#))
- V-09 Ein Optimierungsziel jedes Knotenagenten soll das Hinzugewinnen möglichst vieler Nachbarn seiner Insel sein (erfüllt, siehe [3.4.2](#))
- V-10 Ein Optimierungsziel jedes Knotenagenten kann es sein, neue Inselmitglieder mittels vorab definierter Knotenprioritäten auszuwählen (erfüllt, siehe [3.4.2](#))
- V-11 Ein Optimierungsziel jedes Knotenagenten muss die erweiterte Kommunikationssichtbarkeit seiner Insel sein (erfüllt, siehe [3.4.2](#))
- V-12 Jeder Knotenagent muss mit anderen Knotenagenten, zu welchen er eine Verbindung im Kommunikationsnetz hat, kommunizieren können (erfüllt, siehe [4.3.2](#))
- V-13 Jeder Schalter muss die Schaltbefehle der Knotenagenten, die sich am jeweiligen Ende seiner Kante befinden, empfangen können (erfüllt, siehe [5.2.2](#))
- V-14 Jeder Schalter muss die Schaltbefehle der Knotenagenten, die sich am jeweiligen Ende seiner Kante befinden, bestätigen können (erfüllt, siehe [5.2.2](#))
- V-15 Jeder Schalter muss beim Schaltbefehl schalten können (erfüllt, siehe [5.2.2](#))
- V-16 Das Verfahren muss bei einem nicht lösbaren Szenario das Ziel haben, so viele Agenten wie möglich in Inseln aufzunehmen (erfüllt, siehe [4.3.8](#))

V-17 Das Verfahren muss bei einem nicht lösbaeren Szenario das Ziel haben, Inseln zu einer möglichst großen Insel zu vereinigen (erfüllt, siehe 4.3.8)

Als zentrale Anforderung steht die Wiederherstellung der Stromversorgung in definierten Szenarien (V-01). Dabei ist es wichtig, dass das Verfahren den Prozess durch dezentrale, iterative Zusammenschlüsse von Akteuren zu Inseln und durch die Erweiterung dieser Inseln steuert (V-03). Darüber hinaus muss das Verfahren kompatibel zu den im Szenario definierten Akteuren und Infrastrukturkomponenten sein (V-02). Teilnehmer am Verfahren sind alle Agenten, die über das Kommunikationsnetz erreichbar sind (V-04). Des Weiteren müssen sich im Rahmen des Verfahrens Akteure, die im Szenario durch Leistungsschalter verbunden sind und im Zusammenschluss eine bilanzneutrale Insel bilden, zusammenschalten können (V-05). Der Zusammenschluss ist hierbei durch das Schalten von Leistungsschaltern im Stromnetz vorgesehen (V-06). Eine Weitere Anforderung bezüglich des zu realisierenden Produkts ist die konstante Gewährleistung der Bilanzneutralität innerhalb gebildeter Inseln (V-07). Im Mittelpunkt des Verfahrens stehen Knotenagenten, welche die Akteure des Stromnetzes abbilden und den Wiederaufbau organisieren. Jeder der Agenten verfolgt dabei Optimierungsziele, die es in Form von Anforderungen zu definieren und priorisieren gilt. Als oberste Anforderung – neben dem bilanzneutralen Zusammenschluss – steht hierbei das Optimierungsziel eines jeden Knotenagenten im Rahmen des Verfahrens, seine Insel maximal vergrößern zu wollen (V-08). Das Hinzugewinnen möglichst vieler Nachbarn des Knotenagenten (V-9) ist als weiteres Optimierungsziel zu verstehen. Letztendlich könnten Knotenagenten neben den dargestellten Zielen das Optimierungsziel verfolgen, neue Inselmitglieder anhand von Priorisierungskriterien zu bewerten und den möglichen Zusammenschluss mit dieser Bewertungsgrundlage initiieren oder ablehnen (V-10). Damit Knotenagenten in der Lage sind, ihre Optimierungsziele zu verfolgen, stellt die Projektgruppe Anforderungen im Rahmen des Verfahrens an die Knotenagenten. Dazu gehört, dass Knotenagenten kommunizieren (d.h. Nachrichten an andere Knotenagenten senden und von anderen Knotenagenten empfangen) können, zu denen eine Verbindung im Kommunikationsnetz besteht (V-12). Wesentlich für das Verfahren sind Schalter, die den Zusammenschluss von Inseln ermöglichen. Schalter werden durch Agenten repräsentiert und müssen Schaltbefehle, die von Knotenagenten aus gesendet werden, nicht nur empfangen (V-13), sondern auch bestätigen (V-15) und ausführen können (V-16). Das Verfahren muss sowohl mit lösbaeren, als auch nicht lösbaeren Szenarien umgehen können. Konkret heißt dies, dass das Verfahren bei einem nicht lösbaeren Szenario zum Ziel hat, möglichst viele Knotenagenten in Inseln aufzunehmen (V-16). Darüber hinaus muss das Verfahren gewährleisten, dass sich im genannten Fall Inseln zu einer größtmöglichen Insel vereinigen (V-17).

#### 3.3.2.4 Nicht funktionale Anforderungen

Die nicht funktionalen Anforderungen beschreiben die Dokumentation des Quellcodes gemäß einer von der Projektgruppe erstellten Entwicklungsrichtlinie (N-01), die verwendete Methode der Softwareentwicklung (N-02) und die Terminierung des Verfahrens bei gegebenen Szenarien (N-03). Außerdem kann für kombinatorische Optimierungen die Heuristik COHDA verwendet werden (N-05).

N-01 Der Quellcode muss gemäß einer Entwicklungsrichtlinie, welche die Projektgruppe erstellt hat, nachvollziehbar dokumentiert sein (erfüllt, siehe C)

N-02 Das System und das Verfahren sollen über mehrere Prototypen-Phasen hinweg iterativ entwickelt werden (erfüllt, siehe 2.1)

N-03 Das Verfahren muss in allen gegebenen Szenarien terminieren (erfüllt, siehe 4.3.8)

N-04 Der Algorithmus soll valide sein (erfüllt, siehe 7)

N-05 Das Verfahren kann für kombinatorische Optimierungen die Heuristik COHDA verwenden (erfüllt, siehe 4)

#### 3.3.3 Rahmenbedingungen

Die Rahmenbedingungen beschreiben die Struktur der Projektgruppe als Studentisches Projekt an der Carl von Ossietzky Universität Oldenburg. Dabei werden die individuelle Arbeitszeit der Teilnehmer (R-02), der zeitliche Rahmen (R-01) und die Abgabe des fertigen Produktes (R-04) und des Zwischenberichts (R-03) festgelegt.

R-01 Die Entwicklung des Systems findet im Rahmen einer über ein Jahr laufenden Projektgruppe der Carl von Ossietzky Universität Oldenburg statt

R-02 Die Projektgruppe besteht aus acht Mitgliedern. Das zugehörige Modul umfasst 24 KP, was einem Leistungsumfang von 720 Arbeitsstunden pro Person entspricht

R-03 Am 07.10.2016 ist ein Zwischenbericht abzugeben

R-04 Der Abgabetermin des Produktes sowie des Abschlussberichtes ist der 31.03.2017

## 3.4 Optimierungsziele

In den folgenden Abschnitten werden zunächst in der mathematischen Formulierung die nötigen Variablen, sowie deren Zusammenhänge dargestellt. Des Weiteren werden die für die Lösungsheuristik COHDA relevanten Parameter eingeführt, welche im Abschnitt 3.5 vorgestellt wird. Im Anschluss an die mathematische Formulierung wird die multikriterielle Zielfunktion des Systems in Abschnitt 3.4.2 vorgestellt.

### 3.4.1 Mathematische Formulierung

Die mathematische Formulierung eines Problems bereitet die Grundlage für die Lösung dieses Problems mit mathematischen Werkzeugen. In der Formalisierung sollten die drei im Folgenden erläuterten Grundsätze beachtet werden. Die Formalisierung muss richtig, zulässig und zweckmäßig sein. Die Richtigkeit einer Formalisierung meint die korrekte Abbildung der realen Zusammenhänge. Die Zulässigkeit beschreibt die mathematisch widerspruchsfreie Formulierung des Problems und die Zweckmäßigkeit, dass die mathematische Formalisierung des Problems aus 3.2 dazu geeignet ist, es zu lösen. Eine Formalisierung ist zweckmäßig, wenn sie genau so viele Aspekte der Realität abbildet wie nötig, im Gegenzug aber dort abstrahiert, wo dies möglich ist [Stru10]. Im Folgenden ist die mathematische Formalisierung des Problems der Projektgruppe Blackout-Restart dargestellt. Sie soll einen Überblick über die relevanten Entitäten und Mengenbeziehungen sowie der relevanten Variablen geben. Dazu werden zunächst die einzelnen Agentenarten eingeführt und auf relevante Mengenbeziehungen zwischen diesen eingegangen. Nachdem die benötigten Variablen eingeführt wurden, werden die für eine COHDA-Optimierung relevanten Parameter definiert. Eine kurze erklärende Einführung in die Heuristik COHDA, findet sich im Abschnitt 3.5 Die Definition der Zielfunktion erfolgt im Abschnitt 3.4.2.

- Agenten:

Es werden Knotenagenten und Schalteragenten unterschieden. Die Knotenagenten  $a$  werden mit dem Laufindex  $i$  indiziert, wobei  $i$  Element der natürlichen Zahlen ist.

Knotenagenten:

$$a_i \in \{a_1, \dots, a_n\} = A$$

Im Folgenden wird  $i$  die Identifikation eines Knotenagenten  $a$  darstellen.

Schalteragenten  $s$  werden mit dem Laufindex  $j$  indiziert, wobei  $j$  Element der natürlichen Zahlen ist.

Schalteragenten:

$$s_j \in \{s_1, \dots, s_m\} = S$$

Im Folgenden wird  $j$  die Identifikation der Schalteragenten  $s$  darstellen.

- Schaltzustand:

Der Schaltzustand eines Schalters kann entweder geschlossen, also 1, oder geöffnet, also 0, sein. Der Schaltzustand eines Schalters  $s_j$  wird so durch eine binäre Variable  $\tau_j$  definiert. Diese kann den Wert 0 für geöffnet und 1 für geschlossen annehmen.

- Nachbarschaft eines Knotenagenten im Kommunikationsnetz:

Die Nachbarschaft  $S$  eines Knotenagenten  $a_i$  im Kommunikationsnetz wird definiert durch die Menge der Knotenagenten mit denen er über eine beliebige Kommunikationsart direkt kommunizieren kann. Als Index der Nachbarschaft des Agenten  $a_i$  wird die Bezeichnung  $i$  des Agenten selbst genutzt.

Nachbarschaft des Agenten  $a_i$ :

$$K_i \subseteq A$$

- Nachbarschaft eines Knotenagenten im Stromnetz:

Die Nachbarschaft  $S$  eines Knotenagenten  $a_i$  wird definiert durch die Menge der Knotenagenten mit denen er eine direkte Kante teilt, auf der sich ein Schalteragent  $s_j$  befindet. Als Index der Nachbarschaft des Agenten  $a_i$  wird die Bezeichnung  $i$  des Agenten selbst genutzt.

Nachbarschaft des Agenten  $a_i$ :

$$S_i \subseteq A$$

- Nachbarschaft eines Knotenagenten im Smart-Grid:

Die Nachbarschaft  $U$  eines Knotenagenten  $a_i$  wird definiert durch die Menge der Knotenagenten mit denen er eine direkte Kante im Stromnetz teilt und mit denen er kommunizieren kann. Als Index der Nachbarschaft des Agenten  $a_i$  wird die Bezeichnung  $i$  des Agenten selbst genutzt.

Nachbarschaft des Agenten  $a_i$ :

$$U_i = S_i \cap K_i$$

- **Priorität eines Agenten:**

Die Priorität  $g_i$  eines Knotenagenten gibt dessen Priorisierung für den Wiederaufbau des Netzes an. Sie wird durch einen beliebigen positiven ganzzahligen Wert angegeben.

$$g_i \in \mathbb{N}_0$$

- **Leistungsstufen:**

Jeder Knotenagent hat eine Menge an möglichen Leistungsstufen  $P$ , wobei der Laufindex sich auch hier durch die Identifikation des Agenten ergibt.

Menge möglicher Leistungen des Knotenagenten  $a_i$ :

$$P_i = \{p_1, \dots, p_n\}$$

mit

$$p \in \mathbb{Z}$$

Der tatsächliche Wertebereich ist abhängig von der repräsentierten Entität im Szenario.

- **Ausgewählte Leistungsstufen:**

Ausgewählte Leistung des Agenten  $i$ :

$$p_i^* \in P_i$$

- **Tabuliste:**

Die Tabuliste enthält den Index der Leistungsstufe  $p_i^*$  des Sprecheragenten mit der eine Verhandlung gestartet wurde, aber in der keine neuen Agenten in die Insel aufgenommen werden konnte. Zur Umgehung lokaler Optima wird als nächste Leistungsstufe zum Start einer Verhandlung eine Leistungsstufe  $p_i$  gewählt, die die maximale Distanz zum zuvor gewählten Wert besitzt. Befindet

sich der Index einer Leistungsstufe auf der Tabuliste, kann diese Leistungsstufe für weitere Verhandlungen nicht mehr als initiale Leistungsstufe gewählt werden.

- Insel:

Eine Insel  $I$  ist ein Verbund von Knotenagenten  $a_i$ , bei denen die Schalteragenten  $s_j$  zwischen den Knotenagenten den Zustand geschlossen haben.  $I$  ist eine Insel, wenn zu allen  $a_i \in I$  unter Berücksichtigung der Nachbarschaft  $S_i$  mindestens eine Kante existiert über die alle  $a_i$  verbunden sind.

- Identifikation einer Insel:

Die Identifikation einer Insel  $I$  ergibt sich aus der Identifikation des Inselsprechers  $a_i$  und wird damit als  $I_i$  dargestellt.

- Nachbarschaft der Insel:

Die Nachbarschaft einer Insel bildet sich aus der Menge aller Agenten, welche eine direkte Stromkante mit einem oder mehreren Agenten in der betrachteten Insel haben.

Nachbarschaft der Insel  $I_i$ :

$$N_i \cup \{U_i, \dots, U_n\} \setminus I_i$$

mit

$$I_i = a_i, \dots, a_n$$

- Leistungsbilanz der Insel: Die Leistungsbilanz einer Insel  $I_i$  ergibt sich aus der Summe der ausgewählten Leistungsstufen der Agenten in der Insel.

Leistungsbilanz der Insel  $I$  (mit  $p_i^*$  von Agenten in  $I_i$ ):

$$b_i = \sum_{i=1}^n p_i^*$$

### COHDA spezifische Parameter

Um COHDA auf das Optimierungsproblem der Projektgruppe anwenden zu können, wurden einige Parameter angepasst. Andere können unverändert übernommen werden [Hirn14, 3.1]. Um die folgend genannten Parameter besser zu in einen Kontext zu setzen wird im Abschnitt 3.5 eine einführende Erklärung zu COHDA als Lösungsheuristik gegeben.

- Ziel:

In der Optimierung des entwickelten Systems 3.4.2 handelt es sich um eine Mischform aus lexikographischer Optimierung und einer gewichteten multikriteriellen Optimierungsfunktion. Das Ziel basiert auf der Definition des Wirkleistungsprodukts aus COHDA wurde aber für die Problemstellung der Projektgruppe um die Zielgewichtung erweitert. Es werden die mit einem Laufindex versehenen Gewichte der Zielfunktion versendet, um zum Einem eine Veränderung der Gewichtung der Zielfunktion und zum Anderen ein erneutes Starten der Optimierung mit ausgeschlossenen Nachbarn oder ohne Veränderung zu ermöglichen:

- $\zeta = \{a_i, g_1, g_2, g_3, X, \delta\}$
- $g_1, g_2, g_3$
- $X \subseteq N_i$
- $\delta =$  Zählvariable

Eine neue Konsensfindung kann somit ohne eine Veränderung der Optimierungsfunktion oder der Nachbarschaft, nur durch das Hochzählen der Zählvariable gestartet werden. Die Veränderung dieser Parameter obliegt allein den jeweiligen Inselfprechern.

- Fahrplanauswahl  $\omega_i$ :

Die Fahrplanauswahl in COHDA enthält den Agenten, den ausgewählten Fahrplan und einen Laufindex. So ergibt sich die Fahrplanauswahl folgendermaßen:

$$\omega_i = (a_i, p_i^*, \lambda_i)$$

- $a_i$  ist die ID des Agenten der Fahrplanauswahl.
- $p_i^*$  ist die ausgewählte Leistung des Agenten.
- $\lambda_i$  ist der Laufindex, der mit jeder Erstellung einer neuen Fahrplanauswahl hochgezählt wird.

- Konfiguration  $\Omega$ :

Die Konfiguration ergibt sich aus den aus COHDA bekannten Variablen sowie einer ergänzten Blacklist. Die Blacklist  $\beta$  enthält Knotenagenten  $a_i$  welche in der aktuellen Konfiguration ausgeschlossen werden. Im Kontext der Projektgruppe wurde noch die Nachbarschaft des Agenten hinzugefügt.

$$\beta = \{a_i, \dots, a_n\}$$

$$\Omega = \{\omega_i, \dots, \omega_n; \beta\}$$

$$\text{forall } \omega_i, \omega_j \in \Omega : i \neq j$$

$$U_i$$

- Lösungskandidat  $\gamma$ :

Der Lösungskandidat kann ebenfalls unverändert übernommen werden:

$$\gamma = (a_i, \Omega)$$

Jeder Agent  $i$  verwaltet in seinem Arbeitsgedächtnis genau einen Lösungskandidaten  $\gamma_i$ . Diesen muss er nicht notwendiger Weise selber erstellt haben.

- Arbeitsgedächtnis

Das Arbeitsgedächtnis eines Knotenagenten  $a_i$  ist definiert durch ein Tupel, welches aus Wirkleistungsprodukt, wahrgenommenem Systemzustand und Lösungskandidat besteht.

$$\kappa_i = (\zeta, \Omega_i, \gamma_i)$$

Das Arbeitsgedächtnis ist durch den zugehörigen Knotenagenten  $a_i$  veränderbar, indem er die Informationen in  $\kappa_i$  durch das Empfangen von Nachrichten aktualisiert.

### 3.4.2 Multikriterielle Zielfunktion

Die multikriterielle Optimierung wird innerhalb der lexikografischen Optimierung realisiert. Ziel A bildet dabei das Constraint ab, dass die Leistungsbilanz  $b_i$  einer Insel immer 0 sein muss. Die Ziele B, C und D werden in einer gewichteten Zielfunktion für den zweiten Schritt der lexikografischen Optimierung zusammengefasst und sind nur dann relevant, wenn sich das Ziel A nicht mit allen verhandelnden Knotenagenten umsetzen lässt, also Knotenagenten, welche noch nicht Teil der Insel sind, auf die Blacklist  $\beta$  gesetzt werden müssen, um eine ausgeglichene Leistungsbilanz zu erreichen.

- Ziel A - Ausgeglichenheit der Bilanz:

$$b_i = 0$$

- Ziel B - Maximiere die Inselgröße:

$$|I| \rightarrow \max$$

- Ziel C - Maximiere die maximal sichtbare Nachbarschaft:

$$|N| \rightarrow \max$$

- Ziel D - Maximiere die Anzahl von Priorisierten Knoten in der Insel

$$|G| \rightarrow \max$$

Überführung in eine Gesamtzielfunktion:

$$f(I_i) \rightarrow \max$$

mit

$$f = f_1; f_2$$

als lexikografische Zusammensetzung der Teilziele  $f_1$  und  $f_2$ .

Das heißt:

$$f_1(I_i) : b_i \rightarrow \min$$

$$f_2(I_i) : g_1 \cdot |I| + g_2 \cdot |N| + g_3 \cdot |G| \rightarrow \max$$

mit

$$g_1 + g_2 + g_3 = 1$$

Damit nicht gegensätzlich optimiert wird, muss  $f_2$  mit -1 multipliziert werden.

Die in der Projektgruppe erarbeitete Zielfunktion stellt die Kombination des Constraints einer ausgeglichenen Leistungsbilanz innerhalb der Insel und einer gewichteten Zielfunktion dar. Solange das Ziel A ohne den Einsatz der Blacklist erfüllt ist, werden die Gewichte der Zielfunktion nicht relevant, weil in diesem Fall immer alle Agenten in die Insel aufgenommen werden. Ist dies nicht möglich, wird die Entscheidung darüber, welche Agenten Teil der Insel werden durch die gewichtete Zielfunktion getroffen.

Dabei hat jeder Agent einen Wert für die Sichtbarkeit, die mit seiner Aufnahme gewonnen wird sowie einen ganzzahligen Wert für seine Priorität. Das Ziel der Inselgröße wird berücksichtigt, indem für die noch verfügbare Flexibilität möglichst viele Agenten aufgenommen werden. Diese Werte gehen jeweils in die Zielfunktion ein und beeinflussen somit die Entscheidung des Systems. Bei einer alleinigen Gewichtung der Priorität würden, sofern bilanziell möglich, immer die beziehungsweise der

Knoten mit der höchsten Priorität in der aktuellen Entscheidungssituation gewählt und in die Insel integriert. Analog verhält es sich mit den Zielen B und C. Im Falle einer alleinigen Gewichtung von Ziel B würden immer so viele Agenten wie möglich in die Insel aufgenommen, unabhängig davon ob sich dadurch die Sichtbarkeit für die nächste Verhandlung erweitern würde bzw. ein priorisierter Knoten integriert werden könnte. Bei einer alleinigen Gewichtung der Sichtbarkeit würde immer der Agent mit der höchsten Sichtbarkeitserweiterung in die Insel integriert. Wie die Gewichte in der Zielfunktion zu wählen sind, hängt vom betrachteten Szenario ab. Es kann durchaus sein, dass durch die Gewichtung das Netz in einer Reihenfolge wieder zusammenschaltet wird, die zu einer Sackgasse führt. Auf diese Thematik wird im Abschnitt 7.2.2.6 noch weiter eingegangen

### 3.5 COHDA als Lösungsheuristik

Der Projektgruppe wurde von Seiten der Betreuer COHDA (A Combinatorial Optimization Heuristic for Distributed Agents) als Lösungsheuristik für die Suche einer Lösung im zu entwickelnden Verfahren vorgeschlagen. COHDA soll dabei helfen, Lösungen für kombinatorische Optimierungsprobleme [Hir14] zu finden, welche im Rahmen des zu entwickelnden Verfahrens der Projektgruppe auftreten. Als ein mögliches Anwendungsbeispiel von COHDA lässt sich die prädiktive Einsatzplanung in virtuellen Kraftwerken heranziehen. Das Problem besteht hier darin, mehrere dezentrale Erzeugungseinheiten so zu koordinieren, dass diese zusammen ein vorgegebenes Wirkleistungsprodukt erbringen. Jede dezentrale Erzeugungseinheit wird dabei durch einen Agenten repräsentiert, welcher in der anstehenden Verhandlung einen lokal optimierten Fahrplan beisteuert, um im Verbund das global vorgegebene Wirkleistungsprodukt zu erbringen.

Der Verbund des virtuellen Kraftwerks wird dabei durch einen Sprecher in der Außenwelt repräsentiert. Dieser Sprecher bildet zunächst ein Overlay-Netz, in welchem die Kommunikationsnachbarn der Agenten für die anstehende Verhandlung festgelegt werden. Durch diese Einschränkung der Kommunikation auf feste Nachbarn, wird die Nachrichtenkomplexität reduziert. Dies hilft die Laufzeit des dezentral und autonom arbeitenden Systems zu verringern.

Die optimierten Fahrpläne werden mit dem internen Wissen der Agenten über die jeweilige Erzeugungseinheit, in einer lokalen Optimierung erstellt. Als lokal optimiert gilt ein Fahrplan, wenn er für die Energieanlage im Kontext des Szenarios den optimalen Betriebszustand darstellt. Für eine lokale Optimierung nutzt der Agent sein individuelles Wissen und Fähigkeiten. Das Wissen eines Agenten wird zusätzlich in privates und öffentliches Wissen unterteilt. Den privaten Anteil des Wissens eines

Agenten bilden die lokale Gütefunktion und der Suchraum, welcher alle realisierbaren Fahrpläne des Agenten darstellt.

Das öffentliche Wissen eines Agenten wird in COHDA als Arbeitsgedächtnis bezeichnet. Das Arbeitsgedächtnis enthält das global zu erstellende Wirkleistungsprodukt als Zielvorgabe, seinen wahrgenommenen Systemzustand und somit die Fahrplanauswahl der anderen Agenten, sowie einen Lösungskandidaten in welchem der Agent seine Fahrplanauswahl mit der für ihn höchsten Güte speichert. Die Fähigkeiten eines Agenten sind Wahrnehmung, Entscheidung und Aktion und werden zur lokalen Optimierung genutzt. Während der Wahrnehmung empfängt der Agent Nachrichten und die enthaltenen Informationen zum Arbeitsgedächtnis des sendenden Agenten, also Wirkleistungsprodukt, wahrgenommener Systemzustand und Lösungskandidat und aktualisiert mit diesen Informationen sein Wissen. Im Anschluss nutzt der Agent die Fähigkeit der Entscheidung und führt eine lokale Optimierung unter Berücksichtigung der lokalen und der globalen Güte durch, um so seine Fahrplanauswahl zu treffen. Der ausgewählte Fahrplan wird nun vom Agenten in seinen wahrgenommenen Systemzustand übernommen und die nun aktualisierte Konfiguration aller Fahrpläne hinsichtlich ihrer globalen Güte bewertet. Besitzt die bewertete Konfiguration eine höhere Güte als die bisher bekannte Konfiguration aus dem empfangenen Arbeitsgedächtnisses wird diese übernommen und mit der Fähigkeit Aktion an die direkten Nachbarn im Overlay-Netz versendet. Eine Verhandlung wird gestartet, indem der Sprecher im Anschluss an die Bildung des Overlay-Netzes seine Fähigkeiten Entscheidung und Aktion nutzt und somit ein initiales Arbeitsgedächtnis versendet. Eine Verhandlung ist beendet, wenn eine Konvergenz der wahrgenommenen Systemzustände erreicht ist, also kein Agent eine Konfiguration mit einer höheren Güte mehr finden kann und somit auch keine Arbeitsgedächtnisse mehr versendet werden. Eine ausführlichere Beschreibung von COHDA ist im anhängenden Seminarband [E](#) und der entsprechenden Dissertation von Dr. Christian Hinrichs [[Hinr14](#)] zu finden.

# 4. Algorithmische Umsetzung

In diesem Kapitel wird die Arbeitsweise des Systems auf konzeptioneller Ebene beschrieben. Hierzu wird im folgenden Abschnitt der Gesamt Ablauf des Systems beschrieben, ehe das erarbeitete Zustandskonzept in [4.2](#) dargestellt wird. Ab dem Abschnitt [4.3](#) werden die Zustände detailliert im einzelnen erläutert.

## 4.1 Gesamtüberblick

Die algorithmische Umsetzung und das daraus entstandene Zustandskonzept, welches in Abschnitt [4.2](#) beschrieben wird, beruhen auf der Idee, für das System einen zyklischen Ablauf zu konstruieren, sodass nach einem initialen Start keine Eingriffe von außen mehr notwendig sind, um das System am Laufen zu halten. Das Zustandskonzept wurde so entwickelt, dass für alle Phasen im Systemverlauf klar definierte Zustände und Zustandswechsel vorhanden sind. Das System ist dadurch eindeutig in seinem Ablauf nachzuvollziehen, was insbesondere bei verteilten Systemen unerlässlich ist. Um klare Ein- und Austrittspunkte für die Zustände zu definieren, wurden entry- und exit-Bedingungen für jeden Zustand deklariert.

Das System startet mit dem Erwachen eines Sprechers und der Verbundbildung mit seinen ansprechbaren Nachbarn. Anschließend führt dieser Verbund eine COHDA-Verhandlung durch, um so eine erste Insel zu bilden. COHDA wird hier wie bereits in Abschnitt [3.5](#) beschrieben genutzt, um das vorgegebene Ziel einer ausgeglichenen Leistungsbilanz im Inselnetz zu erreichen. Enthält das Netz mehrere Sprecher findet dieser Vorgang im Netz verteilt statt. Im Anschluss an eine Verhandlung wird geprüft, ob das gefundene Ergebnis gültig ist, also die Leistungsbilanz des Inselnetzes ausgeglichen ist. Ist dies nicht der Fall wird die Verhandlung neu gestartet. Für den Fall, dass eine gültige Lösung vorliegt, prüft der Sprecher, ob sich andere Inseln in der direkten Nachbarschaft befinden um mit diesen zu fusionieren. Ist dies gegeben, gibt der angefragte Sprecher seine Rolle auf und wechselt in seinen Verhalten zu dem eines normalen Agenten. Befinden sich keine weiteren Inseln in der Nachbarschaft, wechselt der Sprecher in einen passiven Zustand und wartet einen im System definierten Zeitraum, ob er von einer anderen Insel zur Inselvereinigung angefragt wird. Wenn in diesem Zeitraum keine Anfrage eingeht, wird der Verbund

wieder mit den umliegenden und ansprechbaren Agenten erweitert und eine neue COHDA-Verhandlung beginnt. Dieser Ablauf wiederholt sich, bis es keine Inseln oder Agenten mehr gibt, die in die Insel aufgenommen werden können oder bis es nicht mehr möglich ist Agenten in die Insel aufzunehmen, da keine ausgeglichene Bilanz gewährleistet werden kann.

## 4.2 Zustandskonzept

In den folgenden Abschnitten werden die Zustände, sowie deren Wechselbedingungen jeweils für einen Sprecher in 4.2.1 und für einen normalen Agenten in 4.2.2 exemplarisch erläutert. Es wurden zwei verschiedene Zustandskonzepte erstellt, damit die Zustände in sich möglichst eindeutig gegliedert und in ihrer jeweiligen Funktion klar gekapselt sind, was auf den erweiterten Aufgabenumfang des Sprechers zurückzuführen ist. Eine solche Umsetzung bietet vor Allem Vorteile für die Nachverfolgbarkeit des Systems und hilft das Verhalten eines verteilten, dezentral organisierten Systems deterministisch zu gestalten.

### 4.2.1 Sprecher

Ein Sprecher-Agent erwacht bei Systemstart und wechselt aus dem initialen SLEEP-Zustand in den Zustand FINDNEIGHBOURS. In FINDNEIGHBOURS versendet er Nachrichten an die umliegenden Agenten um abzufragen, welche davon bereit für eine Verhandlung sind. Alle im SLEEP-Zustand befindlichen Agenten melden ihre Bereitschaft zurück. Bei dieser Abfrage werden für den Verbund nur Agenten berücksichtigt, die mindestens eine Strom- und eine Kommunikationskante in die potentielle Insel haben. Agenten die zwar Nachbarn in Form von Stromkanten aber nicht ansprechbar sind, werden ebenso wenig berücksichtigt wie Agenten die zwar über Kommunikationsstränge erreichbar sind, aber keine direkte Stromkante in die Insel besitzen. Aus den Informationen aus den Antworten der Agenten ergibt sich der Verbund, für den der Sprecher im Zustand OVERLAY das Overlay-Netz erstellt. Nach der Bildung des Overlay-Netzes wechselt der Sprecher in den Zustand COHDA und startet seine lokale Optimierung. Diese lokale Optimierung bildet die Grundlage für das initiale Arbeitsgedächtnis, welches der Sprecher an die Agenten im Overlay-Netz schickt, sodass diese ebenfalls in den Zustand COHDA wechseln. Nach einem erfolgreichem Abschluss der COHDA-Verhandlung, also mit dem Ergebnis einer ausgeglichenen Leistungsbilanz, wechselt der Sprecher in den Zustand JOINISLAND und sendet eine Nachricht zum Zusammenschluss an die Agenten. Der Sprecher wechselt nach dem erfolgreichen Zusammenschluss der Insel in den Zustand FINDISLAND in welchem er nach benachbarten Inseln zur Inselvereinigung sucht. Ab hier müssen mehrere Fälle unterschieden werden.

Im ersten Fall findet der Sprecher direkt eine benachbarte Insel im WAIT-Zustand, sodass er diese für eine Inselvereinigung anfragen kann. Im Falle einer Inselvereinigung wechseln beide Inseln in den Zustand MERGEISLAND. Der angefragte Sprecher gibt im Zuge der Inselvereinigung seine Sprecherrolle ab und wechselt das Zustandskonzept entsprechend zu dem eines normalen Agenten. Die gewachsene Insel sucht nun nach weiteren Inseln zur Vereinigung. Wird eine weitere Insel gefunden, folgt der Zusammenschluss analog zum beschriebenen Prozedere.

Im zweiten Fall findet der Sprecher keine benachbarte Insel zur Inselvereinigung und wechselt nun selbst in den WAIT-Zustand in dem er nun für einen durch das System definierten Zeitraum, auf Anfragen anderer Inseln wartet. Erfolgt eine Anfrage, wechselt der Sprecher wieder in den MERGEISLAND-Zustand und der Zusammenschluss erfolgt wieder analog zum bereits beschriebenen Zusammenschluss. In diesem Fall gibt der Sprecher nun als Angefragter seine Sprecherrolle auf. Erfolgt innerhalb der Wartezeit keine Anfrage einer anderen Insel, wechselt der Sprecher in den FINDNEIGHBOURS-Zustand und sucht neue Agenten zur Inselerweiterung. Werden Agenten gefunden, beginnt der bisher beschriebene Ablauf erneut. Werden keine Agenten mehr gefunden wechselt er wieder in den FINDISLAND-Zustand und sucht erneut nach Inseln bzw. wartet auf Inselanfragen bis die Wartezeit ausläuft und er wieder in den FINDNEIGHBOURS-Zustand wechselt. Diesen Zyklus wiederholt der Sprecher bis das System terminiert. Die folgende Abbildung 4.1 zeigt den vorstehend beschriebenen Ablauf zur besseren Nachvollziehbarkeit. Eine detailliertere

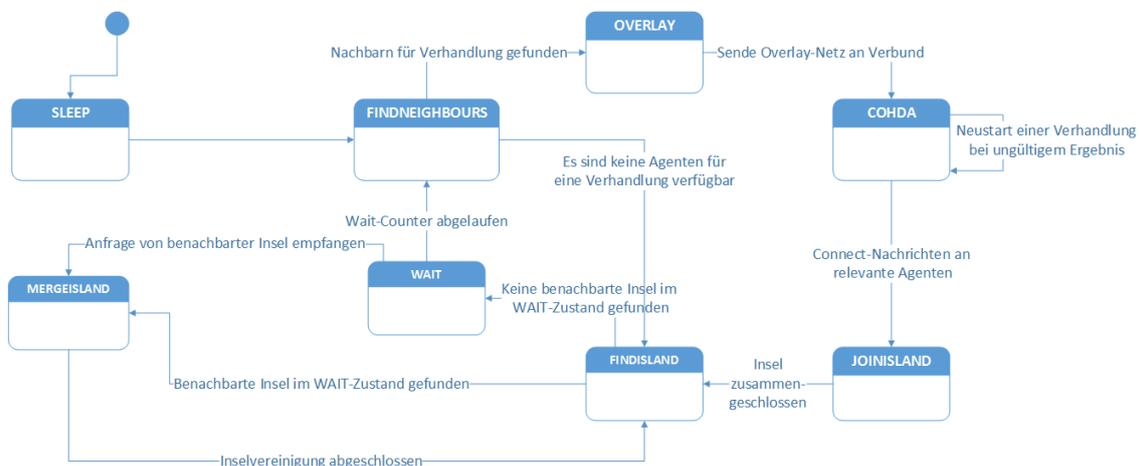


Abbildung 4.1: Zustandskonzept eines Sprechers

Beschreibung aller Zustände, der versendeten Nachrichten, sowie des Wait-Counters erfolgen im Abschnitt 4.3 und seinen Unterabschnitten.

### 4.2.2 Normale Agenten

Der grundsätzliche Ablauf der Zustände eines normalen Agenten unterscheidet sich insbesondere darin von dem eines Sprechers, dass normale Agenten ihre Zustände

nur reaktiv auf einen Nachrichtenempfang wechseln und durch die geringere Zahl an Aufgaben entsprechend weniger Zustände benötigt werden.

Ein Agent befindet sich nach dem Start des Systems zunächst in SLEEP, ehe er eine Anfrage eines Sprechers erhält und er so in den Zustand OVERLAY wechselt. Hier wartet der Agent so lange, bis der Sprecher das Overlay-Netz gebildet hat und sein initiales Arbeitsgedächtnis an die Nachbaragenten des Sprechers versendet hat. Mit dem Empfang eines ersten Arbeitsgedächtnisses wechselt der Agent in den Zustand COHDA, um an der Verhandlung teilzunehmen. Im Anschluss an eine erfolgreiche Verhandlung wechseln die Agenten, welche in den Verbund aufgenommen werden, in den Zustand JOINISLAND, während die anderen wieder in SLEEP wechseln, bis sie für eine neue Verhandlung angefragt werden. Im JOINISLAND-Zustand senden die Agenten eine Connect-Nachricht an alle Schalteragenten auf der Stromkante zwischen den Agenten, welche Teil der neuen Insel sein sollen. Sobald die Schalter geschlossen sind wechseln die Agenten in den Zustand CONNECTED, in welchem sie bleiben, bis eine externe Nachricht ein anderes Verhalten auslöst. Findet der Sprecher eine benachbarte Insel, versetzt er die Agenten in seiner Insel in den MERGEISLAND-Zustand. Analog passiert dies in der benachbarten Insel. Durch das schon beschriebene Aufgeben der Sprecherrolle eines Sprechers und das Austauschen des Arbeitsgedächtnisses zwischen den Inseln werden durch die Agenten alle Schalter auf Kanten zwischen den Inseln geschlossen. Der Inselzusammenschluss ist beendet und die Agenten gehen wieder in den Zustand CONNECTED. Findet der Sprecher keine benachbarte Insel im WAIT-Zustand, wartet dieser für die Dauer der Wartezeit im WAIT-Zustand. Die normalen Agenten bleiben in dieser Zeit im CONNECTED-Zustand. Bei Anfrage des Sprechers durch eine Nachbarinsel wird eine Inselvereinigung wie zuvor beschrieben ausgelöst. Wird keine benachbarte Insel gefunden, bleiben die Agenten in CONNECTED bis der Sprecher neue Verhandlungspartner findet oder das System terminiert. In der folgenden Darstellung 4.2 ist das Zustandsmodell eines normalen Agenten analog zum zuvor beschriebenen Ablauf dargestellt.

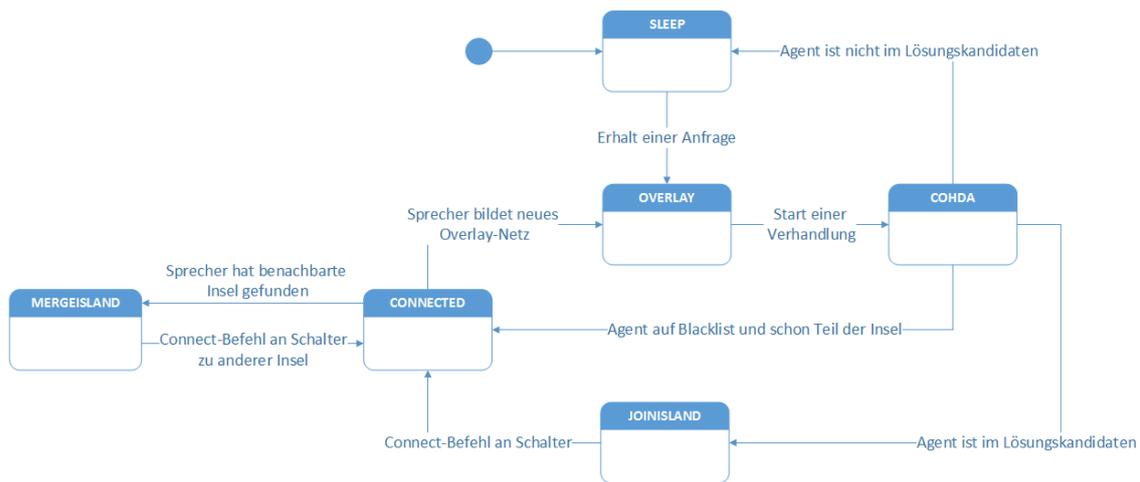


Abbildung 4.2: Zustandskonzept eines Agenten

## 4.3 Verfahren

Im Folgenden wird der genaue algorithmische Ablauf und das Verhalten der Agenten in den einzelnen Zuständen beschrieben. Dabei wird davon ausgegangen, dass der Systemstart bereits erfolgt ist und sich alle Agenten in SLEEP befinden. Das Aufwachen der Sprecher wird durch eine Nachricht vom System initiiert, die die zu verwendenden Gewichte für die Zielfunktion enthält. Der Sprecher speichert diese Gewichte und erstellt seine Insel, in der er zu diesem Zeitpunkt noch alleine ist. Anschließend wechselt er in den Zustand FINDNEIGHBOURS, wodurch der eigentliche Ablauf beginnt.

### 4.3.1 Verbundbildung

Der erste Schritt des Algorithmus besteht in der Bildung eines Verbundes von Energieanlagen. Ziel ist es diesen Verbund im weiteren Verlauf durch die Optimierung der Leistungen der einzelnen Anlagen zu einem Inselnetz zusammen schalten zu können. Das Erstellen des Verbundes läuft in drei Schritten ab.

Zu Beginn muss der Sprecher einer Insel ermitteln, welche Agenten für eine Verbundbildung in Frage kommen. Die Voraussetzung dafür, dass ein Agent berücksichtigt wird, ist, dass er mindestens eine Strom- und eine Kommunikationskante in die Insel des agierenden Sprechers hat. Ob diese an einer oder an verschiedene Anlagen in der Insel anknüpfen ist dabei nicht von Bedeutung.

Abbildung 4.3 zeigt eine beispielhafte Netzsituation. Die Insel bestehend aus den Agenten 1, 2, 3 und 4 hat vier Nachbarn. Agent 5 hat sowohl eine Strom- als auch eine Kommunikationskante in die Insel und ist damit für eine Verhandlung relevant. Agent 6 hat lediglich eine Kommunikationskante in die Insel und kann deshalb erst

berücksichtigt werden, wenn Agent 5 in die Insel integriert ist. Agent 7 hat, wenn auch über zwei verschiedene Agenten, eine Strom- und eine Kommunikationskante in die Insel und wird deshalb ebenfalls berücksichtigt. Agent 8 hingegen hat nur eine Stromkante in die Insel und ist damit zunächst nicht relevant.

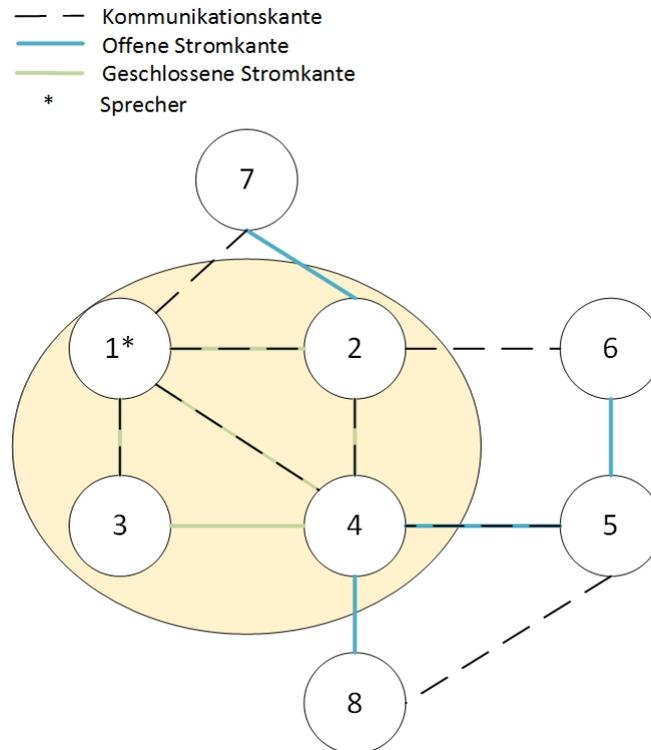


Abbildung 4.3: Nachbarschaftsmöglichkeiten einer Insel

Nachdem der Sprecher ermittelt hat, welche Agenten für eine Verbundbildung in Frage kommen, sendet er an diese eine ReadyForCohda Nachricht. Ein Agent, der eine solche Nachricht erhält und im Zustand SLEEP ist, antwortet mit einem Agree und wechselt in den Zustand OVERLAY. Mit diesem Wechsel des Zustandes ist sichergestellt, dass der Agent nur an einer Optimierung zur Zeit teilnimmt. Ein Agent, der sich bei Erhalt einer ReadyForCohda Nachricht nicht in SLEEP befindet, antwortet immer mit einer Disagree Nachricht, ohne eine weitere Aktion auszuführen.

Der Sprecher wartet, bis er von allen angefragten Agenten eine Antwort bekommen hat. Agenten die mit einem Agree antworten, merkt er sich. Disagree Nachrichten führen zu keiner weiteren Aktion.

Sobald der Sprecher alle Antworten erhalten hat, prüft er, ob mindestens ein Agent mit Agree geantwortet hat. Ist dies der Fall so wechselt er in den Zustand OVERLAY. Haben alle angefragten Agenten mit Disagree geantwortet, so wechselt der Sprecher in FINDISLAND, um eine Erweiterung seiner Insel durch eine Inselvereinigung zu erreichen.

Eine Erweiterung dieser Logik könnte sein, auch Agenten zu berücksichtigen, die nur über andere Agenten eine Strom- und eine Kommunikationskante in die Insel haben. In Abbildung 4.3 wären dies die Agenten 6 und 8. Dies würde ein schnelleres Wachstum und effizientere Verhandlungen bedeuten, da in einer Verhandlung mehr Agenten und damit mehr Flexibilität enthalten sind. Hierfür muss aber eine Logik zur Beurteilung der Netzsituation durch die einzelnen Agenten eingeführt werden. Ein Agent ohne direkte Netzkanten in die Insel dürfte sich nur dann in die Lösung integrieren, wenn auch sein Nachbar, der direkt mit der Insel verbunden ist, zugeschaltet wird.

### 4.3.2 Erstellen des Overlay-Netzes

Ein Sprecher, der Agenten für einen Verbund gefunden hat, erstellt im nächsten Schritt ein Overlay-Netz. Dieses virtuelle Netz bildet das bestehende Kommunikationsnetz ab. Es ordnet jedem Agenten die Nachbarn zu, an die er im Rahmen der Optimierung seine Arbeitsgedächtnisse schicken darf. Das Erstellen eines solchen Overlay-Netzes entspricht dem Ablauf von COHDA, wo dies ebenfalls vorgesehen ist [Hinr14, 4.1]. Die Ausgestaltung des Overlay-Netzes im Detail unterscheidet sich jedoch deutlich. Bei COHDA wird vorausgesetzt, dass zwischen den Agenten eine vollständige Kommunikation herrscht, auf die ein Overlay-Netz gelegt wird. Dieses wird insbesondere zur Verringerung der Kommunikationskomplexität verwendet [Hinr14, 3.6.2]. Dadurch, dass in dem hier betrachteten Szenario von eingeschränkter Kommunikation ausgegangen wird, wurde darauf verzichtet, dem Overlay-Netz eine spezifische Topologie zu geben. Stattdessen bildet es die zur Verfügung stehenden Kommunikationswege uneingeschränkt ab.

Es wird aber zwingend benötigt, da der einzelne Agent nicht wissen kann, welche seiner Nachbarn im erstellten Verbund enthalten sind. Dies weiß nur der Sprecher, der es über das Overlay-Netz den Agenten mitteilt.

Die Erstellung des Overlay-Netzes erfolgt mit Hilfe der Informationen der Insel des Sprechers, sowie den gespeicherten Agenten aus der Verbundbildung. Aus der Vereinigung der Agenten innerhalb der Insel, sowie der neu im Verbund enthaltenen Agenten, ergibt sich die Menge der an der Verhandlung teilnehmenden Agenten. Im Overlay-Netz legt nun der Sprecher für jeden dieser Agenten fest, an welche Nachbarn dieser während der Optimierung seine Arbeitsgedächtnis Nachrichten senden soll. Hierfür werden die Kommunikationsnachbarschaften der einzelnen Agenten mit den an der Verhandlung teilnehmenden Agenten verglichen. Tritt ein Agent in beiden Mengen auf, so wird er dem gerade betrachteten Agenten als Nachbar im Overlay-Netz zugeordnet. Umgekehrt wird gleichzeitig der betrachtete Agent dem Nachbarn zugewiesen. Hierdurch wird sichergestellt, dass auch Agenten außerhalb der Insel,

deren Nachbarschaftsinformationen nicht vorliegen, korrekt berücksichtigt werden. Kommunikationskanten zwischen im Verbund enthaltenen Agenten, die noch nicht Teil der Insel sind, können durch die fehlenden Nachbarschaftsinformationen nicht berücksichtigt werden. Ein beispielhaftes Overlay-Netz ist in Abbildung 4.4 für das Beispiel aus Abschnitt 4.3.1 dargestellt.

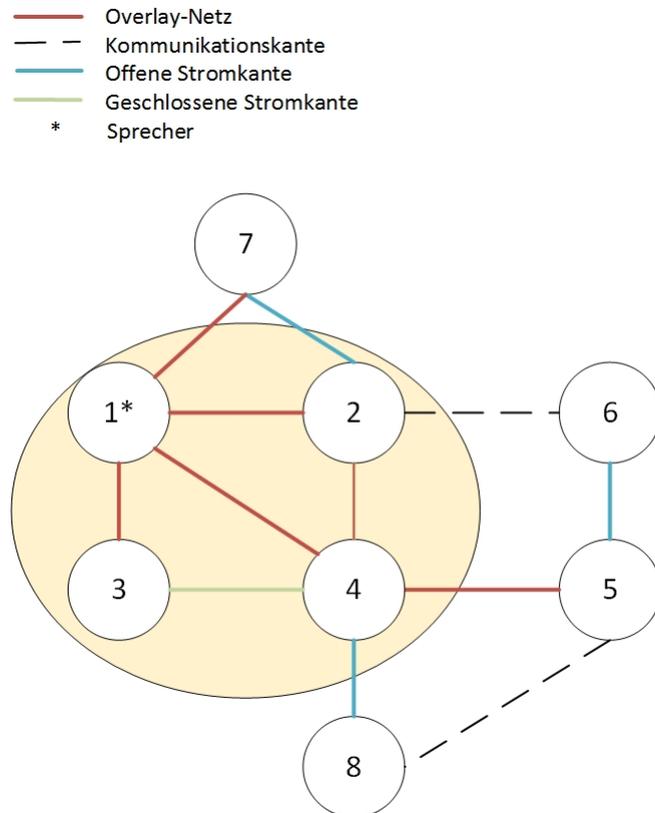


Abbildung 4.4: Overlay-Netz für das Beispiel aus Abb. 4.3

Wenn der Sprecher die Erstellung des Overlay-Netzes beendet hat, sendet er jedem an der Verhandlung teilnehmenden Agenten, inklusive sich selbst, die für ihn relevanten Nachbarschaftsinformationen. Die Nachricht des Sprechers an sich selbst stellt in der aktuellen Implementierung sicher, dass dieser nicht mit der Optimierung beginnt, bevor die anderen Agenten ihre Nachbarschaftsinformationen erhalten haben. In der verwendeten Simulation sind Agenten bisher nicht in der Lage Nachrichten zu speichern und zu einem späteren Zeitpunkt zu verarbeiten. Sollte also eine WorkingMemory-Nachricht aus der Inselbildung vor dem Overlay-Netz bei einem Agenten eintreffen, könnte sie nicht korrekt verarbeitet werden. Da die Agenten jedoch durch die Abfragen aus FINDNEIGHBOURS für die spezifische Insel geblockt sind, könnte die WorkingMemory-Nachricht in einer verbesserten Simulation auch bis zum Erhalt des Overlay-Netzes zwischengespeichert werden, um anschließend weiter verarbeitet zu werden.

Sobald ein Agent in OVERLAY ein Overlay-Netz erhält, speichert er sich die für ihn vorgesehenen Nachbarn für die kommende Verhandlung und wechselt in den Zustand COHDA.

Während der Erarbeitung des Algorithmus gab es Überlegungen auf die Bildung eines Overlay-Netzes inklusive der vorherigen Bereitschaftsabfrage zu verzichten. Dabei sollte die Zusammensetzung des Verbundes über die eingeführte Blacklist (Vgl. Abschnitt 4.3.3.1) umgesetzt werden. Agenten, die nicht bereit sind, an einer Verhandlung teilzunehmen, würden dies dem Sprecher nicht mitteilen, sondern an der Verhandlung unter der Bedingung teilnehmen, dass sie sich immer auf die Blacklist setzen. Der Vorteil hierbei wäre, dass die Schritte der Abfrage der Agenten und der Erstellung des Overlay-Netzes eingespart werden würde. Zudem wäre es denkbar, dass Agenten auch zur Laufzeit noch an einer Optimierung teilnehmen könnten, auch wenn sie zu Beginn nicht dafür bereit waren. Dafür müsste der Agent lediglich sein Paradigma aufgeben, sich konsequent auf die Blacklist zu setzen. Dieser Ansatz wurde jedoch aus zwei Gründen nicht weiter verfolgt. Zuerst hätte dies eine größere Abkehr des Verfahrens von COHDA als Basis-Algorithmus bedeutet, was eine größere Unsicherheit bezüglich der Funktionalität des Algorithmus zur Folge gehabt hätte. Zudem konnte zu dem Zeitpunkt der Entscheidung nicht abschließend geklärt werden, wie in diesem Fall sichergestellt werden kann, dass nur Agenten an einer Optimierung teilnehmen, die auch eine Strom- und eine Kommunikationskante in die spätere Insel haben. Dies bedeutet nicht, dass es ausgeschlossen ist, dass dies möglich ist. Für die Projektgruppe wurde aber die Verfolgung der jetzt umgesetzten Alternative als effizienter und erfolgversprechender beurteilt.

### 4.3.3 Verhandlung mit COHDA

Die Optimierung mit COHDA in einem Verbund von Agenten hat zum Ziel, dass diese sich zu einem leistungsausgeglichene Inselnetz zusammenschließen können. Da COHDA, wie bereits in Abschnitt 3.5 dargestellt wurde, einen solchen Verbund hinsichtlich eines vorgegebenen Wirkleistungsproduktes optimieren kann, ist die Heuristik für diese Problemstellung geeignet. Sie wurde an einigen Stellen an die spezifische Problemstellung des betrachteten Problems angepasst, was bereits auf mathematischer Ebene in Abschnitt 3.4.1 gezeigt wurde. Dies soll nun zunächst noch einmal erläutert und begründet werden, bevor auf das einzelne Verhalten der Agenten im Zustand COHDA eingegangen wird. Auf eine detaillierte Darstellung des Ablaufs von COHDA wird an dieser Stelle verzichtet, dies ist in [Hirr14] zu finden.

#### 4.3.3.1 Anpassungen an COHDA

Für die Anwendung von COHDA im betrachteten Szenario wurden verschiedene, kleinere Anpassungen an COHDA vorgenommen. Diese betreffen das vorgegebene

Wirkleistungsprodukt und die lokale Optimierung. Zudem wurde die Heuristik um eine Blacklist und eine Tabu-Liste ergänzt. Anschließend wird noch auf die Umsetzung der Konvergenzerkennung und die Verbreitung relevanter Informationen der Agenten eingegangen.

### **Ziel**

In COHDA kann dem Verbund, der die Optimierung durchführt, ein Wirkleistungsprodukt vorgegeben werden, welches er als Ziel für die Optimierung nutzt. Dieses wurde durch die Angabe der ID des Sprechers des Verbundes, den Gewichten für die Zielfunktion und einen Counter ersetzt. In der hier betrachteten Problemstellung, dass aus mehreren Anlagen eine leistungsausgeglichene Insel gebildet werden soll, ist das Ziel der Optimierung immer 0. Dies macht die Angabe dieses Ziels überflüssig. Dennoch gibt es einige verhandlungsspezifische Parameter, für die es die Möglichkeit geben soll, diese von Optimierung zu Optimierung zu ändern.

Einen solchen Faktor stellt insbesondere die Gewichtung der Zielfunktion dar. Durch die Mitgabe dieser Informationen im Arbeitsgedächtnis, bietet sich die Möglichkeit die Gewichte im Laufe des Netzwiederaufbaus zu verändern. Hier sind verschiedene Szenarien bis hin zur individuellen Anpassung der Gewichte an verschiedene Netz-situationen denkbar. Eine solche Anpassung ist zum jetzigen Stande aufgrund des Umfangs und der Komplexität nicht umgesetzt.

Weiterhin im Ziel enthalten ist die ID des Agenten, der das Ziel initial erstellt hat. Diese dient als Identifikator für eine Verhandlung. So könnten fehlgeleitete Arbeitsgedächtnisse die Optimierung nicht beeinträchtigen, da eindeutig identifiziert werden kann, ob das Arbeitsgedächtnis zum richtigen Verbund gehört.

Das Ziel wird durch einen Counter versioniert. Dieser gibt an, die wievielte Optimierung im identischen Verbund durch das Ziel spezifiziert wird. Eine Erhöhung des Counters wird bei bestimmten Optimierungsergebnissen durchgeführt und führt zu einem Neustart der Optimierung. Nach einer erfolgreichen Verhandlung, die zu einer Erweiterung der Insel geführt hat, wird der Counter zurückgesetzt. Die genaue Beschreibung der Verarbeitung der Optimierungsergebnisse ist in Abschnitt [4.3.4](#) zu finden.

### **Lokale Optimierung**

In der lokalen Optimierung wählt ein Agent seine für ihn optimale Leistung im Kontext seines wahrgenommenen Systemzustandes aus. In COHDA ist den Agenten dabei die freie Gesinnungswahl gestattet, sofern bei der Endauswahl des Lösungskandidaten die globale Zielfunktion ausschlaggebend ist. Sie können also ihre Leistungsauswahlen mit einer lokalen Zielfunktion bewerten und auswählen. Diese freie

Gesinnungswahl wurde im vorliegenden Szenario eingeschränkt. Das Verfahren soll dazu dienen im Falle eines Blackouts das Stromnetz wieder aufzubauen. Dies ist als Notfallsituation zu betrachten, in der die Präferenzen der einzelnen Anlagenbetreiber als nachrangig betrachtet werden können. Hier hat der möglichst schnelle Wiederaufbau des Stromnetzes Priorität. Deshalb ist es den Agenten vorgegeben, auch in ihrer lokalen Optimierung, bei der Auswahl ihrer Leistung, die globale Zielfunktion, die in Kapitel 3.4.1 vorgestellt wurde, zu benutzen.

### **Blacklist**

Ziel ist es, Agenten unter bestimmten Voraussetzung zu erlauben, sich zum globalen Nutzen aus einer Inselbildung herauszunehmen. Dies dient dazu, auch in Netzsituationen erfolgreiche Verhandlungen führen zu können, die bei Integration aller verfügbaren Agenten nicht ausgeglichen wären. Ein Agent, der sich auf der Blacklist befindet, wird folglich am Ende einer Optimierung nicht in die Insel aufgenommen. Es wurde bewusst eine Blacklist gewählt und nicht eine Leistung 0, da nicht ausgeschlossen werden kann, dass auch Anlagen die an das Stromnetz angeschlossen sind, zwischenzeitlich die Leistung 0 erbringen. Somit bestünde ohne die Blacklist hier keine Eindeutigkeit.

Ein Agent darf sich im vorliegenden Algorithmus unter drei Bedingungen auf die Blacklist setzen. Zunächst einmal muss der Counter des Ziels  $> 1$  sein, das heißt, es muss schon einmal eine Optimierung im gleichen Verbund gegeben haben, die neu gestartet wurde. Dies wurde eingeführt, da ansonsten die Situation aufgetreten ist, dass ein Agent sich auf die Blacklist setzt, obwohl er unter bestimmten Bedingungen in die Insel integriert werden könnte. Diese Problematik ist auf lokale Optima und die Nebenbedingung der Ausgeglichenheit der Insel zurückzuführen. In einem solchen Fall müsste in der Verhandlung zeitweilig eine bezüglich der Inselbilanz schlechtere Lösung akzeptiert werden, um den Agenten integrieren zu können. Dies ist aber durch die Formulierung der Inselbilanz als Nebenbedingung nicht zulässig, wodurch das lokale Optimum nicht verlassen werden kann.

Zur Verbesserung dieses Verhaltens gibt es verschiedene Ansätze. Zum einen könnte die Inselbilanz nicht wie bisher als Constraint, sondern als weiteres Gewicht in die Zielfunktion integriert werden. Hierdurch wären zeitweilig schlechtere Lösungen bezüglich der Bilanz durchsetzungsfähiger gegenüber Lösungen mit einer größeren Insel oder einer verbesserten Sichtbarkeit. Ein weiterer Ansatz ist, das Verhalten bezüglich der Auswahl der Blacklist noch weiteren Bedingungen zu unterwerfen. Eine Idee der Projektgruppe wäre ist das Einsetzen eines Simulated Annealing Ansatz bei der Auswahl der Blacklist. Eine solche Anpassung verspricht eine Verbesserung des Verhaltens des Algorithmus, konnte aber in den Umfang der Projektgruppe nicht

mehr integriert werden, weshalb hier die pragmatische Lösung, dass die Blacklist erst nach einer gescheiterten Optimierung zulässig ist, gewählt wurde.

Eine Alternative zur Blacklist ist der Ausschluss einzelner Agenten durch den Sprecher. Dabei würde nach der Konvergenz einer Verhandlung mit einem unausgeglichene Ergebnis der Sprecher anhand der Daten im Lösungskandidaten einen Agenten von der nächsten Verhandlung ausschließen und diese dann neu starten. Dieses Vorgehen hat allerdings gleich mehrere Nachteile. Zum einen wird hierdurch eine zentral entscheidende Instanz eingeführt. Dies widerspricht der Anforderung an den Algorithmus dezentral zu arbeiten. Zudem kann sich in der Zwischenzeit die Netzsituation so geändert haben, dass durch die Sichtbarkeit neuer Agenten der betroffene Agent doch in die Insel integriert werden könnte. Dieser Ansatz bietet also, verglichen mit der Blacklist, deutliche Defizite und wurde deshalb verworfen.

### **Tabu-Liste**

Wie bereits oben erwähnt, neigt das Verfahren dazu in lokalen Optima zu konvergieren. Um Situationen lösen zu können, in denen weder mit deaktivierter, noch das Aktivieren der Blacklist eine gültige Lösung gefunden werden konnte, wurde eine Tabu-Liste bezüglich der Startwerte eines Inselfprechers eingeführt. Der Startwert eines Sprechers ist diejenige Leistungsauswahl, die der Sprecher als erstes in einer Verhandlung auswählt. Dieser hat damit direkten Einfluss darauf, welche Werte alle anderen Agenten auswählen und damit von wo aus sich die Optimierung in Richtung ausgeglichener Inselbilanz bewegt. Verhandlungen, die mit einer ungültigen Lösung endeten werden auf verschiedene Arten behandelt. Dies wird ausführlich in Abschnitt 4.3.4 dargestellt. Endet eine Optimierung mit einem ungültigen Ergebnis, was bedeutet, dass die Bilanz der ermittelten Insel nicht ausgeglichen ist, wird der verwendete Start-Wert des Sprechers tabu gesetzt. In der nächsten Optimierung beginnt er die Optimierung dann mit dem maximal vom aktuellsten tabu-gesetzten entfernten Wert. Die Tabu-Liste wird nach einer erfolgreichen Optimierung, die zu einer Erweiterung der Insel führt, initialisiert.

### **Konvergenzerkennung**

Für die Konvergenzerkennung verteilter Algorithmen gibt es verschiedene Ansätze. Für COHDA wurden die Verfahren Computation Tree, Static Tree und Splitting-Vektor getestet und als anwendbar evaluiert [Hinr14, 4.3]. Da die Implementierung einer solchen Konvergenzerkennung ohne erkennbaren Erkenntnisgewinn einen erheblichen Aufwand dargestellt hätte, hat sich die Projektgruppe an dieser Stelle für einen pragmatischen Ansatz entschieden. Dieser sieht die Konvergenzerkennung durch das verwendete Multi-Agenten System vor. Dieses prüft, ob in einer Verhandlung noch WorkingMemory Nachrichten verschickt werden. Ist dies nicht der Fall,

wird der Sprecher der jeweiligen Optimierung mit einer Nachricht über die Konvergenz informiert. Dies ist für die Ziele der Projektgruppe zweckdienlich und kann bei Belieben durch eine auch im Feld einsatzfähige Konvergenzerkennung ersetzt werden, ohne, dass dies Einfluss auf die Funktion des Algorithmus hat.

### Informationsverteilung

Für die Verwendung der Zielfunktion benötigen die Agenten Informationen über die Nachbarschaften und die Priorisierung der in der Verhandlung beteiligten Agenten. Die Konfiguration, die in COHDA eigentlich nur die Agenten mit ihren Leistungsauswahlen enthält, wurde deshalb um diese Informationen erweitert.

#### 4.3.3.2 Ablauf im Algorithmus

Im Ablauf des Verfahrens betrachtet, wechseln Agenten aus dem Zustand OVERLAY in den Zustand COHDA, sobald sie ihr Overlay-Netz erhalten haben. Hier beginnt der Sprecher nun sein Arbeitsgedächtnis zu füllen. Dazu erstellt er zunächst das Ziel mit seiner ID, den Gewichten der Zielfunktion, die in der Start-Nachricht vom System enthalten waren und dem Counter. Anschließend wählt der Sprecher seinen Startwert aus. Rein nach der Zielfunktion betrachtet, wäre dies immer der niedrigste Leistungswert des Agenten. Dies wird aber, um von Beginn an möglichst viel Flexibilität in die Verhandlung zu bringen, durch die höchste, dem Agenten zur Verfügung stehenden, Leistung ersetzt. Sollte es sich nicht um die erste Verhandlung in der gegebenen Agentenkonstellation handeln, so wird wie unter Abschnitt 4.3.3.1 beschrieben, diejenige Leistung ausgewählt die maximal von dem aktuellsten Tabu-Eintrag entfernt ist.

Mit der ausgewählten Leistung füllt der Agent sein Arbeitsgedächtnis, erstellt also einen wahrgenommenen Systemzustand und einen Lösungskandidaten. Dieses Arbeitsgedächtnis schickt er dann an seine Nachbarn im Overlay-Netz. Anschließend gibt es keine Unterschiede mehr im Verhalten von Sprechern und normalen Agenten, zudem folgt der Ablauf strikt dem Vorgehen von COHDA.

Erhält ein Agent eine WorkingMemory Nachricht, so geht er nach den in COHDA vorgegebenen Schritten Wahrnehmung - Entscheidung - Aktion vor, die im Folgenden der Vollständigkeit halber verkürzt dargestellt werden. Die ausführliche Beschreibung des Vorgehens befindet sich in [Hinr14, 4.2]. In Abschnitt 3.5 und in dem im Anhang E befindlichen Seminarband wird das Verhalten ebenfalls dargestellt.

### Wahrnehmung

Im Schritt Wahrnehmung überprüft der Agent, ob das erhaltene Arbeitsgedächtnis für ihn neue Informationen erhält. Ist dies das erste Arbeitsgedächtnis, das der Agent

in der aktuellen Verhandlung empfängt, so sind alle erhaltenen Informationen neu. Diese werden dann ohne weitergehende Prüfung übernommen und der Agent fährt mit dem Punkt Entscheidung fort. Hat ein Agent bereits Daten in seinem Arbeitsgedächtnis, so muss das erhaltene Arbeitsgedächtnis äquivalent zu dem Vorgehen in COHDA geprüft werden.

Eine Ausnahme hierbei stellt das Ziel dar. Stellt der Agent fest, dass der Counter des Ziels erhöht worden ist, so initialisiert der Agent sein Arbeitsgedächtnis und verfährt so, als ob dies seine erste WorkingMemory-Nachricht der Verhandlung ist. Dieses Verhalten stellt den Neustart einer Optimierung dar.

Hat sich der Counter im Ziel nicht verändert, so wird der wahrgenommene Systemzustand auf neue Informationen geprüft. Enthält dieser bisher unbekannte Agenten, oder aktuellere Leistungsauswahlen von Agenten, so werden diese Informationen im Arbeitsgedächtnis des Agenten aktualisiert.

Anschließend werden die Lösungskandidaten miteinander verglichen. Unterscheiden sich die Agenten, die in den jeweiligen Lösungskandidaten enthalten sind, so wird ein neuer Lösungskandidat erstellt, der alle bekannten Agenten enthält. Sind in beiden Lösungskandidaten die gleichen Agenten enthalten, so wird derjenige Lösungskandidat mit der höheren Erfüllung der Zielfunktion übernommen. Besitzen beide Lösungskandidaten die gleichen Agenten und die identische Erfüllung, so wird derjenige Lösungskandidat übernommen, der von dem Agenten mit der höheren ID erstellt wurde.

Wurden im Schritt Wahrnehmung Informationen gewonnen, so wird mit dem Schritt Entscheidung fortgefahren. Gab es keine Änderungen am Arbeitsgedächtnis des Agenten, so ist die Verarbeitung der Nachricht beendet und der Agent fährt mit der Bearbeitung der nächsten Nachricht fort.

### **Entscheidung**

Im Schritt Entscheidung optimiert der Agent seine Leistung unter Beachtung der neu gewonnenen Informationen im wahrgenommenen Systemzustand. Dabei ist zu beachten, ob es dem Agenten erlaubt ist, die Blacklist zu nutzen oder nicht. Die Bedingungen hierfür wurden zu Beginn dieses Abschnitts dargestellt.

Aufgrund der geringen Komplexität der Leistungsmöglichkeiten eines Agenten in dem hier betrachteten Szenario, hat sich die Projektgruppe dafür entschieden den Agenten eine exakte Optimierung vorzugeben. Dies bedeutet, dass die Agenten jede ihnen mögliche Leistung ausprobieren und diejenige Leistung auswählen, die die Bilanz der Insel am nächsten an die Ausgeglichenheit bringt. Ist es dem Agenten erlaubt, die Blacklist zu nutzen, so prüft er im Anschluss noch, ob die Bilanz der

Insel ohne seine Beteiligung näher an 0 ist, als mit. Ist dies der Fall, so setzt der Agent sich auf die Blacklist.

Am Ende der Entscheidung überprüft der Agent, ob der wahrgenommene Systemzustand eine höhere Güte als der Lösungskandidat, den der Agent aktuell gespeichert hat, hat. Hierzu wird zuerst von beiden Konfigurationen die Bilanz berechnet. Hat eine der beiden Konfigurationen eine Bilanz näher an null, als die andere, so setzt sich diese durch. Ist die Bilanz bei beiden Konfigurationen gleich gut, so kommt die Zielfunktion zum Tragen. Hierzu ermittelt der Agent von beiden Konfigurationen die Inselgröße, die Größe der Kommunikationsnachbarschaft und die Summe der Prioritäten in den Inseln und setzt diese Werte in die Zielfunktion ein. Für die Gewichtung extrahiert er diese Gewichte aus dem Ziel seines Arbeitsgedächtnisses. Hier setzt sich der wahrgenommene Systemzustand durch, wenn dieser eine höhere Güte, als der Lösungskandidat hat. Ist die Güte gleich groß oder kleiner, übernimmt der Agent die ihm im Lösungskandidaten zugeordnete Leistung als seine neue Auswahl in den wahrgenommenen Systemzustand.

Das beschriebene Vorgehen in der lokalen Optimierung ist erneut eine pragmatische Herangehensweise. Wiederum war aber das gewählte Vorgehen für die Projektgruppe das sinnvollste, da eine Verbesserung der lokalen Optimierung der Zielerreichung dieser Projektgruppe nicht geholfen hätte, sondern dies, wie beispielsweise auch die Konvergenzerkennung, ein Mittel zum Zweck ist. Es erscheint sinnvoll, sobald das Szenario komplexer wird, über die Anwendungen von weniger aufwendigen Optimierungsverfahren nachzudenken. Die jetzt implementierte Optimierung bietet bei höher skalierten Netzen und einer größeren Leistungsauswahl des einzelnen Agenten einen Performanznachteil.

### **Aktion**

Wurde das Arbeitsgedächtnis im Schritt Wahrnehmung verändert, so schickt der Agent dieses im Anschluss an die Optimierung seiner Leistungsauswahl an seine Nachbarn im Overlay-Netz. Anschließend fährt er mit der Verarbeitung der nächsten Nachricht fort.

### **Konvergenzerkennung**

Die Konvergenz einer Optimierung wird, wie oben erläutert, durch das Multi-Agenten System erkannt. Sobald sich ein Lösungskandidat durchgesetzt hat und die Agenten identische Arbeitsgedächtnisse haben, werden keine neuen WorkingMemory-Nachrichten versendet. Dies wird von dem System erkannt und eine Konvergenznachricht an den Sprecher der Optimierung gesendet. Der genaue Ablauf wird in Abschnitt [5.2.2.4](#) beschrieben. Erhält der Sprecher diese Nachricht, wechselt er in

JOINISLAND um das Ergebnis der Verhandlung zu bewerten. Normale Agenten wechseln durch eine Connect-Nachricht in JOINISLAND und durch eine Abort Nachricht in SLEEP, sofern sie keiner Insel angehören, oder in CONNECTED, wenn sie bereits Teil einer Insel sind. Wann welcher Fall eintritt wird im folgenden Kapitel beschrieben.

#### 4.3.4 Verarbeitung des Optimierungsergebnisses

Sobald ein Sprecher die Nachricht des Konvergenzagenten erhalten hat, dass die aktuelle Verhandlung beendet ist, wechselt er in JOINISLAND und prüft das Ergebnis der Verhandlung. In Abbildung 4.5 ist eine Übersicht des Entscheidungsprozesses dargestellt, der dabei durchlaufen wird.

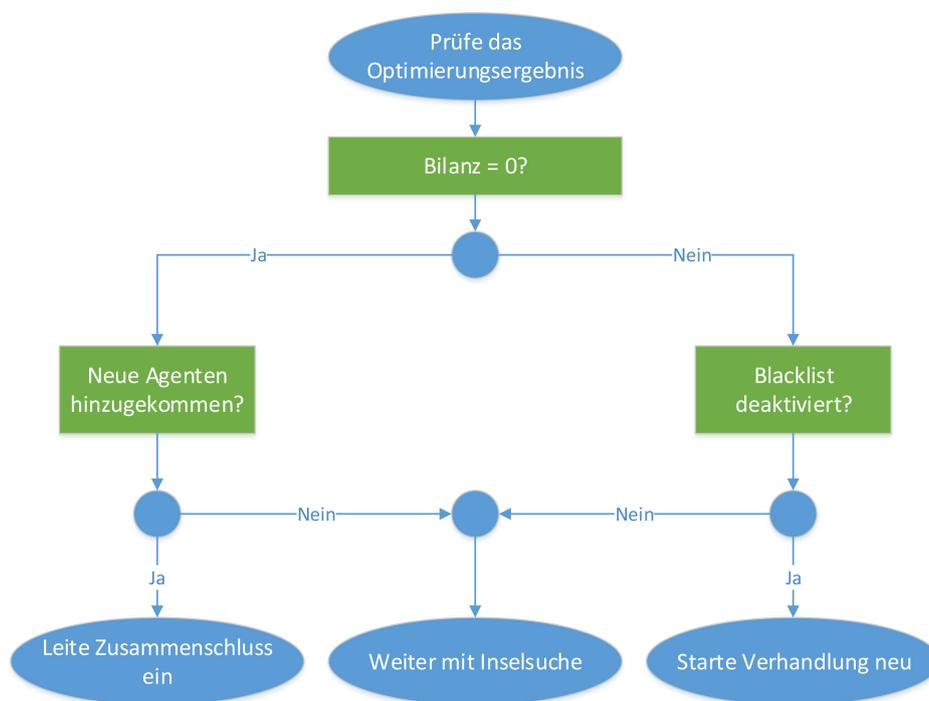


Abbildung 4.5: Verarbeitung des Optimierungsergebnisses

Im Idealfall stellt der Sprecher fest, dass die gefundene Lösung sowohl eine Bilanz von 0 hat, als auch eine Inselvergrößerung ermöglicht. Für eine Inselvergrößerung muss mindestens ein neuer Agent, der nicht auf der Blacklist steht, in der Lösung enthalten sein. Trifft beides zu, leitet der Sprecher die Maßnahmen für die Aufnahme eines oder mehrerer neuer Agenten in seine Insel ein. Die genaue Durchführung der Inselvergrößerung wird in Abschnitt 4.3.5 beschrieben.

Da eine Insel immer eine Bilanz von 0 haben muss, gilt eine Lösung mit einer Bilanz  $\neq 0$  als ungültig. Für die Berechnung der Bilanz werden nur Agenten berücksichtigt, die nicht auf der Blacklist stehen. Tritt dieser Fall ein, wird eine „Death Penalty“

aufgerufen und der aktuelle Verhandlungswert des Sprechers in die Tabuliste geschrieben. Somit wird sichergestellt, dass ein Sprecher nach einer fehlgeschlagenen Verhandlung andere Startparameter wählt, was dem Festhängen in lokalen Optima entgegenwirken soll. Falls die Blacklist in dieser Verhandlungsrunde noch nicht aktiviert war, startet der Sprecher eine neue Verhandlung mit aktivierter Blacklist, an der die gleichen Agenten wie zuvor teilnehmen. Dazu erhöht er den Counter des Ziels, erstellt ein neues Arbeitsgedächtnis, versendet es an seine Verhandlungsnachbarn und wechselt zurück in COHDA. Die Agenten, die an der vorherigen Verhandlung beteiligt waren, befinden sich nach wie vor in COHDA und können daher nahtlos mit der neuen Verhandlung fortfahren. Durch den veränderten Counter erkennen sie, dass die Blacklist verwendet werden darf. Ein ungültiges Verhandlungsergebnis bei aktivierter Blacklist stellt einen Sonderfall dar, der nur eintreten kann, wenn ein Sprecher seine erste Verhandlung durchführt und sich daher alleine in einer Insel befindet. Da sich ein Sprecher nicht selbst auf die Blacklist setzen und in der Regel nicht die Leistung 0 auswählen darf, kommt es zu einer ungültigen Lösung, wenn seine Nachbaragenten die von ihm ausgewählte Leistung nicht ausgleichen können. Der Sprecher schickt in diesem Fall eine Abort-Nachricht an alle Agenten der Verhandlung, wechselt in FINDISLAND und fährt dort mit der in Abschnitt 4.3.6 beschriebenen Vorbereitung einer Inselvereinigung fort. Die anderen Agenten wechseln bei Erhalt der Abbruchnachricht wieder in SLEEP. Sobald sich eine Insel aus mindestens zwei Agenten gebildet hat, gibt es immer eine gültige „Minimallösung“, die gefunden wird, wenn keine neuen Agenten in die Insel aufgenommen werden können. In diesem Fall entspricht das Verhandlungsergebnis der bereits existierenden Insel und alle Agenten außerhalb stehen auf der Blacklist. Eine Erweiterung der Insel könnte also nur durch eine Inselvereinigung oder eine Verhandlung mit anderen Nachbarn erreicht werden. Potentielle Verhandlungsteilnehmer ergeben sich beispielsweise durch neu aktivierte Kommunikation oder wenn ein Inselnachbar, der zuvor mit einer anderen Insel verhandelt hat, dort nicht aufgenommen werden konnte und daher wieder zur Verfügung steht. Die Reaktion des Sprechers auf eine fehlgeschlagene Inselvergrößerung ist identisch mit der Reaktion auf ein ungültiges Verhandlungsergebnis bei eingeschalteter Blacklist. Er sendet eine Abbruchnachricht an alle Verhandlungsteilnehmer und wechselt in FINDISLAND. Die Agenten, die bereits zu seiner Insel gehörten, wechseln bei Erhalt der Abbruchnachricht in CONNECTED, die anderen in SLEEP.

### 4.3.5 Aufnahme von Agenten in eine Insel

Wenn in einer Verhandlung eine gültige Lösung für eine potentielle Inselerweiterung gefunden wurde, müssen die neuen Agenten in die Insel aufgenommen werden. Dazu gehört, dass das Inselobjekt aktualisiert und die entsprechenden Schalter ge-

geschlossen werden. Außerdem müssen die Agenten, die für die Kommunikation in den Niederspannungssträngen zuständig sind, diese aktivieren, sobald sie erstmalig in eine Insel aufgenommen werden. Der Sprecher sendet an alle Agenten, die nicht auf der Blacklist stehen, eine Connect-Nachricht. Die auf der Blacklist stehenden Agenten erhalten eine Abort-Nachricht. Bei Erhalt einer Connect-Nachricht wechselt ein Agent ebenfalls in JOINSISLAND, schickt dem Sprecher Informationen über seine Nachbarschaften und aktiviert seine Kommunikation, falls vorhanden. Die Aktivierung der Kommunikation erfolgt dadurch, dass der Agenten an alle betroffenen Agenten eine Nachricht mit Informationen über die neuen Kommunikationsmöglichkeiten sendet. Diese passen daraufhin ihre Kommunikationsnachbarschaft an. Sofern sie sich in einer Insel befinden, senden sie außerdem eine Nachricht an den Sprecher, um ihn über die veränderte Nachbarschaft der Insel zu informieren. Die Aktualisierung der Kommunikationsnachbarn kann in jedem Zustand durchgeführt werden.

Der Sprecher aktualisiert nach Erhalt aller Antwortnachrichten das Inselobjekt und verschickt es an alle Agenten der Insel. Zum Schluss schicken die Agenten Nachrichten an die entsprechenden Schalter, die geöffnet werden und dadurch den Inselzusammenschluss vervollständigen.

