

Inauguraldissertation

zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaften
des Fachbereichs Wirtschaftswissenschaften
der Universität Osnabrück

Selbst-adaptive modellbasierte Entscheidungsunterstützungssysteme

Konzeption, prototypische Implementierung und
Evaluation einer Architektur für selbst-adaptive
Entscheidungsunterstützungsmodelle

vorgelegt von

Marcel-Philippe Breuer

Osnabrück, Dezember 2019

Dekanin: Prof. Dr. Valeriya Dinger

Referenten: Prof. Dr.-Ing. Bodo Rieger

Prof. Dr. Frank Teuteberg

Tag der Disputation: 19.12.2019

Vorwort

Die vorliegende Dissertation ist während meiner Zeit als wissenschaftlicher Mitarbeiter an der Universität Osnabrück am Fachgebiet für BWL / Management Support & Wirtschaftsinformatik des Instituts für Informationsmanagement und Unternehmensführung (IMU) entstanden. Die Fertigstellung dieser Arbeit wäre ohne die bedeutende Unterstützung von vielen verschiedenen Personen nicht möglich gewesen. Aus diesem Grund möchte ich mich nachfolgend bei diesen Personen bedanken.

Allen voran gebührt meinem Doktorvater Herrn Prof. Dr.-Ing. Bodo Rieger ein besonderer Dank. Durch die ständige Bereitschaft zum wissenschaftlichen Diskurs und das Einräumen vieler Freiräume im Rahmen des Forschungsprozesses hat er wesentlich zum Gelingen der Arbeit beigetragen. Ebenfalls möchte ich meinen Dank an Herrn Prof. Dr. Frank Teuteberg für die Übernahme des Korreferats richten. Hervorheben möchte ich die kollegiale und häufig freundschaftliche Zusammenarbeit am Fachgebiet sowie am Institut. Insbesondere die konstruktiven Diskussionen mit Dr. Axel Benjamins, Dr. Kirsten Liere-Netheler sowie Alexandre Witte haben mich im gesamten Forschungsprozess vorangetrieben und durch die freundschaftlichen Gespräche stets motiviert. Auch allen weiteren Kollegen am IMU danke ich herzlichst für die Zusammenarbeit und die gemeinsame schöne Zeit, besonders Alena Droit, Christian Große Starmann, Katharina Kröger, Albin Sonneck und Jutta Stelter.

Meiner Frau Ann-Katrin Breuer danke ich für die uneingeschränkte Unterstützung sowie dem ständigen positiven Zuspruch vor allem in weniger positiven Phasen des Forschungsprozesses. Insbesondere der Rückhalt in der Endphase der Dissertation bleibt unvergessen. Abschließend bedanke ich mich bei meinen Eltern Margarete und Walter Breuer, die mich seit jeher in allen Lebenslagen bestärken und ohne die das gesamte Promotionsvorhaben nicht möglich gewesen wäre.

Inhaltsverzeichnis

Inhaltsverzeichnis	IV
Abbildungsverzeichnis	VII
Tabellenverzeichnis.....	X
Listingverzeichnis	XI
Formelverzeichnis.....	XII
Abkürzungsverzeichnis	XIII
1 Einleitung.....	1
1.1 Motivation und Zielsetzung	1
1.2 Forschungsmethodik und Aufbau der Arbeit	3
I Die Bedeutung von modellgetriebenen Entscheidungsunterstützungssystemen (M-EUS) und der aktuelle Stand der Forschung von Selbst-Adaption für M-EUS	
2 Modellgetriebene Entscheidungsunterstützungssysteme zur betriebswirtschaftlichen Planung	7
2.1 Planungsprobleme in der Betriebswirtschaft	7
2.2 Modellgetriebene Entscheidungsunterstützungssysteme	10
2.2.1 Definition und Nutzenpotential	10
2.2.2 Entwicklung eines Modells	12
2.3 Grenzen von modellgetriebenen EUS in der betriebswirtschaftlichen Planung	15
3 Selbst-Adaption für robuste Systeme in einem dynamischen Unternehmensumfeld.....	19
3.1 Grundlagen und Einsatzpotential von selbst-adaptiven Systemen.....	19
3.2 Forschungsstand von Selbst-Adaption für M-EUS.....	23
3.2.1 Untersuchungsdesign und Vorgehensweise der Literaturlauswertung	23
3.2.2 Analyse der Rechercheergebnisse	25
3.2.2.1 Kriterien zur Auswertung der Rechercheergebnisse.....	25

3.2.2.2 Konzept-zentrierte Synthese der Ergebnisse 28

3.2.2.3 Schlussbetrachtung des aktuellen Forschungsstandes 36

3.3 Problemrelevanz im praktischen Einsatz von M-EUS..... 37

3.3.1 Untersuchungsdesign und Vorgehensweise 37

3.3.2 Ergebnisse der Umfrage 39

3.4 Forschungslücke und Zieldefinition..... 41

II Entwicklung und Evaluation eines Konzeptes für selbst-adaptive modellgetriebene Entscheidungsunterstützungssysteme

4 Entwicklung eines Konzeptes für selbst-adaptive modellgetriebene Entscheidungsunterstützungssysteme 44

4.1 Anforderungen an das Konzept..... 44

4.2 Entwicklung der Model-/System-Controller (MoSyCo) Architektur 51

4.2.1 Überblick über die Architektur..... 51

4.2.2 Entwurf des System-Controller 52

4.2.3 Entwurf des Observer..... 53

4.2.4 Entwurf des Model-Controller 54

5 Ex ante Evaluation der MoSyCo Architektur 62

5.1 Untersuchungsdesign und Hypothesen 62

5.2 Ergebnisse der Untersuchung..... 67

III Prototypische Implementierung und Evaluation

6 Prototypische Implementierung der MoSyCo Architektur 76

6.1 Überblick über die Komponenten des Prototyps 76

6.2 Repräsentation des Modells 78

6.2.1 Modellübersicht im proprietären Format 78

6.2.2 Modellübersicht im generischen Format 81

6.3 Umsetzung des System-Controller..... 84

6.4 Umsetzung des Observer..... 86

6.4.1 Angewandte Programmiersprache und verwendete Bibliotheken..... 86

6.4.2 Verzeichnisstruktur und Modulübersicht..... 87

6.4.3 Ausgabemöglichkeiten 91

6.4.4 Umsetzung des Reader 93

6.4.5 Umsetzung des Inspector 95

6.4.6 Umsetzung des Plotter..... 97

6.5	Umsetzung des Model-Controller	98
6.5.1	Angewandte Programmiersprache und verwendete Pakete	98
6.5.2	Zeitgesteuerte Prüfung neuer Abweichungen.....	99
6.5.3	Ereignisgesteuerte Ursachenanalyse	101
6.5.4	Methodik zur Entwicklung und Evaluation von Modellanpassungsvorschlägen.....	106
6.5.5	Durchführung von Modellanpassungen	110
6.6	Ganzheitliche Ablaufbeschreibung	111
7	Ex post Evaluation des MoSyCo Prototyp.....	118
7.1	Evaluationsziel und Hypothesen	118
7.1.1	Usability	118
7.1.2	Vertrauen in selbst-adaptive Systeme.....	120
7.2	Untersuchungsdesign	121
7.2.1	Mixed-Methods-Ansatz.....	122
7.2.2	Fragebogengestützter Usability Test	122
7.2.3	Aufgabe im Rahmen des Usability-Tests	125
7.2.4	Stichprobe und Einführungsveranstaltung	130
7.3	Untersuchungsdurchführung	133
7.4	Untersuchungsergebnisse	135
7.4.1	Vorgehen bei der Datenauswertung.....	135
7.4.2	Interpretation der Ergebnisse.....	135
7.4.2.1	Hypothese H7-1: Effektivität.....	136
7.4.2.2	Hypothese H7-2: Effizienz	139
7.4.2.3	Hypothese H7-3: Zufriedenheit	142
7.4.2.4	Hypothese H7-4: Vertrauen in automatisierte Systeme	145
7.4.2.5	Hypothese H7-5: Sicherheit über die Richtigkeit von Modellanpassungen	146
7.4.3	Zusammenfassung der Ergebnisse und Identifikation von Problemen und Verbesserungsmöglichkeiten.....	149
7.5	Limitationen der Untersuchung.....	151
8	Schlussbetrachtung	153
8.1	Zusammenfassung der Forschungsergebnisse	153
8.2	Ausblick auf weiterführende Forschungsarbeiten.....	155
	Anhang.....	XIV
	Literaturverzeichnis	LXIX

Abbildungsverzeichnis

Abbildung 1-1: Forschungsprozess der vorliegenden Arbeit mit Bezug zu den Kapiteln	4
Abbildung 2-1: Prognose- und Planungsprozess.....	8
Abbildung 2-2: Systematisierung der Prognosetechniken	9
Abbildung 2-3: Komponenten eines EUS	11
Abbildung 2-4: Generelle Struktur eines quantitativen Modells.....	13
Abbildung 2-5: Entscheidungsprozess nach Simon (1977, S. 40)	15
Abbildung 3-1: Hierarchie der Self-* Eigenschaften	20
Abbildung 3-2: MAPE-K Zyklus von selbst-adaptiven Systemen.....	21
Abbildung 3-3: Schlagwortwolke der Literaturergebnisse aus Kapitel 3.2.1.....	26
Abbildung 3-4: Boxplot der Probleme mit EUS in einer dynamischen Unternehmensumwelt	40
Abbildung 4-1: Model-/System-Controller (MoSyCo) Architektur	51
Abbildung 4-2: Datenmodell eines PGM für betriebswirtschaftliche EUS-Modelle.....	55
Abbildung 4-3: Property-Graph für einen exemplarischen Modellausschnitt	57
Abbildung 5-1: Verknüpfungen zw. MoSyCo Spezifikationen und initialen Problemen....	62
Abbildung 5-2: Ex ante Evaluation der Nutzungsintention und der wahrgenommenen Nützlichkeit.....	64
Abbildung 5-3: Boxplot der Ergebnisse zum Problem-Lösung-Fit.....	68
Abbildung 5-4: a) Spearman Korrelationskoeffizienten b) Boxplot der Variablen FIT, INT, PU	71
Abbildung 5-5: Boxplot zur Notwendigkeit bestimmter Funktionen der MoSyCo Architektur	72
Abbildung 6-1: Übersicht der Komponenten des Prototyps.....	76

Abbildung 6-2: Systemdynamisches Entscheidungsmodell in Vensim.....	79
Abbildung 6-3: Anteil der Gesamtnachfrage in Abhängigkeit der Abweichung vom Marktpreis	81
Abbildung 6-4: Generisches Entscheidungsmodell in neo4j.....	82
Abbildung 6-5: Cypher Abfrage und Eigenschaften einer Kante.....	83
Abbildung 6-6: exemplarische Gegenüberstellung zweier Läufe in Vensim	84
Abbildung 6-7: exemplarische Ursachenanalyse für den Tagesverkauf.....	85
Abbildung 6-8: Dateiverzeichnisstruktur der Observer Komponente.....	88
Abbildung 6-9: Modulübersicher der Observer Komponente.....	88
Abbildung 6-10: Observer Aktivitätsdiagramm, High-Level.....	90
Abbildung 6-11: GUI der Observer Komponente	92
Abbildung 6-12: Observer Aktivitätsdiagramm, Reader Element.....	94
Abbildung 6-13: Observer Aktivitätsdiagramm, Inspector	96
Abbildung 6-14: Observer Aktivitätsdiagramm, Plotter.....	97
Abbildung 6-15: Model-Controller Aktivitätsdiagramm, readlog Funktion	100
Abbildung 6-16: Model-Controller Aktivitätsdiagramm, StartModelController Funktion.....	102
Abbildung 6-17: Model-Controller Aktivitätsdiagramm, ApplyBSTS Funktion.....	103
Abbildung 6-18: Exponentielle Funktion der Gewichtung	105
Abbildung 6-19: Observer Sequenzdiagramm	113
Abbildung 6-20: Model-Controller Sequenzdiagramm.....	115
Abbildung 6-21: R Kommandozeilenoutput nach erfolgreicher Modellanpassung.....	116
Abbildung 6-22: Neo4j Output nach erfolgreicher Modellanpassung	117
Abbildung 7-1: Erwartete Nachfrage zum Entscheidungszeitpunkt 0	126
Abbildung 7-2: Abhängigkeit zwischen Realitäts- und Entscheidungsmodell	127
Abbildung 7-3: Entwicklung der Nachfrage zum Zeitpunkt 223 in Vensim.....	127
Abbildung 7-4: Realitätsmodell in Vensim	129

Abbildung 7-5: Boxplot über die Vorkenntnisse und Erfahrungen der Teilnehmer	131
Abbildung 7-6: Wirkungsänderung des Export innerhalb Störung 2 (vgl. Tabelle 7-2) ...	138
Abbildung 7-7: Boxplot zur Variable MA1_SQ002 gruppiert nach EG / KG	141
Abbildung 7-8: Boxplot der Variablen MA1_SQ006 und MA1_SQ007 in der EG.....	142
Abbildung 7-9: Boxplot der Variablen EM_SQ001 und PD3_SQ004	145
Abbildung 7-10: Boxplot der Variable MA1_SQ009 gruppiert nach EG / KG	147
Abbildung 7-11: Boxplot der Variablen VA1_SQ001 und VA1_SQ002.....	148

Tabellenverzeichnis

Tabelle 3-1: Einordnung der Recherche in die Taxonomie für Literaturlauswertungen	24
Tabelle 3-2: Berücksichtigte Quellen für die Literaturlauswertung	25
Tabelle 3-3: Themen-zentrierte Übersicht der Literaturlausergebnisse	29
Tabelle 3-4: Zu bewertende Aussagen basierend auf den sechs Self-* Eigenschaften	38
Tabelle 4-1: Zusammenfassung der Anforderungsspezifikationen an das Artefakt	50
Tabelle 5-1: Zu bewertende Aussagen über den Problem-Lösung-Fit	63
Tabelle 5-2: Aussagen zur Nutzungsintention sowie der wahrgenommenen Nützlichkei .	65
Tabelle 5-3: Zu bewertende Spezifikationen der MoSyCo Architektur	67
Tabelle 5-4: Korrelationsanalyse nach Spearman für FIT und INT	70
Tabelle 5-5: Korrelationsanalyse nach Spearman für FIT und PU	70
Tabelle 5-6: Mittelwerte zur Notwendigkeit bestimmter MoSyCo Funktionen.....	73
Tabelle 6-1: Erläuterung der Kommandozeilenparameter.....	89
Tabelle 6-2: Spaltennamen des DataFrame <code>newLines</code> inkl. Variablentypen.....	101
Tabelle 6-3: Mögliche Experimentalmodelle (A, B: unabhängige Variablen; C, D: freie Variablen).....	107
Tabelle 7-1: Fragenkatalog Usability-Test.....	124
Tabelle 7-2: Störungen zur Berechnung der Nachfrage im Realitätsmodell	128
Tabelle 7-3: Korrelationsmatrix über die Zufriedenheit der Leistung des Prototyps	144
Tabelle 7-4: Übersicht der Ergebnisse je Hypothese	149

Listingverzeichnis

Listing 6-1: Kommandozeilenparameter zum Start der Observer Komponente	89
Listing 6-2: Auszug des Loggings der Observer Komponente.....	91
Listing 7-1: Anweisung zum Ausführen des Observer	133

Formelverzeichnis

Formel 4-1: Berechnungslogik einer Subtraktion im PGM (vgl. Abbildung 4-3)	58
Formel 4-2: Berechnungslogik einer Division im PGM (vgl. Abbildung 4-3)	58
Formel 6-1: Berechnung der Preis-Absatz-Funktion in Vensim	80
Formel 6-2: Berechnung des Umsatzes in Vensim	81
Formel 6-3: Formel zur Gewichtung der Datenpunkte	105
Formel 6-4: Multiple lineare Regressionsgleichung	107
Formel 6-5: Fehlerterm einer linearen Regressionsgleichung.....	107
Formel 6-6: Fehlerterm einer gewichteten linearen Regressionsgleichung	108
Formel 6-7: Bestimmtheitsmaß R^2 einer linearen Regression.....	109
Formel 6-8: Wurzel der mittleren Fehlerquadratsumme einer linearen Regression.....	109
Formel 6-9: Mittlerer absoluter Fehler einer linearen Regression	110
Formel 7-1: Berechnung der Nachfrage in Vensim zum Zeitpunkt 0.....	126
Formel 7-2: Berechnung der Nachfrage in Vensim zum Zeitpunkt 95.....	128

Abkürzungsverzeichnis

API	Application Programming Interface
BP	Back Propagation
BSTS	Bayesian Structural Time Series
CSV	Comma-separated values
DE	Differential evolution
DIN	Deutsches Institut für Normung
DWH	Data Warehouse
EG	Experimentalgruppe
EN	Europäische Norm
ES	Evolutionsstrategie
FF	Firefly
FIT	Problem-Lösung-Fit
GA	Genetischer Algorithmus
GMM	Gaussian Mixture Model
GSA	Gravitational Search Algorithm
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
INT	Nutzungsintention
ISO	International Organization for Standardization
KG	Kontrollgruppe
KNN	Künstliche Neuronale Netze
MAE	Mean Absolute Error

MAS	Multi Agenten System
M-EUS	Modellgetriebene Entscheidungsunterstützungssysteme
MoSyCo	Model-/System-Controller Architektur
ODBC	Open Database Connectivity
PAF	Preisabsatzfunktion
PGM	Property Graph Model
PLE	Personal Learning Edition
PSO	Particle Swarm Optimization
PU	Wahrgenommene Nützlichkeit
R ²	Bestimmtheitsmaß
REST	Representational State Transfer
RFID	Radio-frequency identification
RMSE	Root-Mean-Square Error
SOM	Self-Organizing Map
SuOC	System under Observation and Control
TAM	Technology Acceptance Model
TLBO	Teaching-Learning-Based Optimization
UML	Unified Modeling Language
UVP	Unverbindlicher Verkaufspreis
VNS	Variable Neighborhood Search
XML	Extensible Mark-up Language

1 Einleitung

In diesem Kapitel soll die Problemstellung der vorliegenden Arbeit erläutert und darauf aufbauend die Zielsetzung der Arbeit definiert werden (vgl. Kapitel 1.1). Abschließend werden die verwendete Forschungsmethodik und der Aufbau der Arbeit präsentiert (vgl. Kapitel 1.2).