#### 4.3.6 Vorbereitung einer Inselvereinigung

Eine erfolgreiche Inselvereinigung findet immer zwischen einer „aktiven“, anfragenden Insel und einer „passiven“, abwartenden Insel statt. Der Sprecher der aktiven Insel befindet sich in FINDISLAND. Er fragt der Reihe nach alle Inselnachbarn, zu denen mindestens eine Strom- und eine Kommunikationskante besteht, ob sie bereit sind für eine Inselvereinigung. Erst nach Erhalt einer Antwort wird die nächste Anfrage abgeschickt. Die Inselnachbarn können entweder Sprecher, normale Agenten in einer Insel oder normale Agenten ohne Insel sein. Sprecher und alleinstehende Agenten können direkt auf die Anfrage antworten, während Agenten in einer Insel, sofern sie sich in CONNECTED befinden, die Anfrage an ihren Inselsprecher weiterleiten, da sie in diesem Zustand nicht wissen, in welchem Zustand sich der Sprecher befindet. Eine positive Antwort wird nur dann gesendet, wenn sich der Sprecher der angefragten Insel in WAIT befindet und damit derzeit passiv ist. Sobald dieser Sprecher eine Anfrage zur Inselvereinigung erhält, wechselt er in MERGEISLAND, sodass nachfolgende Anfragen automatisch abgelehnt werden. Dadurch wird sichergestellt, dass eine Inselvereinigung nur zwischen zwei Inseln zur gleichen Zeit stattfindet und es nicht zu Überschneidungen kommen kann. Nachteil dieser Vorgehensweise ist die Verzögerung die sich dadurch ergibt, dass es einige Zyklen dauern kann, bis sich zwei Inseln in den richtigen Zuständen treffen. Auch deutlich effizientere Verfahren, wie die mögliche Vereinigung von drei oder mehr Inseln gleichzeitig sind dadurch nicht

möglich. Diese Nachteile wurden jedoch zugunsten der Stabilität bedingt durch bessere Übersichtlichkeit in Kauf genommen.

Hat ein Sprecher alle seine Inselnachbarn gefragt, ohne eine positive Antwort zu erhalten, wechselt er selbst in WAIT. Hier verbleibt er entweder solange, bis er eine Anfrage einer benachbarten Insel erhält, oder bis eine festgelegte Zeitspanne verstrichen ist. In dem von der PG erstellten System wurde diese Zeitspanne durch eine bestimmte Anzahl an Steps repräsentiert, die abhängig von der ID des Agenten festgelegt wurde. Als mögliche Werte wurden Primzahlen gewählt, um zu verhindern, dass zwei Inseln in eine Endlosschleife geraten, in der sie sich immer wieder verpassen. Nach Ablauf dieser Steps würde der Sprecher anschließend wieder in FIND-NEIGHBOURS wechseln. Wenn sich eine passive und eine aktive Insel gefunden haben, wird die Inselvereinigung vollzogen.

### 4.3.7 Inselvereinigung

Für die Vereinigung zweier Inseln muss zum Einen entschieden werden, welcher Sprecher seine Rolle aufgibt und zum Anderen welche Schalter geschaltet werden müssen. Die Wahl des Sprechers wurde im System so umgesetzt, dass der Sprecher der passiven Insel direkt bei Erhalt einer Anfrage, nachdem er eine positive Antwort mit Informationen über seine Insel zurückgesendet hat, seine Sprecherrolle niederlegt. Andere mögliche Auswahlkriterien wären zum Beispiel die Inselgröße oder die ID der Sprecheragenten. Der Ex-Sprecher verbleibt danach in MERGEISLAND, bis die Inselvereinigung beendet wurde. Der aktive Sprecher hat die Aufgabe das Schalten der entsprechenden Schalter zu initiieren. Hierfür nutzt er die Informationen, die er von dem Sprecher der passiven Insel erhalten hat, um zu ermitteln, welche Agenten sich an den Verbindungskanten befinden. Er informiert die Agenten über die Inselvereinigung, die daraufhin ebenfalls in MERGEISLAND wechseln und das Schließen der Schalter veranlassen. Die ausgewählten Leistungen aller Agenten werden im Zuge der Inselvereinigung nicht verändert, da beide Inseln zuvor in sich ausgeglichen waren. Die Inselvereinigung ist damit abgeschlossen; die normalen Agenten, zu denen nun auch der ehemalige Sprecher der passiven Insel gehört, wechseln wieder in CONNECTED. Der Sprecher wechselt zurück in FINDISLAND und sucht innerhalb der aktualisierten Inselnachbarschaft nach weiteren Inseln. Dieses Vorgehen wurde im Hinblick darauf gewählt, dass es sinnvoll erschien, benachbarte Inseln möglichst schnell zu vereinigen. Dadurch entstehen schneller größere und damit stabilere Inseln, anstelle vieler kleinerer. Im Hinblick auf die Ergebnisse der Validierungsphase in Abschnitt 7.2, in denen deutlich wird, dass größere Inseln ein höheres Nachrichtenaufkommen in den Verhandlungen verursachen, wäre es eventuell sinnvoll, hier eine andere Vorgehensweise zu wählen. So könnten die Sprecher beispielsweise erst dann mit der Suche nach Nachbarinseln beginnen, wenn sie zuvor festgestellt haben,

dass es nicht mehr möglich ist, die eigene Insel durch Verhandlungen zu vergrößern.

### 4.3.8 Terminierung des Verfahrens

Es gibt zwei Situationen, die zur Terminierung des Systems führen sollten.

1. Alle Agenten befinden sich in einer Insel
2. Das System stagniert

Die Terminierungserkennung erfolgt, ebenso wie die Konvergenzerkennung innerhalb einer Verhandlung, zentral durch das MiniMAS, siehe Abschnitt 5.2.1, beziehungsweise durch den Konvergenzagenten, der in Abschnitt 5.2.2.4 genauer beschrieben wird. Dieser Konvergenzagent behält den Überblick über alle aktiven Inseln und bekommt von den jeweiligen Sprechern eine Nachricht, wenn eine Insel „fertig“ ist. Eine Insel gilt dann als fertig, wenn sie entweder keine Nachbarn mehr hat, zu denen sowohl eine Strom-, als auch eine Kommunikationskante existiert, oder wenn sie stagniert. In jedem Step prüft das MiniMAS mit Hilfe des Konvergenzagenten, ob es noch aktive Inseln gibt, die nicht fertig sind. Ist dies der Fall, wird der nächste Step ausgeführt und weiterhin Nachrichten zugestellt. Andernfalls wird das System terminiert.

#### **Eine Insel hat keine ansprechbaren Nachbarn mehr**

Im Idealfall tritt diese Situation nur dann auf, wenn das Netz vollständig wieder aufgebaut wurde und es keine Agenten mehr gibt, die sich nicht in einer Insel befinden. Wenn ein Sprecher nach einer Verhandlung oder Inselvereinigung feststellt, dass er keine Nachbarn mehr ansprechen kann, meldet er dies dem Konvergenzagenten, wechselt direkt in WAIT und verbleibt dort ohne zeitliche Begrenzung. War er der einzige noch aktive Sprecher im System, würde an dieser Stelle das System terminieren. Es besteht allerdings auch die Möglichkeit, dass eine Insel keine ansprechbaren Nachbarn mehr hat, weil bestimmte Kommunikationswege noch nicht aktiviert sind oder weil es sich um ein nicht lösbares Netz handelt. In diesem Fall können sich an anderen Stellen im Netz noch aktive Inseln befinden, die normal weiterarbeiten und gegebenenfalls die passive Insel im Laufe des Netzaufbaus aufnehmen können. Eine sinnvolle Erweiterung wäre an dieser Stelle, dass eine in WAIT wartende Insel wieder aktiv werden kann, sobald sich neue Kommunikationsmöglichkeiten ergeben.

### **Eine Insel kann keine neuen Nachbarn aufnehmen**

Um zu erkennen, ob sich eine Insel in einer „Sackgasse“ befindet und keine Neuaufnahme von umliegenden Agenten mehr möglich ist, werden die in Abschnitt 4.3.4 beschriebenen Methoden zur Prüfung des Optimierungsergebnisses verwendet. Von Interesse sind dabei die ungültigen Verhandlungsergebnisse, insbesondere das Eintreten einer Death Penalty trotz eingeschalteter Blacklist und eine Lösung, bei der sich alle neuen Agenten auf der Blacklist befinden. In beiden Situationen wird an alle an der Verhandlung beteiligten Agenten eine Abbruch-Nachricht gesendet. Der Sprecher merkt sich in diesem Fall die IDs der Agenten und prüft beim erneuten Auftreten einer solchen Situation, ob er wieder den gleichen Agenten eine Abbruch-Nachricht sendet. Wenn die Listen identisch sind, erkennt der Sprecher dies als Stagnation und meldet diese dem Konvergenzagenten durch Senden einer Finished-Nachricht. Anschließend führt er weiter den normalen Zyklus aus Agenten- und Inselsuche durch. Sobald sich Änderungen ergeben, zum Beispiel durch neue Kommunikationswege, sodass eine Verhandlung mit anderer Zusammensetzung stattfinden kann, kann der Sprecher diese direkt nutzen. Bei einer erfolgreich durchgeführten Verhandlung würde er sich anschließend wieder als „nicht fertig“ melden.

Befinden sich zwei Inseln nebeneinander, ist immer eine Inselvereinigung möglich. Solange eine Insel also ansprechbare Nachbarn hat, werden so lange die Zustände durchlaufen, bis sich die Inseln gefunden und vereinigt haben. Hier kann es jedoch zu einer vorzeitigen Terminierung kommen, beispielsweise durch folgende Situation: Insel 1 und Insel 2, als einzige Inseln im Netz, liegen nebeneinander, haben aber beide auch noch einzelne Agenten in der Nachbarschaft. Sie führen beide jeweils eine ungültige Verhandlung durch, suchen nach Inseln, verpassen sich jedoch und führen erneut mit den gleichen Agenten eine Verhandlung durch. Da diese wieder ungültig sein wird, melden sie sich beide als fertig beim Konvergenzagenten, woraufhin das System terminiert, obwohl theoretisch noch eine Inselvereinigung möglich wäre. Dem könnte man entgegenwirken, indem der Sprecher sich Informationen darüber einholt, ob sich in seiner Nachbarschaft eine Insel befindet, eher er sich als fertig meldet.

### **Nicht lösbare Netze**

Ein Netz kann aus unterschiedlichen Gründen nicht lösbar sein. Dies können einerseits Probleme in der Netztopologie sein, wie fehlende Strom- oder Kommunikationskanten, die dazu führen, dass Agenten abgeschnitten vom restlichen Netz sind. Andererseits wäre es möglich, dass nicht alle Agenten mit einer Bilanz von 0 zusammengeschaltet werden können, da beispielsweise durch den Ausfall eines Erzeugers Leistung fehlt. Beide Fälle werden durch die zuvor vorgestellten Mechanismen zur Terminierungserkennung abgedeckt. Bei topologischen Problemen würde der Fall

eintreten, dass alle aktiven Inseln keine ansprechbaren Nachbarn mehr haben und sich somit als fertig melden. Die Situation der nicht ausgleichbaren Agenten würde wiederum dadurch abgedeckt werden, dass zweimal hintereinander an die gleichen Agenten eine Abbruchnachricht geschickt werden muss, ohne dass dabei ein neuer Agent aufgenommen werden kann.

# 5. Implementierung

Die Implementierung ist die Umsetzung des in Abschnitt 4.3 beschriebenen Verfahrens zum Lösen der in 3.2 definierten Szenarien. Sie ist außerdem wesentlich zum Erreichen des in Kapitel 3.3.1 festgelegten Ziels Z-04.

In diesem Kapitel wird die zugrundeliegende Systemarchitektur beschrieben. Dabei wird besonders auf die Implementierung des Prototyps eingegangen und es werden verwendete Designentscheidungen und deren Alternativen aufgezeigt. Dazu wird zuerst die Architektur und das Design des Systems erläutert. Dabei stehen ebenso die Auswahl eines geeigneten Multiagentenframeworks sowie die Modularisierung des Systems im Vordergrund. Darauf aufbauend wird die Implementierung des Prototyps näher erläutert. Es erfolgt keine Beschreibung der verwendeten Algorithmen wie in Kapitel 4, sondern eine begründete Erklärung der Umsetzungsstruktur. Zusätzlich werden die Grenzen der Designentscheidungen und damit dieser Implementierung aufgezeigt.

## 5.1 Systemarchitektur

Im Folgenden wird die Systemarchitektur des Systems beschrieben. Dazu wird zuerst auf das generelle Design und die Auswahl der verwendeten Technologien eingegangen. Ein besonderer Fokus liegt dabei auf der Auswahl des Multiagentensystems. Im zweiten Teil dieses Abschnitts folgt die Beschreibung der Implementierung des in Kapitel 4 beschriebenen Verfahrens. Abschließend werden die Grenzen des Systems, welche eben durch die verwendeten Technologien gesetzt sind, kurz erläutert.

### 5.1.1 Grunddesign und verwendete Frameworks

Folgend werden die grundlegenden Designentscheidungen erläutert. Dabei werden sowohl die Objektabhängigkeiten als auch die Prinzipien des objektorientierten Designs und die verwendeten Frameworks dargestellt.

#### 5.1.1.1 Entwicklungsprinzipien

Zu den grundlegenden Designentscheidungen gehören besonders die folgenden Prinzipien der objektorientierten Entwicklung. Das „Single Responsibility“-Prinzip beschäftigt sich mit der Aufteilung logischer Komponenten. Es sind Komponenten zu

programmieren, welche genau eine Verantwortung übernehmen. Dies sorgt, unter anderem, für weniger Code mit unklaren Aufgaben und eine bessere Chance auf Wiederverwendbarkeit. Um das zu erreichen gilt es, folgende Regel einzuhalten: Kohäsion maximieren und Kopplung minimieren. Das bedeutet, es sollen Komponenten entwickelt werden, welche untereinander über spezifische Schnittstellen kommunizieren. Gleichzeitig sollen diese Komponenten aus Modulen bestehen, welche untereinander möglichst viele Abhängigkeiten besitzen, siehe Abschnitt 5.1.3

Das zweite Prinzip lautet „Don't repeat yourself“. Ziel sollte es sein, möglichst granulare Methoden zu entwickeln, welche eine hohe Wiederverwendbarkeit besitzen. Dadurch wird der Programmcode kürzer, übersichtlicher und weniger fehleranfällig. Das dritte und letzte wesentliche Prinzip dieser Projektgruppe ist das „Open-Closed“-Prinzip. Da mit jedem Prototyp neue Funktionen hinzugefügt werden, ist es wichtig spezielle Erweiterungspunkte der einzelnen Komponenten zu gestalten. Dadurch wird der Implementierungsvorgang neuer Funktionen deutlich verkürzt.

#### 5.1.1.2 Objekthierarchie

Insgesamt gibt es fünf verschiedene Typen von Objekten im Pacman-System. Agenten, Actions, Behaviours, Manager und das Multiagentensystem mit einigen Unterstützungsklassen. Die für das Verfahren aus 4.3 entscheidenden Funktionsträger sind die Agenten, Behaviour und Actions. Diese drei besitzen untereinander eine speziell für diesen Prototypen entwickelte Abhängigkeitsbeziehung. Jeder Agent befindet sich zu jedem Zeitpunkt in einem Zustand. Falls ein Ereignis eintritt, entscheidet jeder Agent, anhand dieses Ereignis und seinem aktuellen Zustand, welches Verhalten ausgeführt wird. Dieses ausgewählte „Behaviour“ ist wiederum Träger möglicher Aktionen, die sich in Action-Objekten befinden. Der genauere Aufbau der Behaviours wird in Abschnitt 5.1.3.4 erläutert. Da Klassen der jeweiligen Komponenten untereinander ähnliche Informationen beherbergen, erben sie diese von sogenannten Default-Klassen. Beispielfhaft wird in Abbildung 5.1 die Objekthierarchie der Agenten dargestellt.

Die Hierarchie der Agenten-Klassen, wie in Abbildung 5.1 dargestellt, basiert auf drei Agenten, welche von einem „Default-Agenten“ abstammen. Jeder Agent erbt dabei Attribute wie ID und Methoden wie das Nachrichtenversenden. Dazu hat jeder Agententyp noch spezifische Eigenschaften und Methoden. Beispielsweise kann nur der Schalteragent schalten. Insgesamt hat aber jeder Agent einen Zugriff auf seine individuellen Actions und Behaviours. Diese wiederum haben das gleiche Hierarchiemodell. Diese werden in 5.1.3.4 und 5.1.3.3 genauer beschrieben, genau wie eine konkrete Erklärung der Agenten in 5.2.2 erfolgt. Da die Nachrichtentypen und Zustände unter den Agenten immer gleich sind, werden diese mittels Interfaces aus-

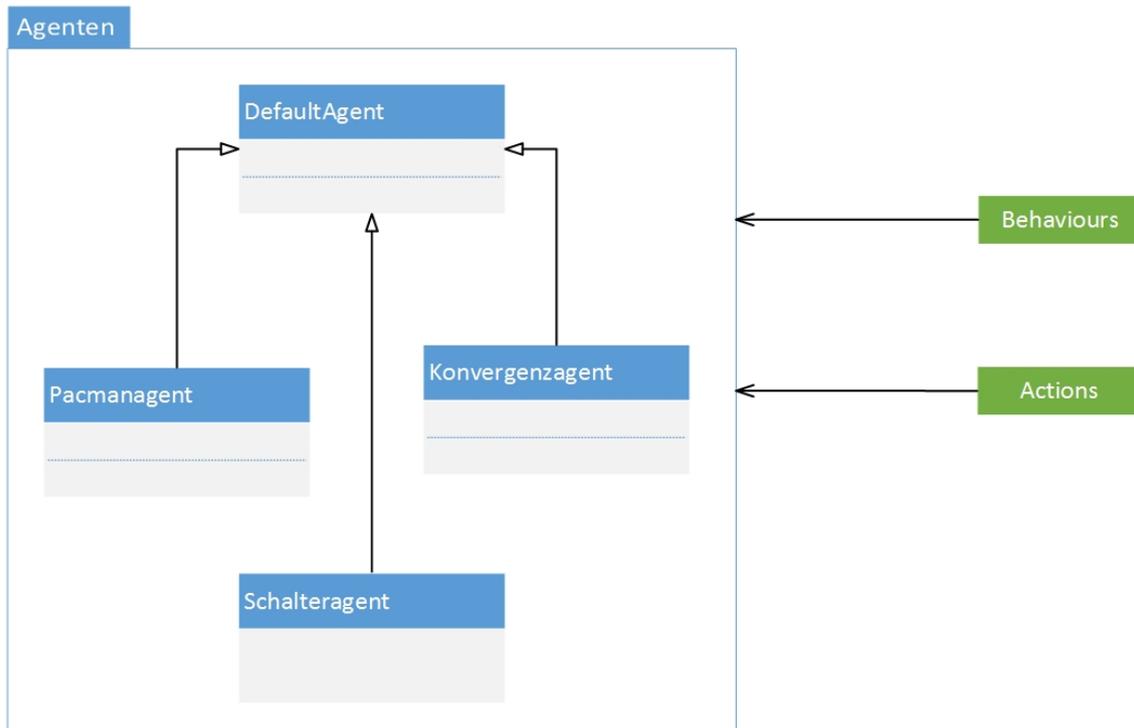


Abbildung 5.1: Klassendiagramm der in der Implementierung verwendeten Hierarchie der Agenten-Klasse

gelagert. Der Vorteil ist, dass beispielsweise der Nachrichtentyp „Connect“ im Code genauso verwendet werden kann, als wäre dieser in der Klasse selbst implementiert.

### 5.1.2 Multiagentenframework

Wie in Abschnitt 1.1 erwähnt, wurde zur Lösung der in Kapitel 3 erläuterten Problemstellung ein Multiagentensystem verwendet. Ein Multiagentenframework besteht aus einem Multiagentensystem, einer Multiagentenplattform, einer Bibliothek von Agenten und gegebenenfalls weiteren Hilfsinstrumenten, wie eine grafischen Oberfläche. Das Multiagentensystem ist dabei die Instanz der Laufzeitumgebung der Agenten. Es ist die Arbeitsumgebung der Agenten und deren Koordination zugleich. Agenten arbeiten innerhalb eines Multiagentensystems nach einem Kooperationsmodell, welches die Kooperation, Konkurrenz oder Unabhängigkeit der Agenten festlegt. Die Agenten, die im Multiagentensystem laufen, werden vorher innerhalb einer der Framework zugehörigen Bibliothek definiert. Dabei bedeutet dies meist eine eigene Programmierung der einzelnen Agenten. Die Multiagentenplattform ist für die Initialisierung, Verwaltung und Steuerung der Agenten des Multiagentensystems zuständig. Sie ist demnach auch ein Bestandteil des Multiagentensystems, genauer es ist die Zentrale. Die Plattform hat außer dem Ansteuern der Agenten noch die Kommunikation zwischen diesen als Aufgabe. Die Projektgruppe hat sich, zu Beginn der Projektphase, auf Java als Programmiersprache festgelegt. Daher wurden nur Frameworks betrachtet, welche in Java implementiert sind. Im Folgendem

werden anhand von JADE und dem MiniMAS aus JOHDA zwei Multiagentenframeworks genauer untersucht.

JADE ist ein freies Multiagentenframework, welches komplett in Java implementiert ist. Es wird als Open Source Software unter LGPL Lizenz vertrieben und beinhaltet Bibliotheken von Klassen zur Erzeugung von Agenten. Es unterstützt die Entwicklung von Agenten und bietet gleichzeitig eine Laufzeitumgebung für diese. Dabei ist es möglich, Agenten dank verschiedener Kommunikationsprotokolle und Standards wie TCP oder FIPA auf verteilten Systemen zu verwenden. Dies würde dem in Abschnitt 3 definierten Problemen und Szenarien entsprechen. Dazu würden Container definiert, welche jeweils Instanzen der JADE Laufzeitumgebung sind. Die Container laufen in separaten Java-Prozessen und beinhalten 0 bis N Agenten. Das ermöglicht auch eine Simulation von dezentralen Systemen auf nur einem Gerät, da Beispielsweise Objekte außerhalb der Container mittels RMI (Remote Method Invocation) Übertragen werden können. Dies sorgt dafür, dass Agenten, welche logisch keine gemeinsamen Ressourcen haben dürften, beim Übertragen von Objekten keine Referenz auf eine gemeinsame Ressource erhalten. Ein Problem hierbei wäre aber das Auftreten von Race Conditions. Eine Race Condition ist ein Ereignis, in welcher das Ergebnis einer Aktion vom Eintreffen einer zeitlich parallel laufenden Aktion beeinflusst wird. [Site17]

Das zweite Framework, welches betrachtet wurde ist das MiniMAS aus JOHDA. JOHDA ist eine COHDA-Implementierung und das MiniMAS wurde von den Betreuern dieser Projektgruppe vorgeschlagen. Dabei handelt es sich um ein reduziertes Multiagentenframework, welches in seiner Grundform aus drei Elementen besteht. Diese sind das MiniMAS selbst, welches die Laufzeitumgebung der Agenten darstellt, Events, die die Nachrichten der Agenten zum nächsten Zeitpunkt beinhalten und den Agenten selbst. Der große Vorteil dieses Multiagentensystems ist jener Minimalismus. Zum einen lässt es sich einfach durch das gesamte System debuggen, da kein fertig compilierter Code verwendet wird. Die Erweiterbarkeit ist bei einer so geringen Anzahl an Klassen komfortabel. Der für die Projektgruppe größte Vorteil war aber die vollständig transparente Nachrichtenverteilung. Diese erfolgt durch das MiniMAS rundenbasiert. Dadurch, dass auch die Agenten rundenbasiert ausgeführt werden, ist es komplett nachvollziehbar, wann eine bestimmte Nachricht durch einen Agenten bearbeitet wird. Daher hat sich die Projektgruppe für das rundenbasierte MiniMAS entschieden, trotz der im späteren Abschnitt 5.3 erwähnten Grenzen.

### 5.1.3 Module

Bei der Implementierung des Prototypen wurde ein modulares System entwickelt, sodass einzelne Module spezielle Aufgaben haben. Dies hat zum einen den Vorteil der Kapselung, sodass verschiedene Aufgaben von verschiedenen Modulen erledigt

werden. Zum anderen bietet eine modulare Entwicklung den Vorteil der Austauschbarkeit. Einzelne Module können durch andere ersetzt werden, wenn beispielsweise andere Eingabe- oder Ausgabeformate gewünscht sind.

Dieser Abschnitt stellt die Module, aus denen das entwickelte System besteht sowie deren Zusammenhang, der in Abbildung 5.2 zu sehen ist. Ein Szenario wird im CSV-Manager-Modul ins System eingegeben. Das eingelesene Netz mit Knoten, Kanten und weiteren Eigenschaften wird ans Multiagentensystem übergeben. Auch wenn die Agenten zum Multiagentensystem gehören, sind sie in Abbildung 5.2 als eigene Komponente dargestellt. Dies hat den Grund, da die Assoziation zu Actions und Behaviours so deutlicher dargestellt werden kann. Der Printer ist für die Ausgaben zuständig, die im Agentensystem, in Actions und in Behaviours getätigt werden.

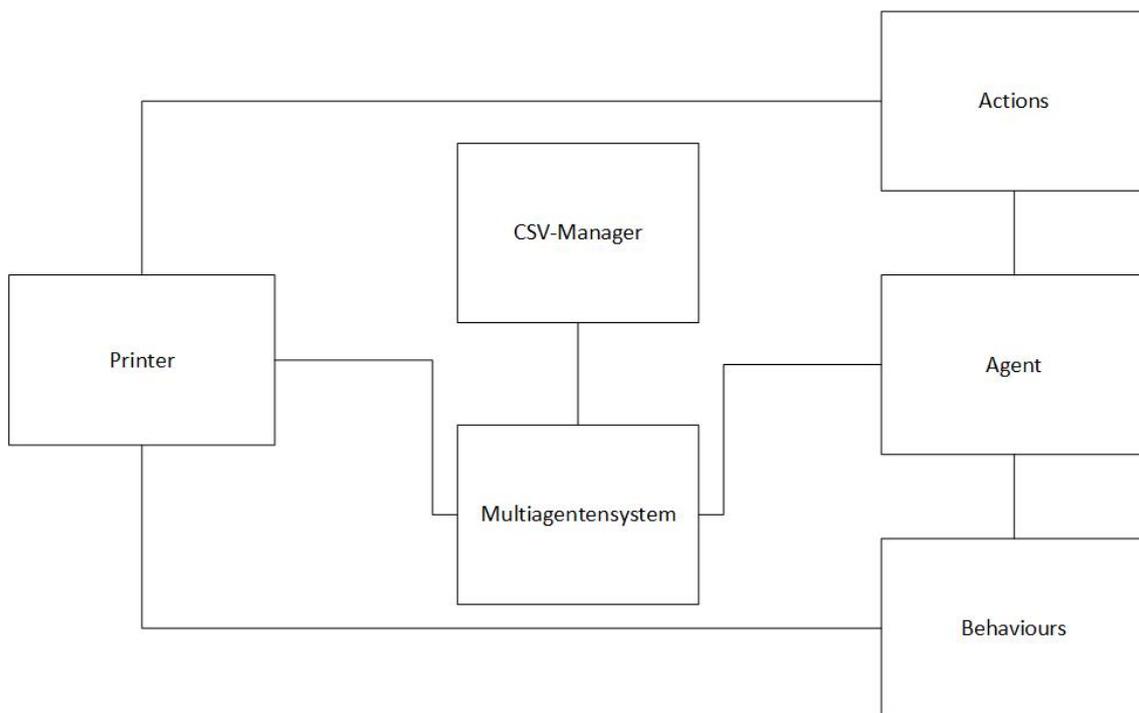


Abbildung 5.2: Module des implementierten Systems

### 5.1.3.1 CSV-Manager

Das zentrale Modul um die Ausführung des Systems zu starten ist der CSV-Manager. Mit Hilfe des CSV-Managers wird das zu lösende Netz in das System portiert. Hierfür werden drei CSV-Dateien für die Knoten, Strom- und Kommunikationskanten eingelesen und der Inhalt dieser für die weitere Verarbeitung vorbereitet. Beim einlesen der Dateien unterscheidet der CSV-Manager zwischen zwei generellen Funktionalitäten. Zum einen das Einlesen von Knoten samt ihren Powerwerten, den Zeitschritten, ihren aktivierbaren Nachbarn und den Prioritäten, zum anderen das Einlesen von fixen Strom- und Kommunikationskanten. Da die Kommunikation

über die aktivierbaren Nachbarn erst nach und nach möglich wird, sind die aktivierbaren Nachbarn den Knoten zuzuordnen und lassen sich nicht direkt wie die fixen Kanten über die Kanten-CSV-Datei einlesen. Beim Einlesen der Knoten geht der CSV-Manager Schritt für Schritt vor. Bei jedem Knoten werden zuerst die Fahrpläne mit den Powerwerten eingelesen, dann die aktivierbaren Nachbarn und zum Schluss die jeweilige Priorität des Knotens. Die Fahrpläne der Agenten werden in einer Liste gespeichert, die dem Multiagentensystem zu Beginn der Ausführung des Systems übergeben wird. Anhand dieser Liste werden die Agenten im System erzeugt. In weiteren Listen werden die aktivierbaren Nachbarn und die Prioritäten gespeichert, welche vom System nach der Erzeugung der Agenten den Agenten hinzugefügt werden. Nachdem die Kanten einer Kantendatei eingelesen wurden, werden sie in Listen gespeichert und dem Multiagentensystem übergeben, damit dieses die Kanten als Verbindungen zwischen den einzelnen Agenten herstellt und speichert.

### 5.1.3.2 Printer

Das Printer-Modul bietet die Möglichkeit, die Konsolenausgaben des Systems individuell zu konfigurieren. Mittels Flags kann hier eingestellt werden, welche Ausgaben gewünscht sind. So ist es möglich, jede Nachricht jedes Typen ausgeben zu lassen, wenn es gewünscht ist. In der Konfiguration, die für die Testphase (s. Kapitel 6) sollten Nachrichten nicht ausgegeben werden, lediglich Inseln und Agenten mit ihren Eigenschaften waren von Belang. Nach Terminierung des Systems wurde ausgegeben, wie oft welche Nachricht versandt wurde, was insbesondere für die Validierung in Kapitel 7 wichtig war.

Ebenso ist es die Aufgabe des Printer-Moduls, die Ausgaben zu formatieren. So wurden Methoden entworfen, die entsprechende Strings generieren und ausgeben. Im Code der Module, die den Printer nutzen, werden diese Methoden aufgerufen. Innerhalb dieser Methoden werden dann entsprechende Flags abgefragt, um den Output nur dann zu liefern, wenn er gewünscht ist.

### 5.1.3.3 Actions

Die Actions kapseln den Programmcode, den die Agenten ausführen, inhaltlich. Diese Kapselung bietet den Vorteil, dass jederzeit klar ist, in welcher Action Änderungen des Agenten-Verhalten vorzunehmen sind. Dies hat den Grund, dass jede Ausführung des Agenten sich einer Action zuordnen lässt. Jede Action ist in Form einer Klasse implementiert, die alle eine Default-Action als Superklasse haben. Der Agent steht mit jeder Action in einer 1:1-Beziehung. So kann der Agent auf die öffentlichen Methoden jeder Action zugreifen, gleichzeitig ist dies innerhalb der Actions umgekehrt möglich. Instanziierungen der Actions finden im Konstruktor des Pacman-Agenten statt, sodass diese Assoziation zu jedem Zeitpunkt existiert.

#### 5.1.3.4 Behaviours

Die Behaviour-Klassen repräsentieren das Zustandskonzept, wie in Abschnitt 4.2 vorgestellt. Der Agent hat ein Attribut, das den aktuellen Zustand wiedergibt. Dies kann in den Behaviours gesetzt werden. Die Behaviours erben alle von einem Default-Behaviour, das einen Superkonstruktor, Entry- und Exit-Methoden und die Nachrichtenverarbeitung bereitstellt, was die Subklassen implementieren.

Entry-Methoden setzen den Zustand des Agenten auf das jeweilige Behaviour, Exit-Methoden verlassen es in dasjenige, das als Argument übergeben wird. Das neue Behaviour wird als neues Objekt initiiert, dessen Konstruktor wiederum die Entry-Methode ausführt. Gibt es Handlungen, die bei Eintritt in einen Zustand jederzeit zu auszuführen sind, so werden innerhalb der Entry-Methode bereits Methoden der Actions aufgerufen. Dies ist beispielsweise beim Eintritt ins Overlay-Behaviour der Fall, wo das Overlay-Netz sofort erstellt wird. Ein Zustandswechsel kann ebenso eine Nachricht mit an den neuen Zustand übergeben, da in einigen Fällen Nachrichten als Trigger für Zustandswechsel fungieren und im jeweils neuen Zustand verarbeitet werden. Die im folgenden Absatz erklärte Nachrichtenverarbeitung findet ebenso in jedem Behaviour statt, da das Verhalten des Systems abhängig von aktuellem Zustand und eingehendem Nachrichtentyp ist.

Jedes Behaviour hat eine eigene Nachrichtenverarbeitung. Hier werden die Nachrichtentypen verzweigt, sodass eine Nachricht eines bestimmten Typen im jeweiligen Behaviour an eine Handlung des Agenten verknüpft ist. Mögliche Handlungen sind Zustandswechsel, Methodenaufrufe aus den Actions oder direkte Rückantworten auf die Nachricht.

#### 5.1.3.5 Graphstream

Mittels Graphstream<sup>1</sup> wurde dem entwickelten Prototypen eine grafische Ausgabe hinzugefügt. Dies dient einer anschaulichen Präsentation sowie einer guten Nachvollziehbarkeit der eingelesenen Netztopologie, andererseits wird so die optionale Anforderung S-03 erfüllt. Graphstream kann die Knoten und Kanten des Netzes visualisieren, mittels Färbungen wurde hier dargestellt, ob Leistungsschalter auf Kanten geschlossen oder offen sind. Dies ermöglicht dem Betrachter, zu sehen, welche Agenten bereits zu Inseln zusammengeschlossen sind. Der zeitliche Verlauf des Prozesses kann mittels Graphstream ebenfalls nachvollzogen werden, da die grafische Oberfläche des Graphstream-Moduls sich stetig aktualisiert. Die grafische Ausgabe des Systems ist exemplarisch in Abbildung 5.3 dargestellt. Hier sind vier Mittelspannungsringe mit davon abgehenden Niederspannungsträngen sichtbar.

---

<sup>1</sup><http://graphstream-project.org/>

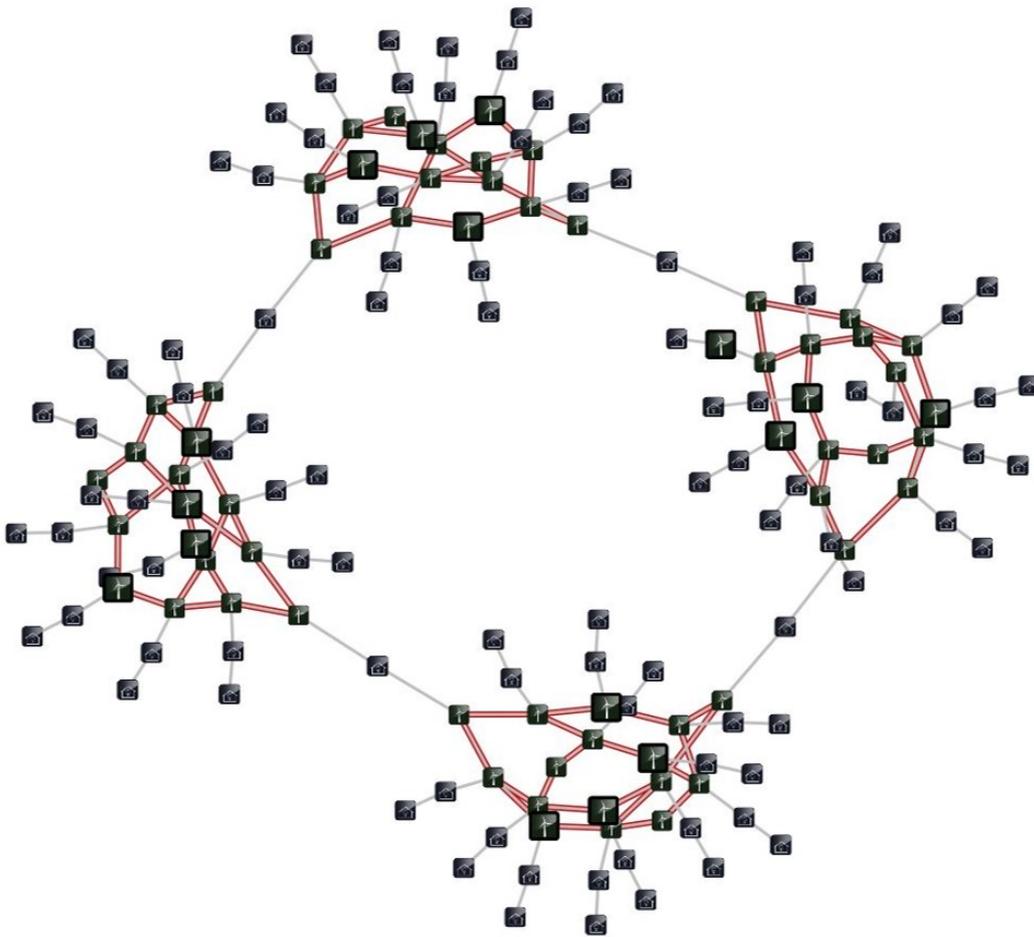


Abbildung 5.3: Visualisierung eines Szenarios mittels Graphstream

## 5.2 Pacman

Nachdem im vorigen Abschnitt die Architektur des Systems ausführlich beschrieben wurde, wird in diesem Abschnitt konkret auf die Implementierung des Verfahrens eingegangen, was der Entwicklung des Pacman-Systems entspricht. Zunächst wird erläutert, wie das Multiagentensystem aufgebaut ist, wie es arbeitet und wie das Programm terminiert. Dann wird auf die Agenten im System eingegangen, die verschiedenen Ausprägungen werden dabei vorgestellt und voneinander abgegrenzt. Zum Schluss werden die erstellten Nachrichtentypen und die Eingabeparameter erklärt.

### 5.2.1 Das MiniMAS

Wie in Abschnitt 5.1.2 erwähnt, wird im Prototypen das MiniMAS von JOHDA verwendet. Um den Entwurf des Verfahrens, wie in Kapitel 4 beschrieben, zu erfüllen und den Prototypen zu strukturieren, wurde das Multiagentensystem deutlich erweitert. Dabei sind zum einen Behaviours und Zustände hinzugefügt worden und zum anderen werden nun Actions für die eigentlichen Agententätigkeiten verwendet. Außerdem sind spezielle Module integriert, welche unter anderem für das Aufbauen der Szenarien, der Ausgabe der Ergebnisse und zur Visualisierung beitragen. Diese Modularisierung wird in Abschnitt 5.1.3 genauer erläutert. Im Folgenden wird das erweiterte Grundgerüst des MiniMAS genauer erläutert. Danach folgt eine Erläuterung der wesentlichen Funktionen der Laufzeitumgebung wie Events, Steps und Terminierung des Systems.

#### 5.2.1.1 Multiagentenplattform

Die Multiagentenplattform ist das zentrale Verwaltungs- und Kommunikations-System der Agentenlaufzeitumgebung. Es werden über diese nicht nur die einzelnen Agenten generiert, sondern auch angesprochen. Die zentralen Aufgaben sind dabei die Verteilung der Nachrichten zum jeweiligen Zeitpunkt und das Ansteuern des jeweiligen Arbeitsschrittes eines Agenten. Da im Pacman-System ein rundenbasiertes Ausführen der Agenten verwendet wird, sind die jeweiligen Zeitschritte ein elementarer Baustein der Funktionalität des Systems und werden hier Steps genannt. Diese werden in Abschnitt 5.2.1.3 genauer erläutert. Der Nachrichtenaustausch der Agenten erfolgt jeweils über sogenannte Events. Dabei besteht ein Event aus einer Nachricht und dem dazugehörigen Step, siehe Abschnitt 5.2.1.2. Der Ablauf des MiniMAS ist in Abbildung 5.4 dargestellt. Zunächst initialisiert das MiniMAS die Agenten. Danach erfolgt innerhalb einer Schleife die Verteilung von Nachrichten und das Ausführen der Agenten. Dabei ist die erste Routine ein Sonderfall, da die Startnachrichten vom Benutzer vergeben werden. Das Schleifenende und damit das Systemende werden von einem separaten Agenten, dem Konvergenzagenten, angesteuert. Dieser und seine Funktionen werden in Abschnitt 5.2.2 genauer beschrieben.

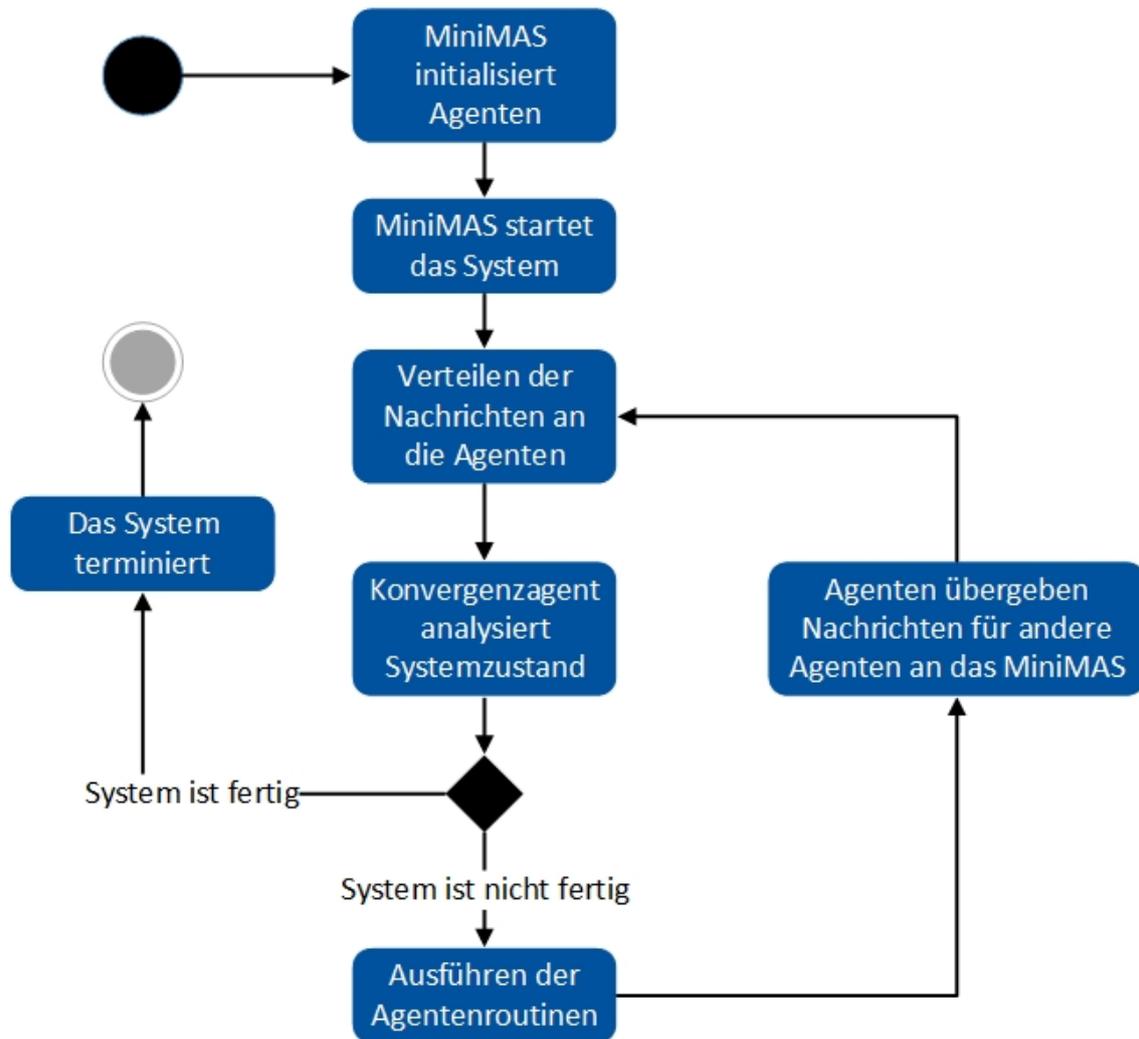


Abbildung 5.4: Aktivitätsdiagramm des MiniMAS

### 5.2.1.2 Events

Events sind die Reaktionsmechanismen des Systems. Ist ein Agent aktiv und trifft für diesen ein Event ein, muss dieser darauf reagieren. Dabei besteht ein Event aus zwei Bestandteilen, einem Step, in dem es Eintritt und einer Nachricht. Eine Nachricht wiederum besteht aus dem Betreff, der ID des Senders, der ID des Empfängers und dem Inhalt der Nachricht. Nachrichten und deren Ausprägungen, den Nachrichtentypen, sind im Abschnitt 5.2.5 näher erklärt. Der Aufbau der Events ist in Abbildung 5.5 dargestellt.

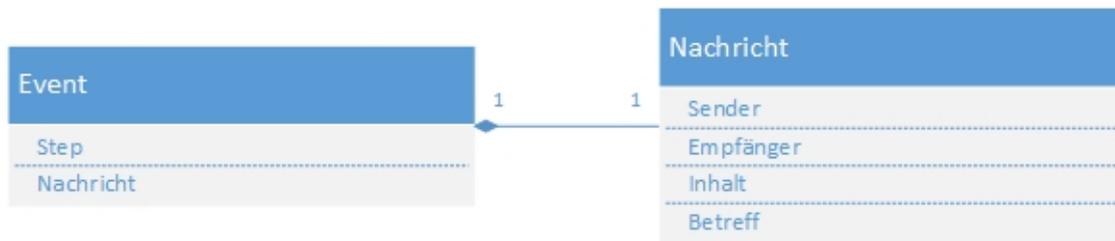


Abbildung 5.5: Aufbau der Events

### 5.2.1.3 Zeitschritte

Wie schon bereits beschrieben läuft das MiniMAS rundenbasiert ab. Das heißt, alle Agenten sind im gleichen Zeitschritt und können demnach nur Events für diesen bearbeiten. Das sorgt für eine deutlich bequemere Fehleranalyse und verhindert Race Conditions. Die Rundenbasiertheit resultiert dabei aus der Schleife des MiniMAS. Wie in Abbildung 5.5 dargestellt, werden erst alle Nachrichten eines Steps gesammelt und dann an die Empfänger verteilt. Die verteilten Nachrichten werden dann sequentiell innerhalb der Agentenroutinen abgearbeitet. Dabei versenden die Agenten in der Regel neue Nachrichten. Diese werden dann wieder gesammelt und wenn alle Agenten ihre Routinen abgeschlossen haben verteilt. Versendet ein Agent eine Nachricht so wird diese automatisch für den nächsten Zeitpunkt erstellt. Dadurch endet ein Zeitpunkt automatisch damit, dass für diesen keine Nachrichten mehr verfügbar sind. Innerhalb eines Zeitschrittes beginnt jeder einzelne Agent, angesteuert vom MiniMAS, mit der Abarbeitung der ihm zur Verfügung stehenden Nachrichten. Dabei ist das Verhalten der Agenten abhängig von Inhalt, Betreff und Sender der Nachricht und vom Status des bearbeiteten Agenten. Dies wird genauer in Kapitel 5.1.3.4 beschrieben. Die Zeitschritte werden in Kapitel 7 als Vergleichsinstrument für die Evaluation und Validierung verwendet.

### 5.2.1.4 Termination des Multiagentensystems

Das ursprüngliche MiniMAS aus JOHDA terminiert, wenn zu einem Zeitpunkt keine Nachrichten verteilt werden. Sendet kein Agent mehr Nachrichten, so muss das Agentensystem in einem Schritt keine einzige Nachricht verarbeitet werden, woraufhin es terminiert. Das MiniMAS, welches im Pacman-System verwendet wird terminiert durch einen speziell dafür entwickelten Konvergenzagenten. Dieser ist nicht nur einmalig im System, er ist auch der einzige Agent, welcher direkten Zugriff auf alle Informationen des Systems hat. Damit ist in diesem Modell eines verteilten Systems eine zentrale Einheit integriert. Diese Vereinfachung der Realität ist aber notwendig, da die Kernaufgaben der Projektgruppe nicht die Implementierung von verteilten Terminierungs- und Konvergenz-Algorithmen sind. Die genauere Beschreibung der Aufgaben des Konvergenzagenten folgt in 5.2.2. Das System terminiert wenn der

Konvergenzagent von allen aktiven Inseln eine Nachricht erhält, dass diese nichts mehr tun können also fertig sind. Dies trifft in den unter Kapitel 4.3.8 definierten Bedingungen ein.

## 5.2.2 Agenten

Das Multiagentensystem beinhaltet verschiedene Typen von Agenten, wobei jeder spezielle Eigenschaften und Aufgaben hat. Zunächst erben alle Agenten vom Default-Agenten, der grundlegende Eigenschaften generalisiert. Diese Typen sind im Folgenden konkret vorgestellt.

### 5.2.2.1 Default-Agent

Der Default-Agent bildet die Superklasse aller weiteren Agententypen und wird nicht instantiiert. Seine Attribute sind solche, die alle Agenten benötigen und somit in der Superklasse generalisiert werden können. Dies sind die ID des Agenten, die Assoziation zum Agentensystem, die zugehörige Insel und die Nachbarschaften. Ebenso können alle Agenten Nachrichten verarbeiten, sodass der Default-Agent eine Queue für die Speicherung eingehender Nachrichten besitzt. Um Ausgaben zu tätigen, ist die Printer-Klasse vonnöten, dieses Attribut ist ebenfalls Teil der Superklasse, da alle Agenten je nach Konfiguration des Printers für Ausgaben zuständig sein können. Neben Gettern und Settern deklariert der Default-Agent die abstrakte Step-Methode, die das Agentensystem für jeden Agenten in jedem Step aufruft. Diese wird durch alle Subklassen individuell implementiert.

### 5.2.2.2 Pacman-Agent

Der Pacman-Agent ist derjenige Agent, der alle Knotenagenten der Szenarien repräsentiert und somit alle die Eigenschaften der Energieanlagen abbildet. Dieser Agent führt das in Kapitel 4 beschriebene Verfahren durch und schaltet die Leistungsschalter. Somit sind alle Eigenschaften, die an die Optimierung, die Verhandlungen und die Bildung von Inseln gekoppelt sind, Attribute dieses Agenten.

Der Pacman-Agent beinhaltet den Planungshorizont, seinen aktuellen Zustand sowie den Zeitschritt, in dem er sich befindet, als Attribute. Dazu kommen alle Datenstrukturen, die zur Verwaltung der Leistungswerte benötigt werden. Dazu gehören die Liste mit allen Werten über alle Zeitschritte und die aktuell ausgewählte Leistung. Der Agent hat zudem ein Arbeitsgedächtnis, und einige Datenstrukturen, die bei der COHDA-Verhandlung beispielsweise zum Zwischenspeichern von Werten wichtig sind. Wie bereits in Abschnitt 5.1.3.3 erwähnt, wird das Verhalten des Pacman-Agenten in die Actions ausgelagert. So enthält der Agent ebenfalls die Assoziation zu diesen. Konstante Werte wie die Priorität des Agenten und die Gewichte der Zielfunktion sind weitere Attribute des Pacman-Agenten.

Der Großteil der Methoden sind solche, die die Attribute manipulieren können, dazu zählen Getter- und Setter-Methoden sowie Methoden, die Listen ergänzen oder Elemente aus ihnen löschen können. Des Weiteren hat der Pacman-Agent die Step-Methode. Diese Methode ruft je nach Zustand ein neues Behaviour auf, sodass dort wie in Abschnitt 5.1.3.4 beschrieben die Nachrichtenverarbeitung stattfinden kann.

### 5.2.2.3 Schalter-Agent

Schalter-Agenten repräsentieren die Leistungsschalter, die im Szenario zwischen zwei Energieanlagen, die durch Pacman-Agenten abgebildet sind, liegen. Somit sind Schalter-Agenten direkt an der Bildung von Inseln beteiligt, ihr Schließen ermöglicht dies gar erst. Einziges Attribut dieses Agenten ist sein Schaltzustand, dieser kann offen oder geschlossen sein, wobei offen der initiale Wert ist. Da das Verhalten des Agenten rein auf Schaltvorgänge limitiert ist, muss die Nachrichtenverarbeitung nicht weiter in Behaviours ausgelagert werden, sie findet direkt in der Step-Methode des Agenten statt. Dabei kann der Agent lediglich Schließbefehle empfangen, woraufhin dieser schließt und diesen Schließbefehl beim Absender der Nachricht bestätigt. Ein geschlossener Schalter kann in dieser Implementierung nicht wieder geöffnet werden, der Algorithmus, der in Kapitel 4 dargestellt wird, sieht dies auch nicht vor. Weitere Methoden und Attribute besitzt dieser Agent nicht, seine Nachbarschaft kennt er durch die Attribute seiner Superklasse.

### 5.2.2.4 Konvergenz-Agent

Der Konvergenz-Agent ist ein Konstrukt, das die Erkennung der Konvergenz von Verhandlungen zentralisiert. Das meint, die Agenten, die dezentral agieren, werden von diesem übergeordneten Agenten überwacht und erfahren die Konvergenz ihrer Verhandlung von ihm. Implementiert wurde dies so, dass ein Konvergenz-Agent, der mit allen Pacman-Agenten kommunizieren kann, in jedem Step des Agentensystems die Konvergenz von allen Verhandlungen prüft. Um Analogien zu den anderen bereits vorgestellten Agenten zu wahren, erfolgt diese kontinuierliche Prüfung in der Step-Methode des Agenten.

Um die Arbeitsweise dieses Agenten deutlich zu machen, werden zunächst dessen Attribute erläutert. Es existiert eine Liste mit aktiven Inseln, sodass jederzeit bekannt ist, welche Inseln vorhanden sind. Zudem sind ihm alle Pacman-Agenten bekannt. Die Information über konvergierte Inseln wird in einer Hashmap gespeichert, die die Insel als Schlüssel und eine boolesche Variable als Wert enthält. Ebenso wird hier gespeichert, welche Inseln bereits konvergiert sind und welche aufgrund einer neu gestarteten Verhandlung die Möglichkeit der Konvergenz besitzen.

Wie auch der Schalter-Agent und gegensätzlich zum Pacman-Agenten benötigt der Konvergenz-Agent kein Zustandsmodell, seine Nachrichtenverarbeitung findet direkt

innerhalb der Step-Methode statt. Dabei kann dem Konvergenz-Agenten einerseits das Ende einer Verhandlung mitgeteilt werden, sodass die sendende Insel in die Liste konvergierter Inseln aufgenommen wird. Andererseits meldet eine Insel ebenso an den Konvergenz-Agenten, wenn eine neue Verhandlung begonnen hat. Dann wird diese Insel analog in die Liste derjenigen Inseln hinzugefügt, die zur Konvergenz fähig sind.

In der Step-Methode wird nach der erwähnten Nachrichtenverarbeitung eine Check-Methode aufgerufen, die nun für die jeweiligen Inseln prüft, ob sie konvergiert sind. Dafür wird für alle Agenten in der Insel geprüft, ob sie unverarbeitete WorkingMemory-Nachrichten (s. 5.2.5) haben. Dies wird ebenso für alle Nachbarn dieser Insel geprüft, allerdings bezieht sich diese Prüfung nur auf Nachrichten aus der Insel. Werden in beiden Schritten keine Nachrichten gefunden, so erhält der Inselsprecher der Insel eine Konvergenz-Nachricht, sodass die Insel von ihrer Konvergenz erfährt. Ebenso ist der Konvergenz-Agent für die Terminierung des Multiagentensystems zuständig. Wurde im System von allen Inseln gemeldet, dass sie nichts mehr tun können, also fertig sind, dann wird die Prozedur des Multiagentensystems abgebrochen.

### 5.2.3 Actions

Es wurden folgende Actions implementiert:

- Connect-Action
- Island-Ask-Action
- Optimize-Action
- Overlay-Action
- Start-Action
- Switch-Action
- Working-Memory-Action

Die Connect-Action beinhaltet alle Handlungen, die zum Ablauf der Inselvereinigung gehören. Die Vorbereitung dieser findet in der Island-Ask-Action statt, hier können andere Agenten gefragt werden, ob sie in einer sind, dazu werden diejenigen Agenten ermittelt, an die diese Frage gestellt werden kann. Alles, was mit der COHDA-Optimierung zu tun hat, beispielsweise sind dies Berechnungen von Bilanzen oder Blacklist-Aktionen, findet in der Optimize-Action statt. Da das Arbeitsgedächtnis

in der Verhandlung eine besondere Rolle spielt und auf ihm sehr viele Operationen ausgeführt werden müssen, wurde ihm eine eigene Action gewidmet, die Working-Memory-Action. Initialisiert wird eine COHDA-Verhandlung mittels Versenden von Overlay-Netzen. Dieses Netz wird in der Overlay-Action erstellt und versendet. Da zu Beginn des Programms einige Initialisierung und das Setzen von Start-Parametern von Nöten sind, wurde hierfür die Start-Action implementiert. Die Default-Action hat die Aufgabe, die 1:1-Beziehung zum Agenten herzustellen und instanziiert ein Objekt des in Abschnitt 5.1.3.2 vorgestellten Printers.

#### 5.2.4 Behaviours

Die implementierten Behaviours entsprechen den in Abschnitt 4.2 beschriebenen Zuständen aus dem Zustandkonzept des Systems. Außerdem dient ein abstraktes Default-Behaviour als Superklasse aller Behaviour. Demnach wurden folgende Behaviours erstellt:

- Default-Behaviour
- COHDA-Behaviour
- Connected-Behaviour
- FindIsland-Behaviour
- FindNeighbours-Behaviour
- MergeIslands-Behaviour
- Overlay-Behaviour
- Sleep-Behaviour
- Wait-Behaviour

In diesen Behaviours wird auf Nachrichten bestimmter Typen reagiert, diese werden im nachfolgenden Abschnitt 5.2.5 erklärt

#### 5.2.5 Nachrichtentypen

Das Agentensystem kommuniziert durch das Versenden von Nachrichten. Sender und Empfänger sind dabei IDs von Agenten, das MiniMAS stellt diese Nachrichten entsprechend zu. Der Inhalt ist als Java-Datentyp „Object“, was Superklasse aller anderen Klassen ist, sodass jedes Objekt im System Inhalt einer Nachricht sein kann. Der Betreff ist dabei ein Integer, der für einen bestimmten Nachrichtentypen

steht. Diese werden in einem Interface definiert, sodass im gesamten System auf die Variablennamen zugegriffen werden kann. Diese Nachrichtentypen spielen eine besondere Rolle im System, da die Behaviours die Nachrichten hiernach filtern und Methoden der Actions aufrufen. In den Behaviours wird dabei auf unterschiedliche Nachrichtentypen unterschiedlich reagiert. Diese Nachrichtentypen sind in Tabelle 5.1 aufgelistet und erklärt. Dabei wird ersichtlich, dass nicht alle Nachrichten einen Inhalt haben. Diese drücken ihre Information allein durch ihren Betreff aus. Im Programm wird hier das leere Objekt „null“ als Parameter eingegeben.

## 5.3 Grenzen des Systems

Die verwendeten Technologien sind im Pacman-System exakt auf die in Abschnitt 3.2 beschriebenen Szenarien ausgerichtet. Da diese nur einen modellhaften Bezug zur Realität haben, ist das Pacman-System von der direkten Nutzung in der Realität abzugrenzen. Zum einen wird bisher die Kommunikation zwischen verteilten Systemen nicht beachtet. Hierbei müsste die Einführung von Kommunikationsprotokollen, wie TCP/IP, und Standards, wie FIPA, eingeleitet werden. Dazu gehört aber auch die Problematik des nicht verteilten Multiagentenframeworks. Anders als bei JADE werden im verwendeten MiniMAS keine prozessgetrennten Container aufgebaut. Das Multiagentensystem läuft komplett in einem Prozess. Dadurch entsteht unter anderem eine problematische Referenzierung der Objekte. Es müssen daher, manuell, Objektkopien beim Nachrichtenversand erstellt werden damit Agenten nicht auf gemeinsamen Ressourcen arbeiten.

Des Weiteren wird durch die klar definierte Arbeitsreihenfolge, welche wiederum aus der Rundenbasiertheit des MiniMAS resultiert, Race Conditions verhindert, da niemals parallel gearbeitet wird. In einem realitätsnäheren Modell sollten die Agenten zumindest parallel arbeiten. Wie oben schon beschrieben müssten Kommunikationsstandards mit berücksichtigt werden. Dabei kann das aktuelle System bisher noch nicht mit Verzögerungen oder Fehlern bei der Kommunikation umgehen. Falls eine Insel, aus solch einem Grund, eine Nachricht zu spät bekäme, könnte das System unbeabsichtigt terminieren. Die Terminierung und Konvergenz sind dabei ebenfalls noch nicht an die Realität angepasst. Wie schon in Abschnitt 4.3.3.1 beschrieben, ist die Implementierung einer verteilten Konvergenzerkennung keine Aufgabe der Projektgruppe. Ebenso kann die Implementierung eines verteilten Terminierungsalgorithmus betrachtet werden. Dies ist aber ebenfalls eine Grenze des aktuellen Prototypen. Des Weiteren sind in diesem System die Schalter und deren Funktionen auf eine triviale Art implementiert. Hier steht die Annahme voraus, dass die Infrastruktur der einzelnen Akteure in der Realität schon durch eine Schnittstelle ansteuerbar ist.

Betreff	Erklärung	Inhalt
Start	Zentrale Nachricht vom System an die Sprecher, um das System zu starten	Gewichte der Zielfunktion
Connect	Nachricht, die das Aufnehmen eines Agenten in eine Insel veranlasst	Insel-Objekt
Abort	Nachricht, die eine Verhandlung abbricht	
WorkingMemory	Versendetes Arbeitsgedächtnis	Arbeitsgedächtnis
AddedMeToIsland	Mit dieser Nachricht meldet sich ein neu aufgenommenener Agent bei der neuen Insel	Insel-Objekt
UpdateIslandObject	Hat die Insel sich verändert, werden die neuen Informationen versendet	Insel-Objekt
OverlayNet	Versendet das Overlay-Netz mit den an der Verhandlung teilnehmenden Agenten	Liste mit Agenten
IslandConvergence	Hiermit meldet der Konvergenz-Agent einer Insel ihre Konvergenz, ebenso wird deren Antwort hiermit gesendet	
AskIfInIsland	Hiermit wird ein Agent gefragt, ob er sich in einer Insel befindet	
IslandAgree	Hiermit wird bestätigt, dass eine Insel-Vereinigung stattfinden kann	
StartedNewNegotiation	Dem Konvergenz-Agenten wird gemeldet, dass eine neue Verhandlung begonnen hat	
Finished	Endet eine Verhandlung, wird diese Nachricht an den Konvergenz-Agenten gesendet	
ReadyForCOHDA	Wird gesendet, um die Bereitschaft zu einer Verhandlung zu signalisieren	
Agree	Positive Antwort auf eine Ready-For-COHDA-Nachricht	
Disagree	Negative Antwort auf eine Ready-For-COHDA-Nachricht	
CloseSwitch	Schaltbefehl an Schalter-Agenten	
SwitchSuccess	Bestätigung des Schaltbefehls	
ActivatedCommunication	Hat sich die Nachbarschaft durch aktivierbare Nachbarn erweitert, so wird dies an jene Nachbarn versendet, damit diese ihre Nachbarschaft aktualisieren	Liste mit Agenten
NewNeighbourhood	Sendet eine veränderte Nachbarschaft einer Insel an beteiligte Agenten	Liste mit Agenten

Tabelle 5.1: Übersicht über die verwendeten Nachrichtentypen.

# 6. Tests

Dieses Kapitel beschreibt die Tests des von der Projektgruppe entwickelten Systems. Zu Beginn wird das dabei gewählte Vorgehen erläutert. Der Abschnitt Vorbereitung erläutert insbesondere die Erstellung der Testfälle und -szenarien, bevor die Durchführung der Tests dargestellt wird. Abschließend erfolgt die Auswertung der Ergebnisse.

## 6.1 Vorgehen

Im Bereich der Softwaretechnik, insbesondere bei verteilten Systemen, ist ein Nachweis der vollständigen Korrektheit des entwickelten Systems beinahe unmöglich [Baum06]. Daher wurden bei den Tests des von der Projektgruppe entwickelten Systems die Ziele verfolgt den Nachweis der Erfüllung der zugrundeliegenden Anforderungen zu erbringen, sowie möglichst viele Fehler aufzufinden und zu beheben.

Aufgrund der hohen Abhängigkeit der verschiedenen Module des Systems untereinander wurden die Tests als Integrationstest durchgeführt. Des Weiteren wurden die Tests auf Basis des Ziels des Nachweises der Erfüllung der Anforderungen mit dem Black-Box-Verfahren durchgeführt. Da das System von vielen Eingabeparametern und auch deren Kombinationen abhängig ist, war eine besondere Herausforderung im Rahmen der Tests eine möglichst große Abdeckung zu erzielen, gleichzeitig den Aufwand in einem handhabbaren Ausmaß zu halten und Mehrfachtests zu vermeiden. Daher wurden zur Erstellung der Testfälle Äquivalenzklassen gebildet. Diese teilen die Wertebereiche der Eingabeparameter in Klassen mit ähnlichen zu erwartendem Verhalten ein. Das Ziel dieses Vorgehens ist mit möglichst wenig Testfällen möglichst viele Fehlerquellen zu überprüfen und somit die Komplexität zu reduzieren [Wetz05].

Auf Basis der aus den gebildeten Äquivalenzklassen abgeleiteten Testfälle wurden im nächsten Schritt Testszenarien entworfen, getestet und dokumentiert. Auftretene Fehler wurden im Rahmen der Testphase analysiert, bewertet und gegebenenfalls direkt behoben. In den folgenden Kapiteln wird die Umsetzung dieses Vorgehens detailliert beschrieben.

## 6.2 Vorbereitung

Dieser Abschnitt beschreibt, welche Schritte zur Vorbereitung der Testphase zu tätigen waren. Zunächst musste identifiziert werden, welche Eingabeparameter das System besitzt, woraufhin diese in Äquivalenzklassen eingeteilt wurden. Durch gezielte Kombination dieser Klassen wurden Testfälle erstellt. Diese Testfälle beinhalten sowohl lösbare als auch nicht-lösbare Szenarien, um alle in Abschnitt 3.3.2 spezifizierten Anforderungen abzudecken.

### 6.2.1 Identifizieren der Eingabeparameter

Um das System in möglichst vielen Ausprägungen zu testen, müssen alle unabhängigen Variablen gefunden werden, die am System einstellbar sind und sein Verhalten beeinflussen. Diese identifizierten Eingabeparameter sind in Tabelle 6.1 aufgeführt und erklärt.

#### Szenario

Die Definition dieser Eingabeparameter wurde dabei von einigen Rahmenbedingungen begrenzt. Das einzulesende Szenario, bestehend aus Strom- und Kommunikationsnetz, musste einer Small-World-Topologie mit vermaschtem Mittelspannungsring und strahlungsförmigen Niederspannungssträngen entsprechen, sodass andere Topologien eines Graphen nicht zu testen waren. Diese Vorgabe ergeben sich aus der Szenariodefinition, die in Abschnitt 3.2 erklärt wurde. Neben der Gewichtung der Zielfunktion waren alle weiteren Eingabeparameter solche, die die Gestaltung des Szenarios betreffen. Dabei muss ein Parameter die Größe dieses Szenario beschreiben, die durch die Anzahl der enthaltenen Knotenagenten beschrieben wird. Diese Agenten werden in einem oder mehreren Mittelspannungsnetzen platziert, die untereinander verbunden sind. Da die Verhandlungen dezentral, also in jedem dieser Netze zunächst separat, geführt werden, hat auch diese Anordnung der Agenten Einfluss auf das System.

#### Ausprägung der Energieanlagen

Des weiteren gibt es im System statische, dynamische und flexible Energieanlagen, die jeweils Erzeuger oder Verbraucher sein können. Da flexible Anlagen aufgrund ihrer Fähigkeit, Leistungswerte auszuwählen, die größte Bedeutung im System haben, ist ihr Anteil verglichen mit Gesamtzahl der Energieanlagen ein Eingabeparameter. Den verbleibenden Anteil im Szenario teilen sich dynamische und statische Anlagen zu gleichen Teilen. Ein wesentlicher Bestandteil des Systems sind die Sprecher, da diese Verhandlungen initiieren und führen. Somit ist die Anzahl dieser eine wichtige Stellgröße im System, hier wird sie als Anteil von flexiblen Anlagen beschrieben.

In der hier stattfindenden Testphase sind alle Sprecher ebenso flexible Erzeuger oder Verbraucher. Beschreibt nun die erwähnte Anzahl der Agenten die Größe des Szenarios, so muss der Parameter „Anteil der Agenten im Mittelspannungsnetz“, wie die Agenten auf die Spannungsniveaus verteilt sind.

### **Anpassungen der Netztopologie**

Weitere wichtige Eingabeparameter sind Eigenschaften der Netztopologie, die beschreiben, wie die Knotenagenten mittels Kanten miteinander verbunden sind. Der Mittelspannungsteil eines Netzes ist als vermaschter Ring definiert. Dies bedeutet eine ringförmige Anordnung, in der zusätzlich sogenannte Shortcuts als Direktverbindungen existieren, die solche Agenten miteinander verbinden, die in der Ringstruktur nicht benachbart sind. Der Grad dieser Vermaschung ist dabei ein wesentlicher Eingabeparameter, da solche Shortcuts die Sichtbarkeit die Möglichkeiten zur Verhandlung erweitern und eigentlich entferntere Netzbereiche direkter zugänglich machen. Niederspannungsstränge haben die Eigenschaft, dass je zwei von ihnen lediglich mittelbar über das Mittelspannungsnetz mit Stromkanten verbunden sind. Jedoch kann es vorkommen, dass diese Stränge eine direkte Kommunikationsverbindung haben. Dies ist ein Eingabeparameter, da unter anderem die Sichtbarkeit von Knoten ein Optimierungsziel ist und diese Niederspannungsknoten so eine höhere Sichtbarkeit haben. So wird die Optimierung beeinflusst, was direkten Einfluss auf die Ausgabe des Systems hat. Niederspannungsnetze sind strahlenförmig, das heißt, sie bestehen aus einem langgezogenen Strang mit beliebig vielen Zweigen. Stellen, an denen sich dieser Strang verzweigt, beeinflussen das Verfahren insofern, als dass hier eine Entscheidung gefällt werden muss, welche Agenten mit in die Insel eingefügt werden. Ist ein solcher Verzweigungspunkt am Rand einer Insel, so gibt es an dieser Stelle zwei mögliche Wege zu gehen, die beide in die Verhandlung miteinbezogen werden. Im Mittelspannungsring ist die Anzahl der Kommunikationskanten im Vergleich zur Anzahl der Stromkanten relevant. Diese entscheidet darüber, ob über Stromkanten verbundene Agenten nur über Umwege ansprechbar sind oder ob - im anderen Fall - Anlagen ansprechbar sind, die aber nicht unmittelbar zuschaltbar sind. Aus den genannten Gründen ist auch dies ein wichtiger Parameter, der im Verfahren eine Rolle spielt.

### **Leistungswerte der Agenten**

Der Planungshorizont beschreibt, für wie viele Zeitschritte in der Zukunft optimiert wird. Da eine Lösung nur gültig ist, wenn die Bilanz einer Insel in allen Zeitschritten Null ist, so wird die Verhandlung durch einen größeren Planungshorizont komplexer, sofern die zukünftigen Werte aller Agenten sich nicht periodisch wiederholen. Ebenso sind die Abstände der Powerwerte eines Agenten ein Eingabeparameter, der das

Verhalten des Systems beeinflusst. Sind diese unregelmäßig verteilt, so erschwert dies das Ausgleichen der Bilanz, da die Flexibilität des Agenten eingeschränkt wird. Ebenso spielt die Anzahl der möglichen Powerwerte eine Rolle, da diese Liste nach der jederzeit bestmöglichen Lösung abgesucht werden muss. Je größer die Liste, desto größer ist der Aufwand der sequentiellen Suche.

### **Zielfunktion**

Im Laufe der Optimierung entscheidet die Gewichtung der Zielfunktion darüber, wie stark die Kriterien Inselgröße, Sichtbarkeit und Zuschaltung priorisierter Agenten einfließen. Letzteres hängt davon ab, wie viele Agenten priorisiert werden und wo diese im Netz platziert werden.

### **Sonstiges**

Was bei den Eingabeparametern keine Rolle spielt, sind konkrete Zahlenwerte. Da die Bilanz einer Insel sich durch einfache Addition aller Powerwerte bildet, ist es lediglich von Bedeutung, wie sich diese Werte zueinander verhalten. Bei den Prioritätswerten der Agenten ist dies analog, hier geht es darum, welcher Agent beziehungsweise welche Insel höher priorisiert ist als ein anderer Lösungskandidat.

Die hier aufgeführten Eingabeparameter bilden die wesentlichen Größen, die für das Verhalten des Systems von Bedeutung sind. Neben Größe und Anordnung der einzelnen Agenten spielt deren Verbindung mittels Kanten eine Rolle. Ferner ist wichtig, welche Leistungswerte die einzelnen Agenten haben, wie regelmäßig diese verteilt sind und wie die Prioritätswerte der Knoten verteilt sind. Dazu kommt die Eigenschaft, für wie viele Zeitschritte in der Zukunft optimiert wird. Neben diesen Eigenschaften, die die Ausgestaltung des Szenarios beschreiben, ist die Wahl der Gewichtungsfaktoren der Zielfunktion entscheidend für das Verhalten des Systems.

Parameter	Beschreibung
Anzahl der Agenten	Zahl der Energieanlagen im zu testenden Netz
Anzahl der Mittelspannungsringe	Gibt an, aus wie vielen einzelnen, aber verbundenen Mittelspannungsnetzen das Szenario besteht
Anteil der Sprecher	Gibt an, welcher Anteil der flexiblen Anlagen als Sprecher fungiert
Anteil der flexiblen Anlagen	Gibt an, wie viele Energieanlagen relativ zur Gesamtzahl der Anlagen flexible Erzeuger oder Verbraucher sind
Anteil der Agenten im MS-Netz	Gibt an, wie viele der Agenten sich im Mittelspannungsteil des Stromnetzes befinden
Kommunikations-Shortcuts im Niederspannungsnetz	Gibt an, wie viele Kommunikations-Verbindungen es zwischen einzelnen Niederspannungssträngen gibt
Planungshorizont	Gibt an, für wie viele Zeitschritte in der Zukunft optimiert wird.
Abstände der Powerwerte	Gibt an, wie regelmäßig die Abstufungen der Powerwerte flexibler Anlagen sind.
Anzahl der Powerwerte flexibler Anlagen	Gibt an, aus wie vielen verschiedenen Werten eine flexible Anlage auswählen kann
Vermaschung des Mittelspannungsringes	Gibt an, wie viele abkürzende Verbindungen es im Mittelspannungsring gibt. Dies meint Verbindungen zweier Agenten mittels Kommunikations- und Stromkanten
Verzweigung der Stränge im Niederspannungsnetz	Gibt an, wie viele Abzweigungen es bei den strahlenförmigen Niederspannungssträngen gibt
Anzahl an Kommunikationskanten im Mittelspannungsring	Gibt an, wie viele Kommunikationsverbindungen im Mittelspannungsteil vorhanden sind.
Knotenprioritäten	Gibt an, wie viele Agenten im Szenario priorisiert werden
Gewichtung der Zielfunktion	Gibt an, welche Gewichtungen die Komponenten der Zielfunktion (Inselvergrößerung, Erweiterung der Sichtbarkeit, Priorisierung) haben

Tabelle 6.1: Liste der Eingabeparameter des Systems und deren Beschreibung

## 6.2.2 Bildung von Äquivalenzklassen

Die in Abschnitt 6.2.1 definierten Parameter sind nun in Äquivalenzklassen einzuteilen, da nicht jeder mögliche Wert eines Parameters testbar ist. Da aufgrund eines zu großen Aufwands nicht jeder Wert, den ein Parameter annehmen kann, in Kombination mit allen anderen Werten anderer Parameter zu kombinieren ist, muss jeder Parameter in Kategorien eingeteilt werden. Dabei sind gleichwertige Ausprägungen zu einer Klasse zusammenzufassen, sodass eine handhabbare Anzahl an Klassen pro Parameter entsteht.

### Anzahl und Ausprägung von Energieanlagen

Die Anzahl der Agenten wurde für die Testphase auf maximal 100 begrenzt. Die Grenze der beiden erstellten Äquivalenzklassen wurde dabei auf 40 gesetzt. Somit gehören zu Klasse 1 Szenarien mit bis zu 40 Agenten. Dies sind kleinere Szenarien, die vom Anwender überschaubar werden können. Bei größeren, hier mit 41 bis 100 Agenten ist dies nicht mehr der Fall, diese stellen viel mehr die verfügbare Rechenleistung auf die Probe. Bei der manuellen Erstellung dieser Netze ist dabei das Verhalten des Systems nur noch schwer vorhersagbar, was das Testen von Szenarien dieser Größe sehr wichtig macht. Bei der Anzahl der Mittelspannungsringe wird lediglich zwischen einem und mehreren unterschieden. Ein Ring ist dabei der Standardfall, bei dem alle zu testenden Eigenschaften zum Vorschein kommen sollten. Mehrere Ringe sorgen dann für mehr Dezentralität, die dahin führen soll, dass sich im Lauf der Verhandlung alle Ringe zu einer großen Insel zusammenschließen. Die Zahl der Sprecher wurde auf 50% der flexiblen Anlagen gesetzt, genauere Performanz-Aussagen zur Anzahl der Sprecher wurden in der folgenden Validierungsphase getroffen. Die zweite Äquivalenzklasse besteht aus Szenarien mit nur einem Sprecher, dieser muss das Netz iterativ aufbauen. Dabei spielen Erreichbarkeit aller Knoten und die Existenz gültiger Zwischenlösungen eine besondere Rolle. Der Agent muss die Insel dabei so lange erweitern, wie es bei ausgeglichener Bilanz eine jeweils bessere Lösung gibt. Der Anteil flexibler Anlagen wurde in drei Äquivalenzklassen eingeteilt. Wenige lagen bei 1-25% aller Agenten vor, viele bei 51-85%. Der dazwischen liegende Bereich von 26-50% beschrieb die Mitte. Hier wurden drei Abstufungen gewählt, da flexible Erzeuger den Aufbau des Netzes stark begünstigen und hier die Einflüsse jeder Klasse beobachtet werden sollte. In zwei Klassen geteilt wurde der Anteil der Agenten im Mittelspannungsnetz. Da die Topologie, bestehend aus Mittelspannungsring und Niederspannungssträngen, vorgegeben war, wurde der maximale Anteil an Agenten im Mittelspannungsteil auf 50% gesetzt. Um hier verschiedene Ausprägungen zu testen, wurde dieser Bereich mittig geteilt, sodass zwei Äquivalenzklassen entstanden sind.

Parameter	Klasse 1	Klasse 2	Klasse 3
Anzahl der Agenten	1-40	41-100	
Anzahl der Mittelspannungsringe	1	>1	
Anteil der Sprecher	1	50%	
Anteil der flexiblen Anlagen	1-15%	16-50%	51-85%
Anteil der Agenten im MS-Netz	1-25%	26-50%	
Kommunikations-Shortcuts im Niederspannungsnetz	0	>0	
Planungshorizont	1	>1	
Abstände der Powerwerte	regelmäßig	unregelmäßig	
Anzahl der Powerwerte flexibler Anlagen	10	100	
Vermaschung des Mittelspannungsringes	0%	10-60%	
Verzweigung der Stränge im Niederspannungsnetz	vorhanden	nicht vorhanden	
Anzahl an Kommunikationskanten im Mittelspannungsring	weniger als Stromkanten	gleich viele wie Stromkanten	mehr als Stromkanten
Knotenprioritäten	vorhanden	nicht vorhanden	
Gewichtung der Zielfunktion	(0,8;0,1;0,1)	(0,1;0,8;0,1)	(0,1;0,1;0,8)

Tabelle 6.2: Auflistung der Äquivalenzklassen der Eingabeparameter des Systems

### Anpassungen der Netztopologie

Kommunikationsshortcuts zwischen je zwei Niederspannungssträngen beeinflussen insbesondere die Sichtbarkeit von Agenten und Inseln. Hier alle Ausprägungen zu testen, wird wie bei allen Parametern aufgrund der Kombinatorik zu aufwendig. Um die wesentlichen Auswirkungen dieses Parameters zu testen, reichen jedoch zwei Äquivalenzklassen. Zum einen gibt es keinerlei dieser Verbindungen, zum anderen

gibt es sie. Im ersten Fall ist eine Kommunikation bzw. Verhandlung mit Teilnehmern aus mehreren Strängen nur möglich, wenn der Mittelspannungsteil die Verbindung herstellt.

Bei der Vermaschung des Mittelspannungsnetz wurde ein geringer bis mittlerer Grad an Vermaschung verwendet. Dies misst sich an der maximal möglichen Zahl an Querverbindung im Ring, was einem vollständig vermaschten Graphen gleichkäme. Die hier gewählte Bereich von 10-60% Vermaschung entspricht einem Ring mit einigen Abkürzungen, was der Realität nahkommt. Der Prozentbereich ist hier so weit gefasst, da bei einer kleinen Zahl an Agenten im Ring eine einzelne zusätzliche Kante einer Steigerung von mehreren Prozentpunkten entsprechen kann. Eine zweite Äquivalenzklasse war ein Ring ohne Maschen, was hier einen Grenzfall darstellte. In diesem Mittelspannungsteil müssen Strom- und Kommunikationskanten keinesfalls kongruent sein. Sind mehr Strom- als Kommunikationskanten vorhanden, so müssen häufiger zuschaltbare Agenten erst über Umwege angesprochen werden. Andersherum existiert häufiger zwischen Agenten eine direkte Kommunikationsverbindung, eine Zuschaltung ist aber erst möglich, wenn die Agenten durch Zuschaltung anderer Agenten Stromnachbarn geworden sind. Hieraus leiten sich zwei Äquivalenzklassen ab, eine dritte ist der Fall, dass gleich viele Kanten beider Art vorhanden sind. Verzweigungen im Niederspannungsnetz stellen die Optimierung häufiger vor Entscheidungen, einen der möglichen Wege zu gehen und gleichzeitig am Ende der Optimierung beide Zweige in die Insel aufgenommen zu haben. So wird diese Parameter in zwei Äquivalenzklassen unterteilt, einmal ohne jegliche Verzweigung, einmal mit dieser.

### **Leistungswerte der Agenten**

Ähnlich erfolgte die Bildung der beiden Äquivalenzklassen des Planungshorizonts. In der Optimierung ist ein großer Planungshorizont zwar komplexer, jedoch muss in jedem Fall für alle vorhandenen Zeitschritte eine gültige Lösung gefunden werden. Die Qualität einer Bilanz errechnet sich auch durch den Betrag des Vektors, der aus den Bilanzwerten aller Zeitschritte gebildet wird. Somit unterscheiden sich Planungshorizonte von mehr als Eins nicht in ihren wesentlichen Eigenschaften. Bei der Gestaltung der Listen mit Powerwerten der Agenten ist ebenso grundsätzlich in zwei Fälle zu unterscheiden. Regelmäßige Verteilungen, also äquidistante Abstände benachbarter Powerwerte sorgen dafür, dass Abweichungen von einer ausgeglichenen Bilanz einfach durch Wahl eines der nächsten Werte ausgeglichen werden. Bei unregelmäßiger Verteilung kann es sein, dass der Wert, der für die Bilanz optimal wäre, in der Liste fehlt. Um die zahlreichen möglichen Fälle hier gut und gleichzeitig handhabbar abzubilden, wurde auch hier lediglich zwischen regelmäßig und unregelmäßig unterschieden. Die Unregelmäßigkeit war dabei manuell zu definieren.

Beispielsweise die Liste [1, 2, 8, 9] wäre solch eine unregelmäßige Verteilung, da der Bereich von 3 bis 7 nicht abgedeckt wird. Die Anzahl der Werte, die ein flexibler Agent nach dem besten Beitrag zur Lösung durchsuchen muss, wurde auf entweder 10 oder 100 festgesetzt. Dies entspricht einer linearen Suche, die einmal mit relativ wenig und einmal mit deutlich mehr Schritten auskommen sollte.

### Sonstiges

Bezüglich Priorisierung war in dieser Testphase zu testen, ob generell priorisierte Agenten in der Optimierung bevorzugt werden. Dies sollte eine korrekte Optimierungsfunktion verifizieren, die dies berücksichtigt. Daher wurde auch hier nur zwischen Szenarien unterschieden, in denen priorisierte Agenten existent waren und solche ohne. Die Gewichte der Zielfunktion wurden in drei Äquivalenzklassen eingeteilt, wobei jeweils eines der drei Ziele dominiert. Diese Klassen wurden abgedeckt, in jedem erstellte Szenario mit allen dieser drei Ausprägungen getestet wurden.

### 6.2.3 Ableiten der Testfälle

Im Anschluss an die in Abschnitt 6.2.2 beschriebene Definition der Äquivalenzklassen, galt es nun auf Basis dieser Klassen einzelne Testfälle abzuleiten. Bei 31 Äquivalenzklassen, verteilt auf 14 Parameter mit 2 oder 3 Äquivalenzklassen pro Parameter, wären  $2^{11} \cdot 3^3 = 55296$  Testfälle abzudecken, wenn alle Kombinationen berücksichtigt werden sollen. Ziel der Testfallableitung war es, eine ausreichend große Abdeckung der Kombinationen zu erhalten, dabei jedoch die Anzahl der Testfälle möglichst gering zu halten. Zur Dokumentation der Testfälle wurde eine Tabelle mit allen Äquivalenzklassen erstellt, die im Anhang in Abschnitt D, Abbildung D.3 zu finden ist. Für jeden Testfall wurde in einer Zeile markiert, welche der Äquivalenzklassen er beinhalten sollte. Da es maximal drei Äquivalenzklassen pro Parameter gab, konnte durch drei Testfälle, im Folgenden Basistestfälle genannt, die Verwendung jeder Äquivalenzklasse in mindestens einem Testfall sichergestellt werden. Die Kombinationen der Äquivalenzklassen in einem Testfall wurden dabei jeweils bewusst gewählt, um verschiedene Extremsituationen zu testen.

So wurde zum Beispiel in Basistestfall 1 der Fokus auf komplizierte Verhandlungen gelegt, indem ein Sprecher das ganze Netz aufbaut, alle Agenten für einen Planungshorizont von 50 Zeitschritten optimieren müssen und es viele flexible Agenten im Netz gibt, die jeweils 100 Leistungswerte zur Auswahl haben. Basistestfall 2 dagegen konzentriert sich eher auf die topologischen Gegebenheiten. Hier wurde eine große Anzahl an Agenten mit vielen Mittelspannungsringen, vielen Shortcuts und Verzweigungen kombiniert, um eine möglichst komplexe Netztopologie zu erreichen. Anschließend wurden weitere Testfälle entwickelt, die besonders die noch fehlenden Kombinationen an Parametern abdecken sollten. Da im Vorfeld nicht ermittelt

wurde, welche Parameter voneinander abhängig sind, wurden hier systematisch alle Kombinationsmöglichkeiten betrachtet. Es wurden zunächst einige Vereinfachungen getroffen, um die Anzahl an Äquivalenzklassen zu reduzieren.

Zum einen wurde der Eingabeparameter „Gewichtung der Zielfunktion“ ausgelassen. Da die Anpassung der Gewichtung im System vorgenommen wird und nicht bei der Erstellung eines Szenarios berücksichtigt werden muss, wurde festgelegt, dass jeder Testfall mit allen drei Gewichtungen durchgeführt werden soll. Zum anderen bekamen mehrere Eingabeparameter eine Standardäquivalenzklasse zugewiesen, die außerhalb der drei Basistestfälle immer gewählt wurde.

- Anzahl der Agenten (1-40): Da es sowohl die Erstellung der Testszenarien, als auch die Auswertung des Tests wesentlich vereinfacht, wenn die Anzahl der Agenten überschaubar bleibt, wurde hier die kleinere Äquivalenzklasse als Standard gewählt.
- Anteil der Sprecher von flexiblen Agenten (50%): Bei einem Netz mit nur einem Sprecher handelt es sich um einen Sonderfall, der in der Regel nicht auftreten sollte. Ziel des Systems ist der Aufbau des Netzes durch die Bildung mehrerer Inseln, die sich nach und nach zusammenschließen.
- Planungshorizont (2): Der Planungshorizont ist nur innerhalb einer Verhandlung von Bedeutung und beeinflusst auch hier nur den Aufwand eines Optimierungsschrittes. Da ein Planungshorizont von 50 die Erstellung der Testszenarien deutlich aufwendiger macht, wurde 2 als Standard gewählt.
- Anzahl der Powerwerte der flexiblen Agenten (10): Hier gilt das gleiche wie für den Planungshorizont, eine größere Anzahl an Werten steigert nur die Komplexität innerhalb einer Verhandlung, hat aber keinen Einfluss auf den restlichen Ablauf des Systems. Um die Erstellung der Testszenarien zu vereinfachen, wurde 10 als Standard gewählt.
- Anzahl der Shortcuts S-Netz (10-60%): Ein Mittelspannungsring ohne Shortcuts wurde hier als Sonderfall betrachtet, weshalb als Standard 10-60% gewählt wurde.

Durch diese Vereinfachungen konnte die Anzahl der Äquivalenzklassen auf 23 reduziert werden, wodurch sich für eine vollständige Kombinationsabdeckung 576 Testfälle ergeben. Um diese Anzahl noch weiter zu verringern, wurden nur die Kombinationen aus jeweils 2 Parametern betrachtet. Dafür wurde eine Matrix erstellt und zunächst alle Kombinationen gekennzeichnet, die bereits durch die drei Basistestfälle abgedeckt wurden. In [Abbildung 6.1](#) ist zu erkennen, dass bereits diese drei

---

Testfälle einen Großteil der Kombinationen beinhalten. Anschließend wurden der Reihe nach weitere Testfälle erstellt, mit dem Ziel, jeweils möglichst viele der noch nicht verwendeten Kombinationen in einem Testfall zu integrieren. Dabei reichten 10 Testfälle aus, um alle noch fehlenden Kombinationen zu berücksichtigen.

	Anzahl Agenten klein (1-30)	Anteil Sprecher 50%	Anteil der flexiblen E&V			Anteil MS-Netz		K-shortcuts 0	Planungshorizont 2-10	Abstände Powerwerte		Anzahl Powerwerte 10	Shortcuts 10-60%	Verzweigung		Anzahl der MS-Ringe 1	Anzahl der Shortcuts K-Netz			Priorität nein ja	
			1-15%	16-50%	51-85%	1-25%	26-50%			regelmäßig	unregelmäßig			ja	nein		weniger	gleich	mehr		
Anzahl Agenten	klein(1-30)																				
Anteil Sprecher	50%																				
Anteil der flexiblen E/V	1-15%																				
	16-50%																				
	51-85%																				
Anteil Agenten MS-Netz	1-25%																				
	26-50%																				
K-shortcuts	0																				
Planungshorizont	>0																				
Abstände Powerwerte	regelmäßig																				
	unregelmäßig																				
Anzahl Powerwerte	10																				
	Shortcuts																				
Verzweigung	ja																				
	nein																				
Anzahl der MS-Ringe	1																				
	>1																				
Anzahl der Shortcuts K-Netz	weniger als S-Netz																				
	gleich wie S-Netz																				
	mehr als S-Netz																				
Priorität	nein																				
	ja																				

Abbildung 6.1: Kombinationsmatrix der Basisnetze

### 6.2.4 Nicht-lösbare Szenarien

Die vorgestellten Szenarien sind alle so entworfen worden, dass sie lösbar sind, was bedeutet, dass alle Agenten am Ende in einer Insel sind. Jedoch definierten die Anforderungen, wie in Abschnitt 3.3.2 dargestellt, auch, dass das System ebenso terminieren muss, wenn eine Lösung mit allen Agenten nicht existiert. Dann sollte das System so lange arbeiten, wie sich vorhandene Inseln bei ausgeglichener Bilanz noch erweitern lassen. Um die Erfüllung dieser Anforderungen zu testen, wurden hierfür eigene Testfälle definiert, die in diesem Abschnitt vorgestellt werden.

Zunächst musste erörtert werden, in welchen Fällen ein Szenario nicht lösbar sein konnte. Eine Möglichkeit war der Fall, dass Agenten aufgrund von ihren Leistungswerten nicht mehr in Inseln aufgenommen werden können. Dies kommt vor, wenn es keine Konfiguration von Powerwerten innerhalb der Inseln gibt, die einen benachbarten Agenten ausgleichen können.

Zudem kann ein Netz nicht vollständig lösbar sein, wenn ein Teil von ihm nicht erreichbar ist. Um mit einem Agenten verhandeln und ihn zuschalten zu können, muss zu ihm sowohl eine Verbindung via Strom- und Kommunikationskante existieren. Ein Sonderfall ist hierbei, dass zu Beginn der Optimierung ein Bereich des Netzes mittels Kommunikationskanten nicht erreichbar ist, dieser aber zu einem späteren Zeitpunkt nach einer Zuschaltung erreichbar ist. In diesem Fall ist das Netz lösbar und zählt wird in diesem Abschnitt nicht behandelt. Mit diesen Erkenntnissen wurden die in Tabelle 6.3 beschriebenen Testfälle erstellt.

ID	Beschreibung	Basis-Testfall
14	Von einem Niederspannungsstrang aus gibt es keine Kommunikationskanten zum restlichen Graphen, daher ist dieser nicht zuschaltbar.	1
15	Von einem Niederspannungsstrang aus gibt es keine Stromkanten zum restlichen Graphen, daher ist dieser nicht zuschaltbar.	3
16	Ein Agent hat eine Leistung -1000, dies macht eine Aufnahme in eine Insel unmöglich.	1

Tabelle 6.3: Definition der Testfälle nicht-lösbarer Szenarien

Bei Testfall 16 ist zu erwarten, dass der Agent, der die Leistung von -1000 hat, in keine Insel aufgenommen wird, das System trotzdem in einer Insel terminiert, in der alle Agenten des Szenarios außer diesem sind. Analog verhält es sich bei den Testfällen 14 und 15, nur dass hier ein ganzer Niederspannungsstrang nicht mit aufgenommen werden kann, während das System terminiert. Erreichbarkeit im Netz setzt Verbindungen mittels Strom- und Kommunikationskanten voraus. Daraus resultiert, dass die Nicht-Erreichbarkeit durch das Fehlen von mindestens einer dieser Kantenart

bedingt ist. Daher wurden zwei Testfälle definiert, wo jeweils eine Kantenart zur Verbindung fehlt. Der Zusammenhang zwischen den hier aufgeführten Testfällen und den Basis-Testfälle, die in Tabelle 6.3 erwähnt werden, wird in Abschnitt 6.2.5 verdeutlicht

### 6.2.5 Erstellung der Testszenarien

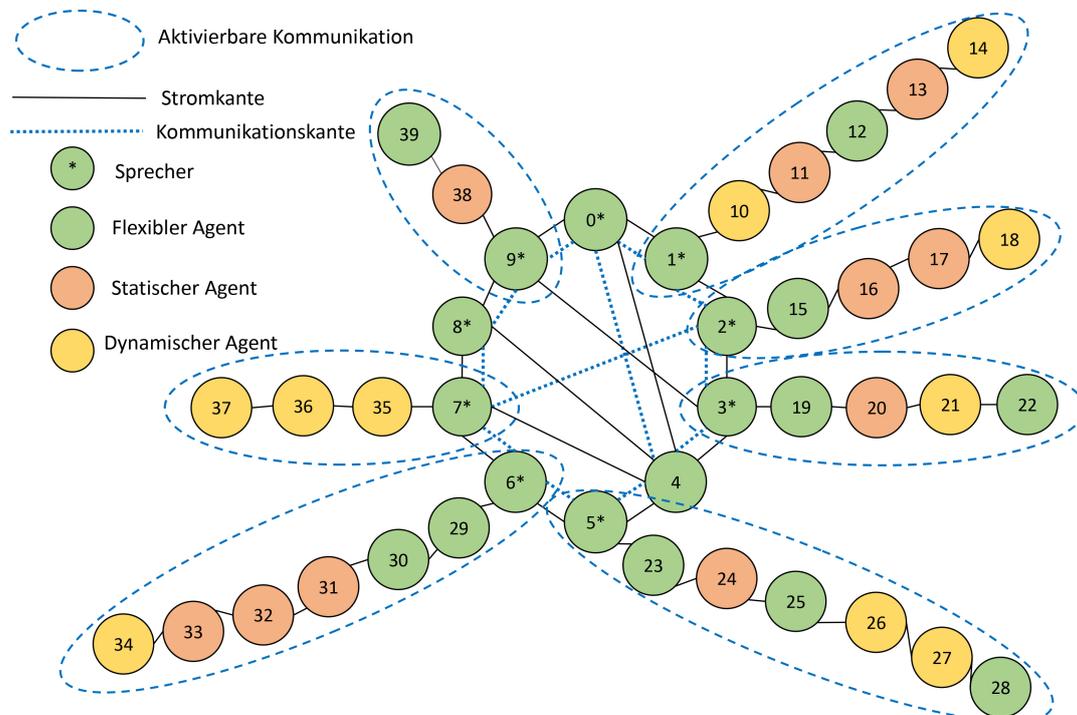


Abbildung 6.2: Graphische Darstellung von Testfall 10

Nach der Definition der Testfälle, die in den Abschnitten 6.2.3 und 6.2.4 beschrieben wurde, mussten als letzter Vorbereitungsschritt die Testszenarien erstellt werden. Dies umfasste einerseits die graphische Darstellung des jeweiligen Szenarios, sowie die Erstellung der dazugehörigen CSV-Dateien, die anschließend bei der Durchführung der Testphase in das System eingelesen wurden. Durch die Äquivalenzklassen wurden viele Designentscheidungen beim Netzentwurf vorgegeben, wie zum Beispiel die Anteile der Agenten im Mittelspannungs- bzw. Niederspannungsnetz. Dennoch wurden nicht alle Parameter zur Beschreibung eines konkreten Netzes durch die Äquivalenzklassen abgedeckt. Einige dieser fehlenden Parameter, wie die Positionierung der Sprecher, wurden durch die Vorgabe allgemeiner Designkriterien festgelegt, die in späteren Absätzen dieses Abschnitts vorgestellt werden. An anderen Stellen, wie zum Beispiel bei der Wahl der konkreten Länge der einzelnen NS-Stränge, mussten individuelle Entscheidungen getroffen werden. Generelles Ziel dabei war, alle Szenarien, in dem durch die Äquivalenzklassen vorgegebenen Rahmen, möglichst unterschiedlich zu gestalten, um dadurch viele verschiedene Situationen zu provozieren und die Testabdeckung zu erhöhen. Ergebnis dieses Arbeitsschrittes sollten

einerseits die drei CSV-Dateien zum Einlesen des Szenarios in das System und andererseits eine graphische Darstellung des jeweiligen Netzes sein. Auf den Abbildungen 6.2 und 6.3 sind zwei der Testfallnetze zu sehen, die unterschiedliche Äquivalenzklassen abdecken. So haben sie zum Beispiel eine unterschiedliche Anzahl an Mittelspannungsringen, eine unterschiedliche Anzahl an Shortcuts und in 6.3 sind Verzweigungen in den NS-Strängen zu sehen, die bei 6.2 nicht existieren.

Im Folgenden werden nun kurz die allgemeinen Designkriterien, sowie die individuellen Designentscheidungen beschrieben, die bei der Netzgestaltung, neben der Verwendung der korrekten Äquivalenzklassen, berücksichtigt werden sollten.

### **Agenten**

Es wurde vorausgesetzt, dass immer genauso viele statische wie dynamische Agenten im Netz vorhanden sind, sodass die Mengenanteile auf Basis der durch die Äquivalenzklassen vorgegebenen Anteil der flexiblen Agenten berechnet werden konnte. Da der Anteil an Sprechern innerhalb der flexiblen Agenten vorgegeben war und Sprecher zwingend Erzeuger sein mussten, wurden bereits Vorgaben zur Verteilung der Erzeuger und Verbraucher getroffen. Für dynamische Agenten gab es keinerlei Vorgaben, sodass hier frei gewählt werden konnte, während statische Agenten nur Verbraucher sein konnten. Grundsätzlich sollte bei der Verteilung der Leistungen darauf geachtet werden, dass das Netz mehr Verbraucher als Erzeuger enthält, da dies eher der Realität entspricht. Die Leistungswerte konnten frei gewählt werden, hier gab es durch die Äquivalenzklassen nur die Vorgabe, ob es sich um regelmäßige oder unregelmäßige Werte handeln soll. Zusätzlich galt die Grundlage, dass das Netz lösbar sein sollte, sofern es sich nicht um einen explizit unlösbaren Testfall handelte. Die Leistungswerte mussten also so gewählt werden, dass sie sich gegenseitig ausgleichen konnten.

### **Netztopologie**

Nicht vorgegeben war die genaue Verteilung der unterschiedlichen Agententypen im Netz und blieb damit den jeweiligen Netzgestaltern überlassen. Die einzige Vorgabe bestand darin, dass Sprecher flexible Erzeuger sein mussten, die sich im Mittelspannungsring befinden, sodass je nach Sprecheranzahl ein gewisser Prozentsatz des Mittelspannungsringes aus flexiblen Agenten bestand. Es wurde festgelegt, dass zwei Mittelspannungsringe immer durch mindestens einen Niederspannungsagenten verbunden sein mussten. Hinsichtlich der Kommunikation war vorgegeben, dass alle Niederspannungsstränge nur durch aktivierbare Kommunikation erreicht werden können, wobei bei der entsprechenden Äquivalenzklasse auch zwei NS-Stränge durch

einen Agenten aktiviert werden konnten. Länge, Anzahl und Positionierung der NS-Stränge konnte dagegen frei gewählt werden, solange der Anteil der Agenten innerhalb dieser Stränge der für den jeweiligen Testfall ausgewählten Äquivalenzklasse entsprach.

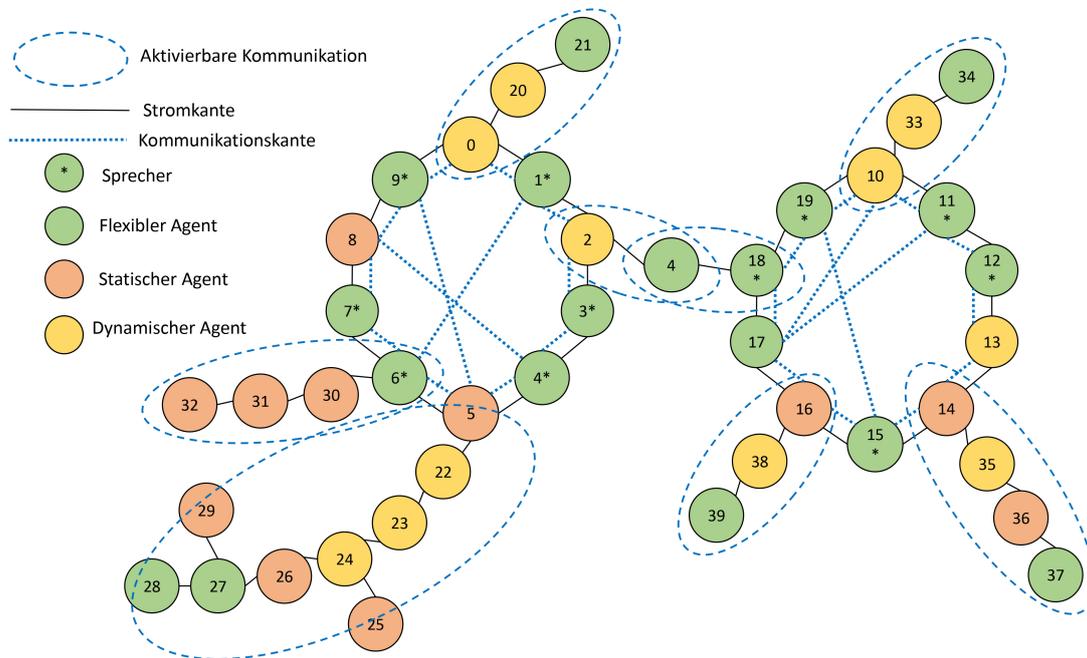


Abbildung 6.3: Graphische Darstellung von Testfall 8

## 6.2.6 Mapping von Testfällen und Anforderungen

Diese Testphase sollte die Erfüllung der Anforderungen, die in Abschnitt 3.3.2 definiert wurden, an das Systems verifizieren. Dies bezieht sich sowohl auf die funktionalen als auch die nicht-funktionalen Anforderungen.

Die funktionalen Anforderungen an das Szenario wurden direkt beim Erstellen der CSV-Dateien für die Netze berücksichtigt. Es werden ungerichtete Graphen mit Knotenagenten, Kommunikations- und Stromnetz erstellt.

Die Anforderungen bezüglich des Systems erfüllte dabei das Java-Programm. Es erfüllt die drei System-Anforderungen. Das Mutliagentensystem bildet die Infrastruktur ab (S-01), die Protokollierung erfolgt mittels Konsolenausgaben (S-02) und Graphstream sorgt zudem für eine grafische Darstellung (S-03).

Demnach bezog sich die Testphase insbesondere auf die Anforderungen an das Verfahren sowie einige nicht-funktionale Anforderungen. Dabei werden in allen erstellten Szenarien jeweils alle Verfahrens-Anforderungen abgedeckt. Grund hierfür ist, dass das Verfahren durch die Gesamtheit dieser Anforderungen definiert wird. Dieses Verfahren wurde hier in der Testphase nicht in einzelnen Komponenten getestet, sodass

immer alle Anforderungen von Bedeutung waren. Durch das Testen mit verschiedenen Gewichtungen der Zielfunktion wurden die definierten Optimierungsziele mit verschieden starkem Einfluss berücksichtigt.

Des Weiteren ist eine nicht-funktionale Anforderung, dass das System in allen lösba- ren und nicht lösba- ren Szenarien terminieren muss. Das bedeutet, dass das Verfahren bei vollständig lösba- ren Netzen alle Agenten in eine Insel aufnehmen muss. Ebenso bedeutet dies aber, dass es enden muss, wenn nicht alle aufgenommen werden kön- nen. Aus diesem Grund wurden die in Abschnitt 6.2.4 definierten Testfälle erstellt und getestet.

## 6.3 Durchführung

Nachdem auf Basis der identifizierten Eingabeparameter, den daraus abgeleiteten Äquivalenzklassen und der erstellten Testfälle die Testszenarien entworfen wurden, konnten diese im System getestet werden.

In der Dokumentation der Tests wurden folgende Informationen festgehalten:

- ID des Testfalls zur eindeutigen Identifikation
- Das Eingabeszenario in Form der Namen der erstellten CSV-Dateien
- Die im System eingestellte Gewichtung der Zielfunktion
- Das Testergebnis in Form der Konsolenausgaben
- Eventuell aufgetretene Fehler
- Die Ursache der aufgetretenen Fehler
- Der Umfang der Fehlerbehebung
- Das Testergebnis in Form der Konsolenausgaben nach der Fehlerkorrektur

Das erwartete Testergebnis war stets, dass das System terminiert und für lösba- re Szenarien dabei eine ausgeglichene Insel bildet. Im Falle der nicht-lösba- ren Szenarien, welche in Abschnitt 6.2.4 beschrieben werden, war die Erwartungshaltung, dass das System so lange weiterarbeiten soll, wie sich vorhandene Inseln bei ausgeglichener Bilanz noch erweitern lassen. Die erstellten Testszenarien wurden in koordinierter Vorgehensweise ins System eingespielt und die Ergebnisse dokumentiert. Dabei soll- ten potenziell festgestellte Fehler dokumentiert und deren Auswirkungen und Mög- lichkeiten zur Behebung analysiert werden. Nach der Korrektur eines Fehlers war dabei ein vollständiger Re-Test der Testszenarien vorgesehen. Die Zusammenfassung der Ergebnisse und deren Interpretation sind im Abschnitt 6.4.1 zu finden.

## 6.4 Auswertung

Im Folgenden sollen die Ergebnisse der Testphase kurz vorgestellt und interpretiert werden, sowie ein Fazit zur Testphase gezogen werden.

### 6.4.1 Ergebnisse und Interpretation

Größtenteils ließen sich die Testszenarien fehlerfrei ausführen, lieferten die erwarteten Ergebnisse und galten damit als bestanden. Das erwartete Ergebnis entsprach jeweils einem vollständig aufgebauten Netz mit der Bilanz 0, bzw. im Falle der nicht lösbaren Szenarien einem teilweise aufgebauten Netz mit ausgeglichenen Inseln.

Bei der Ausführung von Testfall 3 konnte ein Fehler im System ermittelt werden. Es kam zu einer vorzeitigen Terminierung des Systems, bedingt durch eine ungünstige Vorgehensweise der Terminierungserkennung. In dem entsprechenden Szenario beginnen zwei Sprecher mit dem Aufbau zweier Inseln. Insel 1 beendet zweimal eine Verhandlung mit den gleichen Agenten auf der Blacklist und meldet daher dem Konvergenzagenten, dass sie fertig ist, entsprechend dem Vorgehen bei Stagnation aus Abschnitt 4.3.8. Anschließend fährt sie fort mit dem Zyklus der Nachbar- und Inselnsuche. Zu diesem Zeitpunkt gibt es also zwei aktive Inseln im System, von denen eine fertig ist. Insel 0 hat unterdessen die Inselnachbarschaft von Insel 1 erreicht und befindet sich im WAIT-Zustand, wo sie von Insel 1 eine Anfrage bekommt. Der Sprecher von Insel 0 schickt eine Agree-Nachricht an Insel 1 mit seinem eigenen Inselobjekt und gibt seine Sprecherrolle auf, womit er aus den aktiven Inseln gelöscht wird. Im nächsten Schritt stellt das System also fest, dass die Anzahl der aktiven Inseln der Anzahl der fertigen Inseln entspricht und terminiert dadurch, bevor die Inselvereinigung durchgeführt werden kann. Dieser Fehler trat nur bei diesem Testfall und auch nur bei einer bestimmten Gewichtung auf, da eine andere Gewichtung zu einer anderen Reihenfolge des Netzaufbaus führte, wodurch die Inselvereinigung von Insel 0 ausging und nicht von Insel 1. Ein möglicher Lösungsansatz wäre, dass das Aufgeben der Sprecherrolle unabhängig von der Löschung aus den aktiven Inseln stattfindet, sodass Insel 0 sich erst dann aus den aktiven Inseln entfernt, wenn die Inselvereinigung durchgeführt wurde. Andersherum müsste Insel 1 sich in diesem Zusammenhang auch als „nicht fertig“ bei dem Konvergenzagenten melden, sobald sie eine positive Rückmeldung von Insel 0 erhalten hat. Eine weitere Möglichkeit wäre, dass eine als fertig gemeldete Insel nicht mehr agiert, sondern in WAIT verbleibt und nur noch von außen angesprochen werden kann. Dies könnte allerdings auch nachteilig wirken, sodass das System mit mehreren passiven Inseln terminiert, obwohl es noch möglich wäre weitere Agenten aufzunehmen. Aus Zeitgründen und in Anbetracht der Tatsache, dass dieser Fehler nur bei der Kombinationen mehrerer ungünstiger Umstände auftrat und sich rein auf die Terminierung des Systems und nicht auf den Algorithmus bezog, wurde keine Fehlerkorrektur vorgenommen.

Auffallend war, dass nur zwei der Testfälle unterschiedliche Ergebnisse bei unterschiedlicher Gewichtung zeigten. Um das Verhalten der Zielfunktion in möglichst unterschiedlichen Situationen testen zu können, und dementsprechend die Chance für die Aufdeckung von Fehlern zu erhöhen, müssten die Testfälle explizit darauf ausgelegt werden.

### 6.4.2 Fazit

Die meiste Arbeit der Fehlersuche und -behebung fand bereits im Vorfeld der Testphase statt. Erweiterungen des Algorithmus wurden jeweils direkt im Anschluss an die Implementierung getestet. Hierzu wurden minimale Testszenarien verwendet, die explizit im Hinblick auf das Testen der Erweiterungen erstellt wurden. Dabei wurde versucht, möglichst viele Sonderfälle zu berücksichtigen. Dadurch hatte das System bereits eine gute Grundstabilität erreicht, die auch der Kombination aller Funktionalitäten und größeren Testszenarien in verschiedenen Ausprägungen standhielt.

Bei den Eingabeparametern für die Erstellung der Testszenarien wurden die möglichen Leistungen der Agenten nur im Hinblick auf Anzahl und Regelmäßigkeit der Werte bei den flexiblen Erzeugern/Verbrauchern betrachtet. Nicht berücksichtigt wurde die Verteilung der statischen/dynamischen/flexiblen Agenten im Netz sowie die Verteilung der Werte, dies konnte beim Design der Szenarien frei entschieden werden. Um die Erstellung der Szenarien möglichst einfach zu halten, wurden dabei oft die gleichen oder sehr ähnliche Werte für alle Agenten gewählt, entsprechend ob es sich um einen statischen, dynamischen oder flexiblen handelt. Dies führt zu einer deutlich vereinfachten Ausgangslage für den Netzaufbau, die so in der Realität vermutlich nicht gegeben wäre.

Die Tests galten als bestanden, wenn das erwartete Endergebnis erreicht wurde. Zwar wurde auch der Verlauf des Systems stichprobenartig überprüft, eine exakte Überprüfung konnte jedoch aufgrund der Komplexität nicht sichergestellt werden. Ebenso wurde nicht überprüft, ob das System den jeweils „optimalen“ Weg wählt, wobei die fehlende Variation bei unterschiedlicher Gewichtung darauf hindeutet, dass es bei den meisten Testszenarien ohnehin nur einen möglichen Weg gab.

Das fehlerfreie Durchlaufen fast aller Tests lässt zwar nicht auf ein fehlerfreies System schließen, doch es konnte gezeigt werden, dass das System bei allen Test stabil lief. Das Hauptziel der Tests, die Bestätigung der Anforderungen, konnte ebenfalls erreicht werden, wie in Abschnitt [6.2.6](#) gezeigt wurde.

# 7. Validierung

Das Kapitel der Validierung beschreibt die Untersuchung der Güte des entwickelten Algorithmus. Dazu wird zuerst der Aufbau dieser Untersuchung beschrieben. Anschließend folgt die Darstellung der erhobenen Ergebnisse und deren Interpretation, bevor das Kapitel mit einem Fazit und einem Ausblick abgeschlossen wird.

## 7.1 Aufbau

Der Aufbau der Validierung beschreibt die zu untersuchenden Validierungskriterien sowie die relevanten Eingabeparameter. Außerdem werden in diesem Abschnitt Konstruktionsvorschriften für die Erstellung der Validierungs-Szenarien erläutert.

### 7.1.1 Validierungskriterien

Zu Beginn der Validierungsphase wurden die Kriterien ermittelt, auf deren Grundlage die Bewertung des Systems erfolgen sollte. Basis hierfür bildete das Paper „Evaluation Guidelines for Asynchronous Distributed Heuristics in Smart Grid Applications“ [HiSo14]. In diesem Paper werden mehrere Kriterien für die Validierung verteilter Systeme vorgestellt. Dabei wird unterschieden zwischen Zeroth-, First- und Higher-Order Criteria. Zeroth-Order Kriterien können entweder erfüllt werden oder nicht und sind unabhängig von den getesteten Szenarien. First-Order Kriterien werden auf Basis der Ergebnisse von Experimenten mit definierten Szenarien bewertet. Higher-Order Kriterien zeigen quantifizierbare Effekte mit Vektoren oder Matrizen als Ausgabeparameter und erfordern eine Reihe von Experimenten mit variierenden Szenarien.

#### 1. Zeroth-Order

- (a) Korrektheit: Das System liefert eine gültige Lösung für das gegebene Problem.
- (b) Terminierung: Das System terminiert in einer endlichen Zeitspanne.
- (c) Selbststabilisierung: Das System kann sich selbstständig in gültige Startzustände bewegen.

## 2. First-Order

- (a) Performanz: Beschreibt die Güte der gefundenen Lösung des Systems, zum Beispiel durch einen Fitnesswert.
- (b) Effizienz: Benötigte Ressourcen des Systems, zum Beispiel gemessen durch Laufzeit, Zielfunktionsaufrufe oder Anzahl der versendeten Nachrichten.

## 3. Higher-Order

- (a) Skalierbarkeit: Das Verhalten des Systems bei verschiedenen Skalierungen der Eingabeparameter.
- (b) Robustheit: Wie reagiert das System auf Störungen, zum Beispiel verzögerte Nachrichten.

Nach der Definition der Kriterien beschreibt das Paper zwei Ansätze für konkrete Arten der Evaluation, die analytische und die empirische Methode. Bei der analytischen Methode wird das Verfahren mathematisch formal untersucht, während die empirische auf der Ausführung des Algorithmus und den dadurch erhobenen Daten basiert. Wichtig dabei ist die Wahl eines sinnvollen „Design of Experiments“, was die Wahl der Eingabeparameter, Definition der Szenarien und Anzahl der Ausführungen umfasst.

Die Projektgruppe entschied sich für einen empirischen Ansatz zur Validierung, da ein analytischer für die der Projektgruppe zur Verfügung stehenden Zeit nicht umsetzbar gewesen wäre. Dafür wurden Szenarien festgelegt, basierend auf denen in Abschnitt 7.1.2 beschriebenen Eingabeparametern sowie dem in Abschnitt 7.1.3 beschriebenen Standard-Mittelspannungsring. Anschließend wurden die oben genannten Kriterien in Bezug auf das erstellte System betrachtet und bewertet, inwieweit sie für die Validierung des Systems eine Rolle spielen und welche Ausgabeparameter dafür erfasst werden müssen. Die Ergebnisse dieser Überlegungen werden in den folgenden Abschnitten vorgestellt.

### 7.1.1.1 Korrektheit

Korrektheit bedeutet im Falle des hier beschriebenen Systems einerseits, dass sich ausgeglichene Inseln bilden, also das System korrekt arbeitet. Andererseits umfasst Korrektheit aber auch, dass, wenn es eine Lösung gibt, bei der sich alle Agenten in einer Insel befinden, diese auch gefunden werden muss. Das korrekte Arbeiten des Systems ist leicht zu beweisen, da durch die in Abschnitt 4.3.4 vorgestellte Death Penalty sichergestellt wurde, dass Agenten, die sich nicht ausgleichen können, sich nicht zu einer Insel zusammenschalten. Es kann also nicht zu falschen Ergebnissen kommen. Das garantierte Finden der „richtigen“ Lösung ist dagegen schwierig formal

zu zeigen und würde noch weitere Anpassungen am System benötigen. Beispielsweise ist es derzeit nicht möglich, einmal entstandene Inseln wieder aufzulösen, also Schritte rückgängig zu machen, wodurch das System unter bestimmten Umständen in eine Sackgasse laufen kann. Es kann also davon ausgegangen werden, dass es mindestens ein lösbares Netz gibt, das vom System nicht gelöst werden könnte. Daher wurde in Absprache mit den Betreuern entschieden, dieses Kriterium in der Validierungsphase nicht weiter zu berücksichtigen und das korrekte Arbeiten des Systems durch die Testphase als ausreichend bewiesen zu betrachten.

### 7.1.1.2 Terminierung

Die Terminierung des Systems kann in zwei voneinander unabhängige Abschnitte unterteilt werden. Einerseits die Terminierung des gesamten Systems und andererseits die Terminierung - bzw. Konvergenz - innerhalb der im System stattfindenden Verhandlungen. Da die Verhandlungen auf COHDA basieren und hierfür bereits der Beweis für die Terminierung erbracht wurde (s.[Hinr14, 5]), ist es sinnvoll, diesen Teil des Systems losgelöst vom Rest zu betrachten.

Der Beweis für die Terminierung von COHDA basiert auf den folgenden drei Stufen:

1. Stufe: Nach einer endlichen Zeitspanne sind die wahrgenommenen Systemzustände und Lösungskandidaten aller Verbundteilnehmer vollständig, das heißt sowohl im wahrgenommenen Systemzustand, als auch im Lösungskandidaten jedes Agenten befindet sich jeder andere Agent mit einer ausgewählten Leistung. Ein wesentlicher Teil von COHDA ist die Übernahme von neuen Informationen. Sobald ein Agent neue Informationen erhält, werden diese übernommen und an die Nachbarn weiterverbreitet. Über das Kommunikationsnetz ist jeder Agent über eine endliche Anzahl von Kanten mit jedem anderen Agent verbunden, sodass Informationen nur eine endliche Anzahl von Schritten brauchen, um einen Agenten zu erreichen. Dadurch ist sichergestellt, dass in einer endlichen Zeitspanne jeder Agent von jedem anderen Agenten eine veröffentlichte Leistung erhalten hat.

2. Stufe: Nach einer endlichen Zeitspanne wird von einem Verbundteilnehmer eine finale Lösung ermittelt. Es gibt eine endliche Anzahl an Lösungen und dementsprechend auch eine endliche Anzahl an finalen Lösungen. Eine finale Lösung entspricht einem lokalen Optimum und bedeutet, dass kein Verbundteilnehmer durch Auswahl einer anderen Leistung etwas an dieser Lösung verbessern könnte. Betrachtet man die Summe der Güte aller Lösungskandidaten der Verbundteilnehmer in einem Zeitschritt stellt man fest, dass diese nur gleich bleiben oder sich verbessern kann. Dies ergibt sich aus der Tatsache, dass ein Agent nach einer Optimierung niemals einen schlechteren Lösungskandidat übernimmt. Es kann außerdem gezeigt werden, dass diese „Gesamtgüte“ nur eine endliche Anzahl an Schritten gleich bleiben kann, bevor

das System terminiert. Die Güte bleibt dann gleich, wenn ein Agent einen Lösungskandidaten erstellt, der genauso gut ist, wie der bisher beste, allerdings eine höhere ID hat. Der Worst Case wäre also, dass jeder Verbundteilnehmer einen Lösungskandidaten mit unterschiedlicher ID aber gleicher Güte hat. In diesem Fall würde es eine endliche Anzahl an Schritten dauern, bis der Lösungskandidat mit der höchsten ID von jedem anderen Agenten erhalten und übernommen wurde. Zusätzlich ist festgelegt, dass wenn ein Agent die Möglichkeit hat, einen Lösungskandidaten zu verbessern, er dies auch innerhalb seiner Optimierungsphase umsetzt. Es werden also so lange bessere Lösungen erzeugt, bis kein Agent eine bessere findet. Daher muss zu einem endlichen Zeitpunkt ein Agent einen nicht verbesserbaren Lösungskandidaten finden.

3. Stufe: Nach einer endlichen Zeitspanne hat sich diese finale Lösung im gesamten Verbund ausgebreitet und es kommt zur Terminierung. Wie bei der ersten Stufe werden auch hier wieder neue Informationen über das Kommunikationsnetz verbreitet. Nach einer endlichen Zeitspanne erhält also jeder Agent die finale Lösung. Erhält er sie zum ersten Mal, wird er sie an seine Nachbarn weiterschicken, die sie ihm daraufhin, sofern sie sie ebenfalls zum ersten Mal erhalten, wieder zurückschicken. In diesem Fall stellt der Agent fest, dass es keine Änderungen mehr gab, sodass er seinerseits keine neue Nachricht verschickt. Somit terminiert das Verfahren, indem nach einer endlichen Zeitspanne kein Agent mehr neue Informationen verschickt.

Das im System genutzte Verfahren für die Verhandlungen unterscheidet sich von COHDA hauptsächlich in zwei Punkten. Zum einen wird grundsätzlich zunächst auf die Bilanz 0 hin optimiert und die Zielfunktion kommt erst in einem zweiten Schritt bei gleichwertigen Lösungen zum Einsatz. Zum anderen haben Agenten durch die Blacklist die Möglichkeit, sich selbst aus dem Lösungskandidaten auszuschließen. Dabei stehen sie jedoch weiterhin im Lösungskandidaten drin und nehmen auch weiterhin an der Verhandlung teil. Beide Punkte haben keinen größeren Einfluss auf die Funktionalität von COHDA und auf die oben beschriebene Beweisführung der Terminierung, sodass diese für das System übernommen werden kann.

Für die Terminierung des gesamten Systems, müsste man die in Abschnitt 4.3.8 beschriebenen Ablaufschritte betrachten und prüfen inwieweit garantiert werden kann, dass Endlosschleifen erkannt und abgefangen werden. Die Terminierung nach Zusammenschluss aller Agenten zu einer Insel ist dabei vergleichsweise leicht zu zeigen. Der letzte Schritt vor einem vollständigen Netzaufbau ist entweder eine Verhandlung oder eine Inselvereinigung. In beiden Fällen wäre der nächste Schritt des Sprechers seine neuen Inselnachbarn zu fragen, ob diese bereit sind für eine Verhandlung. Da er in diesem Fall keine Inselnachbarn mehr hat, würde er keine Nachrichten mehr verschicken. Normale Agenten reagieren nur auf Anfragen von außen, verschicken

jedoch nicht eigenständig Nachrichten. Die Terminierung kann also dadurch erkannt werden, dass im gesamten System keine Nachrichten mehr gesendet werden. Schwieriger zu zeigen ist dagegen die Terminierung eines nicht lösbaeren Netzes, da hier einzelne Inseln noch aktiv sein können. Hier wurde in Absprache mit den Betreuern festgelegt, dass der Beweis der Terminierung ausreichend durch die Testphase abgedeckt wurde.

### 7.1.1.3 Selbststabilisierung

Selbststabilisierung wurde in dem System nicht näher betrachtet, da vorausgesetzt wurde, dass sich das System nach einem Blackout in einem definierten Startzustand befindet. Alle Schalter sind geöffnet, es wird nirgendwo Strom erzeugt oder verbraucht, alle Agenten befinden sich im Zustand SLEEP und es existiert eine minimale Kommunikation, sodass mindestens ein Sprecher aufwachen und mit dem Netzaufbau beginnen kann.

### 7.1.1.4 Performanz

Zur Bewertung der Güte des Systems gibt es mehrere potentielle Ansätze. Eine Möglichkeit wäre eine Aussage darüber, wieviel Prozent des Netzes wieder aufgebaut werden konnte. Hierbei müsste berücksichtigt werden, dass dieser Anteil nicht von einem vollständigen Netz ausgeht, sondern von der bestmöglichen Lösung. Diese müsste ebenfalls genauer definiert werden, da sich die optimalen Lösungen je nach gewünschtem Ergebnis unterscheiden können. Beispielsweise müsste festgelegt werden, ob eine einzelne Insel mit 20 Agenten höher bewertet wird, als 15 Inseln mit je zwei Agenten. Erstere wäre deutlich stabiler, während letztere mehr Netzknoten mit Strom versorgen. Möglich wäre auch ein Fokus auf die frühzeitige Zuschaltung priorisierter Knoten, um sicherzustellen, dass kritische Infrastrukturen bevorzugt mit Strom versorgt werden. Dafür müsste ermittelt werden, wie schnell ein priorisierter Agent in einem Szenario maximal zugeschaltet werden könnte, um die Ergebnisse des Systems entsprechend einordnen zu können. Zusätzlich könnte auch die Anzahl der Verhandlungen betrachtet werden, im Hinblick darauf, dass sie den aufwendigsten Teil des Systems darstellen. Allerdings ist diese Anzahl alleine nicht aussagekräftig, da die Menge der Optimierungen pro Verhandlung stark variieren kann, abhängig von der Anzahl der Agenten, des möglichen Lösungsraums und der Topologie des Overlay-Netzes. Hier müsste auch die Anzahl der versendeten Arbeitsgedächtnisse mit einbezogen werden. Eine weitere relevante Größe in Bezug auf die Verhandlungen ist die Anzahl der eingetretenen Death Penalties bei ausgeschalteter Blacklist, da in diesen Fällen eine Verhandlung quasi umsonst stattgefunden hat. Dieser Fall tritt immer dann ein, wenn in einer Verhandlungsrunde nicht alle neuen Nachbarn hinzugenommen werden können, was stark abhängig von dem Aufbau des Netzes und

der Agenten ist. In Netzen, bei denen dieser Fall häufig auftritt, würde das System also wesentlich schlechter abschneiden als bei einem Netz wo die Inseln von Anfang an genug Kapazität haben, alle Nachbarn aufzunehmen. Da in der Projektgruppe nur mit fiktiven Leistungswerten gearbeitet wurde, kann keine Aussage darüber getroffen werden, inwiefern das Neustarten von Verhandlungen unter realistischeren Bedingungen eine Rolle spielen würde. Ein weiteres Beispiel für Verhandlungen, die zu keiner Inselvergrößerung führen, sind diejenigen, bei denen nach Abschluss alle potentiellen neuen Agenten auf der Blacklist stehen („Everybody-On-Blacklist“).

Die Schwierigkeit bei der Bestimmung dieser Performanzkriterien besteht in der Erstellung der Testszenarien. Diese müssten speziell so gewählt werden, dass es zu mehreren potentiellen Verzweigungen im Systemablauf kommt, um dann bewerten zu können, welchen Weg das System wählt. Dafür müsste jeder Schritt per Hand durchlaufen und nachvollzogen werden, welcher Agent sich zu welchem Zeitpunkt im welchen Zustand befindet, was bei mehr als 10 Agenten zu unzuverlässig werden würde. Zwar wurden die benötigten Daten im Zuge der Validierung erhoben, haben aber nur eine begrenzte Aussagekraft, da die optimalen Ergebnisse nicht bekannt sind und daher die Vergleichswerte fehlen. Die für die Bestimmung der Performanz erhobenen Ausgabekriterien sind:

- Anzahl der Verhandlungen insgesamt
- Anzahl der aufgetretenen Death Penaltys
- Anzahl der Verhandlungen Everybody-On-Blacklist

#### 7.1.1.5 Effizienz

Für die Bestimmung der Effizienz des Systems wurde zum einen die Anzahl der versendeten Nachrichten, aufgeteilt nach Nachrichtentyp, erfasst und zum anderen die Anzahl der benötigten Steps bis zur Terminierung des Systems. Die Schritte werden vom in Abschnitt 5.2.1 beschriebenen MiniMAS verwaltet und sind unabhängig von der eigentlichen Laufzeit, da die Dauer der Ausführung eines Steps von der Anzahl der Nachrichten abhängt, die in diesem Step von den einzelnen Agenten verarbeitet werden müssen. Die Hauptfrage, die sich hier stellte, war die Wahl der Eingabegröße und die Frage danach, welche Parameter welchen Einfluss auf die Kriterien zur Effizienzbewertung haben, bzw. welche Parameter ggf. voneinander abhängig sind. Das dafür genutzte Vorgehen wird in Abschnitt 7.1.2 und 7.1.3 beschrieben.

#### 7.1.1.6 Skalierbarkeit

Um die Skalierbarkeit zu testen, wurden die in Abschnitt 7.1.2 vorgestellten Parameter in unterschiedlichen Skalierungsstufen betrachtet. Von besonderem Interesse war

dabei die Skalierung der Agentenanzahl, um Aussagen darüber treffen zu können, wie das Verhalten des Systems bei großen bis sehr großen Netzen ist.

#### 7.1.1.7 Robustheit

Das Kriterium der Robustheit kann sich auf zwei Bereiche beziehen. Einerseits die Robustheit gegenüber Fehler im Systemablauf, wie zum Beispiel verloren gegangene Nachrichten oder dem plötzlichen Ausfall von Agenten und andererseits die Robustheit des Systems bei nicht lösbaren Netzen. Ersteres wurde bei der Entwicklung des Systems nicht berücksichtigt, da die Fehlerfreiheit der Kommunikation und der Agenten vorausgesetzt wurde. Daher gibt es bisher keine Mechanismen zum Abfangen derartiger Probleme. Im Bezug auf Letzteres wurde das System so konzipiert, dass nicht lösbare Szenarien erkannt werden und das System terminiert. Wie schon bei der Terminierung in Abschnitt 4.3.8 erläutert, ist es jedoch nicht möglich zu garantieren, dass das System bei jedem nicht lösbaren Szenario terminiert.

### 7.1.2 Eingabeparameter

In der Validierung wurden verschiedene Parameter untersucht. Dabei wurden zum einen Eingabeklassen verwendet, die die Skalierbarkeit des Netzes untersuchen, wie die Anzahl der Agenten und die Länge der Niederspannungsstränge. Zum anderen wurden Parameter untersucht, die sich auf die Ausprägung des jeweiligen Netzes auswirken. Dies sind insbesondere die Anzahl der Sprecher, der Anteil flexibler Erzeuger und Verbraucher im Netz, sowie der Anteil von Shortcuts im Mittelspannungsring. Bei allen Parametern wurde untersucht, welchen Einfluss diese auf die Performanz und die Effizienz des Systems haben. Im Anschluss wurde geprüft, wie stark die Zielfunktion und die Priorisierung von Agenten auf den Netzaufbau wirken. Im Folgenden sollen die einzelnen Eingabeparameter kurz erläutert und die Erwartungen an ihre Auswirkungen formuliert werden.

#### Anzahl der Agenten

Die Anzahl der Agenten im Netz wurde auf zwei unterschiedliche Arten variiert. Zum einen wurden an einem, wie in Abschnitt 7.1.3 vorgestellten, Standard-MS-Ring die Niederspannungsstränge in verschiedenen Skalierungen verlängert. Zum anderen wurden Netze entworfen, die aus mehreren aneinander geschlossenen Standard-MS-Ringen bestehen. Dies sind wichtige Eingabeparameter, da der Algorithmus später auch in größeren Netzen einsetzbar sein soll. Deshalb ist es wichtig, die Performanz und Effizienz in Abhängigkeit der Netzskalierung zu betrachten.

In dem für den Algorithmus entworfenen Szenario haben Niederspannungsstränge die Eigenschaft, dass in ihnen, nach der Aktivierung des zugehörigen Agenten im Mittelspannungsring, vollständige Kommunikation herrscht. Im Gegensatz dazu können

aber die Knoten im Niederspannungsnetz nur nacheinander zugeschaltet werden. In Abschnitt 4.3.2 wurde dargestellt, dass auf eine Einschränkung der Kommunikationswege während einer Optimierung über das Overlay-Netz verzichtet wurde. Diese Konstellation lässt vermuten, dass die Länge der Niederspannungsstränge sowohl einen Einfluss auf das Nachrichtenaufkommen, als auch auf die Gesamtzahl der Verhandlungen haben könnte. Das Ergebnis hierzu befindet sich in Abschnitt 7.2.2.1.

Durch die Erhöhung der Anzahl der aneinander geschlossenen Standard-MS-Ringe sollte der Einfluss der Anzahl der Agenten ohne den vermuteten negativen Einfluss der Niederspannungsstränge untersucht werden. Dies sollte dazu dienen ein besseres Bild des Einflusses der Netzgröße auf die Effizienz und Performanz des Algorithmus zu bekommen. Das Ergebnis hierzu befindet sich in Abschnitt 7.2.2.2.

### **Anzahl der Sprecher**

Sprecher initiieren den Wiederaufbau des Netzes. Sie beginnen mit der Inselbildung und führen Inselvereinigungen durch. Ihre Anzahl könnte also einen direkten Einfluss darauf haben, wie viele Verhandlungen geführt werden, wann Inselvereinigungen durchgeführt werden und wie sich die Inseln im Netz ausbreiten. Es wird deshalb erwartet, dass die Anzahl der Sprecher einen signifikanten Einfluss, insbesondere auf die Performanz des Verfahrens, hat. Die Untersuchung hierzu befindet sich in Abschnitt 7.2.2.3.

### **Anzahl flexibler Erzeuger und Verbraucher**

Durch ihre Flexibilität bringen die hier betrachteten Erzeuger und Verbraucher mehr Spielraum in die untersuchten Netze. Es gibt durch sie mehr Wege zu einer gültigen Lösung und einen generell größeren Lösungsraum. Hierdurch wird die Frage untersucht, ob eine Erhöhung der Anzahl flexibler Erzeuger und Verbraucher die Performanz und Effizienz des Verfahrens beeinflusst. Das Ergebnis wird in Abschnitt 7.2.2.4 dargestellt.

### **Anzahl der Shortcuts im Mittelspannungsring**

Die Erhöhung der Anzahl der Shortcuts, hierbei werden Strom- und Kommunikationsnetz gleichgestellt, im Mittelspannungsnetz ermöglicht wiederum mehr Lösungswege beim Wiederaufbau des untersuchten Netzes. In Abschnitt 7.2.2.5 wird untersucht, wie sich die Skalierung der Shortcuts im Mittelspannungsnetz auf die Performanz und Effizienz des Verfahrens auswirkt.

## Gewichtung der Zielfunktion

In der Zielfunktion werden die Inselgröße, die Sichtbarkeit der Insel, sowie die Priorisierung von Agenten in der Insel betrachtet. Hier liegt die Besonderheit darin, dass die Zielfunktion erst zum Tragen kommt, sollten in einer Verhandlung nicht alle Agenten in die Insel integriert werden können. Wie sich die Gewichtung der Zielfunktion in diesen Situationen auswirkt, wird in Abschnitt [7.2.2.6](#) untersucht.

## Priorisierung

Um bestimmte Agenten bevorzugt in eine Insel integrieren zu können, wurde die Priorisierung von Agenten eingeführt. Diese kommt, wie auch die Gewichtung der Zielfunktion, ebenfalls dann zum Tragen, wenn nicht alle Agenten in einer Optimierung in eine Insel integriert werden können. Die Untersuchung der Priorisierung ist in Abschnitt [7.2.2.7](#) zu finden.

## Kombinationen von Eingabeparametern

Im Anschluss an die Untersuchung der oben genannten Eingabeparameter wurden diese zum Teil miteinander kombiniert. So sollte der Einfluss der Parameter auf Effizienz und Performanz in Abhängigkeit voneinander evaluiert werden. Es wurden dabei die Anzahl Agenten und die Länge der Niederspannungsstränge jeweils mit verschiedenen Anteilen an Sprechern im Netz untersucht. Das Ergebnis ist in Abschnitt [7.2.2.8](#) zu finden.

### 7.1.3 Konstruktionsvorschriften

Um die beschriebenen Eingabeparameter möglichst vergleichbar zu validieren, wurden einige Konstruktionsvorschriften zur Netzmodellierung festgelegt und ein standardisierter Mittelspannungsring entworfen. Die wichtigste Anforderung an diesen standardisierten Mittelspannungsring war dabei die Skalierbarkeit.

Die von den Ausprägungen des standardisierten Mittelspannungsringes unabhängigen Konstruktionsvorschriften der Netzmodellierung waren hierbei:

- Die Priorität der Knotenagenten sollte stets 1 betragen
- Es sollten keine Kommunikationsshortcuts zwischen Niederspannungssträngen existieren
- Der Planungshorizont sollte 2 betragen
- Das Netz sollte lösbar sein
- Es sollte keine Verzweigung der Stränge im Niederspannungsnetz existieren

- Die Gewichtung der Zielfunktion sollte mit folgenden Werten ausgeprägt sein:
  - Sichtbarkeit = 0,3
  - Inselgröße = 0,3
  - Priorisierung = 0,4

Im Folgenden werden die Eigenschaften des Standard-Mittelspannungsringes erläutert, welcher in Abbildung 7.1 dargestellt ist.

Der Mittelspannungsring besteht aus 16 Agenten, wovon die Agenten mit den IDs 2, 6, 10 und 14 als Sprecher definiert sind. Die Agenten in dem Mittelspannungsring sind jeweils mit einer Strom- und Kommunikationskante miteinander verbunden. In den angeschlossenen Niederspannungsnetzen befinden sich insgesamt 24 Agenten, welche als aktivierbare Nachbarn zugeschaltet werden können. Die Agenten mit den IDs 0, 4, 8 und 12 besitzen für mögliche Verbindungen zu anderen Mittelspannungsringen keinen angeschlossenen Niederspannungsstrang. Die Verbindungen zu anderen Mittelspannungsringen werden mithilfe von Konnektor-Knoten realisiert. Dies sind Agenten, welche zwischen zwei Mittelspannungsringen platziert werden und von beiden Seiten als aktivierbare Nachbarn erreichbar sind. Bei dem Entwurf eines Szenarios mit mehreren Mittelspannungsringen galt hierbei die Vorgabe, dass die Mittelspannungsringe in quadratischer Anordnung zueinander hochskaliert werden sollten.

Innerhalb des Mittelspannungsringes befinden sich gleichmäßig viele flexible Erzeuger und Verbraucher, welche sich gegenseitig abwechseln. Diese haben in beiden Zeitschritten jeweils Powerwerte zwischen 1 und 5 beziehungsweise -1 und -5 zur Auswahl. In den Niederspannungssträngen sind gleichmäßig viele dynamische sowie statische Erzeuger und Verbraucher verteilt. Dabei besitzen die dynamischen Erzeuger die Powerwerte 1 im ersten Zeitschritt und 2 im zweiten Zeitschritt und die dynamischen Verbraucher die Powerwerte -1 im ersten Zeitschritt und -2 im zweiten Zeitschritt. Für die statischen Erzeuger und Verbraucher sind die Powerwerte 2 und -2 vorgesehen. Die Agenten wurden in den Niederspannungssträngen dabei so verteilt, dass sich möglichst ein Erzeuger neben einem Verbraucher und dynamische neben statischen Agenten befinden. Weiterhin sind innerhalb des Mittelspannungsringes insgesamt vier Shortcuts eingebaut, welche gleichermaßen als Strom- und Kommunikationskante ausgeprägt sind.

Mit Hilfe der beschriebenen Konstruktionsvorschriften konnten relativ einfach skalierbare Netze zur Validierung der Eingabeparameter entworfen werden. Dabei wurden, wo nötig, die standardisierten Vorgaben durch die Anforderungen der jeweiligen Eingabeparameter aufgehoben. Beispielsweise die Anzahl der Sprecher bei der Unter-

suchung des Eingabeparameters „Anzahl der Sprecher“, da diese den Betrachtungsgegenstand in diesem Fall darstellten oder auch die Höhe der Powerwerte, welche zur Betrachtung der Länge der Niederspannungsstränge angepasst werden musste, um eine Lösbarkeit der Netze zu gewährleisten.

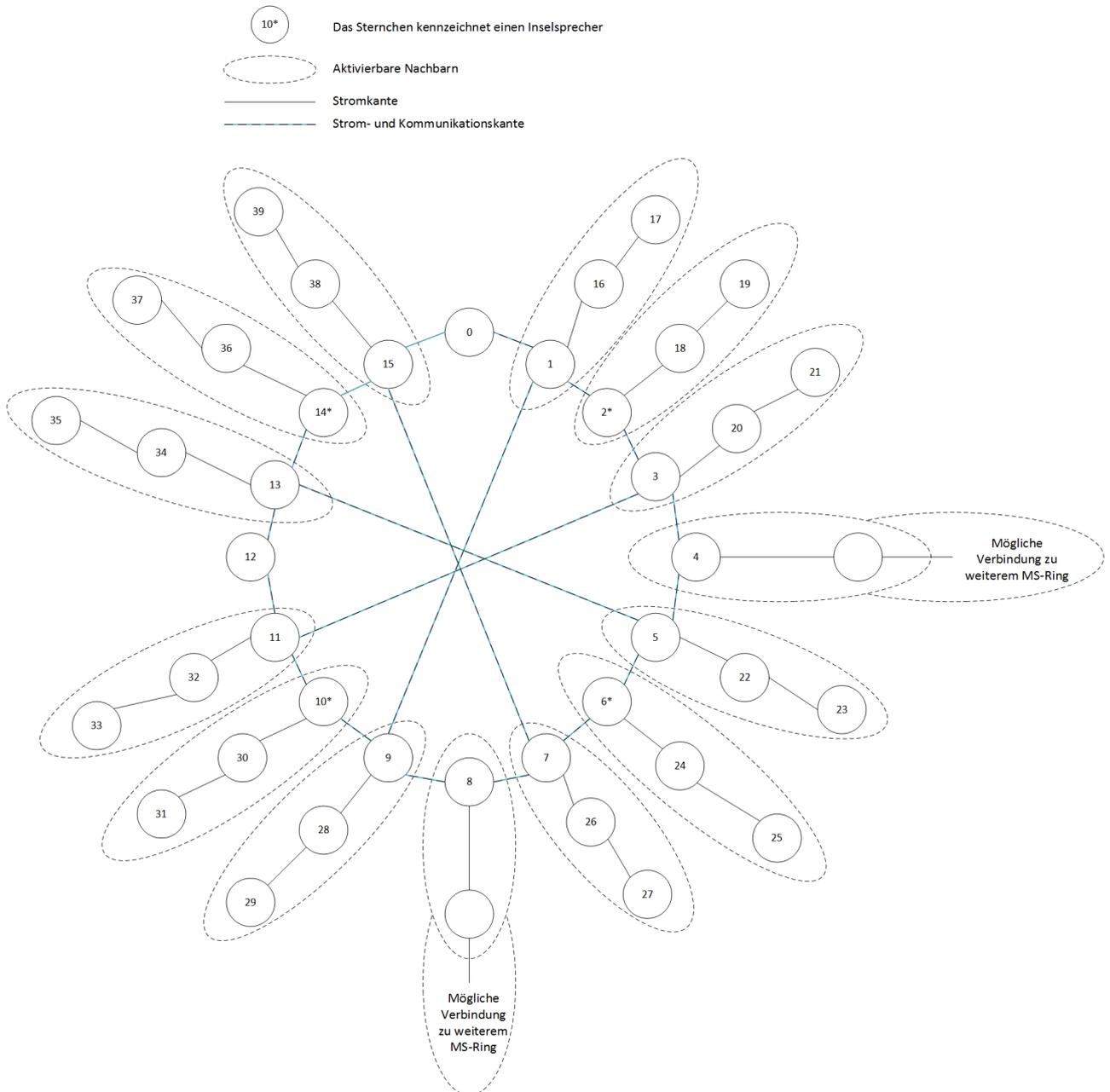


Abbildung 7.1: Standardisierter Mittelspannungsring

## 7.2 Ergebnisse und Interpretation

Im folgenden Abschnitt werden die Ergebnisse der Validierung sowie deren Interpretation dargestellt. Dabei werden sowohl allgemeine Erkenntnisse als auch die Ergebnisse der einzelnen Eingabeparameter und deren Kombinationen erläutert. Die ausführlichen Ergebnisse befinden sich im Anhang in Abschnitt A.

### 7.2.1 Allgemeine Erkenntnisse

Die Validierung lieferte einige parameterunabhängige oder -übergreifende Erkenntnisse, die hier zusammengefasst dargestellt werden.

Die Anzahl der Nachrichten wird größtenteils bestimmt durch die Anzahl der versendeten Arbeitsgedächtnisse. Für eine Reduzierung der Nachrichten sollte also bei den Verhandlungen angesetzt werden. Beispielsweise könnte versucht werden, die Anzahl der Verhandlungen insgesamt zu reduzieren oder die Anzahl der Verhandlungen mit vielen Agenten. Dafür müsste der Systemablauf so angepasst werden, dass sich zunächst viele kleinere Inseln bilden und Inselvereinigungen erst stattfinden, wenn die Inseln sich nicht mehr selbstständig vergrößern können. Ein anderer Ansatz wäre eine Anpassung des Overlay-Netzes, da bisher im System alle zur Verfügung stehenden Kommunikationskanten auch genutzt wurden. Im Zuge der Evaluation von COHDA wurde gezeigt, dass die Wahl des Overlay-Netzes einen deutlichen Einfluss auf die Nachrichtenanzahl hat [Hirr14, 7.3]. Eine eingeschränkte Kommunikation, zum Beispiel in Form einer Small World Topologie, benötigt dabei deutlich weniger Nachrichten im Gegensatz zu einem vollvermaschten Netz.

Death Penalty und Everybody-On-Blacklist tritt nur sehr selten auf, bedingt durch das einfache Design der Szenarien, und kann daher keine Aussagekraft zur Güte des Systems liefern, siehe auch Abschnitt 7.1.1.7 .

Bei der Betrachtung des Verlaufs des Netzaufbaus, insbesondere bei größeren Netzen, wurde festgestellt, dass sich der Aufbau mit zunehmendem Fortschritt verlangsamt. Zu Beginn werden sehr schnell viele Agenten in Inseln aufgenommen, während die meiste Systemlaufzeit gegen Ende dafür benötigt wird, die letzten Agenten ans Netz anzuschließen. Der Grund für diese zunehmende Verlangsamung liegt vermutlich in der Arbeitsweise des MiniMAS. Mit zunehmendem Fortschritt des Netzaufbaus erhöht sich die Anzahl der Agenten in einer Verhandlung, wodurch mehr Nachrichten versendet werden, die von den Agenten abgearbeitet werden müssen. Da das MiniMAS die Agenten jeweils sequenziell und nicht parallel aufruft, dauert ein einzelner Step umso länger, je mehr Nachrichten die Agenten haben. Bei größeren Netzen wurde außerdem eine hohe Rechenleistung benötigt, bedingt durch die große Anzahl an Nachrichten und die dementsprechend große Anzahl an ausgeführten Optimierungen. Dabei ist weniger die Anzahl der Agenten entscheidend, sondern hauptsächlich die Topologie des Netzes, die großen Einfluss auf die Anzahl der versendeten Nachrichten hat, wie auch an den Ergebnissen in Abschnitt 7.2.2.1 und 7.2.2.2 zu sehen ist. Um die Leistung des Systems zu verbessern, könnte einerseits, wie bereits erwähnt, eine Anpassung des Overlaynetzes durchgeführt werden oder andererseits die Implementierung im Hinblick auf Speichernutzung optimiert werden.

## 7.2.2 Versuchsergebnisse

Die folgenden Abschnitte beschreiben die Versuchsergebnisse der einzelnen Eingabeparameter und deren Kombinationen sowie jeweils die Interpretation dieser Ergebnisse.

### 7.2.2.1 Länge der Niederspannungsstränge

Bei der Eingabeklasse der Länge der Niederspannungsstränge wurde der Standard Mittelspannungsring verwendet. An diesem wurde die Länge der Niederspannungsstränge schrittweise von 1 bis 50 erhöht.

In der Auswertung fällt auf, dass zwischen allen gemessenen Parametern und der Länge der Niederspannungsstränge ein deutlicher Zusammenhang besteht. Eine Ausnahme stellt dabei die Vergrößerung von 0 auf 1 Agenten in den Niederspannungssträngen dar, der kaum Auswirkungen beigemessen werden können. Dies liegt darin begründet, dass dieser Agent eine Stromkante in den Mittelspannungsring hat. Er kann deshalb sofort, ohne, dass erst andere Agenten im Niederspannungsstrang angeschlossen werden müssen, in eine Verhandlung integriert werden. Deshalb kann er bereits im gleichen Zyklus, in dem der Mittelspannungsring verarbeitet wird, berücksichtigt werden - ohne, dass eine neue Verhandlung benötigt wird. Death-Penalty oder Everybody-on-Blacklist Situationen sind nicht aufgetreten und werden deshalb hier nicht betrachtet.

Die Anzahl der Verhandlungen steigt, mit der oben benannten Ausnahme, augenscheinlich linear mit der Anzahl der Agenten in den Niederspannungssträngen (vgl. Abb. 7.2). Alle Agenten eines Niederspannungsnetzes können, durch die im Szenario definierte Topologie [vgl. Abschnitt 3.2), nur nacheinander in die Insel integriert werden. Insofern braucht die Hinzunahme jeder weiteren Stufe in den Niederspannungssträngen mindestens eine neue Verhandlung. Hierdurch verhält sich auch die Anzahl der Steps (vgl. Abb. 7.2) augenscheinlich linear zur der Länge der Niederspannungsstränge.

An dieser Stelle stellt sich die Frage, inwiefern die in der Szenariodefinition vorgesehene Konstruktion der Niederspannungsstränge sinnvoll ist. Wenn Agenten entlang eines Niederspannungsstranges unabhängig von ihrer Reihenfolge hinzu geschaltet werden könnten, wäre von einer erheblichen Steigerung der Performanz des Algorithmus auszugehen. Inwieweit diese Veränderung des Szenarios der Weiterentwicklung des Verfahrens dienlich ist, kann hier nicht abschließend beurteilt werden.

Der starke Anstieg des Nachrichtenaufkommens in Abbildung 7.2 ist zum einen durch den deutlichen Anstieg an benötigten Verhandlungen, zum anderen auch durch die vollständige Kommunikation in den Niederspannungssträngen zu erklären. So hat

jeder Agent in den Niederspannungsnetzen in einer Verhandlung alle bereits in einer Insel enthaltenen Agenten aus seinem Strang als Nachbarn im Overlay-Netz. Dies bedeutet, dass bei einer Länge des Niederspannungsstrangs von 50 Agenten, ein Agent jede seiner WorkingMemory-Nachrichten mindestens an die 49 anderen Agenten im Niederspannungsnetz und den zugehörigen Agenten im Mittelspannungsnetz sendet. Dies erzeugte ein solch hohes Nachrichtenaufkommen, dass es durch die der Projektgruppe zur Verfügung stehenden Ressourcen nicht mehr bearbeitet werden konnte.

Wie bereits in den Abschnitten 4.3.2 und 7.2.1 dieses Berichts dargelegt, wurde auf die Bildung einer speziellen Topologie im Overlay-Netz verzichtet. Für COHDA wurden in [Hinr14, 7.3.1] verschiedene Netztopologien und ihre Auswirkungen auf das Nachrichtenaufkommen untersucht und gezeigt, dass durch eine geeignete Topologie das Nachrichtenaufkommen reduziert werden kann. Dies lässt für den hier untersuchten Algorithmus schließen, dass durch eine Anpassung der Topologie im Overlay-Netz das Nachrichtenaufkommen positiv beeinflusst werden kann.

Abschließend lässt sich durch die Ergebnisse schließen, dass die Länge der Niederspannungsstränge einen deutlich ungünstigen Einfluss auf die Performanz und die Effizienz des Verfahrens hat. Diese liegen sowohl in der Szenariodefinition, als auch in der fehlenden Anpassung der Topologie des Overlay-Netzes begründet.

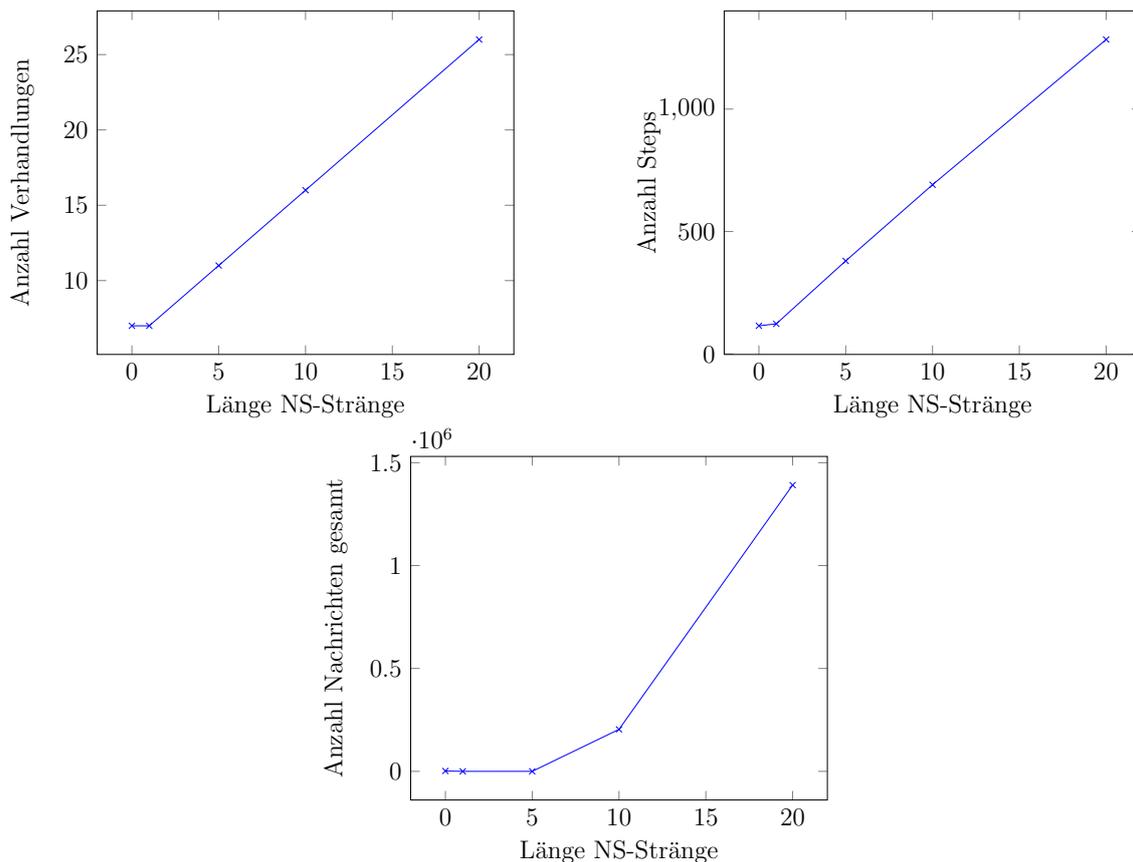


Abbildung 7.2: Validierungsergebnisse bei Skalierung der Länge der Niederspannungsstränge

### 7.2.2.2 Anzahl Agenten

Für die Validierung des Eingabeparameters „Anzahl der Agenten“ wurde die Anzahl der Mittelspannungsringe entsprechend der Vorgabe der Einhaltung der quadratischen Anordnung erhöht. Dabei wurde erst ein Mittelspannungsring und zwei Mittelspannungsringe in Reihe vermessen und anschließend ein Netz aus vier (2x2), neun (3x3) und 16 (4x4) quadratisch angeordneten Mittelspannungsringen. Aufgrund der in 7.1.3 beschriebenen Konstruktionsvorschriften stehen die Anzahl der Mittelspannungsringe und die Anzahl der Agenten, wie in Abbildung 7.3 zu erkennen, in einem linearen Verhältnis zueinander.

Analog der Ergebnisse der Untersuchung der Länge der NS-Stränge, welche prinzipiell auch eine Erhöhung der Anzahl der Agenten bedeutet, lässt sich beobachten, dass die Anzahl der Nachrichten, die Anzahl der Verhandlungen und die Anzahl der Steps mit der Anzahl der Agenten korreliert. Diese Ergebnisse sind in Abbildung 7.3 dargestellt. Allerdings ist der Anstieg des Nachrichtenaufkommens deutlich geringer als bei der Erhöhung der Länge der NS-Stränge, da durch die fixe Länge der NS-Stränge in Höhe von zwei das Nachrichtenaufkommen durch die vollständige Kommunikation und die sequenzielle Zuschaltung der NS-Agenten nicht überproportional ansteigt.

Der lineare Anstieg der geführten Verhandlungsrunden kann durch den symmetrischen Aufbau der Netze, basierend auf den Konstruktionsvorschriften, erklärt werden. Da alle Mittelspannungsringe gleich aufgebaut sind, sind dort auch ähnlich viele Verhandlungen zu erwarten.

Auch der Anstieg der Anzahl der Steps bestätigt diese These. Beim Anstieg von einem auf zwei Mittelspannungsringe erhöht sich die Anzahl der benötigten Steps lediglich um einen, von 203 auf 204. Dies ist dadurch zu erklären, dass, ausgehend von den beiden einzelnen Mittelspannungsringen, zuerst zusammenhängende Inseln gebildet werden, welche sich schlussendlich über die einzige mögliche Verbindung zusammenschließen. Da es bei der weiteren Erhöhung der Anzahl der Mittelspannungsringe mehrere Verbindungen zwischen den Mittelspannungsringen gibt, steigt die Anzahl der benötigten Steps danach schneller an, da die Möglichkeiten der Inselzusammenschlüsse komplexer werden. Die Situationen einer Death Penalty oder Everybody-On-Blacklist sind beim Vermessen der Anzahl der Agenten nicht aufgetreten.

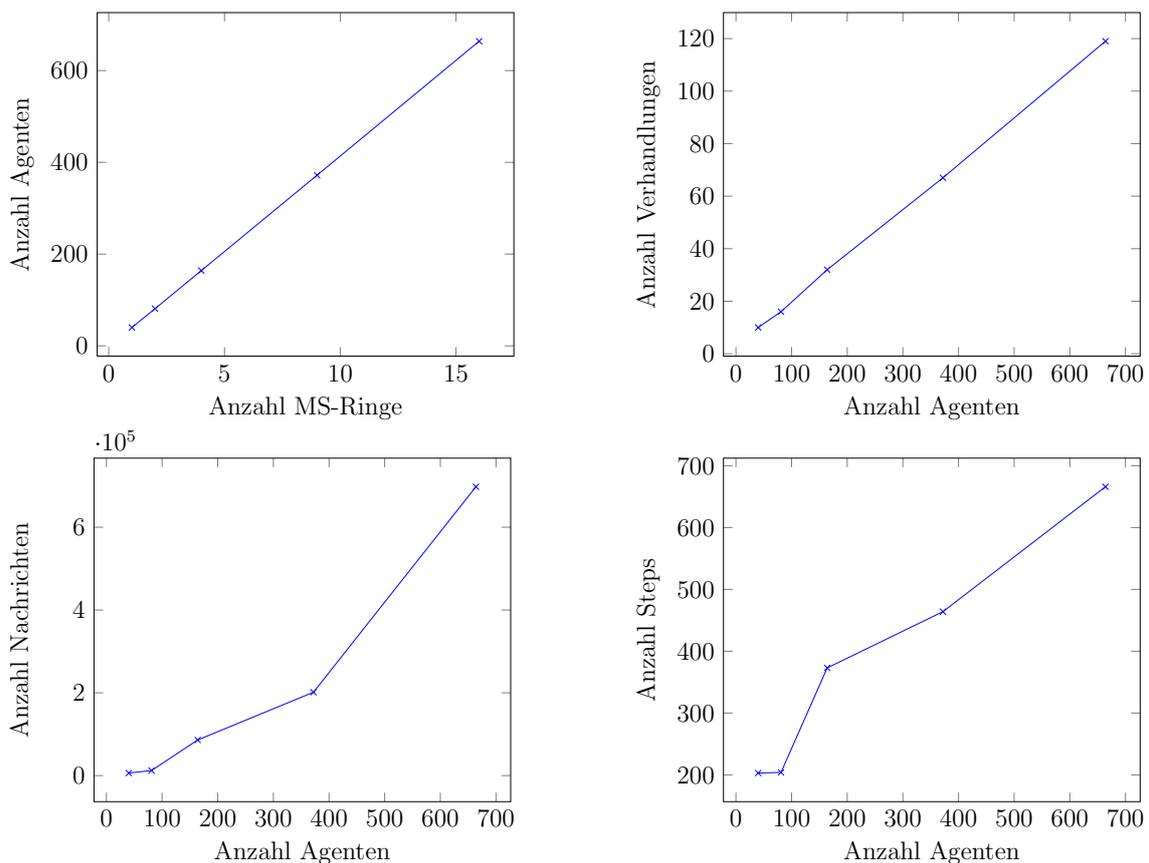


Abbildung 7.3: Validierungsergebnisse bei Skalierung der Anzahl der Agenten

### 7.2.2.3 Anzahl Sprecher

Die Anzahl der Sprecher wurde in einem Standard-MS-Ring von 1 bis 16 variiert. Sprecher wurden nur im Mittelspannungsteil platziert, wobei bei 16 Sprechern der

komplette MS-Ring aus Sprechern bestand. Die Verteilung der Sprecher bei weniger als 16 erfolgte dabei möglichst gleichmäßig. Wie in Abbildung 7.4 zu sehen, nimmt die Anzahl der Nachrichten tendenziell mit Anzahl der Sprecher ab, was darauf hindeutet, dass eine größere Sprecheranzahl zur Reduktion von Nachrichten führen kann. Da in mehreren kleineren Verhandlungen weniger Nachrichten ausgetauscht werden als in wenigen großen, erscheint dies logisch. Allerdings ist auch zu sehen, dass die optimale Sprecheranzahl für minimalen Nachrichtenaufwand bei 4 liegt. Abhängig von der Topologie des Netzes können sich Sprecher auch gegenseitig blockieren, weshalb im Einzelfall auch eine niedrigere Anzahl das Optimum sein kann. Die Reduzierung von Nachrichten durch die Erhöhung der Sprecheranzahl kann also nicht pauschalisiert werden, sondern muss für einzelne Szenarien geprüft werden. Berücksichtigt werden muss außerdem, dass es sich bei den Sprechern in diesem Fall immer um flexible Erzeuger handelte, wodurch sie eine gute Grundlage hatten, um verschiedene Agenten auszugleichen. Ein statischer oder dynamischer Agent hätte weniger Variabilität, was dazu führen könnte, dass diese Sprecher den Netzaufbau eher blockieren. Würde man als Voraussetzung für die Sprecherrolle festlegen, dass der jeweilige Agent flexibel sein sollte, könnte die Sprecheranzahl in einem realen Netz nicht beliebig variiert werden, sondern müsste an die Gegebenheiten angepasst werden.