1.1 Motivation und Zielsetzung

Führungskräfte müssen Entscheidungen in einem immer komplexeren, sich stark verändernden und volatilen Geschäftsumfeld treffen (vgl. Power 2013, S. 1). Diese Umweltbedingungen erhöhen die Risiken für Fehlentscheidungen von Managern und Unternehmen. Um diese Risiken zu reduzieren und zu kontrollieren, werden potentielle Entscheidungsalternativen mit Hilfe von Entscheidungsunterstützungssystemen (EUS) evaluiert (vgl. Power 2013, S. 1). Diese computergestützte Lösung verspricht, den menschlichen Entscheidungsprozess effektiver und effizienter zu gestalten (vgl. Power 2013, S. 2), weshalb der Bedarf an Entscheidungsunterstützungssystemen in der Praxis sehr hoch ist (vgl. Power 2013, S. 1).

Die Entscheidungssituationen sind durch die Dynamik sowie Multi-Dimensionalität des betriebswirtschaftlichen Kontexts geprägt und können insbesondere durch den Einsatz von Modellen (sog. M-EUS) unterstützt werden (vgl. Turban et al. 2014, S. 43). Die Hauptkomponente eines modellgetriebenen EUS ist die Modellbank, die für die Bereitstellung eines geeigneten Modells in verschiedenen Problemsituationen verantwortlich ist (vgl. Turban et al. 2014, S. 96 f.). Die Qualität einer modellbasierten Entscheidungsunterstützung ist im Wesentlichen von einem validen sowie aktuellen Modell abhängig und dieses ist deshalb maßgeblich für den wahrgenommenen Nutzen, aber auch für die Akzeptanz eines modellbasierten EUS verantwortlich (vgl. Liang 1986, S. 3).

Die Validität eines Modells hängt u. a. von der Modellierung der Unternehmensumwelt ab (vgl. Sauter 2010, S. 130 f.), die zunehmend dynamischer ist und deshalb eine große Herausforderung für Entscheidungsunterstützungssysteme darstellt (vgl. Liu et al. 2009, S. 262). Um einen falschen Modelloutput zu verhindern, sollte für eine zuverlässige Entscheidungsgrundlage das Modell regelmäßig überprüft und ggf. aktualisiert werden (vgl. Power 2013, S. 75 f.). Diese kontinuierliche *Modellaktualisierung* (Problem 1) ist notwendig, um Fehler und Ungenauigkeiten zu vermeiden (vgl. Sauter 2010, S. 147), die in der

Regel durch die zeitabhängige und dynamische Realität hervorgerufen werden (vgl. Walker et al. 2003, S. 13 f.). Diese Fehler können bei der initialen Erstellung eines Modells (vgl. Oden und Prudhomme 2011, S. 262) oder durch eine Veränderung in der Unternehmensumwelt auftreten (vgl. Walker et al. 2003, S. 9). In beiden Fällen kann die Struktur des Modells oder eine falsche Kalibrierung von Parametern für die Fehler verantwortlich sein (vgl. Oden und Prudhomme 2011, S. 262; Walker et al. 2003, S. 9 f.).

Aufgrund der Multi-Dimensionalität von betriebswirtschaftlichen Planungsproblemen weisen die zur Lösung eingesetzten Entscheidungsunterstützungsmodelle eine hohe strukturelle Komplexität auf (vgl. Power 2002, S. 169). Diese Art der Komplexität wird durch die Vielzahl an Elementen und Beziehungen in einem Modell bedingt (vgl. Kirchhof 2013, S. 12). Durch die starke Dynamik in der Unternehmensumwelt ist auch die Ambiguität der jeweiligen Elemente hoch (vgl. Kirchhof 2013, S. 14). Diese unsicherheitsgeprägte Vieldeutigkeit bewirkt, dass neben der strukturellen Komplexität auch die funktionale Komplexität stark ausgeprägt ist (vgl. Kirchhof 2013, S. 18). Diese beiden *Komplexitätsausprägungen* (Problem 2) können zu Fehlern bei der initialen Modellerstellung führen, aber auch die notwendige kontinuierliche Modellaktualisierung beeinträchtigen.

Für die Erstellung und Wartung von Modellen sind in Unternehmen in der Regel Experten verantwortlich (vgl. Power 2013, S. 94), die sich durch eine hohe Expertise in einem bestimmten Bereich auszeichnen (vgl. Josefiok et al. 2015, S. 296). Sie sind jedoch selten in einem Unternehmen vertreten, häufig nicht verfügbar und überdurchschnittlich teure Arbeitskräfte (Josefiok et al. 2015, S. 292). Der *Aufwand* einer manuellen Modellwartung (Problem 3) ist deshalb für das Unternehmen ein teurer Prozess, der aufgrund der begrenzten Verfügbarkeit von Experten oft vernachlässigt wird, womit die Validität sowie Aktualität von Modellen nicht garantiert werden kann.

Zur Lösung der drei beschriebenen Probleme wird in dieser Arbeit das Prinzip der Selbst-Adaption untersucht, welches durch eine kontinuierliche Überwachung das Systemverhalten evaluiert und Anpassungen an einem System vornimmt, sofern die Vorgaben und Erwartungen nicht erfüllt werden oder die Funktionalität bzw. Leistung verbesserungsfähig ist (vgl. Laddaga 1999, S. 27). Im Rahmen der Forschung zur Selbst-Adaption von Systemen sind weitere Eigenschaften (sog. *Self*-* Eigenschaften) eines selbst-adaptiven Systems definiert sowie ein Vorgehensmodell (*MAPE-K* Zyklus) entwickelt worden, um eine zunehmende Dynamik in der Umwelt eines Systems regulieren zu können. Diese Erkenntnisse werden in der vorliegenden Arbeit genutzt, um Modelle innerhalb eines Entscheidungs-

unterstützungssystems mit Hilfe der Selbst-Adaption kontinuierlich realitätsnah zu repräsentieren, damit ein Entscheider stetig bestmöglich unterstützt werden kann. Hierfür wird eine Architektur entwickelt, die ein modellgetriebenes Entscheidungsunterstützungssystem auf Grundlage des Prinzips der Selbst-Adaption konzeptionell erweitert, und prototypisch realisiert. Zum einen wird damit das Ziel verfolgt, durch eine kontinuierliche Überwachung die Notwendigkeit einer *Modellaktualisierung* zu identifizieren. Zum anderen zielt die Erweiterung darauf ab, die *Komplexität* zu regulieren, indem Methoden für Modellanpassungen bereitgestellt werden, die sowohl parametrische als auch strukturelle Veränderungen ermöglichen. Sowohl die Überwachung des Systems als auch die Erstellung sowie Evaluation von Anpassungsmöglichkeiten des Modells sollen nach dem Prinzip der Selbst-Adaption automatisiert erfolgen, damit der *Aufwand* einer manuellen Modellwartung reduziert wird.

1.2 Forschungsmethodik und Aufbau der Arbeit

Zur Erreichung der Zielsetzung (vgl. Kapitel 1.1) orientiert sich die vorliegende Dissertation forschungsmethodisch an der Design Science Research (DSR) (Hevner et al. 2004). Dabei steht das Erstellen von innovativen Artefakten im Vordergrund, die zur Lösung von Problemen beitragen (vgl. Hevner et al. 2004, S. 75 f.). Mit jeder Konstruktion solcher Artefakte ist es erforderlich, dem *Build* und *Evaluate* Muster zu folgen (vgl. Hevner et al. 2004, S. 77). Die *Evaluate*-Phase ist in Anlehnung an Sonnenberg und vom Brocke (2012) aufgebaut, die den ursprünglichen Forschungsablauf von Hevner (2004) verfeinern und eine perspektivische Trennung der Evaluation in eine *ex ante* und *ex post* Phase vorschlagen (vgl. Sonnenberg und vom Brocke 2012, S. 392). Dabei werden die vier *Build*-Aktivitäten Problemidentifikation, konzeptionelles Design eines Artefaktes, prototypische Instanziierung eines Artefaktes und Nutzung eines Artefaktes jeweils durch die *Evaluate*-Aktivitäten eins bis vier überprüft (vgl. Sonnenberg und vom Brocke 2012, S. 392). Die *ex ante* Evaluation wird vor einer prototypischen Instanziierung durchgeführt und umfasst die *Evaluate*-Aktivitäten eins und zwei, die *ex post* Evaluation wird durch die dritte und vierte Evaluation definiert (vgl. Sonnenberg und vom Brocke 2012, S. 392). Neben dem *Build* und *Evaluate* Muster ist die Koexistenz der *Relevanz* und *Rigorosität* ein fundamentaler Aspekt von DSR (vgl. Hevner et al. 2004, S. 80). Demnach soll ein Forschungsprojekt die Ergebnisse aus dem *Build* und *Evaluate* Zyklus zur Sicherstellung der praktischen *Relevanz* in der Anwendungsdomäne prüfen (vgl. Hevner 2007, S. 88 f.). Für eine rigorose Forschung soll die Arbeit auf wissenschaftlichen Grundlagen basieren und die Erkenntnisse

aus dem Forschungsprozess sollen der wissenschaftlichen Wissensbasis hinzugefügt werden (vgl. Hevner 2007, S. 89 f.). Somit soll insgesamt eine rigorose, aber auch praxisrelevante Forschung durch DSR angestrebt werden.

Die vorliegende Arbeit orientiert sich an diesen Vorgaben und betrachtet gestaltungsorientierte Forschungsschritte (*Build*) sowie anschließende *Evaluationen*. Diese Aktivitäten können gemäß Abbildung 1-1 in drei Teile untergliedert werden, in denen jeweils ein Kapitel als ein *Build*- und ein *Evaluate*-Schritt interpretiert werden kann.

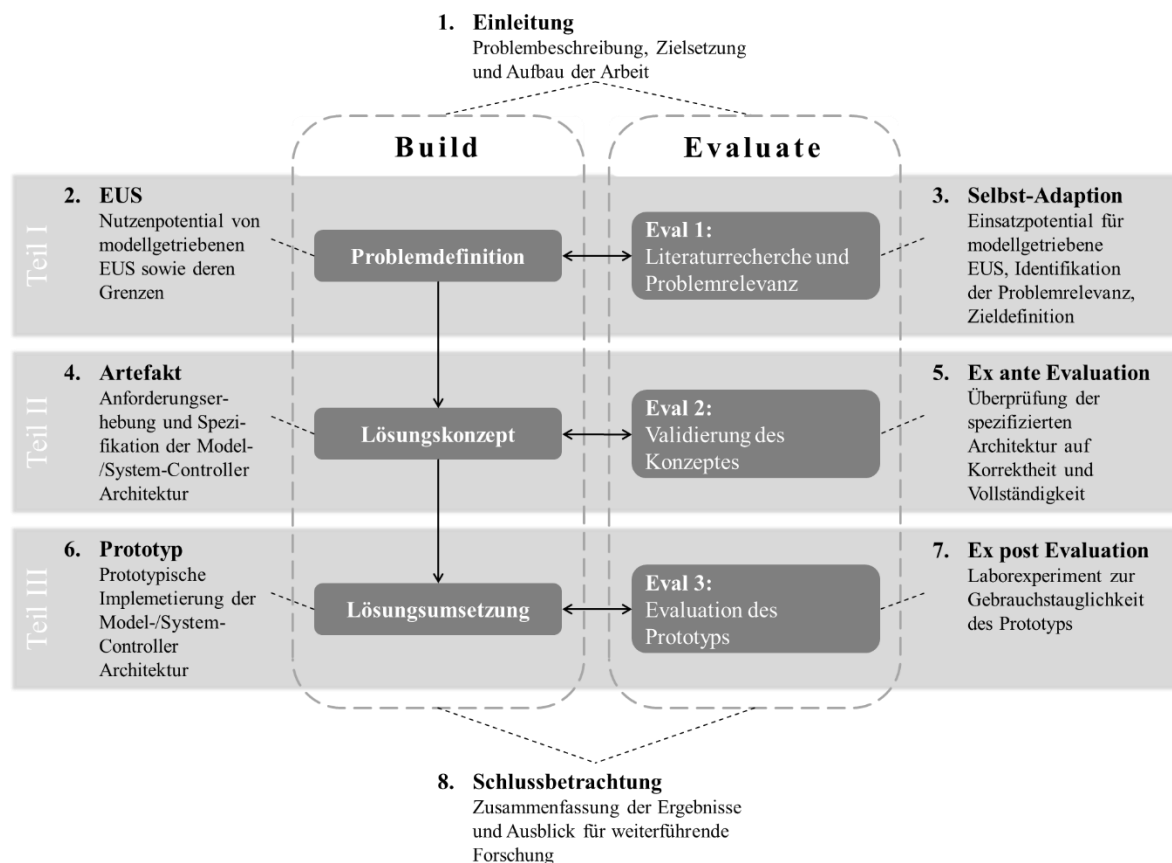


Abbildung 1-1: Forschungsprozess der vorliegenden Arbeit mit Bezug zu den Kapiteln

Quelle: In Anlehnung an Hevner et al. (2004, S. 77); Sonnenberg und vom Brocke (2012, S. 392)

In Kapitel eins findet die Identifikation sowie Motivation des Problems statt, worauf aufbauend ein potentieller Lösungsansatz vorgestellt und ein Ziel der Arbeit definiert wird (vgl. Kapitel 1.1). In Kapitel zwei werden die Grundlagen modellgetriebener Entscheidungsunterstützungssysteme erläutert und die Problemstellung der vorliegenden Arbeit anhand der Grenzen von modellgetriebenen EUS präzisiert. Anschließend wird in Kapitel drei der potentielle Lösungsansatz der Selbst-Adaption vorgestellt. Im Rahmen der Eval 1 wird eine Literaturrecherche durchgeführt, um verwandte Arbeiten und Theorien zu analysieren (vgl. Kapitel 3.2). Weiterhin wird durch eine Umfrage die Problemrelevanz in der

Praxis überprüft (vgl. Kapitel 3.3). Im Anschluss erfolgt in Kapitel 4 die Definition der Anforderungen, die von einer Lösung des Problems erfüllt werden müssen (vgl. Kapitel 4.1). Die Anforderungen basieren auf den Grundlagen zu dem Themengebiet der modellgetriebenen Entscheidungsunterstützungssysteme sowie den Prinzipien der Selbst-Adaption. Die Konzeption und Entwicklung einer innovativen Lösung findet in Kapitel 4.2 statt. Diese wird anschließend in Kapitel 5 durch eine Umfrage auf ihre Vollständigkeit und Korrektheit überprüft. Dieser Schritt entspricht der Eval 2 und schließt damit die ex ante Evaluation ab. Die prototypische Instanziierung des Lösungskonzeptes wird in Kapitel 6 beschrieben und im Rahmen eines Laborexperimentes in Kapitel 7 auf die generelle Gebrauchstauglichkeit getestet. Die Ergebnisse der vorliegenden Arbeit werden in der Schlussbetrachtung zusammengefasst und abschließend wird ein Ausblick auf mögliche weiterführende Forschungsvorhaben gegeben (vgl. Kapitel 8).

Der gesamte Forschungsprozess soll nach den Grundsätzen der Design Science Research durch eine Kommunikation der wissenschaftlichen Erkenntnisse als zusätzliches Evaluationsmedium komplettiert werden und für eine rigorose Forschung sorgen (vgl. Hevner et al. 2004, S. 90). Begleitend zum Forschungsprozess ist damit bereits begonnen worden (Breuer 2016, 2018); die Publikation der vorliegenden Dissertation als Monographie soll den Prozess abschließen.