In Abbildung 7.4 wird zudem gezeigt, wie sich die Anzahl der Verhandlungen mit der Skalierung der Sprecher entwickelt. Es kann nicht pauschal gesagt werden, dass weniger Verhandlungen effizienter sind, da wenige große Verhandlungen beispielsweise einen wesentlich höheren Nachrichtenaufwand haben als viele kleine. Der Verlauf der Kurve ist ebenfalls durch die Topologie des Netzes begründet. Da sich die Sprecher nur im Mittelspannungsring befinden und die Agenten im Niederspannungsnetz erst dann erreichbar sind, wenn die Kommunikation aktiviert wurde, kann die erste Verhandlungsrunde nur im MS-Ring stattfinden. Zunächst steigt die Anzahl an Verhandlungen mit Anzahl der Sprecher, da diese noch wenig genug sind, um sich nicht gegenseitig zu blockieren, sodass viele kleinere Verhandlungen durchgeführt werden können. Mit zunehmender Anzahl kommt es schließlich dazu, dass es vor Beginn der ersten Verhandlungsrunde erst einige Inselvereinigungen gibt, sodass sich die Sprecheranzahl deutlich reduziert und damit auch die Anzahl an Verhandlungen.

Zuletzt wird noch das Verhältnis der Steps zur Anzahl der Sprecher dargestellt. Hier wird deutlich, dass die Anzahl der Steps nicht unbedingt mit der Anzahl der Nachrichten korrelieren muss, da beispielsweise große Verhandlungen in wenigen Steps viele Nachrichten erzeugen, während viele kleine Inseln, die sich gegenseitig oft verpassen oder für längere Zeit in WAIT warten, viele Steps durchlaufen, ohne dabei viele Nachrichten zu versenden. Dennoch ist eine leichte Tendenz zu erkennen, dass mehr Sprecher es in weniger Steps schaffen, das Netz wieder aufzubauen.

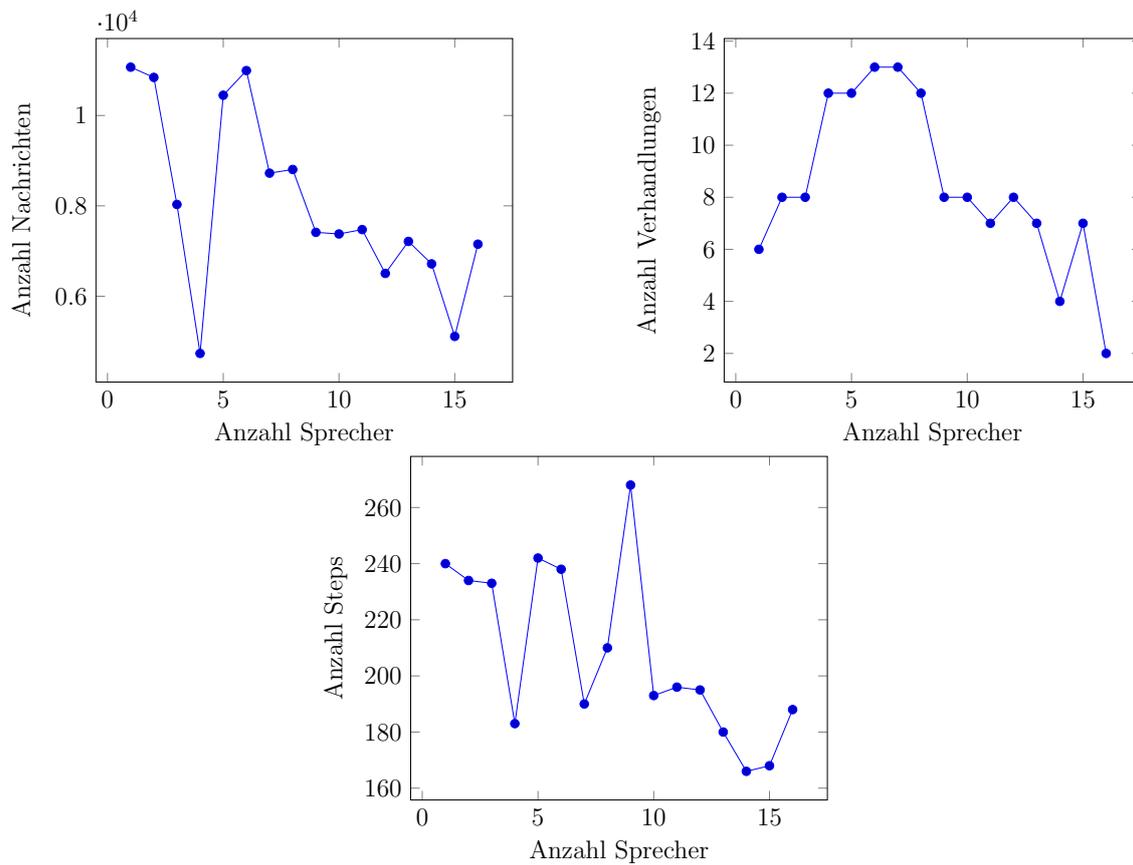


Abbildung 7.4: Validierungsergebnisse bei Skalierung der Anzahl der Sprecher

#### 7.2.2.4 Anteil flexibler Erzeuger und Verbraucher

Der Einfluss der flexiblen Erzeuger und Verbraucher auf Performanz und Effizienz wurde anhand der Variation des prozentualen Anteils dieser im Standard-MS-Ring gemessen. Dabei wurden nacheinander 15%, 50% und 85% der Agenten mit Flexibilität ausgestattet. Niedrigere Parameter wurden nicht getestet, da dafür die Netzkonstellation im verwendeten Standardnetz hätte angepasst werden müssen, was die Ergebnisse verzerrt hätte.

In Abbildung 7.5 ist zu sehen, dass mit dem Anstieg des Anteils der flexiblen Erzeuger und Verbraucher die Anzahl der Verhandlungen sinkt. Gleichzeitig zeigt sie, dass der Anteil der Verhandlungen, die mit einer Death-Penalty enden, abnimmt. Dies lässt sich dadurch erklären, dass ein Netz durch mehr Flexibilität auf eine größere Anzahl von Wegen lösbar ist. Dies ist dadurch bedingt, dass flexible Agenten die einzigen sind, die in einer Optimierung überhaupt Spielräume bei der Auswahl ihrer Leistungen haben. Dynamische und statische Agenten können entweder Teil der Insel werden oder sich auf die Blacklist setzen.

Bei Betrachtung der Entwicklung des Nachrichtenaufkommens in Abbildung 7.5 fällt auf, dass eine Erhöhung der Flexibilität zu einer Verringerung des Nachrichtenaufkommens führt. Dies ist unter anderem auf die Verringerung der Anzahl der Verhand-

lungen zurückzuführen. Ein weiterer vermuteter Einfluss ist die Beschleunigung von Verhandlungen, dass also Lösungen schneller gefunden werden und dadurch weniger Nachrichten versendet werden müssen. Darauf schließen lässt auch die Verringerung der Steps in Abbildung 7.5. Da hierzu allerdings keine Daten erhoben wurden, bleibt dies eine Vermutung.

Der Anteil der Flexibilität in einem Netz zeigt somit einen positiven Einfluss auf die Performanz und die Effizienz des Algorithmus. Zu bedenken ist hierbei aber, dass dieser Parameter in der Realität nicht variiert werden kann, sondern von dem zu lösenden Netz vorgegeben ist.

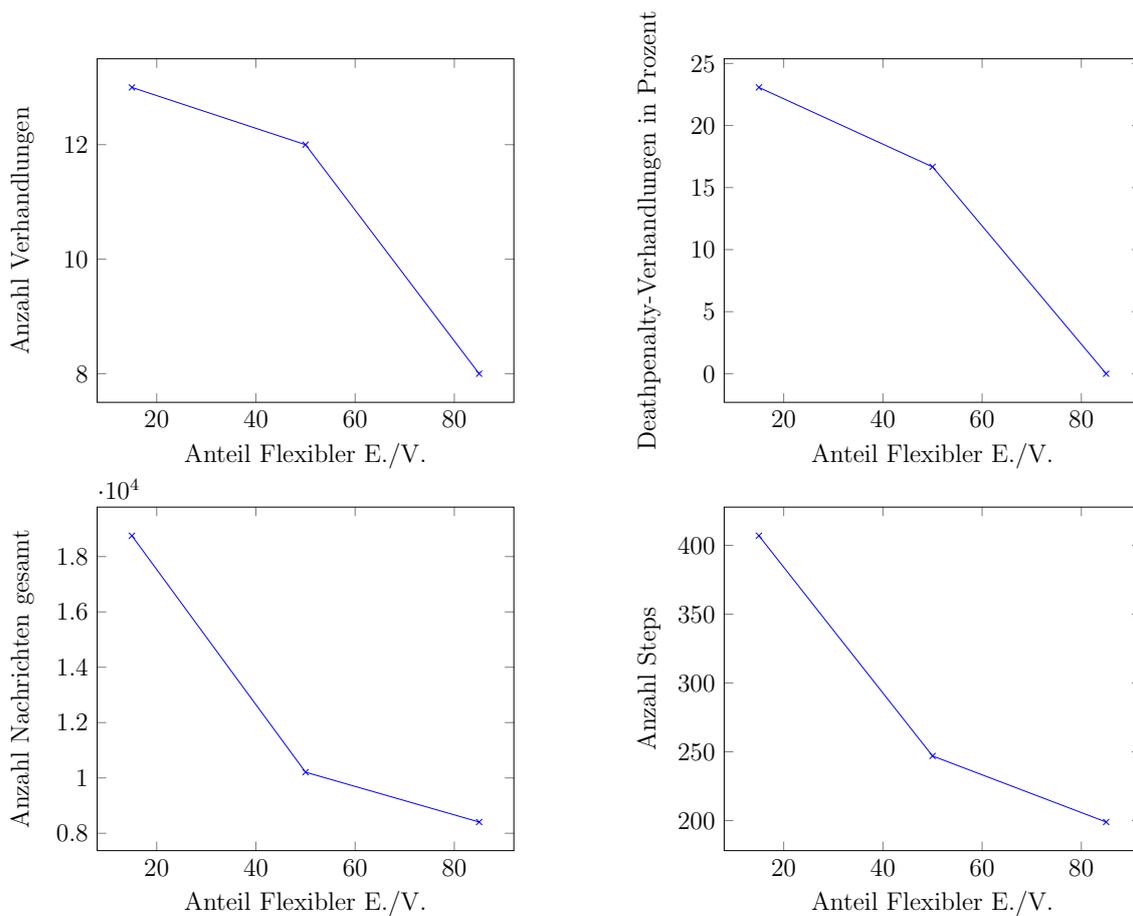


Abbildung 7.5: Skalierung des Anteils flexibler Erzeuger und Verbraucher

### 7.2.2.5 Anzahl Shortcuts

Bei der Anzahl der Shortcuts wurden die Auswirkungen von Kommunikations- und Stromshortcuts innerhalb des Mittelspannungsringes auf die Ausgabeparameter untersucht. Shortcuts stellen in diesem Sinne Kanten zwischen Knoten dar, welche im Mittelspannungsring nicht direkt nebeneinander liegen. Hierbei ist, im Rahmen des Versuchsaufbaus, eine Strom- auch gleichzeitig eine Kommunikationskante. Getestet wurden verschiedene Vermaschungsstufen innerhalb des Mittelspannungsringes. Dabei wurden folgende Ausprägungen untersucht:

- 0 Shortcuts
- 1 Shortcut
- 4 Shortcuts
- 8 Shortcuts
- 50 Shortcuts
- 105 Shortcuts (= Vollvermaschung)

Die Ergebnisse dieser Untersuchung sind in 7.6 einzusehen. Grundsätzlich ist zu erkennen, dass eine Erhöhung der Shortcuts nicht zwangsläufig zu einer besseren, beziehungsweise schnelleren, Lösbarkeit der Netze führt. Die Anzahl der benötigten Verhandlungen und Steps kann im Rahmen dieses Versuchsaufbaus als nicht korreliert bezeichnet werden. Die Anzahl der Nachrichten weist in etwa einen linearen Verlauf auf, was durch die Erhöhung der Menge der Kommunikationskanten, in Form der Shortcuts, zu erklären ist.

Insgesamt lässt der Versuchsaufbau, aufgrund des hohen Abstraktionsgrades im Vergleich zur Realität, jedoch keine allgemeinen Rückschlüsse auf den positiven Einfluss des Vorhandenseins von Shortcuts zu. Im vorliegenden Versuchsaufbau sind alle Knoten im Mittelspannungsnetz auch ohne Shortcuts über Strom- und Kommunikationskanten zu erreichen. Wäre ein Knoten beispielsweise kommunikationstechnisch nur über einen Shortcut erreichbar, so würde die Lösbarkeit des Netzes von diesem Shortcut abhängen und diesen unverzichtbar machen. Der Einfluss von Shortcuts hängt also maßgeblich mit dem Aufbau des Netzes zusammen, was in der Realität eine gegebene Größe darstellt.

Es konnte noch beobachtet werden, dass bei einer Vollvermaschung mit 105 Shortcuts zwei Mal eine Death Penalty aufgetreten ist. Dies ist auf komplexe Verhandlungssituationen aufgrund der Vielzahl an möglichen Verhandlungspartnern zurückzuführen. In allen anderen Ausprägungen konnten keine Death Penalty oder Everybody-On-Blacklist-Situationen festgestellt werden.

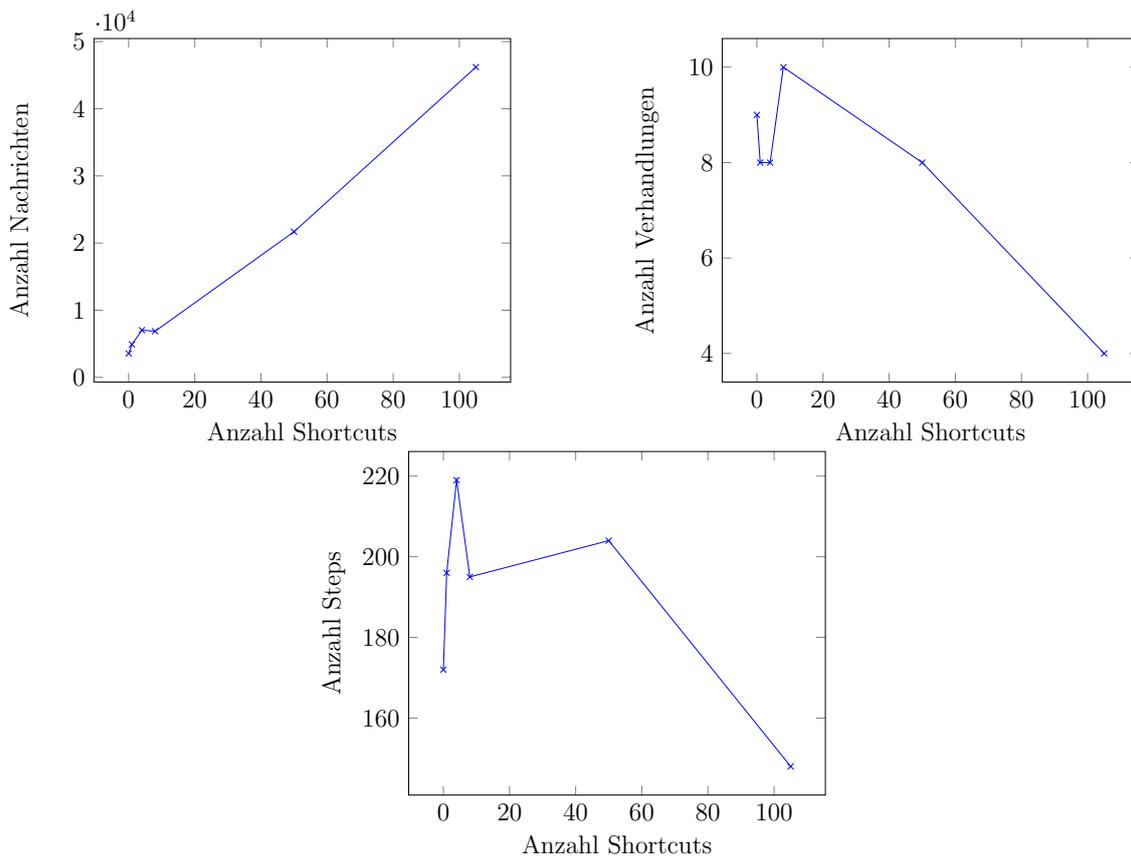


Abbildung 7.6: Validierungsergebnisse bei Skalierung der Anzahl der Shortcuts

### 7.2.2.6 Gewichtung der Zielfunktion

Die Zielfunktion besteht aus drei Gewichten, deren Reihenfolge im Folgenden als fix angesehen wird:

- Sichtbarkeit
- Inselgröße
- Priorisierung

Wie bereits in Abschnitt 7.1.2 beschrieben, konnte zur Untersuchung der Gewichtung der Zielfunktion, sowie der damit eng verknüpften Priorisierung, nicht das im Abschnitt 7.1.3 beschriebene Szenario verwendet werden. Der Grund hierfür ist, dass die Zielfunktion erst zum Tragen kommt, wenn während einer Verhandlung nicht alle Agenten in die Insel integriert werden können und gleichzeitig zwischen der Aufnahme unterschiedlicher Agenten gewählt werden kann. Da die Bilanz in allen Fällen Null beträgt, wird die Zielfunktion zur Bestimmung der besten Lösung gewählt. Die beste Lösung kann dabei je nach Gewichtung variieren. Daher wurde für diese Anforderungen ein spezielles Netz entworfen, welches in 7.7 abgebildet ist.

Agent 0 ist alleiniger Sprecher des Netzes und startet somit den Netzaufbau. Agent 1 kann hierbei noch ohne Probleme zugeschaltet werden. Im folgenden Schritt wird eine Verhandlung mit den Agenten 2, 3, 8 und 9 gestartet. Diese können aufgrund der Ausprägung der Powerwerte nicht im selben Schritt in die Insel aufgenommen werden, sondern es muss gewählt werden, ob Agent 2, 3 oder die Agenten 8 und 9 aufgenommen werden. Somit greift an dieser Stelle die Zielfunktion sowie deren Gewichtung. Agent 2 stellt dabei einen priorisierten Knoten dar. Agent 3 besitzt das Potenzial über seine aktivierbaren Nachbarn die Sichtbarkeit der Insel zu erhöhen. Die Agenten 8 und 9 repräsentieren die Möglichkeit, die Inselgröße zu maximal zu erweitern.

Im Rahmen der Validierung wurde nun untersucht, wie sich verschiedene Gewichtungen der Zielfunktion auf das Ergebnis eines Systemdurchlaufs auswirken. Dazu wurde festgehalten, welche Agenten am Ende des Durchlaufs nicht in die Insel mit aufgenommen werden konnten. Dieses Ergebnis ist in Tabelle 7.1 zu sehen. Dabei wurde der Agent 2 mit einer Priorität in Höhe von 2 versehen, um den Einfluss der Höhe der Priorität gering zu halten und das Ergebnis damit nicht zu verzerren. Es ist zu erkennen, dass bei einseitiger Ausprägung der Gewichtung die Zielfunktion relativ stark greift und sich am Ende meistens einige Agenten nicht in der Insel befinden. Die nach der Gewichtung bevorzugt zu behandelnden Agenten sind jedoch stets in der Insel enthalten. Bei moderaterer Ausprägung der Gewichtung findet der Algorithmus in dem gewählten Szenario jeweils eine Lösung, welche alle Agenten beinhaltet. Dem Aufbau des Szenarios geschuldet, traten während des Programmdurchlauf in den meisten der betrachteten Fällen Death Penalties oder Everybody-On-Blacklist-Situationen auf.

Die Untersuchung der Gewichtung der Zielfunktion hat gezeigt, dass die Gewichtung die gewünschten Lösungen bevorzugt und damit Auswirkungen auf den Programmablauf und dessen Ergebnis hat. Damit kann die Gewichtung der Zielfunktion die Performanz des Systems sowohl positiv als auch negativ beeinflussen. Gleichzeitig hängt dieser Einfluss stark vom Aufbau des Netzes ab. Daraus ergeben sich außerdem Potenziale die Gewichtung der Zielfunktion dynamisch während des Netzaufbaus anzupassen, um so gezielt die Performanz des Algorithmus zu verbessern.

Im Rahmen der Validierung ist dabei noch aufgefallen, dass in bestimmten Situationen eine zu hohe Gewichtung der Sichtbarkeit zu einem Everybody-On-Blacklist führen kann. Der Grund hierfür ist, dass eine Insel mit zwei Nachbarn eine höhere Sichtbarkeit hat als eine Insel, die beide Nachbarn aufnimmt. Ein Lösungsvorschlag der Projektgruppe wäre, hierbei mithilfe einer Penalty-Funktion die Güte einer Everybody-On-Blacklist-Lösung zu reduzieren.

Gewichtung der Zielfunktion (Sichtbarkeit; Inselgröße; Priorisierung)	Nicht aufgenommene Agenten
0; 0; 1	3, 5, 6, 7
0; 1; 0	-
1; 0; 0	2, 8, 9, 5, 6, 7
0,2; 0,2; 0,6	3, 5, 6, 7
0,2; 0,6; 0,2	-
0,6; 0,2; 0,2	2, 8, 9, 5, 6, 7
0,4; 0,4; 0,2	-
0,4; 0,2; 0,4	-
0,2; 0,4; 0,4	-
0,33; 0,33; 0,33	-

Tabelle 7.1: Untersuchung der Gewichtung der Zielfunktion

### 7.2.2.7 Priorisierung

Aufbauend auf dem in Abschnitt 7.2.2.6 beschriebenen Szenario, wurde die bevorzugte Berücksichtigung von Agenten innerhalb einer Verhandlung auf Basis der Höhe der Priorität untersucht. Dazu wurde die Höhe der Priorität mit dem Zeitpunkt der Aufnahme des priorisierten Agenten innerhalb einer Verhandlung verglichen. Hierfür wurde Agent 2 aus dem in Abbildung 7.7 dargestellten Netz mit Prioritäten in Höhe von 2, 5 und 10 versehen. Zusätzlich wurden verschiedene Gewichtungen der Zielfunktion, analog der Ausprägungen in Tabelle 7.1, untersucht.

In Tabelle 7.2 ist dieser Vergleich beispielhaft für folgende Gewichtung der Zielfunktion angestellt worden:

- Sichtbarkeit = 0,4
- Inselgröße = 0,4
- Priorisierung = 0,2

Der Vergleich zeigt, dass Agent 2 bei einer höheren Priorisierung entsprechend früher in die Insel aufgenommen wird. Die Verschiebung dieses Zeitpunkts stößt aufgrund der Ausprägung des Szenarios zwar an seine Grenzen, zeigt aber, dass die Höhe der Priorisierung Auswirkungen auf den Zeitpunkt hat. Zusätzlich werden bei der Priorisierung in Höhe von 10 die Agenten 3, 5, 6 und 7 nicht mit in die Insel aufgenommen, was die Wirksamkeit der Priorisierung zusätzlich bestätigt. Diese beiden

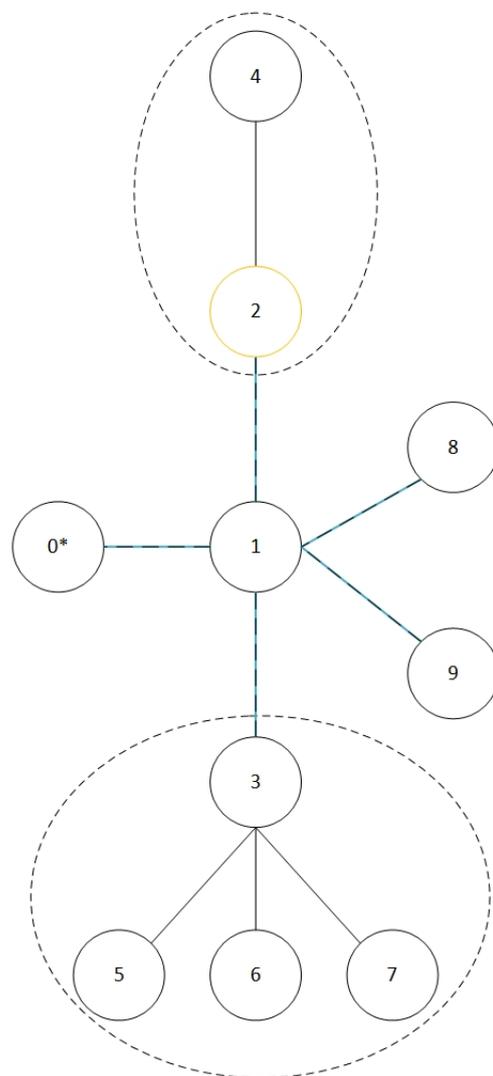
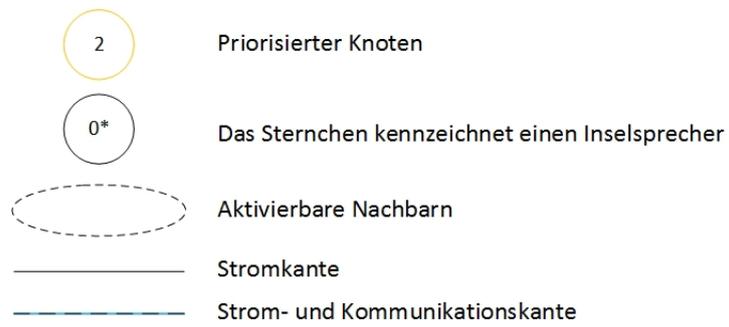


Abbildung 7.7: Szenario zur Validierung der Gewichtung der Zielfunktion und der Priorisierung

Trends, dass der priorisierte Agent früher zugeschaltet wird und gleichzeitig bei höherer Priorität Agenten nicht mit aufgenommen werden können, lässt sich auch für andere Gewichtungen der Zielfunktion ablesen.

Höhe der Priorität	Step der Aufnahme / Gesamtanzahl Steps
2	94 / 123
5	51 / 124
10	49 / 132

Tabelle 7.2: Step der Aufnahme der priorisierten Agenten

### 7.2.2.8 Kombination von Parametern

Für die Validierung kombinierter Parameter wurden zum einen die Anzahl der Sprecher mit der Länge der NS-Stränge und zum anderen die Anzahl der Sprecher mit der Anzahl der MS-Ringe kombiniert, also letztendlich die Anzahl der Sprecher mit der Anzahl der Agenten in unterschiedlichen Netztopologien. Dies waren die Parameterkombinationen, bei denen der größte Erkenntnisgewinn zu erwarten war.

Die Gewichtung der Zielfunktion sowie die Priorisierung sind nur innerhalb einzelner Verhandlungen von Bedeutung, wenn es innerhalb dieser Verhandlung mehrere Lösungen mit gleicher Bilanz gibt, unter denen gewählt werden muss. Das Entstehen dieser gleichwertigen Lösungen hängt nur von den Leistungswerten der Agenten in dieser Verhandlung ab und nicht von der Gesamtzahl der Agenten, der Anzahl der Sprecher, dem Anteil der flexiblen Agenten oder der Anzahl der Shortcuts. Die Ergebnisse von der Skalierung der flexiblen Agenten in Abschnitt 7.2.2.4 lassen vermuten, dass der positive Einfluss einer größeren Anzahl flexibler Agenten auch bei größeren Netzen bzw. auch bei einer größeren Sprecheranzahl gegeben wäre. Allerdings ist der Anteil der flexiblen Erzeuger und Verbraucher im Stromnetz keine Variable auf die das System Einfluss hat, sodass es nicht zielführend erschien, diesen Einfluss auf die Effizienz des Systems zu zeigen. Die Ergebnisse der Anzahl der Shortcuts in Abschnitt 7.2.2.5 zeigen, dass analog zu COHDA die Anzahl der Nachrichten mit der Anzahl der Kommunikationsmöglichkeiten steigt, weshalb eine Reduktion der Kommunikationswege sinnvoll ist. Auch die Ergebnisse aus Abschnitt 7.2.2.1 zur Länge der NS-Stränge zeigen, dass vollständige Kommunikation in größeren Netzabschnitten das Nachrichtenaufkommen massiv erhöht. Dieses Phänomen würde auch bei höherer Sprecheranzahl bestehen bleiben und durch eine größere Anzahl Agenten nur verstärkt werden, weshalb auf diese Kombinationen verzichtet wurde.

Für die getesteten Kombinationen wurde nicht die vollständige Skalierung der Sprecher von 1-16 betrachtet, sondern ausgewählte Werte, um die Anzahl an Testfällen

überschaubar zu halten. Betrachtet wurden die Sprecherzahlen 1,2,4,8,11 und 16. Die Positionierung der Sprecher orientierte sich an der Verteilung der Sprecher in Abschnitt 7.2.2.3. Die jeweilige Sprecheranzahl wurde anschließend mit allen Skalierungen der Klasse „Anzahl der Agenten“ und der Klasse „Länge der NS-Stränge“ kombiniert. Die wichtigsten Erkenntnisse sollen im Folgenden vorgestellt werden, die ausführlichen Ergebnisse befinden sich im Anhang.

### Kombination Anzahl der Sprecher und Länge der NS-Stränge

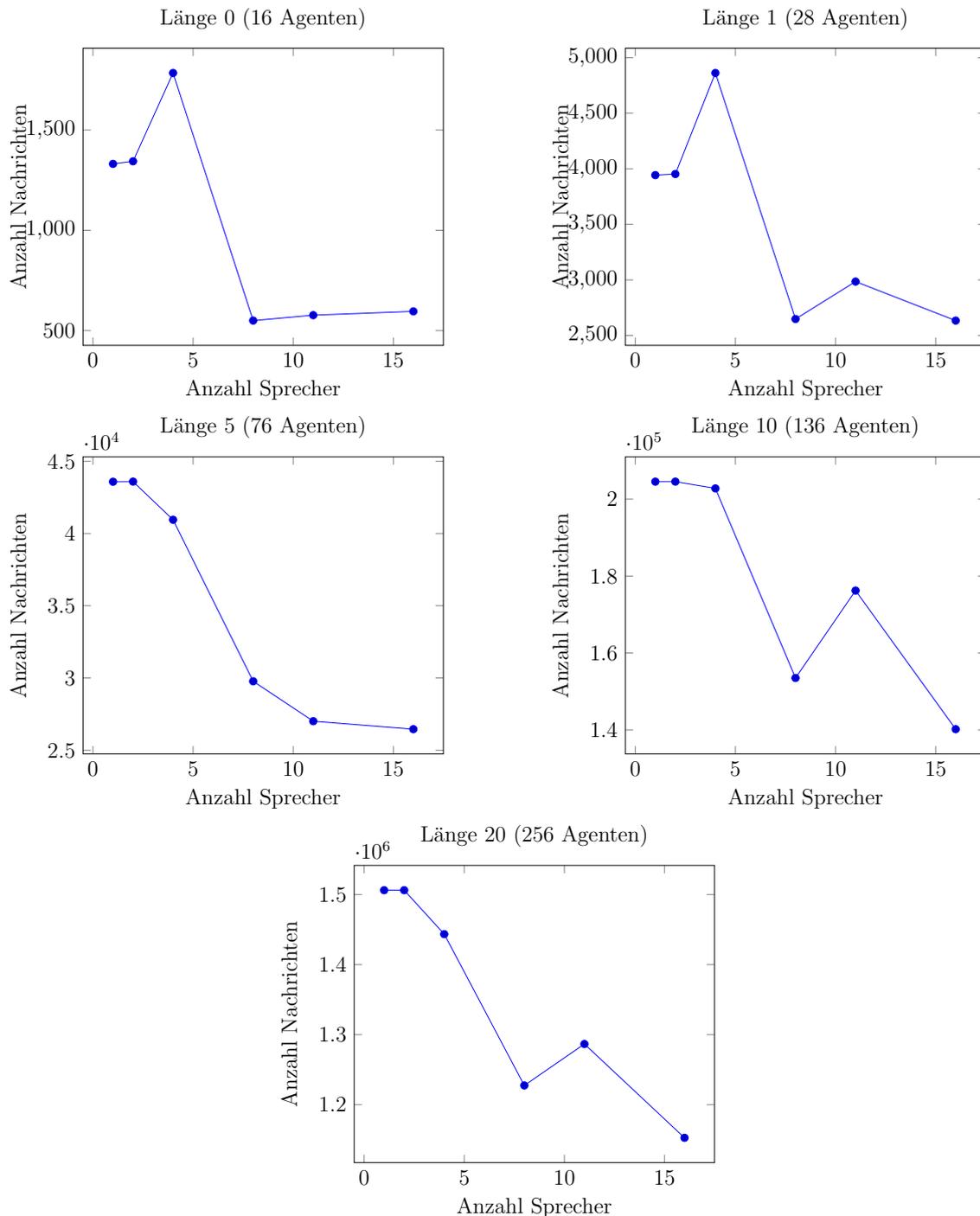


Abbildung 7.8: Kombination Sprecher und Länge NS-Stränge

Abbildung 7.8 zeigt die gesammelten Ergebnisse von Nachrichtenanzahl und Anzahl der Sprecher in den verschiedenen Skalierungen der NS-Stränge. Es ist zu erkennen, dass tendenziell mit Erhöhung der Sprecheranzahl die Anzahl der Nachrichten abnimmt. Mit zunehmender Länge der NS-Stränge wird dieser Effekt jedoch geringer. Betrachtet man jeweils die Anzahl der Nachrichten bei einem und bei 16 Sprecher und berechnet um wieviel Prozent sich die Nachrichtenzahl verringert hat, kommt man auf folgende aufgerundete Ergebnisse:

- Länge NS-Stränge 0: 56%
- Länge NS-Stränge 1: 34%
- Länge NS-Stränge 5: 40%
- Länge NS-Stränge 10: 32%
- Länge NS-Stränge 20: 24%

Dieses Verhalten liegt darin begründet, dass die Sprecher nach jeder Verhandlungsrunde zunächst versuchen, eine Inselvereinigung durchzuführen. Je länger die NS-Stränge, desto mehr Verhandlungsrunden müssen durchgeführt werden und desto wahrscheinlicher ist es, dass eine Inselvereinigung stattfindet, ehe der komplette Strang in die initiale Insel aufgenommen werden kann. Dadurch finden die letzten Verhandlungsrunden mit wenigen, sehr großen Inseln statt, was ein hohes Nachrichtenaufkommen verursacht, bedingt durch die vollständige Kommunikation in den NS-Strängen. Es ist allerdings auch zu sehen, dass die Abnahme nicht stetig und gleichmäßig erfolgt.

### **Kombination Anzahl der Sprecher und Anzahl der MS-Ringe**

Um die Symmetrie zu erhalten, wurden bei diesen Kombinationen die Anzahl der Sprecher in allen MS-Ringen gleichzeitig erhöht.

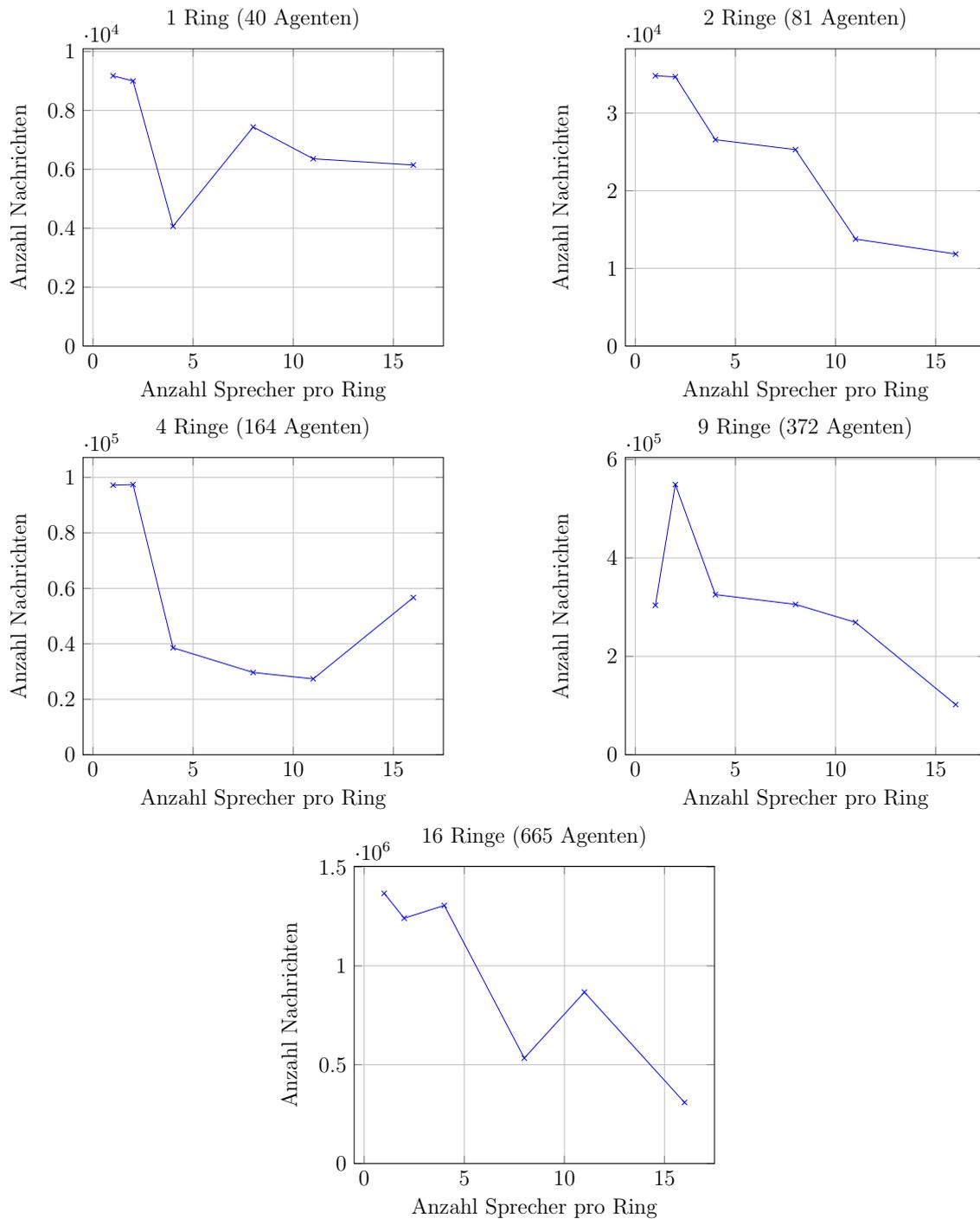


Abbildung 7.9: Kombination Sprecher und Anzahl MS-Ringe

Die in Abbildung 7.9 dargestellten Ergebnisse zeigen das Verhältnis der Nachrichtenanzahl zur Anzahl der Sprecher in den MS-Ringen. Betrachtet man zunächst um wieviel Prozent die Nachrichtenanzahl mit maximaler Sprecheranzahl verringert wurde zeigt sich, dass eine größere Reduktion als bei der Kombination mit der Länge der NS-Stränge erreicht wird:

- Anzahl MS-Ringe 1: 33%
- Anzahl MS-Ringe 2: 66%
- Anzahl MS-Ringe 4: 42%
- Anzahl MS-Ringe 9: 67%
- Anzahl MS-Ringe 16: 78%

Es muss allerdings berücksichtigt werden, dass das Verhältnis der normalen Agenten zu Sprechern bei diesen Kombinationstests auch mit zunehmender Agentenzahl praktisch gleich bleibt durch die gleichmäßige Erhöhung der Sprecher in den MS-Ringen. Dies ist bei der Kombination mit der Länge der NS-Stränge nicht der Fall und es liegt nahe, dass bei einer Erhöhung der Sprecheranzahl im gesamten Netz und nicht in den einzelnen Ringen, die Reduktion der Nachrichten weniger deutlich ausfallen würde. Außerdem ist zu erkennen, dass die Reduzierung mit höherer Agentenanzahl zunimmt, anstatt wie bei den NS-Strängen abzunehmen. Je mehr MS-Ringe das Netz enthält, desto mehr profitiert es also von einer höheren Sprecheranzahl. Von der generellen Tendenz der Nachrichtenreduktion abgesehen wird auch deutlich, dass bei bestimmten Anzahlen von Sprechern der Nachrichtenversand zunimmt, bedingt dadurch, dass sich Sprecher auch gegenseitig blockieren können.

Als generelles Fazit kann gesagt werden, dass sich eine höhere Sprecheranzahl in der Regel positiv auf das Nachrichtenaufkommen auswirkt. Nicht betrachtet wurde die Positionierung der Sprecher. Es kann aber davon ausgegangen werden, dass die Sprecheranzahl nur dann positiven Einfluss hat, wenn die Sprecher so positioniert werden, dass sie nicht nur von anderen Sprechern umgeben sind, sondern normale Verhandlungen führen können. Zum anderen müssen sie in der Lage sein, wenige Agenten ausgleichen zu können, sodass sich viele kleinere Inseln bilden, die sich später vereinigen können. Die optimale Sprecheranzahl hängt von der Netztopologie ab und kann daher auch in unterschiedlichen Netzabschnitten verschieden ausfallen.

## 7.3 Fazit

In der Validierung sollte das in Abschnitt 4.3 vorgestellte Verfahren auf Performanz und Effizienz geprüft werden. Hierfür wurden die Parameter

- Anzahl Nachrichten
- Anzahl Steps

- Anzahl Verhandlungen
- Anzahl Death Penalty Verhandlungen
- Anzahl Everybody-on-Blacklist Verhandlungen

unter verschiedenen Szenarioausprägungen und Skalierungen untersucht.

Hierbei ergaben sich verschiedene Erkenntnisse, die zum einen Optimierungspotential, als auch wichtige Einflussparameter des Systems aufzeigten.

Zum einen ist deutlich geworden, dass der Umgang mit Niederspannungssträngen verbessert werden muss. Diese verursachen in ihrer momentanen Ausprägung bei einer Skalierung einen signifikant negativen Einfluss auf Effizienz und Performanz des Verfahrens. Diese Problematik war im Vorfeld der Validierung erwartet worden. Durch eine Anpassung der Overlay-Netz Logik und Anpassungen am Szenario, könnte an dieser Stelle ohne großen Aufwand Abhilfe geschaffen werden. Da der Hauptanteil der Nachrichten durch die COHDA Optimierungen entsteht, macht es Sinn, hier auf die bereits durch [Hir14] erarbeiteten Erkenntnisse bezüglich des Nachrichtenaufwandes zurückzugreifen. Das Skalieren der Anzahl der Agenten durch die Erhöhung der Standard-MS-Ringe hat ein positiv zu bewertendes Ergebnis geliefert, so dass nach Anpassung der oben benannten Mängel erwartet werden kann, dass Effizienz und Performanz bei einer Skalierung der Netzgröße erhalten bleiben.

Als weitere Einflussgrößen wurden die Anteile von flexiblen Erzeugern und Verbraucher, sowie die Anzahl der Sprecher identifiziert. Der Anteil von Flexibilität im Netz stellt allerdings eine gegebene Größe dar und spielt deshalb bei der Bewertung des Algorithmus eine untergeordnete Rolle. Die Anzahl der Sprecher hingegen kann im Rahmen der gegebenen Netzausprägung, wobei ein Sprecher eine schwarzstartfähige Energieanlage repräsentieren muss, beeinflusst werden. Hier hat sich gezeigt, dass diese Einflussgröße sich in der Skalierung zwar grundlegend positiv auf die Effizienz und die Performanz auswirkt, aber der Zusammenhang doch komplex ist. An dieser Stelle würde sich eine weitergehende Untersuchung lohnen, die auch die Platzierung der Sprecher im Netz berücksichtigt und so deren Einfluss noch genauer beurteilen kann. Dadurch könnte die Anzahl und Positionierung der Sprecher für verschiedene Netze optimiert und damit die Effizienz und Performanz des Algorithmus verbessert werden.

Die Untersuchung der Zielfunktion und der Priorisierung hat gezeigt, dass diese den Netzaufbau aktiv beeinflussen können. Auch hier muss der genaue Einfluss noch durch weitere Untersuchungen erfasst werden. Dies würde eine Optimierung der Gewichtung für spezifische Netze möglich machen, wodurch eine Einsatzfähigkeit des Algorithmus für unterschiedlich ausgeprägte Netze unterstützt wird.

---

Zusammengefasst bietet der Algorithmus eine gute Grundlage, die durch einfache Anpassungen in ihrer Performanz und Effizienz noch deutlich verbessert werden kann. Zudem wird für die Zukunft neben anderen Erweiterungen interessant sein, diejenigen Stellschrauben noch genauer zu evaluieren, mit denen der Wiederaufbau eines Stromnetzes aktiv verbessert und beeinflusst werden kann.

## 8. Fazit und mögliche Erweiterungen

Der Startschuss der Projektgruppe fiel im April 2016. Begonnen wurde mit der Auseinandersetzung der Projektmitglieder mit verschiedenen thematischen Aspekten der Aufgabenstellung im Rahmen der Seminarphase. Danach befasste sich die Projektgruppe mit der Spezifikation der Problemstellung in Form der Definition des zugrundeliegenden Szenarios sowie der Erarbeitung eines Lösungskonzeptes. Auf Basis dieser Erkenntnisse wurden die Anforderungen an das zu erstellende Produkt abgeleitet. Über mehrere Prototypphasen hinweg wurde der Algorithmus entwickelt und stetig erweitert. Dies brachte eine Anpassung der Projektorganisation mit sich, da für eine effiziente Zusammenarbeit klare Verantwortlichkeiten und definierte Kommunikationsabläufe notwendig waren. Im Anschluss an die Fertigstellung des dritten und finalen Prototypens wurde das System ausführlich getestet und einer Validierung unterzogen. Die Ziele waren der Nachweis der Erfüllung der Anforderungen, das Finden und Beheben möglichst vieler Fehler sowie die Untersuchung der Güte des entwickelten Algorithmus.

Die im Rahmen der Projektgruppe an das System gestellten Anforderungen konnten, wie in Abschnitt 3.3.2 beschrieben, vollständig erfüllt werden. Das Ergebnis der Projektgruppe basiert hierbei auf einem soweit von der Realität abstrahierten Szenario, das trotz der komplexen Problemstellung ein hoher Erkenntnisgewinn erzielt werden konnte. Der Projektgruppe ist es gelungen, mit der Erarbeitung des Algorithmus und dessen prototypischen Umsetzung, bereits wichtige Herausforderungen des Wiederaufbaus eines Smart Grid nach einem Blackout zu bewältigen. Hierbei ist insbesondere der dezentral und selbstständig organisierte Ablauf des Algorithmus zu nennen, welcher dabei den unterschiedlichen Aufbau der Strom- und Kommunikationsnetze berücksichtigt. Gleichzeitig konnten Weiterentwicklung- und Optimierungspotenziale für zukünftige Forschungsarbeiten identifiziert werden. Diese werden im Folgenden kurz dargestellt.

Zur Verbesserung der Performanz und Effizienz gibt es insbesondere in der Behandlung der Niederspannungsstränge Potenziale. Sowohl durch die Aufhebung der sequenziellen Zuschaltung der Agenten in den Niederspannungssträngen als auch

durch eine Optimierung der Overlay-Netz-Logik können an dieser Stelle relativ einfach Verbesserungen erzielt werden. Weitere Fortschritte und Erkenntnisgewinne sind außerdem durch eine Erweiterung der Simulation zu erwarten. Als mögliche Potenziale sind hierbei die Einbindung in ein Simulationsframework, eine realitätsnähere Abbildung der Energieanlagen sowie die Integration einer Netzsimulationen zu nennen. Dem Algorithmus würde die Möglichkeit bereits geschlossene Leistungsschalter wieder zu öffnen, also Agenten den Austritt aus einer Insel zu ermöglichen, einen größeren Spielraum zur Findung von Lösungen ermöglichen. Weiterhin bietet eine Überprüfung und Erweiterung der Zielfunktion sowie eine Optimierung der Gewichte für spezifische Netzsituationen weitere Potenziale.

# Literaturverzeichnis

- [Ampr17] Amprion. Netzfrequenz. <http://www.amprion.net/netzfrequenz>, 2017. Letzter Zugriff am 30.03.2017.
- [Baum06] D.-I. G. Baumann. Was verstehen wir unter Test? Abstraktionsebenen, Begriffe und Definitionen, October 2006. Letzter Zugriff 26.03.2017.
- [BSBB17] BSI und BBK. Kritische Infrastrukturen - Definition und Übersicht. <http://www.kritis.bund.de/SubSites/Kritis/DE/Einfuehrung/>, 2017. Letzter Zugriff am 29.03.2017.
- [Grü14] R. Grünwald. Moderne Stromnetze als Schlüsselement einer nachhaltigen Energieversorgung. <http://www.tab-beimbundestag.de/de/untersuchungen/u9700.html>, 2014. Letzter Zugriff am 28.03.2017.
- [GRST<sup>+</sup>10] S. B. Ghosn, P. Ranganathan, S. Salem, J. Tang, D. Loegering und K. E. Nygard. Agent-oriented Designs for a Self Healing Smart Grid. 2010.
- [Hinr14] C. Hinrichs. *Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid*. Dissertation, 2014.
- [HiSo14] D. C. Hinrichs und M. Sonnenschein. Evaluation Guidelines for Asynchronous Distributed Heuristics in Smart Grid Applications. In *Proceedings of the 28th EnviroInfo 2014 Conference*, 2014.
- [MeOD14] K. Mets, J. A. Ojea und C. Develder. Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, NO. 3, THIRD QUARTER 2014* 16(3), 2014.
- [NCBI17] NCBI. The Ubiquity of Small-World Networks. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3604768/>, 2017. Letzter Zugriff am 30.03.2017.

- [OFFI16] OFFIS. Documentaion Mosaik. <https://mosaik.offis.de/docs/>, 2016. Letzter Zugriff am 28.03.2017.
- [Posl07] S. Poslad. Specifying protocols for multi-agent systems interaction. 2(4), November 2007.
- [RSLK15] C. Rehtanz, A. Seack, S. Lehnhoff und O. Krause. *Planung und Betrieb von Smart Grids*, S. 10–15. Springer Nature. 2015.
- [Site17] J. Site. Introduction to Jade, 2017. Letzter Zugriff 20.03.2017.
- [Stru10] J. Struckmeier. Was ist mathematische Formulierung, 2010. Letzter Zugriff am 28.03.2017.
- [Umwe10] Umwelt-Bundesamt. Energieziel 2050: 100 Prozent Strom aus erneuerbaren Quellen. <http://www.umweltbundesamt.de/publikationen/energieziel-2050>, 2010. Letzter Zugriff am 29.03.2017.
- [Wetz05] T. Wetzlmaier. *Systematische Testfallgenerierung für den Black-Box-Test*. GRIN Verlag. 1. Auflage, 2005.

# Abkürzungen

COHDA	Combinatorial Optimization Heuristic for Distributed Agents
CSV	Comma-separated values
FIPA	Foundation for Intelligent Physical Agents
JADE	Java Agent Development Framework
LGPL	GNU Lesser General Public License
MS	Mittelspannung
NS	Niederspannung
Pacman	Power and Communication Management
PG	Projektgruppe
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
QM	Qualitätsmanagement
UML	Unified Modeling Language

---

# Anhang

---

## A. Validierungsergebnisse

## Validierungsergebnisse - Länge der NS-Stränge

Testfall Nr.	Länge der NS-Stränge	Anzahl Agenten	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DeathPenalty	Anzahl EverybodyOnBlacklist	Gesamtanzahl der Steps
			Gesamt	Ready for Cohda	WorkingMemory				
1	0	16	1.835	17	1.295	44	7	0	115
2	1	28	2.335	36	1.599	66	7	0	123
3	5	76	37.043	77	33.889	121	11	0	380
4	10	136	204.026	137	195.802	181	16	0	691
5	20	256	1.391.198	257	1.365.634	301	26	0	1.284

## Validierungsergebnisse - Anzahl der Agenten

Testfall Nr.	Anzahl der MS-Ringe	Anzahl der Agenten	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DeathPenalty	Anzahl EverybodyOnBlacklist	Gesamtanzahl der Steps
			Gesamt	Ready for Cohda	WorkingMemory				
1	1	40	6186	53	4656	90	10	0	203
2	2	81	12285	90	9532	167	16	0	204
3	4	164	86088	189	76806	383	32	0	373
4	9	372	201360	402	179316	854	67	0	464
5	16	664	697655	827	652933	1749	119	0	666

## Validierungsergebnisse - Anzahl der Sprecher

Testfall Nr.	Anzahl der Sprecher	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl Death Penalty	Anzahl EverybodyOnBlacklist	Gesamtanzahl der Steps
		Gesamt	Ready for Cohda	WorkingMemory				
1	1	11070	39	9740	37	6	0	240
2	2	10843	44	9387	56	8	0	234
3	3	8031	43	6792	79	8	0	233
4	4	4734	64	3290	99	12	0	183
5	5	10448	66	8696	127	12	0	242
6	6	10994	54	9291	113	13	0	238
7	7	8724	51	7222	118	13	0	190
8	8	8805	55	7271	146	12	0	210
9	9	7414	59	6146	138	8	0	268
10	10	7378	76	5912	164	8	0	193
11	11	7475	69	6038	139	7	0	196
12	12	6505	86	4946	165	8	0	195
13	13	7213	69	5856	131	7	2	180
14	14	6716	65	5463	116	4	0	166
15	15	5111	89	3652	162	7	2	168
16	16	7152	81	5786	152	2	0	188

### Validierungsergebnisse - Anteil flexibler Erzeuger und Verbraucher

Testfall Nr.	Anteil flex. Erz. + Vbr.	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DeathPenalty	Anzahl EverybodyOnBlacklist	Gesamtanzahl der Steps	
		Gesamt	Ready for Cohda	WorkingMemory					Askifiniland
1	1%	15600	44	13653	72	16	6	0	354
2	15%	18753	48	16869	96	13	3	0	407
3	50%	10208	56	8675	109	12	2	0	247
4	85%	8407	41	7144	84	8	0	0	199

## Validierungsergebnisse - Anzahl der Shortcuts

Testfall Nr.	Anzahl Shortcuts	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DeathPenalty	Anzahl EverybodyOnBlacklist	Anzahl Steps
		Gesamt	Ready for Cohda	WorkingMemory				
1	0	3591	40	2457	63	9	0	172
2	1	4897	39	3728	61	8	0	196
3	4	7026	41	5783	84	8	0	219
4	8	6865	58	5309	89	10	0	195
5	50	21686	70	19573	130	8	0	204
6	105	46191	85	43432	103	4	2	148

## Validierungsergebnisse - Gewichtung der Zielfunktion und Priorisierung

Nr.	Prio- risierung	Gewichte ZF	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DP	Anzahl EOB	Anzahl Steps	Step, der Aufnahme des Prio-Knotens	Agenten, die nicht aufgenommen wurden	
			Gesamt	Ready for Cohda	WM							Askifin-Island
1	2	0;0;1 (Priority)	636	11	475	11	8	3	2	132	49	3,5,6,7
2	2	0;1;0 (Islandsziele)	820	13	592	12	6	2	0	123	96	-
3	2	1;0;0 (Visibility)	795	17	646	17	6	2	2	133	-	2, 8, 9, 5, 6, 7
4	2	0;33;0,33;0,33	641	14	431	13	6	2	0	124	51	-
5	2	0;2;0;4;0,4	641	14	431	13	6	2	0	124	51	-
6	2	0;4;0;2;0,4	785	14	559	13	6	2	0	123	94	-
7	2	0;4;0;4;0,2	785	14	559	13	6	2	0	123	94	-
8	2	0;6;0;2;0,2	795	17	646	17	6	2	2	133	-	2, 8, 9, 5, 6, 7
9	2	0;2;0;6;0,2	849	13	621	12	6	2	0	123	96	-
10	2	0;2;0;2;0,6	636	11	475	11	8	3	2	132	49	3,5,6,7
11	5	0;0;1	636	11	475	11	8	3	2	132	49	3,5,6,7
12	5	0;1;0	820	13	592	12	6	2	0	123	96	-
13	5	1;0;0	795	17	646	17	6	2	2	133	-	2, 8, 9, 5, 6, 7
14	5	0;33;0,33;0,33	636	11	475	11	8	3	2	132	49	3,5,6,7
15	5	0;2;0;4;0,4	636	11	475	11	8	3	2	132	49	3,5,6,7
16	5	0;4;0;2;0,4	636	11	475	11	8	3	2	132	49	3,5,6,7
17	5	0;4;0;4;0,2	641	14	431	13	6	2	0	124	51	-
18	5	0;6;0;2;0,2	785	14	559	13	6	2	0	123	94	-
19	5	0;2;0;6;0,2	636	11	475	11	8	3	2	132	49	3,5,6,7
20	5	0;2;0;2;0,6	636	11	475	11	8	3	2	132	49	3,5,6,7
21	10	0;0;1	636	11	475	11	8	3	2	132	49	3,5,6,7
22	10	0;1;0	820	1	592	12	6	2	0	123	96	-
23	10	1;0;0	795	17	646	17	6	2	2	133	-	2, 8, 9, 5, 6, 7
24	10	0;33;0,33;0,33	636	11	475	11	8	3	2	132	49	3,5,6,7
25	10	0;2;0;4;0,4	636	11	475	11	8	3	2	132	49	3,5,6,7
26	10	0;4;0;2;0,4	636	11	475	11	8	3	2	132	49	3,5,6,7
27	10	0;4;0;4;0,2	636	11	475	11	8	3	2	132	49	3,5,6,7
28	10	0;6;0;2;0,2	636	11	475	11	8	3	2	132	49	3,5,6,7
29	10	0;2;0;6;0,2	636	11	475	11	8	3	2	132	49	3,5,6,7
30	10	0;2;0;2;0,6	636	11	475	11	8	3	2	132	49	3,5,6,7

## Validierungsergebnisse - Kombination Länge der NS-Stränge und Anzahl der Sprecher

Testfall Nr.	Anzahl Sprecher	Länge der NS-Stränge	Anzahl Agenten	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DeathPenalty	Anzahl EverybodyOnBlacklist	Gesamtanzahl der Steps
				Gesamt	Ready for Cohda	WorkingMemory				
1.1	1	0	16	1331	15	977	13	4	0	122
1.2	2	0	16	1344	16	938	22	5	0	123
1.3	4	0	16	1784	17	1244	44	7	0	103
1.4	8	0	16	550	30	34	90	7	0	60
1.5	11	0	16	577	40	25	90	4	0	46
1.6	16	0	16	596	57	0	110	0	0	55
2.1	1	1	28	3943	27	3149	25	5	0	181
2.2	2	1	28	3953	28	3093	36	6	0	180
2.3	4	1	28	4862	29	3848	65	8	0	159
2.4	8	1	28	2648	35	1780	102	9	0	161
2.5	11	1	28	2985	66	1899	149	6	0	149
2.6	16	1	28	2634	69	1660	140	1	0	126
3.1	1	5	76	43583	75	40069	73	9	0	413
3.2	2	5	76	43593	76	40013	84	10	0	412
3.3	4	5	76	40952	77	37307	116	12	0	387
3.4	8	5	76	29772	89	26553	171	15	0	372
3.5	11	5	76	27009	146	23186	247	14	0	391
3.6	16	5	76	26456	117	23338	188	5	0	345
4.1	1	10	136	204524	135	195450	133	14	0	693
4.2	2	10	136	204534	136	195394	144	15	0	692
4.3	4	10	136	202763	137	193678	176	17	0	675
4.4	8	10	136	153517	149	145338	231	20	0	647
4.5	11	10	136	176228	215	167174	322	20	0	694
4.6	16	10	136	140210	177	132252	248	10	0	619
5.1	1	20	256	1506309	255	1478915	253	24	0	1267
5.2	2	20	256	1506319	256	1478859	264	25	0	1266
5.3	4	20	256	1443528	257	1416363	296	27	0	1227
5.4	8	20	256	1227409	269	1202110	351	30	0	1204
5.5	11	20	256	1286612	335	1260058	442	30	0	1268
5.6	16	20	256	1152676	297	1127838	368	20	0	1166

## Validierungsergebnisse - Kombination Anzahl der Agenten und Anzahl der Sprecher

Testfall Nr.	Anzahl Sprecher (pro Ring)	Anzahl Agenten	Anzahl Nachrichten			Anzahl Verhandlungen	Anzahl DeathPenalty	Anzahl EverybodyOnBlacklist	Gesamtanzahl der Steps
			Gesamt	Ready for Cohda	WorkingMemory				
1.1	1	40	9173	39	7843	37	6	0	240
1.2	2	40	8997	44	7541	56	8	0	234
1.3	4	40	4069	64	2625	99	12	0	183
1.4	8	40	7435	55	5901	146	12	0	210
1.5	11	40	6356	69	4919	139	7	0	196
1.6	16	40	6145	81	4779	152	2	0	188
2.1	1	81	34803	79	31016	81	12	0	532
2.2	2	81	34674	85	30695	111	15	0	535
2.3	4	81	26571	99	22905	208	18	0	301
2.4	8	81	25296	92	21679	232	20	0	259
2.5	11	81	13780	154	10558	307	13	0	224
2.6	16	81	11859	169	8770	318	5	0	194
3.1	1	164	97216	171	88762	203	21	0	681
3.2	2	164	97409	176	88686	246	25	0	679
3.3	4	164	38563	199	30881	393	34	0	273
3.4	8	164	29670	239	22495	614	44	0	266
3.5	11	164	27365	371	19747	777	28	0	346
3.6	16	164	56695	366	47627	762	12	0	359
4.1	1	372	303814	418	279526	508	48	0	701
4.2	2	372	548981	415	519508	627	54	0	1158
4.3	4	372	325.416	481	301.246	1.009	67	0	647
4.4	8	372	305573	562	281364	1582	78	0	737
4.5	11	372	269024	648	244434	1545	63	0	687
4.6	16	372	101927	912	80179	1807	24	0	394
5.1	1	665	1367127	738	1314549	1011	76	0	1088
5.2	2	665	1241343	840	1188237	1398	87	0	1152
5.3	4	665	1305823	921	1250455	2095	123	0	1039
5.4	8	665	533246	910	493336	2450	132	0	753
5.5	11	665	866921	1005	823236	2799	96	1	1145
5.6	16	665	308803	1335	270129	3016	35	0	518

# B. Anwenderhandbuch

## Einleitung

Dieses Anwenderhandbuch soll es einem potentiellen Nutzer ermöglichen und ihn in die Lage versetzen, Pacman zu verwenden. Erklärt werden deshalb auf den folgenden Seiten die Leistungsumfänge sowie die Funktionalitäten des Programms, die Voraussetzungen, die für den Start erfüllt werden müssen und wie ein Anwender mit Ausnahme- beziehungsweise Fehlersituationen umgehen kann.

## Voraussetzungen

Eine zwingende Voraussetzung für das benutzen von Pacman ist, dass eine aktuelle Java-Version auf dem verwendeten Gerät installiert ist. Des Weiteren wird eine Entwicklungsumgebung, von der Projektgruppe wurde einheitlich Eclipse verwendet, benötigt. Wenn diese Voraussetzungen bereits erfüllt sind, muss das Pacman-Projekt nach Eclipse importiert werden. Zuletzt wird noch ein Netz benötigt, dass das Programm zu lösen hat. Das System erwartet, dass sich diese im Ordner data befinden, wie in [B.1](#) dargestellt ist.

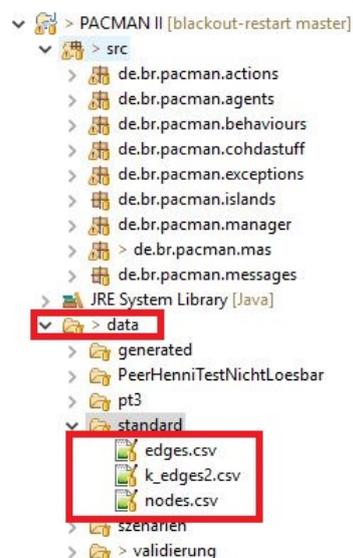


Abbildung B.1: Ordnerstruktur für Netze

```
private static String SAMPLES = "Testfall_15_Nodes.csv";
private static String S_EDGES = "Testfall_15_S_Edges.csv";
private static String K_EDGES = "Testfall_15_K_Edges.csv";

private static Printer printer = new Printer();
private static String FOLDER = "pt3";
```

Abbildung B.2: Auswahl Netze

## Eingabeparameter

Bei Pacman ist es möglich, verschiedene Eingabeparameter zu wählen und in einem gewissen Rahmen zu verändern. Einige Eingabeparameter verändern das Ergebnis, andere Parameter wirken sich lediglich auf die Laufzeit oder auch auf die Konsolenausgaben aus.

Bei dem wohl wichtigsten Eingabeparameter handelt es sich um das Netz, welches eingelesen werden soll, wie auf der Abbildung B.2 zu sehen ist. Hierfür müssen drei CSV-Dateien ausgewählt werden. Eine Datei für die Knoten, welche die Erzeuger und Verbraucher darstellen. In ihr wird jeder einzelne Knoten mit seinen Powerwerten, seiner Priorität und seinen möglichen aktivierbaren Nachbarn aufgezählt. Die zweite der drei CSV-Dateien ist für die Stromkanten. In ihr werden einfach alle vorhandenen Stromkanten aufgezählt. Die letzte der CSV-Dateien ist für die Kommunikationskanten, in der eben diese aufgezählt werden. Zu beachten ist bei den CSV-Dateien der Dateipfad. Er lässt sich über die Variable Folder anpassen. Alle drei CSV-Dateien müssen sich zwingend im selben Ordner befinden.

Beim zweiten wichtigen Eingabeparameter handelt es sich um den Planungshorizont. Dieser muss auf die Anzahl der Zeitschritte, die in der Knoten-CSV-Datei vorhanden sind, angepasst werden. Erfolgt dies nicht, kommt es beim Ablauf des Programms gezwungenermaßen zu Fehlern.

Die nächsten Eingabeparameter sind die Gewichte, mit denen die Optimierungsziele versehen werden sollen. Diese sollten in der Summe eins ergeben und können auch direkte Auswirkungen auf das Ergebnis haben. Wenn die Hinzunahme von Knoten mit hoher Priorität durch die Gewichte wichtiger ist, als eine erhöhte Sichtbarkeit, dann werden Knoten mit höherer Priorität bei entsprechender Priorisierung der Zielfunktion früher mit in eine Insel mit aufgenommen.

Der letzte Eingabeparameter, der sich auf direkt auf das Ergebnis auswirkt, sind die Sprecher, die im Netz vorhanden sein sollen. In B.3 abgebildeter Zeile werden die IDs der Sprecher eingetragen. Die ID der jeweiligen Sprecher leitet sich aus der Knoten-CSV-Datei ab.

Der erste Eingabeparameter, der nicht mit dem Ergebnis, aber mit der Laufzeit zu tun hat, ist der Garbagecollector. Sobald dieser angeschaltet wird, wird verhindert,

```
private static int[] speaker = new int[]{0,1};
```

Abbildung B.3: Auswahl Sprecher

```
private boolean SHOW_SOLUTIONCANDIDATE_OUTPUT = false;
private boolean SHOW_WORKINGMEMORY_OUTPUT = false;
private boolean SHOW_SWITCHSTATE_OUTPUT = false;
private boolean SHOW_SWITCHCOMMAND_OUTPUT = false;
private boolean SHOW_SENDMESSAGE_OUTPUT = false;
private boolean SHOW_RECEIVEMESSAGE_OUTPUT = false;
private boolean SHOW_SCENARIO = false;
private boolean SHOW_ISLANDS = true;
private boolean SHOW_AGENTS = false;
private boolean SHOW_WRONGBEHAVIOUR_OUTPUT = false;
```

Abbildung B.4: Auswahl Ausgaben

dass der Speicher bei zu großen Netzen überläuft und es bleibt steht auf lange Sicht mehr Arbeitsspeicher zur Verfügung, da dieser immer wieder bereinigt wird.

Bei den letzten Eingabeparametern handelt es sich um die Konsolenausgaben, dargestellt in Abbildung B.4. Über den Printer kann der Anwender entscheiden, welche Konsolenausgaben er angezeigt haben möchte. Dies ist überaus sinnvoll, wenn ein Netz mit vielen Agenten gelöst werden soll, da man sich jeden Agenten in jedem Step und jede Insel anzeigen lassen könnte. Bei einer hohen Anzahl an Agenten führt die Anzeige dieser schnell zu einer sehr unübersichtlichen Ausgabe.

## Erstellung von eigenen Netzen

Selbstverständlich ist es auch möglich, eigene Netze zu erstellen. Hierzu ist es notwendig, eine CSV-Datei für die Knoten, eine CSV-Datei für die Stromkanten sowie eine für die Kommunikationskanten zu erzeugen. Es empfiehlt sich, das gewünschte Netz vorher zu skizzieren.

Nach Skizzierung des Netzes sollten die Knoten in der CSV-Datei erstellt werden. Hierfür bekommt jeder Knoten eine eigene Zeile. Zuerst werden die möglichen Powerwerte eines Zeitschrittes durch Kommata voneinander getrennt, notiert. Wenn die Powerwerte für einen Zeitschritt notiert wurden, wird der Zeitschritt vom nächsten Zeitschritt durch ein Semicolon getrennt. Nun werden die Powerwerte des nächsten Zeitschritts eingetragen, durch Semicolon vom nächsten Zeitschritt getrennt, bis alle Zeitschritte notiert sind. Zu beachten ist hierbei, dass alle Knoten die selbe Anzahl an Zeitschritten haben müssen. Für flexible oder statische Verbraucher und Erzeuger werden in jedem Zeitschritt einfach die selben Powerwerte eingetragen.

Wenn die Zeitschritte eines Knotens eingetragen sind, folgt eine Raute (#). Gefolgt hiervon werden die IDs der aktivierbaren Nachbarn des Knotens, getrennt durch Kommata, eingetragen. Wenn diese notiert sind, folgt eine weitere Raute. Sofern

der Knoten über keine aktivierbaren Nachbarn verfügt, folgt die Raute sofort. Als letztes wird in jeder Zeile die Priorität des jeweiligen Knotens eingetragen. Die Priorität eines Knoten wird als natürliche Zahl angegeben, die Powerwerte können reelle Zahlen annehmen.

Beispiel:

```
1,2,4,5;1,2,4,5;1,2,4,5#15,16,17#2
-2;-2;-2#2,3,5#5
1;2;3;##1
```

Die IDs der Knoten werden automatisch nach der jeweiligen Zeile in der Knoten-CSV-Datei vergeben. Das heißt, der erste Knoten in der Datei bekommt die ID 0, der zweite Knoten die 1, der dritte die 2 und so fort.

Die Strom- und Kommunikationskanten werden analog zueinander erstellt. Dafür werden in einer Zeile jeweils die IDs von den zwei zu verbindenden Knoten durch ein Komma getrennt in eine Zeile geschrieben.

Beispiel:

```
12,15
12,16
13,14
```

Es ist nicht notwendig, Kanten doppelt zu notieren. Falls dies doch geschieht, werden die doppelten Kanten entfernt. Zu beachten ist dabei, dass der erste Agent die ID 0 hat. Dies ist einer der häufigen Fehler, die bei der Erstellung von Netzen entstehen.

Wenn man die Dateien erstellt hat, sollte man sie sinnvoll benennen. Empfehlenswerte Endungen für die Dateien wären:

```
_nodes.csv
_k_edges.csv
_s_edges.csv
```

Bei der Skizzierung des Netzes sollte man sich auch überlegt haben, welche Agenten die Sprecher sind. Es empfiehlt sich, die IDs dieser in einer zu den CSV-Dateien zugehörigen Textdatei einzutragen, damit man sie bei Eintragung der CSV-Dateien in das System direkt in die Zeile für die Sprecher einfügen kann.

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 80, Size: 14
    at java.util.ArrayList.rangeCheck(Unknown Source)
    at java.util.ArrayList.get(Unknown Source)
    at de.br.pacman.mas.Main.connectSEdges(Main.java:166)
    at de.br.pacman.mas.Main.main(Main.java:78)
```

Abbildung B.5: Fehler in einer der Dateien für die Kanten

## Fehlererkennung

Insgesamt gibt es drei große Möglichkeiten, wo der Anwender selbst Fehler machen kann, sodass das ihm vorliegende Ergebnis von dem erhofften beziehungsweise dem von ihm erwarteten Ergebnis abweicht.

Die erste Möglichkeit liegt bei dem Entwurf des Netzes selbst. Hier könnte der Anwender sich verrechnet und falsche Powerwerte gewählt haben, sodass das Netz nicht lösbar ist, sowie er es sich vielleicht vorgestellt hat, oder er könnte vergessen haben, Kanten zu zeichnen, sodass das Netzdesign unlösbar wurde. Diese Möglichkeit soll im Rahmen des Anwenderhandbuchs nicht weiter betrachtet werden.

Die zweite Möglichkeit liegt bei der Erstellung der CSV-Dateien. Hier könnte der Anwender beispielsweise einen Zahlendreher einbauen, in einer Zeile verrutschen oder einfach eine Ziffer zu viel eintragen, sodass ein Knoten angesprochen werden soll, den es in Wirklichkeit gar nicht gibt. Die Ausgabe dieses Fehlers ist in [Abbildung B.5](#) dargestellt. Hier wurde bei einem Netz mit 14 Agenten anstatt einer 8 eine 80 eingetragen. Logischerweise kann dieser Agent nicht gefunden werden, was zu einer Index-Out-Of-Bound-Exception führt. Die gleiche Fehlermeldung gäbe es, wenn eine falsche Ziffer in der Datei für die Kommunikationskanten eingetragen wird. Sollte bei den aktivierbaren Nachbarn dieser Fehler gemacht werden, so gibt es keine Fehlermeldung. Allerdings wird man am Ende der Laufzeit des Systems unweigerlich sehen, dass in der Kommunikationsnachbarschaft noch Agenten vorhanden sind, welche in keine Insel aufgenommen wurden.

Die dritte Möglichkeit liegt bei der Eintragung der CSV-Dateien in die Main-Methode des Systems. Hier können ganz verschiedene Fehler gemacht werden. Die einfachsten Fehler wären, dass die CSV-Dateien durch einen Rechtschreibfehler falsch benannt sind, zu sehen in [B.6](#), der Pfad nicht korrekt ist, zu sehen in [B.7](#) oder die Dateien richtig benannt sind, aber nicht in den entsprechenden Ordner kopiert wurden. Das erwartete Ergebnis kann auch dadurch vom vorliegenden Ergebnis abweichen, dass die Gewichte falsch gewählt wurden, beziehungsweise vergessen wurde, diese anzupassen. Auch möglich ist, dass vergessen wurde, den Planungshorizont auf das aktuelle Szenario anzupassen, wie in [B.8](#), oder nicht die korrekten Sprecher gewählt wurden.

```
Exception in thread "main" java.lang.NullPointerException
    at de.br.pacman.mas.Main.connectSEdges(Main.java:165)
    at de.br.pacman.mas.Main.main(Main.java:78)
```

Abbildung B.6: Eine Datei wurde falsch benannt

```
Exception in thread "main" java.lang.NullPointerException
    at de.br.pacman.manager.CsvManager.readSamples(CsvManager.java:127)
    at de.br.pacman.mas.Main.main(Main.java:53)
```

Abbildung B.7: Der Ordnerpfad wurde falsch benannt

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 2, Size: 2
    at java.util.ArrayList.rangeCheck(Unknown Source)
    at java.util.ArrayList.get(Unknown Source)
    at de.br.pacman.agents.PacmanAgent.checkPower(PacmanAgent.java:270)
    at de.br.pacman.agents.PacmanAgent.chooseNegotiationPeer(PacmanAgent.java:221)
    at de.br.pacman.actions.OptimizeAction.nextCohdaRound(OptimizeAction.java:387)
    at de.br.pacman.actions.OverlayAction.manageOverlayNet(OverlayAction.java:105)
    at de.br.pacman.behaviours.OverlayBehaviour.getMessages(OverlayBehaviour.java:40)
    at de.br.pacman.behaviours.OverlayBehaviour.<init>(OverlayBehaviour.java:21)
    at de.br.pacman.agents.PacmanAgent.checkState(PacmanAgent.java:158)
    at de.br.pacman.agents.PacmanAgent.step(PacmanAgent.java:95)
    at de.br.pacman.mas.MiniMas.execute(MiniMas.java:116)
    at de.br.pacman.mas.Main.main(Main.java:121)
```

Abbildung B.8: Der Planungshorizont ist falsch

## C. Programmierkonventionen

# Programmierkonventionen der PG Blackout Restart

Grundlegend für die Softwareentwicklung dieser Projektgruppe sind die Folgenden Prinzipien der Objektorientierten Entwicklung.

## 1. *Single Responsibility Principle*

- Harte Aufteilung in Logische Zusammenpassende Module: **Actions** (Kohäsion maximieren und Kopplung minimieren)
- Code mit unklaren Aufgaben wird entfernt
- **Jedes Modul dient einem Zweck**

## 2. *Don't repeat yourself*

- Granulare Methoden: dabei Abwägung zwischen Wiederverwendbarkeit und Komplexität

## 3. *Open-Closed-Principle*

- Anpassbarkeit ohne Große Veränderungen eines Moduls
- Module können durch erweiterungspunkte neue Funktionalitäten bereitstellen

Um die Prototypenentwicklung zu Überwachen und das Verteilte Arbeiten zu erleichtern wird die Versionsverwaltungssoftware git verwendet. Dabei wird im Rahmen des Aufgabenmanagements jeder Branch in Abhängigkeit eines Jira-tickets gestellt. Der Name eines spezifischen Branchs ist demnach äquivalent mit der ID des dazugehörigen Jira-tickets.

Für die Code-Dokumentation verlassen wir uns auf die Grundlegenden Features von Javadoc.

Um Zusätzlich für Konsistenz in der Entwicklung zu sorgen werden im Folgenden Benennungen und Abläufe definiert. Diese Sind nur als Erweiterung und Hinweise auf gängige Programmierkonventionen zu betrachten.

### **Dateien und Packages:**

- 1) Jede Klasse wird in einer eigenen Datei implementiert.
- 2) Jede Datei gehört zu einem logischen Package.
- 3) Packages werden nach ihrer logischen Identität benannt. Zum Beispiel: de.br.pacman.models oder de.br.pacman.agents.

### **Klassen und Methoden:**

- 1) Es wird möglichst granular entwickelt. Das heißt, es werden möglichst logisch zusammenhängende Blöcke in Methoden gekapselt um diese dadurch wiederverwendbar zu machen und die Übersicht zu verbessern. Daraus folgt das kein Code doppelt geschrieben werden muss. **Don't repeat yourself**
- 2) Klassen und Methoden werden nach ihrem Verwendungszweck benannt!
- 3) Es werden Javadoc Kommentare über wichtigen Funktionen, aber jeder Klasse verwendet. Triviale Funktionen wie Getter und Setter sind hiervon ausgeschlossen.

- 4) Zusätzlich werden Code Änderungen wie folgt Kommentiert. Dabei wird der ursprüngliche Code auskommentiert und als Kopie behalten.

```
/*MODBEG TD-BR-123: Kommentar */
```

```
//old stuff
```

```
New Stuff
```

```
/*MODEND TD-BR-123: Kommentar */
```

- 5) Jede wichtige Klasse erhält eine toString Methode. Dadurch werden die Consolen-Ausgaben deutlich einfacher.

### Logik und Fehlervermeidung

- 1) Um Fehler vorzubeugen sind in jeder Methode mit Parametern null Prüfungen zu implementieren.
- 2) Dabei wird bei unerwarteten Eingaben ein Fehlerhandling erfolgen (Entweder initialwert einbauen, Abbruch der Methode oder Exceptions!)

### Git und Branches

- 1) Jede Aufgabe wird in eigenem Branch entwickeln. Dazu im Jira Aufgabe aussuchen, Branch erstellen und ihm den Namen des Tickets geben.
- 2) Das Erzeugen eines Branches erfolgt immer vom aktuellen Master
- 3) Da sich Änderungen im Master ergeben haben können, wird vor Abschluss einer Aufgabe der aktuelle Master in den eigenen Branch gemerged.
- 4) Es wird nur vom Repositoriums-beauftragtem im Master gearbeitet.
- 5) Falls Dateien hinzugefügt werden, zum Beispiel libraries, dann werden diese im lib Ordner hinzugefügt.
- 6) Bei Abschluss einer Aufgabe und Fertigstellung wird sowohl der Jira-task auf Abgeschlossen gesetzt und zusätzlich dem Repositoriums-beauftragtem eine Nachricht gesendet.

### Git Merge:

Der Repositoriums-beauftragte Merged am Freitag die fertiggestellten Branches. Dabei ist es seine Aufgabe, den Code zu sichten, und gegebenenfalls den Merge abzulehnen. Dann hat dieser Kommentare mit der Begründung jener Ablehnung zu schreiben.

Ablehnungen dienen ausschließlich der Gewährleistung der Funktionalität des Systems und Einhaltung dieser Konventionen. Die Kommentare sind in Folgender Form in dem abgelehnten Branch zu finden:

```
//QM-TD: Kommentar
```

Des Weiteren ist es nicht die Aufgabe des Repositoriums-beauftragten die einzelnen Branches auf den aktuellen Masterstand zu Mergen. Dies ist die Aufgabe der Programmierer.

Bei großen Systemänderungen und Bugfixes darf der Repositoriums-beauftragte jederzeit den Master aktualisieren. Hier ist aber von den Erstellern des Codes eine Statusmail über den internen Verteiler zu tätigen. In dieser muss erklärt werden warum die Veränderungen notwendig sind.



## D. Abbildungen

# Projektplan

Hervorgehobener Zeitraum: 30

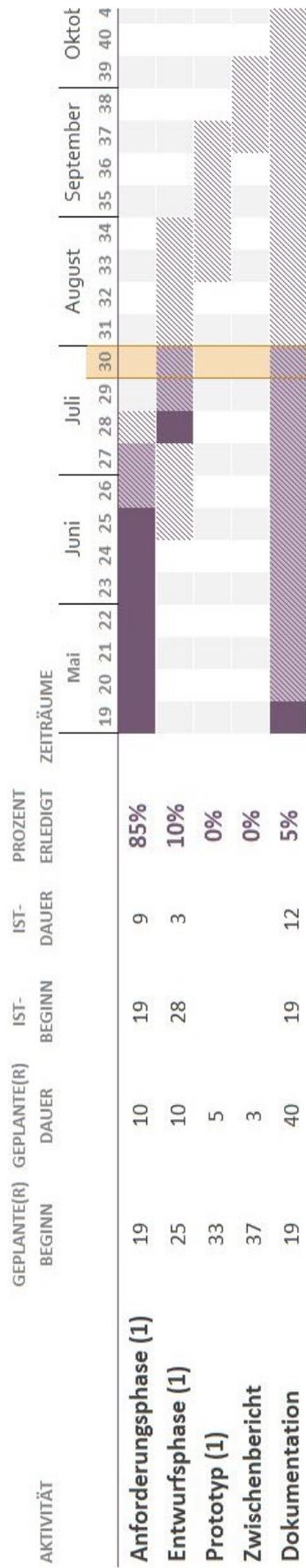


Abbildung D.1: Zeitplan des ersten Semesters

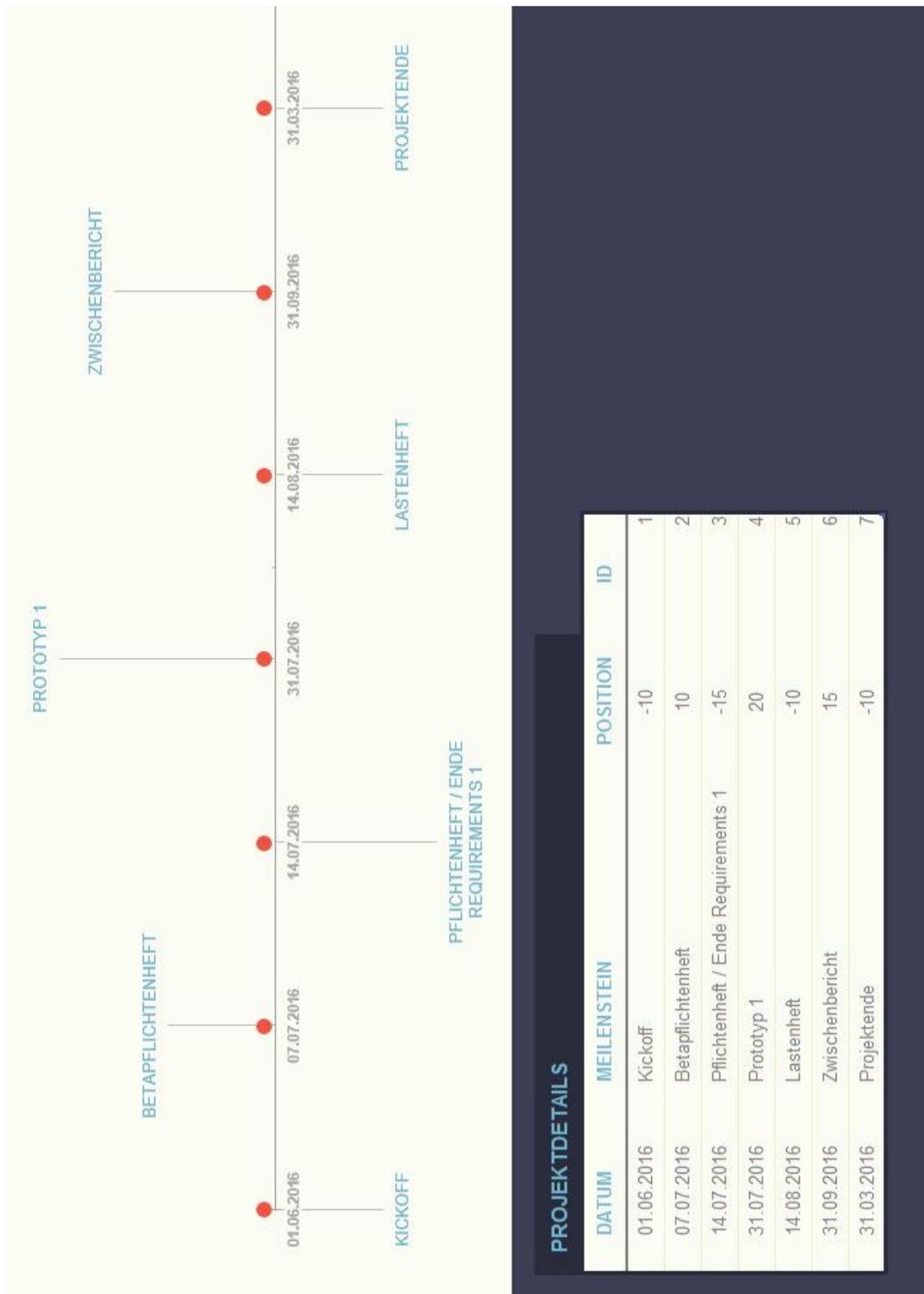


Abbildung D.2: Meilensteine des ersten Semesters



## **E. Seminarband**

---

# Seminarband der Projektgruppe Blackout - Restart

Sommersemester 2016

---

# Inhaltsverzeichnis

<b>1</b>	<b>Projektmanagement</b> .....	1
	Daniel Palm	
1.1	Grundlagen des Projektmanagements .....	1
1.2	Projektphasen .....	5
1.3	Vorgehensmodelle zur Projektdurchführung .....	8
1.4	Fazit .....	12
	<b>Literaturverzeichnis</b> .....	14
<b>2</b>	<b>Requirements Engineering &amp; SGAM</b> .....	15
	Hendrik Buhl	
2.1	Einleitung .....	15
2.2	Requirements Engineering .....	16
2.3	SGAM .....	22
2.4	Fazit .....	28
	<b>Literaturverzeichnis</b> .....	30
<b>3</b>	<b>COHDA - A Combinatorial Optimization Heuristic for Distributed Agents</b> .....	31
	Stephanie Alexandra Warsch	
3.1	Einleitung .....	31
3.2	Problemstellung und Motivation .....	32
3.3	Grundlagen von COHDA .....	33
3.4	Ablauf der Konsensbildung .....	36
3.5	Ausblick auf die Projektgruppe .....	40
	<b>Literaturverzeichnis</b> .....	42
<b>4</b>	<b>Kommunikationsnetze/-kanäle/-simulation</b> .....	43
	Erik Hogen	
4.1	Historische Entwicklung von Kommunikationsnetzen .....	43

Inhaltsverzeichnis	iii
4.2    Kommunikationsnetze heute	44
4.3    Kommunikation im Smart Grid	49
4.4    Simulation von Kommunikationsnetzen	51
<b>Literaturverzeichnis</b>	56
<b>5    Stabiler Netzbetrieb und Wiederaufbau</b>	59
Sanja Stark	
5.1    Grundlagen	59
5.2    Stromnetze von heute	63
5.3    Stromnetze von morgen	71
5.4    Fazit	75
<b>Literaturverzeichnis</b>	76
<b>6    Kommunikationsnetze bei Großstörungen</b>	78
Peer Vincent Stubbe	
6.1    Einleitung	78
6.2    Kommunikationsnetze in Deutschland	79
6.3    Aufbau der Kommunikationsnetze	80
6.4    Großstörungen	83
6.5    Fazit	88
<b>Literaturverzeichnis</b>	89
<b>7    Modellierung und Simulation von Smart-Grids</b>	90
Maximilian Kreutz	
7.1    Einführung	90
7.2    Darstellung der Problemstellung und Begriffsklärung	91
7.3    Simulationssoftware für Smart-Grids	95
7.4    Fazit	101
<b>Literaturverzeichnis</b>	102
<b>8    Verteilte Berechnung von minimalen Spannbäumen</b>	103
Tim Dombrowski	
8.1    Einleitung	103
8.2    Minimale Spannbäume	104
8.3    Berechnung minimaler Spannbäumen	105
8.4    Wiederaufbau eines Stromnetzes	114
8.5    Fazit	115
<b>Literaturverzeichnis</b>	116



# Kapitel 1

## Projektmanagement

Daniel Palm

**Zusammenfassung** In der vorliegenden Seminararbeit beschäftigt sich der Autor dieser Arbeit im Rahmen des Kernmoduls “Projektgruppe” und der daraus im SS2016 entstandenen Projektgruppe Blackout/Restart innerhalb der Seminarphase mit dem Thema des Projektmanagements. Ziel der Arbeit ist es, dem Leser einen kurzen Einblick über die Grundzüge und Vorgehensmodelle des Projektmanagements zu gewährleisten und im Speziellen näher auf ausgewählte Vorgehensmodelle in Anbetracht der Projektdurchführung der PG Blackout/Restart einzugehen, um als optimale Vorbereitung für den Beginn der Projektphase zu dienen. Ferner soll das aus dieser Arbeit gewonnene Wissen an die Kommilitonen der Projektgruppe vermittelt werden und einen Beitrag zur Grundlagenbildung innerhalb der Seminarphase liefern. Da das Thema des Projektmanagements sehr weitläufig ist, wurde der Fokus im Rahmen dieser Arbeit auf ausgewählte Themengebiete gesetzt. Im ersten Kapitel werden einige Grundlagen des Projektmanagements dargelegt. Dazu werden die Begrifflichkeiten ‘Projekt’ und ‘Projektmanagement’ erläutert und ferner auf Hauptaufgaben und Rollen des Projektmanagements eingegangen. Nachdem im zweiten Kapitel auf die klassischen Projektphasen näher eingegangen wird, behandelt das dritte Kapitel allgemeine Vorgehensmodelle zur Projektdurchführung in der Softwareentwicklung. Anschließend werden in Bezug auf die Umsetzung der Projektgruppe Blackout/Restart einige konkret ausgewählte Modelle in ihren Grundzügen vorgestellt.

### 1.1 Grundlagen des Projektmanagements

Das nachstehende Kapitel befasst sich mit den Grundzügen des Projektmanagements in seiner theoretischen Form.

---

Carl von Ossietzky Universität Oldenburg  
E-mail: [daniel.palm@uni-oldenburg.de](mailto:daniel.palm@uni-oldenburg.de)

### **1.1.1 Projekt(management)**

Bisweilen hat sich keine einheitliche Definition für Projekt durchgesetzt. Laut DIN-Norm 69901 ist ein Projekt „ein Vorhaben, das im Wesentlichen durch die Einmaligkeit seiner Bedingungen in ihrer Gesamtheit gekennzeichnet ist, z.B. Zielvorgabe, zeitliche, finanzielle, personelle und andere Begrenzungen, Abgrenzung gegenüber anderen Vorhaben, projektspezifische Organisation“. Laut PMI hingegen ist ein Projekt „eine zeitlich beschränkte Anstrengung zur Erzeugung eines einmaligen Produktes, Dienstes oder Ergebnisses“. Gemeinsam haben alle Definitionen, dass sie eine unbestimmte Anzahl von Kriterien bzw. Eigenschaften von Projekten beinhalten. So können unter anderem folgende Kriterien für Projekte in den verschiedenen Definitionen aufgeführt sein[6].

- Zielvorgabe
- Einmaligkeit
- Endlichkeit
- Projektteam
- Bedeutung
- Komplexität
- Umfang
- Interdisziplinarität

Unter Projektmanagement versteht man allgemein das *Management* von Projekten, das erforderlich ist, um Projekte in einer bestimmten Zeit mit bestimmten Ressourcen zu einem bestimmten Ergebnis zu bringen. Die DIN 69901 definiert Projektmanagement daher als Gesamtheit von Führungsaufgaben, -organisation, -techniken, -mitteln für die Abwicklung eines Projektes.

Zusammengefasst lässt sich festhalten, dass Projektmanagement das Planen, Steuern und Überwachen von Projekten zur Aufgabe innehat.

### **1.1.2 Projektorganisation**

Die Projektorganisation setzt sich mit den Fragen auseinander, wie zeitlich befristete Projekte in die bestehende Organisationsstruktur von Unternehmen eingebettet werden können und wie diese Projekte intern zu strukturieren sind. Die meisten Definitionen um Projektorganisation heben die Abwicklung eines Projektes entsprechend des Entwicklungsprozesses (Ablauforganisation) und die Einbindung des Projektes in die Unternehmensstruktur (Aufbauorganisation) hervor[4]. Je nach Projektgröße und dem Grad der Bereichsüberschreitung der beteiligten Mitarbeiter wird zwischen drei Grundtypen von Organisationsformen für die dauerhafte Einbettung eines Projektes in die Unternehmensstruktur unterschieden. Dazu zählen Linien-, Stab- und Matrix-Projektorganisation. Die Linienorganisation im Projektmanagement definiert die anfallenden Aufgaben einer Abteilung in Form eines Projektes. Es besteht im Gegenzug dazu, keine eigenständige Projektorganisation, sondern ein

ausgewählter Projektleiter inklusive Zeit- und Kostenplan für das Projekt. Bei der Matrixorganisation wird typischerweise nur die Projektleitung und ggf. ein kleines Team aus der Linienstruktur herausgelöst. Alle anderen Projektbeteiligten werden dem Projekt nur zeitlich beschränkt und mit Kapazitätsanteilen zugeordnet. Bei der Staborganisation spielt der Projektleiter die Funktion eines Stabes, d.h. er führt lediglich koordinierende und beratende Tätigkeiten aus und hat keinerlei Führungs- und Weisungsbefugnisse. Dem Projekt wird darüber hinaus keine eigene Organisationseinheit zugeschrieben, sodass die Projektbeteiligten in ihren üblichen Bereichen verbleiben. Der Unterschied zur Matrixorganisation besteht darin, dass Projektbeteiligte in der Matrixorganisation unter der Führung zweier Vorgesetzte (Projektleiter und Vorgesetzter des Fachbereichs) stehen.

Da die Projektgruppe Blackout/Restart keine Projektorganisation im Sinne einer Aufbauorganisation vornehmen muss, wird auf eine ausführlichere Beschreibung der verschiedenen Organisationsformen verzichtet.

Des Weiteren stellt ein wesentlicher Bestandteil der Projektorganisation die Benennung und Beschreibung von Rollen mit jeweiligen Verantwortungen und Befugnissen dar. Zu den wichtigsten Rollen innerhalb der Projektorganisation zählen Projektausschuss, Auftraggeber, Anwender, Projektleiter und Projektmitarbeiter.

### ***1.1.3 Aufgaben des Projektmanagements***

Einen international anerkannten Standard bezüglich der Aufgaben bzw. Tätigkeiten des Projektmanagements stellen die sogenannten neun Wissensfelder des Projektmanagements (s. Abb. 1.1) dar, die als Projektmanagementdisziplinen vom PMI 2013 aufgestellt worden sind. Die jeweiligen Tätigkeiten im Projektmanagement sind auf diese Inhalte ausgerichtet und können dabei mehrere Wissensbereiche gleichzeitig betreffen.

Das Integrationsmanagement im Projektmanagement beschreibt die Prozesse, die für eine gute Koordinierung und Integration der unterschiedlichen Aktivitäten eines Projekts erforderlich sind. Es umfasst die Projektplanentwicklung, die Projektplandurchführung und das Änderungswesen. Das Projektumfangsmanagement befasst sich mit der laufenden Planung und Kontrolle des Leistungsfortschritts im Projekt. Im Rahmen des Umfangsmanagements wird in regelmäßigen Abständen überprüft, ob sich das Projekt „auf dem richtigen Weg“ befindet. Zum Projektumfangsmanagement gehören die Projektinitiierung, die Inhalts- und Umfangsplanung, die Leistungsdefinition, die Leistungsverifizierung sowie die Leistungsüberwachung. Das Zeit- und Terminmanagement soll sicherstellen, dass ein Projekt termingerecht fertiggestellt wird. Zum Zeit- und Terminmanagement gehören Vorgangsdefinition, Festlegung der Vorgangsfolgen, Vorgangsdauerschätzung, Terminplanentwicklung und Terminplanüberwachung. Das Kostenmanagement beschreibt alle erforderlichen Prozesse, die sicherstellen sollen, dass das Projekt im geplanten und genehmigten Kostenrahmen fertiggestellt wird. Zum Kostenmanagement gehören Einsatzmittelplanung, Kostenschätzung, Budgetierung sowie Kostenüberwachung. Das Qualitäts-

management in Projekten soll sicherstellen, dass die vom Auftraggeber definierten Qualitätsansprüche eingehalten oder sogar übertroffen werden. Dazu gehören Qualitätsplanung, Qualitätssicherung und Qualitätslenkung. Die Hauptaufgabe des Personalmanagements ist es, dafür zu sorgen, dass die am Projekt beteiligten Mitarbeiter so effizient wie möglich eingesetzt werden. Dem Personalmanagement können folgende Funktionen und Aufgaben zugeordnet werden: Projektorganisation, Personalakquisition und Teamentwicklung. Das Kommunikationsmanagement im Projekt hat zum Ziel, sämtliche Projektinformationen rechtzeitig und angemessen zu erstellen, zu sammeln, zu verbreiten, abzulegen sowie zu definieren. Hierzu gehören der Aufbau eines Informations- und Berichtswesens, die Informationsverteilung, die Fortschrittsermittlung sowie der administrative Abschluss. Das Risikomanagement umfasst alle Prozesse, die sich mit der Durchführung der Risikomanagementplanung sowie der Überwachung und Identifizierung von Projektrisiken befassen. Darüber hinaus erfolgt bei der Identifikation von Projektrisiken die Einleitung entsprechender Gegenmaßnahmen. Das Wissensfeld Beschaffungsmanagement beinhaltet die Beschaffung von Waren und Leistungen außerhalb der Organisation sowie die dazugehörige Vertragsgestaltung. In diesen Bereich fallen Beschaffungsvorbereitung, Angebotsvorbereitung, Einholen von Angeboten, Lieferantenauswahl, Vertragsgestaltung und Vertragserfüllung.



**Abb. 1.1** Die neun Wissensfelder des Projektmanagements[4]

## 1.2 Projektphasen

Nachdem nun ausgewählte Grundlagen des Projektmanagements dargestellt worden sind, beschäftigt sich dieses Kapitel mit den klassischen Projektphasen.

Nach Kessler/Winkelhofer ist eine Projektphase ein definierter Abschnitt im Projekt mit einem wichtigen Teilergebnis.[6] Jede Projektphase hat dabei ein definiertes Ziel im Projektablauf und eine bestimmte Anzahl von Arbeitsschritten. Die klassische Gliederung der Projektaktivitäten in Phasen ist in Reinform eine Vorgehensweise nach dem Wasserfallmodell, kann jedoch auch iterativ angelegt sein, damit z. B. bestimmte Projektergebnisse nochmals überarbeitet werden können bzw. auf sich ändernde Anforderungen eingegangen werden kann.

Die Literatur (s.[6], [4]) stellt ein allgemeines Phasenmodell, bestehend aus Initialisierungs-, Planungs-, Durchführungs- und Abschlussphase, dar, das im Folgenden näher erläutert werden soll.

### 1.2.1 Initiierungsphase

Die Initiierungsphase stellt die erste Projektphase dar und bildet die Grundlage eines Projektes. In ihr werden Vorgaben getroffen, die für die nächste Phase, der Planungsphase, als verbindlich gelten. Die Formalisierung der Initiierungsphase erfolgt von Unternehmen zu Unternehmen unterschiedlich und obliegt der entsprechenden Projektkultur. Dennoch kann die Initiierungsphase als wichtige Voraussetzung für die erfolgreiche Projektdurchführung gesehen werden. In ihr sind die Ziele des Projektes zu klären, um darauf aufbauend eine erste Grobplanung durchführen zu können. Allgemein lassen sich die Aktivitäten dieser Phase auf folgende reduzieren:

- Gründung des Projektes
- Definition des Projektziels
- Organisation des Projektes
- Organisation des Prozesses

Der erste formale Schritt zur Gründung eines Projektes stellt der Projektantrag dar, den es auszuarbeiten gilt. Er gilt zur Klärung der Frage 'Was soll genehmigt werden?' und enthält alle projektrelevanten Informationen. Dazu zählen u.a. Projektbegründung, -beschreibung, -verantwortungen, -ressourcen und Zielerreichungsstrategie. Mit der Genehmigung wird der Projektantrag offiziell zum Projektauftrag. Durch den Projektauftrag werden mehrere Vereinbarungen zwischen Auftraggeber und Projektleiter getroffen. Dazu zählen Projektziel, -nutzen, -laufzeit, Meilensteine sowie sämtliche Zwischentermine, Messgrößen des Projekterfolges, Art und Weise des Berichtwesens, Anforderungen an die Projektdokumentation, Projektbudget und Vollmachten und Weisungsbefugnisse des Projektleiters.[6] Der Projektauftrag stellt ein fester Bestandteil des Projektes dar, an denen sich alle Projektbeteiligte im weiteren Verlauf des Projektes orientieren können. Neben der eindeutigen Definition

des Projektziels erfolgt zusammen mit dem Auftraggeber die Erstellung des Lastenhefts. Aus diesem geht letztendlich hervor, was das Projektteam unter welchen Rahmenbedingungen schaffen soll. Dabei schafft die Projektleitung die für das Projekt notwendige Infrastruktur, wobei der Auftraggeber die Gremien im Umfeld des Projektes einrichtet. Zuletzt gilt es, organisatorische Rahmenbedingungen für das Projekt zu klären. Hier erfolgt typischerweise eine Rollenverteilung und die Festlegung einer Organisationsstruktur (s. Abb. 1.1.2).

### ***1.2.2 Planungsphase***

Üblicherweise startet die Planungsphase mit dem sogenannten Kick-off-Meeting, welches die erste gemeinsame Besprechung aller Projektbeteiligter darstellt. Es werden das Projekt, die Teams und Rahmenbedingungen (Ziele, Kosten, Zeitplanung) vorgestellt.

Als wichtigster Schritt der Planungsphase lässt sich die Erstellung des Projektstrukturplans (PSP) festhalten. Dieser beinhaltet alle Teilaufgaben und Arbeitspakete des gesamten Projektes und stellt die Grundlage für die weitere Termin- und Ablaufplanung, die Ressourceneinplanung und die Kostenplanung dar. Da der PSP als Grundlage für das Projekt angesehen werden kann, wird er auch als „Plan der Pläne“<sup>[9]</sup> bezeichnet. Steht zunächst der PSP, müssen im Anschluss die Arbeitspakete definiert werden. Ein Arbeitspaket ist dabei ein planbares Element, das nicht weiter untergliedert werden kann, und eindeutige Bedeutung und Verantwortlichkeit zugeschrieben wird. Inhalte werden in der Arbeitspaketbeschreibung festgehalten und umfassen u.a. allgemeine Projektinformationen (wie z.B. Projekt, Projektleiter), Beschreibung (Zweck und Inhalt), Ziele, Verantwortlichkeiten, Risiken, Ressourcen, Termininformationen und Kosten.

Ein weiteres Element der Planungsphase stellt die Terminplanung dar. Projektleiter stehen vor der schier unlösbaren Aufgabe, Annahmen über zukünftige Zustände und deren zeitliche Abhängigkeiten zu treffen, um eine möglichst genaue Terminplanung zu treffen. Projekt-Terminplanungen stellen wichtige Eckpfeiler für den Projekterfolg dar und können in Grob- als auch Fein-Terminplanung untergliedert werden<sup>[7]</sup>. Grundsätzliche Hilfsmittel stellen Phasenpläne, Meilensteinplan, Aktionsplan, Aktivitätenliste, Netzplan und Balkenplan (s. Abb. 1.2) dar. Ziele einer erfolgreichen Terminplanung sind auf der einen Seite Aussagen über den zeitlichen Verlauf des Projekts zu treffen und auf der anderen Seite die Möglichkeit zu bieten, während der Projektdurchführungsphase Soll-Ist-Vergleiche durchzuführen um dementsprechend bei planmäßigen Abweichungen gegensteuern zu können.

Die Grobterminplanung wird i.d.R. mit Hilfe von Meilensteinplänen durchgeführt. Meilensteine markieren den Start einer Phase und stellen zugleich Phasenabschluss der vorherigen Phase dar. Sie stellen Teilziele im Projektfortschritt dar, die zu bestimmten Terminen erreicht werden müssen und

- Standortbestimmungen des Projektes,
- Bilanzziehungen des bisherigen Projektverlaufes und

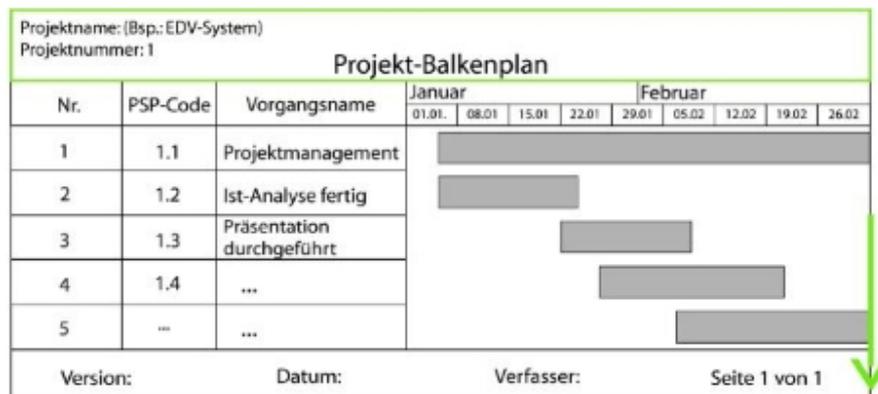


Abb. 1.2 Beispielhafter Balkenplan[5]

- Entscheidungspunkte für Leitlinien des weiteren Projektverlaufes dar.[6]

Damit stellt ein Meilenstein ein überprüfbares Zwischenergebnis dar, welches mit Inhalt und Terminen definiert ist und eine Beurteilung des Projektfortschritts ermöglicht. Die Grobterminplanung gibt somit immer einen Überblick über das Gesamtprojekt, während die Feinplanung die Definition von Teilaufgaben und Arbeitspaketen in einem überschaubaren Bereich vorsieht. Da die Erstellung und Pflege eines solchen Plans mit sehr viel Aufwand betrieben wird, sollte die Projektleitung stets die Frage nach dem Nutzen im Vergleich zum Aufwand innehalten. Eine bis ins Detail aufgestellte Feinplanung vernachlässigt die Dynamik und Komplexität eines Projektes, das durch diese Eigenschaften mit sehr viel Änderungen verbunden ist. Jedes Detail feinzuplanen kann also eine Zeitverschwendung darstellen, die der Projektleiter abzuwägen hat.

Ein weiteres Element der Planungsphase ist die Kostenplanung. Grob zusammengefasst werden die im Laufe des Projektes anfallenden Kosten kalkuliert, Kosten für Arbeitspakete und Ressourcen berechnet und das Projekt einer Wirtschaftlichkeitsbeurteilung unterzogen. Da dieser Aspekt aber für die Projektgruppe Blackout/Restart irrelevant ist, wird auf eine genauere Ausführung an dieser Stelle verzichtet.

### 1.2.3 Durchführungsphase

Die Durchführungsphase umfasst zwei Seiten: Zum einen die Durchführung und Kontrolle des Projektfortschritts und zum anderen Planungs- und Korrekturmaßnahmen bei anfallenden Planabweichungen (Steuerung). Controlling und Steuerung sind daher zentrale Elemente der Durchführungsphase, die auch als Realisierungsphase bezeichnet werden kann.

Die Durchführung und Kontrolle des Projektfortschritts beinhaltet während der Projektrealisierung die PM-Aufgaben Informationsbereitstellung und Kommunikation, Durchführung von Soll-Ist-Vergleichen und Dokumentation[1]. Ersterer Punkt beinhaltet, dass das Projektmanagement allen Projektbeteiligten über den Projektfort-

schritt im ausreichenden Maße regelmäßig und zeitnah informiert. Dazu bedient es sich Hilfsmittel wie Fortschrittsberichte, Meetings, Präsentationen, usw. Die Durchführung von Soll-Ist-Vergleichen dient während der Durchführungsphase der frühzeitigen Vorhersage möglicher Abweichungen von der Projektplanung, der Identifikation bereits eingetretener Abweichungen und der Möglichkeit der schnellen Reaktion auf beschriebene Abweichungen. Die Dokumentation sollte im Projektgeschehen möglichst lückenlos erfolgen, was üblicherweise durch die Ablage projektrelevanter Daten und Unterlagen auf elektronischem Wege geschehen. Sie beinhaltet z.B. Ergebnisprotokolle, Präsentationen, Berichte, dokumentierte Planungsschritte, Projektstrukturplanung usw.

#### ***1.2.4 Abschlussphase***

Ist das Ziel erreicht worden steht der unmittelbare Abschluss des Projektes bevor. Damit einher gehen Abschlusspräsentation, -bericht, -besprechung, Produktabnahme und die Auflösung des Projektteams. Typische PM-Aufgaben in dieser Phase sind die Evaluierung zur Messung des Projekterfolgs, die Reflexion der Umstände der Projektbeteiligten (Was lief gut? Was lief nicht so gut? Was kann besser gemacht werden?), die Erstellung des Projektabschlussberichts und die Entlastung und Auflösung der Projektorganisation.

Da es keine weitere Zusammenarbeit der an der Projektgruppe Blackout/Restart beteiligten Personen geben wird, spielt die Reflexion und Evaluierung eine nebensächliche Rolle. Eine zentrale Rolle für die Projektgruppe wird der auf Basis der Projektdokumentation verfasste Projektabschlussbericht darstellen. Dieser enthält alle für das Projekt relevante Daten, Ergebnisse und Berichte darüber, was die Projektgruppe gemacht und erreicht hat. Letztendlich erfolgt die Entlastung des Projektteams als vorletzte Instanz typischerweise die Unterzeichnung des Abschlussberichts durch den Auftraggeber, der ggf. Nachbesserungen und Nacharbeiten veranlassen kann. Es folgt die Auflösung der Projektorganisation.

### **1.3 Vorgehensmodelle zur Projektdurchführung**

Im Projektmanagement und im speziellen in der Softwareentwicklung haben sich verschiedene Vorgehensmodelle entwickelt. Dazu werden in diesem Kapitel zwischen klassischen, modernen und agilen Vorgehensmodellen unterschieden, die im Folgenden mit Beispielen näher erläutert werden sollen.

### ***1.3.1 Klassische Vorgehensmodelle***

Klassische Vorgehensmodelle werden auch sequenzielle Vorgehensmodelle in der Softwareentwicklung genannt. Man fasst unter ihnen Wasserfall- und Schleifenmodelle zusammen und gliedern das Projekt in sequenziell hintereinander ablaufende Phasen. In den verschiedenen Modellen können beispielsweise folgende Phasen im Softwareentwicklungsprozess berücksichtigt werden:[6]

- Problemanalyse und Grobplanung,
- Systemspezifikation und Planung,
- System- und Komponentenentwurf,
- Implementierung und Komponententest,
- System- und Integrationstest,
- Betrieb und Wartung.

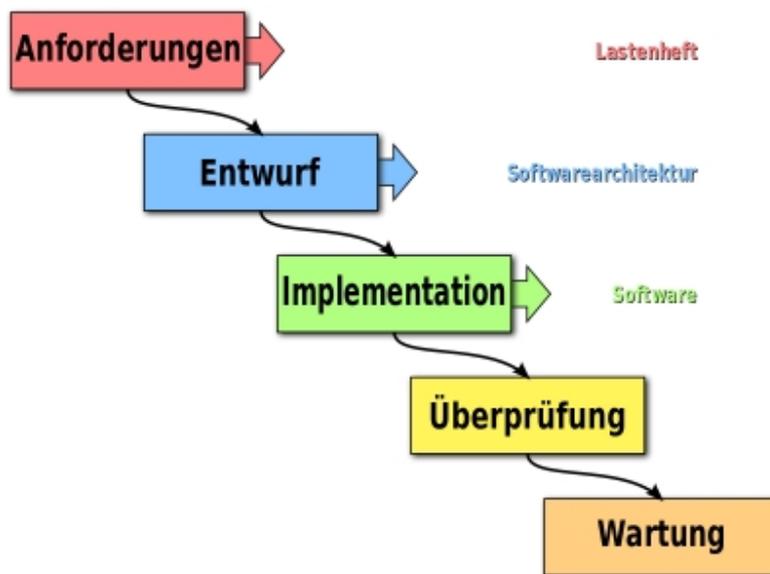
Typischerweise basieren klassische Vorgehensmodelle auf einem Top-down Ansatz und werden durch sequenzielles Vorgehen mit definierten Phasen und Ergebnissen gekennzeichnet. Dabei kann nur in eine Phase vorangeschritten werden, sofern die vorherige Phase vollständig abgeschlossen worden ist. Phasen werden prinzipiell durch das Erreichen der von Beginn an definierten Phasenzielen beendet und führen zum direkten Voranschreiten des Projektfortschritts. Rückschritte in frühere Phasen sind i.d.R. nicht bzw. nur bedingt möglich. Beispiele für klassische Vorgehensmodelle sind das Wasserfall- und Spiralmodell.

#### **1.3.1.1 Wasserfallmodell**

Das Wasserfallmodell stellt das berühmteste Vorgehensmodell sowohl in der Softwareentwicklung als auch im Projektmanagement dar. Es wurde 1970 erstmals entwickelt und 1980 von Barry W. Boehm als Wasserfallmodell bezeichnet. Es sich durch einen Top-down-Ansatz aus, wobei Rücksprünge nur zu benachbarten, vorausgehenden Phase möglich sind. Die jeweiligen Phasen laufen sequenziell hintereinander ab und werden durch einen Validierungsprozess beendet (s. Abb. 1.3). Der Übergang in die nächste Phase erfolgt, falls keine Mängel dabei registriert werden. Das Wasserfallmodell zeichnet sich durch eine hohe Effizienz beim Einsatz von Projekten mit konstanten Anforderungen aus und ist dank seiner außerordentlichen Komplexitätsbeherrschung auch für große Projekte ausgelegt. Für die Projektgruppe Blackout/Restart könnte sie in Abstimmung der Projektbeteiligten eine priorisierte Rolle in der ersten Phase des Requirements Engineering spielen.

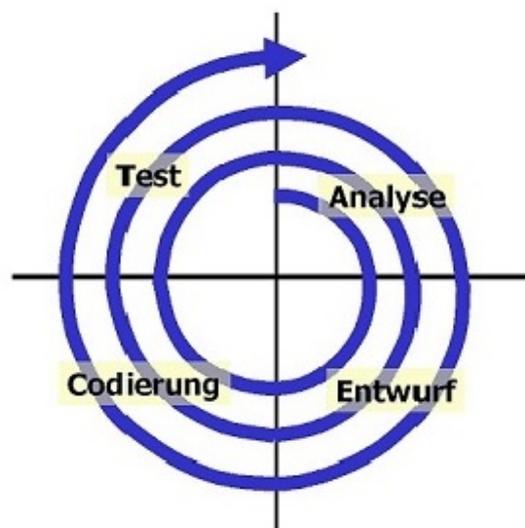
#### **1.3.1.2 Spiralmodell**

Das Spiralmodell wurde im Jahr 1988 von Barry W. Boehm vorgestellt und stellt eine Verfeinerung des Wasserfallmodells dar. Es beschreibt einen evolutionären Prozess, der auf der einen Seite den gesamten Projektaufwand, auf der anderen Seite den



**Abb. 1.3** Beispielhaftes Schema des Wasserfallmodells [3]

Projektfortschritt in den einzelnen Zyklen aufzeigt. Die Entwicklung mit Hilfe des Spiralmodells erfolgt durch das Durchlaufen der Phasen Entwicklung, Integration, Qualitätssicherung und Planung (s. Abb. 1.4). Jede Phase weist mehrere Windung mit mehreren Arbeitspaketen auf. Wie beim Wasserfallmodell ist auch beim Spiralmodell ein Validierungsschritt vorgesehen.



**Abb. 1.4** Beispielhaftes Schema des Spiralmodells[2]

Das Spiralmodell eignet sich besonders für mittel bis große Projekte mit volatilen Anforderungen.

### 1.3.2 Moderne Vorgehensmodelle

Moderne Vorgehensmodelle werden durch verschiedene Faktoren beim Übergang von klassischen Modellen charakterisiert. Dazu zählen eine höhere Flexibilität gegenüber ändernden Rahmenbedingungen, bessere Transparenz der Projekte und Ergebnisse und verstärkte Einbindung des Endnutzers in den Projektverlauf. Nach Aichele und Schönberger sind demnach das V-Modell und das Rational-Unified-Process-Modell (RUP-Modell) modernen Vorgehensmodellen zuzuordnen.[6]

#### 1.3.2.1 V-Modell

Das von Barry B. Boehm im Jahr 1979 entwickelte V-Modell ist ein in der Praxis weit verbreitetes Modell zur Entwicklung von Informationssystemen. Als eine Weiterentwicklung des Wasserfallmodells beinhaltet das V-Modell weitere Elemente als die reine Softwareherstellung: Qualitätssicherung, Konfigurationsverwaltung und Projektmanagement sind inbegriffen (s. Abb. 1.5). Im Laufe der Zeit haben sich eine Reihe von Weiterentwicklungen etabliert. Es erschien zunächst das V-Modell 97 als Vorgabe für den Einsatz im zivilen und militärischen Bereich und anschließend das heute im öffentlichen Bereich gültige V-Modell XT. Das V-Modell findet unter anderem in verschiedensten Branchen - wie Banken, Versicherungen und Automobilindustrie - Gebrauch.

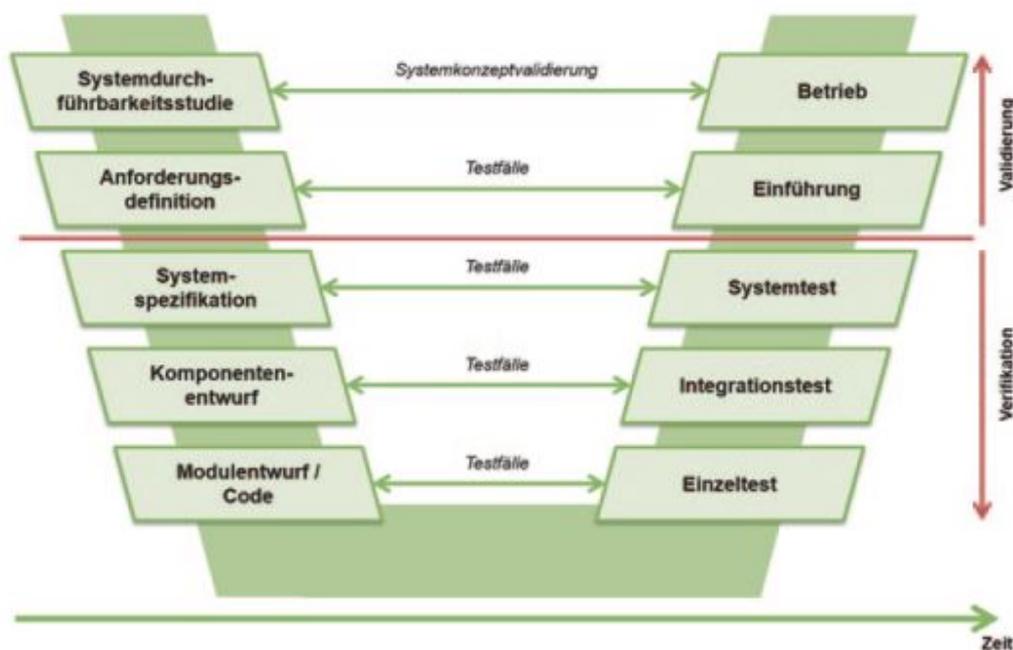


Abb. 1.5 Beispielhaftes Schema des V-Modells[4]

### **1.3.3 Agile Vorgehensmodelle**

Agile Vorgehensmodelle sollen vor allem eine höhere Flexibilität liefern und zeichnen sich durch das frühe Einarbeiten der Codierung, starke Einbeziehung der Nutzer, kontinuierliches Testen und die Weiterentwicklung der Architektur aus.[8]. Dabei entstammen agile Vorgehensmodelle aus der Grundidee der japanischen Lean-Management-Bewegung, die die systematische Vermeidung von Ressourcenverschwendung und Beschleunigung der Abarbeitung von Kundenaufträgen vorsieht. So sollen die Vorteile aus dieser Bewegung durch agile Vorgehensmodelle auf die Softwareentwicklung übertragen werden, was 2001 durch das „Agile Manifest“ festgehalten wurde. Zentrale Ideen sind die Reduktion der Entwurfsphase auf ein Mindestmaß, die Erzeugung früh ausführbarer Softwareprodukte und eine hohe Kundennähe. Im Vergleich zu klassischen Vorgehensmodellen sind agile Vorgehensmodelle durch relativ kurze Iterationen geprägt, wobei ein für den Kunden greifbarer Prototyp nach jeder Iteration entwickelt wird.

#### **1.3.3.1 Scrum**

Scrum ist ein agiles Vorgehensmodell, das seinen Ursprung in der Softwaretechnik hat. Es ist dem Rugby entlehnt und wurde 1995 von Schwaber, Sutherland und Beedle formalisiert, bevor es in einem Regelwerk niedergeschrieben wurde.

Scrum geht von der Tatsache aus, dass Projekte komplex und nicht von Anfang an planbar sind. Daher wird nach Scrum lediglich am Anfang ein grober Rahmen vereinbart, wobei einige wenige Regeln geltend gemacht werden. Der Scrum-Prozess setzt sich typischerweise aus den drei Rollen (Scrum Master, Product Owner, Team), drei Artefakten (Produkt Backlog, Sprint Backlog, Release) und fünf Aktivitäten zusammen (s. Abb. 1.6).

Generell eignet sich Scrum für Projekte mit unklaren Zielen und sich ändernden Anforderungen. Es bietet eine erhöhte Transparenz auf den Projektstand und mögliche Risiken und stellt geringe Ansprüche an die Projektverwaltung. Dafür setzt Scrum ein hohes Maß an Eigenverantwortlichkeit der Projektbeteiligten und an Kommunikationsaufwand voraus. Für die Projektgruppe Blackout/Restart könnte nach der ersten Phase gut mit Scrum aufgrund der genannten Vorzüge gearbeitet werden, auch aus dem Aspekt heraus, dass einige Projektbeteiligte bereits Scrum-Erfahrung besitzen, was eine intensive Einarbeitung in das Themengebiet erübrigt bzw. erleichtert.

## **1.4 Fazit**

Im ersten Kapitel wurden ausgewählte Grundlagen des Projektmanagements dargestellt. Die Begrifflichkeiten 'Projekt' und 'Projektmanagement' wurden erläutert, Hauptaufgaben und Rollen des Projektmanagements dargelegt. Nachdem im zweiten

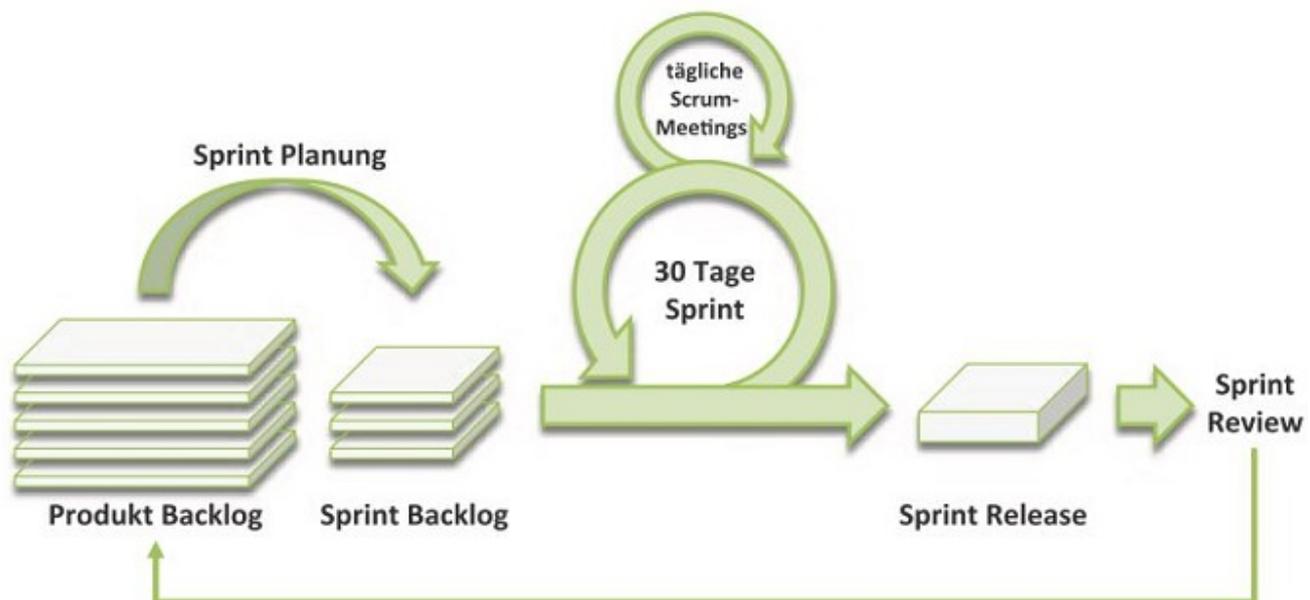


Abb. 1.6 Beispielhaftes Schema von Scrum [4]

Kapitel auf die klassischen Projektphasen näher eingegangen worden ist, wurden ausgewählte Vorgehensmodelle zur Projektdurchführung in der Softwareentwicklung vorgeführt, die der Projektgruppe Blackout/Restart von Nutzen sein könnten. Dies liegt aber in der Entscheidungsbefugnis der Projektbeteiligten, die sich im weiteren Vorgehen auf ein bestimmtes Vorgehensmodell für die verschiedenen Projektphasen einigen müssen. Dies Bedarf einer Diskussion mit anschließender Beurteilung, welches Modell für die Projektgruppe am sinnvollsten ist.

# Literaturverzeichnis

- [1] Durchführung und Controlling. Webseite. URL <http://www.pm-handbuch.com/durchfuhrung-und-controlling/>. Letzter Zugriff 21.05.2016
- [2] Spiral-Modell. Webseite. URL <http://ftp.diebuerger.de/sse/Vorgehensweise/Fenster/SpiralSeite11.htm>. Letzter Zugriff 21.05.2016
- [3] Wasserfall-Modell. Webseite. URL [https://de.wikipedia.org/wiki/Wasserfallmodell#/media/File:Waterfall\\_model-de.svg](https://de.wikipedia.org/wiki/Wasserfallmodell#/media/File:Waterfall_model-de.svg). Letzter Zugriff 21.05.2016
- [4] Aichele, C., Schönberger, M.: IT-Projektmanagement. Effiziente Einführung in das Management von Projekten. Springer Vieweg (2014)
- [5] Hollensteiner, L., U., J.: Balkenplan. Webseite. URL <http://www.projektnachwuchs.de/balkenplan/>. Letzter Zugriff 21.05.2016
- [6] Kessler, H., Winkelhofer, G.: Projektmanagement, 3 edn. Springer (2002)
- [7] Kraus G Westermann, R.: Projektmanagement mit System. Organisation, Methoden, Steuerung, 4 edn. Springer Gabler (2014)
- [8] Kuhrmann, M.: Agile Vorgehensmodelle. Webseite. URL <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Agile-Vorgehensmodelle/index.html>. Letzter Zugriff 21.05.2016
- [9] Motzel, E.: Projektmanagement-Lexikon. Wiley-Verlag (2010)

# Kapitel 2

## Requirements Engineering & SGAM

Hendrik Buhl

**Zusammenfassung** Requirements Engineering spielt bei der Durchführung von Softwareprojekten eine herausragende Rolle. Mit einem ausführlichen und gut geplanten Requirements Engineering können die Grundlagen für ein erfolgreiches Projekt gelegt werden. Anforderungen müssen hier ermittelt und spezifiziert werden, ehe ein Anforderungsdokument verabschiedet wird. Die Formulierung von Anforderungen kann mittels natürlicher Sprache erfolgen, dazu können Modellierungssprachen diese gezielt ergänzen. Für die Modellierung bietet sich einerseits die UML an, dazu kann im Smart-Grid-Kontext SGAM als Modell genutzt werden. SGAM kann ein Smart Grid aus verschiedenen Perspektiven ausgehend von Use Cases bis hin zur physischen Ebene beschreiben. Für die Modellierung der einzelnen Use Cases können dann wiederum andere, beliebige Modellierungssprachen eingesetzt werden. Das Requirements Engineering wird für die Projektgruppe Blackout - Restart eine wichtige Rolle spielen, der Einsatz von SGAM muss hierbei diskutiert werden

### 2.1 Einleitung

In dieser Arbeit, die im Rahmen der initialen Seminarphase der Projektgruppe „Blackout - Restart“ erstellt wird, soll die Bedeutung des Requirements Engineering beleuchtet werden. Dabei werden Bezüge zum speziellen Kontext der Aufgabe der Projektgruppe hergestellt. Besondere Beachtung findet in dieser Arbeit zudem die SGAM-Methode, die speziell zur Modellierung von Anwendungen im Bereich Smart Grids geeignet ist. SGAM steht dabei für *Smart Grid Architecture Model*. Die Arbeit soll sich dabei von einer reinen Darstellung geeigneter Werkzeuge des Requirements Engineering unterscheiden, indem gleichzeitig dessen Relevanz für die Projektgruppe beurteilt wird.

---

Carl von Ossietzky Universität Oldenburg  
E-mail: [hendrik.buhl1@uni-oldenburg.de](mailto:hendrik.buhl1@uni-oldenburg.de)

## 2.2 Requirements Engineering

### 2.2.1 Grundlagen

Requirements Engineering (RE) bezeichnet den Prozess, die Anforderungen an ein zu entwickelndes System zu ermitteln und zu formulieren. Dabei handelt es sich um eine Teildisziplin der Softwareentwicklung, die die erste Phase der Entwicklung bildet, auf die Entwurf und Implementierung sowie Tests folgen [5]. Dieser Disziplin kommt in der Softwareentwicklung eine hohe Bedeutung zu. Etwa ein Drittel aller begonnenen Softwareprojekte wird nicht beendet, wobei der Hauptgrund für den Abbruch von Projekten darin liegt, dass initial erstellte Anforderungen später geändert werden mussten. Dies zeigt die Wichtigkeit, zu Beginn eines Projekts eindeutig formulierte und von allen Beteiligten akzeptierte Anforderungen erstellt zu haben. Ein weiteres Indiz für die Bedeutung des RE ist die Feststellung, dass 43 % aller festgestellten Softwarefehler auf Ungenauigkeiten in der Phase des RE zurückzuführen sind. Hauptfehlerquelle bei Anforderungen sind dabei sprachliche Fehler, also unklar bzw. mehrdeutig formulierte Anforderungen. Zu den sprachlichen kommen noch inhaltliche oder logische Fehler, wobei falsche Sachverhalte oder Widersprüche dargestellt werden. Um der steigenden Bedeutung des RE beizukommen, ist der Aufwand für das RE in der Vergangenheit stark angestiegen. So ergab eine Studie im Jahr 2001, dass 16 % der Projektkosten sowie 39 % der Zeit hierfür aufgewendet wurde. Eine weitere Studie zeigte eine Abnahme der Kostenüberschreitung von Projekten, wenn der Aufwand für das RE erhöht wird [1].

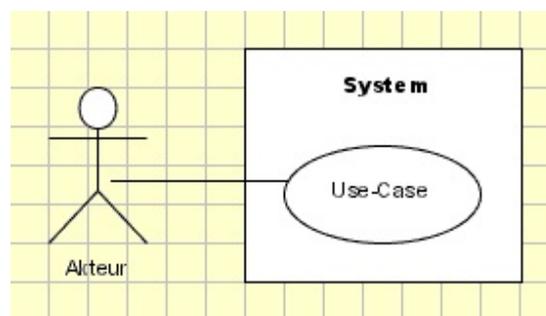
### 2.2.2 Aktivitäten

Der erste Schritt im RE ist das Definieren einer Vision. Dabei wird formuliert, wie die Zukunft durch das neue System aussehen soll. Hierbei geht es nicht darum, auf welche Art und Weise diese Vision verwirklicht werden soll. Vielmehr dient die Vision als gemeinsamer Leitgedanke aller Stakeholder. Im nächsten Schritt werden aus einer Vision Ziele abgeleitet. Diese spezifizieren und konkretisieren die Vision, indem wichtige Eigenschaften des Systems genannt werden. Zu einer Vision können mehrere Ziele gehören [1]. Im Fall der Projektgruppe ergibt sich die Vision aus der Aufgabenstellung und besteht aus bilanzneutralem Wiederaufbau des Stromnetzes nach einem Stromausfall. Alle Stakeholder, also hier die Studierenden der Projektgruppe sowie die verantwortlichen Lehrenden müssen dennoch Visionen und Ziele gemeinsam ausformulieren.

Mit den Zielen werden in dieser Phase des RE bereits Use Cases aufgestellt. Diese Anwendungsfälle fassen funktionale Einheiten des Systems zusammen. Zu einem Use Case gehören dann wiederum mehrere Anforderungen. Anforderungen können in funktionale und nicht-funktionale Anforderungen unterteilt werden, was in Ab-

schnitt 2.2.3 erläutert wird. In den folgenden Schritten werden die Use Cases immer weiter verfeinert. Abbildung 2.1 zeigt exemplarisch ein Use-Case-Diagramm, das neben den Use Cases die Grenzen des Systems sowie Akteure und deren Interaktion mit den Anwendungsfällen zeigt [7].

Bevor nun Anforderungen ermittelt werden können, müssen wichtige Rahmenbedingungen geklärt werden. Diese können die Systementwicklung aus organisatorischer oder technischer Sicht beeinflussen und müssen daher stets beachtet werden. Deswegen müssen sie in dieser frühen Phase des RE geklärt werden. Im Kontext der Projektgruppe Blackout - Restart ist beispielsweise von einem Notbetrieb der Stromversorgung die Rede. Hier muss klar definiert werden, wie dieser Betrieb aussieht und welche Ressourcen in diesem Fall zur Verfügung stehen. Ebenso muss beispielsweise klar sein, welche technische Infrastruktur zur Kommunikation beim Wiederaufbau des Netzes zur Verfügung steht [1].



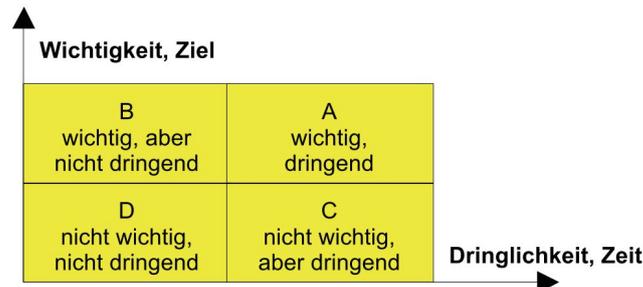
**Abb. 2.1** Use-Case-Diagramm, das Use Case, System und Akteur zeigt [6].

Auf Visions- und Zieldefinition sowie die Beachtung der Rahmenbedingungen folgt das Formulieren von Anforderungen an das System. Diese bilden nach Visionen und Zielen die nächste Stufe der Konkretisierung in der Beschreibung des Systems. Bis die Anforderungen vollständig festgelegt sind, werden mehrere Aktivitäten des RE durchlaufen, die erste ist dabei die Ermittlung der Anforderungen. Daran können alle Stakeholder mitwirken, mögliche Techniken dabei sind Befragung, Beobachtung und Brainstorming. Dabei ist es üblich, diese drei Techniken zu kombinieren [7]. Für die Arbeit der Projektgruppe ist insbesondere eine Befragung der Auftraggeber nach konkreten Vorstellungen denkbar, dazu kann innerhalb der Gruppe ein Brainstorming stattfinden. Nach der Aktivität des Ermittlens liegen zunächst beispielsweise Notizen, Gesprächs- oder Beobachtungsprotokolle vor, die im nächsten Schritt zu konkreten Anforderungen spezifiziert werden.

Die Spezifikation von Anforderungen besteht aus einer Analyse der Notizen aus der Anforderungsermittlung. Ziel ist es, konkrete Anforderungen in natürlicher Sprache oder in einer Modellierungssprache zu formulieren, worauf im Abschnitt 2.2.7 näher eingegangen. Am Ende der Anforderungsspezifikation steht ein Anforderungsdokument, indem alle Anforderungen auf die vorher festgelegte Art und Weise dokumentiert sind. Solch ein Dokument beinhaltet neben den Anforderungen in jedem

Fall eine Einleitung, Rahmenbedingungen sowie notwendige Erklärungen zu den Anforderungen. Außerdem sind die aufgestellten und stetig verfeinerten Use Cases ein wichtiger Bestandteil des Dokuments.

Ein weiterer Schritt, der zur Spezifikation von Anforderungen gehört, ist die Priorisierung. In der Regel werden zahlreiche Anforderungen aufgestellt, die nicht alle von derselben Wichtigkeit sind. Dafür sind zunächst verschiedene Prioritätsgrade zu definieren, die den Anforderungen zugeordnet werden. Eine Möglichkeit zur Priorisierung ist das Eisenhower-Schema, das 4 verschiedene Prioritäten abhängig von Dringlichkeit und Wichtigkeit definiert, wie Abbildung 2.2 zeigt.



**Abb. 2.2** Priorisierung von Anforderungen nach dem Eisenhower-Schema [5].

Beim Anforderungsdokument kann zwischen Auftraggeber- und Auftragnehmersicht unterschieden werden, die Artefakte nennen sich dann Lasten- und Pflichtenheft. Während das Lastenheft aus Auftraggebersicht beschreibt, was in der Umsetzung gefordert wird. Das Pflichtenheft dagegen enthält das Lastenheft und beschreibt zu realisierende Anforderungen [1].

Im nächsten Schritt wird das Anforderungsdokument überprüft, woran alle Stakeholder beteiligt werden. Es soll festgestellt werden, ob die in Abschnitt 2.2.4 definierten Qualitätskriterien erfüllt sind. In diesem Schritt müssen zwei zentrale Fragen beantwortet werden. Die erste ist, ob alle Wünsche der Auftraggeber erfasst wurden. Bei der zweiten Frage geht es darum, ob die aufgestellten Anforderungen korrekt formuliert wurden und keine Unvollständigkeiten oder Unklarheiten mehr vorhanden sind. Wurden Probleme gefunden, so wird das Lösen dieser versucht, was in eine neue Validierung der Anforderungen übergeht. Wenn keine Probleme mehr im Anforderungsdokument gefunden werden, können die Anforderungen verhandelt werden. Beim Verhandeln der Anforderungen müssen die Stakeholder sich auf ein gemeinsames Anforderungsdokument verständigen. Dabei können aufgrund verschiedener Interessen und Kenntnisse Konflikte auftreten, sodass Kompromisse getroffen werden müssen. Liegt nun ein Anforderungsdokument vor, das keine Fehler mehr enthält und mit dem alle Stakeholder einverstanden sind, so wird das Anforderungsdokument verabschiedet, womit die initiale Phase der Anforderungsermittlung abgeschlossen ist. Im weiteren Projektverlauf müssen die Anforderungen weiter beachtet werden, ein gutes Anforderungsmanagement deckt Änderungen auf

und garantiert eine gute Nachvollziehbarkeit der Anforderungen von der Modellierung bishin zum Programmcode [7].

### ***2.2.3 Funktionale und nicht-funktionale Anforderungen***

[1] liefert eine Definition für Anforderungen, die lautet: „Anforderungen legen fest, was man von einem Softwaresystem als Eigenschaften erwartet“. Bei Anforderungen wird grundlegend zwischen funktionalen und nicht-funktionalen Anforderungen unterschieden. Funktionale Anforderungen beschreiben, was das System tut, während die nicht-funktionalen das „Wie“ beschreiben. Daraus ergibt sich, dass beide Arten im Zusammenhang zueinander stehen, indem die nicht-funktionalen Anforderungen sich auf bestimmte Funktionen beziehen. Die nicht-funktionalen Anforderungen können sich beispielsweise auf Zuverlässigkeit, Verfügbarkeit oder Sicherheitsaspekte einer funktionalen Anforderung beziehen. Die nicht-funktionalen Anforderungen lassen sich in weitere Arten von Anforderungen aufteilen. Beispiele hierfür sind Qualitätsanforderungen, rechtliche Anforderungen oder technische Anforderungen [1].

### ***2.2.4 Qualitätskriterien für Anforderungen***

Aufgrund der hohen Bedeutung des RE sind an die Formulierung von Anforderungen selbst hohe Anforderungen gestellt. Werden diese Qualitätsmerkmale verletzt oder vernachlässigt, so verzögert sich die Verabschiedung der Anforderungen oder in späteren Entwicklungsschritten treten gar Fehler auf. Wichtige Qualitätskriterien für Anforderungen sind in Tabelle 2.1 dargestellt. Dabei muss eine Anforderung korrekt sein, also die Systemfunktion auch tatsächlich beschreiben. Die Eindeutigkeit einer Anforderung kann insbesondere durch eine Modellierungssprache gewährleistet werden. Dazu kommen Qualitätskriterien wie Vollständigkeit, Konsistenz, Überprüfbarkeit und Realisierbarkeit. Die Lösungsneutralität untermauert eine wichtige Eigenschaft im RE. Es gibt hier primär, was ein System erfüllen muss, der Weg wird hierbei weniger beachtet [1, 5, 7].

### ***2.2.5 Anforderungsattribute***

Die Bestandteile einer Anforderung werden als Attribute bezeichnet. Ein Attribut liefert Informationen über einen bestimmten Aspekt der Anforderung. In [1] und [7] sind zahlreiche Attribute genannt, von denen einige in Tabelle 2.2 zusammengestellt wurden. ID, Beschreibung, Sicht und Querbezüge sind dabei eher beschreibende Attribute, die für ein geordnetes Anforderungsmanagement der Projektgruppe wichtig

**Tabelle 2.1** Qualitätskriterien für Anforderungen [1, 7].

Kriterium	Beschreibung
Korrektheit	Das System soll erfüllen, was die Anforderung beschreibt
Eindeutigkeit	Alle Stakeholder interpretieren die Anforderung gleich.
Vollständigkeit	Die beschriebene Funktionalität wird vollständig beschrieben.
Konsistenz	Es gibt keine Widersprüche innerhalb der Anforderung.
Überprüfbarkeit	Es ist möglich, die Umsetzung der Anforderung nachzuprüfen.
Realisierbarkeit	Unter gegebenen Rahmenbedingungen kann die Anforderung umgesetzt werden.
Lösungsneutralität	Es wird nur das "Was" vorgeschrieben, der Weg zum Ziel wird offen gelassen.

**Tabelle 2.2** Auswahl an Anforderungsattributen [1, 7].

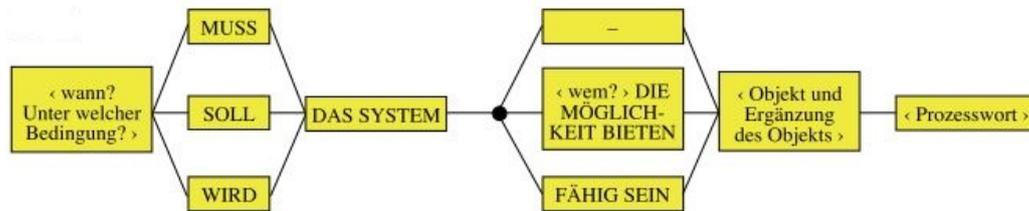
Attribut	Beschreibung
ID	Eindeutiger Identifikator
Typ	Funktional oder nicht-funktional
Beschreibung	Formulierung der Anforderung mittels natürlicher Sprache oder eines Modells
Sicht	Beschreibt die Anforderung die Statik, die Dynamik oder die Logik des Systems?
Querbezüge	mögliche Abhängigkeiten von anderen Anforderungen
Abnahmekriterien	Z.B. Testszenario
Priorität	Priorität der Anforderung
Stabilität	Wahrscheinlichkeit der späteren Überarbeitung
Aufwand	geschätzter Aufwand der Realisierung
Autor	Autor und Verantwortlichkeit für die Anforderung

sind. Durch Abnahmekriterien von Anforderungen wird es bereits in einer frühen Projektphase notwendig sein, Testszenarien zu entwickeln. Für Priorität und Stabilität müssen zunächst eigene Maße definiert werden, ein Beispiel für die Priorisierung bietet Abschnitt 2.2.2. Durch die Festlegung der Verantwortlichkeiten für jede Anforderung können Aufgaben gemäß der Kompetenzen verteilt werden, dazu ist die verantwortliche Person klar benannt [1, 7].

### 2.2.6 Natürlichsprachliche Formulierung

Bei der natürlichsprachlichen Formulierung von Anforderungen muss insbesondere auf Eindeutigkeit und Verständlichkeit geachtet werden, [7] formuliert hierfür einige Regeln. Dabei wird jede Anforderung durch genau einen Hauptsatz beschrieben. In diesem Satz müssen die in Tabelle 2.2 genannten Attribute vorkommen. Zur Formulierung eignet sich eine Anforderungsschablone, die eine einheitliche Formulierungsvorlage bietet. Ein Beispiel bietet hier die Schablone aus [7], wie sie in Abbildung 2.3 gezeigt wird. In dieser Formulierung spielt speziell die Wahl des Prädikats eine besondere Rolle im Bezug auf die Verbindlichkeit einer Anforderung. Das Wort „muss“ impliziert dabei eine Pflichtenforderung, während „soll“ einen Wunsch darstellt. Eine Absichtserklärung, die zukünftige Entwicklungen berücksichtigen soll, kann durch das Wort „wird“ ausgedrückt werden. Dazu kommt noch das

Schlüsselwort „kann“, was mögliche Lösungen vorschlägt, die aber nicht verbindlich umgesetzt werden müssen. In Abbildung 2.3 ist zusätzlich eine Vorbedingung der Anforderungen vorhanden, diese besagt, unter welcher Voraussetzung diese Anforderung relevant ist. Die Vorbedingung kann beispielsweise lauten „wenn das System aktiviert wurde“. Da der nach der Schablone formulierte Satz nicht alle Attribute



**Abb. 2.3** Anforderungsschablone zur Formulierung natürlichsprachlicher Anforderungen [5].

enthalten kann, kann eine Anforderung mit allen Attributen zusammen tabellarisch dokumentiert werden. Der formulierte Satz ist dabei beim Attribut „Beschreibung“ einzuordnen [5, 7].

### 2.2.7 Modellierung

Neben der natürlichsprachlichen Formulierung von Anforderungen eignen sich Modellierungssprachen ebenfalls zur Darstellung von Anforderungen. Während natürliche Sprache von allen Beteiligten zumindest syntaktisch verstanden wird, kann es bei Modellierungen weniger zu Mehrdeutigkeiten kommen. Insbesondere die Nutzung von UML-Modellen mit ihrer einheitlichen Syntax bietet sich hier an. Abhängig von der Sicht auf das System ist das geeignete Modell zur Modellierung einer Anforderung zu wählen. Sind dynamische bzw. prozedurale Aspekte darzustellen, eignen sich insbesondere Sequenzdiagramm, Aktivitätsdiagramm oder Zustandsdiagramm. Soll dagegen die Statik eines Systems beschrieben werden, kann ein Klassendiagramm Zusammensetzungen und Hierarchien abbilden.

Modellierung spielt ebenfalls eine bedeutende Rolle, wenn das Requirements Engineering abgeschlossen wird und in die Phase des Systementwurfs übergeht. Hier muss ein Produktmodell erstellt werden, was danach wiederum implementiert wird. Mit modellierten Anforderungen wird dieser Übergang erleichtert und es können erstellte Modelle direkt wieder verwendet werden [1]. Ein vielseitiges und ganzheitliches Modell zur Modellierung von Smart Grids und deren Anforderungen ist *SGAM* (*Smart Grid Architecture Model*), das im nächsten Kapitel ausführlich vorgestellt wird.

## 2.3 SGAM

SGAM ist ein Modell, das Smart Grids aus verschiedenen Sichten beschreiben kann. Das Modell wurde im Jahr 2012 von der *Smart Grid Coordination Group* entwickelt, um für Smart Grids einen einheitlichen Standard zur Modellierung einzuführen. Grundlage des Modells sind dabei die fünf Schichten, die *SGAM Layers*. Jede Schicht repräsentiert eine spezielle Sicht auf das System. Innerhalb jeder Schicht werden die Bestandteile des Systems in der Ebene platziert, die sich mittels *Domains* und *Zones* aufspannt. Ein Beispiel für ein SGAM-Modell ist in Abbildung 2.4 zu sehen. Bei Betrachtung der Schichten ist die vertikale Abdeckung von besonderer Bedeutung. Jeder Systembestandteil einer Schicht muss in allen Schichten eine Entsprechung am selben Punkt der Ebene haben [4].

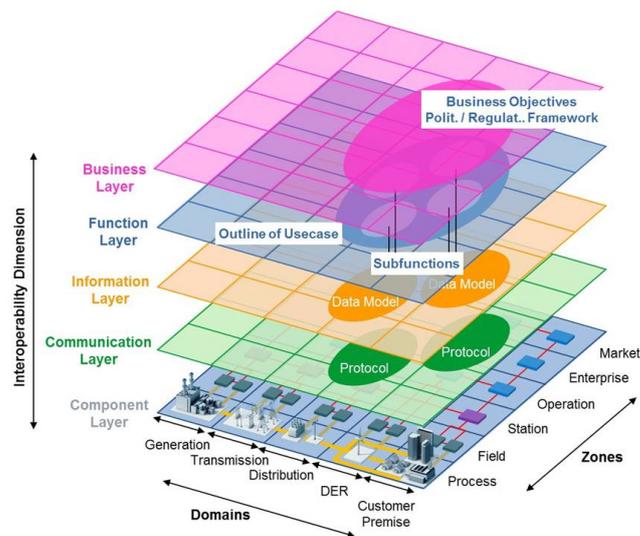


Abb. 2.4 Beispielhaftes SGAM-Modell [3].

### 2.3.1 Schichten

Jede der Schichten repräsentiert eine Sicht auf das System. Tabelle 2.3 zeigt einen Überblick über die fünf Schichten und ihre Aufgaben. Die Business Layer stellt dabei das System aus einer „Business“-Sicht dar. Dort werden sog. High Level Use Cases definiert, also Anwendungsfälle, die das System aus sehr ganzheitlicher, geschäftlicher Sicht. Darunter, in der Function Layer, wird das System in funktionale Einheiten zerlegt, die innerhalb der Ebene platziert werden. Zudem werden Verbindungen zwischen den Funktionseinheiten gezogen, wenn diese miteinander in Interaktion treten. In der Function Layer werden demnach einzelne Algorithmen des Gesamtsystems beschrieben, die mithilfe der darunter liegenden Schichten im-

**Tabelle 2.3** Schichten des SGAM[7].

Schicht	Aufgabe
Business Layer	Politische/wirtschaftliche Rahmenbedingungen, High Level Use Cases
Function Layer	Funktionale Einheiten, Primary Use Cases, Algorithmen
Information Layer	Übertragene Informationsobjekte
Communication Layer	Übertragungsprotokolle und -technologie
Component Layer	Physisch vorhandene Komponenten, Rechner, Stromnetz

plementiert werden können. In der dritten Schicht, der Information Layer, geht es um die Kommunikation zwischen den Systembestandteilen. Hier wird modelliert, welche Informationsobjekte ausgetauscht werden. Wie diese gesendet und empfangen werden, beschreibt die Communication Layer. In dieser werden Datenmodelle und Übertragungsprotokolle modelliert. Auch die Übertragungstechnologie wie z.B. Ethernet wird hier beschrieben. Zuletzt stellt die Component Layer dar, auf welchen Geräten die darüber beschriebenen Funktionen ausgeführt werden. Verbindungen repräsentieren hier reell vorhandene Verbindungen zwischen Komponenten. Zusätzlich können hier bereits vorhandene Komponenten wie z.B. das Verteilnetz eingefügt werden [4].

### 2.3.2 Domänen

Eine Dimension jeder Schicht ist die Domäne. Die Wahl der Domäne sagt etwas darüber aus, wo in der Wertschöpfungskette der Energieerzeugung der jeweilige Teil des Systems anzusiedeln ist. Die fünf SGAM-Domänen sind in Tabelle 2.4 aufgelistet. Diese Domänen decken Energieerzeugung und -verbrauch in allen Be-

**Tabelle 2.4** Domänen des SGAM [4].

Domäne	Beschreibung
Bulk Generation	Große Energieerzeuger, Kraftwerke
Transmission	Übertragungsnetz
Distribution	Verteilnetz
DER (Distributed Electrical Resources)	Kleinere Energieerzeugungsanlagen im Verteilnetz
Customer Premises	Erzeuger und Verbraucher beim Endkunden

reichen ab. Die Energie kann in großen Kraftwerken erzeugt werden, ehe sie mittels Übertragungs- und Verteilnetz zum Endnutzer gelangt. Im Verteilnetz können zudem weitere kleinere Erzeugungsanlagen vorhanden sein. Alle diese Orte finden sich in den SGAM-Domänen wieder, sodass alle Teile eines zu entwickelnden Systems einer Domäne zugeordnet werden können[2, 4].

### 2.3.3 Zonen

Die zweite Dimension jeder Ebene bilden die SGAM-Zonen. Während die Domänen den Ort eines Prozesses beschreiben, geht es bei den Zonen eher um die Art des Prozesses. Beispielsweise wird dabei unterschieden, ob es direkt um die Transformation von Energie oder um dessen Management im Unternehmen geht. Die SGAM-Zonen sind in Tabelle 2.5 dargestellt.

**Tabelle 2.5** Zonen des SGAM [4].

Zone	Beschreibung
Process	Transformation von Energie, direkt beteiligte Geräte
Field	Kontrolle und Monitoring der Transformation, Sensorik/Aktorik
Station	Funktionale Einheiten der Field-Zone, Zusammenfassung mehrerer Sensoren
Operation	Management und Kontrolle
Enterprise	Betriebliche Prozesse, Personal, Fortbildung, Logistik
Market	Markt, Verbindung zum Kunden

Die Zonen schreiben somit hierarchisch Bereiche der Energiewirtschaft. Die erste Zone ist die Process-Zone, in der es direkt um die Transformation von Energie geht. Die Überwachung dieser bzw. das Monitoring dieses Prozesses mit Hilfe von Sensorik und Aktorik gehört dann der Field-Zone an, während die Station-Zone mehrere Field-Einheiten zusammenfasst. Dagegen ist die Operation-Zone schon weiter von technischen Aspekten entfernt, hier geht es um Management und Kontrolle der Prozesse der zuvor beschriebenen Zonen. In der nächsten Zone, der Enterprise-Zone, sind dann Prozesse auf Unternehmensebene angesiedelt, diese Zone umfasst konkrete Unternehmensbereiche wie Logistik, Fortbildung oder Personal. Die letzte und in dieser Hierarchie höchste Zone ist die Market-Zone, diese umfasst den Markt und bezieht damit Aspekte, die den Endnutzer betreffen, mit ein. Während die Enterprise-Zone Prozesse innerhalb eines Unternehmens umfasst, ist in der Market-Zone über den Markt Kontakt zur Welt außerhalb eines Unternehmens vorhanden [4].

### 2.3.4 Einsatz von SGAM

SGAM eignet sich für die vollständige Modellierung eines Smart-Grid-Systems ausgehend von den ermittelten Anforderungen. Hier gibt [4] ein Beispiel, wie die Schichten schrittweise und aufeinander aufbauend entwickelt werden können. Wie in Kapitel 2.2 erwähnt, beinhaltet des Requirements Engineering bereits eine Use-Case-Analyse. Mit Hilfe der erstellten Use Cases wird hier mit der Modellierung begonnen, durch deren Verfeinerung werden dann alle Schichten vervollständigt.

Die Use Cases ergeben sich aus Zielen, den sog. *Business Goals*, die mit Business Cases einhergehen. Daraus werden *High Level Use Cases* (HLUC) abgeleitet, die

sich wiederum in *Primary Use Cases* (PUC) aufteilen. Dieser Zusammenhang zwischen HLUC und PUC ist in Abbildung 2.5 abgebildet, wobei hier nur ein HLUC aus einem System mit mehreren ausgewählt wurde. Dieses Vorgehen ist sehr ähnlich dem allgemeinen Vorgehen im RE, wo aus Zielen Use Cases abgeleitet und verfeinert werden. Die PUC bilden jetzt einzelne Funktionsblöcke bzw. Algorithmen des

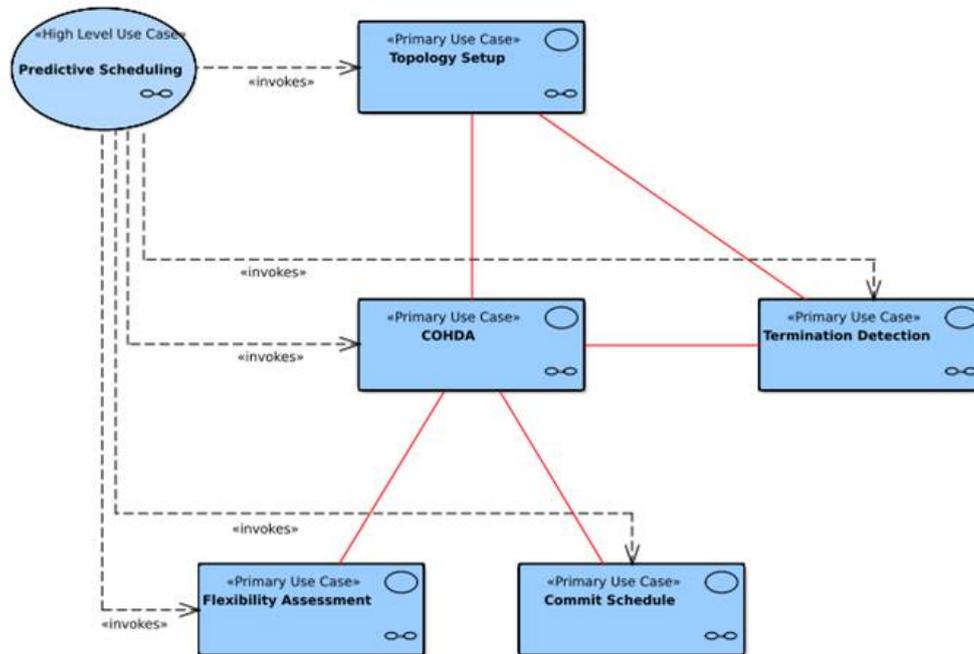
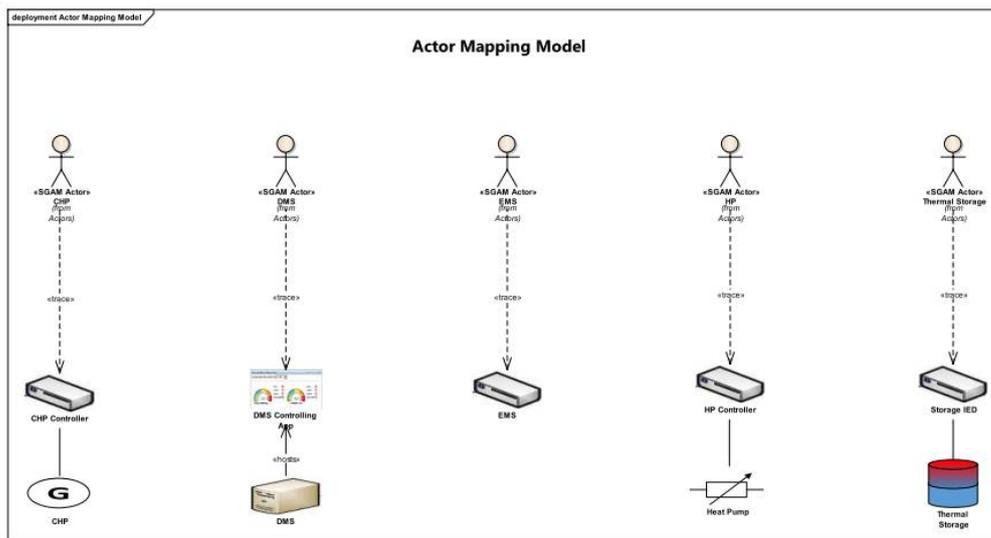


Abb. 2.5 Zusammenhang zwischen HLUC und PUC beim SGAM [2].

Systems und können in der Function Layer platziert werden. Zudem werden Verbindungen zwischen einzelnen PUC gezogen, wenn diese miteinander interagieren. Für die Modellierung von hoher Bedeutung sind zudem Akteure, die *SGAM Actors*. Während die PUCs beschreiben, was das System hier leistet, geht es beim Erstellen der Akteure um die Frage, wer daran beteiligt ist. Die konkrete Interaktion, die in diesem Use Case stattfindet, kann dann mittels UML-Sequenz- oder Aktivitätsdiagramm dargestellt werden. Für jeden PUC müssen die Akteure identifiziert und ebenfalls in der Function Layer platziert werden. Nach der Entwicklung der Function Layer wird ersichtlich, welche Domänen und Zonen relevant sind, sodass der zum jeweiligen HLUC gehörige Business Case modelliert werden kann. Somit sagt die Business Layer und ein zugehöriger Business Case darüber aus, welche Bereiche insgesamt der SGAM Ebene insgesamt betroffen sind. Die Business Layer fasst somit die einzelnen Bestandteile der Function Layer zusammen. Diese Gesamtheit ist aus geschäftlicher Sicht relevant, die darunter liegenden funktionalen Einheiten eher aus Entwicklersicht [4].