TEIL I

Die Bedeutung von modellgetriebenen Entscheidungsunterstützungssystemen (M-EUS) und der aktuelle Stand der Forschung von Selbst-Adaption für M-EUS

2 Modellgetriebene Entscheidungsunterstützungssysteme zur betriebswirtschaftlichen Planung

Basierend auf der Motivation sowie der Zielsetzung der Arbeit (vgl. Kapitel 1.1) wird im Folgenden der betriebswirtschaftliche Planungsprozess (vgl. Kapitel 2.1) erläutert, der durch den Einsatz von modellgetriebenen Entscheidungsunterstützungssystemen verbessert werden kann (vgl. Kapitel 2.2). Abschließend werden zum besseren Verständnis der Problemstellung die Grenzen von modellgetriebenen EUS detailliert erläutert (vgl. Kapitel 2.3).

2.1 Planungsprobleme in der Betriebswirtschaft

Eine der zentralen Aufgaben der Unternehmensführung stellt die *Planung* dar (vgl. Wild 1980, S. 12). In der Literatur gibt es eine Vielzahl an Definitionen von „Planung“ (bspw. Adam 1996, S. 3; Dörner 1996, S. 234; Schmalen und Pechtl 2013, S. 99; Stölzle 2007, Sp. 1383). Dabei wird unter Planung häufig eine Beschreibung von Vorgehensweisen verstanden, die dazu dienen sollen, auf ein in der Zukunft liegendes selbstdefiniertes Ziel erfolgreich hinzuarbeiten (vgl. Schmalen und Pechtl 2013, S. 99), indem Handlungsmöglichkeiten für die Zukunft definiert werden (vgl. Dörner 1996, S. 234; Stölzle 2007, Sp. 1383). Für die Planung sind sowohl interne als auch externe Informationen grundlegend erforderlich, wobei die Daten entweder schon vorhanden sind oder noch beschafft werden müssen (vgl. Adam 1996, S. 35; Stölzle 2007, Sp. 1383). Informationen aus der Vergangenheit können dabei dazu dienen, potentielle zukünftige Zustände besser einschätzen und vorherzusagen zu können (Pieroth 2013, S. 30). Generell ist ein wichtiger Bestandteil von Planungen, die Konsequenzen unterschiedlichen Handelns zu verdeutlichen (vgl. Adam 1996, S. 5), um sich auf dieser Basis für eine aus mehreren Handlungsalternativen zu entscheiden (vgl. Pieroth 2013, S. 31f). Für den weiteren Verlauf der vorliegenden Arbeit wird die Planung im Unternehmenskontext auf Basis der in der Literatur gängigen Definitionen als ein Vorgang der *Zieldefinierung* (vgl. Stölzle 2007, Sp. 1383), der *Informationsbeschaffung und -auswertung* (vgl. Adam 1996, S. 3), sowie der Definition und Wahl verschiedener *Handlungsmöglichkeiten* (vgl. Dörner 1996, S. 234; Schmalen und Pechtl 2013, S. 99), die zur *Zielerreichung* führen können, unter Beachtung der jeweiligen *Konsequenzen* (vgl. Dörner 1996, S. 234) und unter *Unsicherheit* (vgl. Schmalen und Pechtl 2013, S. 99; Stölzle 2007, Sp. 1383), verstanden.

Die Informationen, die für die Aktivität einer Planung notwendig sind, werden im Zuge vorangegangener Prognosen für den Planungsprozess aufbereitet (vgl. Abbildung 2-1).



Abbildung 2-1: Prognose- und Planungsprozess

Quelle: In Anlehnung an Pieroth (2013, S. 32)

Die Rahmenbedingungen wie die Fristigkeit oder der Detaillierungsgrad einer Planung beeinflussen die Gestaltung des Planungsprozesses (vgl. Pieroth 2013, S. 32). Ein Verständnis dieser Rahmenbedingungen ist notwendig, um die Anforderungen an die vorgelagerte *Prognose* definieren zu können (vgl. Abbildung 2-1). Bereits in dieser Phase werden Daten verarbeitet, die im Zusammenhang mit den Entscheidungen im Rahmen der Unternehmensplanung stehen (vgl. Pieroth 2013, S. 31). *Vergangenheits- und Gegenwartsdaten* über Beobachtungen und Umweltzustände werden aufbereitet sowie analysiert und dienen als Basis zum Erstellen einer Prognose. Anhand von Prognosen werden Ereignisse beschrieben, die möglicherweise in der Zukunft eintreten können (*Zukunftsdaten*) (vgl. Bergmann und Bungert 2011, S. 266). Diese fließen in die *Planung* ein und unterstützen die *Entscheidungsfindung*.

Bei der Unternehmensplanung wird zwischen kurz-, mittel- und langfristigen Prognosen unterschieden (vgl. Pieroth 2013, S. 37). Das Spektrum an möglichen Ereignissen wird mit zunehmendem Prognosehorizont größer, denn damit wächst die Unsicherheit der zukünftigen Zustände (vgl. Vogel 2015, S. 14). Prognosen können auf unterschiedliche Ebenen des Managements ausgerichtet sein (vgl. Hanke und Reitsch 1981, S. 3). Um den Planungsprozess der verschiedenen Managementebenen zu unterstützen, können unterschiedliche *Prognoseverfahren* angewandt werden (vgl. Makridakis und Wheelwright 1989, S. 18). Die Wahl des Prognoseverfahrens hängt primär von dem zu prognostizierenden Zeitraum ab (vgl. Brockhoff 1977, S. 11) sowie vom Umfang und der Qualität der Daten (vgl. Brockhoff 1977, S. 12). Hinsichtlich der Auswahl des Verfahrens werden grundsätzlich zwei Ansätze zum Erstellen einer Prognose unterschieden (vgl. Treyer 2010, S. 28): quantitativ bzw. datenorientiert und qualitativ bzw. meinungsorientiert (vgl. Abbildung 2-2). Die *quantitative Prognosemethode* nimmt an, dass in der Vergangenheit erhobene Daten repräsentativ für die Zukunft sind (vgl. Makridakis und Wheelwright 1989, S. 51). *Qualitative Prognosemethoden* können zusätzlich zu den Vergangenheitsdaten auf Beurteilungen über

zukünftige Problemsituationen, die meist auf Expertenmeinungen basieren, beruhen (vgl. Vogel 2015, S. 11f). Diese Informationen können nicht immer faktisch belegt werden, sondern basieren oftmals auf subjektiven Einschätzungen (vgl. Vogel 2015, S. 12). Die Qualität einer qualitativen bzw. meinungsorientierten Prognose hängt stark von der Subjektivität des Experten ab (vgl. Götze 1991, S. 35). Eines der weitverbreitetsten quantitativen Prognoseverfahren stellt die *Zeitreihenanalyse* dar (vgl. Vogel 2015, S. 9), mit der *Entwicklungen* basierend auf historischen Daten prognostiziert werden (vgl. Adam 1996, S. 199). „Unter einer *Zeitreihe* versteht man die Entwicklung eines Merkmals (oder einer Merkmalsgruppe), dessen Werte zu bestimmten Zeitpunkten oder für bestimmte Zeitabschnitte im Zeitablauf erfasst und dargestellt werden“ (Mittelbach 2015, S. 51). Bei *Wirkungsprognosen* werden hingegen kausale Zusammenhänge zwischen Variablen und der Ausprägung des prognostizierten Ereignisses abgebildet (vgl. Adam 1996, S. 199). Für qualitative Prognosen werden u. a. Verfahren wie das *Brainstorming* oder auch *Befragungen* von Experten durchgeführt (vgl. Abbildung 2-2).

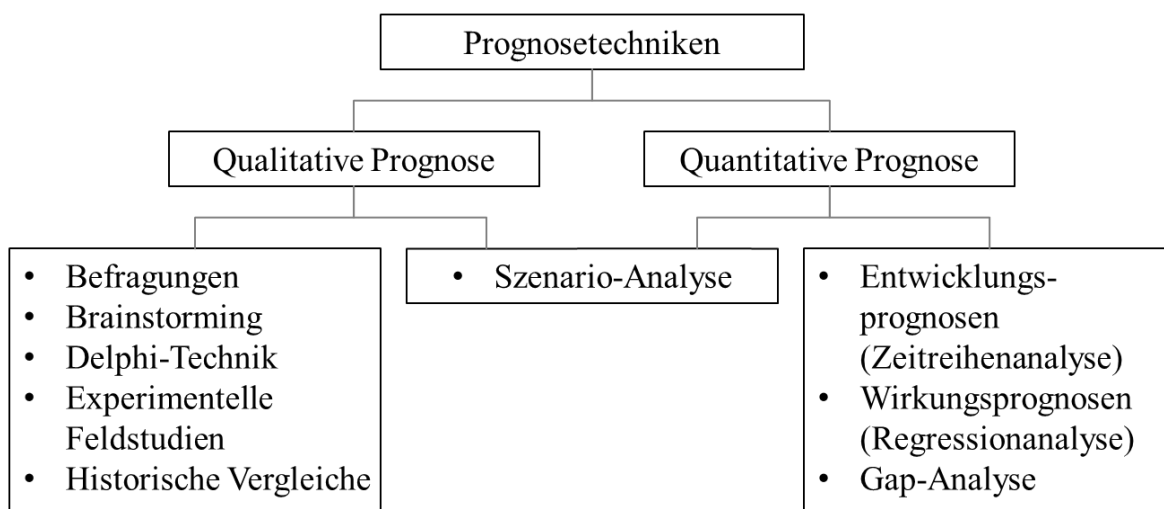


Abbildung 2-2: Systematisierung der Prognosetechniken

Quelle: In Anlehnung an Vogel (2015, S. 11)

Eine Kombination aus qualitativer und quantitativer Prognose stellt die *Szenario-Analyse* dar (vgl. Abbildung 2-2). „Unter einem Szenarium versteht man [...] eine hypothetische Folge von Ereignissen, die konstruiert wird, um kausale Zusammenhänge zwischen Entscheidungen und ihren Auswirkungen zu analysieren“ (Vogel 2015, S. 13). Es handelt sich um eine Verknüpfung der beiden Prognoseverfahren, da die Systemzusammenhänge in der Ausgangssituation mittels qualitativer Verfahren initial erfasst werden und die Zukunftsprognosen durch quantitative Verfahren berechnet werden (vgl. Vogel 2015, S. 14). Diese

Prognosen können den Planungsprozess unterstützen, indem diverse Handlungsalternativen gegeneinander bei der Entscheidungsfindung evaluiert werden und final eine zielgerichtete Entscheidung getroffen wird (vgl. Abbildung 2-1).

2.2 Modellgetriebene Entscheidungsunterstützungssysteme

Zur Definition des zentralen Forschungsgegenstandes wird nachfolgend der relevante Themenbereich der Entscheidungsunterstützungssysteme vorgestellt. Hierfür werden jene Systeme definiert und das Nutzenpotential zur betriebswirtschaftlichen Planung aufgezeigt (vgl. Kapitel 2.2.1). Die im Rahmen dieses Forschungsvorhabens fokussierte Klasse der modellgetriebenen EUS wird im Anschluss betrachtet und dabei näher auf die Entwicklung eines Modells eingegangen (vgl. Kapitel 2.2.2).

2.2.1 Definition und Nutzenpotential

Zur Unterstützung einer Entscheidungssituation, die regelmäßig und wiederkehrend im Rahmen des Planungsprozesses auftritt (vgl. Weber 2005, S. 271), werden häufig IT-Systeme eingesetzt (vgl. Keen und Scott Morton 1978, S. 1). Derartige entscheidungsorientierte Systeme werden Decision Support System (DSS) genannt (vgl. Gluchowski et al. 2008, S. 63; Power 2002, S. 2; Sprague und Carlson 1982, S. 4). Eine gängige deutsche Übersetzung lautet Entscheidungsunterstützungssysteme (EUS) (vgl. Gluchowski et al. 2008, S. 62). EUS sollen den Planungs- und Entscheidungsprozess effektiv unterstützen, indem das Urteilsvermögen eines Entscheiders erweitert wird (vgl. Gluchowski et al. 2008, S. 63) und damit die Entscheidungsqualität sowie die Geschwindigkeit zur Problemlösung verbessert werden (vgl. Power 2002, S. 32f). Damit werden die Möglichkeiten im Entscheidungsprozess durch den IT-Einsatz zwar erweitert, die Beurteilung des Menschen wird allerdings weiterhin benötigt, denn ein EUS ersetzt keine Personen (vgl. Turban et al. 2014, S. 75). EUS können unterschiedlich klassifiziert werden (vgl. Turban et al. 2014, S. 79). Die geläufigste Klassifikation von Power (2002, S. 12) basiert auf der Arbeit von Alter (1980) und unterscheidet zwischen modell-, wissens-, daten-, kommunikations- und dokumentenbasierten EUS. Bei einer Kombination von zwei oder mehr Klassen wird von zusammengesetzten EUS gesprochen (vgl. Turban et al. 2014, S. 79). Trotz dieser unterschiedlichen Ausprägungsformen ist in der Literatur eine generelle Architektur zu finden, die aus vier Komponenten besteht: einer Dialogschnittstelle, einer Datenbank, einer Modell- und Methodenbank und einer Ablaufsteuerung (vgl. Power 2002, S. 17). Die Kompo-

nenten sind in Abbildung 2-3 dargestellt und im Grundgedanken für die meisten EUS-Klassen gültig (vgl. Power 2002, S. 16 f.).

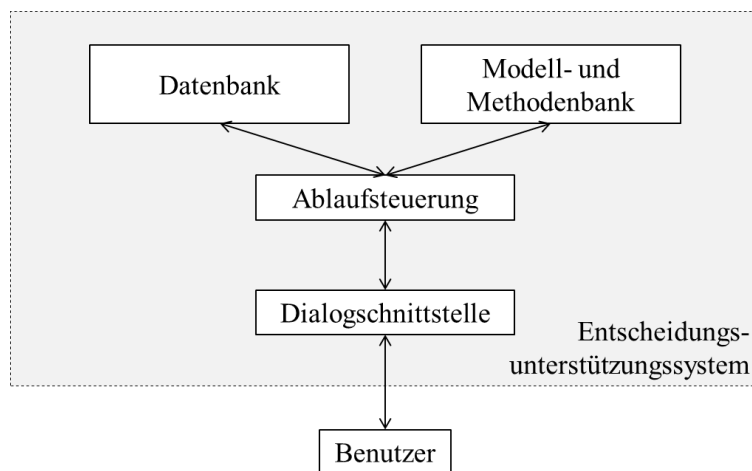


Abbildung 2-3: Komponenten eines EUS

Quelle: In Anlehnung an Power (2002, S. 17, 2013, S. 19)

Über die *Dialogschnittstelle* kann ein Benutzer gewünschte Aktionen oder Eingaben an das EUS übergeben, aber auch Informationen empfangen (vgl. Power 2002, S. 18). Die Akzeptanz eines EUS hängt maßgeblich von der Gestaltung der Dialogschnittstelle ab (vgl. Turban et al. 2014, S. 86), weshalb durch eine Separierung dieser Komponente eine höhere Anpassungsfähigkeit ermöglicht werden soll (vgl. Krallmann und Rieger 2001, S. 186). Die *Ablaufsteuerung* fungiert als Steuerelement zwischen den restlichen drei Komponenten (vgl. Krallmann und Rieger 2001, S. 186). In Abhängigkeit der Benutzeraktionen werden Daten ausgewählt und Modelle angewandt, um dem Benutzer gewünschte Ergebnisse zurückzugeben (vgl. Krallmann und Rieger 2001, S. 186). Die *Datenbank* stellt die notwendigen Informationen in Form von Daten aus der Gegenwart und Vergangenheit bereit (vgl. Power 2002, S. 18). In der Regel werden diese Informationen aus einer unternehmensinternen Datenbank bezogen (vgl. Turban et al. 2014, S. 86). In Abhängigkeit der Anwendungsdomäne können auch externe Quellen verwendet werden (vgl. Power 2002, S. 18). Die *Modell- und Methodenbank* greift in Abhängigkeit der Problemstellung auf unterschiedliche Modelle und Analysewerkzeuge zurück (vgl. Sauter 2010, S. 15). Diese Komponente sorgt für eine kontext- und modellsensitive Unterstützung des Benutzers (vgl. Sauter 2010, S. 15). Bei einem Modell handelt es sich um eine vereinfachte Repräsentation oder auch Abstraktion der Realität (vgl. Turban et al. 2014, S. 42). Die Analyse unterschiedlicher Szenarien (vgl. Kapitel 2.1) ist eine zentrale Aufgabe eines modellgetriebenen EUS (vgl. Power 2002, S. 33) und wird durch Werkzeuge aus der Methodenbank ermög-

licht (vgl. Power 2013, S. 36). Dies können bspw. Was-wäre-wenn-Analysen (engl. what-if analysis) sein (Drazan 1995, S. 418). Diese Analysen werden durch Modelle ermöglicht, anhand derer Simulationen durchgeführt und Szenarien gegeneinander evaluiert werden können (vgl. Power 2002, S. 171). Der Anwendungsfokus liegt bspw. auf der Optimierung einer Kennzahl oder der Simulation von Auswirkungen einer Entscheidung (vgl. Turban et al. 2014, S. 80).