Nachdem die Business Layer und die Function Layer entwickelt wurden, werden den ermittelten logischen Systembestandteilen physische Komponenten zugeordnet, die in der Component Layer modelliert werden. Dafür findet für jeden SGAM Actor die direkte Zuordnung einer Komponente statt. Dies sind in der Regel Rechner oder Steuergeräte, des weiteren können zusätzliche Ressourcen wie das elektrische Verteilnetz oder ein Energiespeicher hinzugefügt werden. Abbildung 2.6 zeigt die Zuordnung der Steuergeräte zu den Akteuren und der Ergänzung um weitere Teile wie das Netz, ein Wärmespeicher oder eine -pumpe. Diese Elemente werden im



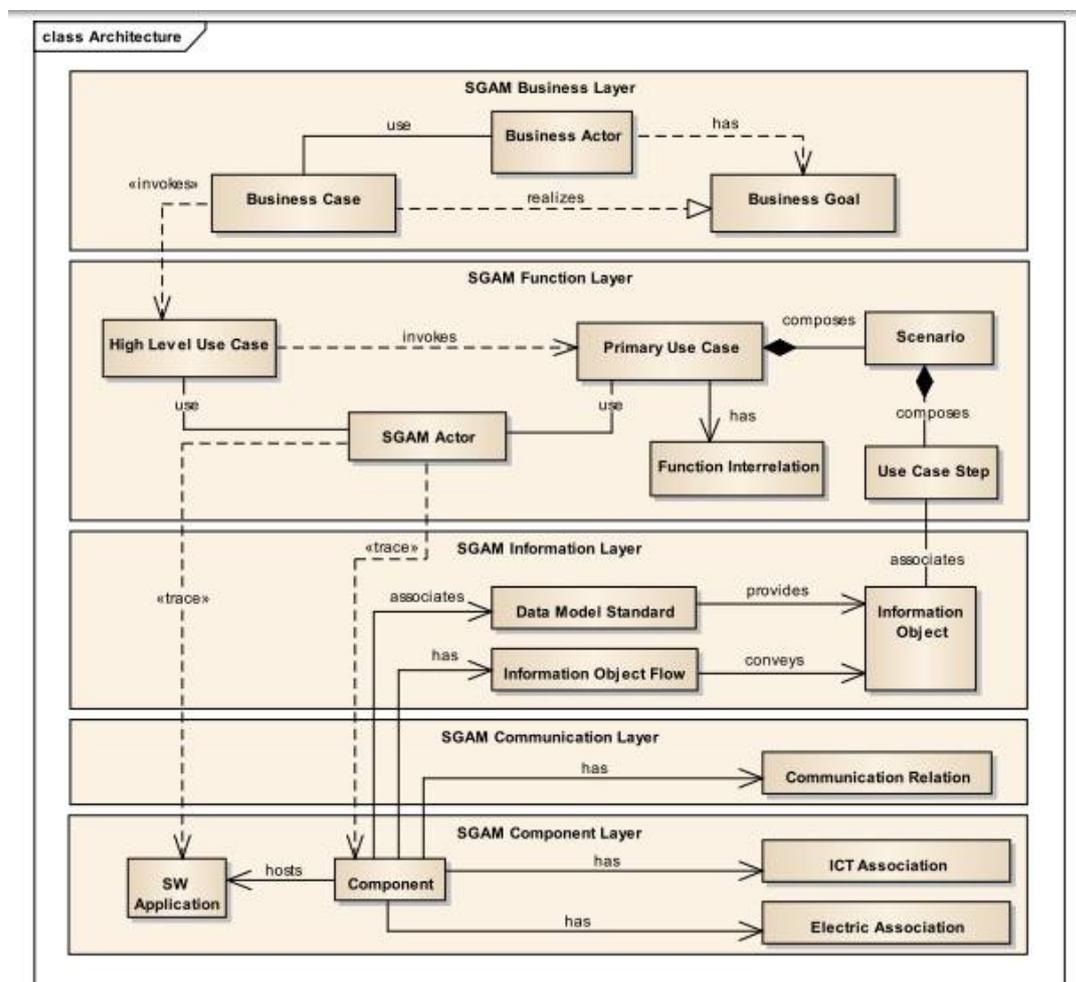
**Abb. 2.6** Zuordnung der SGAM Actors zu physischen Komponenten. Zusätzliche Komponenten werden ergänzt, sodass alle benötigten Geräte hier abgebildet werden [4].

nächsten Schritt in die Component Layer integriert, dabei sind sie in den Domänen und Zonen wie ihre zugehörigen PUC. Verbindungen zwischen Elementen der Component Layer repräsentierten physisch vorhandene Verbindungen.

Nachdem Use Cases ermittelt und auf zugehörige Komponenten abgebildet wurden, werden in der Information Layer Informationsflüsse des Systems spezifiziert. Da bereits die Bestandteile der Component Layer modelliert wurden, können diese ohne ihre Verbindungen zueinander in die Information Layer übernommen werden. Die Einordnung in Domänen und Zonen ist somit bereits abgeschlossen. Nun ist es in dieser Schicht die Aufgabe, Informationsflüsse zwischen den Komponenten einzuzeichnen. Dazu werden gerichtete Pfeile verwendet, die mit dem übertragenen Informationsobjekt beschriftet werden. So wird sichtbar, welche Objekte zwischen welchen Komponenten versendet werden und in welche Richtung die Kommunikation gerichtet ist. Es kann durch diese Modellierung sofort erkannt werden, ob die Kommunikation bidirektional ist oder ob eine Komponente ein reiner Sender oder Empfänger ist.

Um das SGAM-Modell fertigzustellen, ist außerdem relevant, auf welche Art und Weise die zuvor definierten Informationen versendet werden. Dies modelliert die Communication Layer. Genau wie bei der Entwicklung der Information Layer werden die Elemente der Component Layer ohne Verbindungen übernommen. Jetzt wird überall, wo in der Information Layer ein beliebig gerichteter Informationsfluss existiert, ein beidseitig gerichteter Verbindungspfeil gezogen. Durch die Beschriftung dieses Pfeils wird zuletzt noch spezifiziert, mit welchem Übertragungsprotokoll und mit welcher Technologie die Daten übertragen werden. Beispielhaft könnten Daten gemäß des OPC-UA-Standards mittels einer Ethernet-Verbindung übermittelt werden.

Um die gesamte Modellierung des SGAM-Modells und die Abhängigkeiten der Schichten voneinander auf allgemeiner Basis zu veranschaulichen, eignet sich das SGAM-Metamodell, was in Abbildung 2.7 zu sehen ist. Insbesondere kann hier in



**Abb. 2.7** SGAM Metamodell, das die Abhängigkeiten zwischen den einzelnen Schichten veranschaulicht [4].

den oberen Schichten der Zusammenhang von Business Case, HLUC und PUC nachvollzogen werden. Ein PUC benötigt Aktuere, die eine direkte Entsprechung in der Component Layer haben. Diese Komponenten besitzen nun Informationsflüsse zu anderen, die zur Information Layer gehören. Außerdem haben diese Komponenten physische Verbindungen zur Kommunikation, dessen Technologie in der Communication Layer modelliert wird.

## 2.4 Fazit

Ein ausführliches und sorgfältiges Requirements Engineering ist eine wichtige Voraussetzung für die erfolgreiche Umsetzung eines Projektes. Folglich spielt das RE auch die Projektgruppe Blackout - Restart eine herausragende Rolle und sollte mit ausreichend Zeit bedacht werden. Wie viel Zeit dafür beansprucht wird, hängt von der restlichen Projektplanung ab, jedoch sollte im Zweifelsfall keine Zeit am RE gespart werden, da dies häufig zu Fehlern in der Umsetzung führt. Stakeholder der Projektgruppe sind lediglich die Studierenden als Auftragnehmer sowie die Lehrenden als Auftraggeber, zwischen diesen Gruppen ist insbesondere im RE die Kommunikation wichtig. Die Studierenden werden nach der Visions- und Zieldefinition die Anforderungen zunächst ermitteln und formulieren, dabei muss eng mit den Lehrenden als Auftraggeber zusammengearbeitet werden. Beim Verabschieden dieser müssen die Lehrenden miteinbezogen werden und ihr Einverständnis geben, da dieser Schritt zwingend alle Stakeholder einbezieht. Danach kann das Projekt in die nächste Phase gehen.

SGAM bietet die Möglichkeit, ein System umfassend und auf verschiedenen Perspektiven zu modellieren. Der Einsatz dieser Methode kann dabei eng mit dem RE verknüpft werden. Die Entwicklung der Business sowie der Function Layer geht einher mit einer Ziel- und Use-Case-Definition. Darauf aufbauend kann das System in den anderen Schichten spezifiziert werden, was mit einem logischen sowie technischen Systementwurf zu beschreiben ist. Somit kann die Modellierung von Anforderungen vollständig in SGAM eingebettet werden.

Durch das Schichtenmodell kann sofort erkannt werden, auf welcher Steuereinheit ein Algorithmus ausgeführt wird und mit welchem Protokoll eine Information verschickt wird. Das Modell bietet sich dafür an, in einer Entwurfsphase eines Projekts einen Überblick über alle Komponenten zu geben. Da die Modellierung mit SGAM zeitaufwändig ist, eignet sie sich besonders im Rahmen klassischer Projektmodelle, in denen es eine ausführliche Konzeptphase gibt, ehe die Umsetzung beginnt. Gleichzeitig liefert ein fertiges SGAM-Modell durch die Elemente der Function Layer eine Vorlage für den Beginn einer modularen Systementwicklung. Die einzelnen Module können dann z.B. mit einer agilen Methode entwickelt werden. Somit bietet SGAM ein Grundgerüst bzw. eine Übersicht über ein zu entwickelndes System als Grundlage für die Implementierung. Für die konkrete Modellierung einzelner Use Cases

können dann beliebige Modellierungssprachen eingesetzt werden, womit ein direkter Übergang vom RE zur Entwurfsphase geschaffen wird.

Im Rahmen der PG Blackout - Restart kann SGAM eingesetzt werden, wie es im vorherigen Absatz beschrieben wurde. Die Entscheidung für oder gegen SGAM hängt insbesondere von der Wahl des Projektmodells ab, da die Modellierung aller Schichten viel Zeit in Anspruch nehmen würde. Bei einem klassischen Projektmodell wie dem Wasserfallmodell könnte sich die SGAM-Modellierung an das Requirements Engineering anschließen bzw schon währenddessen beginnen. Dabei werden die im RE aufgestellten Ziele und Use Cases in der Modellierung verwendet. Ein weiteres Argument für die Wahl von SGAM ist die konkrete Ausrichtung auf den Bereich Energie und die Berücksichtigung der Infrastruktur mit Erzeugern, Verbrauchern und Netz. Von besonderer Wichtigkeit wird in der Projektgruppe ebenfalls das Zeitmanagement sein. Die Projektdauer ist klar vorgegeben, insofern sollte das RE in der Gesamtplanung ausreichend berücksichtigt werden und zu einem festgelegten Termin abgeschlossen werden. Dasselbe gilt für den Systementwurf, für den Zeit aufgewendet werden muss, insbesondere bei der Nutzung von SGAM. Sollte die Wahl auf die Nutzung von SGAM fallen, so bietet [4] eine Toolbox an, mit der ein System schrittweise und systematisch nach SGAM entwickelt werden kann.

# Literaturverzeichnis

- [1] Balzert, H.: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3., Aufl. edn. Spektrum Akademischer Verlag, Heidelberg, Neckar (2009). URL [http://aleph.bib.uni-mannheim.de/F/?func=find-b&request=308066812&find\\_code=020&adjacent=N&local\\_base=MAN01PUBLIC&x=0&y=0](http://aleph.bib.uni-mannheim.de/F/?func=find-b&request=308066812&find_code=020&adjacent=N&local_base=MAN01PUBLIC&x=0&y=0)
- [2] Hinrichs, C., Sonnenschein, M.: Design, analysis and evaluation of control algorithms for applications in smart grids. CoRR **abs/1506.03663** (2015). URL <http://arxiv.org/abs/1506.03663>
- [3] Maus, D., Pflipsen, A.J., Schwabe, P.: Planning, Simulation & Evaluation for Energy Grids. RWTH Aachen. URL <https://www.ti.rwth-aachen.de/research/smartgrid/>. Retrieved 10 May 2016
- [4] Neureiter, C.: Introduction to the „SGAM“ Toolbox. Salzburg (2014)
- [5] Partsch, H.: Requirements Engineering systematisch. EXamen. press Series. Springer Berlin Heidelberg (2010). URL <https://books.google.de/books?id=8CZtjArMxAwC>
- [6] Schäling, B.: Der moderne Softwareentwicklungsprozess mit UML (2010). URL <http://www.highscore.de/uml/>
- [7] Winter, A.: Vorlesungsfolien Requirements-Engineering and -Management. Universität Oldenburg (2015)

# Kapitel 3

## COHDA - A Combinatorial Optimization Heuristic for Distributed Agents

Stephanie Alexandra Warsch

**Zusammenfassung** Im Rahmen der Energiewende stellt die organisatorische Integration dezentraler Energiewandlungsanlagen in den Strommarkt ein neuartiges Koordinationsproblem dar. Die Heuristik COHDA soll im Rahmen einer prädiktiven Einsatzplanung einen Verbund von Energieanlagen ermächtigen, ein gegebenes Wirkleistungsprodukt zu produzieren. Ein möglicher Anwendungsfall wäre, dieses an einer Strombörse verkaufen zu können. Auf Basis eines Multi-Agenten Systems zur Steuerung der Energieanlagen wird dabei eine Optimierung der Fahrpläne der einzelnen Anlagen durchgeführt, die zu einer globalen Lösung - dem Produzieren des gegebenen Wirkleistungsproduktes - führt. Dabei werden für den einzelnen Anlagenbetreiber Autonomie und Datenschutz gewahrt. Gleichzeitig erhält dieser, aber auch der Abnehmer des Stroms, durch die prädiktive Ausgestaltung der Optimierung eine erhöhte Planungssicherheit. In dieser Arbeit wird zunächst auf die Ausgestaltung des Agentensystem eingegangen. Anschließend wird der Ablauf der Heuristik erläutert. Abschließend wird ein Ausblick auf einen möglichen Einsatz der Heuristik im Rahmen der Projektgruppe Blackout-Restart gegeben.

### 3.1 Einleitung

In seiner im Mai 2015 veröffentlichten Dissertation „Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid“ hat Christian Hinrichs die Heuristik COHDA veröffentlicht.[1] Diese soll durch ein Multi-Agenten System einen Verbund von Energieanlagen dazu befähigen produzierten Strom an einer regulären Strombörse zu handeln.

Hintergrund hierfür ist die starke Förderung des Ausbaus von Erneuerbaren Energien. Das Erneuerbare Energien Gesetz regelt die bevorzugte Einspeisung von Strom aus Erneuerbaren Energien.[3] Dennoch müssen die so geförderten Anlagen langfris-

---

Carl von Ossietzky Universität Oldenburg

E-mail: [stephanie.alexandra.warsch@uni-oldenburg.de](mailto:stephanie.alexandra.warsch@uni-oldenburg.de)

tig nicht nur technisch, sondern auch organisatorisch vollständig in das Stromnetz integriert werden.[5]

Der starke Ausbau von Anlagen zur Erzeugung von Strom aus erneuerbaren Energien hat unter anderem Auswirkungen auf die regionale Verteilung von Energieerzeugungsanlagen. Die Entwicklung geht von einer zentralen Erzeugung durch große Kraftwerke hin zu einer dezentralen Stromerzeugung vieler kleinerer Energiewandlungsanlagen. [5] Die organisatorische Integration dieser dezentralen Energieerzeuger in das Energienetz bringt Koordinationsprobleme mit sich, für deren Lösung Mechanismen entwickelt werden müssen. [6] Einen solchen Koordinationsmechanismus stellt COHDA dar.

Ziel dieser Seminararbeit ist es, diese Heuristik möglichst leicht verständlich vorzustellen, um der Projektgruppe Blackout-Restart eine Basis für eine mögliche Anwendung zu geben.

### **3.2 Problemstellung und Motivation**

Nimmt man den Stromhandel an einer Strombörse als Beispiel des regulären Strommarktes, so kann ein Ziel der organisatorischen Integration sein, kleinere, dezentrale Energiewandlungsanlagen in die Lage zu versetzen, an einer solche Börse Strom anzubieten. Ein Beispiel hierfür wäre das Produkt "Peakload" an der EPEX Spot SE. [4] Hierbei müssen aber offensichtlich Verpflichtungen bezüglich des in der Zukunft zu liefernden Stroms eingegangen werden. Dabei stoßen die eher prognoseunsicheren Erzeuger aus Erneuerbaren Energien an ihre Grenzen. Es muss also eine Möglichkeit gefunden werden, Schwankungen in der Stromerzeugung auszugleichen. Zu beachten ist hierbei das Potential von prognosesichereren Energiewandlungsanlagen, wie beispielsweise Kraft-Wärme-Kopplungsanlagen.[2] Ziel von COHDA ist es unter anderem auch, die Freiheiten, die solche Anlagen bieten auszunutzen, um Schwankungen ausgleichen zu können. Ein weiterer zu beachtender Punkt ist, dass viele kleinere Energiewandlungsanlagen im Privatbesitz sind. Dies bedingt wiederum eine große Anzahl von Einzelinteressen, die beachtet werden müssen. Gleichmaßen folgen hieraus aber auch Anforderungen an den Datenschutz der einzelnen Betreiber. Das heißt es wird ein Organisationsmechanismus benötigt, der die Prognoseunsicherheit für Stromerzeuger und -abnehmer reduziert, dabei Autonomie und Datenschutz der Anlagenbetreiber berücksichtigt.

## 3.3 Grundlagen von COHDA

### 3.3.1 Voraussetzungen

Grundlage der Heuristik ist ein Verbund von Energieanlagen. Hierdurch wird das Problem der Prognoseunsicherheit entgegengewirkt, da innerhalb des Verbundes Schwankungen kompensiert werden können. Vorbild hierfür ist der Ansatz der Dynamischen Verbundbildung des Forschungsverbundes „Smart Nord – Intelligente Netze Norddeutschland“.[7] Dieser Verbund von Energieanlagen ist vorab zu bilden und wird als gegeben angenommen. Durch COHDA soll der Verbund befähigt werden, gemeinsam ein Wirkleistungsprodukt zu produzieren. Als Wirkleistungsprodukt wird im folgenden die im Planungszeitraum in Form eines bestimmten zeitlichen Verlaufes produzierte Wirkleistung bezeichnet. Diese Wirkleistungsprodukt wird ebenfalls als gegeben vorausgesetzt. Für jede der im Verbund enthaltenen Energieanlagen ist außerdem bekannt, welche Fahrpläne sie im zu planenden Zeitraum fahren könnte. Fahrpläne beschreiben für eine Anlage einen zeitlichen Verlauf der Produktion einer Wirkleistung in einem betrachteten Planungshorizont. Um Nachrichten zwischen den Agenten des Systems austauschen zu können, müssen diese über Kommunikationsnetze miteinander verbunden sein.

### 3.3.2 Ausgestaltung des Multi-Agenten Systems

#### Allgemein

Basis von COHDA ist ein Multi-Agenten System. Dabei wird jede Energiewandlungsanlage des vorab definierten Verbundes durch einen Softwareagenten repräsentiert und gesteuert. Einer der Agenten wird als Sprecher des Verbundes definiert. Dieser dient als Schnittstelle zu externen Systemen und übernimmt im weiteren Verlauf administrative Aufgaben.

#### Wissen

Jeder Agent hat einen Wissensspeicher mit definierten Aufbau. Dieser beinhaltet die Menge der ausführbaren Fahrpläne seiner Energiewandlungsanlage. Diese Menge von realisierbaren Fahrplänen wird als Suchraum bezeichnet. Des Weiteren kennt der Agent die Präferenzen seiner Energieanlage in Form einer lokale Gütefunktion. In dieser sind die Eigeninteressen des Anlagenbetreibers abgebildet. Sowohl der Suchraum, als auch die lokale Gütefunktion sind privates Wissen des Agenten. Im Laufe der Heuristik wird keiner der anderen Agenten Wissen hiervon erlangen.

Zusätzlich hat ein Agent ein öffentliches Arbeitsgedächtnis. Dieses hat drei Bestandteile. Zum einen das dem Verbund vorgegebene, zu produzierende Wirkleis-

tungsprodukt. Dieses ändert sich im hier betrachteten Fall während der Optimierung nicht und wird mit der ersten erhaltenen Nachricht gespeichert.

Außerdem wird ein wahrgenommener Systemzustand im Arbeitsgedächtnis verwaltet. Dieser enthält alle dem Agenten bekannten, aktuellen Fahrplanauswahlen der anderen Agenten im Verbund. Jeder Agent versieht einen Fahrplan, den er einmal ausgewählt hat, mit einem Index. Der aktuellste Fahrplan eines Agenten ist also derjenige mit dem höchsten Index.

Prinzipiell wird eine Sammlung von Fahrplanauswahlen der Agenten des Verbundes als Konfiguration bezeichnet. Eine Konfiguration, die alle im Verbund enthaltenen Agenten berücksichtigt, ist eine vollständige Konfiguration. Dabei wird die Menge der in der Konfiguration mit einer Fahrplanauswahl enthaltenen Agenten als Schlüsselmenge bezeichnet.

Zuletzt enthält das Arbeitsgedächtnis eines Agenten einen Lösungskandidaten. Dies ist diejenige, dem Agenten bekanntem, Konfiguration, die derzeit die höchste globale Güte erreicht. Zu einem Lösungskandidaten wird immer gespeichert, welcher Agenten diesen initial erstellt hat. Tabelle 1.1 fasst das Wissen der Agenten noch einmal zusammen.

**Tabelle 3.1** Wissen der Agenten

Öffentlich	Privat
Wirkleistungsprodukt	Suchraum
Wahrgenommener Systemzustand	lokale Gütefunktion
Lösungskandidat	

Der Unterschied zwischen dem wahrgenommene Systemzustand und dem Lösungskandidaten ist also, dass der wahrgenommene Systemzustand die aktuellsten Fahrplanauswahlen der Agenten beinhaltet, während der Lösungskandidat die besten Fahrplanauswahlen hinsichtlich der globalen Güte beinhaltet.

Tabelle 1.2 fasst die eben erläuterten Begrifflichkeiten noch einmal zusammen.

## Fähigkeiten

Ein Agent hat zudem definierte Fähigkeiten. Diese werden in die Prozesse Wahrnehmung, Entscheidung und Aktion unterteilt.

Der Schritt Wahrnehmung bedeutet, dass ein Agent Nachrichten empfangen und speichern kann. Dabei kann er beurteilen, ob die in den Nachrichten enthaltenen Informationen für ihn relevant sind. Wie diese Selektion verläuft wird im Verlauf dieser Arbeit weiter erörtert.

Anschließend geht der Agent zum Schritt Entscheidung über. Hier geht es für den Agenten darum einen für ihn optimalen Fahrplan auszuwählen. Beachtung finden hierbei die lokale und, wenn so vom Anlagenbetreiber gewollt, die globale Güte.

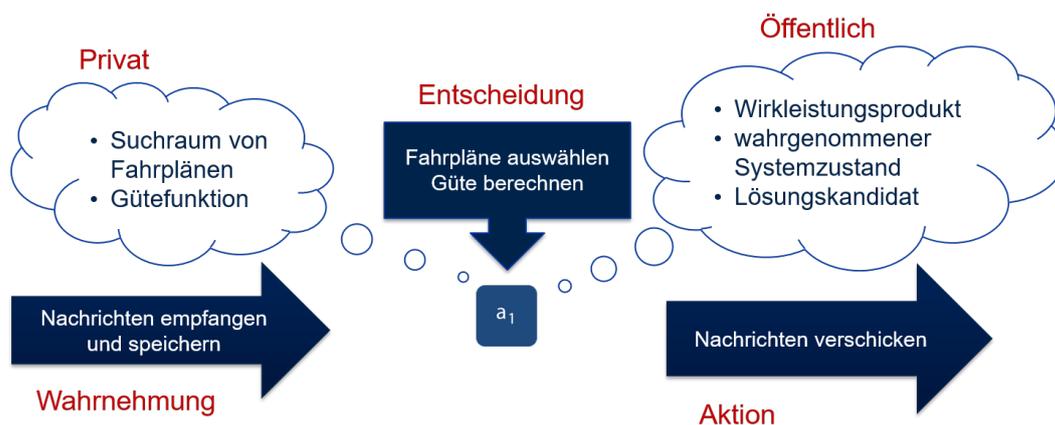
**Tabelle 3.2** Übersicht Definition

Begriff	Bedeutung
Wirkleistungsprodukt	Erbrachte Wirkleistung die einem gewollten Zeitverlauf entspricht
Globale Gütefunktion	Funktion die einem Wirkleistungsprodukt und einem Einsatzfahrplan des Verbundes einen Gütewert zuordnet
Lokale Gütefunktion	Funktion die einem Fahrplan einen Gütewert bezüglich der Präferenzen der Anlage zuordnet
Suchraum	Menge an Fahrplänen die eine Energiewandlungsanlage im vorgegebenen Zeitraum ausführen kann
Fahrplanauswahl	Aktuelle Auswahl des Fahrplans des Agenten. Wird mit einem Laufindex versehen
Konfiguration	Menge von Fahrplanauswahlen von Agenten
Schlüsselmenge	Menge der in einer Konfiguration enthaltenen Agenten
wahrgenommener Systemzustand	Teil des Arbeitsgedächtnisses eines Agenten. Eine Konfiguration mit allen dem Agenten bekannten, aktuellen Fahrplanauswahlen von ihm selber und den anderen Agenten im Verbund
Lösungskandidat	Teil des Arbeitsgedächtnisses eines Agenten. Beste, bezüglich der globalen Güte, dem Agenten bekannte Systemkonfiguration

Den ausgewählten Fahrplan kann er in seinen wahrgenommenen Systemzustand integrieren und diese Konfiguration hinsichtlich ihrer globalen Güte bewerten.

Abschließend, im Schritt Aktion, ist ein Agent in der Lage sein Arbeitsgedächtnis als Nachricht an andere Agenten zu verschicken.

Abbildung 1.1 fasst die Ausgestaltung der Agenten noch einmal zusammen.

**Abb. 3.1** Wissen und Fähigkeiten von Agenten

### 3.3.3 Kommunikation zwischen den Agenten

Ein wichtiger Aspekt ist die Kommunikation der Agenten untereinander. Wie schon beschrieben, kann ein Agent Nachrichten verschicken und empfangen. Die Nachrichten werden dabei über das gegebene Kommunikationsnetz versendet. Um den Nachrichtenaufwand zu reduzieren wird vor Beginn der Konsensbildung ein Overlay-Netz gebildet. Dies ist ein virtuelles Kommunikationsnetz, in dem jedem Agenten direkte Nachbarn zugewiesen werden mit denen er kommunizieren darf. Die eigentliche Kommunikation läuft dabei weiterhin über das bestehende Kommunikationsnetz. Abbildung 1.2. zeigt eine mögliche Reduktion der Kommunikationswege durch ein Overlay-Netz.

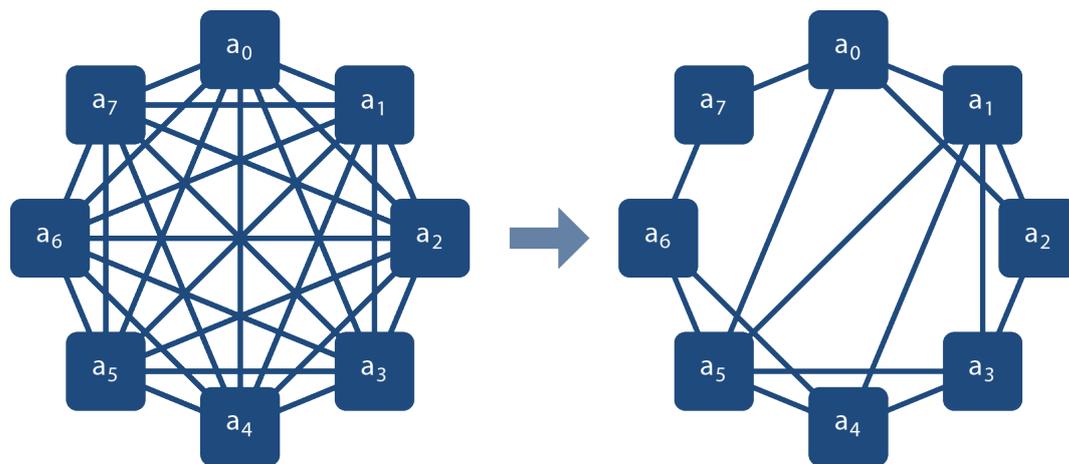


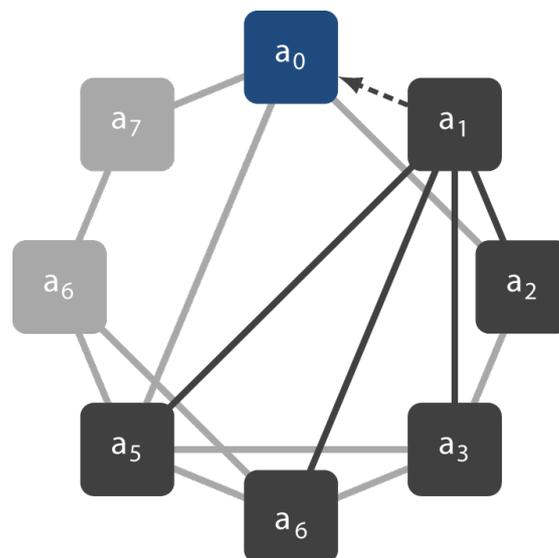
Abb. 3.2 Reduzierung des möglichen Kommunikationsnetzes durch ein Overlay Netz [1]

Trotzdem verteilen sich die Informationen aller Agenten im gesamten Netz. Dies rührt daher, dass eine Nachricht immer aus dem gesamten Arbeitsgedächtnis eines Agenten besteht. Durch den wahrgenommenen Systemzustand und den Lösungskandidaten in dem Arbeitsgedächtnis werden somit auch Informationen über andere im Verbund befindliche Agenten versendet. Abbildung 1.3 zeigt dies beispielhaft für die Kommunikation von Agent 1 zu Agent 0.

## 3.4 Ablauf der Konsensbildung

### 3.4.1 Ablauf der Optimierung

Erhält nun ein Agent durch eine Nachricht das Arbeitsgedächtnis eines anderen Agenten, beginnt er damit, sein Arbeitsgedächtnis zu aktualisieren. Dies ist der Schritt der Wahrnehmung. Dazu betrachtet er einzeln die Inhalte des ihm zugesandten Arbeitsgedächtnisses.



**Abb. 3.3** Agentenübergreifende Informationsweitergabe[1]

### 1. Wirkleistungsprodukt

- 1.1. Hat der Agent bisher kein Wirkleistungsprodukt in seinem Arbeitsgedächtnis gespeichert, so übernimmt er dasjenige aus der erhaltenen Nachricht.

### 2. Wahrgenommener Systemzustand

- 2.1. Wenn in dem erhaltenen Systemzustand Fahrpläne von bisher unbekanntem Agenten enthalten sind, übernehme diese.
- 2.2. Wenn in dem erhaltenen Systemzustand Fahrpläne vorhanden sind, die zwar von bekannten Agenten stammen, aber aktueller sind, so ersetze die alten durch die neuen Fahrpläne.

### 3. Lösungskandidat

- 3.1. Wenn der erhaltene Lösungskandidat Agenten enthält, die bisher nicht im Lösungskandidaten vorhanden waren, wird der Lösungskandidat übernommen.
- 3.2. Wenn der erhaltene Lösungskandidat Agenten enthält, die bisher nicht im Lösungskandidaten vorhanden waren, ihm aber Agenten fehlen, die im eigenen Lösungskandidaten enthalten sind, werden die neuen Elemente übernommen und ein neuer Lösungskandidat erzeugt.
- 3.3. Wenn der erhaltene Lösungskandidat genau die gleichen Agenten enthält, aber eine höhere globale Güte hat, als der eigene Lösungskandidat, wird der neue Lösungskandidat übernommen.
- 3.4. Wenn der erhaltene Lösungskandidat die gleichen Agenten enthält und die gleiche globale Güte hat, wie der eigene Lösungskandidaten, aber von einem Agenten mit einem höheren Index erstellt wurde, wird der neue Lösungskandidat übernommen.

In allen anderen Fällen wird das Arbeitsgedächtnis nicht verändert.

Wurde im vorangegangenen Schritt das Arbeitsgedächtnis modifiziert, so geht der Agent zum Schritt Entscheidung über und optimiert seine eigne Fahrplanauswahl mit den neuen Informationen. Die Art und Weise, wie er dies tut ist im freigestellt. Auch die Priorisierung von lokaler und globaler Güte ist ihm überlassen. Einzig allein verpflichtet sich ein Agent dazu einen einmal veröffentlichten Fahrplan immer wieder anzunehmen, sofern dieser in einer Systemkonfiguration eine höhere globale Güte verspricht, als eine seiner neueren Fahrplanauswahlen.

Hat der Agent dann einen Fahrplan ausgewählt, so prüft er, ob seine wahrgenommene Systemkonfiguration mit seinem neuen Fahrplan eine größere globale Güte hat, als der im Arbeitsgedächtnis gespeicherte Lösungskandidat. Ist dem so, so ist die wahrgenommene Systemkonfiguration ein neuer Lösungskandidat und wird als solcher im Arbeitsgedächtnis gespeichert. Die neue Fahrplanauswahl wird in den wahrgenommenen Systemzustand übernommen. Ist die globale Güte mit der neuen Fahrplanauswahl niedriger, so übernimmt der Agent den ihm im Lösungskandidaten zugeteilten Fahrplan als neuen Fahrplan und speichert diesen in seiner wahrgenommenen Systemkonfiguration.

Abschließend geht er zum Schritt Aktion über und versendet sein Arbeitsgedächtnis an seine ihm zugeordneten Nachbarn. Diese beginnen dann wiederum mit den Schritten Wahrnehmung - Entscheidung - Aktion.

### ***3.4.2 Konvergenz***

Die Informationsverteilung in dem Verbund der Energieanlagen, hin zu einem Konsens, verläuft in drei Schritten.

Der erste Schritt ist, dass alle Agenten ein vollständiges Arbeitsgedächtnis haben. Das bedeutet, dass die Schlüsselmengen des wahrgenommenen Systemzustandes und die des Lösungskandidaten alle im Verbund befindlichen Agenten enthält.

Schritt zwei ist, dass ein Lösungskandidat gefunden wurde, der durch keinen Agenten mehr verbessert werden kann. Da bei der Aktualisierung des Arbeitsgedächtnisses bei gleicher globaler Güte derjenige Lösungskandidat übernommen wird, der den höheren Agentenindex hat, breitet sich genau dieser Lösungskandidat aus.

Im Schritt drei kommt es zu dem Zustand, dass alle Agenten ein identische gefülltes Arbeitsgedächtnis haben. Das bedeutet, dass auch der wahrgenommene Systemzustand jedes Agenten exakt derselbe ist. Hierzu kommt es, da ein Agent dazu gezwungen ist, einen Fahrplan für seine Anlage aus einem Lösungskandidaten zu übernehmen, sofern diese Systemkonfiguration eine höhere globale Güte hat, als die eigene. Diesen Fahrplan übernimmt er dann als seine aktuelle Fahrplanauswahl mit einem neuen Index in seinen wahrgenommenen Systemzustand und versendet dies an die anderen Agenten.

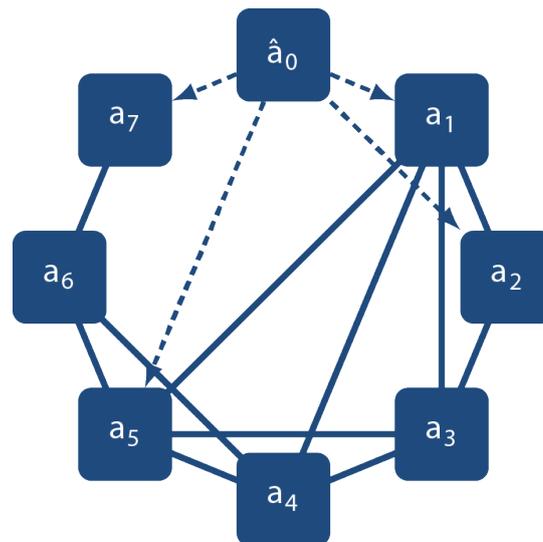
In diesem Zustand sind also wahrgenommener Systemzustand und Lösungskandidaten jedes Agenten übereinstimmend. Hieraus folgt, dass keine neuen Informa-

tionen mehr verschickt werden können. Wenn ein Agent eine Nachricht erhält hat er im Schritt Wahrnehmung keine Veranlassung mehr sein Arbeitsgedächtnis zu ändern. Dadurch werden die Schritte Entscheidung und Aktion übersprungen und keine Nachricht versendet. Das System kommt zum Stillstand und der finale Lösungskandidat ist gefunden. Hierdurch kennt nun jeder Agent denjenigen Fahrplan, den er im Erfüllungszeitraum auszuführen hat.

### 3.4.3 Start der Heuristik

Im Ausgangszustand des Multi-Agenten Systems kennt nur der Sprecher des Verbundes als Schnittstelle zur Außenwelt das vom Verbund zu erbringende Wirkleistungsprodukt. Bevor er aber mit der eigentlichen Optimierung beginnt, ermittelt er das Overlay-Netz und teilt jedem Agenten seine Nachbarn mit, an die dieser seine Nachrichten zu adressieren hat.

Anschließend beginnt der Sprecher seine Schritte Entscheidung und Aktion. Er füllt sein Arbeitsgedächtnis entsprechend seines, auf Basis des ihm bekannten Wirkleistungsproduktes, ausgewählten Fahrplanes. Dieses versendet er dann an seine Nachbarn.



**Abb. 3.4** Der Sprecher versendet sein Arbeitsgedächtnis an seine Nachbarn. Modifiziert nach[1]

Durch diesen Schritt ist die Konsensbildung gestartet, da mit Erhalt der Nachricht die jeweiligen Agenten ihre Schritte Wahrnehmung - Entscheidung - Aktion durchführen.

### 3.5 Ausblick auf die Projektgruppe

COHDA ist eine Heuristik, die darauf zielt mit einem Verbund von Energieanlagen ein gegebenes Wirkleistungsprodukt im Zuge einer prädiktiven Einsatzplanung zu produzieren. Durch die Ausgestaltung ohne zentrale Institution werden dabei Autonomie und Datenschutz der Anlagenbetreiber gewahrt.

Für die Projektgruppe Blackout-Restart gilt es, die Möglichkeiten zu klären, unter denen COHDA für eine Lösung der Aufgabenstellung zum Einsatz kommen könnte. Hierzu werden im Folgenden einige Punkte erläutert.

- **Nachrichtenaufkommen**

Das Vorhandensein von Kommunikationsnetzen ist für den Einsatz von COHDA unerlässlich. Für die Projektgruppe ist vorgegeben, dass das Kommunikationsnetz im Notbetrieb zur Verfügung steht. Hier gilt es anzusetzen und zu klären, in welcher genauen Ausprägung dieses Netz zur Verfügung steht. Auf Basis dessen kann dann beurteilt werden, ob das Nachrichtenaufkommen durch COHDA tragbar ist.

- **Formulierung eines kombinatorischen, verteilten Optimierungsproblems**

Elementar ist ebenfalls, ob ein Optimierungsproblem formuliert werden kann, welches durch COHDA gelöst werden kann. Beispielhaft hierfür ist das gegebene Wirkleistungsprodukt. Dieses ist auch als Ansatz für die Projektgruppe nicht auszuschließen. Wobei es dann zu klären gilt, wie dieses ermittelt werden kann. Offen ist, wie dies unter den Gegebenheiten eines Blackouts geschehen kann.

- **Konvergenz und globale Güte**

Zu klären ist ebenfalls, welcher Zeitraum für die Findung eines finalen Lösungskandidaten akzeptabel ist. Im Zusammenhang dazu steht die Ausgestaltung der globalen Gütefunktion. Der Spielraum, der beim Erreichen des Wirkleistungsproduktes zur Verfügung steht muss ermittelt werden.

- **Bildung des Verbundes**

Hinzu kommt, dass zur Ausführung von COHDA ein Verbund von Energiewandlungsanlagen vorhanden sein muss. Die Zusammenstellung dieses Verbundes scheint sehr interessant für die Projektgruppe zu sein. Vielleicht bietet sich die Möglichkeit, die Ausgestaltung solcher Verbünde hinsichtlich der Prognostizierbarkeit und Stabilität zu optimieren. Hierfür bietet das Forschungsprojekt „Smart Nord – Intelligente Netze Norddeutschland“ mit dem Ansatz der dynamischen Verbundbildung wertvolles Wissen. [7]

- **Ermittlung der Suchräume**

Zu klären ist auch, wie ein Agent Wissen über die ausführbaren Fahrpläne seiner Energieanlage erhält. Gerade bei wetterabhängigen Anlagen wird dies keine triviale Aufgabe sein, deren Ergebnis aber wichtig für die Anwendung von COHDA ist.

- **Verhalten bei einer Störung**

Interessant erscheint auch die in [1, Kap. 3.8] aufgezeigte Möglichkeit der reaktiven Einsatzplanung zu sein. Inwieweit dieses relevant für Blackout-Restart sein wird, kann hier noch nicht abgeschätzt werden.

- **Erhalt von Autonomie**

Über die lokale Gütefunktion können Präferenzen der Anlagebetreiber im Ablauf von COHDA berücksichtigt werden. Fraglich ist, ob die im Rahmen eines Blackouts einschränkbar oder sogar komplett zu vernachlässigen ist.

COHDA bietet eine Heuristik, die für die Projektgruppe nützlich erscheint, um dezentrale Energieanlagen koordinieren zu können. Jedoch gilt es vor der Entscheidung für einen Einsatz des Algorithmus die notwendigen Voraussetzungen von COHDA mit den Gegebenheiten und Anforderungen abzugleichen, um sicher zu gehen, dass der Einsatz auch möglich und sinnvoll ist.

# Literaturverzeichnis

- [1] Christian Hinrichs: Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid. DOI 10.13140/2.1.1792.0008
- [2] Christian Hinrichs, Michael Sonnenschein: A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents. International Journal of Bio-Inspired Computation **8** (2016). (in press)
- [3] Deutsche Bundesregierung: Erneuerbare-Energien-Gesetz vom 21. Juli 2014 (BGBl. I S. 1066), das zuletzt durch Artikel 2 Absatz 10 des Gesetzes vom 21. Dezember 2015 (BGBl. I S. 2498) geändert worden ist
- [4] EPEX Spot SE: DAY-AHEAD-AUKTION. URL <https://www.epexspot.com/de/marktdaten/dayaheadauktion>
- [5] Martin Tröschel: Aktive Einsatzplanung in holonischen VirtuellenKraftwerken
- [6] Sebastian Lehnhoff: Dezentrales vernetztes Energiemanagement: Ein Ansatz auf Basis eines verteilten adaptiven Realzeit-Multiagentensystems. Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden (2010). DOI 10.1007/978-3-8348-9658-2. URL <http://dx.doi.org/10.1007/978-3-8348-9658-2>
- [7] Smart Nord: Smart Nord - Intelligente Netze Norddeutschland. URL <http://www.smartnord.de>

# Kapitel 4

## Kommunikationsnetze/-kanäle/-simulation

Erik Hogen

**Zusammenfassung** Die Arbeit gibt einen Überblick über Kommunikationsnetze im Kontext des Smart Grid mit einem Schwerpunkt auf Simulationsframeworks. Als Herleitung wird kurz die historische Entwicklung von Kommunikationsnetzen skizziert, bevor die wichtigsten Charakteristika von Kommunikationsnetzen erläutert werden. Als integrierter Bestandteil des Smart Grid werden die besonderen Anforderungen an das Kommunikationsnetz dargelegt und die verschiedenen Anwendungsfälle aufgezeigt. Als Entscheidungsgrundlage für die Auswahl eines Simulationsframeworks für Kommunikationsnetze im Rahmen der Projektgruppe wurde ein Kriterienkatalog entwickelt, welcher auf verschiedene Frameworks angewendet wurde, um eine Nutzungs-Empfehlung auszusprechen.

### 4.1 Historische Entwicklung von Kommunikationsnetzen

Kommunikationsnetze finden ihren Ursprung bereits 6000 v. Chr. als die Ureinwohner in Afrika damit begannen sich über längere Distanzen mithilfe von Trommeln auszutauschen. Über die Rauchzeichen der Indianer in Amerika und den Brieftauben im Mittelalter wurde die Kommunikation immer weiter ausgebaut [2]. Mit der Telegrafie und den Morsezeichen wurden Kommunikationsnetze Mitte des 19. Jahrhunderts technisch [19]. Als wichtigste technische Errungenschaften bis in unser heutiges Zeitalter sind dabei die Festnetztelefonie, der Mobilfunk und das Internet zu nennen [3, 17, 18]. Mit der bereits avisierten fünften Mobilfunkgeneration und der Entwicklung des semantischen Web wird das „Internet der Dinge“ in Zukunft zunehmend an Bedeutung gewinnen [25].

## **4.2 Kommunikationsnetze heute**

Nach einer Einführung in die historische Entwicklung von Kommunikationsnetzen und einem Ausblick auf die wichtigsten Trends, wird im Folgenden das Kommunikationsnetz in seiner heutigen Ausprägung dargestellt.

### ***4.2.1 Definition Kommunikationsnetz***

Zu Beginn wird eine Definition des Begriffes Kommunikationsnetz vorgenommen.

Unter einem Kommunikationsnetz versteht man allgemein „ein Transportsystem für den Nachrichtenverkehr zwischen Menschen und Systemen. Ein Kommunikationsnetz stellt die Ende-zu-Ende-Verbindung zwischen den Endteilnehmern beziehungsweise den Teilnehmergeräten zur Verfügung, damit diese miteinander kommunizieren können [14].“

Vereinfacht ausgedrückt stellt ein Kommunikationsnetz die Infrastruktur zum Austausch von Informationen zwischen Teilnehmern über räumliche Entfernungen dar.

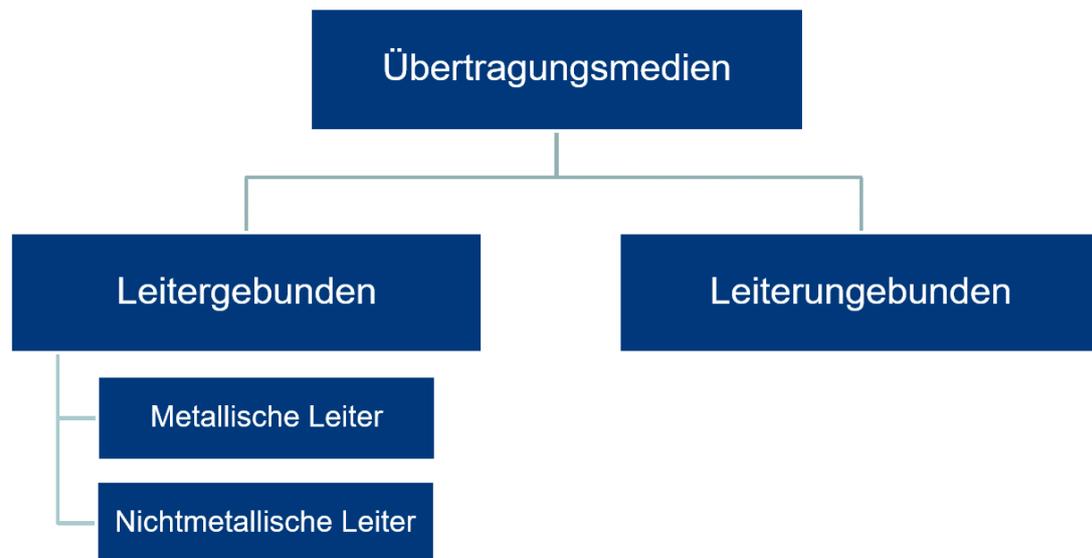
### ***4.2.2 Charakterisierung von Kommunikationsnetzen***

Da Kommunikationsnetze eine Vielzahl von Diensten bereitstellen und diverse Anwendungsfälle unterstützen, besitzen die Netze verschiedene Ausprägungen. In diesem Kapitel werden die wichtigsten Unterscheidungsmerkmale vorgestellt.

#### **4.2.2.1 Übertragungsmedien**

Übertragungsmedien stellen in einem Kommunikationsnetz den Weg dar, auf welchem die Daten übertragen werden [12].

Grundsätzlich unterscheidet man hierbei in leitergebundene und leiterungebundene Übertragungsmedien.



**Abb. 4.1** Übersicht über die Übertragungsmedien in Kommunikationsnetzen<sup>1</sup>

Leitergebundene Übertragungsmedien bieten den Vorteil, dass sie eine hohe Datensicherheit und schnelle Übertragungsgeschwindigkeiten ermöglichen. Allerdings sind die Kosten für das Verlegen der Kabel und das Material selbst sehr hoch. Außerdem bieten sie wenig Flexibilität und es ist keine mobile Nutzung möglich. Die Vorteile der leitergebundenen Übertragungsmedien sind hierbei die Nachteile der leiterungebundenen Übertragungsmedien und umgekehrt.

Leitergebundene Übertragungsmedien finden meist Anwendung in der Raum- und Gebäudeverkabelung, im Telefonnetz und im größten Teil des Internets [13]. Es wird unterschieden in metallische und nichtmetallische Leiter.

Bei metallischen Leitern wird das Informationssignal als nieder- oder hochfrequenter elektrischer Wechselstrom über ein Kabel übertragen. Das am häufigsten genutzte Medium ist dabei das Kupferkabel, da es eine einfache Handhabung bietet und relativ kostengünstig ist. Allerdings ist es störanfällig, z.B. beim Übersprechen auf benachbarte Leitungen. Weiterhin können Kommunikationssignale auch über Stromleitungen übertragen werden, indem das Signal per Trägerfrequenztechnik auf die Leitung aufmoduliert wird. Diese Technik nennt sich Power Line Communication (PLC). Die bekannteste Anwendung hierfür findet sich im haushaltsnahen Bereich und ist das Babyfon. Die Technik ist schnell und kostengünstig zu realisieren, da die Infrastruktur in Form des Stromnetzes meist schon verfügbar ist. In der Praxis haben sich jedoch immer wieder Probleme mit Störfrequenzen gezeigt und außerdem ist die Qualität des Kommunikationssignals abhängig von der Güte der Stromleitungen [23].

<sup>1</sup> Quelle: Eigene Darstellung

Nichtmetallische Leiter nutzen den Lichtweg als Informationsträger. Dabei wird das Informationssignal als elektromagnetische Welle im Frequenzbereich des sichtbaren Lichts innerhalb eines leitenden Materials übertragen [7]. Das verbreitetste Medium ist hierbei das Glasfaserkabel. Es bietet besonders hohe Übertragungsraten und Reichweiten, ist allerdings beispielsweise im Vergleich zum Kupferkabel schwer zu verarbeiten, verursacht hohe Kosten und ist empfindlicher gegen mechanische Belastungen [16].

Bei leiterungebundenen Übertragungsmedien wird das Informationssignal als hochfrequente elektromagnetische Welle durch den freien Raum übertragen. Beispiele hierfür sind Bluetooth, RFID, WLAN, Richtfunk, Mobilfunk oder Satellitenübertragungen [12].

#### 4.2.2.2 Räumliche Ausdehnung

Ein natürlicher Parameter für eine weitere Klassifizierung von Kommunikationsnetzen, ist die geographische Ausdehnung des Netzes [1].

Hierbei wird üblicherweise in vier Kategorien unterschieden.

##### 1. LAN (Local Area Network)

LANs sind meist Firmen- oder Heimnetzwerke. Die Ausdehnung beträgt in der Regel weniger als zehn Kilometer. Die bereitgestellten Zugangsraten sind niedriger als in den Übertragungsnetzen und betragen zurzeit etwa 10-100 Mbit/s.

##### 2. MAN (Metropolitan Area Network)

MANs haben meist eine Ausbreitung über die Fläche einer Stadt (bis zu 50 Km Reichweite) und werden deshalb auch Stadtnetze genannt. Sie verbinden mehrere LANs und nutzen in der Regel die gleichen Techniken wie diese.

##### 3. WAN (Wide Area Network)

WANs sind Fernnetze und überspannen teilweise ganze Länder. Es handelt sich in der Regel um öffentliche Netze und der Betrieb liegt in der Hand der Provider<sup>2</sup>.

##### 4. GAN (Global Area Network)

Ein GAN ist ein weltumspannendes Netz, welches die Rechner und Netzer über mehrere Kontinente hinweg verbindet. Dabei werden z.B. Satelliten und Überseekabel als Übertragungsmedien genutzt. Das bekannteste GAN stellt das Internet dar [12, 21].

#### 4.2.2.3 Vermittlungstechnik

Die Aufgabe der Vermittlungstechnik ist, in einem Kommunikationsnetz einen Übertragungsweg zwischen zwei beliebigen Teilnehmern herzustellen bzw. zu vermitteln. Hierbei wird üblicherweise in Leitungsvermittlung und Speichervermittlung

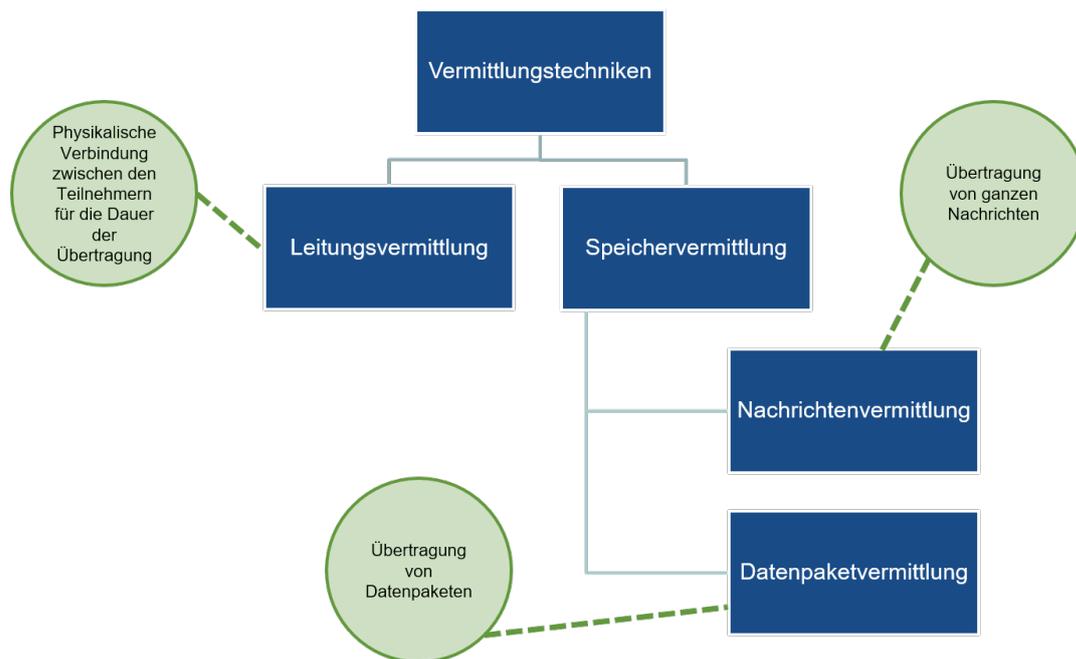
---

<sup>2</sup> Ein Provider ist ein Telekommunikationsdiensteanbieter, welcher für das gewerbliche Erbringen von Telekommunikationsdiensten für die Öffentlichkeit verantwortlich ist

unterschieden, wobei Speichervermittlung nochmals in Nachrichtenvermittlung und Datenpaketvermittlung aufgeteilt ist [9].

In der Leitungsvermittlung wird eine physikalische Verbindung zwischen Sender und Empfänger über eine oder mehrere Vermittlungseinheiten für die Dauer der Übertragung bereitgestellt. Diese Verbindung bzw. dieser Kanal ist für die gesamte Dauer belegt. Übliche Anwendungsfälle für diese Vermittlungstechnik sind Sprach- oder Videoübertragungen [15]. Allerdings ist die Leitungsvermittlung für die Rechnerkommunikation eher ungeeignet, da das Datenaufkommen meist unregelmäßig und der Kanal damit möglicherweise schlecht ausgelastet ist [11].

Die Speichervermittlung ermöglicht es, Nachrichten in einem Zwischenspeicher der Vermittlungsstelle zu speichern, bis ein Weitertransport möglich ist. Eine dauerhafte Verbindung wie bei der Leitungsvermittlung ist dabei nicht notwendig. Die Nachrichten werden einzeln und nacheinander über das Transportnetz übermittelt und belegen den Kanal nur für die Dauer der Übermittlung. In der Nachrichtenvermittlung wird hierbei immer die ganze Nachricht übertragen. Dies kann bei langen Nachrichten problematisch sein, da der Kanal lange belegt ist bevor die nächste Nachricht übertragen werden kann. Zur Übertragung per Datenpaketvermittlung werden die Nachrichten in Pakete zerteilt und einzeln und unabhängig voneinander über das Transportnetz übermittelt. Diese Vermittlungstechnik findet vor allem bei zeitkritischen Anwendungen, wie beispielsweise E-Mail, Anwendung [11].



**Abb. 4.2** Übersicht über die Vermittlungstechniken in Kommunikationsnetzwerken<sup>3</sup>

<sup>3</sup> Quelle: Eigene Darstellung

#### 4.2.2.4 Ausblick auf weitere Klassifizierungsmöglichkeiten

Neben den beschriebenen Charakteristika von Kommunikationsnetzen gibt es noch weitere Klassifizierungsmöglichkeiten, welche im Rahmen dieser Arbeit nicht näher erläutert werden. Der Vollständigkeit halber werden die wichtigsten im Folgenden aufgelistet:

- Simplex/Duplex
- Verbindungsorientiert/-los
- Bestätigt/Unbestätigt
- Synchron/Asynchron
- Temporär/statisch/permanent
- Stationär/quasistationär/mobil
- Nutzerbezogen/Endgerätebezogen
- Nutzung/Tarifierung [8]

### 4.2.3 Standardisierung von Kommunikationsnetzen

Um eine herstellerunabhängige Kompatibilität von Kommunikationsnetzen zu gewährleisten sind internationale Standards notwendig [26]. Im Bereich der Kommunikationsnetze gibt es dafür Kommunikationsprotokolle, welchen einen Satz von Regeln beschreiben, nach denen die Kommunikation zwischen zwei oder mehreren Beteiligten abgewickelt werden muss [24]. Dazu werden im Folgenden die zwei wichtigsten Standards vorgestellt und voneinander abgegrenzt.

#### 4.2.3.1 OSI-Referenzmodell

Das OSI<sup>4</sup>-Referenzmodell beschreibt in mehreren Schichten, welche Voraussetzungen gegeben sein müssen, damit verschiedene Netzwerkkomponenten miteinander kommunizieren können.

Alle Schritte, die vom Sender bis zum Empfänger gemacht werden müssen, werden während der Übertragung in einem Protokoll festgehalten, damit jede einzelne Station auf diesem Weg weiß, wohin das Paket möchte, woher es kommt und welche Eigenschaften es hat. Das OSI-Schichten-Modell sorgt durch diesen Standard dafür, dass in einem Netzwerk Komponenten und Software verschiedener Hersteller miteinander arbeiten können [22].

Die Schichten werden dabei in die transportorientierten Schichten (Schicht 1-4) und die anwendungsorientierten Schichten (Schicht 5-7) unterteilt.

---

<sup>4</sup> OSI = Open System Interconnection

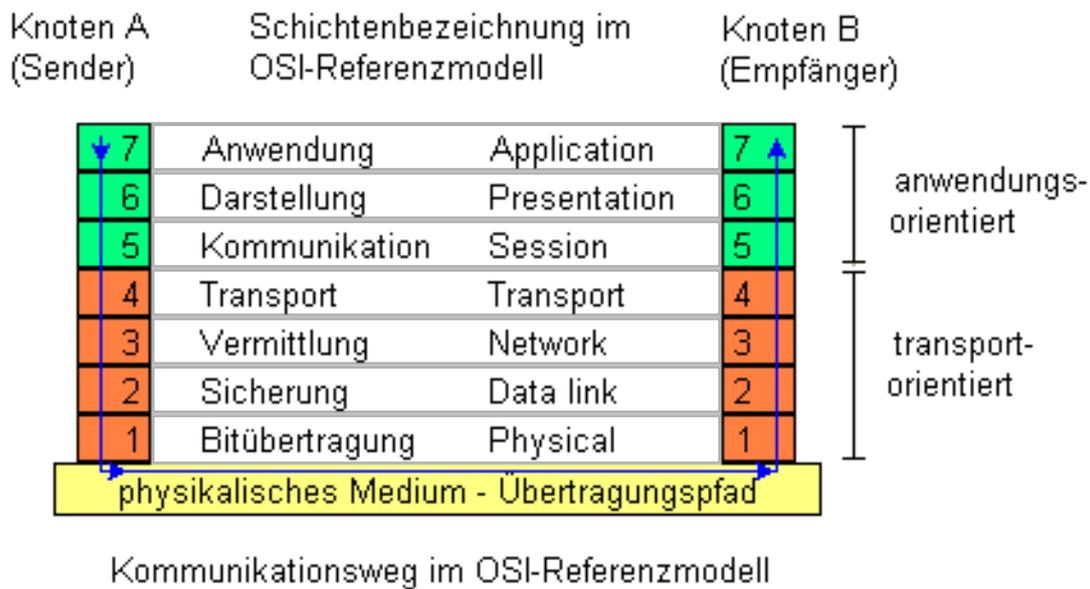


Abb. 4.3 Das OSI-Referenzmodell<sup>5</sup>

#### 4.2.3.2 TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) ist die grundlegende Kommunikationssprache des Internets [27]. Es basiert wie das OSI-Modell auf unterschiedlichen Schichten, welche ebenfalls in anwendungsorientierte und transportorientierte Schichten unterteilt werden können [29]. Allerdings besitzt das TCP/IP-Modell weniger Schichten als das OSI-Modell und ist innerhalb der Schichten teilweise etwas anders ausgeprägt. Grundsätzlich lässt sich TCP/IP als grundlegende Kommunikationssprache dem Internet zuordnen, während OSI allgemein für Kommunikationsnetze Anwendung findet [20].

### 4.3 Kommunikation im Smart Grid

#### 4.3.1 Besondere Anforderungen des Smart Grid

Zur echtzeitfähigen Überwachung und Steuerung der Komponenten innerhalb des Smart Grid sind Kommunikationsnetze unabdingbar und stellen deshalb einen integrierten Bestandteil des Smart Grid dar. Dabei sind folgende Anforderungen an die Kommunikationsnetze besonders relevant [5]:

<sup>5</sup> Quelle: [http://elektroniktutor.de/internet/net\\_pict/osi.gif](http://elektroniktutor.de/internet/net_pict/osi.gif)

- Ausfallsicherheit / Verfügbarkeit im Falle eines Blackouts
- Deterministische Bandbreitenverfügbarkeit
- Echtzeitfähigkeit
- Abdeckung einer sehr großen Zahl an Messsensoren
- Wirtschaftlichkeit der Lösung
- Latenzanforderungen
- Datensicherheit

Je nach Anwendungsfall im Smart Grid sind die genannten Anforderungen unterschiedlich stark gewichtet [10].

### ***4.3.2 Ausprägung von Kommunikationsnetzen im Smart Grid***

Um die Ausprägung von Kommunikationsnetzen im Smart Grid darstellen zu können, werden zuerst die verschiedenen Anwendungsfälle erläutert.

Die Anwendungsfälle können in zwei Kategorien unterteilt werden. Zum einen die verbrauchernahen Applikationen und zum anderen die netzmanagementorientierten Applikationen.

Unter verbrauchernahen Applikationen versteht man:

- Erweiterte (Strom-)Messinfrastruktur / Smart Metering
- Lastmanagement / Demand Side Management
- E-Mobilität / Electric vehicles

Als netzmanagementorientierte Applikationen zählen:

- Dezentrale Energieerzeugung und Energiespeicherung / Distributed Energy Resources and Storage
- Verteilnetzmanagement / Distribution Grid Management [10]

Da im Rahmen der Projektgruppe besonders die netzmanagementorientierten Applikationen eine Rolle spielen, werden im Folgenden die besonderen Charakteristika der Kommunikationsarchitektur für diese Anwendungsfälle dargestellt. Als etablierter Kommunikationsstandard in der Energieverteilung liegt die Verwendung der IEC 61850<sup>6</sup> nahe. Diese bietet sowohl eine gute Grundlage für einen modularen Ausbau als auch die Voraussetzung für eine einheitliche Kommunikations- und Datenstruktur. Als Übertragungsmedien bieten sich leitungsgebundene (Glasfaser, Kupferkabel, PLC) oder drahtlose (GSM, UMTS, LTE, WiMax, Richtfunk, Digitaler Betriebsfunk, Tetra) Technologien an. Die Aspekte der Übertragungsrate, Datensicherheit, Ausfallsicherheit und des Echtzeitverhalten spielen in diesem Umfeld eine wichtige Rolle [4].

---

<sup>6</sup> Die Norm IEC 61850 beschreibt ein allgemeines Übertragungsprotokoll für die Schutz- und Leittechnik in elektrischen Schaltanlagen der Mittel- und Hochspannungstechnik (Stationsautomatisierung)

## 4.4 Simulation von Kommunikationsnetzen

Im Rahmen der Projektgruppe wird die Simulation von Kommunikationsnetzen eine Rolle spielen. Aus diesem Grund wurden innerhalb dieser Arbeit verschiedene Simulationsframeworks für Kommunikationsnetze untersucht. Ziel dieser Untersuchungen war eine Empfehlung je Framework für den Einsatz in der Projektgruppe, welche als Entscheidungsgrundlage dient, sobald die Nutzung innerhalb der Projektphase relevant wird. Dazu wurde ein Kriterienkatalog entwickelt.

Der Kriterienkatalog beantwortet folgende Fragen:

1. Was ist der **Fokus** des jeweiligen Frameworks?
2. Handelt es sich um **kommerzielle** oder **Open-Source** Software?
3. Wie ist die **Bedienbarkeit** der Software? Welcher Aufwand zur Einarbeitung ist zu erwarten?
4. In welcher **Programmiersprache** erfolgt die Simulation?
5. Wie gut ist die **Eignung** des Frameworks im Kontext von Simulationen im Umfeld des **Smart Grid**?

Im Folgenden wird dieser Kriterienkatalog auf die untersuchten Frameworks angewendet. Dazu wird das jeweilige Framework kurz vorgestellt, die Antworten auf den Kriterienkatalog tabellarisch dargestellt und eine abschließende Empfehlung zur Nutzung in der Projektgruppe abgegeben.

### 4.4.1 Network Simulator (ns-2 und ns-3)

Der Network Simulator findet starke Anwendung im Bereich der eventgesteuerten Netzwerksimulation für Zwecke der Forschung und der Lehre. Der Fokus des Frameworks liegt dabei auf Internet-Systemen. Darüber hinaus kommt die Software auch im Bereich des Smart Grid häufig zum Einsatz, insbesondere bei Co-Simulationsansätzen [6].

**Tabelle 4.1** Anwendung des Kriterienkatalog auf das Framework Network Simulator (ns-2 und ns-3) [6]

Fokus	Kommerziell / Open Source	Bedienbarkeit	Programmiersprache	Eignung im Smart Grid
Netzwerksimulation zu Forschungszwecken	Open Source	keine GUI/IDE	C++; Python Integration in ns-3; Auch für Java verfügbar (Java Network Simulator)	Häufiger Einsatz im Smart Grid, gute Eignung bei Co-Simulationsansätzen

Der Network Simulator kann für die Nutzung in der Projektgruppe empfohlen werden. Die Gründe hierfür sind der Fokus auf Forschungszwecke und die kostenfreie Verfügbarkeit.

#### 4.4.2 OMNeT++

Das Open-Source Framework OMNeT++ wurde für die Simulation von Kommunikationsnetzen und verteilten Systemen im Allgemeinen entwickelt. Aufgrund des allgemeinen Designs eignet sich die Software für verschiedene Anwendungsfälle, wie drahtlose Netzwerke, Geschäftsprozessmodellierung oder Peer-to-Peer Netzen. Der Schwerpunkt liegt jedoch bei den Kommunikationsnetzen.

OMNeT++ ist stark verbreitet im akademischen Umfeld und in der Simulation von Kommunikationsnetzen im Smart Grid und findet dort bereits in vielen Testszenarien und Business Cases Anwendung. Es existieren Erweiterungen für verschiedene Modelle bzw. Problemstellungen, welche jedoch teilweise kostenpflichtig sind [6].

**Tabelle 4.2** Anwendung des Kriterienkatalog auf das Framework OmNet++ [6]

Fokus	Kommerziell / Open Source	Bedienbarkeit	Programmiersprache	Eignung im Smart Grid
Schwerpunkt auf Kommunikationsnetze und verteilte Systeme im Allgemeinen, aber auch Geschäftsprozessmodellierung	Open Source	GUI/IDE vorhanden	C++	Sehr verbreitet in der Simulation von Kommunikationsnetzen im Smart Grid in diversen Testszenarien

OmNet++ erhält ebenfalls eine Nutzungsempfehlung, hauptsächlich aufgrund der starken Verbreitung in der Simulation von Kommunikationsnetzwerken im Smart Grid, der kostenfreien Verfügbarkeit und des Vorhandenseins einer grafischen Benutzeroberfläche.

#### 4.4.3 NeSSi (Network Security Simulator)

Das Framework NeSSi (Network Security Simulator) wurde zur Simulation von eventgesteuerten Netzwerk Simulationen vom DAI-Labor (Distributed Artificial Intelligence Laboratory) entwickelt und dabei von der Deutschen Telekom gesponsert.

NeSSi hat einen starken Fokus auf Aspekte, welche die Netzwerksicherheit betreffen. Dieser Ansatz findet auch im Bereich der Kommunikationsnetze im Smart Grid Anwendung. Darüber hinaus bietet NeSSi einen integrierten Ansatz zur Untersuchung und Optimierung von Agentenbasierten Smart Grid Management Systemen [6].

**Tabelle 4.3** Anwendung des Kriterienkatalog auf das Framework NeSSi (Network Security Simulator) [6]

Fokus	Kommerziell / Open Source	Bedienbarkeit	Programmiersprache	Eignung im Smart Grid
Starker Fokus auf Netzwerksicherheit; Häufiger Einsatz bei agentenbasierten Smart Grid Management Systemen	Open Source	GUI/IDE vorhanden	Agent-Framework JIAC (Java)	Geeignet zur Untersuchung von Sicherheitsaspekten in Kommunikationsnetzen und der Performanz Agentenbasierter Smart Grid Management Systeme

Für das Framework NeSSi kann eine eingeschränkte Empfehlung ausgesprochen werden, da ein starker Fokus auf der Netzwerksicherheit liegt. Je nach Ausprägung der Anforderungen in der Projektgruppe kann das Framework dennoch interessant sein, auch deshalb, da es häufig bei der Evaluation und Optimierung agentenbasierter Smart Grid Management Systeme eingesetzt wird.

#### 4.4.4 OPNET Modeler

Der OPNET Modeler dient der Simulation von eventgesteuerten Netzwerken. Er wird in verschiedenen Bereichen der Simulation von Kommunikationsnetzen eingesetzt und bietet dabei bereits eine breite Auswahl an Modellen und unterstützten Testszenarien. Der OPNET Modeler ermöglicht eine einfache Integration von externen Objekten, Bibliotheken und anderen Simulationsframeworks, was eine wichtige Voraussetzung für Co-Simulationsansätze darstellt [6].

**Tabelle 4.4** Anwendung des Kriterienkatalog auf das Framework OPNET Modeler [6]

Fokus	Kommerziell / Open Source	Bedienbarkeit	Programmiersprache	Eignung im Smart Grid
Simulation zur Analyse der Netzwerkperformance	Kommerziell	Visual high level interface	C / C++	Gute Eignung im Kontext von Co-Simulationen; Bietet bereits eine breite Auswahl an Modellen

Der OPNET Modeler würde aufgrund seiner Ausrichtung und Ausprägung zwar eine Rolle für die Nutzung in der Projektgruppe spielen, erhält jedoch aufgrund seiner Kommerzialität keine Empfehlung für einen Einsatz.

#### 4.4.5 NetSim

NetSim wird zur Simulation von Kommunikationsnetzen im Bereich der Verteidigungsindustrie eingesetzt. Es dient speziell der Modellierung und Simulation von Kommunikationsprotokollen im Bereich der Forschung und Entwicklung [28].

**Tabelle 4.5** Anwendung des Kriterienkatalog auf das Framework NetSim [28]

Fokus	Kommerziell / Open Source	Bedienbarkeit	Programmiersprache	Eignung im Smart Grid
Verteidigungsnetze; Forschung und Entwicklung	Kommerziell	GUI vorhanden	C	Keine Eignung für den Einsatz im Smart Grid

Aufgrund des Fokus auf Verteidigungsnetze und der Kommerzialität kann NetSim ebenfalls nicht für einen Einsatz innerhalb der Projektgruppe empfohlen werden.

#### 4.4.6 Fazit

Für die Nutzung in der Projektgruppe kann eine Einsatz-Empfehlung für die Frameworks Network Simulator und OmNet++ ausgesprochen werden. Eine eingeschränkte Empfehlung erhält das Framework NeSSi und der OPNET Modeler und NetSim

sind aufgrund des unpassenden Fokus bzw. der kommerziellen Nutzung nicht für den Einsatz in der Projektgruppe zu empfehlen.

# Literaturverzeichnis

- [1] Awad, G.: LAN-MAN-WAN-GAN. Presentation (o.J.). Online erhältlich unter <http://www1.tfh-berlin.de/~awad/Modul-Klassifizierung.pdf>; abgerufen am 15. Mai 2016.
- [2] BASE, ein Unternehmen der Telefónica Germany GmbH und Co. OHG: Die Geschichte der Telekommunikation – Folge 2. Website (2010). Online erhältlich unter <http://blog.base.de/die-geschichte-der-telekommunikation-2/>; abgerufen am 15. Mai 2016.
- [3] BASE, ein Unternehmen der Telefónica Germany GmbH und Co. OHG: Die Geschichte der Telekommunikation – Folge 4. Website (2010). Online erhältlich unter <http://blog.base.de/die-geschichte-der-telekommunikation-4/>; abgerufen am 15. Mai 2016.
- [4] BDEW Bundesverband der Energie- und Wasserwirtschaft e. V.: Smart Grids in Deutschland - Handlungsfelder für Verteilnetzbetreiber auf dem Weg zu intelligenten Netzen. Website (2012). Online erhältlich unter [https://www.bdew.de/internet.nsf/res/86B8189509AE3126C12579CE0035F374/\\$file/120327%20BDEW%20ZVEI%20Smart-Grid-Broschuere%20final.pdf](https://www.bdew.de/internet.nsf/res/86B8189509AE3126C12579CE0035F374/$file/120327%20BDEW%20ZVEI%20Smart-Grid-Broschuere%20final.pdf); abgerufen am 15. Mai 2016.
- [5] Benigni, A., Monti, A., Ponci, F.: Simulation for the design of smart grid controls. In: 2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS) - at IEEE SmartGridComm 2011, pp. 73–78 (2011)
- [6] Develder, C., Mets, K., Ojea, J.A.: Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis. IEEE COMMUNICATIONS SURVEYS AND TUTORIALS **16**(3), 1771–1796 (2014)
- [7] Elektronik-Kompendium: Übertragungsmedium. Website (o.J.). Online erhältlich unter <http://www.elektronik-kompendium.de/sites/kom/0211192.htm>; abgerufen am 15. Mai 2016.
- [8] Elektronik-Kompendium: Kommunikationstechnik. Website (o.J.). Online erhältlich unter <http://www.elektronik-kompendium.de/sites/>

- [kom/index.htm](#); abgerufen am 15. Mai 2016.
- [9] Elektronik-Kompendium: Vermittlungstechnik. Website (o.J.). Online erhältlich unter <http://www.elektronik-kompendium.de/sites/kom/0211191.htm>; abgerufen am 15. Mai 2016.
- [10] Fraunhofer-Einrichtung für Systeme der Kommunikationstechnik ESK: Smart Grid Communications 2020 - Fokus Deutschland. Website (2011). Online erhältlich unter [http://www.esk.fraunhofer.de/content/dam/esk/dokumente/SmartGrid\\_Studie\\_dt\\_web\\_neu.pdf](http://www.esk.fraunhofer.de/content/dam/esk/dokumente/SmartGrid_Studie_dt_web_neu.pdf); abgerufen am 15. Mai 2016.
- [11] Hübner, U.: Vermittlungsverfahren. Website (1997). Online erhältlich unter [http://www.erasmus-reinhold-gymnasium.de/info/angew\\_inf/Vermittlungsverfahren.htm](http://www.erasmus-reinhold-gymnasium.de/info/angew_inf/Vermittlungsverfahren.htm); abgerufen am 15. Mai 2016.
- [12] Hörnig, V.: Übertragungsmedien in der Kommunikationstechnik. Präsentation (2010). Online erhältlich unter <https://www.rrze.fau.de/ausbildung/berufsausbildung/pdf/uebertragungsmedien.pdf>; abgerufen am 15. Mai 2016.
- [13] ITWissen - Das große Online-Lexikon für Informationstechnologie: Übertragungsmedien - Transmission Medium. Website (o.J.). Online erhältlich unter <http://www.itwissen.info/definition/lexikon/Uebertragungsmedium-transmission-medium.html>; abgerufen am 15. Mai 2016.
- [14] ITWissen - Das große Online-Lexikon für Informationstechnologie: Kommunikationsnetz - Communication Network. Website (o.J.). Online erhältlich unter <http://www.itwissen.info/definition/lexikon/Kommunikationsnetz-communication-network.html>; abgerufen am 15. Mai 2016.
- [15] ITWissen - Das große Online-Lexikon für Informationstechnologie: Vermittlungstechnik - switching technology. Website (o.J.). Online erhältlich unter <http://www.itwissen.info/definition/lexikon/Vermittlungstechnik-switching-technology.html>; abgerufen am 15. Mai 2016.
- [16] Kafka, G.: Kupfer, Glasfaser, Kunststoff und Funk – Für jede Datenübertragung das richtige Medium. Website (2007). Online erhältlich unter <http://tinyurl.com/hesvxre>; abgerufen am 15. Mai 2016.
- [17] Konitzer, A.: Geschichte des Internets. Website (o.J.). Online erhältlich unter <https://www.lmz-bw.de/geschichte-internet.html>; abgerufen am 15. Mai 2016.
- [18] Mobilfunk-Geschichte: Mobilfunk Geschichte. Website (o.J.). Online erhältlich unter <http://www.mobilfunk-geschichte.de>; abgerufen am 15. Mai 2016.
- [19] Mobilfunk-Geschichte: Telegrafie als Teil der Mobilfunk Geschichte. Website (o.J.). Online erhältlich unter <http://www.mobilfunk-geschichte.de/telegrafie.html>; abgerufen am 15. Mai 2016.

- [20] Netzwerk Blog: Vergleich von OSI- und TCP/IP Modell. Website (2011). Online erhältlich unter <http://www.netzwerkblog.com/vergleich-von-osi-und-tcpip-modell/>; abgerufen am 15. Mai 2016.
- [21] Netzwerk-Treff: LAN - MAN - WAN. Website (o.J.). Online erhältlich unter <http://www.netzwerk-treff.de/lanmanwan.htm>; abgerufen am 15. Mai 2016.
- [22] Netzwerke.com: OSI-Schichten-Modell. Website (o.J.). Online erhältlich unter <http://www.netzwerke.com/OSI-Schichten-Modell.htm>; abgerufen am 15. Mai 2016.
- [23] Reder, B.: Kurzeinführung Powerline-Communications: Powerline als Alternative zu WLAN? Website (2009). Online erhältlich unter <http://www.crn.de/netzwerke-storage/artikel-7717.html>; abgerufen am 15. Mai 2016.
- [24] Seitz, P.D.r.n.h.J.: Vermittlungstechnik - Kommunikationsnetze. Presentation (o.J.). Online erhältlich unter [http://zack1.e-technik.tu-ilmenau.de/~webkn/lehre\\_old/VermTechKomNet/V01-Einfuehrung.pdf](http://zack1.e-technik.tu-ilmenau.de/~webkn/lehre_old/VermTechKomNet/V01-Einfuehrung.pdf); abgerufen am 15. Mai 2016.
- [25] Startmobile: Mobilfunk Generationen - Ein Überblick über die Mobilfunk Generationen. . Website (2012). Online erhältlich unter <http://www.startmobile.net/mobilfunk-generationen/>; abgerufen am 15. Mai 2016.
- [26] Technische Universität Ilmenau: Kommunikationsnetze. Presentation (2006). Online erhältlich unter <http://yukon.e-technik.tu-ilmenau.de/~seitz/Kommunikationsnetze-WS0203/V02-Prinzipien.pdf>; abgerufen am 15. Mai 2016.
- [27] TechTarget: TCP/IP (Transmission Control Protocol/Internet Protocol). Website (2008). Online erhältlich unter <http://www.searchnetworking.de/definition/TCP-IP-Transmission-Control-Protocol-Internet-Protocol>; abgerufen am 15. Mai 2016.
- [28] TETCOS: NetSim v9. Website (2016). Online erhältlich unter [http://www.tetcos.com/netsim\\_gen.html](http://www.tetcos.com/netsim_gen.html); abgerufen am 15. Mai 2016.
- [29] Themadbrain: Vergleich zwischen TCP/IP und OSI. Website (o.J.). Online erhältlich unter [http://www.themadbrain.de/Netzwerktechnik/VergleichTCP\\_OSI.htm](http://www.themadbrain.de/Netzwerktechnik/VergleichTCP_OSI.htm); abgerufen am 15. Mai 2016.

# Kapitel 5

## Stabiler Netzbetrieb und Wiederaufbau

Sanja Stark

**Zusammenfassung** Die folgende Arbeit beschäftigt sich mit dem Thema *Stabiler Netzbetrieb und Wiederaufbau*.

Zum Einstieg wird zunächst ein kurzer Überblick über die Grundlagen der Elektrotechnik sowie der Stromerzeugung gegeben.

Anschließend wird im zweiten Teil der Aufbau des Stromnetzes in Deutschland dargestellt, sowie der Begriff der Stabilität erläutert. Davon ausgehend werden mögliche Ursachen für Störungen des stabilen Netzbetriebes behandelt, die schlimmstenfalls zu einem Zusammenbruch führen können.

Schließlich werden die verschiedenen Wiederaufbaumaßnahmen vorgestellt, mit denen nach einem solchen *Blackout* der stabile Netzbetrieb wieder hergestellt werden soll.

Der dritte Teil beschäftigt sich mit der Weiterentwicklung der heutigen Stromnetze. Es werden verschiedene Technologien vorgestellt, mit denen der stabile Netzbetrieb auch unter den in der Zukunft veränderten Bedingungen gewährleistet werden kann. Zum Schluss werden einige aktuelle Forschungsprojekte vorgestellt, die sich mit dem stabilen Netzbetrieb und dem Wiederaufbau zukünftiger Stromnetze beschäftigen.

### 5.1 Grundlagen

Um das Konzept eines stabilen Netzbetriebes besser nachvollziehen zu können, werden in den folgenden Abschnitten zunächst die wichtigsten elektrotechnischen Begriffe, sowie die Vorgehensweise der Stromerzeugung erläutert. [18]

### 5.1.1 Definitionen

#### Strom

Im Kontext der Energieerzeugung bezeichnet Strom die Bewegung von Elektronen in einem leitfähigen Material. Er wird in Ampere (A) gemessen und durch das Formelzeichen  $I$  dargestellt.

#### Spannung

Spannung ist die Arbeit, die aufgebracht werden muss, um Strom zu erzeugen. Sie wird in Volt (V) gemessen und mit dem Formelzeichen  $U$  gekennzeichnet.

#### Widerstand

Jedes Material bremst die Elektronenbewegung zu einem gewissen Grad ab, je nach Leitfähigkeit, Länge und Querschnitt des Leiters. Diese Abbremsung wird als Widerstand bezeichnet und in Ohm ( $\Omega$ ) gemessen. Das zugehörige Formelzeichen ist  $R$ .

#### Ohmsches Gesetz

Die Abhängigkeiten zwischen Strom, Spannung und Widerstand werden durch das Ohmsche Gesetz beschrieben, das wie folgt lautet:

$$U = R \cdot I. \quad (5.1)$$

Nach  $I$  umgestellt besagt diese Formel, dass mehr Strom fließen kann, je größer die Spannung und je kleiner der Widerstand ist.

#### Leistung

Der Begriff *Leistung* umfasst die durch Strom erzeugte elektrische Energie, die ein Verbraucher letztendlich für die Nutzung seiner elektrischen Geräte verwendet. Sie lässt sich mit der Formel

$$P = U \cdot I \quad (5.2)$$

berechnen und wird in der Einheit Watt (W) angegeben.

Die Formel zeigt, dass es für die Höhe der Leistung keine Rolle spielt, ob eine hohe Spannung und niedriger Strom oder niedrige Spannung und hoher Strom verwendet wird. Diese beiden Werte können also in beliebigem Verhältnis zueinander stehen, solange die gewünschte Leistung erreicht wird.

#### Energie

Unter dem Begriff Energie versteht man die in einer bestimmten Zeitspanne erzeugte Leistung. In der Regel wird im Bereich der Stromerzeugung mit Wattstunden (Wh) gerechnet.

#### Verlust

Bei der Übertragung von Strom kommt es immer zu einer Differenz zwischen der im Kraftwerk erzeugten und der beim Verbraucher ankommenden Leistung. Diese

Differenz wird als Verlust bezeichnet. Der Verlust kann berechnet werden durch:

$$P_{\text{Verlust}} = R \cdot I^2. \quad (5.3)$$

Je mehr Strom fließt, desto größer ist der Verlust. Daher ist es für den Transport von Leistung über weite Strecken effizienter, nicht den Strom, sondern die Spannung zu erhöhen.

### **5.1.2 Wechselspannung/Wechselstrom**

Im deutschen Stromnetz wird in der Regel Wechselstrom übertragen, also Strom, der seine Polarität ändert. Damit dies geschieht, muss ebenfalls eine Wechselspannung angelegt werden. Sie sorgt dafür, dass sich die Elektronen nicht in eine Richtung bewegen (vom negativen zum positiven Pol, wie es bei Gleichspannung der Fall wäre), sondern in regelmäßigen Intervallen ihre Richtung wechseln und somit auf der Stelle schwingen. Die Anzahl der Richtungswechsel pro Sekunde wird als Frequenz bezeichnet und in Hertz (Hz) gemessen.

Die Nutzung von Wechselstrom im Stromnetz hat sich aus der Tatsache entwickelt, dass es zu Beginn des Netzaufbaus nicht ohne weiteres möglich war, die bei Gleichstrom anliegende Spannung für die Übertragung zu vergrößern oder zu verkleinern. Aufgrund der Verlustrechnung (5.3) ist es jedoch sinnvoll, Strom zunächst mit hoher Spannung zu übertragen, um die Verluste möglichst gering zu halten.

Spannungsänderungen bei Wechselstrom lassen sich dagegen leicht über Transformatoren realisieren.

Der beim Endverbraucher ankommende Wechselstrom wird je nach Bedarf in den Netzteilen wieder in Gleichstrom umgewandelt, da viele elektrische Geräte nur mit Gleichstrom funktionieren. [18]

Aufgrund des technologischen Fortschritts ist mittlerweile auch die Transformation von Gleichstrom finanzierbar, sodass er vermehrt im Netz eingesetzt wird. Beispielsweise findet die Übertragung der in den Offshore-Windparks erzeugten Leistung zum Festland mittels Gleichstrom statt. [1]

### **5.1.3 Wirk- und Blindleistung**

Neben dem Vorteil der Spannungstransformation, hat die Verwendung von Wechselstrom auch Nachteile.

Im Gegensatz zu Gleichstrom kann bei Wechselstrom die übertragene Leistung nicht automatisch zu 100% vom Verbraucher genutzt werden, sondern teilt sich auf in einen so genannten Wirk- und einen Blindleistungsteil.

Wirkleistung ist die von den Verbrauchern nutzbare Energie, während Blindleistung zwar das Netz belastet, jedoch nicht verwendet werden kann. Sie entsteht durch die

Phasenverschiebung der Frequenzen von Spannung und Strom.

Wechselspannung und Wechselstrom schwingen immer mit der gleichen Frequenz und wechseln dabei zwischen positiven und negativen Werten. Die Leistung berechnet sich aus dem Produkt von Strom und Spannung (s. 5.2). Wenn es keine Phasenverschiebung zwischen Strom und Spannung gibt, sind ihre Werte zu jedem Zeitpunkt entweder beide negativ oder beide positiv, sodass das Produkt immer positiv wird. In diesem Fall erhält man 100% Wirkleistung.

Kommt es jedoch zu einer Phasenverschiebung, die durch Spulen und Kondensatoren im Stromnetz verursacht wird, erhält man zwischenzeitlich sowohl positive als auch negative Werte, deren Produkt ebenfalls negativ wird. Es existiert sozusagen „negative Leistung“, die einen Teil der positiven Leistung resorbiert.

Schlimmstenfalls erhält man so, bei einer Phasenverschiebung von  $90^\circ$ , 100% Blindleistung, da in diesem Fall genauso viel „negative“ wie positive Leistung erzeugt wird.

Ziel ist es grundsätzlich, die Blindleistung im Stromnetz so gering wie möglich zu halten. Es gibt zwei verschiedene Arten von Blindleistung, je nachdem, ob Strom oder Spannung „vorläuft“, also in welche Richtung die Phasenverschiebung stattfindet. Dies hängt wiederum davon ab, ob sie durch einen induktiven (Spule, Transformator, Asynchronmotor) oder durch einen kapazitiven (Kondensator, Erdkabel) Verbraucher erzeugt wird. Man spricht jeweils von induktiver bzw. kapazitiver Blindleistung.

Diese beiden Arten heben sich gegenseitig auf, sodass es durch gezielten Einsatz der einen oder anderen Art möglich ist, den Anteil der Blindleistung im Netz zu reduzieren. [16]

### 5.1.4 Funktionsweise der Kraftwerke

Die meisten Kraftwerke erzeugen Strom mit Hilfe eines Generators (s. Abb. 5.1), der kinetische in elektrische Energie umwandelt. Durch Drehung eines Magneten in einer Spule wird eine Spannung erzeugt, sodass Strom fließen kann.

Der Antrieb des Generators erfolgt entweder direkt, z.B. durch Wasser- oder Windkraft, oder indirekt durch eine Turbine. Durch Wasserdampf oder Gas bringt die Turbine große Räder in Bewegung, die diese Drehbewegung an den Generator weitergeben.

Eine Ausnahme bilden Photovoltaikanlagen, die ohne Umweg über einen Generator Sonneneinstrahlung direkt in elektrische Energie umwandeln.

In den Kraftwerken wird ein Wechselstrom mit einer Frequenz von 50Hz und einer



Abb. 5.1 Übersicht Kraftwerksarten (unvollständig)

Spannung zwischen 10.000 und 60.000 Volt erzeugt. Weil diese Spannung für den Transport über weite Strecken noch zu niedrig ist, wandelt ein Transformator sie vor der Einspeisung in das Stromnetz um. [6] [7]

Um mehrere Erzeuger problemlos in einem Stromnetz zusammenschalten zu können, ist eine einheitliche Frequenz wichtig.

Die in Deutschland übliche Frequenz von 50Hz hat sich, unter Beachtung verschiedener Aspekte, Ende des 19./Anfang des 20. Jahrhunderts etabliert. Während für eine reine Kraftübertragung Werte von 25Hz als ideal angesehen wurden, benötigten Bogenlampen mindestens 42Hz, da sie sonst flackerten, und Transformatoren sogar optimalerweise 60Hz.

Als Kompromiss wurde schließlich 1930 die Normfrequenz von 50Hz festgelegt. [13]

## 5.2 Stromnetze von heute

Die nächsten Abschnitte behandeln das Thema des stabilen Netzbetriebs und Wiederaufbaus im Bezug auf das derzeit in Deutschland bestehende Stromnetz.

### 5.2.1 Netzaufbau

Deutschland ist Teil des europäischen UCTE<sup>1</sup>-Verbundnetzes. Dieses Verbundnetz besteht aus den in Zentraleuropa liegenden Ländern und ist in mehrere so genannte Regelzonen aufgeteilt, wobei üblicherweise eine Regelzone genau ein Land umfasst. Deutschland bildet hier eine Ausnahme, da es 5 Regelzonen enthält.

Jede Regelzone ist über Kuppelstellen mit ihren Nachbarn verbunden, über die Strom in- und exportiert werden kann.

Das Stromnetz teilt sich in vier verschiedene Spannungsebenen auf (s. Tabelle 5.1). Stromtransport über größere Strecken wird auf der Höchst- und Hochspannungsebene in Übertragungsnetzen realisiert, die von Übertragungsnetzbetreibern (ÜNB) überwacht werden. Für jede Regelzone ist genau ein ÜNB zuständig; die in Deutschland arbeitenden ÜNB sind Tennet TSO, 50Hertz Transmission, Amprion und TransnetBW.

Für den Transport des Stroms zu den Verbrauchern sind wiederum Verteilnetze und deren Verteilnetzbetreiber (VNB) verantwortlich. Sie umfassen teilweise die Hochspannung, sowie Mittel- und Niederspannungsebene. [8]

---

<sup>1</sup> Union for the Coordination of Transmission of Electricity

**Tabelle 5.1** Spannungsebenen im UCTE-Verbundnetz

Höchstspannung	220 oder 380 kV	Kohlekraftwerke Atomkraftwerke Offshore-Windparks	
Hochspannung	60 oder 110 kV	Gaskraftwerke Kohlekraftwerke Wasserkraftwerke Windparks	industrielle Großabnehmer
Mittelspannung	3-30 kV	BHKW <sup>a</sup> Biomasseanlagen Wasserkraftanlagen Windenergieanlagen Solarparks	industrielle u. gewerbliche Abnehmer
Niederspannung	230 oder 400 V	kleine Solaranlagen	Privatverbraucher industrielle u. gewerbliche Abnehmer

<sup>a</sup> Blockheizkraftwerk

### 5.2.1.1 Bilanzkreise

Alle Regelzonen sind aufgeteilt in Bilanzkreise. Ein Bilanzkreis besteht sowohl aus Erzeugern als auch aus Verbrauchern und kann als ein virtuelles Energiemengenkonto angesehen werden. Ziel ist es, dieses Konto immer ausgeglichen zu halten, also genau so viel Energie zu erzeugen, wie auch verbraucht wird. Dabei müssen Bilanzkreise jedoch nicht in sich abgeschlossen sein, sondern können beispielsweise zuviel erzeugte Leistung in andere Bilanzkreise übertragen, die dafür weniger produzieren müssen.

Für den Ausgleich zwischen Erzeugung und Verbrauch eines Bilanzkreises ist jeweils ein Bilanzkreisverantwortlicher zuständig. Es werden Last- und Erzeugungsprognosen angefertigt, aus denen so genannte Fahrpläne erstellt werden, die reglementieren, welches Kraftwerk, wann, wieviel Strom erzeugen soll.

Zur Erstellung der Prognosen werden Wetterverhältnisse, Jahreszeit, Wochentag, Tageszeit, besondere Ereignisse und allgemeine Erfahrungswerte berücksichtigt und auf dieser Basis der voraussichtliche Energiebedarf berechnet.[19]

### 5.2.2 Stabiler Netzbetrieb

Ein stabiler Netzbetrieb ist Teil der Systemdienstleistungen, die von den ÜNB bereitgestellt werden müssen. Sie bilden die Grundlage für die Qualität und Versorgungssicherheit eines Stromnetzes.

Die wichtigsten Kriterien für ein stabiles Stromnetz sind eine stabile Spannung sowie eine stabile Frequenz. Beide Werte dürfen nur innerhalb eines festgelegten Bereiches schwanken. In Bezug auf Spannung unterscheidet sich dieser Wertebereich abhängig

von der Spannungsebene. Sie darf jeweils nur  $\pm 10\%$  von der für die jeweilige Ebene festgelegten Nennspannung abweichen. Die Frequenz muss sich dagegen immer in einem Bereich von 50Hz bewegen. [2]

### 5.2.2.1 Spannungsstabilität

Eine stabile Spannung ist wichtig, um die fehlerfreie Arbeit der ans Netz angeschlossenen Geräte zu gewährleisten.

Während eine zu niedrige Spannung lediglich die Funktionalität beeinträchtigt, kann eine zu hohe Spannung irreparable Schäden hervorrufen. Daher ist es die Aufgabe der Netzbetreiber, die Werte in einem Bereich von  $\pm 5-10\%$  um die Sollspannung zu halten.

Die Netzspannung wird durch verschiedene Faktoren beeinflusst. Zum einen gibt es einen natürlichen Spannungsabfall entlang der Stromleitungen, verursacht durch die Widerstände der Betriebsmittel wie Kabel, Schalter und Transformatoren. Zum anderen können auch das An- und Abschalten von Verbrauchern oder Störungen im Netz Spannungsschwankungen verursachen.

Maßnahmen zur Spannungserhaltung können entweder direkt im Kraftwerk oder in Form von regelbaren Transformatoren vorgenommen werden. Auch die gezielte Einspeisung von Blindleistung kann für Spannungsanpassungen genutzt werden. Kapazitive Blindleistung erhöht und induktive Blindleistung verringert die Spannung, weshalb die Stabilität in direktem Zusammenhang mit einer ausgeglichenen Blindleistungsbilanz steht. [14] [5] [17]

### 5.2.2.2 Frequenzstabilität

Im Gegensatz zur Spannung wird die Frequenz nicht von Blind-, sondern von der Wirkleistung beeinflusst. Wird mehr Wirkleistung aus dem Netz bezogen als erzeugt, wird die fehlende Energie von den rotierenden Massen in den Generatoren abgezogen. Die Generatoren werden langsamer und die Frequenz sinkt. Dagegen führt ein Überschuss an Wirkleistung dazu, dass die Generatoren beschleunigt werden und die Frequenz steigt.

Damit ist die Frequenz ein direkter Indikator für das derzeitige Verhältnis zwischen Erzeugung und Verbrauch und eine stabile Frequenz von 50Hz bedeutet, dass dieses Verhältnis sich im gewünschten Gleichgewicht befindet.

Zur Frequenzstabilisierung müssen, je nach Abweichung, Kraftwerke hoch- oder heruntergefahren oder Verbraucher zu- oder abgeschaltet werden. [14] [5]

### 5.2.2.3 Regelenergie

Um kleinere Frequenzschwankungen auszugleichen, gibt es die so genannte Regelenergie. [2]

Man unterscheidet zwischen positiver und negativer Regelleistung, je nachdem, ob mehr Energie produziert oder überschüssige Energie abgefangen werden muss. Hauptanbieter sind regelfähige Kraftwerken, die bei Bedarf ihre Produktion kurzfristig anpassen können.

Zu den weiteren Bezugsmöglichkeiten gehört das Starten schnell anlaufender Kraftwerke (positive Leistung), der Einsatz von Pumpspeicherkraftwerken (positive oder negative Leistung) oder die Trennung von Kunden mit Laststeuerung (negative Leistung). Letztere sind in der Lage, für begrenzte Zeit ihre Arbeitsprozesse ohne Strom fortzuführen, beispielsweise große Kühllhäuser.

Es gibt drei verschiedene Arten von Regelenergie, die unterschiedlich schnell aktiviert werden können und dementsprechend nacheinander zum Einsatz kommen.

**Momentanreserve** bezeichnet den Trägheitseffekt der in den Generatoren rotierenden Massen. Aufgrund des großen Verbundnetzes ist dieser Effekt nicht unerheblich und kleinere Schwankungen werden automatisch abgefangen.

**Primärleistung** muss innerhalb von 30 Sekunden vollständig bereitstehen und bis zu 15 Minuten stabil gehalten werden können. Sie wird von allen Übertragungsnetzbetreibern des Verbundes gemeinsam bereitgestellt und automatisch abgerufen, sobald die Frequenz einen bestimmten Grenzwert über- oder unterschreitet. Sie dient lediglich der Stabilisierung des Systems und der Netzfrequenz auf einem neuen Arbeitswert, ist jedoch nicht in der Lage, die Frequenz wieder auf ihren Sollwert von 50Hz zu bringen.

Dies ist die Aufgabe der **Sekundärleistung**. Sie muss innerhalb von 5min zur Verfügung stehen und sollte nach spätestens 15min die Primärleistung vollständig abgelöst haben, sodass diese dem Netz wieder zur Verfügung steht. Die Sekundärleistung wird zwar ebenfalls automatisch aktiviert, jedoch nicht mehr im kompletten Verbundnetz, sondern nur noch innerhalb der Regelzone, in der die Ursache für die Frequenzabweichung liegt.

Um den gewünschten Frequenzwert zu erreichen, werden die Leistungsflüsse in dieser Zone bzw. von und zu anderen Regelzonen überprüft und Erzeugung und Verbrauch entsprechend angepasst.

**Tertiärleistung** (Minutenreserve) kommt bei länger andauernden Störungen zum Einsatz, wie beispielsweise bei Kraftwerksausfällen oder Prognosefehlern in den Fahrplänen. Sie muss innerhalb von 15min aktiviert und bis zu einer Stunde aufrecht erhalten werden. Die Aktivierung erfolgt nicht mehr automatisch, sondern wird bei Bedarf von dem ÜNB der betroffenen Regelzone manuell angefordert.

#### 5.2.2.4 (n-1)-Sicherheit

Bei einem System mit den Ausmaßen des Verbundnetzes muss berücksichtigt werden, dass es zeitweise zur Störung eines Netzbetriebsmittels oder einer Erzeugungseinheit kommen kann.

Um zu verhindern, dass bereits der Ausfall einer einzelnen Komponente größere Störungen verursacht, muss das so genannte (n-1)-Kriterium beachtet werden. Es besagt, dass trotz eines solchen Ausfalls die Netzstabilität weiter gewährleistet sein

muss. Das Netz soll seine Übertragungs- und Verteilfunktionen weiter ohne Einschränkungen erfüllen können.

Dabei gilt es besonders zu beachten, dass die Grenzwerte der Betriebsmittel zu keinem Zeitpunkt überschritten werden, was zu einer Notabschaltung und damit einer Ausweitung der Störung führen könnte.

Um sicher zu stellen, dass das (n-1)-Kriterium jeder Zeit erfüllt ist, sollte jeder Übertragungsnetzbetreiber in regelmäßigen Abständen eine Berechnung zur Prüfung der (n-1)-Sicherheit in seinem System durchführen. [3]

### **5.2.3 Störungen**

Störungen im Stromnetz können prinzipiell durch zwei verschiedene Ursachen bedingt sein, die in den folgenden Abschnitten vorgestellt werden sollen.

#### **5.2.3.1 Schädigung der Infrastruktur**

Der Begriff Infrastruktur umfasst in diesem Fall alle Betriebsmittel, die zur Übertragung der Leistung benötigt werden, wie Transformatoren, Schaltanlagen, Übertragungsleitungen, Erdkabel, etc.

Schädigungen der Infrastruktur können durch Witterungseinflüsse, Unfälle oder auch gezielte Anschläge verursacht werden. Sobald die Schäden so groß sind, dass die (n-1)-Sicherheit nicht mehr gewährleistet ist, kann die Energie nicht mehr zu den Verbrauchern transportiert werden.

Vorfälle dieser Art sind in der Regel regional begrenzt, sodass es nicht zu einem Zusammenbruch des kompletten Höchstspannungsnetzes kommt. Allerdings muss je nach Größe der Schäden mit einer relativ langen Reparaturzeit gerechnet werden. Als Beispiel sei hier das Schneechaos im Münsterland 2005 genannt, bei dem es durch eine ungünstige Kombination aus Schneefall und Wind zu einer hohen Belastung der Übertragungsleitungen des Hochspannungsnetzes kam. Dies führte zu einem stellenweisen Zusammenbruch der Leitungen, was wiederum eine großflächige Störung verursachte, die sich über fünf Tage erstreckte.

Oberste Priorität hatte die Versorgung der vom Ausfall betroffenen Haushalte mittels Notstromaggregaten und provisorischen Leitungen, ehe die Schäden innerhalb einer Woche repariert und die Mittel- und Niederspannungsnetze wieder vollständig zugeschaltet wurden. [4]

#### **5.2.3.2 Verletzung der Systemparameter**

Als Verletzung der Systemparameter wird das Über- oder Unterschreiten bestimmter Grenzwerte bezeichnet, woraufhin sich Netzbetriebsmittel selbst abschalten und es zu einem großflächigen Zusammenbruch des Netzes auf Höchstspannungsebene

kommt.

Ursache für derartige Störungen sind in der Regel entweder ein Frequenz- oder ein Spannungskollaps. Ein Frequenzkollaps tritt auf, wenn deutlich mehr Leistung aus dem Netz abgezogen als eingespeist wird, ein Spannungskollaps wird in der Regel durch das automatische Abschalten von Teilen des Netzes nach einer Überspannung verursacht.

Auch wenn die Auswirkungen weitreichender sind, als bei einer Schädigung der Infrastruktur, ist die Ausfallzeit generell kürzer. Da keine aufwendigen Reparaturen vorgenommen werden müssen, kann unmittelbar nach dem Zusammenbruch mit dem Aufbau des Netzes begonnen werden.

Für diese Art von Netzzusammenbruch existierten Wiederaufbaukonzepte, die regelmäßig in Simulationen getestet und angepasst werden.

Ein Beispiel für eine derartige Störung ist der Stromausfall 2006 in Europa, der im Emsland durch das geplante Abschalten einer Überlandleitung verursacht wurde. Aufgrund unzureichender Prüfung und Kommunikation der beteiligten Übertragungsnetzbetreiber wurde der Strom falsch umgeleitet, was zur Überlastung einer Leitung führte, die sich daraufhin selbst abschaltete und eine Kettenreaktion auslöste.[3]

### 5.2.3.3 5-Stufen-Plan

Der 5-Stufen-Plan ist eine Vorgehensweise zur Netzstabilisation bei Unterfrequenz. Er kommt zum Einsatz, wenn die Frequenz soweit gesunken ist, dass die Regelenergie nicht mehr ausreicht, um sie wieder zu stabilisieren. [2] Da eine Unterfrequenz

**Tabelle 5.2** 5-Stufen-Plan [2]

Stufe 1	49,8 Hz	Alarmierung des Personals und Einsatz der noch nicht mobilisierten Erzeugungsleistung auf Anweisung des ÜNB, Abwurf von Pumpen.
Stufe 2	49,0 Hz	Unverzögerter Lastabwurf von 10 - 15 % der Netzlast.
Stufe 3	48,7 Hz	Unverzögerter Lastabwurf von weiteren 10 - 15 % der Netzlast.
Stufe 4	48,4 Hz	Unverzögerter Lastabwurf von weiteren 15 - 20 % der Netzlast.
Stufe 5	47,5 Hz	Abtrennen aller Erzeugungsanlagen vom Netz.

durch einen zu hohen Verbrauch verursacht wird, werden nach und nach Verbraucher vom Netz genommen. Sollte dies nicht ausreichen, werden schließlich auch die Erzeugungsanlagen vom Netz getrennt, da ab einer Frequenz von 47,5 Hz Schäden an den Generatoren auftreten können. [11] In Folge dessen kommt es zu einem Black-out.

Im Falle einer Überfrequenz werden, statt einer Abtrennung von Verbrauchern, Erzeugungsanlagen abgeschaltet.

### **5.2.4 Wiederaufbau**

Um nach einem *Blackout* das Stromnetz möglichst schnell und reibungslos wiederaufbauen zu können, sind vorausschauende Planungen wichtig. Neben den am Simulator trainierten Wiederaufbaukonzepten, müssen auch Teile des Netzes auf einen Wiederaufbau ausgelegt sein.

Viele Kraftwerke benötigen zum Hochfahren aus dem ausgeschalteten Zustand sehr viel Energie, die sie im Normalbetrieb aus dem funktionierenden Stromnetz beziehen würden. Steht diese Energie bei einem Blackout nicht zur Verfügung, benötigt man Kraftwerke, die trotz fehlender Stromzufuhr von außen lauffähig sind, und somit die erste Energie für einen Wiederaufbau bereitstellen können. [2]

#### **5.2.4.1 Kraftwerke im Eigenbedarf**

Ein Kraftwerk kann sich im Eigenbedarf „fangen“, wenn es nach einem Blackout in der Lage ist, seine Produktion soweit herunterzufahren, dass es genau soviel Leistung produziert, wie es für die eigenen wichtigsten Funktionalitäten benötigt. Von diesem Zustand aus kann es dann seine Produktion wieder erhöhen, wenn Verbraucher dem Netz zugeschaltet oder Energie für den Start eines anderen Kraftwerks benötigt wird. Im Fall von Kernkraftwerken ist die Eigenbedarfssicherung auch im Hinblick auf die Sicherheitsvorkehrungen wichtig, da hier beispielsweise ein Ausfall der Kühlung schwerwiegende Folgen haben kann.

#### **5.2.4.2 Schwarzstartfähige Kraftwerke**

Als schwarzstartfähig bezeichnet man ein Kraftwerk, das aus dem abgeschalteten Zustand ohne Energiezufuhr von außen gestartet werden kann, wie zum Beispiel Wasserkraft- oder Druckluftspeicherkraftwerke.

In der Regel sind schwarzstartfähige Kraftwerke auch gleichzeitig fähig im Eigenbedarf zu arbeiten, sodass sie nach dem Hochfahren zunächst in eine Haltephase übergehen.

#### **5.2.4.3 Kommunikation**

Um das Stromnetz wieder aufbauen zu können, reicht es jedoch nicht, nur die Energieerzeugung sicherzustellen. Das Zusammenschalten der einzelnen Netzteile erfordert viel Kommunikation, sowohl zwischen den einzelnen Betriebsmitteln, als auch

zwischen den ÜNB. Daher ist es wichtig, im Falle eines *Blackouts*, einen Notbetrieb der Kommunikationskanäle sicherzustellen. Ebenso muss die Funktionalität der Leitstellen und Schaltanlagen für einen Wiederaufbau gewährleistet sein.

#### **5.2.4.4 Inselnetzfähigkeit**

Inselfähige Kraftwerke können die Frequenz innerhalb eines abgeschlossenen Inselnetzes selbstständig stabil halten, sowie die plötzliche Lastzunahme bei Hinzuschalten eines weiteren Kraftwerkes ausregeln.

Die Vorgehensweise des Wiederaufbaus unterscheidet sich je nach Ausgangssituation nach der Störung. Mögliche Netzzustände lassen sich grob in vier verschiedene Varianten gliedern, die im Folgenden näher beschrieben werden sollen.[15]

#### **5.2.4.5 Regionale Störung**

Bei einer regionalen Störung kommt es nur in einem begrenzten Teil des Netzgebietes zu einem Ausfall, der Rest des Netzes ist nicht betroffen und befindet sich im Normalbetrieb. Eine solche Störung kann beispielsweise durch die in Kapitel 1.2.3 beschriebenen Schädigung der Infrastruktur hervorgerufen werden.

Hier muss individuell nach Rahmenbedingungen und Ausmaß gehandelt werden. Dazu erfolgt zunächst eine Störanalyse, ggf. Reparaturarbeiten und anschließend das schrittweise Zuschalten der spannungslosen Betriebsmittel.

#### **5.2.4.6 Stabiler Betrieb nach Lastabwurf**

Diese Situation wird erreicht, wenn nach der Durchführung der Stufen 2 bis maximal 4 des 5-Stufen-Plans das Netz wieder einen stabilen Zustand erreicht hat. In diesem Fall ist das Versorgungssystem an sich noch intakt und nur ein Teil der Verbraucher ohne Strom.

Nach einer Störanalyse muss zunächst der offensichtliche Mangel an erzeugter Leistung behoben werden, ehe die Verbraucher sukzessive wieder zugeschaltet werden. Während des Zuschaltens muss die Netzfrequenz genau überwacht und bei Bedarf der Zuschaltvorgang abgebrochen werden, falls die Frequenz erneut zu sinken beginnt.

#### **5.2.4.7 Blackout bei intakten Nachbarnetzen**

Von einem *Blackout* spricht man, wenn das gesamte Gebiet einer Regelzone spannungslos ist, also auf allen Netzebenen und auf allen Leitungen keine Spannung

anliegt und dementsprechend kein Strom fließt. Im Hinblick auf den Wiederaufbau wird unterschieden, ob sich (mindestens) eine benachbarte Regelzone noch im Normalbetrieb befindet, oder ob die umliegenden Netze ebenfalls spannungslos sind. Bevor der eigentliche Wiederaufbau beginnt, wird das Netz in mehreren Schritten durch Schaltvorgänge aufgetrennt:

1. **Abtrennung vom Verbundnetz:** Alle Leitungen, die aus der Regelzone in benachbarte Regelzonen führen, werden getrennt, sodass die vom Blackout betroffenen Zone ein klar abgegrenztes, beobachtbares Netzgebiet bildet.
2. **Vertikale Netztrennung:** Die einzelnen Spannungsebenen werden voneinander getrennt.
3. **Horizontale Netztrennung:** Die einzelnen Spannungsebenen werden wiederum in mehrere Teilnetze aufgeteilt, wobei die Größe so gewählt wird, dass jedes Teilnetz mit einem einzigen Schritt unter Spannung gesetzt werden kann. Das Mittelspannungsnetz ist dabei durch die Trennung vom Hochspannungsnetz bereits in ausreichend kleine Teilnetze zerlegt, sodass sich die horizontale Netztrennung hauptsächlich auf Höchst- und Hochspannungsebene bezieht.

Der Wiederaufbau beginnt nun, indem ein Höchstspannungsteilnetz über die Leitung zu einer benachbarten Regelzone unter Strom gesetzt wird. Das intakte Nachbarnetz gibt somit Spannung und Frequenz vor, sodass beide Netze miteinander synchronisiert sind.

Anschließend werden nach und nach alle Kraftwerke zugeschaltet, wobei man zunächst bei den sich im Eigenbedarf befindlichen Kraftwerken sowie den schnellstartenden beginnt, die dann wiederum Anfahrleistung für andere Kraftwerke bereitstellen können.

Zum Schluss werden die unteren Spannungsebenen zugeschaltet.

Die ÜNB der betroffenen Regelzone sind jeweils nur für das Höchst- und Hochspannungsnetz zuständig, während die Versorgung der Kunden in den Mittel- und Niederspannungsnetzen durch die VNB geschieht.

#### **5.2.4.8 Blackout ohne intakte Nachbarnetze**

Die Vorgehensweise des Wiederaufbaus bei einem Blackout ohne intakte Nachbarnetze ist identisch zu der eben beschriebenen Vorgehensweise. Der einzige Unterschied besteht darin, dass die Spannungsvorgabe für den ersten Teil des Höchstspannungsnetzes nicht von außen stattfindet, sondern von einem schwarzstartfähigen Kraftwerk oder einem Kraftwerk in Eigenbedarf bereitgestellt werden muss.

### **5.3 Stromnetze von morgen**

Die Entwicklung des Stromnetzes hin zu einer größeren Anzahl dezentraler Erzeugungsanlagen und einer vermehrten Einspeisung durch erneuerbare Energien stellt

einige Herausforderungen dar. Die Strategien zum Wiederaufbau des Netzes nach einem Blackout müssen angepasst und die Besonderheiten von erneuerbaren Energien in Bezug auf die Netzstabilität beachtet werden.

Wichtig ist sowohl ein genereller Ausbau des Netzes, um beispielsweise die im Norden von Offshore-Windparks produzierte Leistung in den Süden zu transportieren, als auch der Einsatz von intelligenten Systemelementen zur Optimierung der Netzleistungsfähigkeit, zusammengefasst unter dem Begriff „Smart Grid“.

Im Folgenden sollen Technologien vorgestellt werden, die helfen können, das Stromnetz von morgen stabil zu halten, sowie einige aktuell laufende Forschungsprojekte, die sich mit möglichen Lösungen für die Herausforderungen von Smart Grids beschäftigen.

### ***5.3.1 Intelligentes Stromnetz***

Gemeinsam mit den VNB hat der BDEW<sup>2</sup> diverse technische Komponenten analysiert, die im Hinblick auf die Stabilisierung des Smart Grids am meisten Potential zeigten. [21]

Um den aktuellen Netzzustand beurteilen und bei Abweichungen von Nennwerten rechtzeitig reagieren zu können, bietet sich eine Überwachung der Netzsituation mittels Sensorik an. Die Sensoren können Informationen über Spannung, Strom, Frequenz und Temperatur der Betriebsmittel liefern. Durch Kombination der erfassten Werte mit einer systemoptimierenden Netz-, Einspeise- und Verbrauchssteuerung, kann sich das Stromnetz somit in einem gewissen Rahmen selbst stabilisieren.

Für die Automatisierung des Verteilnetzes ließen sich intelligente Ortsnetzstationen einsetzen. Sie würden eine schnellere Störanalyse und Fehlerortung ermöglichen, die Wartung verbessern, sowie Transparenz über die genauen Lastflüsse von und zu den Verbrauchern bieten. Zusätzlich könnte eine solche Station die aktive Lastverteilung übernehmen und damit helfen, das Netz stabil zu halten.

Eine weitere Möglichkeit zur automatisieren Netzregulation ist die Nutzung von intelligenten Transformatoren. Ein so genanntes Smart IED<sup>3</sup> berechnet die benötigte Transformatorstufe, die verwendet werden muss, um die Spannung im dahinterliegenden Netz stabil zu halten und stellt diese automatisch ein. Besonders im Hinblick auf Windkraft und Solarenergie ist diese Technik interessant, da Sonneneinstrahlung und Wind sehr kurzfristige Hoch- bzw. Tiefwerte erreichen können, was wiederum zu Spannungsspitzen bzw. Spannungsflauten führt.

Da ein solches Smart Grid maßgeblich auf die Kommunikation der beteiligten Geräte angewiesen ist, ist die fehlerlose Übertragung der benötigten Informationen essentiell. Auch, bzw. gerade wenn es zu einem Blackout kommt, muss ein stabiler Austausch gewährleistet sein. Die Vorgaben für einheitliche Kommunikations- und

---

<sup>2</sup> Bundesverband der Energie- und Wasserwirtschaft

<sup>3</sup> Intelligent Electronic Device

Datenstrukturen sowie die Voraussetzungen für eine stabile Kommunikation, sind im Systemstandard IEC<sup>4</sup> 61850 festgelegt.

Ein anderer Bereich, der eher indirekt die Stabilität eines Smart Grid unterstützen könnte, ist die Entwicklung besserer Stromspeichertechnologien.

Da Windkraft- und PV-Anlagen nicht nach Bedarf, sondern nur abhängig von den Wetterverhältnissen produzieren können, kann es leicht zu einer Überproduktion in bedarfsschwachen oder einer Unterproduktion in bedarfsstarken Zeiten kommen.

Um diese asynchrone Produktion abzufangen und das Gleichgewicht zwischen Erzeugung und Verbrauch zu halten, wäre der Einsatz von leistungsfähigen Speichersystemen optimal.

### 5.3.2 Aktuelle Forschung

Ein wichtiges Element zur Stabilisierung der Netzfrequenz ist die in Kapitel 1.2.2 beschriebene Regenergie. Diese wird momentan durch konventionelle Kraftwerke bereitgestellt. Im Hinblick auf das Ziel den Anteil erneuerbarer Energien im Netz zu erhöhen, oder sogar die konventionellen Energien vollständig zu ersetzen, stellt sich die Frage, ob erneuerbare Erzeugungsanlagen ebenfalls regelfähig sein können.

Mit dieser Fragestellung beschäftigt sich das Projekt *Kombikraftwerk 2*, bei dem unter anderem das Fraunhofer IWES<sup>5</sup>, Siemens AG, Enercon, der Deutsche Wetterdienst und die Universität Hannover zusammen arbeiten.

Der Begriff *Kombikraftwerk* bezeichnet eine Zusammenschaltung verschiedener Energieerzeugungsanlagen. Ein solches Kombikraftwerk, nur bestehend aus erneuerbaren Energien Anlagen, bildet die Basis für das Projekt.

Das Vorgängerprojekt *Kombikraftwerk* sollte zeigen, dass es möglich ist, den Strombedarf allein aus erneuerbaren Energien zu decken. Im Folgeprojekt wollen die Beteiligten nun zeigen, dass auch die nötige Netzstabilität durch erneuerbare Energien erreicht werden kann. Das Kombikraftwerk soll per Bereitstellung von Regenergie auf Schwankungen im Netz reagieren.

Dabei wurden zunächst Szenarien für ein rein erneuerbares Stromversorgungssystem berechnet, um den Bedarf an benötigten Systemdienstleistungen, also die Höhe der Regenergie, zu ermitteln. Anschließend wurden Konzepte zur Bereitstellung der Regenergie entwickelt und in Simulationen getestet, ehe in einem Feldtest mehrere erneuerbare Erzeugungsanlagen zusammenschaltet wurden.

Die bisherigen Ergebnisse deuten darauf hin, dass die Bereitstellung von Regenergie durch den alleinigen Einsatz erneuerbarer Energien möglich ist.

Wie in Kapitel 1.2.3 dargelegt, können Störungen im Stromnetz nicht vollständig verhindert werden, weshalb ein Smart Grid nicht nur in der Lage sein muss, das

---

<sup>4</sup> International Electrotechnical Commission

<sup>5</sup> Insitut für Windenergie und Energiesystemtechnik

funktionierende Netz stabil zu halten, sondern es auch nach einem Störfall wieder aufzubauen.

Das Projekt *Netz:Kraft*[9], das unter anderem von den Übertragungsnetzbetreibern in Deutschland geleitet wird, beschäftigt sich mit der Frage, wie erneuerbare Energien in den Netzwiederaufbau eingebaut werden können. Dafür sollen einerseits die bereits vorhandenen Konzepte zum Wiederaufbau weiterentwickelt und auf die Besonderheiten erneuerbarer Energien angepasst und andererseits untersucht werden, ob Ausfälle durch den Einsatz dezentraler Anlagen in Versorgungsinseln verkürzt werden können.

Das Projekt *LINDA* [12] findet in einem kleineren Rahmen statt und entstand aus der Zusammenarbeit des Verteilnetzbetreibers LVN, den Bayerischen Elektrizitätswerken, der Hochschule Augsburg und der Technischen Universität München sowie Spezialisten für Wasserkraft, Automatisierungstechnik, Gas- und Diesellaggregaten und Leittechnik.

Es beschäftigt sich mit dem stabilen Betrieb von Inselnetzen bei Großstörungen. Die im Projekt betrachtete Ausgangslage beinhaltet einerseits einen hohen Anteil dezentraler, erneuerbarer Erzeugungsanlagen innerhalb der Insel, sowie das Fehlen einer eigenen Kommunikationsinfrastruktur.

Die Problematik besteht darin, dass dezentrale Erzeugungsanlagen, wie beispielsweise PV-Anlagen, automatisch mit ans Netz angeschlossen werden, sobald der Verbraucher hinzugeschaltet wird, auf dessen Dach sich diese Anlage befindet. Dies erschwert die Stabilisierung des Netzes, da es nicht möglich ist, Verbraucher und Erzeuger getrennt voneinander zu behandeln.

Ein möglicher Ansatz wäre, die Kraftwerke in einem solchen Fall mit leicht erhöhter Frequenz laufen zu lassen, da sich PV-Anlagen erst ab einer Frequenz unter 50,05 Hz mit dem Netz synchronisieren. Auf diese Weise könnte man das Inselnetz erst über die zentralen Kraftwerke stabilisieren, ehe man die dezentralen Erzeuger hinzunimmt und die Leistung der Kraftwerke entsprechend anpasst. [10]

Ein Feldversuch soll in der Gemeinde Niederschönenfeld in Bayern durchgeführt werden. Hier wird zunächst ein Blackout simuliert und anschließend ein stabiles Inselnetz aufgebaut. Dabei soll es drei Versuche unterschiedlichen Umfangs geben.

1. Versuch: Zunächst wird nur ein Wasserkraftwerk hochgefahren und mit einer Biogasanlage synchronisiert. Beide Kraftwerke laufen dann im Eigenbedarf.
2. Versuch: Zu den beiden laufenden Kraftwerken werden alle privaten Haushalte sowie die dort angeschlossenen PV-Anlagen zugeschaltet.
3. Versuch: Das stabile Inselnetz soll um ein weiteres Wasserkraftwerk erweitert werden, zum Beweis, dass es möglich ist, das Netz mit weiteren Teilen zu synchronisieren und dabei stabil zu halten.

Mit einem anderen Ansatz arbeitet das Projekt *Kickstarter*. [20]

Dieses Projekt entstand durch Zusammenarbeit der Energieversorgung Schwerin, Uni Rostock, dem Energieversorger WEMAG AG sowie dem Technologieunternehmen Younicos AG.

Grundlage bildet ein von WEMAG und Younicos gebauter Batteriespeicher, der be-

reits regelfähig ist und nun um die Fähigkeit zum Schwarzstart erweitert werden soll. Ziel ist es, ausgehend von diesem Batteriespeicher, nach einem Blackout ein stabiles Inselnetz aufzubauen.

## 5.4 Fazit

Ein stabiles Netz ist dann gewährleistet, wenn das Verhältnis zwischen Stromerzeugung und Stromverbrauch ausgeglichen ist. Die Sicherstellung eines stabilen Netzbetriebs ist umfangreich und bedarf guter Vorbereitung und Planung. Da es in einem Netz dieser Größenordnung nicht möglich ist, den exakten Verbrauch vorherzusagen, muss jeder Zeit Regelleistung bereitstehen, um Schwankungen auszugleichen.

Durch den vermehrten Einsatz von Windkraft- und Solaranlagen kann es zu größeren, weniger planbaren Schwankungen in der Stromerzeugung kommen. Der zunehmende Anteil dezentraler Erzeugungsanlagen erschwert zudem den Wiederaufbau des Netzes nach einem Blackout, da man nicht mehr von einer rein in den höheren Spannungsebenen stattfindenden Stromerzeugung ausgehen kann.

Aktuelle Forschungsgruppen beschäftigen sich mit verschiedenen Ansätzen, um die heutige Netzstabilität auch in einem System mit verteilter, dezentraler Erzeugung und ohne den Einsatz konventioneller Kraftwerke zu gewährleisten.

Besonders wichtig im Hinblick auf den Wiederaufbau ist die Kommunikation zwischen Erzeugern und Verbrauchern, um einen automatisierten, auf die aktuellen Bedingungen angepassten Aufbau des Netzes zu ermöglichen.

# Literaturverzeichnis

- [1] Bernd Schöne: Die Zukunft gehört dem Gleichstrom. Webseite. URL <http://www.welt.de/wissenschaft/article111446330/Die-Zukunft-gehört-dem-Gleichstrom.html>. Letzter Zugriff 22.05.2016
- [2] Berndt, H., Hermann, M., Kreye, H., Reinisch, R., Scherer, U., Vanzetta, J.: Transmission Code 2007 – Netz- und Systemregeln der deutschen Übertragungsnetzbetreiber. Tech. Rep. Version 1.1, VDN (2007)
- [3] Bundesnetzagentur: Bericht der Bundesnetzagentur über die Systemstörung im deutschen und europäischen Verbundsystem am 4. November 2006 (2006). URL [http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Energie/Unternehmen\\_Institutionen/Versorgungssicherheit/Berichte\\_Fallanalysen/Bericht\\_9.pdf?\\_\\_blob=publicationFile&v=1](http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Energie/Unternehmen_Institutionen/Versorgungssicherheit/Berichte_Fallanalysen/Bericht_9.pdf?__blob=publicationFile&v=1)
- [4] Bundesnetzagentur: Untersuchungsbericht über die Versorgungsstörungen im Netzgebiet des RWE im Münsterland vom 25.11.2005 (2006). URL [http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Energie/Unternehmen\\_Institutionen/Versorgungssicherheit/Berichte\\_Fallanalysen/Bericht\\_12.pdf?\\_\\_blob=publicationFile&v=2](http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Energie/Unternehmen_Institutionen/Versorgungssicherheit/Berichte_Fallanalysen/Bericht_12.pdf?__blob=publicationFile&v=2)
- [5] Crastan, V.: Elektrische Energieversorgung 2, 2 edn. Springer (2009)
- [6] Crastan, V.: Elektrische Energieversorgung 1, 3 edn. Springer (2012)
- [7] Daniela Sasonow: Wie wird eigentlich Strom erzeugt? Webseite. URL <http://blog.vattenfall.de/wie-wird-strom-erzeugt/>. Letzter Zugriff 22.05.2016
- [8] Forschung Stromnetze: Struktur des Stromnetzes. Webseite. URL <http://forschung-stromnetze.info/basisinformationen/struktur-des-stromnetzes>. Letzter Zugriff 22.05.2016
- [9] Heckmann, W.: Netzwiederaufbau unter Berücksichtigung zukünftiger Kraftwerkstrukturen. Webseite. URL <http://www.energiesystemtechnik.iwes.fraunhofer.de/de/>

- [schnelleinstieg-wirtschaft/themen/netzkraft.html](http://schnelleinstieg-wirtschaft/themen/netzkraft.html).  
Letzter Zugriff 22.05.2016
- [10] Kerber, G.: Netzwiederaufbau mit erneuerbaren Energien. Webseite. URL <http://forschung-stromnetze.info/projekte/netzwiederaufbau-mit-erneuerbaren-energien/>. Letzter Zugriff 22.05.2016
- [11] Leuschner, U.: Die Netzfrequenz darf nur minimal schwanken. Webseite. URL <http://www.udo-leuschner.de/basiswissen/SB124-04.htm>. Letzter Zugriff 22.05.2016
- [12] LEW Verteilnetz GmbH: LINDA – Lokale Inselversorgung mit erneuerbaren Energien (2016). URL <https://www.lew-verteilnetz.de/CVP/Downloads/Netzgebiet/Pilotprojekt-LINDA-Flyer.pdf>
- [13] Neidhöfer, G.: Der Weg zur Normfrequenz 50 Hz. Bulletin **17**, 29–34 (2008)
- [14] Piller Group GmbH: Spannungsstabilisierung in Inselnetzen und netzgekoppelten Eigenerzeugungssystemen (2014). URL <http://www.piller.com/de-DE/documents/2132/power-conditioning-of-microgrids-and-co-generation-systems-de-de.pdf>
- [15] Rasch, P.: Netzwiederaufbaukonzept/Training bei E.ON Netz. In: VDE Kongress 2004 Berlin – Innovationen für Menschen Band 1 (2004)
- [16] Rüdiger Paschotta: Blindstrom. Webseite. URL <https://www.energie-lexikon.info/blindstrom.html>. Letzter Zugriff 22.05.2016
- [17] Rüdiger Paschotta: Spannungshaltung. Webseite. URL <https://www.energie-lexikon.info/spannungshaltung.html>. Letzter Zugriff 22.05.2016
- [18] Swissgrid AG: Grundlagen kurz erklärt (2013). URL [https://www.swissgrid.ch/dam/swissgrid/company/publications/de/grundlagen\\_elektrotechnik\\_de.pdf](https://www.swissgrid.ch/dam/swissgrid/company/publications/de/grundlagen_elektrotechnik_de.pdf)
- [19] TenneT: Bilanzkreise. Webseite. URL <http://www.tennet.eu/de/kunden/bilanzkreise.html>. Letzter Zugriff 22.05.2016
- [20] Younicos, WEMAG, Stadtwerke Schwerin, Uni Rostock: Eine Batterie für alle Fälle: WEMAG-Speicher zeigt Schwarzstartfähigkeit (2016). URL [http://www.stadtwerke-schwerin.de/swsr/swsr\\_dms/psfile/docfile/8/2016\\_01\\_2556b4885e22043.pdf](http://www.stadtwerke-schwerin.de/swsr/swsr_dms/psfile/docfile/8/2016_01_2556b4885e22043.pdf)
- [21] ZVEI, BDEW: Smart Grids in Deutschland – Handlungsfelder für Verteilnetzbetreiber auf dem Weg zu intelligenten Netzen (2009). URL [https://www.bdew.de/internet.nsf/res/86B8189509AE3126C12579CE0035F374/\\$file/120327%20BDEW%20ZVEI%20Smart-Grid-Broschuere%20final.pdf](https://www.bdew.de/internet.nsf/res/86B8189509AE3126C12579CE0035F374/$file/120327%20BDEW%20ZVEI%20Smart-Grid-Broschuere%20final.pdf)

# Kapitel 6

## Kommunikationsnetze bei Großstörungen

Peer Vincent Stubbe

**Zusammenfassung** Ziel dieser Arbeit ist es, die Auswirkungen von Großstörungen auf Kommunikationssysteme zu analysieren. Hierfür werden zunächst die Kommunikationsnetze untersucht und dargestellt, und im Anschluss auf die Großstörungen bei Kommunikationsnetze eingegangen. Die zentralen Ergebnisse der Arbeit sind, dass die Kommunikationsnetze nicht ausreichend gegen Großstörungen geschützt sind und ein Blackout somit verheerende Auswirkungen auf die Verfügbarkeit von Kommunikationsnetzen hat.

### 6.1 Einleitung

Moderne Kommunikationssysteme spielen in der Welt, insbesondere in Europa eine große Rolle. Fast alles ist vernetzt und ohne diese Kommunikationssysteme ist ein Leben, wie wir es heute führen, schlicht und ergreifend nicht mehr denkbar. Gerade deshalb sollten die Kommunikationssysteme nicht einfach als selbstverständlich hingenommen werden, schließlich kann es dazu kommen, dass Kommunikationssysteme durch Störungen ausfallen und nicht mehr zur Verfügung stehen. Auf den folgenden Seiten werden die Auswirkungen von Großstörungen auf Kommunikationsnetze analysiert. Dazu wird zunächst auf die verschiedenen Kommunikationsnetze in Deutschland eingegangen, wie zum Beispiel das Internet oder der Mobilfunk. Nachdem die Kommunikationsnetze beschrieben wurden, soll es um den Aufbau der Kommunikationsnetze gehen. Nachdem die Grundlagen geschaffen wurden, wird auf die Großstörungen eingegangen. Es werden die bisherigen Maßnahmen zur Sicherung der Kommunikationsnetze beschrieben sowie die Folgen von Großstörungen für die Kommunikationsnetze. Danach wird auf die zu ergreifenden Maßnahmen bei Großstörungen eingegangen. Zum Schluss wird das Themen mit einem abschließenden Fazit abgerundet.

---

Carl von Ossietzky Universität Oldenburg  
E-mail: [peer.vincent.stubbe@uni-oldenburg.de](mailto:peer.vincent.stubbe@uni-oldenburg.de)

## 6.2 Kommunikationsnetze in Deutschland

In Deutschland gibt es eine Vielzahl verschiedener Kommunikationsnetze. Bevor auf die verschiedenen Arten eingegangen wird, wird noch ein wenig Allgemeines zu den Netzen gesagt.

Ein Kommunikationsnetz ist in der Nachrichtentechnik eine Bezeichnung für die zusammengefassten Merkmale des Nachrichtenverkehrs in einem Nachrichtenetz. Im engeren Sinne ist das Kommunikationsnetz eine Einrichtung oder auch eine Infrastruktur für die Übermittlung von Informationen. Hierfür werden beim Kommunikationsnetz Nachrichtenverbindungen zwischen mehreren Endstellen hergestellt. Wichtig für Kommunikationsnetze ist die Standardisierung von Schnittstellen. In Datennetzen geschieht dies zum Beispiel durch die Orientierung an hierarchisch aufgebauten Architekturmodellen mit standardisierten Protokollebenen. Klassische Beispiele dafür sind das OSI-Schichtenmodell beziehungsweise das TCP-IP Modell (Vgl. [8]).

### 6.2.1 Mobilfunk

Zum Mobilfunk gehören nicht nur Handys. Diese gehören zwar auch zum Mobilfunk dazu, spiegeln aber längst noch nicht die komplette Palette wieder. Neben Mobiltelefonen zählen zum Mobilfunk auch noch in Fahrzeugen eingebaute Wechselsprechgeräte, wie zum Beispiel der "Taxifunk". Auch die Telemetrie, häufig eingesetzt bei Drohnen, die See- und Binnenschiffahrtfunkdienste, sowie der Jedermannfunk, bekannt durch WalkieTalkies oder Bluetooth, sowie der Amateurfunk sind Teilbereiche des Mobilfunks. Neben diesen Bereichen zählt auch die Satellitengestützte Kommunikation zum Mobilfunk.

Der Mobilfunk ist in zwei Bereiche eingeteilt, nämlich den öffentlichen Mobilfunk sowie den nicht öffentlichen Mobilfunk. Zum nicht öffentlichen Mobilfunk gehören der Flugfunk und der Betriebsfunk. Die Nutzer des Betriebsfunks sind beispielsweise die Polizei, die Feuerwehr oder andere Hilfsorganisationen. Diese fasst man unter dem BOS-Funk (Behörden und Organisationen mit Sicherheitsaufgaben) zusammen. Außerdem wird der Betriebsfunk von Taxiunternehmen, Unternehmen des Baugewerbes oder auch von Verkehrsbetrieben genutzt.

Im Gegensatz zum Jedermannfunk oder der Telemetrie ist für das klassische Mobiltelefon eine Infrastruktur notwendig, damit es genutzt werden kann. Diese Infrastruktur besteht aus Kern- und Zugangsnetzen. Die Kernnetze verbinden die verschiedenen Zugangsnetze, während die Zugangsnetze den Teilnehmern, beziehungsweise den Endanwendern den Zugang zum Netz ermöglichen. Im Regelfall handelt es sich bei den Zugangsnetzen um Funknetze, womit die Mobilität der Teilnehmer sichergestellt wird (Vgl. [9]).

### **6.2.2 Festnetztelefonie**

Die Festnetztelefonie umfasst alle öffentlichen, leitungsgebundenen Telefonnetze. Nicht öffentliche Telefonnetze sind in Deutschland beispielsweise die Telefonnetze der Bundeswehr sowie die der Deutschen Bahn AG. Genau wie das Mobilfunknetz teilt sich das Festnetz in Kern- und Zugangsnetze. Die Kernnetze verbinden die verschiedenen Zugangsnetze. Die Zugangsnetze, auch als letzte Meile bezeichnet, verbinden die Endanwender mit den Vermittlungsknoten. Die Endgeräte der Festnetztelefonie lassen sich in zwei Gruppen aufteilen. Jene, mit einem analogen Anschluss und solche, die über den DSL-Anschluss beziehungsweise Voice over IP funktionieren (Vgl. [6]).

### **6.2.3 Internet**

Das Internet besteht durch die Zusammenschaltung unterschiedlicher Netzwerke. Hauptsächlich handelt es sich bei ihnen um Firmennetzwerke, Providernetzwerke, und die die Rechner der Kunden angeschlossen sind, oder Universitäts- und Forschungsnetzwerke. Diese Netze sind durch Router mit dem Backbone-Netz verbunden, einem Kernbereich von Kommunikationsnetzen mit sehr hohen Datenübertragungsraten (Vgl. [7]).

Die Kommunikation zwischen Personen über das Internet ist auf viele Weisen möglich. Die am meisten verwendeten sind Mails, Chat, Videochat, Internettelefonie oder über Foren im Internet.

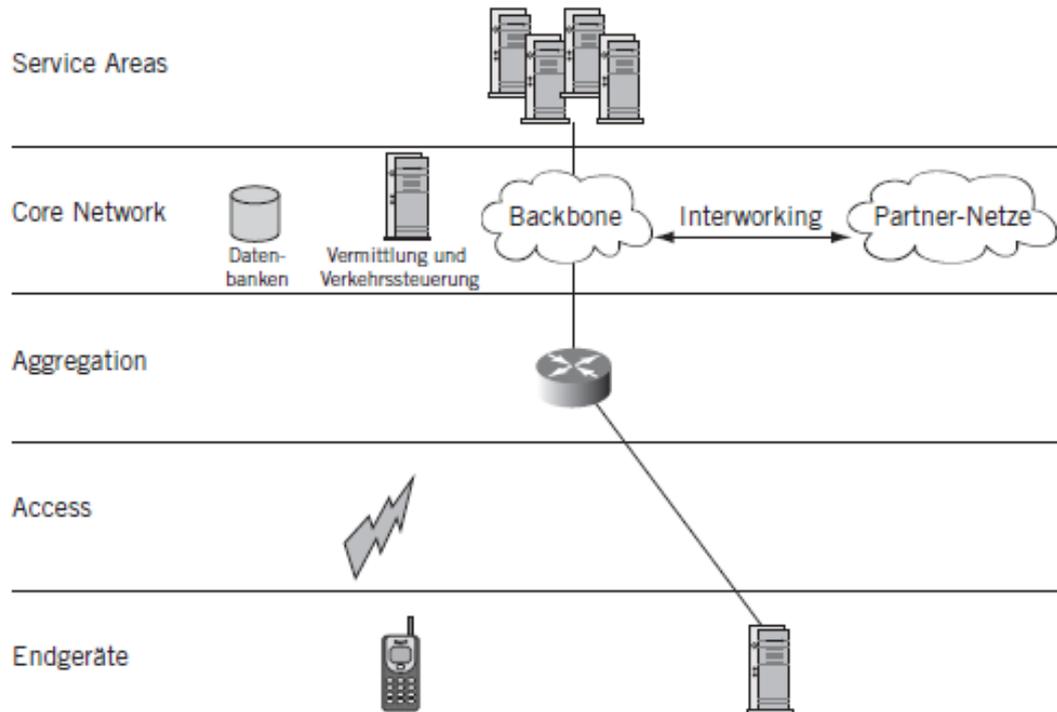
### **6.2.4 Rundfunk**

Der Rundfunk bezeichnet die Übertragung von Informationen jeglicher Art an die Öffentlichkeit. Dies geschieht über den Hörfunk, bei dem ein Radio das Empfangsgerät ist, oder über Bild und Ton, wobei ein Fernsehgerät das Empfangsgerät sein kann. Seit einigen Jahren ist es auch problemlos möglich, die Inhalte des Rundfunks über das Internet zu empfangen. Viele Radio- oder Fernsehsender bieten hierfür Live-streams an. Der große Vorteil beim Rundfunk ist, dass große Teile der Bevölkerung erreicht und mit Informationen versorgt werden können (Vgl. [10]).

## **6.3 Aufbau der Kommunikationsnetze**

Bei modernen Kommunikationsnetzen, auch Next Generation Networks genannt, handelt es sich um hochkomplexe, die ganze Welt umspannende Kommunikationsnetze. Diese Netze sind fast zu 100 Prozent von Strom abhängig, sind besonders

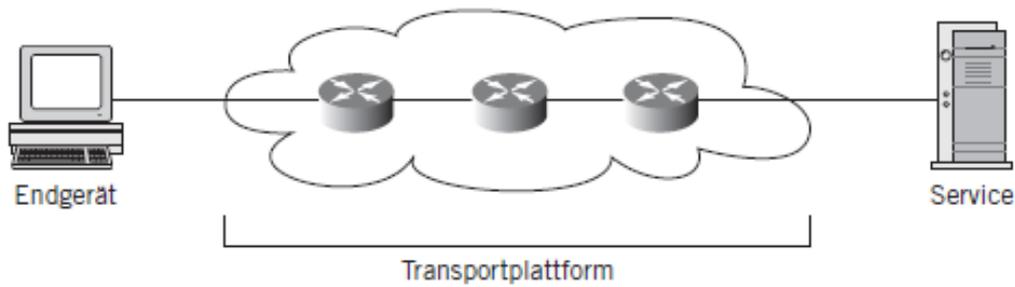
kritische Infrastrukturen und auch eng mit weiteren kritischen Strukturen verwurzelt. Außerdem unterliegen moderne Kommunikationsnetze einer sehr hohen Innovationsdynamik. Die Next Generation Networks verbinden Hard- und Software verschiedener Hersteller und sind nur mit hohem Aufwand an Steuerungs- und Überwachungstechnik, sowie mit geschultem Personal zu betreiben (Vgl. [5]).



**Abb. 6.1** Schema eines Next-Generation-Networks

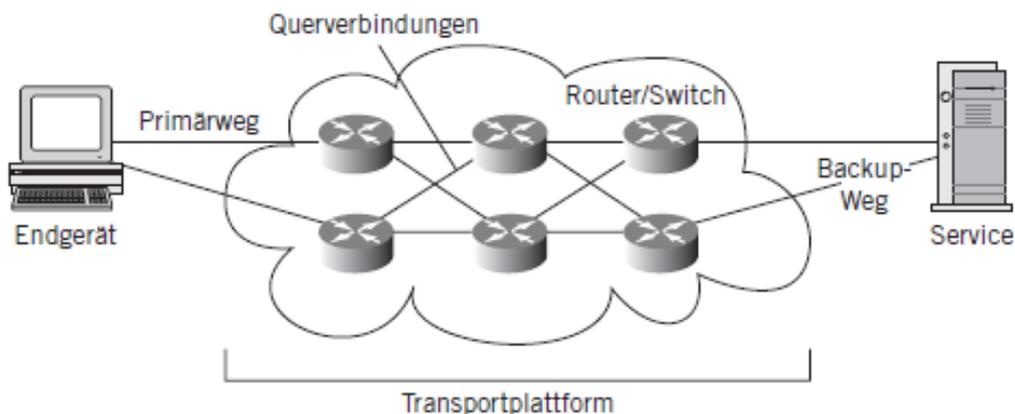
Im Normalfall besteht ein Next Generation Network aus dem Bereich der Endgeräte, welche verschiedene Dienste anfordern und in Anspruch nehmen, dem Bereich der Zugangnetze, bestehend aus Access und Aggregation, einem zentralen Core Network, welches aus einer dienst- und zugangsunabhängigen Plattform besteht, dem Backbone, den Vermittlungs- und Verkehrssteuerungselementen sowie den Netzübergängen zu anderen Netzen, auch Gateway genannt. Außerdem beinhaltet das Next Generation Network zentrale Datenbanken zur Teilnehmerverwaltung und dem Identitätsmanagement. Des Weiteren gibt es den Bereich der Dienstleistung, den Service Areas. Diese Teilbereiche bestehen aus einer Vielzahl von Technologien, die von verschiedenen Herstellern bereitgestellt werden können (Vgl. [5]).

Ein besonders wichtiger Bestandteil des Aufbaus der Kommunikationsnetze ist das Netzdesign. Die Abbildung 6.2 zeigt ein einfaches Modell eines Telekommunikationsnetzes mit hintereinander geschalteten Einzelkomponenten. Die Dienstverfügbarkeit des Gesamtsystems ist bei diesem Design allerdings viel zu gering, da der Ausfall einer Komponente zum Erliegen der gesamten Kommunikation führt. Dies ist der Grund, weshalb dieses Design in der Praxis nicht angewendet wird (Vgl. [5]).



**Abb. 6.2** Transportplattform ohne Redundanz

Durch das Einfügen zusätzlicher Komponenten entstehen Redundanzen. Dadurch wird die Verfügbarkeit höher als bei den Einzelkomponenten, mit dem Ziel einer Verfügbarkeit von nahezu 100 Prozent. Bei den Redundanzen werden die Einzelkomponenten jeweils über einen sogenannten Primärweg als auch über einen Ersatzweg an das übrige Netz angeschlossen. Wichtig dabei ist die räumliche Trennung von Primär- und Ersatzweg, damit nicht beide Wege durch das selbe Störereignis, wie zum Beispiel einem Schneesturm, betroffen werden. Bei den Redundanzen richtet man sich nach dem  $n+1$  Konzept. Für eine Anzahl  $n$  an Netzknoten muss jeweils ein Backup-Knoten bereitstehen. Hierbei ist  $n$  häufig eins.



**Abb. 6.3** Transportplattform mit Redundanz

Durch den Einsatz von Protokollen wird die Umschaltung zwischen Primär- und Ersatzweg vorgenommen. Bei einigen Protokollen gibt es zunächst eine aktive Verbindung, welche 100 Prozent der Last trägt bis diese ausfällt. Dann übernimmt die Ersatzleitung die komplette Last. Ein anderes Protokoll gibt vor, dass sich die Verbindungen die Last von Beginn an aufteilen, jede Verbindung also zunächst 50 Prozent trägt. Wenn eine der Verbindungen ausfällt, übernimmt die andere danach 100 Prozent der Last (Vgl. [5]).

## 6.4 Großstörungen

Eine Großstörung liegt vor bei Spannungslosigkeit im gesamten oder in großen Teilen des Übertragungsnetzes eines Übertragungsnetzbetreibers oder im gesamten oder in größten Teilen des Verteilungsnetzes eines Verteilungsnetzbetreibers oder in mehreren Netzen von benachbarten Netzbetreibern oder in Netzteilen eines oder mehrerer benachbarter Übertragungs- bzw. Verteilungsnetze." [1]

Entstehen kann die Großstörung durch eine Vielzahl menschengemachter, technischer oder klimatischer Ereignisse. Beispiele für die Ursachen sind gravierende Beeinträchtigungen der Netzsteuerungsprozesse durch menschliches oder technisches Versagen, Störungen der technischen Infrastrukturen durch organisierte Kriminalität, ein schweres Naturereignis, wie zum Beispiel Starkregen mit Hochwasser, Sturm, Schnee oder Eis. Auch eine Pandemie ist als Ursache denkbar, wenn durch einen extrem hohen Krankenstand Mitarbeiter zuhause bleiben um Angehörige zu pflegen beziehungsweise selber krank werden. Auch durch die immer dezentraler werdende Stromversorgung, welche nicht immer gleich viel Strom produziert, steigt das Risiko für Ausfälle. Denkbar ist auch ein Angriff durch terroristische Gruppe (Vgl. [2]).

### 6.4.1 Bisherige Maßnahmen zur Sicherung der Kommunikationsnetze

Die bisherigen Maßnahmen zur Sicherung der Kommunikationsnetze sind vielfältig. Sie reichen von gesetzlichen Grundlagen bis zum Netzdesign.

Im Grundgesetz Artikel 87[1] wird verlangt, dass der Bund im Bereich des Postwesens sowie der Telekommunikation flächendeckend angemessene sowie ausreichende Dienstleistungen gewährleistet. Im Telekommunikationsgesetz ist geregelt, dass der Betreiber gewährleisten muss, dass ein Telefonnotruf erfolgen kann. Außerdem sind Anbieter von Telekommunikationsdienstleistungen für die Öffentlichkeit dafür zuständig, „angemessene technische Vorkehrungen oder sonstige Maßnahmen zum Schutze gegen Störungen, die zu erheblichen Beeinträchtigungen von Telekommunikation führen, und gegen äußere Angriffe und Einwirkungen von Katastrophen zu treffen“ (Vgl. [2]).

Ein weiteres wichtiges Gesetz im Katastrophenfall ist auch das AFuG (Amateurfunkgesetz), welches den Amateurfunkdienst als Funkdienst definiert, der unter anderem zur Unterstützung von Hilfsaktionen wahrgenommen werden kann.

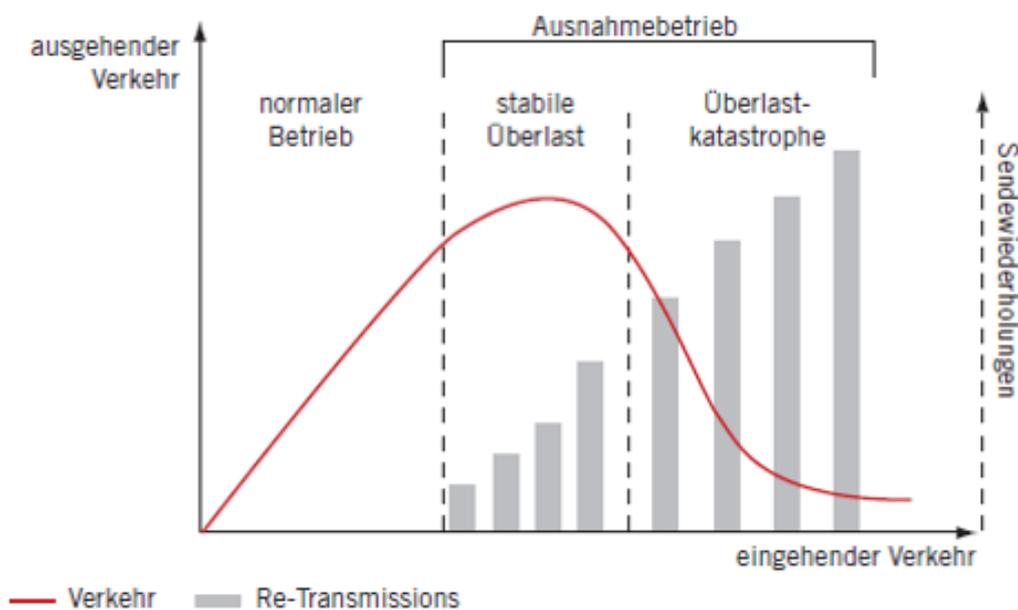
Eine weitere Maßnahme zur Sicherung der Kommunikationsnetze ist das redundante Netzdesign, welches die Verfügbarkeit der Kommunikationsnetze erheblich erhöht.

Außerdem werden die einzelnen Netzknoten durch weitere Maßnahmen geschützt. Diese Maßnahmen sind zum Beispiel eine redundante und stabile Stromversorgung, Notstromaggregate, falls die normale Stromversorgung ausfällt, Überspannungsschutz, Anlagen zur Brandfrüherkennung sowie getrennte Brandabschnitte für

redundante Komponenten, damit diese nicht durch das selbe Feuer beschädigt beziehungsweise zerstört werden (Vgl. [4]).

### 6.4.2 Folgen von Großstörungen für Kommunikationsnetze

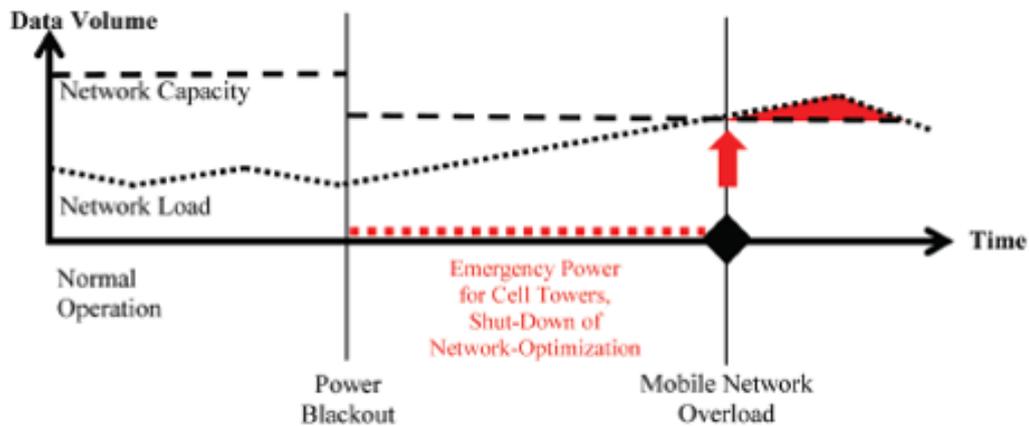
Die Grafik 6.4 zeigt die verschiedenen Betriebszustände eines TK-Netzes. Neben dem normalen Betrieb gibt es noch den Ausnahmebetrieb, der sich in die stabile Überlast sowie in die Überlastkatastrophe aufteilen lässt. Der Ausnahmebetrieb ist notwendig, um kritische Betriebszustände zu beheben. Dies sind zum Beispiel der Ausfall eines gesamten Netzknotens durch Störungen der lokalen Infrastruktur, welcher nicht in einer vertretbaren Zeit zu beheben ist, oder die Überlast eines Netzknotens bzw. des gesamten TK-Systems, welche nicht durch freie Redundanzen zu beheben ist. Zur Überlast kommt es, wenn der Verkehr so stark ansteigt, dass die Lastgrenze des Systems überschritten wird. Wenn das der Fall ist, werden vermehrt Senderwiederholungen gesendet, die die Verarbeitungskapazität weiter vermindern, es kommt zum Domino-Effekt. Je nach Architektur des Systems kann es im Extremfall dazu führen, dass kein Verkehr mehr verarbeitet werden kann und es zum totalen Netzausfall kommt (Vgl. [4]).



**Abb. 6.4** Betriebszustände eines Telekommunikationsnetzes

Abbildung 6.5 zeigt die Folgen eines Blackouts am Beispiel eines Mobilfunknetzes. Die gestrichelte Linie zeigt die Netzkapazität, die gepunktete Linie zeigt die Netzlast. Durch den Blackout muss das Kommunikationsnetz durch Notstrom versorgt werden. Da nicht mehr genügend Energie zur Verfügung steht, kann das Netzwerk nicht mehr ausreichend optimiert werden, sodass die Netzkapazität sinkt.

Gleichzeitig steigt die Netzlast immer weiter an, sodass es zu einer Netzüberlast kommt (Vgl. [3]).



**Abb. 6.5** Blackout mit den Konsequenzen für mobile communication

Allerdings sind nicht nur die Kommunikationsnetze von der Stromversorgung abhängig, sondern zwischen den kritischen Infrastrukturen gibt es wechselseitige Abhängigkeiten. So ist beispielsweise die Stromversorgung auf funktionstüchtige Informations- und Kommunikationssysteme angewiesen. Bewertet wurden die Abhängigkeiten in der Grafik 6.6 mit einem Wert von 0 bis 3. Insgesamt haben die Bereiche Telekommunikation und Informationssysteme auf alle 31 Teilsektoren der schweizerischen kritischen Infrastruktur einen Auswirkungswert von 45 von 93 möglichen, die Stromversorgung einen Wert von 68 (Vgl. [2]). Hieraus ergibt sich für die Schweiz ein Handlungsbedarf. Die gegenseitigen Abhängigkeiten müssen weiter untersucht werden, damit umfassende Konzepte zum Schutz erarbeitet werden können. Des Weiteren wird die Risikokommunikation gefördert, da das Bewusstsein über die Bedeutung kritischer Infrastrukturen und die Folgen von möglichen Störungen häufig unterschätzt werden. Deshalb werden Betreiber von kritischen Infrastrukturen, Akteure aus der Wirtschaft, aber auch die Bevölkerung über die Gefahren und Risiken aufgeklärt.

Ausfall des Teilsektors →	DEPENDENZEN DER TEILSEKTOREN		
Auswirkung auf Teilsektor ↓	Stromver- sorgung	Telekom- munikation	Informations- systeme und -netze
Stromversorgung	–	2	1
Telekommunikation	3	–	3
Informationssysteme und -netze	3	2	–
Internet	3	3	3
Instrumentations-, Automations- und Überwachungssysteme	3	3	3
Rundfunk und Medien	3	2	3
auf alle 31 Teilsektoren	68	45	45

**Abb. 6.6** Abhängigkeiten zwischen verschiedenen kritischen Infrastrukturen

Die Grafik 6.7 zeigt, wann voraussichtlich die verschiedenen Komponenten der Kommunikationssysteme im Falle eines Blackouts ausfallen würden. Bei der Festnetztelefonie werden zuerst die digitalen Endgeräte ausfallen, wenige Stunden später auch die analogen Endgeräte und die Vermittlungsstellen. In den Mobilfunknetzen sind es nicht die Endgeräte, die sofort ausfallen würden, da diese bei geringer Nutzung und aufgeladenem Zustand einige Tage durchhalten können. Das Problem im Mobilfunk wäre die Vermittlungstechnik, die die Einwahl in die Netze ermöglicht. Beim Satellitenfunk liegt der Schwachpunkt wiederum bei den Endgeräten, die je nach Ladezustand und Nutzung ebenfalls einige Tage durchhalten könnten. Von Vorteil beim Satellitenfunk ist, dass sowohl die Vermittlungstechnik als auch die Backbone-netze im Prinzip nicht vom Blackout betroffen sind. Die Nutzung des heimischen Internets würde durch einen Stromausfall direkt unmöglich werden. Wenn der Router zuhause nicht mehr mit Strom versorgt wird, kann man auch nicht mehr ins Internet gehen. Bemerkenswert ist, dass der Funk der Behörden und Organisationen mit sicherheitskritischen Aufgaben insgesamt nur für einige wenige Stunden ausreicht. Prinzipiell könnte man davon ausgehen, dass Polizei, Feuerwehr etc. auf einen Stromausfall vorbereitet sind. Deren Notstromaggregate reichen allerdings nur für zwei Stunden. Die Möglichkeit für Bürgern Informationen über den Rundfunk zu sammeln, wird durch einen Blackout stark eingeschränkt. Fernsehen wird unmöglich, da die Funktionsfähigkeit der Endgeräte direkt vom Strom abhängt. Zum wichtigsten Informationsmedium für die Bürger wird der Hörfunk in Form eines Batteriebetriebenen Radios oder einem Autoradio. Über diese Möglichkeit können im „Idealfall“ sogar über mehrere Wochen Informationen und Nachrichten versendet werden. Im Gegensatz zu den Behörden sind öffentlich-rechtliche Sendeanstalten

verhältnismäßig gut aufgestellt. Problematisch ist, dass der Großteil der Bürger auf einen länger andauernden Stromausfall schlecht, beziehungsweise gar nicht vorbereitet ist, was den Empfang von Nachrichten, abgesehen durch Hörfunk, erschwert bis unmöglich macht (Vgl. [2]).

**TAB. 3** ZEITLICH GESTUFTE AUSFÄLLE IM SEKTOR »INFORMATIONSTECHNIK UND TELEKOMMUNIKATION«

	Endgeräte	Vermittlungstechnik	Backbonenetze
Festnetz (mit analogem Endgerät)	**	**	****
Festnetz (DSL-Anschluss, VoIP)	*	**	****
Mobiltelefon	***	**	****
Satellitenfunk	***	****	****
BOS	***	**	**
Internet	*	**	****
Behördenetze	***	***	***
Fernsehen	*	***	***
Hörfunk (batteriebetriebenes Radio, Autoradio)	****	****	****

**Abb. 6.7** Ausfälle der Kommunikationsnetze bei einem Blackout

### 6.4.3 Maßnahmen bei Großstörungen

Im Fall der Störung eines gesamten Netzknotens wird in erster Linie eine Verkehrs-umleitung auf freie Redundanzen bzw. Reserveknoten vorgenommen. Im Falle einer Überlast im gesamten TK-System kommt es zu einer Lastabwehr. Dadurch wird verhindert, dass die verschiedenen Netzkomponenten noch stärker verlangsamt werden, beziehungsweise, dass sie vollständig zusammenbrechen. Möglichkeiten der Lastabwehr sind die Verlangsamung der Übertragungsgeschwindigkeit, die Abweisung neuer Verbindungsanforderungen oder, am drastischsten, der Lastabwurf, indem bestehende Verbindungen unterbrochen werden, um zu verhindern, dass das System mehr mit Sender-Wiederholungen oder Re-Transmissionen als mit dem eigentlichen Verkehr beschäftigt ist, damit die Überlastkatastrophe abgewendet wird (Vgl. [4]).

Wichtig ist gerade zu Anfang der Störung auch, dass die Bevölkerung genau informiert wird. Dies sollte einerseits über den Hörfunk geschehen, andererseits

bietet es sich an, dass zum Beispiel die Polizei durch die Straßen fährt und über Lautsprecher die Menschen über den aktuellen Zustand und das richtige Verhalten aufklärt. Wenn der Bevölkerung konsequent mitgeteilt wird, dass es zurzeit zwecklos ist, zu telefonieren und sich so Informationen einzuholen, und diese es dann auch nicht mehr machen, werden die noch funktionierenden Kommunikationsnetze nicht noch stärker belastet und brechen nicht zusammen.

## 6.5 Fazit

Je nach Größe der Störung sind die Folgen für die Kommunikationsnetze enorm. Während kleinere Störungen nahezu problemlos durch freie Redundanzen abgefangen werden können, fällt bei einer großen Störung das komplette Netz über kurz oder lang aus. Die mangelnde Vorsorge im Bereich der Kommunikationsnetze für eine Großstörung ist zwar alarmierend, dennoch nicht so dramatisch wie in anderen kritischen Infrastrukturen, wie der Wasserver- und Entsorgung, die ebenfalls größtenteils vom Strom abhängig ist und sich direkt auf das physische Wohlbefinden der Bevölkerung auswirkt.

Es ergibt sich auf jeden Fall ein Handlungsbedarf in der Form, dass die Kommunikation auch bei einem längeren Stromausfall sichergestellt bleibt. Dies könnte durch Versorgung durch mehr Notstromaggregate geschehen oder durch den Einkauf von zusätzlichen Kommunikationsmitteln, wie sie im Amateur- und Satellitenfunk genutzt werden.

Da es für die Wiederherstellung der Stromversorgung von Vorteil ist, wenn Kommunikationsnetze zur Verfügung stehen, bietet sich die Nutzung der Kommunikationsnetze des Bundes an, da diese Netze in der Regel länger als 48 Stunden über Notstrom betrieben werden können.