2.2.2 Entwicklung eines Modells

Die Entwicklung eines Modells wird häufig von Experten durchgeführt (vgl. Power 2002, S. 158) und basiert auf subjektiven Einschätzungen über Variablenzusammenhänge oder auch Parameterschätzungen (vgl. Power 2002, S. 159). Besonders bei Modellen mit einem weitreichenden Prognosehorizont oder vielen externen Einflussfaktoren ist der subjektive und damit qualitative Einfluss bei der Modellerstellung stärker ausgeprägt (vgl. Power 2002, S. 168). In Abhängigkeit des Entscheidungsprozesses und der Zielsetzung können verschiedene Modelle gewählt werden (vgl. Turban et al. 2014, S. 143). Es gibt unterschiedliche Modelle für Optimierungsprobleme, die sich in der technischen Lösung unterscheiden (vgl. Turban et al. 2014, S. 144). Kann bspw. aufgrund einer zu hohen Komplexität rechnerisch nicht das Optimum bestimmt werden, gibt es für solche Fälle heuristische Modelle (vgl. Turban et al. 2014, S. 144). Simulationsmodelle werden genutzt, um eine gewünschte Handlungsalternative aus mehreren Möglichkeiten auszuwählen, indem potentielle Entwicklungen simuliert und bewertet werden (vgl. Turban et al. 2014, S. 144). Für diesen Zweck nutzen EUS quantitative Modelle (vgl. Power 2002, S. 169; Turban et al. 2014, S. 143), wohingegen qualitative Ansätze primär in Expertensystemen eingesetzt werden (vgl. Turban et al. 2014, S. 143). Der Fokus dieser Arbeit liegt auf modellgetriebenen EUS, die einen quantitativen Ansatz haben, weshalb im Folgenden die Komponenten eines quantitativen Modells zur Entscheidungsunterstützung betrachtet werden.

Quantitative Modelle zur Entscheidungsunterstützung bestehen aus vier Komponenten: Ergebnisvariablen, Entscheidungsvariablen, Variablen oder auch Parameter, die nicht direkt kontrolliert werden können (unkontrollierbare Variablen), und Hilfs- bzw. Zwischenvariablen (vgl. Abbildung 2-4). Diese Komponenten werden durch mathematische Operatoren miteinander verknüpft, womit die Wirkungsstruktur eines Modells definiert wird. Darüber hinaus kann es in Abhängigkeit der Problemsituation notwendig sein, Verzögerungseffekte modellieren zu können (vgl. Bossel 2004a, S. 118).

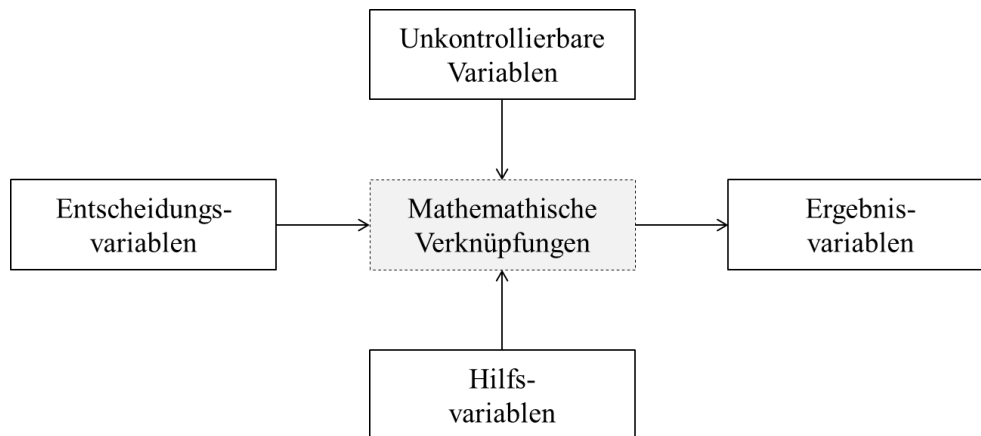


Abbildung 2-4: Generelle Struktur eines quantitativen Modells

Quelle: In Anlehnung an Turban et al. (2014, S. 145)

Die Entscheidungsvariablen bestimmen durch ihren Wert die Höhe bzw. Ausprägung der Ergebnisvariablen. Mit den Entscheidungsvariablen werden unterschiedliche Handlungsalternativen beschrieben, Beispiele hierfür sind Entscheidungen über Investmentbeträge oder das verfügbare Werbebudget (vgl. Turban et al. 2014, S. 146). Die Entscheidungen werden anhand der Ergebnisvariablen bewertet. Diese bestimmen, wie effektiv oder zielführend eine Handlungsalternative ist (vgl. Turban et al. 2014, S. 146). Die Ergebnisvariablen hängen von den Entscheidungs-, Hilfs- sowie unkontrollierbaren Variablen ab und werden deshalb auch als abhängige Variablen umschrieben (vgl. Turban et al. 2014, S. 146). Bei Hilfsvariablen handelt es sich um Variablen, die zur Verständlichkeit des Modells beitragen und als Zwischenspeicher von Ergebnissen verwendet werden (vgl. Turban et al. 2014, S. 147). Beispielsweise kann die Mitarbeiterproduktivität (entspricht der Ergebnisvariable) durch vorgelagerte Hilfsvariablen wie die Zufriedenheit und das Gehalt des Mitarbeiters modelliert werden (vgl. Turban et al. 2014, S. 147). Bei den unkontrollierbaren Variablen handelt es sich um Faktoren, die nicht durch den Entscheidungsträger beeinflusst werden können und somit externe Einflüsse darstellen (vgl. Turban et al. 2014, S. 146). Diese Faktoren stammen aus der Unternehmensumwelt und über deren Entwicklung sowie Wirkung müssen gegebenenfalls Annahmen getroffen werden. Während des Modellierungsprozesses werden die relevanten Variablen identifiziert und die Abhängigkeiten untereinander quantifiziert (vgl. Turban et al. 2014, S. 145). Zur Identifikation der Wirkungsstruktur innerhalb eines Modells, d. h. Variable A wirkt auf B, ist in der Regel das Fachwissen eines Experten notwendig (vgl. Bossel 2004a, S. 27). Zur Quantifizierung einer Wirkung, d. h. zur Ermittlung der Wirkungsintensität, können hingegen computergestützte Analysemethoden eingesetzt werden (vgl. Bossel 2004a, S. 22). Oftmals handelt es sich bei den unkontrollierbaren

Variablen um Limitationen für den Entscheidungsträger, weshalb sie auch als Restriktionen interpretiert werden können (vgl. Turban et al. 2014, S. 146). Diese Restriktionen müssen nicht starr sein, sondern können auch als eine Funktion der Zeit implementiert werden und wirken damit dynamisch (vgl. Bossel 2004a, S. 43). Die externen Umwelteinflüsse haben einen entscheidenden Einfluss auf das Systemverhalten, besonders wenn diese grundlegende Wirkungsbeziehungen des Systems abschwächen oder stärken (vgl. Bossel 2004a, S. 43).

Über die externen Umwelteinflüsse und ihre Wirkung besteht in der Praxis jedoch selten eine vollständige Transparenz, sodass die Wirkungsstruktur und die Parameter nur teilweise ermittelt werden können (vgl. Bossel 2004a, S. 53). In diesem Fall entsteht eine Mischform aus den beiden Modellierungsansätzen „Verhalten nachahmen“ und „System nachbilden“ (vgl. Bossel 2004a, S. 52f). Wenn das Systemverhalten durch ein Modell *nachgeahmt* wird, wird das Verhalten des nachzubildenden Systems durch eine beliebige Modellstruktur erreicht. Dabei ist das Originalsystem oftmals eine Blackbox und die tatsächliche Wirkungsstruktur ist nicht nachvollziehbar, weshalb auf Basis von Verhaltensbeobachtungen ein Modell gebildet wird, um eine Nachahmung des Verhaltens zu erzielen (vgl. Bossel 2004a, S. 52). Wird hingegen das System *nachgebildet*, wird die wesentliche Systemstruktur auch in dieser Art im Modell abgebildet. Demnach ist die Wirkungsstruktur des Originalsystems bekannt (Whitebox) und es kann ein strukturgetreues Modell nachgebildet werden (vgl. Bossel 2004a, S. 53). Die Mischform dieser beiden Ansätze ist in der Realität am häufigsten vertreten (vgl. Bossel 2004a, S. 53). Die Wirkungsstruktur und Parameter können in diesem Fall nur teilweise ermittelt werden. Es wird versucht, die Wirkungsstruktur des Systems nach besten Kenntnissen nachzubilden. Anhand von Verhaltensbeobachtungen wird versucht, die unbekannteren Strukturelemente sowie Modellparameter nachzuahmen (vgl. Bossel 2004a, S. 53). Bei diesem Greybox-Ansatz sind die Wirkungszusammenhänge in den Grundzügen bekannt. Insbesondere bei betriebswirtschaftlichen Entscheidungsmodellen ist keine volle Transparenz gegeben und der Mischformansatz wird verfolgt, da z. B. über das Kaufverhalten der Kunden nur Annahmen getroffen werden können (vgl. Jockisch und Rosendahl 2009, S. 35). Das Kaufverhalten der Kunden entspricht in diesem Fall externen Unternehmenseinflüssen, die in ihrer Struktur unbekannt sind und nachgeahmt werden müssen. Hierfür wird der verhaltensbeschreibende Ansatz zur Bildung jener Modellteile eingesetzt (vgl. Bossel 2004a, S. 53). Das Modell zur Entscheidungsunterstützung ist in

diesem Fall jedoch nur für jene Bedingungen und Aktionen gültig, die auch in dem betrachteten Beobachtungszeitraum aufgetreten sind (vgl. Bossel 2004a, S. 54).

2.3 Grenzen von modellgetriebenen EUS in der betriebswirtschaftlichen Planung

Die Abhängigkeit zwischen einer erfolgreichen oder nicht erfolgreichen Entscheidung und dem Modellierungsprozess hat Simon (1977) in dem Grundlagenwerk zum Entscheidungsprozess verdeutlicht. Der Prozess kann in die Phasen *Intelligence*, *Design*, *Choice* und *Review* untergliedert werden (vgl. Simon 1977, S. 41).

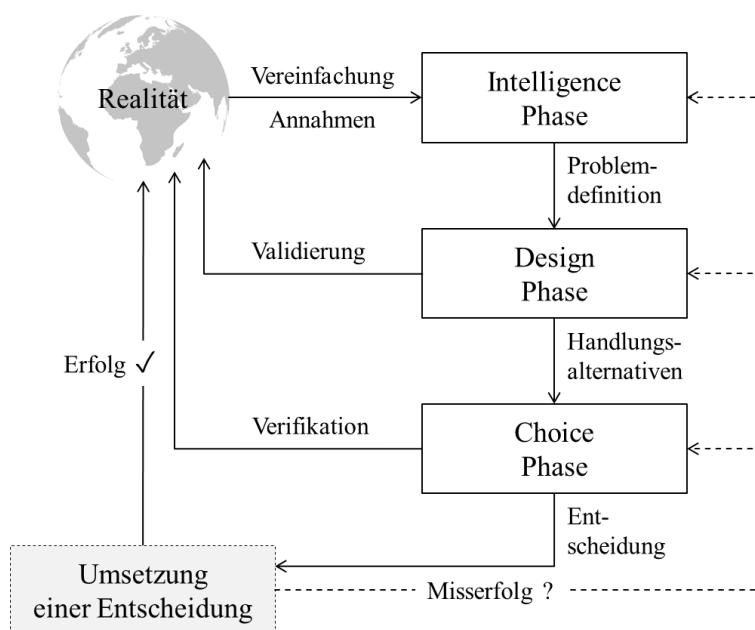


Abbildung 2-5: Entscheidungsprozess nach Simon (1977, S. 40)

Quelle: In Anlehnung an Turban et al. (2014, S. 44)

In der Phase *Intelligence* wird das Problem fokussiert und die Unternehmensumwelt nach Faktoren, die eine Entscheidung oder auch die Wirkung einer Entscheidung beeinflussen können, untersucht (vgl. Simon 1977, S. 40). Im Vordergrund dieser Phase steht die Diagnose der Entscheidungssituation, um diese möglichst detailliert beschreiben zu können (vgl. Simon 1977, S. 40) und als Ergebnis eine Problemdefinition formuliert zu haben (vgl. Abbildung 2-5). Diese Phase hängt davon ab, wie Menschen bei der Analyse eines Problems vorgehen. Nach dem funktionalistischen Ansatz von Newell und Simon (1972) ist es wichtig, dass der Mensch eine interne Repräsentation des Problems erzeugen kann, damit das Problem ganzheitlich verstanden werden kann (vgl. Newell und Simon 1972, S. 810).

In der *Design* Phase wird ein Modell erstellt, welches die zur Lösung des zugrundeliegenden Problems relevanten Faktoren entsprechend repräsentiert. In diesem Schritt werden

Annahmen getroffen und die Realität vereinfacht, damit eine finale Wirkungsstruktur für das Modell entwickelt werden kann (vgl. Turban et al. 2014, S. 44). Jeder Modellentwurf in dieser Phase soll validiert werden, indem für unterschiedliche Lösungsalternativen die Entwicklung durch das Modell simuliert wird und diese Werte anhand der Daten aus der Realität validiert werden (vgl. Turban et al. 2014, S. 44).

In der Phase *Choice* wird aus den potentiellen Handlungsalternativen eine Lösung ausgewählt (vgl. Simon 1977, S. 41). Diese wird auf ihre Durchführbarkeit in der Realität geprüft (Turban et al. 2014, S. 44). Nach einer erfolgreichen Verifikation wird die Entscheidung umgesetzt (vgl. Simon 1977, S. 41).

In der letzten Phase *Review* wird eine getroffene Entscheidung nach Erfolg oder Misserfolg bewertet (vgl. Simon 1977, S. 41). Eine Entscheidung ist dann erfolgreich, wenn das initiale Problem durch die Entscheidung gelöst werden kann (vgl. Turban et al. 2014, S. 44). Sind die Auswirkungen einer Entscheidung nicht wie erwartet, erfolgt eine Rückkopplung zu den vorangegangenen drei Phasen (vgl. Abbildung 2-5), um die Ursachen des Fehlverhaltens zu analysieren.

Die Unternehmensumwelt hat einen starken Einfluss auf den gesamten Entscheidungsprozess (vgl. Turban et al. 2014, S. 8). Insbesondere eine rasante technologische Entwicklung sowie zunehmende gesellschaftliche Einflüsse stellen externe Faktoren dar, die den Entscheidungsprozess bedingen und die maßgeblich die Komplexität erhöhen (vgl. Turban et al. 2014, S. 8f). Komplexität entsteht durch die Interaktion der Elemente untereinander sowie mit der Systemumwelt und wird durch die Vielzahl und Vielfalt der Elemente sowie ihrer Relationen bestimmt (vgl. Kirchhof 2013, S. 18). Diese Einflussfaktoren prägen die strukturelle Komplexität eines Systems (vgl. Kirchhof 2013, S. 17). Die strukturelle Komplexität wird auch als objektive Komplexität bezeichnet, da diese ausschließlich von dem zu modellierenden System abhängt (vgl. Kirchhof 2013, S. 12). In der betriebswirtschaftlichen Planung handelt es sich in der Regel um Problemstellungen, deren Abhängigkeiten mehrdimensional sind und die deshalb eine hohe strukturelle Komplexität aufweisen (vgl. Power 2002, S. 169). Im Gegensatz dazu beschreibt die Dimension der funktionalen Komplexität, wie die Vieldeutigkeit der strukturellen Komplexität erfasst und abgebildet wird und wie damit umgegangen wird (vgl. Kirchhof 2013, S. 18). Konkret geht es um menschliche bzw. subjektive „Schwierigkeiten im Bestimmen von Zielen, im Erkennen von Unsicherheiten, Unbestimmtheiten, Abhängigkeiten, Mustern und Risiken, und in der Auswahl der jeweils relevanten System- und Umwelt-System-Ausschnitte“ (Kirchhof 2013, S. 18).

Insbesondere in den Phasen Intelligence und Design (vgl. Abbildung 2-5) sind die jeweiligen Ergebnisse maßgeblich von der Handhabung der Komplexität abhängig (vgl. Simon 1977, S. 46). Aufgrund der hohen strukturellen Komplexität bei der modellgetriebenen betriebswirtschaftlichen Planung (vgl. Power 2002, S. 169) sowie einem durch den Menschen bedingtes begrenztes Verarbeitungsvermögen im Umgang mit der funktionalen Komplexität (vgl. Kirchhof 2013, S. 18) kann die *Komplexität* im Allgemeinen beim Einsatz von modellgetriebenen EUS in der unternehmerischen Planung zu Problemen führen.

In der Design Phase von betriebswirtschaftlichen Planungsproblemen findet aufgrund intransparenter Wirkungsstrukturen von externen Unternehmenseinflüssen häufig der verhaltensbeschreibende Ansatz Anwendung (vgl. Kapitel 2.2.2). Da der verhaltensbeschreibende Ansatz auf Beobachtungen aus der Vergangenheit basiert, ist die Gültigkeit dieser Modellteile für zukünftige Entwicklungen nicht immer gegeben. Verändern sich die Bedingungen, die zum Zeitpunkt der Modellentwicklung wirkten, ist das Modell nicht mehr repräsentativ (vgl. Bossel 2004a, S. 54). Die Güte einer Entscheidung sowie die Akzeptanz des eingesetzten EUS hängen jedoch maßgeblich von der Qualität des angewandten Modells ab (vgl. Liang 1986, S. 3; Pieroth 2013, S. 28). Eine regelmäßige Überprüfung und Aktualisierung des Modells ist deshalb notwendig, um Fehler und Ungenauigkeiten zu verhindern (vgl. Sprague und Carlson 1982, S. 262). Angesichts der rasanten technologischen Entwicklung (vgl. Turban et al. 2014, S. 8f) müssen diese Überwachungs- und Analyseprozesse in Abhängigkeit zu den Aktualisierungsraten der Umwelt stehen. Deshalb wird für EUS empfohlen, dass die Überwachung und Analyse nahezu in *Echtzeit* zu erfolgen hat (vgl. Pick und Weatherholt 2012, S. 8). Zur Gewährleistung dieser regelmäßigen Aktualisierungen soll bei der Entwicklung eines EUS die Verantwortlichkeit für die Wartung und Pflege eines Modells spezifiziert werden (vgl. Power 2002, S. 91). Trotz dieser Ansätze kommt es aufgrund einer dynamischen Unternehmensumwelt häufig zu Fehlern bei dem Einsatz von modellgetriebenen EUS (vgl. Walker et al. 2003, S. 13f). Diese Fehler können entstehen, wenn das Modell initial erstellt wird (vgl. Oden und Prudhomme 2011, S. 262) oder wenn sich im Laufe der Zeit die Umweltbedingungen ändern (vgl. Walker et al. 2003, S. 9). In beiden Fällen können Strukturbrüche oder Parameteränderungen für die Fehler verantwortlich sein (vgl. Bossel 2004a, S. 56; Oden und Prudhomme 2011, S. 262; Walker et al. 2003, S. 9 f.). Beim Einsatz von modellgetriebenen EUS kann es demnach in einer dynamischen Unternehmensumwelt zu Problemen kommen, wenn eine *Aktualisierung* und *Wartung* des zugrundeliegenden Modells nicht regelmäßig erfolgt.

Die Wartung der Modelle hängt von den dafür verantwortlichen Personen ab. In der Regel sind in einem Unternehmen Experten für die Entwicklung von modellgetriebener Unterstützung verantwortlich (vgl. Power 2013, S. 94). Experten zeichnen sich durch eine hohe Expertise in einem bestimmten Bereich aus und können überdurchschnittlich gut Problemsituationen auch bei einer unvollständigen oder unpräzisen Informationsgrundlage lösen (vgl. Josefiok et al. 2015, S. 296). Unternehmen haben jedoch Schwierigkeiten entsprechende Experten zu rekrutieren. Falls diese trotzdem vorhanden sind, sind sie häufig nicht verfügbar und darüber hinaus überdurchschnittlich teure Arbeitskräfte (Josefiok et al. 2015, S. 292). Der *Aufwand* einer Modellwartung ist deshalb für Unternehmen ein teurer Prozess und kann sich aufgrund der begrenzten Verfügbarkeit von Experten negativ auf die Häufigkeit der Aktualisierungen auswirken.

Zusammengefasst können drei Probleme von modellgetriebenen Entscheidungsunterstützungssystemen identifiziert werden. Aufgrund der von Dynamik geprägten Unternehmensumwelt ist sowohl die strukturelle als auch die funktionale *Komplexität* von betriebswirtschaftlichen Modellen sehr hoch. Durch die rasante technologische Entwicklung ist eine kontinuierliche *Wartung und Aktualisierung* des Modelles notwendig. Wird diese Wartung manuell durchgeführt, ist dieser *Aufwand* mit sehr hohen Kosten verbunden.

3 Selbst-Adaption für robuste Systeme in einem dynamischen Unternehmensumfeld

Aufgrund der steigenden Dynamik in der Unternehmensumwelt ist es notwendig, auf Veränderungen reagieren zu können (vgl. Kapitel 2.3). Die gestiegene Dynamik wird auch in der Literatur zu Entscheidungsunterstützungssystemen erkannt, weshalb eine generische EUS-Umgebung für zukünftige Forschungsvorhaben vorgeschlagen wird, in der sich das System in Abhängigkeit unterschiedlicher Entscheidungssituationen und Umweltfaktoren selbst konfigurieren und anpassen kann, um eine möglichst hohe Anpassungsfähigkeit zu ermöglichen (vgl. Liu et al. 2009, S. 280). Das Prinzip der Selbst-Adaption stellt einen Lösungsansatz dar, der die geforderte Selbst-Konfiguration sowie -Anpassung ermöglicht und deshalb im Folgenden erläutert und in Hinblick auf die Eignung untersucht wird (vgl. Kapitel 3.1). Im Anschluss wird mit Hilfe einer Literaturlauswertung der aktuelle Forschungsstand selbst-adaptiver modellgetriebener Entscheidungsunterstützungssysteme vorgestellt (vgl. Kapitel 3.2). Neben dieser wissenschaftlichen Perspektive wird die Relevanz des Problems in der Praxis im Rahmen einer Umfrage zur Abdeckung der Eval 1-Aktivität (vgl. Kapitel 1.2) untersucht (vgl. Kapitel 3.3). Auf dieser Basis wird die Forschungslücke und die Zielsetzung für die Konzeption eines selbst-adaptiven modellgetriebenen Entscheidungsunterstützungssystems beschrieben (vgl. Kapitel 3.4).

3.1 Grundlagen und Einsatzpotential von selbst-adaptiven Systemen

Selbst-adaptive Systeme evaluieren das eigene Verhalten und verändern dieses, wenn die Evaluation ergibt, dass das System nicht die Vorgaben und Erwartungen erfüllt oder wenn die Funktionalität bzw. Leistung verbesserungsfähig ist (vgl. Laddaga 1999, S. 27). Das Präfix „Selbst“ drückt aus, dass das System autonom entscheidet, wie es sich selbst verwaltet, damit Anpassungen an die Umwelt umgesetzt werden können (vgl. Macías-Escrivá et al. 2013, S. 7268). Einige selbst-adaptive Systeme können gänzlich ohne ein menschliches Eingreifen funktionieren, da eine Steuerung durch übergeordnete Zielvorgaben (z. B. durch definierte Strategien) hilfreich ist und deshalb in manchen Systemen so eingesetzt wird (vgl. Brun et al. 2009, S. 49).

Das Prinzip der Selbst-Adaption kann in sechs Eigenschaften (sogenannte *Self*-* Eigenschaften) zerlegt werden, die in einer dreistufigen Hierarchie angeordnet werden können

(vgl. Andersson et al. 2009, S. 38). In dieser Hierarchie ist die Selbst-Adaption eine übergeordnete und grundlegende Eigenschaft, die auf zwei verschiedenen Ebenen in Haupt- und Primitiveigenschaften zerlegt wird (vgl. Abbildung 3-1).

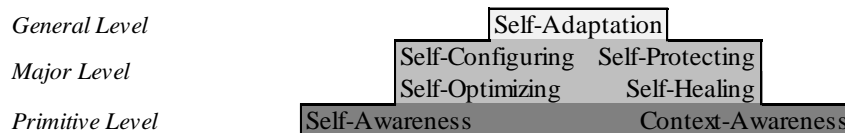


Abbildung 3-1: Hierarchie der Self-* Eigenschaften

Quelle: In Anlehnung an Salehie und Tahvildari (2009, S. 5)

Die grundlegende bzw. allgemeine Ebene (*General Level*) enthält die Eigenschaft der Selbst-Adaption. Ein System mit einer solchen Eigenschaft besteht typischerweise aus vielen interagierenden Komponenten, denen teilweise nur begrenzte Informationen über das Gesamtsystem zur Verfügung stehen (vgl. Oreizy et al. 1999, S. 58). Die Komponenten übernehmen unterschiedliche Funktionen, deren Aufgaben in den unteren Hierarchieebenen definiert werden.

Das *Major Level* besteht aus den vier Eigenschaften *Self-Configuring*, *Self-Protecting*, *Self-Optimization* und *Self-Healing* (vgl. Abbildung 3-1). *Self-Configuring* beschreibt die Eigenschaft, dass ein System sich automatisiert anpasst und dies in Abhängigkeit von übergeordneten Strategien passiert, die sich bspw. an den Geschäftszielen orientieren (vgl. Kephart und Chess 2003, S. 42). Die Eigenschaft der *Self-Optimization* umfasst die Steuerung der Leistung und Ressourcenzuweisung, um auf Anforderungen verschiedener Benutzer oder Umstände zu reagieren (vgl. Lalanda et al. 2013, S. 36). Dabei ist sich das System der optimalen Leistung bewusst und kann die aktuelle Leistung messen sowie beurteilen (vgl. Sterritt 2005, S. 80). *Self-Healing* ermöglicht das Entdecken, Diagnostizieren und Reagieren auf Störungen (vgl. Kephart und Chess 2003, S. 43). Das Antizipieren von potentiellen Problemen ist ebenfalls eine relevante Aufgabe, die durch das Diagnostizieren ermöglicht werden soll, um präventive Aktionen einzuleiten (vgl. Salehie und Tahvildari 2009, S. 6). *Self-Protecting* ist die Fähigkeit, im Speziellen Sicherheitsrisiken zu erkennen und gegensteuernde Maßnahmen einzuleiten, um drohende Probleme zu vermeiden (vgl. Lalanda et al. 2013, S. 36).

Auf der untersten Ebene, dem *Primitive Level*, werden die Eigenschaften *Self-Awareness* und *Context-Awareness* geführt (vgl. Abbildung 3-1). *Self-Awareness* beschreibt die Fähigkeit, dass ein System den eigenen Zustand sowie das eigene Verhalten beurteilen kann (vgl. Sterritt 2005, S. 80). Die *Context-Awareness* bezieht sich auf das operative sowie externe

Umfeld, das ebenfalls erfasst und analysiert werden muss (vgl. Salehie und Tahvildari 2009, S. 6). Da das primäre Ziel eines selbst-adaptiven Systems die Anpassung an wechselnde Umweltbedingungen ist, hat der Kontext des Systems einen maßgeblichen Einfluss. Kontext kann definiert werden als jede Information, die verwendet werden kann, um die Situation einer Entität zu charakterisieren, die für ein selbst-adaptives System relevant ist (vgl. Lalanda et al. 2013, S. 28). Der Kontext selbst kann durch unterschiedliche Eigenschaften charakterisiert werden (vgl. Lalanda et al. 2013, S. 29). Manche Kontexte können vollständig beobachtbar sein. In diesem Fall liegt jede Art von Information zu jeder Zeit vor. Im entgegengesetzten Fall können Informationen über den Kontext nur teilweise verfügbar sein. Hierbei handelt es sich häufig um zufallsbedingte Umgebungen, bei denen ein zukünftiger Zustand des Systems von getätigten Aktionen abhängt, aber auch von einigen unbekanntem Faktoren beeinflusst werden kann, die nicht vorherzusagen sind (vgl. Lalanda et al. 2013, S. 30). Die Dynamik des Kontextes bestimmt, wie häufig und wie stark sich die Umgebung eines Systems ändert und hat ebenfalls einen Einfluss auf die Leistung eines selbst-adaptiven Systems (vgl. Lalanda et al. 2013, S. 30).

Zur Umsetzung der oben aufgeführten Prinzipien und Eigenschaften innerhalb eines selbst-adaptiven Systems schlagen Kephart und Chess (2003) den sogenannten *MAPE-K* Zyklus vor. Es handelt sich um das Akronym für *Monitor, Analyse, Plan, Execute* und *Knowledge* (vgl. Kephart und Chess 2003, S. 44).

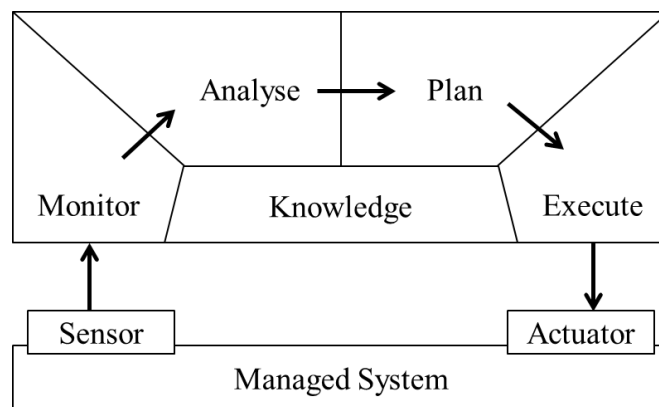


Abbildung 3-2: MAPE-K Zyklus von selbst-adaptiven Systemen

Quelle: In Anlehnung an Kephart und Chess (2003, S. 44)

Der Zyklus spezifiziert die Abfolge von Prozessen, die zur Umsetzung eines selbst-adaptiven Systems notwendig sind. Über *Sensoren* wird das zu kontrollierende (*managed*) System im Schritt *Monitor* überwacht (vgl. Kephart und Chess 2003, S. 44). In der Phase *Analyse* werden die beobachteten Daten untersucht und bewertet, was bspw. anhand von

Grenzwerten, ob ein Anpassungsbedarf am System besteht, möglich ist (vgl. Brun et al. 2013, S. 34). Sind Anpassungen erforderlich, werden entsprechende Maßnahmen in der Komponente *Plan* entwickelt und evaluiert (vgl. Brun et al. 2013, S. 35). Eine geplante Maßnahme wird in der Phase *Execute* im überwachten und kontrollierten System via *Aktuatoren* umgesetzt (vgl. Kephart und Chess 2003, S. 44). Jede Prozesskomponente kann auf die *Knowledge Bank* zugreifen, um relevante Ereignisse zu protokollieren und für zukünftige Analysen zur Verfügung zu stellen (vgl. Brun et al. 2009, S. 59).

Durch die untergeordneten Eigenschaften der Selbst-Adaption (vgl. Abbildung 3-1) sowie den *MAPE-K* Zyklus (vgl. Abbildung 3-2) wird der Grundgedanke verfolgt, dass ein selbst-adaptives System so handelt, dass eine vorgegebene Leistung in einer sich verändernden Umwelt realisiert wird (vgl. Macías-Escrivá et al. 2013, S. 7268). Konkret werden selbst-adaptive Systeme eingesetzt, um eine hohe Komplexität unter dynamischen Rahmenbedingungen zu handhaben (vgl. Salehie und Tahvildari 2009, S. 2). Insbesondere die *Context-Awareness* stellt ein wichtiges Prinzip dar, mit Hilfe dessen beim Einsatz von modellgetriebenen Entscheidungsunterstützungssystemen der dynamische Handels- bzw. Geschäftsmarkt überwacht werden kann (vgl. Kapitel 2.3). Auch ein robustes Verhalten gegenüber unerwarteten Ereignissen ist ein Einsatzgrund von selbst-adaptiven Systemen und wird durch die Eigenschaft *Self-Configuring* ermöglicht (vgl. Salehie und Tahvildari 2009, S. 2), womit im Rahmen von M-EUS zum Beispiel eine stark schwankende und teilweise intransparente Nachfrage von Kunden besser erfasst werden kann (vgl. Kapitel 2.3). Durch diese Eigenschaft kann die strukturelle Komplexität (vgl. Kapitel 2.3) von modellgetriebenen Entscheidungsunterstützungssystemen reduziert werden. Die funktionale Komplexität (vgl. Kapitel 2.3) kann durch das autonome Handeln eines selbst-adaptiven Systems ebenfalls eingeschränkt werden, indem bestimmte Aufgaben dem Menschen durch das System abgenommen werden. Konkret stellt die Eigenschaft *Self-Configuring* für diesen Fall einen automatisierten Lösungsansatz, bspw. durch den Einsatz von Algorithmen, bereit, der neben einer Komplexitätsreduktion auch zu einer Senkung des Wartungs- und Aktualisierungsaufwandes beitragen kann (vgl. Kapitel 2.3). Das Monitoring des *MAPE-K* Zyklus sowie konkret die *Self-Healing* Eigenschaft sorgen für eine stetige Kontrolle des zu überwachenden Systems, womit die Aufgabenlast der Experten und der Wartungsaufwand weiter reduziert werden können.