# Literaturverzeichnis

- [1] Kießling, A.: Großstörung (im elektrizitätsversorgungsnetz). Webseite. URL [https://teamwork.dke.de/specials/7/Wiki-Seiten/Gro%C3%9Fst%C3%B6rung%20\(im%20Elektrizit%C3%A4tsversorgungsnetz\).aspx](https://teamwork.dke.de/specials/7/Wiki-Seiten/Gro%C3%9Fst%C3%B6rung%20(im%20Elektrizit%C3%A4tsversorgungsnetz).aspx). Letzter Zugriff 22.05.2016
- [2] Petermann, P., H., B., Lüllmann, A., Poetzsch, M., Riehm, U.: Was bei einem blackout geschieht - folgen eines langandauernden und großräumigen stromausfalls. Tech. Rep. LBNL-57661, CEC-500-2006-01, TAB (2011)
- [3] Reuter, C.: Communication between power blackout and mobile network overload. International Journal of Information Systems for Crisis Response and Management **6**(2), 38–53 (2014)
- [4] Römer, S.: Planung, aufbau und betrieb moderner telekommunikationsnetze. WissenHeute **62**(11), 18–26 (2009)
- [5] Römer, S., Peitzsch, M.: Ausfallsicherheit von kommunikationsnetzen. WissenHeute **62**(2), 26–36 (2009)
- [6] Unbekannt: Festnetz. Webseite. URL <https://de.wikipedia.org/wiki/Festnetz>. Letzter Zugriff 19.05.2016
- [7] Unbekannt: Internet. Webseite. URL <https://de.wikipedia.org/wiki/Internet>. Letzter Zugriff 15.05.2016
- [8] Unbekannt: Kommunikationssystem. Webseite. URL <https://de.wikipedia.org/wiki/Kommunikationssystem>. Letzter Zugriff 22.05.2016
- [9] Unbekannt: Mobilfunk. Webseite. URL <https://de.wikipedia.org/wiki/Mobilfunk>. Letzter Zugriff 23.05.2016
- [10] Unbekannt: Rundfunk. Webseite. URL <https://de.wikipedia.org/wiki/Rundfunk>. Letzter Zugriff 19.05.2016

# Kapitel 7

## Modellierung und Simulation von Smart-Grids

Maximilian Kreutz

**Zusammenfassung** Diese Arbeit gibt einen Überblick über die Grundlagen der Modellierung und Simulation von Smart-Grids. Es werden die grundlegenden Konzepte der Modelltheorie erläutert und in diesem Kontext die Besonderheiten und Eigenschaften von Smart-Grids aufgezeigt. Es werden verschiedene Simulationsformen wie Co-Simulation und Hardware-in-the-Loop vorgestellt. Außerdem wird auf das Potential von Multiagentensystem für die Steuerung und Automatisierung von Smart-Grids eingegangen. Abschließend wird das Co-Simulationsframework MOSAIK vorgestellt und dessen Vorzüge bei der Co-Simulation von Smart-Grids dargestellt.

### 7.1 Einführung

Die nicht zuletzt durch den Einsatz von erneuerbaren Energien geänderten Anforderungen an die elektrischen Verteilnetze, haben auch Einfluss auf die Werkzeuge zur Modellierung und Simulation dieser elektrischen Verteilnetze. Die Einspeisung des Stroms aus erneuerbaren Energien erfolgt meist aus kleinen dezentralen Energieerzeugungsanlagen wie z.B. Windkraftanlagen oder Photovoltaikanlagen, welche in ihrer Erzeugung vom Wetter abhängig sind und daher durch Prognosemodelle in ihrer Verfügbarkeit nur geschätzt werden können. Durch die schlechtere Planbarkeit und somit kurzfristigere Steuerung der erneuerbaren Energien, wird eine neue Form von elektrischen Verteilnetzen benötigt. Diese Netze sollen mit Hilfe der Kommunikationsnetze die Einspeisung der kleinen dezentralen Kraftwerke koordinieren und somit eine gleichmäßige Einspeisung in das Stromnetz gewährleisten. Diese Kopplung von Strom- und Kommunikationsnetz nennt man Smart-Grids [5]. Die Besonderheiten der Modellierung und Simulation dieser Smart-Grids wird in dieser

---

Carl von Ossietzky Universität Oldenburg  
E-mail: [maximilian.kreutz@uni-oldenburg.de](mailto:maximilian.kreutz@uni-oldenburg.de)

Arbeit betrachtet. Dabei soll ein besonderes Augenmerk auf die besonderen Eigenheiten von Smart-Grids gelegt werden und wie diese in Modellierung und Simulation zu berücksichtigen sind.

## 7.2 Darstellung der Problemstellung und Begriffsklärung

Bevor es zu einer Simulation kommen kann wird ein Modell benötigt, welches von einem System abgeleitet wird. Daher sollen diese Begriffe im Folgenden definiert und erläutert werden.

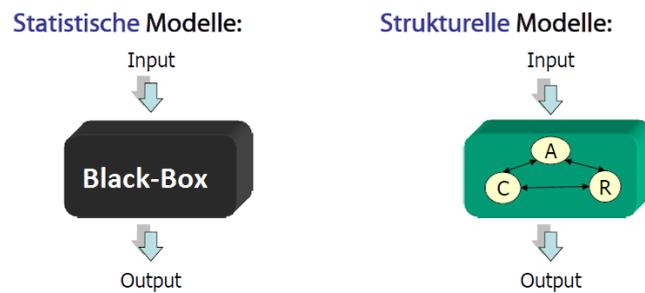
### 7.2.1 System

Dieser Abschnitt soll einen generellen Überblick geben, was ein System ist und abgrenzen, welche Art von System im weiteren Verlauf der Arbeit modelliert und simuliert werden soll. Im Allgemeinen lassen sich eine Vielzahl an unterschiedlichen System unterscheiden, wie z.B. Öko-Systeme, technische Systeme, Softwaresysteme oder eben auch wie für diese Arbeit relevant Energieversorgungssysteme und Kommunikationssysteme. Ein System besteht dabei aus Elementen und deren Beziehungen untereinander. Diese Beziehungen lassen sich durch Abhängigkeiten und Austauschprozesse definieren. Aus den Beziehungen ergibt sich die innere Struktur des Systems. Ein System besitzt außerdem einen über die Zeit veränderlichen Zustand. Zustandsveränderungen werden durch die interne Mechanismen, also die Abhängigkeiten und Austauschbeziehungen oder aber durch externe Einflüsse ausgelöst. Unter externen Einflüssen werden Umgebungsveränderungen von außerhalb des Systems verstanden. Ein System lässt sich von seiner Umwelt abgrenzen, wobei die Grenzen vom Betrachter definiert werden. Ein System kann auch aus mehreren Teilsystemen bestehen. Diese Abgrenzung bedeutet aber keine Isolation von der Umwelt, da das betrachtete System mit seiner Umwelt interagieren kann, bzw. von der Umwelt verändert werden kann und umgekehrt [12]. Für diese Arbeit relevant sind wie schon im Vorfeld genannt die Energieversorgungssysteme in Form der elektrischen Verteilnetze, sowie die darin enthaltenen Energieerzeuger, -verbraucher, und -speicher, sowie die Kommunikationssysteme in ihren unterschiedlichsten Ausprägungen, wie z.B. LTE, Glasfaser oder auch Power-Line-Communication (PLC), worunter die Nutzung des Stromnetzes zur Kommunikation verstanden wird [5]. Energieversorgungs- und Kommunikationssystem bilden zusammen das zu untersuchende Smart-Grid-System. Die Zusammensetzung dieser Systeme kann je nach Anforderung an die Leistung variieren und ist daher nur solange konstant, bis eine andere Zusammensetzung der bisherigen und ggf. zusätzlicher Elemente für ein besseres System gehalten werden. Im Bezug auf das Thema der Projektgruppe ist dies z.B. der Fall, wenn das Zuschalten weiterer Teilsysteme die Gesamtleistung des Systems verbessert. Alternativ wäre auch das Abschalten eines Teilsystems und das

dafür ausgleichende Zuschalten eines anderen ggf. leistungsfähigeren Teilsystems denkbar. In beiden genannten Fällen ändert sich die Zusammensetzung des Systems, da die geänderte Konfiguration den Systemzweck besser erfüllt.

### 7.2.2 Modell

Modelle erfassen Änderungen von System- und Umweltgrößen, unter der Voraussetzung der Messbarkeit bzw. Beobachtbarkeit dieser der untersuchten Systemeigenschaften bzw. Systemgrößen. Ein Modell lässt sich auf verschiedenen Ebenen betrachten. Es kann hier zwischen Reduktionismus und Holismus unterschieden werden. Während beim Holismus die Makro-Ebene des Modell, also das Modell als Ganzes betrachtet, wird beim Reduktionismus das Modell auf Mikro-Ebene seiner Elemente untersucht. Der innere Aufbau, sowie Austauschprozesse und Abhängigkeiten auf Mikro-Ebene beeinflusst das Verhalten des gesamten Modells auf Makro-Ebene. Es existiert allerdings auch der Begriff der Emergenz, welcher beschreibt, dass es Beobachtungen auf Makro-Ebene geben kann, welche auf Mikro-Ebene nicht erklärbar sind. In Bezug auf die Projektgruppe können die einzelnen Energieerzeuger, -speicher und -verbraucher, sowie die Kommunikationstechnik als Mikro-Ebene betrachtet werden. Das Zusammenwirken auf Mikro-Ebene bewirkt dann Änderungen im Gesamtsystem, im Projektgruppenkontext die Wiederherstellung der Stromversorgung und der Kommunikation. Ein Modell dient der Klärung einer bestimmten Fragestellung und hat somit einen Zweck. Je nach Zweck des Modells sind daher unterschiedliche Elemente und deren Zusammenhänge des Modells relevant [12]. Stachowiak charakterisiert den Begriff des Modells so, dass ein Modell nicht alle Eigenschaften des Originals erfasst, sondern das ein Modell abstrahiert [10]. Je nach Einsatzzweck lassen sich verschiedene Modellarten definieren. Es existieren beschreibende Modelle, welche Aussagen darüber geben sollen, wie ein System aufgebaut ist. Eine weitere Modellart bilden die erklärenden Modelle. Sie sollen die Fragen klären, warum das Modell das beobachtete Verhalten zeigt und welche Elemente dafür verantwortlich sind. Zusätzlich zu den genannten Modellarten gibt es auch noch die so genannten Prognosemodelle welche häufig auch als Vorhersagemodelle bezeichnet werden. Mit ihnen soll untersucht werden wie sich das gegebene System in einer geänderten Umgebung verhalten würde, oder welche internen bzw. externen Einflüsse das Systemverhalten verbessern oder verschlechtern. Mit Prognosemodellen kann auch untersucht werden, wie sich verschiedene Modellierungsansätze in einer gegebenen Umwelt verhalten. Als weitere Modellarten lassen sich statistische Modelle sowie strukturelle Modelle unterscheiden. Statistische Modelle untersuchen In- und Output eines Modells und vergleicht sie mit den Systembeobachtungen, wobei der innere Aufbau nicht weiter untersucht wird (Black-Box-Modell). Strukturelle Modelle hingegen untersuchen den Einfluss des inneren Aufbaus des Modells auf In- und Output (White-Box-Modell). Die folgende Darstellung fasst die beiden Konzepte schematisch zusammen.



**Abb. 7.1** Schematische Darstellung Statistisches und Strukturelles Modell [11]

Ein weiterer bei der Modellierung zu beachtender Punkt ist Wahl einer geeigneten zeitlichen Auflösung. Wird der Zustand des Modells zu bestimmten oder unbestimmten diskreten Zeitpunkten untersucht oder wird der Modellzustand immer bei Eintreten eines Ereignisses bestimmt. Sind Modellzweck, -art und die zeitliche Auflösung bestimmt, kann das betrachtete System in ein Modell überführt werden [12]. Hierzu gibt es zwei Vorgehensweisen, zum Einen kann das System anhand der Wirkbeziehungen in Teilsysteme zerlegt werden, welche dann im Einzelnen modelliert werden. Diese Vorgehensweise funktioniert allerdings nur dann, wenn die entsprechenden Wirkbeziehungen zwischen den Teilsystemen bekannt sind. Ist dies nicht der Fall, greift die zweite Vorgehensweise, die Zerlegung anhand der Systemgrößen. Für diese Vorgehensweise wird zunächst versucht die untersuchte Größe so weit wie möglich vom Einfluss der anderen Parameter zu isolieren und so kleine Teilmodelle zu bilden. Diese Teilmodelle werden anschließend zusammengeführt und bilden so die Modellierung des Gesamtsystems [11]. Da die Wirkbeziehungen in Smart-Grids sehr komplex sind, wird versucht anhand der ersten Vorgehensweise Teilsysteme zu isolieren und diese dann entsprechend zu modellieren. Durch sukzessives Zusammenschalten von neu gestarteten Teilsystemen soll im Falle eines Blackouts so das Gesamtsystem wiederhergestellt werden. Dabei werden sogenannte Zellen gebildet, welche jeweils einen eigenständigen Regelkreis bilden und eine vollständigen Satz der nötigen technischen Akteure enthalten. Die Elemente des Energieversorgungsnetzes sind bei Energiequellen, Energiespeicher und Energieverbraucher. Diese werden ergänzt durch die Netzbetriebsmittel, welche aus Energienknoten zur physikalischen Leistungsverteilung und den Energietransportleitungen bestehen. Elemente des Kommunikationssystems sind Kommunikationsknoten und -leitungen für den Informationstransport, Kommunikationsbrücken zwischen Wirkungsdomänen, sowie die Dienstplattformen als physische Ausführungsumgebungen. Als dritte Gruppe gehören noch die informationsverarbeitenden Elemente des Automatisierungssystems, welche aus Mess-, Stell-, Steuer-, und Regeleinrichtungen bestehen, in den vollständigen Satz der technischen Akteure. Werden diese Akteure nun zusammengebracht um die zu simulierende Zelle abzubilden müssen die Eigenschaften des zu simulierenden Systems beachtet werden [2]. Solche Eigenschaften können z.B die verfügbare Bandbreite und Latenz des Kommunikationsnetzes sein oder auch das Umstand das einige Erzeuger und Verbraucher gekoppelt sind, etwa wenn eine PV-Anlage auf einem Hausdach installiert ist. Ebenso müssen entsprechende Puffer für

den Ausgleich größerer Lastverschiebungen bedacht werden, z.B. wenn ein Elektro-Auto von einer Zelle in eine andere wandert. Der Jahresverbrauch eines Elektroautos entspricht in etwa dem eines kleinen Haushalts, weshalb solche Lastverschiebungen in der Planung berücksichtigt werden sollten [5]. Im Falle eines Blackouts können Pufferspeicher je nach Leistungsfähigkeit als Überbrückungsreserve genutzt werden bis die eigentlichen Energieerzeuger wieder hochgefahren sind oder als Unterstützung genutzt werden, wenn die Leistung der verfügbaren Energieversorgungseinheiten allein noch nicht ausreicht um bspw. Kommunikationsanlagen wieder in Betrieb zu nehmen. Im nächsten Schritt könnte die nun verfügbare Kommunikationsalge genutzt werden um weitere Energieversorgungseinheiten im Smart-Grid anzusteuern und hochzufahren. Eine schematische Darstellung der Strom und Datenflüsse in einer Smart-Grid Zelle ist in der folgenden Abbildung dargestellt.

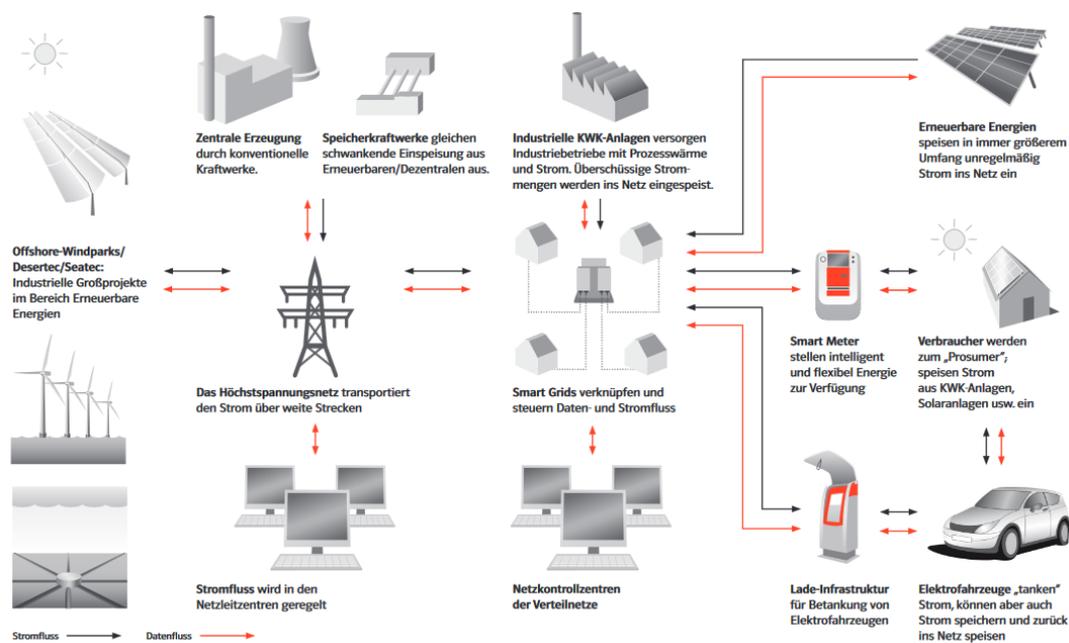


Abb. 7.2 Strom- und Datenfluss in einem Smart-Grid [3]

### 7.2.3 Simulation

Die Simulation ist ein entscheidender Schritt in der Konzeption und Gestaltung von Smart-Grids, da es sich sowohl beim Kommunikationsnetz als auch beim Energieversorgungsnetz um kritische Infrastrukturen handelt. Laut dem Bundesamt für Bevölkerungsschutz und Katastrophenhilfe wird kritische Infrastruktur wie folgt definiert: "Kritische Infrastrukturen (KRITIS) sind Organisationen oder Einrichtungen mit wichtiger Bedeutung für das staatliche Gemeinwesen, bei deren Ausfall oder Beeinträchtigung nachhaltig wirkende Versorgungsengpässe, erhebliche Störungen

der öffentlichen Sicherheit oder andere dramatische Folgen eintreten würden." [1]  
Die folgende Darstellung zeigt die Sektoren kritischer Infrastrukturen.

Sektoren Kritischer Infrastrukturen	
Energie	Transport und Verkehr
Informationstechnik und Telekommunikation	Finanz- und Versicherungswesen
Gesundheit	Staat und Verwaltung
Wasser	Medien und Kultur
Ernährung	

**Abb. 7.3** Sektoren kritischer Infrastrukturen [1]

Es lässt sich ableiten, dass auch andere kritische Infrastrukturen, wie z.B. die Wasserversorgung oder Informationsmedien, abhängig von den Kommunikations- bzw. Energienetzen sind, weshalb diese besonders zuverlässig funktionieren müssen. Daher können Versuche nicht mit den Netzen selbst durchgeführt werden, sondern müssen möglichst praxisnah und detailliert simuliert und getestet werden. Hierzu existieren verschiedene Ansätze, wie die Co-Simulation, Hardware-in-the-Loop-Simulationen oder auch die Nutzung von Multiagentensystemen für die Steuerung von Smart-Grids, welche besondere Anforderungen an die Simulation des Kommunikationsnetzes stellen. Auf diese Ansätze wird im späteren Verlauf dieser Arbeit genauer eingegangen.

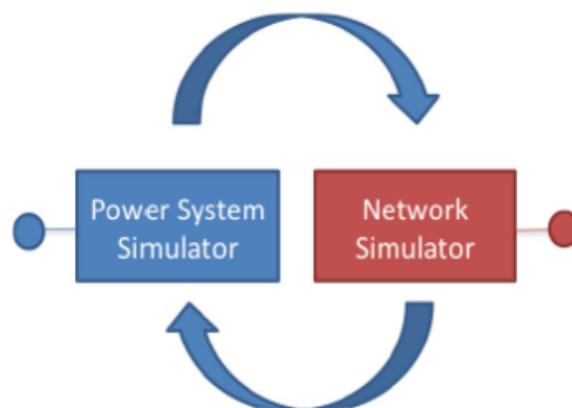
## 7.3 Simulationssoftware für Smart-Grids

Der mehrschichtige Aufbau und die Vielzahl und Variationen der in einem Smart-Grid vorhandenen Elemente stellt eine größten Herausforderungen für die Simulationssoftware dar. Anders als bei der bisherigen zentralen Energieerzeugung durch wenige große Kraftwerke, müssen nun viele kleine dezentrale Einheiten simuliert und reguliert werden. Dabei zeichnen sich Smart-Grids besonders durch ein hohes Kommunikationsaufkommen zwischen den dezentralen Energieerzeugungseinheiten aus, welche ihre eingespeiste Leistung koordinieren müssen um einen stabilen Netzbetrieb zu ermöglichen. Dabei kann die Kommunikation durch mangelnde Bandbreiten, hohe Latenzen oder Verbindungsabbrüche behindert werden [5]. Die Simulationssoftware muss also in der Lage sein solche Effekte zu berücksichtigen, um so Rückschlüsse auf das Systemverhalten unter realen Bedingungen zuzulassen. Ebenfalls sollten die relevanten Kommunikationsmedien unterstützt werden, wie z.B Internet, Mobilfunk oder PLC. Auch das Energienetz stellt eine Vielzahl an Anforderungen an die Simulationssoftware, da diese verschiedenste Formen von

Energieerzeugern und -speichern unterstützen muss. Eventuell müssen zusätzlich Anforderungen des Gas-Netzes berücksichtigt werden, etwa wenn sich in der simulierten Zelle ein gasbetriebenes Blockheizkraftwerk (BHKW) befindet [9]. Es müssen auch stochastische Modelle unterstützt werden, da diese bei der Simulation von erneuerbaren Energien eine große Rolle spielen, etwas bei der Simulation von Windparks oder von Photovoltaik-Anlagen. Hier steigt der Berechnungsaufwand mit der Zahl der zu simulierenden Energieerzeuger. Da bisherige Software häufig für die Simulation einer der beiden relevanten Netzarten in Smart-Grids entwickelt und optimiert wurde, also entweder für die Simulation von Energieversorgungsnetzen oder für die Simulation von Kommunikationsnetzen, wurde der Ansatz der Co-Simulation entwickelt [5]. Diese Ansatz soll im folgenden Abschnitt weiter erläutert werden.

### 7.3.1 Co-Simulation

Der Vorteil der Co-Simulation liegt darin, dass jeweils für den Einsatzzweck optimierte und bewährte Simulationssoftware zum Einsatz kommt. Das Modell wird also in den für das jeweilige Netz optimierten Umgebung simuliert. Das Energienetz dann in einer bereits bestehenden Simulationsumgebung für Energienetze simuliert und das Kommunikationsnetz entsprechend in einer Kommunikationssimulationssoftware. Mit dieser Methode lässt sich auch der Entwicklungsaufwand eines völlig neuen Simulationssoftware für den angestrebten Zweck umgehen, indem man bereit erprobte Lösungen kombiniert. Es ist außerdem davon auszugehen, dass sich durch den Einsatz bereits erprobter und bewährter Simulationssoftware, Vorteile in den Bereichen Verlässlichkeit und Skalierbarkeit generieren lassen. Die beiden Simulationsprogramme simulieren jeweils das entsprechende Netz und geben den Zustand als Input für das andere Programm weiter, damit dieses auf Basis dieser Daten den nächsten Simulationsschritt berechnen kann [5]. Dieser Ablauf ist in der folgenden Abbildung schematisch dargestellt.



**Abb. 7.4** Schematische Darstellung Co-Simulation [5]

Eines der Kernprobleme bei der Co-Simulation von Smart-Grids ist die unterschiedliche zeitliche Auflösung der Simulationsprogrammen. Während Energienetz-Simulationssoftware in konstanten Schritten rechnet und so versucht das Energienetz im Gleichgewicht zu halten, agieren viele Simulationsprogramme für Kommunikationsnetze ereignisbasiert. Das bedeutet, dass das Programm den nächsten Schritt erst bei auftreten eines Ereignisses bestimmt, etwas beim Verschicken einer Nachricht [5]. Die Unterschiede zwischen diesen zeitlichen Auflösungen wird in der folgenden Abbildung illustriert.

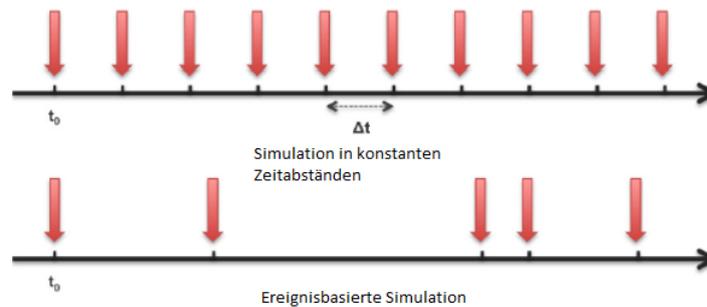


Abb. 7.5 Darstellung der Simulationsschritte [5]

Diese unterschiedlichen Auflösungen führen bei der Co-Simulation zu Synchronisierungsproblemen, da gesendete Nachrichten ggf. erst zu spät vom anderen Programm verarbeitet werden. Ein möglicher Lösungsansatz besteht darin für beide Simulationsprogramme feste Synchronisierungspunkte zu bestimmen, welche dafür sorgen, dass es in regelmäßigen Abständen zu einem Informationsaustausch zwischen den beiden Programmen kommt. Aber auch dieser Ansatz ist noch nicht optimal, da es zwischen diesen Synchronisationspunkten zu Verzögerungen kommen kann, wie in der folgenden Abbildung zu erkennen ist [5].

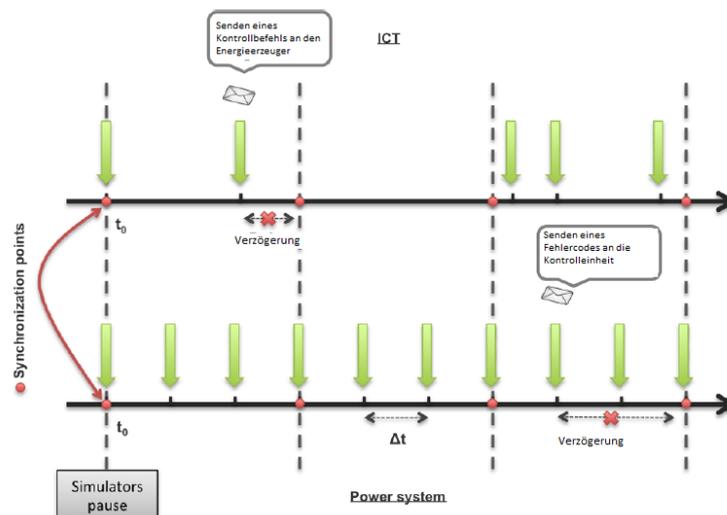


Abb. 7.6 Lösungsansatz des Synchronisationsproblems durch feste Synchronisationspunkte [5]

Insbesondere das Energienetz hat besondere Anforderungen an die zeitliche Auflösung der Simulationssoftware. Hier variiert die Zeitspanne je nach Anwendungsfall zwischen einigen Mikrosekunden, z.B. beim Ausgleich von Blitzeinschlägen und ganzen Tagen z.B. bei der Optimierung der Nutzung erneuerbarer Energien [5]. Die folgende Abbildung gibt einen guten Überblick über Anwendungsfälle und die nötige zeitliche Auflösung.

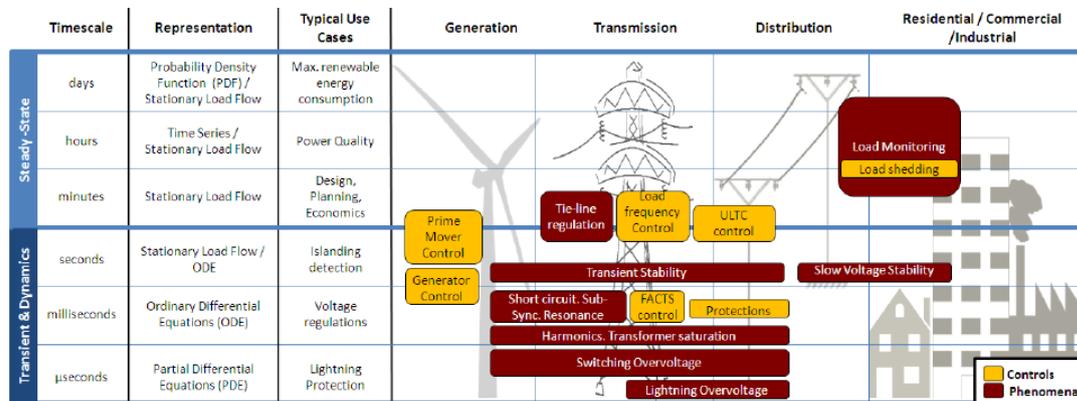


Abb. 7.7 Zeitliche Auflösung nach Anwendungsfall [5]

Der Austausch der Daten zwischen den Simulationsprogrammen kann durch ein Framework koordiniert werden, welches über entsprechende Interfaces mit den Programmen kommuniziert und den In- und Output koordiniert und das zu simulierende Szenario bereitstellt [5]. Ein solches Framework ist beispielsweise Mosaik, welches in einem späteren Abschnitt genauer betrachtet werden soll. Um die Simulationsergebnisse realitätsnäher zu gestalten gibt es die Möglichkeit Hardwarekomponenten in den Simulationsprozess mit einzubinden. Dieses Vorgehen wird auch als Hardware-in-the-Loop-Simulation bezeichnet und soll im folgenden Abschnitt näher betrachtet werden.

### 7.3.2 Hardware-in-the-Loop

Hardware-in-the-Loop-Simulationen ermöglichen die Simulation komplexer eingebetteter Hardwaresysteme. Einige Komponenten sind als Hardware vorhanden, während andere durch Software und entsprechende mathematische Modelle simuliert werden. Hierzu wird der Hardware an Sensoren und Messstellen durch die Software entsprechende Eingangsdaten vorgegeben, wie sie sonst von der simulierten Hardware kommen würde. So lässt sich zum Einen das Verhalten einiger Systemkomponenten und zum Anderen das Verhalten zu teurer, zu größer oder zu gefährlicher Systemkomponenten unter sehr realitätsnahen Bedingungen untersuchen. In der Realität sind trotzdem Abweichungen möglich, da Effekte wie Latenzen oder andere technische Beeinträchtigungen nie komplett durch Modelle erfasst werden können [5]. Um

die Einflüsse durch Beschränkungen des Kommunikationsnetzes auf die Steuerung der Energieerzeuger zu minimieren, empfiehlt es sich den Energieerzeugungseinheiten eine möglichst hohe Autonomie zu ermöglichen. Dies kann z.B. durch den Einsatz von Multiagentensystemen geschehen, wie im folgenden Abschnitt erläutert wird.

### 7.3.3 Multiagentensysteme

Der Einsatz von Multiagentensystemen in Smart-Grids ermöglicht die dezentrale und teilautonome Steuerung von Energieerzeugerzellen. Dazu wird jeder Erzeugungseinheit ein Agent zugewiesen. Diese Agenten werden durch einen übergeordneten Agenten koordiniert, sodass die Erzeugungseinheiten zusammen einen festgelegten Output erzeugen. Dazu bekommen die Agenten ein Set an möglichen Fahrplänen mit aus denen Sie dann anhand der Fahrpläne der umliegenden Zellen und der gegebenen Anforderungen einen wählen. Über die Implementierung von maschinellen Lernalgorithmen lassen sich auch Fahrpläne optimieren oder neue Fahrpläne erstellen. Durch den Aufbau mit einem koordinierenden Agenten lässt sich im Fehlerfall die Störquelle schneller eingrenzen. Hierzu fragt der Koordinationsagent den Status der ihm unterstellten Energieerzeugungseinheiten ab und kann so ggf. andere Erzeugungseinheiten anweisen, einen anderen Fahrplan zu wählen um mehr Energie einzuspeisen bis die Störung an anderer Stelle behoben ist. So kann eine schnelle und automatisierte Rückkehr zum Normalbetrieb realisiert werden [2]. Wie ein solches Multiagentensystem aufgebaut sein kann, zeigt die folgende schematische Darstellung.

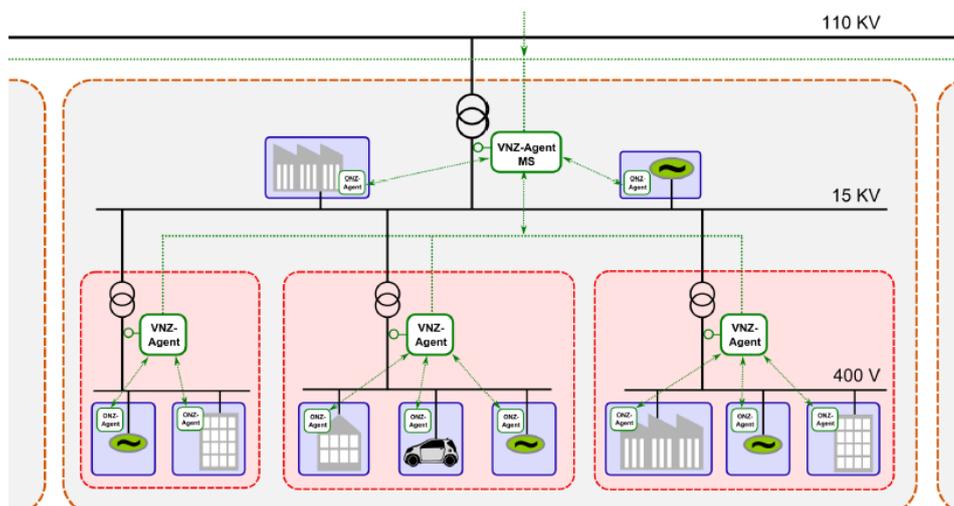


Abb. 7.8 Schematischer Aufbau eines Zellbasierten Multiagentensystems [2]

Das dieses Konzept funktioniert, wurde mit mehreren funktionierenden Demonstratoren gezeigt. So existiert an der National Technical University of Athens ein funktionierendes multiagentengesteuertes Teststromnetz. Auch die Analysefunktion wurde bereit erfolgreich an der University of Strathclyde umgesetzt. Unter dem Namen PEDDA (Protection Engineering Diagnostic Agents) wurde ein Multiagentensystem zur Automatisierung der Analyse von elektrischen Systemen erstellt [4]. Daher scheint ein Multiagentensystem als Ansatz einer automatischen Wiederherstellung der Stromversorgung nach einem Totalausfall (Blackout) als sehr vielversprechend, da sich so Zelle für Zelle automatisch in einen normalen Betriebszustand versetzen kann und sich dann mit den umliegenden Zellen koordinieren kann [2]. So kann durch das schrittweise Zusammenschalten der Zellen wieder das gesamte Energieversorgungsnetz und Kommunikationsnetz wieder hergestellt werden.

### **7.3.4 *mosaik***

mosaik ist ein Co-Simulationsframework für Smart-Grid-Simulationen, welches es ermöglicht, bestehende Simulationsmodelle zu verwenden und zu kombinieren um so große Smart-Grid-Szenarien mit eine Vielzahl an simulierten Einheiten zu simulieren. Diese Szenarien lassen sich dann also Grundlage für den Test von Multiagentensystemen oder zentraler Steuerstrategien nutzen [7]. mosaik besteht aus vier Hauptkomponenten. Die erste Komponente ist die SimAPI, welche das Kommunikationsprotokoll zwischen mosaik und den Simulatoren definiert. Die zweite Komponente ist die ScenarioAPI, welche die Szenariodefinition in Python erlaubt. Die dritte Komponente ist der SimulationManager welcher die Simulationsprozesse steuert und mit diesen kommuniziert. Der Simulationmanager kann drei Aktionen ausführen, einen neuen Simulationsprozess starten, die Verbindung zu bereits laufenden Prozessen herstellen und ein Simulationsmodul importieren und im laufen Prozess ausführen, wenn dieses in Python 3 geschrieben ist. Die vierte und letzte Komponente ist der Simulator welcher die ereignisdiskrete Simulationsbibliothek SimPy zur koordinierten Simulation des Szenarios. mosaik ist in der Lage Simulatoren mit verschiedenen zeitlichen Auflösungen zu kombinieren. mosaik dabei kann Abhängigkeiten zwischen verschiedenen Simulationen erkennen und einen Simulator zu einen Simulationsschritt zwingen, wenn dessen Daten in einem anderen Simulator benötigt werden. Dabei ist das modulare Design von mosaik in zwei Ebenen aufgeteilt. Dem mosaik Kernelement, welches den Datenaustausch zwischen den verbundenen Simulatoren und deren Ausführung koordiniert. Diese Simulatoren bilden die zweite Ebene, das mosaik-Ökosystem. In dieser Ebene finden sich die verschiedenen Simulatoren für die Co-Simulation, welche je nach Bedarf mit dem koordinierenden Kernelement verbunden werden können [6]. Dies ist in der folgenden Abbildung nochmal verdeutlicht.



**Abb. 7.9** Modulares Design mosaik [8]

mosaik bietet also eine Vielzahl an Anpassungsmöglichkeiten für die verschiedensten Anwendungsbereiche und besitzt außerdem die Möglichkeit die Synchronisierungsproblematik in der Co-Simulation durch das Erzwingen von Simulationsschritten zu minimieren. Im Hinblick auf die recht offene Aufgabenstellung der Projektgruppe, nach einem Blackout das Kommunikation und Stromversorgung automatisch wieder herzustellen, bietet mosaik die nötige Flexibilität und Erweiterbarkeit verschieden Lösungsansätze für diese Problemstellung zu erarbeiten.

## 7.4 Fazit

Modellierung und Simulation bieten gerade im Hinblick auf die Analyse von Smart-Grids die Möglichkeit diverse Szenarien zu testen und so einen stabilen und zuverlässigen Betrieb kritischer Infrastrukturen zu gewährleisten. Durch den Einsatz von Multiagentensystemen wird ein hoher Grad an Autonomie der einzelnen Energieerzeuger ermöglicht. Damit dies in der Praxis fehlerfrei funktioniert, ist es wichtig dieses Systeme möglichst praxisnah und ohne Gefahr für den Betrieb zu testen. Simulationen und insbesondere Hardware-in-the-Loop-Simulationen bieten hier eine Vielzahl an Möglichkeiten verschiedene Betriebszustände und Steuerungsmechanismen zu testen und zu evaluieren. Co-Simulationsframeworks wie mosaik ermöglichen eine optimale Simulation verschiedenster Szenarien und lassen sich durch einen modularen Aufbau beliebig auf neue Szenarien und Anwendungsfälle erweitern. Das Paper von Mets et al. bietet einen Überblick über verschiedene Simulationsprogramme, sowie deren Stärken und Schwächen und bietet daher eine gut Hilfestellung bei der Auswahl der Simulationsumgebungen für die Co-Simulation mit z.B. mosaik.

# Literaturverzeichnis

- [1] BBK: Kritische Infrastrukturen (2016). URL [http://www.bbk.bund.de/DE/AufgabenundAusstattung/KritischeInfrastrukturen/kritischeinfrastrukturen\\_node.html](http://www.bbk.bund.de/DE/AufgabenundAusstattung/KritischeInfrastrukturen/kritischeinfrastrukturen_node.html). Bundesamt fuer Bevoelkerungsschutz und Katastrophenhilfe
- [2] Benze, J., Huebner, C., Kiessling, A.: Informatik schafft communities (2011)
- [3] EON: Eon-netze (2016). URL [http://www.eon.com/content/dam/eon-com/de/downloads/e/E.ON\\_NETZE10.pdf](http://www.eon.com/content/dam/eon-com/de/downloads/e/E.ON_NETZE10.pdf)
- [4] McArthur, S.D.J., Davidson, E.M., Catterson, V.M., Dimeas, A.L., Hatziargyriou, N.D., Ponci, F., Funabashi, T.: Multi-agent systems for power engineering applications - part i: Concepts, approaches, and technical challenges. IEEE TRANSACTIONS ON POWER SYSTEMS **22**(4) (2007)
- [5] Mets, K., Ojea, J.A., Develder, C.: Combining power and communication network simulation for cost-effective smart grid analysis. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, NO. 3, THIRD QUARTER 2014 **16**(3) (2014)
- [6] OFFIS: Documentation mosaik (2016). URL <http://mosaik.readthedocs.io/en/latest/index.html>
- [7] OFFIS: Homepage mosaik (2016). URL <https://mosaik.offis.de/>
- [8] OFFIS: Modulares design mosaik (2016). URL <http://mosaik.readthedocs.io/en/latest/ecosystem/index.html>
- [9] Ponci, F., Monti, A., Benigni, A.: Simulation for the design of smart grid controls. 2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS) - at IEEE SmartGridComm 2011 (2011)
- [10] Stachowiak, H.: Allgemeine Modelltheorie. Springer Verlag (1973)
- [11] Vogel, U.: Einfache dgl und simulationstool vensim. Vorlesungsfolien 3 (2013). WS 2013/2014
- [12] Vogel, U.S.M.: Modellierung und simulation oekologischer systeme. Vorlesungsfolien 1 (2013). WS 2013/2014

# Kapitel 8

## Verteilte Berechnung von minimalen Spannbäumen

Tim Dombrowski

**Zusammenfassung** Das Ziel dieser Ausarbeitung ist es, die verteilte Berechnung von minimalen Spannbäumen am Beispiel des GHS (Gallager, Humblet und Spira) Algorithmus zu untersuchen und mit zentralen Verfahren zu vergleichen. Außerdem soll ein möglicher Bezug zu der Problematik der Projektgruppe, dem Wiederaufbau eines Stromnetzes, diskutiert werden. Es werden daher verschiedene Vorgehensweisen beschrieben und dazu Algorithmen vorgestellt, die diese abbilden. Zu Beginn wird nach einer Definitionsphase eine Methode am Beispiel des greedy Algorithmus von Kruskal beschrieben. Da dieser sequentiell mit zentraler Steuerung funktioniert, wird zusätzlich der GHS Algorithmus als Implementierung einer verteilten Methode zum Bestimmen von minimalen Spannbäumen vorgestellt. Bei der abschließenden Diskussion wird deutlich, dass eine vollständige Übernahme des GHS Algorithmus für diese Projektgruppe nur bedingt möglich ist.

### 8.1 Einleitung

Minimale Spannbäume können nicht nur zum Lösen von Rundreiseproblemen verwendet werden. Sowohl in der Elektro- wie auch bei der Netzwerktechnik können minimale Spannbäume optimale Leitungswege abbilden.

Im Rahmen der Projektgruppe Blackout-Restarts erfolgt eine Auseinandersetzung des Themas: Verteilte Berechnung von minimalen Spannbäumen. Dabei wird besonders der Bezug zur Aufgabe der Projektgruppe, dem verteilten Wiederaufbau eines Stromnetzes, aufgenommen.

Zu Beginn dieser Hausarbeit werden minimale Spannbäume definiert. Darauf folgt die Bestimmung eben dieser mittels klassischer Verfahren wie dem Algorithmus von Kruskal. Anschließend wird der von Gallager, Humblet und Spira entwickelte GHS Algorithmus beschrieben, welcher minimale Spannbäume verteilt bestimmen

---

Carl von Ossietzky Universität Oldenburg  
E-mail: [tim.dombrowski@uni-oldenburg.de](mailto:tim.dombrowski@uni-oldenburg.de)

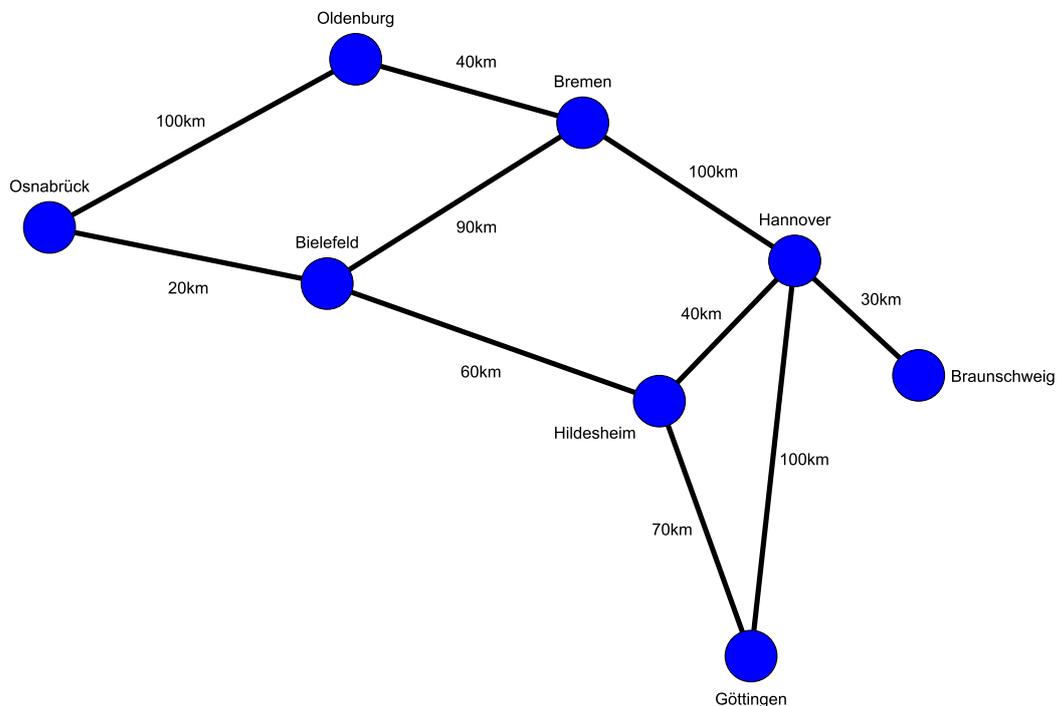
kann. Abschließend wird die Möglichkeit der Übertragung dieses Verfahrens auf den sukzessiven Wiederaufbau eines Stromnetzes diskutiert.

## 8.2 Minimale Spannbäume

Im Folgenden werden zunächst Graphen definiert, um danach Spannbäume, vor allem minimale Spannbäume, beschreiben zu können.

### 8.2.1 Graphen

Graphen sind, so Tittmann, mathematische Modelle für netzartige Strukturen. Sie bestehen aus einer Menge von Knoten  $N$  (für Nodes) und einer Menge von Kanten  $E$  (für Edges).[8] Graphen sind entweder gerichtet oder ungerichtet. Bei gerichteten Graphen sind die Kanten in eine Richtung definiert. Beispielsweise wäre in so einem Graphen die Kommunikation zweier Knoten nur in eine Richtung möglich. Bei Ungerichteten erfolgt eine bidirektionale Richtung der Kanten. Außerdem können die Kanten eines Graphen entweder gewichtet oder ungewichtet sein.[8]. In dieser Ausarbeitung wird sich ausschließlich auf ungerichtete Graphen mit gewichteten Kanten bezogen. Abbildung (8.1) zeigt ein Beispiel für solch einen Graphen.



**Abb. 8.1** Beispiel für einen gewichteten Graphen

## 8.2.2 Minimale Spannbäume

Graphen können verschiedene Teilgraphen enthalten. Sie bestehen genauso wie ein Graph aus einer Menge von Knoten und Kanten. Spannbäume sind solche Teilgraphen, welche jeweils alle Knoten eines Graphen miteinander verbinden ohne dass die Kanten dabei Zyklen bilden. [6] Jedem Spannbaum ist ein Gewicht zugeordnet, welches die Summe der Gewichte aller im Spannbaum enthaltenen Kanten ist.[6] Ein minimaler Spannbaum (kurz *mst* aus dem englischen für minimum spanning tree) ist jener Spannbaum dessen Gewicht unter allen Spannbäumen eines Graphen minimal ist. Hat jede Kante eines Graphen ein unterschiedliches Gewicht, so wäre der minimale Spannbaum zusätzlich einzigartig.[6]

## 8.3 Berechnung minimaler Spannbäumen

Minimale Spannbäume können mittels verschiedener Verfahren bestimmt werden. Da der Schwerpunkt dieser Arbeit auf der verteilten Bestimmung liegt, beschränkt sich der erste Abschnitt auf das Verfahren nach Kruskal. Darauf folgt der von Gallager, Humblet und Spira entwickelte Ansatz des GHS Algorithmus.

### 8.3.1 Grundsätzliche Ansätze

Grundsätzlich werden die minimalen Spannbäume bei diesen Verfahren sequentiell mit zentraler Steuerung bestimmt. Das bedeutet, dass die Algorithmen zu jeder Zeit Kenntnisse über den gesamten Graphen und damit über die Knoten, die Kanten und die Gewichte haben. Dementsprechend können und werden die minimalen Spannbäume iterativ von einer einzelnen Instanz aufgebaut. Im Folgenden folgt die Umsetzung eines solchen Ansatzes im Algorithmus von Kruskal.

#### 8.3.1.1 Algorithmus von Kruskal

Kruskal beschreibt in seinem Paper eine Methode, um auf möglichst einfachem Weg den minimalen Spannbaum eines Graphen zu bestimmen. Der Algorithmus arbeitet mit einer Menge von Kanten  $E$ , welche zunächst aufsteigend ihres Gewichts sortiert werden. Daraufhin wird iterativ der minimale Spannbaum aufgebaut indem solange  $E$  noch Kanten enthält, bei jedem Iterationsschritt eine Kante  $e \in E$  ausgewählt wird. Diese wird aus  $E$  entfernt und es wird überprüft, ob  $e$  mit dem aktuellen minimalen Spannbaum  $mst$  keinen Zyklus bildet, dann wird  $e$  dem  $mst$  hinzugefügt. Ansonsten wird  $e$  verworfen.[6]

Nachfolgend ist eine Pseudocode Implementierung dieses Algorithmus angegeben.

```

1 public Mst kruskal() {
2     E = Menge aller Kanten eines Graphen G;
3     Mst mst = new Mst();
4     sort(E);
5     while (E.notEmpty()) {
6         e = E.next();
7         E.remove(e);
8         if (mst.getNoCircleWith(e)) {
9             mst.add(e);
10        }
11    }
12    return mst;
13 }

```

Der Algorithmus benötigt in jedem Fall  $|E|$  Iterationen. Zusätzlich muss  $E$  sortiert werden. Dies führt nach der Arbeit von Katajainen und Nevalainen zu der in (8.1) angegebenen Zeitkomplexität.[3]

$$O(|E| * \text{Log}(|E|)). \quad (8.1)$$

### 8.3.2 Verteilte Berechnung minimaler Spannbäume

Da die oben dargestellte Methode nur zentral mit Kenntnis über sämtliche Kantengewichte funktioniert, folgt in diesem Abschnitt eine Methode zum verteilten Bestimmen minimaler Spannbäume. Zunächst folgt eine Erörterung der grundsätzlichen Vorgehensweise. Daraufhin folgt die Beschreibung des von Gallager, Humblet und Spira entwickelten Algorithmus.

#### 8.3.2.1 Grundsätzliche Idee

Die verteilte Berechnung minimaler Spannbäume erfolgt mittels eines Multiagentensystem. Jeder Knoten ist dabei ein Agent und Kanten sind die Kommunikationswege zwischen diesen. Die Idee ist, dass die Knoten sich über die kürzesten Kanten zu Fragmenten zusammenschließen und dadurch, iterativ größere Fragmente entstehen und sie gleichzeitig weniger werden. Zum Schluss bleibt nur ein Fragment welches den minimalen Spannbaum darstellt übrig.[2]

Im Folgenden wird der GHS Algorithmus, welcher nach diesem Vorgehen funktioniert, detailliert beschrieben.

#### 8.3.2.2 GHS Algorithmus

Wie oben beschrieben, erfolgt die Bestimmung des minimalen Spannbauums mittels eines Multiagentensystems. Ein Multiagentensystem besteht aus einer Menge von

mehreren Agenten, welche verteilt eine Problemlösung generieren.[5] Bei dem hier beschriebenen Vorgang, werden Knoten als Agenten definiert.

### *Agentensystem*

Ein Softwareagent, nach Kim, ist mit speziellen Eigenschaften ausgestattet. Diese seien eine gewisse Autonomie, ein dem Handeln zugrunde liegenden Rationalitätskalkül und das Erreichen von als erstrebenswert angesehenen Zielen. Außerdem seien Agenten entweder proaktiv oder reaktiv, haben Zustände und sind kommunikativ.[5] Damit das Multiagentensystem eine Aufgabe erfüllt, müssen die Agenten untereinander kommunizieren. Die Zustände der Agenten, in welchen sie sich bei diesem Verfahren befinden können, werden in Tabelle 8.2 dargestellt.

### *Kommunikationswege*

Diese Kommunikation erfolgt mittels Nachrichten über die Kanten zwischen den Knoten und in beide Richtungen. Um die Kommunikation, innerhalb und außerhalb eines Fragmentes zu erleichtern, können Kanten drei verschiedene Kennzeichnungen haben. Zur besseren Übersicht sind in Tabelle 8.3 sämtliche Nachrichtentypen, welche für das Verfahren benötigt werden, dargestellt. Ist eine Kante als *branch* markiert, liegt diese innerhalb des Fragments. Eine *basic* Kante ist zwischen zwei verschiedenen Fragmenten aufgespannt. Eine *rejected* Markierung einer Kante bedeutet, dass die beiden Knoten der Kante im Fragment sind, aber die *rejected* Kante selbst nicht. [2]

Die verschiedenen Kantentypen werden zusätzlich in Tabelle 8.1 beschrieben.

**Tabelle 8.1** Kantentypen

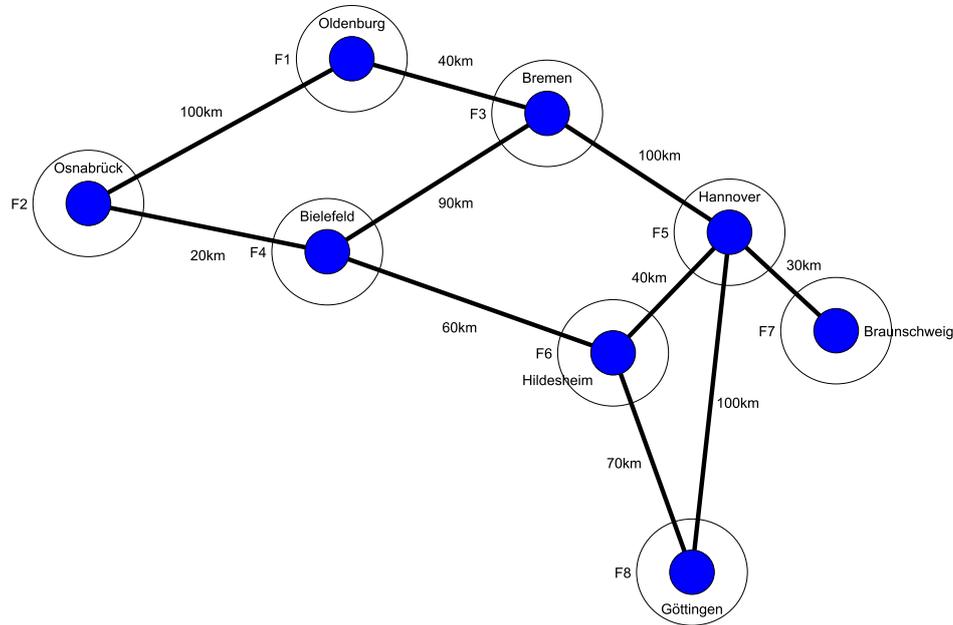
Kantentyp	Bedeutung
<i>branch</i>	Kante ist innerhalb des Fragments
<i>basic</i>	Kante ist die Verbindung zweier Fragmente
<i>rejected</i>	Kante ist außerhalb, aber die angrenzenden Knoten innerhalb des Fragments

### *Fragmente*

Jeder Knoten, des Systems, führt den selben lokalen Algorithmus aus.[2] Wie oben beschrieben ist die Idee dieses Verfahrens, Fragmente zu erzeugen, welche iterativ wachsen. Das geschieht entweder durch den Zusammenschluss von zwei Fragmenten oder das Absorbieren eines Fragments durch ein anderes. Dies wird unter dem Punkt *Change-Core* beschrieben. Ein Fragment, hat in der Regel ein Level und eine ID. Das Level ist abhängig von der Anzahl der Zusammenschlüsse, welche ein Fragment vollzogen hat. Die ID lässt sich aus dem Kern des Fragments generieren. Dieser ist die Kante des letzten Zusammenschluss.[2]

Zu Beginn ist jeder Knoten ein eigenes Fragment mit dem Level 0 und dem Status *sleep*. Daraus ergibt sich ein Sonderfall für den Algorithmus.[2] Abbildung 8.2 stellt diesen Sonderfall beispielhaft dar.

In Abbildung 8.2 ist jeder Knoten ein eigenes Level 0 Fragment. Es gab noch keinen



**Abb. 8.2** Alle Knoten sind eigene Fragmente

Zusammenschluss daher ist keine Kante innerhalb eines Fragments und jede ist eine *basic* Kante. Außerdem befindet sich jeder Knoten im Zustand *sleep*.

**Tabelle 8.2** Status der Agenten

Zustand	Bedeutung
<i>sleep</i>	Der Agent ist nicht aktiv und wartet auf ein spontanes Erwachen
<i>find</i>	Der Agent sucht seine leichteste <i>basic</i> Kante
<i>found</i>	Der Agent wartet auf eine <i>initiate</i> Nachricht um wieder in den Zustand <i>find</i> zu wechseln

### *Sonderfall*

Der Sonderfall ergibt sich daraus, dass jeder Knoten ein eigenes Fragment, mit Level 0, darstellt und sich im Zustand *sleep* befindet. Dies ist nur bis zum ersten Zusammenschluss irgendwelcher Fragmente der Fall. Die Knoten erwachen spontan, ändern ihren Status auf *find* und suchen mittels ihres eigenen Algorithmus nach ihrer leichtesten Kante. Sobald diese gefunden wurde, wird über diese eine *connect*

Nachricht gesendet. Davor wird die Kante noch als *branch* markiert und der Status des Knoten ändert sich zu *found*. Da dies alle Fragmente durchführen, warten die Knoten nach dem Versand auf eine *connect* Nachricht von der Gegenseite des *branch*. Nachrichten bestehen hierbei in der Regel aus einem Sender, einem Empfänger, einen Nachrichten Typen und einem Inhalt. Bei der *connect* Nachricht ist der Nachrichtentyp *connect* und der Inhalt, das Level des Senderfragments. Sobald die *connect* Nachricht beim Empfänger eintrifft, verbinden sich die beiden Fragmente zu einem neuen Level 1 Fragment. Der *branch* wird dazu zum Kern des Fragments und daraus ergibt sich die ID des neuen Fragments.[2]

Für den in Abbildung 8.2 dargestellten Graphen würden sich nach dem Sonderfall,

**Tabelle 8.3** Nachrichtentypen

Nachricht	Inhalt	Bedeutung
<i>initiate</i>	Level und ID	Bekanntgabe des Levels und der ID innerhalb eines Fragments
<i>test</i>	Level und ID	Anfrage wohin die Kante führt
<i>reject</i>		Antwort auf <i>test</i> : Kante ist <i>rejected</i>
<i>accept</i>		Antwort auf <i>test</i> : Kante ist <i>basic</i>
<i>report</i>		Gewicht der leichtesten <i>basic</i> Kante dem <i>core</i> bekannt geben
<i>change – core</i>		Blattknoten soll <i>connect</i> Nachricht versenden
<i>connect</i>	Level	Anfrage für Zusammenschluss

die in Abbildung 8.3 dargestellten Fragmente bilden. Dabei hat Fragment *F2* beispielsweise Fragment *F4* eine *connect* Nachricht gesendet. Die Kante wurde vor dem senden als *branch* markiert. *F4* hat ebenfalls mit einer *connect* Nachricht geantwortet und beide haben sich zu einem Level 1 Fragment *F2'* Zusammengeschlossen. Der *core* dieses neuen Fragments ist der *branch* über den die *connect* Nachrichten versendet wurden.

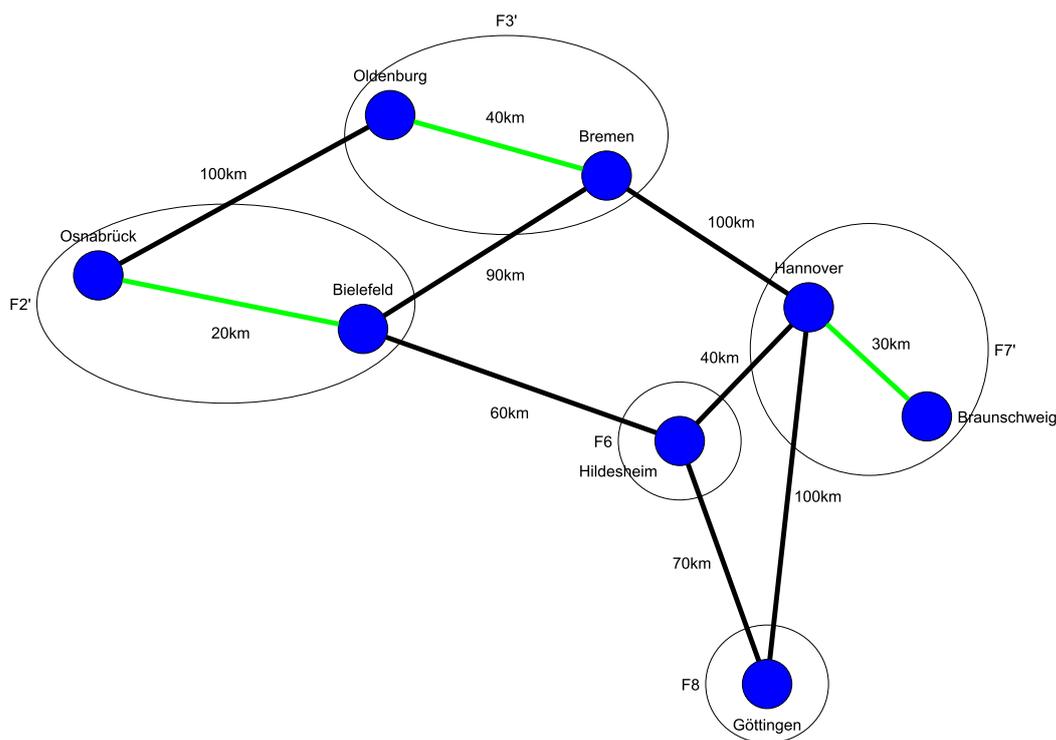
### *Broadcast*

Nach diesem Sonderfall existiert mindestens ein Level 1 Fragment. Der allgemeine Teil des Algorithmus kann in drei Bereiche, *Broadcast*, *Convergence* und *Change-Core*, aufgeteilt werden.[2]

Im *Broadcast* senden die beiden Knoten am Kern eine *initiate* Nachricht mit der neuen ID und dem neuen Level des Fragments über die *branches* an die anderen Knoten des Fragments. Diese kennen nach Empfang ihr Fragment und ändern ihren Status auf *find*.[2]

### *Convergence*

Im Status *find* sucht zuerst jeder Blattknoten seine leichteste *basic* Kante. Ein Blattknoten ist ein Knoten mit nur einem *branch*. Dazu sendet jeder Knoten über seine leichteste nicht *branch* Kante eine *test* Nachricht mit dem Level und der ID des eige-



**Abb. 8.3** Die ersten Level 1 Fragmente sind entstanden

nen Fragments. Der Empfängerknoten überprüft nun anhand der ID und des Levels, ob die Nachricht aus dem eigenen Fragment gesendet wurde. Ist dies der Fall, wird eine *reject* Nachricht als Antwort zurückgesendet und beide Knoten, also Sender und Empfänger, markieren die Kante als *rejected*. Ansonsten wird ein *accept* gesendet und beide Knoten wissen, dass diese Kante den Zustand *basic* hat. Danach wird iterativ mit der nächsten Kante weitergemacht, bis alle Kanten untersucht sind.[2] Nachdem die Blattknoten ihre jeweilige leichteste *basic* Kante gefunden haben, senden diese eine *report* Nachricht mit dem Gewicht ihrer leichtesten Kante über ihren *branch*. Die inneren Knoten warten darauf, dass sie über jeden eigenen *branch* außer einem, eine *report* Nachricht empfangen. Danach überprüfen sie mittels des oben schon beschriebenen Ablaufs, ihre leichteste *basic* Kante und senden dann über den verbleibenden *branch* entweder einen *report* mit der leichtesten *basic* Kante oder sie leiten den leichtesten *report* weiter. Zusätzlich speichert jeder Knoten die Kante deren Gewicht oder *report* weitergeleitet wird und setzt seinen Zustand von *find* auf *found*. Dieses Verfahren endet, sobald die beiden Kern-Knoten *report* Nachrichten empfangen. Falls kein Knoten seine leichteste *basic* Kante bestimmen kann, ist der minimale Spannbaum gefunden und der Algorithmus ist beendet.[2]

### *Change-Core*

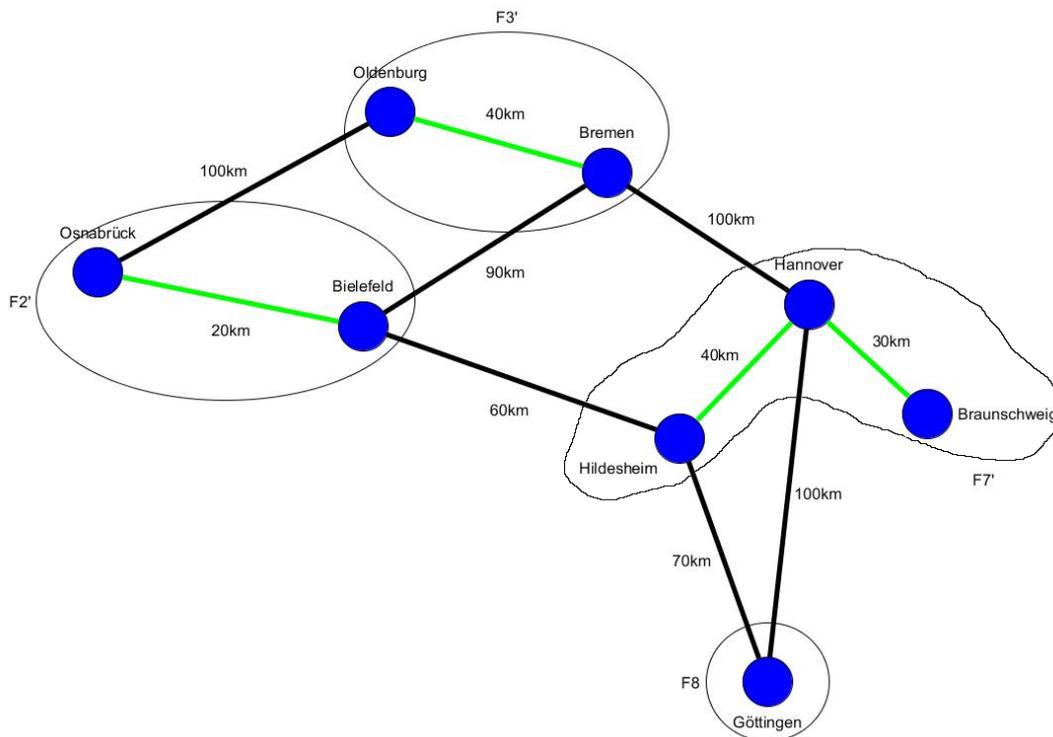
Nachdem der Weg zur leichtesten *basic* Kante des Fragments bekannt ist, beginnt der sogenannte Change-Core Abschnitt. Die beiden Kern Knoten senden nun über

den Weg zur leichtesten *basic* Kante des Fragments eine *change – core* Nachricht. Der Blattknoten dessen *basic* Kante die leichteste des Fragments ist, sendet nun über diese, genauso wie im Sonderfall oben beschrieben, eine *connect* Nachricht mit dem Level des eigenen Fragments und markiert die Kante als *branch*.

Empfängt ein Knoten eine *connect* Nachricht, überprüft dieser ob das Level des Senderfragments größer, kleiner oder gleich dem des eigenen Fragments ist. Wie oben schon erwähnt, gibt es zwei verschiedene Möglichkeiten zwei Fragmente zusammenzufügen. Wenn das Level des Sender Fragments kleiner als das des Empfänger Fragments ist, wird es absorbiert, dann ist der Kern, das Level und die ID des „neuen“ Fragments die des größeren der beiden. Die zweite Möglichkeit zum Verbinden zweier Fragmente ist der Zusammenschluss. Dies geschieht ausschließlich bei Fragmenten gleichen Levels. Dann ist der *branch* zwischen den beiden Fragmenten der neue Kern, die ID leitet sich von diesem Kern ab und das Level des neuen Fragments steigt um eins. Falls das Level des Senderfragments größer als des Empfängers ist, wird an dieser Stelle solange gewartet, bis der Sender ein geeignetes Level hat.[2]

Um die Entscheidung über den Zusammenschluss den anderen Knoten im neuen Fragment mitzuteilen, beginnt der Algorithmus wieder beim Broadcast mit dem Versenden einer neuen *initiate* Nachricht.[2]

Als Veranschaulichung dieses Prozesses dienen Abbildung 8.4, 8.5 und 8.6 welche in Folgendem erläutert werden.

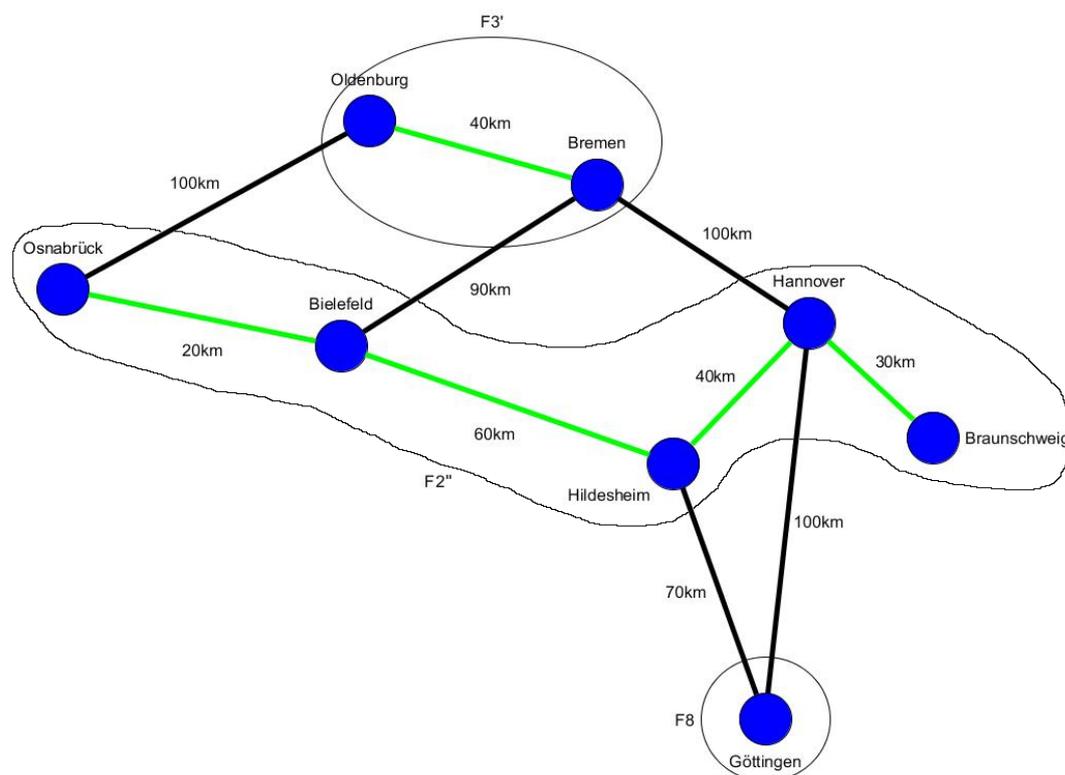


**Abb. 8.4** Fragmente nach dem ersten Change-Core

### Beispielszenario

Abbildung 8.4 zeigt die Absorption des Level 0 Fragments  $F_6$  durch das Level 1 Fragment  $F_7'$ . Dabei wurde in der Convergence die leichteste *basic* Kante der einzelnen Knoten bestimmt. Da Braunschweig keine *basic* Kante besitzt, ist die leichteste *basic* Kante von Hannover auch die des Fragments. Diese ist jene zwischen Hildesheim und Hannover, mit dem Gewicht von 40Km. Im Change-Core wird nun über diese Kante ein *connect* von Hannover gesendet. Da Hildesheim ebenfalls ein *connect* über diese Kante gesendet hat aber ein geringeres Level hat, wird das Fragment absorbiert.

Die Abbildung 8.5 zeigt den Zusammenschluss der Level 1 Fragmente  $F_7'$  und  $F_2'$



**Abb. 8.5** Fragmente nach dem zweiten Change-Core

zum Level 2 Fragment  $F_2''$ . Dabei wurde in der Convergence die leichteste *basic* Kante der einzelnen Knoten bestimmt und mittels *report* Nachrichten deren Gewicht zum *core* gesendet. In diesem Beispiel, ist die leichteste *basic* Kante für Hildesheim die Kante zu Bielefeld. Für Hannover, welches einen *report* von Hildesheim empfängt ist es eben dieser. Im Change-Core wird nun eine *change – core* Nachricht vom *core* zu Hildesheim gesendet. Dieser Knoten sendet über seine leichteste *basic* Kante, welche nun als *branch* markiert ist, eine *connect* Nachricht mit dem Level und der ID des eigenen Fragments. Da das Fragments  $F_2'$  ebenfalls ein *connect* über den *branch* sendet und beide Fragmente ein gleich hohes Level haben, verbinden

diese sich zum Fragment  $F2'$  mit dem Level 2. Dazu senden die beiden Empfänger der *connect* Nachrichten am *branch* nun *initiate* Nachrichten mit dem neuen Level und der neuen ID des neuen Fragments.

Die Abbildung 8.6 zeigt den minimalen Spannbaum innerhalb des Level 2 Frag-

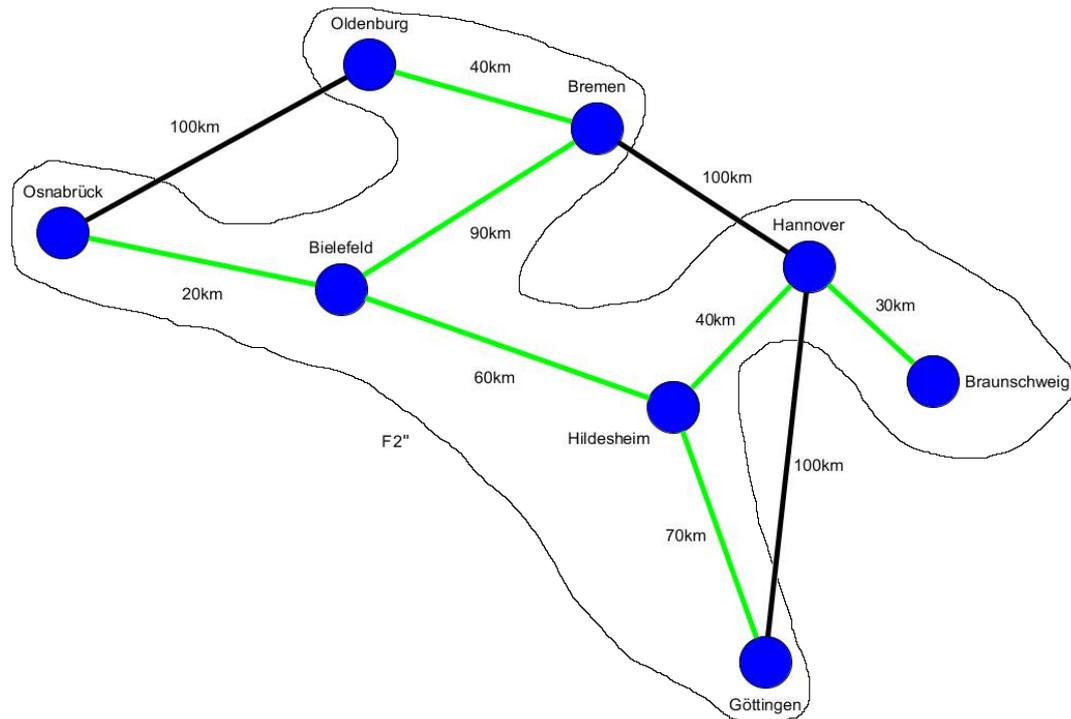


Abb. 8.6 Der minimale Spannbaum

ments  $F2''$ . Dazu wurden nach dem Zwischenschritt aus Abbildung 8.5 noch das Level 1 Fragment  $F3'$  und das Level 0 Fragment  $F8$  von  $F2''$  absorbiert.

### Aufwand

Den Aufwand Ihres Algorithmus, zum Bestimmen des minimalen Spannbaums eines Graphen, geben Gallager, Humblet und Spira als Zeitkomplexität und durch die maximal benötigte Anzahl an Nachrichten an. Die Zeitkomplexität ist in Formel (8.2) angegeben. [2] Die maximal benötigte Menge an Nachrichten beschreiben die Autoren mit der Formel (8.3). [2]

$$O(|N| * \text{Log}(|N|)). \quad (8.2)$$

$$O(5 * |N| * \text{Log}(|N|) + 2|E|). \quad (8.3)$$

### 8.3.2.3 Alternativen

Im Folgenden werden zwei weitere Verfahren, minimale Spannbäume verteilt zu Bestimmen, dargestellt. Zum einen ein Rekursiver Algorithmus und zum anderen ein nearest Neighbor Algorithmus. Bei beiden folgenden Beschreibungen wird sich auf den Vergleich zum GHS Algorithmus beschränkt.

#### *A Recursive, Distributed Minimum Spanning Tree Algorithm for Mobile Ad Hoc Networks*

Die Autoren dieses Papers stellen in diesem ein rekursives Verfahren zum Bestimmen minimaler Spannbäume dar. Dabei wird ein initialer Spannbaum mittels lokaler Optimierung verbessert. Das Verfahren soll hier nicht genauer untersucht werden, dennoch ist der unterschied zum GHS verfahren deutlich.[1] Hierbei wird ein initialer Spannbaum rekursiv verbessert während beim GHS Algorithmus ein minimaler Spannbaum verteilt generiert wird. Die Laufzeitkomplexität geben die Autoren mit der Formel (8.4) und die maximale Nachrichtenmenge mit (8.5) an.[1] Ein großer Nachteil dieses Verfahrens ist der Startspannbaum, in dem sich jeder Knoten zu Beginn befindet. Dafür müssen jedem Knoten zumindest seine Kanten bekannt sein, welche sich in dem Spannbaum befinden.

$$O(|N| * \text{Log}|N|). \quad (8.4)$$

$$O(|N| * \text{Log}(|N|) + |E|). \quad (8.5)$$

#### *A fast distributed approximation algorithm for minimum spanning trees*

Bei diesem Verfahren wird der minimale Spannbaum verteilt und mittels eines nearest neighbor Algorithmus erzeugt. Dazu bekommt jeder Knoten zu Beginn einen Rang und versucht sich mit dem nächsten Knoten höheren Ranges zu verbinden.[4] Die Laufzeitkomplexität geben die Autoren mit der Formel (8.6) und die maximale Nachrichtenmenge mit (8.7) an.[4]

$$O(D + L * \text{Log}(N)). \quad (8.6)$$

$$O(|E| * \text{Log}(L) * \text{Log}(N)). \quad (8.7)$$

## 8.4 Wiederaufbau eines Stromnetzes

Die Hauptaufgabe dieser Projektgruppe ist der sukzessive agentenbasierte Wiederaufbau eines Stromnetzes.[7] Dazu sollte im Idealfall ein Inselsystem entwickelt werden, welches autonom ohne Steuerung wächst. So eine Insel wäre beispielsweise der Zusammenschluss einiger Gemeinden und Stromlieferanten. Die verteilte Bestimmung minimaler Spannbäume, insbesondere der GHS-Algorithmus wäre in der

Lage den Wachstumsprozess dieser Inseln abzubilden. Dazu würde die Optimierung nicht über die Kantengewichte ablaufen, sondern über einen bilanzneutralen Energiehaushalt.[7] Um diesen zu bestimmen, könnte eine Implementierung des COHDA Algorithmus in Betracht gezogen werden. Dieser kann zum bestimmen optimierter Fahrpläne in Inseln verwendet werden. Außerdem stellt der GHS Algorithmus ein ziemlich genaues Kommunikationsmodell dar, welches als Grundlage für die Kommunikation des benötigten Multiagentensystems dienen kann.

## 8.5 Fazit

Die verteilte Bestimmung minimaler Spannbäume unterscheidet sich im Kern deutlich von Methoden wie dem Algorithmus von Kruskal. Besonders die Autonomie und die Fragmentierung sind hierbei zu nennen.

Durch eben diese ist es möglich mit Algorithmen wie dem GHS netzartige Strukturen in Inseln aufzubauen. Dieser Punkt wird auch im Abschnitt, *Wiederaufbau eines Stromnetzes*, deutlich. Dabei wird beschrieben, dass die verteilte Bestimmung minimaler Spannbäume hauptsächlich für den Wachstumsprozess und die dabei nötige Kommunikation verwendet werden könnte. Insgesamt ist der Ansatz sehr interessant und es sollten noch weitere Algorithmen analysiert werden, um einen Vergleich der Methodiken und des Aufwandes zu ermöglichen, da besonders bei großen Graphen eine hohe Anzahl an Nachrichten vermittelt werden muss.

# Literaturverzeichnis

- [1] Arslan, O., Koditschek, D.E.: A recursive, distributed minimum spanning tree algorithm for mobile ad hoc networks (2014)
- [2] GALLAGER, R., Humblet, P., Spira, P.: A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems* **5**(1), 66–77 (1983)
- [3] Katajainen, J., Nevalainen, O.: An alternative for the implementation of kruskal’s minimal spanning tree algorithm. *Science of Computer Programming* **3**(2), 205–216 (1983)
- [4] Khan, M., Pandurangan, G.: A fast distributed approximation algorithm for minimum spanning trees. *Lecture Notes in Computer Science* **4167**(1), 355–369 (2006)
- [5] Kim, S.: Kooperierende intelligente softwareagenten. *Wirtschaftsinformatik* **44**(1), 53–63 (2002)
- [6] Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* **7**(1), 48–50 (1956)
- [7] Lehnhoff, S., Sonnenschein, M., Bremer, J., Nannen, O., Hinrichs, C.: Pg einföhrung blackout/restart. *Foliensatz, Universität Oldenburg* (2016)
- [8] Tittmann, P.: *Graphentheorie: Eine anwendungsorientierte Einföhrung*, 2 edn. Carl Hanser Verlag (2011)