Die eingangs identifizierten Probleme modellgetriebener Entscheidungsunterstützungssysteme in Form einer erhöhten Komplexität, der Notwendigkeit häufiger Aktualisierungen

sowie eines hohen menschlichen Wartungsaufwands könnten durch die aufgeführten Eigenschaften eines selbst-adaptiven Systems sowie dem in der Literatur vorgeschlagenen *MAPE-K* Zyklus potentiell reduziert oder gar gelöst werden. Das Einsatzpotential von *selbst-adaptiven* Systemen im Rahmen von Entscheidungsunterstützungssystemen ist somit groß. Konkret setzt die Anpassung eines betriebswirtschaftlichen *Modells* kontinuierliche Überwachungs- und Analyseprozesse voraus, die aufgrund der steigenden dynamischen Unternehmensumwelt in nahezu *Echtzeit* erfolgen sollen (vgl. Kapitel 2.3). Auf Grundlage dieser kursiv formatierten Begriffe wird im Folgenden der Forschungsstand einer kontinuierlichen Selbst-Adaption von modellgetriebenen Entscheidungsunterstützungssystemen näher untersucht.

3.2 Forschungsstand von Selbst-Adaption für M-EUS

Zur Untersuchung des Forschungsstandes wird zu Beginn das Untersuchungsdesign und die Vorgehensweise der Literaturlauswertung erläutert (vgl. Kapitel 3.2.1). Im Anschluss werden die Ergebnisse analysiert und vorgestellt (vgl. Kapitel 3.2.2).

3.2.1 Untersuchungsdesign und Vorgehensweise der Literaturlauswertung

Der Forschungsstand wird durch die Analyse und Systematisierung von Forschungsarbeiten im Bereich der modellgetriebenen Entscheidungsunterstützungssysteme sowie der Selbst-Adaption untersucht. Zur Veranschaulichung des Untersuchungsdesigns und der Vorgehensweise der durchgeführten Literaturrecherche wird in der Tabelle 3-1 die empfohlene Taxonomie (vgl. Randolph 2009, S. 8) nach Cooper (1988) verwendet. Die Einordnung der durchgeführten Literaturrecherche im Rahmen dieser Forschungsarbeit erfolgt durch die farblich hervorgehobenen Felder.

Der **Fokus** der Recherche liegt auf den Forschungsergebnissen der im Rahmen der Literaturrecherche identifizierten Beiträge. Das **Ziel** der Untersuchung ist es, die zentralen Aspekte der Forschung zu ermitteln, wofür frühere Forschungsarbeiten analysiert werden, um Forschungsfragen für zukünftige Arbeiten identifizieren zu können (vgl. Cooper 1988, S. 109). Bei der Untersuchung wird eine neutrale Position eingenommen, indem die existierende Literatur primär zusammengefasst und auf eine Interpretation anhand persönlicher Einstellungen verzichtet wird (vgl. Cooper 1988, S. 11).

Tabelle 3-1: Einordnung der Recherche in die Taxonomie für Literaturlauswertungen

Quelle: Eigene Tabelle in Anlehnung an Cooper (1988, S. 107 ff.)

Merkmale	Kategorien			
Fokus	Forschungs- ergebnis	Forschungs- methode	Theorien	Erfahrungen o. Anwendungen
Ziel	Integration	Identifikation zentraler Aspekte		Kritik
Perspektive	Neutrale Darstellung		Vertretung eines Standpunktes	
Literaturumfang	Zentral oder grundlegend	Repräsentativ	Selektiv	Vollständig
Organisation	Historisch	Thematisch		Methodisch
Zielgruppe	Spezialisierte Wissenschaftler	Allgemeine Wissenschaftler	Praktiker	Öffentlichkeit

Es werden die beiden Datenbanken Web of Science sowie EBSCO host durchsucht, da diese u. a. für Recherchen im Bereich der Wirtschaftsinformatik empfohlen werden (vgl. Schryen 2015, S. 299). Die EBSCO host Datenbank greift wiederum auf eine Vielzahl anderer Datenbanken wie bspw. Business Source Premier, EconLit oder MLA International Bibliography zu (vgl. Schryen 2015, S. 299). Auf Basis der zu untersuchenden Forschungsfelder (vgl. Kapitel 3.1) wurde im März 2015 der Suchstring ((((((“self adaptive”) OR (“self-adaptive”) OR (“auto adaptive”) OR (“auto-adaptive”)))) AND (“continuous”) OR (“real-time”) OR (“real time”)))) AND (“Model” OR “Modelling” OR “Parameter”)) gebildet und zum Durchsuchen der Datenbanken verwendet. Die Suche resultierte in beiden Datenbanken in zusammen 52 Treffer. Aufgrund von sieben Dubletten wurden final 45 Beiträge identifiziert (vgl. Tabelle 3-2). Da noch weitere Datenbanken für wissenschaftliche Publikationen existieren (vgl. Schryen 2015, S. 299), wird von einem selektiven **Literaturumfang** ausgegangen (vgl. Tabelle 3-1).

Die **Organisation** der Literaturlaufbereitung erfolgt thematisch, sodass Beiträge mit demselben abstrakten Konzept gemeinsam betrachtet und vorgestellt werden (vgl. Cooper 1988, S. 112). Da die Literaturrecherche im Rahmen einer Dissertation stattfindet, richtet sich die Auswertung primär an die **Zielgruppe** der spezialisierten Wissenschaftler, kann aufgrund des sehr anwendungsorientierten Ansatzes der Entscheidungsunterstützungssysteme allerdings auch für die Praxis von Relevanz sein.

Tabelle 3-2: Berücksichtigte Quellen für die Literaturlauswertung

Quelle: Eigene Tabelle

Datenbank	Suchgebiet	Restriktionen	Ergebnisse	Relevant
EBSCO host	Title & Abstract	Scholarly (Peer Reviewed)	11	9
Web of Science	Topic & Title	Categories: "Computer Science", „Business Economics“, OR "Management Science"	34	27
			45	36

In den 45 identifizierten Beiträgen sind mehrere enthalten, die aus fachfremden Bereichen stammen. Die vorliegende Arbeit ist im Forschungsbereich der Wirtschaftsinformatik angesiedelt, weshalb für die Synthese der Literatur die Betriebswirtschaftslehre sowie die Informatik und im weitesten Sinne auch die Ingenieurwissenschaften im Fokus stehen. Insgesamt werden neun Beiträge aufgrund ihres Anwendungsbereiches im Rahmen einer ersten Bewertung als nicht relevant eingestuft (vgl. Schryen 2015, S. 302). Darunter fallen fünf Beiträge aus dem Bereich der Medizin (Herrera et al. 2009; Klein 1991; Metzger 2000; Wang et al. 2014; Xue et al. 2007), einer aus dem Bereich der Meteorologie (Schreider et al. 2001), jeweils einer aus dem Bereich der Biologie (Sbarciog et al. 2013) sowie der Chemie (Zhu et al. 2013) und ein Beitrag aus der Musikwissenschaft (Wang und Huang 2010). Dies führt zu 36 als relevant identifizierte Beiträge, welche im Folgenden analysiert werden (vgl. Tabelle 3-2).

3.2.2 Analyse der Rechercheergebnisse

Mit dem Ziel, die Rechercheergebnisse strukturiert analysieren zu können, werden im Vorfeld Kriterien zur Auswertung hergeleitet (vgl. Kapitel 3.2.2.1). Anhand dieser Kriterien werden die Ergebnisse Konzept-zentriert dargestellt (vgl. Kapitel 3.2.2.2) und die Erkenntnisse zur Bewertung des Forschungsstandes abschließend zusammengefasst (vgl. Kapitel 3.2.2.3)

3.2.2.1 Kriterien zur Auswertung der Rechercheergebnisse

Um einen Überblick über die Inhalte der Ergebnisse der Literaturrecherche zu bekommen, wird eingangs eine Analyse der Schlagworte (Keywords) betrieben. Weiter soll auf Basis von häufigen Schlagworten eine Struktur zur Präsentation sowie zur Analyse der Literatur abgeleitet werden (vgl. Schryen 2015, S. 305).

Für die Analyse wurden im Vorfeld die Schlagwörter in Anlehnung an die Stammformreduktion beim Text Mining bearbeitet (vgl. Aggarwal 2018, S. 17 f.). So wurde stets der Singular (bspw. System anstatt Systems) sowie das Substantiv von Adjektiven (bspw. Adaptation anstatt Adaptive) verwendet. In einem zweiten Schritt wurden die Leerzeichen und Bindestriche innerhalb eines Schlagwortes entfernt, sodass durch die Anwendung dieser Regeln aus „Self-Adaptive“ „SelfAdaptation“ entsteht.

Für die Schlagwortanalyse wird die absolute Häufigkeit berücksichtigt. Ein Schlagwort muss mindestens zweimal vorkommen, damit es bei der Schlagwortanalyse berücksichtigt wird. Das Ziel der Analyse ist, Kriterien zur Synthese der Literatur zu identifizieren (vgl. Schryen 2015, S. 305). Abbildung 3-3 enthält die Schlagwortwolke, die die Information der Häufigkeit eines Schlagwortes visualisiert. Die häufigsten Schlagworte sind *Differential Evolution* und *Self-Adaptation* mit jeweils acht Nennungen, weshalb diese beiden Schlagworte am größten dargestellt werden.



Abbildung 3-3: Schlagwortwolke der Literaturergebnisse aus Kapitel 3.2.1

Quelle: Eigene Darstellung

Die Schlagworte aus Abbildung 3-3 werden durch Begriffe dominiert, die **Verfahren** zur Datenanalyse beschreiben. So gehören der genetische Algorithmus (*Genetic Algorithm*), die Evolutionsstrategien (*Evolution Strategy*) sowie die Differentialevolution (*Differential Evolution*) zu evolutionären Algorithmen (*Evolutionary Algorithms*), die sich die Prinzipien der biologischen Evolution zum Vorbild nehmen, um komplexe **Optimierungsprobleme** (*Optimization*) zu lösen (vgl. Gerdes et al. 2004, S. 1). Der genetische Algorithmus verwendet dafür in seiner ursprünglichen Form nach Holland (1975) eine binäre Kodie-

nung, womit dieser Ansatz insbesondere für kombinatorische Optimierungsprobleme geeignet ist (vgl. Gerdes et al. 2004, S. 2). Im Gegensatz zum genetischen Algorithmus wird im Rahmen der Evolutionsstrategie das Problem direkt zumeist durch reelle Zahlen, und nicht wie beim genetischen Algorithmus durch binäre Werte, repräsentiert (vgl. Weicker 2015, S. 134). Die Differentialevolution (*Differential Evolution*) basiert ebenfalls wie die beiden vorangegangenen Algorithmen auf den Evolutionsprinzipien Selektion, Rekombination und Mutation. Die Differentialevolution arbeitet ebenso mit reell-wertigen Individuen (vgl. Storn und Price 1997, S. 341), es wird jedoch ein zusätzlicher Operator berücksichtigt, der mit Hilfe von Differenzvektoren eine Mischung aus Rekombination und Mutation bewirkt, womit der potentielle Suchraum nach Lösungen für den Algorithmus vergrößert wird (vgl. Weicker 2015, S. 170). Durch diesen Ansatz soll der Suchprozess nach einem potentiellen globalen Optimum verbessert werden (vgl. Storn und Price 1997, S. 342). Neben den Verfahren für Optimierungsprobleme finden sich in der Schlagwortwolke auch Begriffe wieder, die gemäß der Differenzierung für Data-Mining Methoden nach Fayyad et al. (1996, S. 44 f.) **Segmentierungsprobleme** (*Partitional Clustering*) sowie **Klassifikationsprobleme** (*Self-Organizing Map*) adressieren. Bei partitionierenden Clustering-Verfahren werden während des Clustering die Distanzen zwischen den untersuchten Objekten bestimmt (vgl. Backhaus et al. 2018a, S. 438). Mit Hilfe von Self-Organizing Maps können Klassifikationsprobleme effizient gelöst werden (vgl. Kohonen 1990, S. 1464).

Darüber hinaus werden in Abbildung 3-3 unterschiedliche Fokusse (*Parameter Adaptation*, *Strategy Adaptation* sowie *Adaptation Sensorimotor Control*) einer Anpassung (*Self-Adaptation*) dargestellt. In Abhängigkeit der eingesetzten Algorithmen werden mit dem Prinzip der Selbst-Adaption unterschiedliche Ziele verfolgt. Da die Funktionsweise und Effektivität eines Algorithmus stark von der Einstellung der Parameter abhängig ist, wird in einigen Beiträgen das Prinzip der Selbst-Adaption für eine in Abhängigkeit des Anwendungsfalles geeignete Parametrisierung (*Parameter*) des Algorithmus genutzt. Auch während der Lösungsfindung eines Algorithmus wird das Prinzip der Selbst-Adaption genutzt, um im Zuge der Optimierung die Suchstrategie (*Strategy*) anzupassen. Neben dem Ziel, die Selbst-Adaption für die Konfiguration von Algorithmen einzusetzen, wird das Prinzip in zwei Beiträgen zur automatisierten Steuerung von sensorischen und motorischen Leistungen genutzt (*Sensorimotor*). Diese Schlagworte können unter dem Kriterium **Anpassungsfokus** zusammengefasst werden, bei dem die Ziele der Selbst-Adaption differenziert wer-

den. Das Kriterium wird neben den Verfahren zur Auswertung der Rechercheergebnisse verwendet.

Das dritte Kriterium wird auf Basis der Schlagworte *Background Subtraction* sowie *Evolvable Hardware* definiert (vgl. Abbildung 3-3) und soll die Beiträge in Abhängigkeit des **Anwendungsgebietes** konkreter untersuchen. Im Rahmen der Bildverarbeitung wird das Prinzip der Selbst-Adaption für Heuristiken eingesetzt, um den Hintergrund eines Bildes zu extrahieren (*Background Subtraction*), damit die Objekte, die im Vordergrund stehen sollen, hervorgehoben werden. Ein weiteres Anwendungsgebiet wird als *Evolvable Hardware* bezeichnet, bei dem die evolutionären Ansätze aus dem Bereich der Biologie mit den algorithmischen Optimierungsansätzen aus dem Bereich der Informatik kombiniert werden, um spezialisierte Elektronik ohne eine manuelle Steuerung betreiben zu können (vgl. Gordon und Bentley 2002, S. 279 ff.).

Durch die drei Kriterien **Verfahren**, **Anpassungsfokus** sowie **Anwendungsgebiet** lassen sich die häufigsten Schlagworte der identifizierten Literatur (vgl. Abbildung 3-3) einordnen, weshalb diese Kriterien im Folgenden zur Analyse der Rechercheergebnisse verwendet werden.

3.2.2.2 Konzept-zentrierte Synthese der Ergebnisse

In der Synthesephase einer Literaturrecherche werden dem Leser die Ergebnisse präsentiert. Eine Konzept-zentrierte Präsentation wird gegenüber einer chronologischen oder auch Autoren-zentrierten Darstellung für Literaturanalysen in der Wirtschaftsinformatik empfohlen (vgl. Schryen 2015, S. 305). Bei einer Konzept-zentrierten Synthese werden die Rechercheergebnisse anhand eines Kriteriums untereinander differenziert. Da die Menge der Schlagworte aus dem Kriterium **Verfahren** in der Schlagwortwolke (vgl. Abbildung 3-3) dominieren, wird dieses Kriterium als zentrales Konzept zur differenzierten Präsentation der Ergebnisse ausgewählt. Nachgelagert werden die Kriterien **Anpassungsfokus** sowie **Anwendungsgebiet** innerhalb der jeweiligen Verfahren untersucht. Tabelle 3-3 stellt die identifizierten Ergebnisse in Abhängigkeit der unterschiedlichen Verfahren dar. Auf erster Ebene können die Beiträge entsprechend dem Ziel des Verfahrens (Optimierung, Klassifikation, Clustering) unterschieden werden (vgl. Tabelle 3-3). Wenige Beiträge können nicht in diese Kategorien eingeordnet werden, weshalb diese unter „Sonstige“ geführt werden. Trotzdem kann in dieser Gruppe entsprechend des Forschungsschwerpunktes eine Unterscheidung vorgenommen werden: Framework, Studie, Vorgehensmodell.

Tabelle 3-3: Themen-zentrierte Übersicht der Literaturergebnisse

Quelle: Eigene Tabelle

Themenbereiche			Autoren	
Optimierung	populationsbasierte Heuristiken	Evolutionär	Genetic Algorithm (GA)	(Ameduri et al. 2009; Happe et al. 2013; Hoorfar 2007; Keymeulen et al. 1998; Srinivasa et al. 2005)
			Evolution Strategy (ES)	(Bäck 1996; Wong et al. 2014)
			Differential Evolution (DE)	(Kovacevic et al. 2014; Neri und Tirronen 2010; Pan et al. 2011b; Qin et al. 2009; Teo 2006; Zhao et al. 2011; Zhong und Zhang 2012)
		Schwarmintelligenz	Particle Swarm Optimization (PSO)	(Wang et al. 2011)
			Firefly Algorithmus (FF)	(Niknam et al. 2012)
			Wasp-based Multi Agent System (MAS)	(Meyyappan et al. 2007, 2008)
			Gravitational Search Algorithm (GSA)	(Niknam et al. 2013)
		Lehr- und Lerninspiertert	Teaching-Learning-Based Optimization (TLBO)	(Rao et al. 2012)
Klassifikation	Künstliche Neuronale Netze	Überwachtes Lernen	Back Propagation (BP)	(Chen et al. 2009; Ford et al. 2013)
		Unüberwachtes Lernen	Self-Organizing Map (SOM)	(Lee et al. 2011; Ramirez-Quintana und Chacon-Murguia 2015)
Clustering	partitionierende Algorithmen	iterierende Distanzverfahren	Gaussian Mixture Model (GMM)	(Bhaskar et al. 2010; Chen und Ellis 2014; Hao-li et al. 2006; Ji et al. 2006)
Sonstige			Framework	(Filieri et al. 2012; Goldman et al. 2001; Musliner et al. 1999; Pan et al. 2011a; Zhao et al. 2003)
			Studie	(Babae und Babamir 2013; Quitadamo und Zambonelli 2008)
			Vorgehensmodell	(Garcia-Diaz et al. 2012)

Im Folgenden werden die Beiträge entsprechend der Verfahrensziele Optimierung, Klassifikation, Clustering sowie der Kategorie „Sonstige“ Konzept-zentriert vorgestellt.

Optimierung

Im Themenbereich *Optimierung* werden Heuristiken betrachtet, die alle populationsbasiert sind. Bei einer populationsbasierten Optimierung beginnt die Suche ausgehend von unter-

schiedlichen Startpunkten parallel (vgl. Beheshti und Shamsuddin 2013, S. 3) und wird auf mehrere Individuen verteilt (vgl. Beheshti und Shamsuddin 2013, S. 6). In Anlehnung an die biologische Fachterminologie wird bei einem einzelnen Lösungskandidaten von einem Individuum und bei einer Menge von Individuen von einer Population gesprochen (vgl. Weicker 2007, S. 24).

Evolutionäre Algorithmen orientieren sich an den genetischen Operatoren Selektion, Rekombination und Mutation (vgl. Weicker 2015, S. 9 f.). Diese Operatoren werden sowohl vom genetischen Algorithmus (GA, engl. Genetic Algorithm), der Evolutionsstrategie (ES, engl. Evolution Strategy) als auch von der Differentialevolution (DE, engl. Differential Evolution) genutzt.

Das Prinzip der Selbst-Adaption wird im Rahmen des *genetischen Algorithmus* fünfmal genannt (vgl. Tabelle 3-3), wobei der Beitrag von Ameduri (2009) das Prinzip der Selbst-Adaption nicht berücksichtigt, sondern einmalig im Abstract zur Definition von Smart Materials verwendet (vgl. Ameduri et al. 2009, S. 1529). Der genetische Algorithmus (GA) basiert auf der Arbeit von Holland (1975) und versucht mit einer Menge von Individuen, die gemeinsam eine Population bilden, eine Lösung für ein Optimierungsproblem zu finden. Das Optimierungsverfahren repräsentiert ein Problem binär, beruht auf dem evolutionären Prinzip „Survival of the fittest“ und nutzt die genetischen Operatoren (vgl. Holland 1975). Srinivasa et al. (2005) setzen den GA für eine dynamische Generierung eines Benutzerprofils auf Basis von persönlichen Interessen ein, damit Empfehlungen aus einem kontinuierlichen Strom von XML Dokumenten erstellt werden (vgl. Srinivasa et al. 2005, S. 562). Das Erstellen eines Benutzerprofils in Abhängigkeit der Interessen wird als selbst-adaptiv deklariert (vgl. Srinivasa et al. 2005, S. 536), in dem Beitrag wird jedoch nicht erläutert, wie das Prinzip der Selbst-Adaption in dem zugrundeliegenden genetischen Algorithmus eingesetzt wird. Happe et al. (2013) nutzen eine selbst-adaptive Strategie zur Verteilung von Ressourcen bei der Bildverarbeitung mit dem Ziel, eine effiziente Ressourcennutzung zu erzielen. Hierfür werden auf Basis der sich verändernden Objektgrößen die Bearbeitungsdauer mit Hilfe von Verfahren zur Partikelfilterung, einem Algorithmus auf Basis des GA (vgl. Del Moral 1996), geschätzt und in Echtzeit Rechenkapazitäten aktiviert oder deaktiviert (vgl. Happe et al. 2013, S. 95). Hoorfar (2007) setzt das Prinzip der Selbst-Adaption für einen dynamischen Mutationsoperator ein. Mit dem genetischen Algorithmus werden Probleme aus dem Bereich der Elektrotechnik optimiert, bei denen optimale Frequenzbereiche für frequenzselektive Techniken wie Antennen- oder Mikrowellen gesucht

werden (vgl. Hoorfar 2007, S. 528). Keymeulen et al. (1998) nutzen den genetischen Algorithmus zur Navigation von Robotern im Labor und versuchen durch das Prinzip der Selbst-Adaption eine autonome Navigation für die Evolvable Hardware, bei der es sich um Elektronik handelt, die eigene Verhaltensmuster entwickeln kann, einzusetzen (vgl. Keymeulen et al. 1998, S. 359). Ein selbst-adaptiver genetischer Algorithmus kommt in verschiedenen Forschungsdomänen zum Einsatz. In der Informatik zur Bildbearbeitung (vgl. Happe et al. 2013) sowie zur Analyse von XML Dokumenten (vgl. Srinivasa et al. 2005), in der Elektrotechnik zur Steuerung von Frequenzbereichen (vgl. Hoorfar 2007) sowie im Bereich der Robotic bzw. Evolvable Hardware (vgl. Keymeulen et al. 1998), der aufgrund der Kombination von Informatik und Elektronik als interdisziplinär bezeichnet werden kann.

Die *Evolutionstrategie* (ES) basiert ebenfalls wie der GA auf Individuen in einer Population sowie den genetischen Operatoren (vgl. Rechenberg und Eigen 1976). Das Problem wird im Vergleich zum GA jedoch durch reelle Zahlen repräsentiert und der Fokus liegt auf der Mutation als maßgebender Suchoperator für eine optimale Lösung (vgl. Rechenberg und Eigen 1976). Bäck (1996) betrachtet eine selbst-adaptive Strategie zur Mutation, bei der mit Hilfe von Lernraten während des Optimierungsprozesses die Mutation bewertet und beeinflusst werden kann (vgl. Bäck 1996, S. 12). Wong et al. (2014) stellt im Vergleich zu Bäck eine selbst-adaptive Populationsgröße vor, die sich während des Optimierungsprozesses an der besten Fitness orientiert (vgl. Wong et al. 2014, S. 652), um in Echtzeit eine Produktionsplanung auf Basis von RFID Sensordaten zu optimieren (vgl. Wong et al. 2014, S. 647).

Die *Differential evolution* (DE) arbeitet ähnlich wie die Evolutionstrategie, jedoch wird die Mutation erweitert und es findet eine Kombination aus Mutation sowie Rekombination statt. Dabei werden Vektoren zwischen allen Individuen innerhalb einer Population als Grundlage für die Mutation eines Individuums genommen, womit die Vielfalt einer potentiellen Lösung erweitert wird (vgl. Storn und Price 1997, S. 343 f.). Bei sechs der insgesamt sieben ermittelten Beiträge zur Differential evolution handelt es sich jeweils um einen Methodenbeitrag, der versucht, den Steuerungsparameter zur Generierung der Vektoren zu optimieren (vgl. Kovacevic et al. 2014; Neri und Tirronen 2010; Pan et al. 2011b; Qin et al. 2009; Teo 2006; Zhao et al. 2011). Der Gedanke der Selbst-Adaption wird genutzt, indem auf Grundlage der vorangegangenen Phasen die Fitness evaluiert wird und durch den Parameter zukünftige Mutationen zielgerichtet gesteuert werden (vgl. Neri und Tirronen

2010; Pan et al. 2011b; Qin et al. 2009; Teo 2006; Zhao et al. 2011). Kovacevic et al. (2014) nutzen für dieses Ziel mit der variablen Nachbarschaftssuche (VNS, engl. variable neighborhood search) einen zusätzlichen Ansatz, um bei den diversen Mutationsmöglichkeiten eine Lösung zu finden, die nicht in einem lokalen Optimum stagniert (vgl. Kovacevic et al. 2014, S. 157). In dem Beitrag von Zhong und Zhang (2012) wird die Differentialevolution mit einer selbst-adaptiven Parametersteuerung eingesetzt, um im Bereich der Bildverarbeitung das Subpixel-Problem, das bspw. bei der Gewinnung von Informationen aus großflächigen Satellitenaufnahmen auftritt, zu lösen (vgl. Zhong und Zhang 2012, S. 1306).

Bei *Schwarmintelligenz Algorithmen* agieren Individuen gleicher Art und Größe. Die Einzelindividuen besitzen häufig begrenzte Fähigkeiten und sind untereinander gleichrangig. Im Vergleich zu evolutionären Algorithmen erfolgt eine Optimierung nicht durch die genetischen Operatoren, sondern durch die Kooperation entsteht ein Informationsaustausch zwischen den Individuen, womit in der Gemeinschaft auch komplexere Aufgaben gelöst werden können (vgl. Weicker 2015, S. 177 ff.).

Die *Partikelschwarm Optimierungsmethode* (PSO, engl. Particle Swarm Optimization) betrachtet reell-wertige Probleme, deren Lösungsprozess durch das Nachahmen und Lernen von benachbarten Individuen gekennzeichnet ist (vgl. Kennedy und Eberhart 1995, S. 1943). Wang et al. (2011) nutzen das Prinzip der Selbst-Adaption, um die Effizienz des Algorithmus bei der Lösungssuche zu verbessern, indem mehrere Suchstrategien parallel auf Basis der vorangegangenen Lösungsgüte erstellt werden und im Anschluss in Abhängigkeit einer Wahrscheinlichkeitsverteilung die Vielversprechendste verfolgt wird (vgl. Wang et al. 2011, S. 4515).

Der *Firefly* (FF) Algorithmus orientiert sich am Leuchtverhalten eines Schwarms von Glühwürmchen (vgl. Yang 2009, S. 170 f.). Niknam et al. (2013) modifizieren das ursprüngliche Verfahren, indem ähnlich wie bei der Differentialevolution eine vektorbasierte Mutation stattfindet (vgl. Niknam et al. 2012, S. 638). Die Mutationsstrategie wird iterativ angepasst, damit der Algorithmus effektiv bei Echtzeit-Anwendungen arbeitet und lokale Optima verhindert werden (vgl. Niknam et al. 2012, S. 639).

Ein *Wespen-basiertes Multi Agenten System* (MAS, engl. Multi Agent System) orientiert sich am Verhalten einer Wespenkolonie, in der es unterschiedliche Aufgaben gibt (vgl. Van Dyke Parunak 1997, S. 79). Das Schwarmverhalten besteht aus der Wespeninteraktion untereinander sowie der Interaktion mit der Umwelt, jeweils in Abhängigkeit der Aufgaben

(vgl. Meyyappan et al. 2007, S. 5160). Durch Grenzwerte kann bestimmt werden, ob eine Wespe ihrer Aufgabe nachkommt und wie stark der Einsatz zur Aufgabenerfüllung ist (vgl. Meyyappan et al. 2007, S. 5160). Dieser Grenzwert sowie die Anzahl an Wespen in einer Kolonie werden in dem Beitrag von Meyyappan et al. (2007) so adaptiert, dass in Abhängigkeit eines Produktionsaufkommens für eine Vielzahl an Produktionsmaschinen ein optimaler Belegungsplan in Echtzeit erstellt wird (vgl. Meyyappan et al. 2007, S. 5157). In einem weiteren Beitrag verbessert Meyyappan et al. (2008) die Verhandlungsstrategie der Wespen zur Ermittlung einer optimalen Maschinenbelegung (vgl. Meyyappan et al. 2008, S. 259).

Der *Gravitationssuchalgorithmus* (GSA, engl. Gravitational Search Algorithm) basiert auf dem Prinzip der Gravitationskraft im Universum, bei dem sich Teilchen in Abhängigkeit ihrer Masse sowie Entfernung anziehen (vgl. Rashedi et al. 2009, S. 2232). Niknam et al. (2013) entwickeln diesen Ansatz weiter, indem zwei Parameter zur Steuerung der Gravitationskraft sowie der Anzahl an Teilchen bzw. Agenten dynamisch je Iteration angepasst werden, um lokale Optima bei der Stromversorgungs- und Spannungssteuerung zu vermeiden (vgl. Niknam et al. 2013, S. 742 ff.).

Die *lehr- und lerninspirierten Verfahren* basieren auf der Wirkung des Einflusses eines Lehrers auf die Leistung der Lernenden. Eine Population an Lernenden versucht, iterativ zu lernen und eine Lösung für ein Problem zu finden. Im Rahmen der Problemlösung kann das Lernen durch den Lehrer erfolgen oder durch Interaktion unter den Lernenden. Der Lehrer wird als eine hochgelehrte Person angesehen, die ihr Wissen mit den Lernenden teilt, weshalb je Iterationsschritt das bisherige Optimum durch den Lehrer repräsentiert wird (vgl. Rao et al. 2011, S. 304). Der *Teaching-Learning-Based Optimization* (TLBO) Algorithmus wird ebenfalls durch einen zusätzlichen Parameter gesteuert, indem Strafen für schlechte Lösungen eingeführt werden, womit über die gesamte Laufzeit eine Selbstanpassung erzielt werden soll (vgl. Rao et al. 2011, S. 1456).

Klassifikation

Mit dem Ziel der *Klassifikation* werden *Künstliche Neuronale Netze* (KNN) in den Rechercheergebnissen eingesetzt. Grundlegend kann die Arbeitsweise eines KNN nach der Art des Lernens unterschieden werden. Beim *überwachten Lernen* wird das Netz mit einem Datensatz, in dem die Ausprägung der Zielvariable bekannt ist, trainiert (vgl. Ertel 2016, S. 301). Beim *unüberwachten Lernen* erfolgt das Training nicht anhand der Überprüfung der Zielvariable, da deren Ausprägung i. d. R. nicht bekannt ist (vgl. Görz et al. 2014, S. 380).

Stattdessen versucht das KNN die unbekanntesten Muster und Strukturen in den Daten zu erkennen (vgl. Görz et al. 2014, S. 380). Bei der *Back-Propagation* handelt es sich um eine Methode des überwachten Lernens (vgl. Tabelle 3-3), bei der die Ausgabe des KNN mit dem bekannten Zielwert verglichen wird und die Abweichung als Fehler zurück an die Eingabeschicht des KNN propagiert wird (vgl. Rumelhart et al. 1986, S. 534 f.). Chen et al. (2009) nutzen dieses Verfahren für eine selbst-adaptive Steuerung von Schweißrobotern, die je Schweißobjekt via Bildverarbeitung die Schweißintensität sowie -position bestimmen (vgl. Chen et al. 2009, S. 576). Ford et al. (2013) nutzen das Prinzip, um verdächtige Bieter bei online Auktionen zu identifizieren. Die Selbst-Adaption sorgt dafür, dass die Klassifikationsregeln in Abhängigkeit neuer Marktbedingungen sowie anderer verdächtiger Verhaltensweisen angepasst werden (vgl. Ford et al. 2013, S. 657). *Selbstorganisierende Karten* (SOM, engl. Self-Organizing Map) gehören zu der Kategorie des unüberwachten Lernens (vgl. Tabelle 3-3) und werden primär zur Mustererkennung eingesetzt (vgl. Kohonen 1990, S. 1467). Ähnlich wie Ford et al. (2013) setzen Lee et al. (2011) die selbstorganisierenden Karten ein, um selbst-adaptiv auf sich verändernde Umweltbedingungen zu reagieren und durch Mustererkennung gefährliche Anomalien in Softwareanwendungen in Echtzeit zu identifizieren (vgl. Lee et al. 2011, S. 14891). Das Anwendungsgebiet von Ramirez-Quintana und Chacon-Murguia (2015) ist die Bildverarbeitung, bei der mittels SOM versucht wird, Objekte innerhalb eines Bildes dynamisch zu erkennen und vom Hintergrund zu trennen (vgl. Ramirez-Quintana und Chacon-Murguia 2015, S. 1137).

Clustering

Zum Clustering werden *partitionierende* Verfahren eingesetzt, die während des Clustering die *Distanzen* zwischen den untersuchten Objekten bestimmen (vgl. Backhaus et al. 2018a, S. 438). Ein *Gaussian Mixture Model* (GMM) arbeitet nach dem Gaußschen Ansatz und versucht anhand von Wahrscheinlichkeitsverteilungen iterativ das Clustering zu verbessern (vgl. Xu und Jordan 1996, S. 129). Das Einsatzgebiet dieses Algorithmus liegt primär in der Bildverarbeitung zur Identifikation von Objekten (vgl. Bhaskar et al. 2010; Chen und Ellis 2014; Hao-li et al. 2006; Ji et al. 2006), um bspw. den Verkehr zu überwachen (Ji et al. 2006) oder Fußgänger zu erkennen (vgl. Hao-li et al. 2006). Alle Beiträge setzen die Selbst-Adaption dafür ein, dass Schwellwerte zur Identifikation sowie Segmentierung von sich bewegenden Objekten in Abhängigkeit des Bildhintergrundes bestmöglich gesetzt werden (vgl. Bhaskar et al. 2010, S. 1133; Chen und Ellis 2014, S. 35; Hao-li et al. 2006, S. 880; Ji et al. 2006, S. 647).

Sonstige

In der Kategorie Sonstige werden Beiträge zusammengefasst, die keine Methode fokussieren, sondern ein *Framework* vorstellen, die Ergebnisse einer *Studie* präsentieren oder ein *Vorgehensmodell* entwickeln.

Filieri et al. (2012) entwickeln ein *Framework* für selbst-adaptive Software, bei der auch nicht-funktionale Anforderungen kontinuierlich berücksichtigt werden (vgl. Filieri et al. 2012, S. 164 f.). Goldman et al. (2001) sowie Musliner et al. (1999) stellen eine Architektur zur Planung und Zuweisung von Hardware-Ressourcen für Softwareanwendungen vor, die selbst-adaptiv in Abhängigkeit des Angebotes sowie der Nachfrage an Ressourcen erfolgen soll (vgl. Goldman et al. 2001, S. 6; Musliner et al. 1999, S. 23). Ein *Framework* zur Verarbeitung von Animationen für Zeichensprache auf unterschiedlichen Plattformen wird von Pan et al. (2011a) vorgestellt, damit eine selbst-adaptive Wiedergabe von Videos in Abhängigkeit der Bildschirmgröße sowie der Auflösung ermöglicht wird (vgl. Pan et al. 2011a, S. 213).

Babae und Babamir (2013) präsentieren eine *Studie* zu zentralen Aspekten Service-orientierter Systeme, die sie im Rahmen einer Literaturrecherche identifizieren (vgl. Babae und Babamir 2013, S. 214). Im Ergebnis sollten die Systeme zur Laufzeit selbst-adaptiv auf neue Anforderungen von Stakeholdern, geänderte Geschäftsziele oder unerwartete Marktsituationen reagieren können, um einen zuverlässigen und fehlertoleranten Service zu bieten (vgl. Babae und Babamir 2013, S. 259 f.). Quitadamo und Zambonelli (2008) forschen an derselben Problemstellung zur effizienten Ressourcenzuweisung wie Goldman et al. (2001) und Musliner et al. (1999), konzentrieren sich in ihrer Studie jedoch auf die autonome Kommunikation innerhalb einer Systemlandschaft (vgl. Quitadamo und Zambonelli 2008, S. 457). Zhao et al. (2003) stellen ein *Framework* vor, durch das eine selbst-adaptive Synchronisation von Uhren im Rahmen von Grid-Computing, das durch eine zeitlich asynchrone Intranet- und Internet-Umgebung geprägt ist, ermöglicht wird (vgl. Zhao et al. 2003, S. 434).

Garcia-Diaz et al. (2012) entwickeln ein *Vorgehensmodell* für eine kontinuierliche Integration im Rahmen von Softwareentwicklungsprojekten, die aus mehreren Anwendungskomponenten bestehen (vgl. Garcia-Diaz et al. 2012, S. 479). Durch dieses *Vorgehensmodell* soll selbst-adaptiv auf sich verändernde Rahmenbedingungen reagiert werden können, die sich durch den Kunden, Feedbackgespräche oder auch Benutzertests ergeben (vgl. Garcia-Diaz et al. 2012, S. 480).

3.2.2.3 Schlussbetrachtung des aktuellen Forschungsstandes

In den Ergebnissen der Literaturrecherche sind viele Algorithmen vertreten, die den Heuristiken zuzuordnen sind (vgl. Tabelle 3-3). Dabei handelt es sich um ältere, etablierte Ansätze wie den genetischen Algorithmus (vgl. Holland 1975), aber auch aktuellere Ansätze wie den Gravitationssuchalgorithmus, der in der letzten Dekade entwickelt wurde (vgl. Rashedi et al. 2009). Gemein ist allen, dass Steuerungsparameter für eine effiziente Arbeitsweise des Algorithmus notwendig sind (vgl. Borkar und Sarode 2018, S. 170). Im Fall des GA sind die zu kalibrierenden Parameter die Populationsgröße, die Rekombinations- sowie die Mutationsrate (vgl. Kapitel 3.2.2.2). Die adäquate Kalibrierung der Steuerungsparameter ist ein entscheidender Schritt im Rahmen einer heuristischen Optimierung (vgl. Borkar und Sarode 2018, S. 170) zur Vorbeugung von lokalen Optima (vgl. Weicker 2015, S. 53). Um dieses Problem zu regulieren, wird das Prinzip der Selbst-Adaption eingesetzt, durch das in Abhängigkeit eines Optimierungsproblems sowie der iterativen Lösungsgüte eine Heuristik effizient arbeiten soll (vgl. Kapitel 3.2.2.2). Der Fokus der identifizierten Forschungsarbeiten liegt demnach auf der Kalibrierung Algorithmen-spezifischer Parameter in Abhängigkeit der Umweltbedingungen.

Das Prinzip der Selbst-Adaption wird in den Ergebnissen jedoch nicht zur Anpassung von betriebswirtschaftlichen Modellen eingesetzt. Beim Einsatz von quantitativen Modellen in einem Entscheidungsunterstützungssystem sind diese zumeist einer dynamischen Unternehmensumwelt ausgesetzt (vgl. Kapitel 2.3). Aufgrund dessen sind neben parametrischen auch strukturelle Anpassungen notwendig, bei denen die Wirkungen innerhalb eines Modells neu bestimmt werden (vgl. Kapitel 2.3). Der selbst-adaptive Ansatz findet in der Literatur jedoch keine Anwendung für strukturelle Anpassungen (vgl. Kapitel 3.2.2.2). Darüber hinaus sind die Hauptanwendungsbereiche die Bildverarbeitung und die effiziente Ressourcenverteilung bei der Anwendung von Software (vgl. Kapitel 3.2.2.2). Eine Untersuchung im Bereich der Entscheidungsunterstützungssysteme und konkret für modellgetriebene Systeme hat gemäß den Ergebnissen der Literaturanalyse nicht stattgefunden (vgl. Kapitel 3.2.2.2).

Die Selbst-Adaption wird in dem vorliegenden Forschungsansatz für einen neuen Zweck genutzt, indem neben der Anpassung von Parametern auch strukturelle Änderungen in Abhängigkeit der Umwelt vorgenommen werden. Dieser neue Anpassungsfokus hängt von der Anwendungsdomäne der modellgetriebenen Entscheidungsunterstützungssysteme ab, in der das Prinzip der Selbst-Adaption zur Anpassung von Modellen noch nicht untersucht

wurde (vgl. Kapitel 3.2.2.2). Bei der Anpassung der Wirkungsstruktur von Modellen ist insbesondere die Vielfalt der Komplexität ein besonderes Merkmal (vgl. Kapitel 2.3). Bei der Kalibrierung von Parametern steht die Vielzahl der Komplexität im Vordergrund, weshalb sich der Anpassungsfokus dieses Forschungsansatzes von den Ergebnissen der Literaturrecherche unterscheidet. Zusammenfassend lässt sich festhalten, dass der selbst-adaptive Ansatz bereits in anderen Forschungsbereichen existiert, bspw. zur Bildverarbeitung oder Produktionssteuerung, jedoch keine Anwendung zur Anpassung von Modellen im Bereich der Entscheidungsunterstützungssysteme gefunden hat. Forschungsmethodisch kann die vorliegende Arbeit deshalb als „Exaptation“ bezeichnet werden, bei der bekannte Lösungen für Probleme in anderen Domänen eingesetzt werden (vgl. Gregor und Hevner 2013, S. 347). In der Wirtschaftsinformatik ist diese Vorgehensweise aufgrund des häufigen technischen Fortschrittes weit verbreitet (vgl. Gregor und Hevner 2013, S. 347). Die Herausforderung im Vergleich zu den bestehenden Ansätzen der Selbst-Adaption besteht darin, wie quantitative Entscheidungsunterstützungsmodelle angepasst werden können und wie neben den parametrischen Änderungen im Speziellen strukturelle Veränderungen vorgenommen werden können.

3.3 Problemrelevanz im praktischen Einsatz von M-EUS

Im vorangegangenen Kapitel ist mit dem Forschungsstand die Rigorosität dieses Forschungsvorhabens durch die Literaturrecherche aufgezeigt worden. Entsprechend dem Design Science Ansatz nach Hevner et al. (vgl. Kapitel 1.2) wird in diesem Kapitel die praktische Relevanz untersucht, damit gezielt Probleme von Unternehmen durch das avisierte Artefakt adressiert werden können (vgl. Hevner et al. 2004, S. 80). Hierfür wird eine Umfrage durchgeführt, deren Untersuchungsdesign in Kapitel 3.3.1 vorgestellt wird und die Ergebnisse in Kapitel 3.3.2 herausgearbeitet werden.

3.3.1 Untersuchungsdesign und Vorgehensweise

Die potentiellen Probleme, die beim Einsatz von modellgetriebenen Entscheidungsunterstützungssystemen auftreten können (vgl. Kapitel 2.3), werden im Rahmen einer Umfrage evaluiert. Hierfür sollen Aussagen über Probleme im Umgang mit modellgetriebenen EUS von Anwendern aus der Praxis beurteilt werden. Auf Grundlage dessen soll die Relevanz und der Bedarf der Forschung beurteilt werden.

In einem ersten Schritt wird der Nutzen von EUS in der Praxis erfragt. Um die mit der Nutzung von EUS verbundenen Risiken und Probleme genauer zu untersuchen, sollten die Teilnehmer in einem zweiten Schritt verschiedene Probleme bewerten, die während der Nutzung auftreten können. Die jeweiligen Probleme in der Umfrage basieren auf den sechs Eigenschaften der Selbst-Adaption (vgl. Abbildung 3-1). Diese Eigenschaften sind, wie in Kapitel 3.1 erläutert, in der Literatur definiert worden, um Veränderungen im Kontext und in der Umgebung eines Systems zu erkennen und darauf reagieren zu können. Auf Basis dieser Eigenschaften werden Probleme im Umgang mit EUS formuliert. Die Antworten repräsentieren einen Indikator für die Problemsituation von EUS in einer dynamischen Unternehmensumwelt, womit die Relevanz des Forschungsansatzes für die unternehmerische Praxis aufgezeigt werden soll (vgl. Hevner et al. 2004, S. 80). Die zu bewertenden Aussagen sind der Tabelle 3-4 zu entnehmen.

Tabelle 3-4: Zu bewertende Aussagen basierend auf den sechs Self-* Eigenschaften

Quelle: Eigene Tabelle

Eigen-schaft	Variablen-name	Aussage
Self-Awareness	PL1_PLm1	Das EUS kann die eigenen Aktionen und Maßnahmen (bspw. empfohlene Entscheidungen) nicht selbst beurteilen.
Context-Awareness	PL1_PLm2	Das EUS ist sich nicht über das operative Umfeld, in dem es agiert (bspw. Kunden oder Lieferanten), bewusst.
	PL1_PLm3	Das EUS kennt nicht die externen Systeme (bspw. ERP-Systeme), mit denen es interagiert.
Self-Configuring	PL1_PLm4	Das EUS muss oftmals von externen Akteuren (z. B. Entwickler) angepasst und gewartet werden.
	PL1_PLm5	Der Output des EUS ist nach Veränderungen in der Umwelt (z. B. Ölpreisänderung) oder nach veränderten Rahmenbedingungen (z. B. neue Zulieferer) nicht mehr gültig.
Self-Healing	PL1_PLm6	Das EUS kann Störungen oder Probleme nicht erkennen bzw. diagnostizieren, sodass eine Reaktion darauf nicht möglich ist.
	PL1_PLm7	Das EUS kann Störungen oder Probleme nicht prognostizieren und nicht proaktiv darauf hinweisen oder reagieren.
Self-Optimizing	PL1_PLm8	Das EUS stellt sich nicht auf die Bedürfnisse unterschiedlicher Endanwender ein.
	PL1_PLm9	Das EUS kann die eigene Konfiguration nicht selbst optimieren.
Self-Protecting	PL1_PLm10	Das EUS kann sich nicht eigenständig vor Fehlern schützen.
	PL1_PLm11	Das EUS erkennt keine drohenden Risiken und kann sich somit auch nicht davor schützen.

Zur Erläuterung der Aussagenkonstruktion wird im Folgenden die Eigenschaft *Self-Configuring* betrachtet, durch die sich ein System automatisiert anpassen kann. Die Anpassung geschieht in Abhängigkeit übergeordneter Vorgaben autonom (vgl. Kapitel 3.1). Basierend auf dieser Funktionsbeschreibung ist das Problem „Das EUS muss oftmals von externen Akteuren (z. B. Entwickler) angepasst und gewartet werden“ formuliert worden. Äquivalent sind für alle Eigenschaften aus der Hierarchie der *Self*-* Eigenschaften (vgl. Abbildung 3-1) jeweils ein bis zwei Probleme im Umgang mit einem EUS formuliert worden. Die Antwortskala orientiert sich an einer fünfstufigen Likert-Skala und reicht von 1 (Trifft nicht zu) bis 5 (Trifft zu) (vgl. Schnell 2005, S. 187 f.). Der entwickelte Fragebogen¹ ist webbasiert, um eine hohe Erreichbarkeit und eine schnelle Durchführung gewährleisten zu können (vgl. Jacob et al. 2013, S. 110). Der Fragebogen wurde über die Webanwendung *LimeSurvey* (vgl. Engard 2009, S. 272) erstellt und war vom 06. Februar 2017 bis 05. März 2017 aktiv. Es sind 44 Unternehmen über private Netzwerke kontaktiert worden. Die akquirierten Unternehmen variieren hinsichtlich der Branche sowie der Unternehmensgröße. Unter anderem sind 4 Unternehmen im DAX kontaktiert worden, die detaillierte Beschreibung der Stichprobe folgt in Kapitel 3.3.2. Da alle Antworten anonymisiert wurden, lassen sich keine Rückschlüsse auf konkrete Unternehmen oder Personen ziehen.

3.3.2 Ergebnisse der Umfrage

An der Umfrage haben 50 Personen teilgenommen. In der Stichprobe sind 16% kleine (< 50 Mitarbeiter), 23% mittlere (50-249), 18% große (250-999) und 16% sehr große (1000-5000) Unternehmen sowie 27% mit mehr als 5000 Beschäftigten. Das verarbeitende Gewerbe sowie die Logistik sind mit 37% bzw. 17% die am stärksten vertretenen Branchen. Es folgen die Bereiche "Wasserversorgung und Abfallentsorgung", das Baugewerbe (je 8%), die Energieversorgung und die öffentliche Verwaltung (je 5%). 36 von 50 (72%) Teilnehmern kennen EUS und setzen diese Art System in ihrem Unternehmen ein, während 11 Teilnehmer (22%) EUS kennen, diese aber nicht im operativen Geschäft nutzen. Drei Teilnehmer kennen EUS nicht, weshalb diese Antworten bei der Auswertung nicht berücksichtigt wurden. Insgesamt konnte durch die webbasierte Umfrage eine breite und heterogene Stichprobe von EUS-Nutzern identifiziert werden.

In einem ersten Schritt wird der Nutzen eines EUS im operativen Betrieb erfragt. Durch den Einsatz von EUS steigern mehr als zwei Drittel der Teilnehmer ihre Effizienz, daher

¹ Der vollständige Fragebogen ist dem Anhang I zu entnehmen.

empfehlen 75% den Einsatz eines EUS (vgl. Anhang II). Das Risiko, Entscheidungen auf der Grundlage falscher Informationen aus einem EUS zu treffen, nehmen etwa 20% der Teilnehmer wahr, jedoch können 45% das Risiko nicht ausreichend einschätzen (vgl. Anhang III). Um die mit der Nutzung von EUS verbundenen Risiken und Probleme genauer zu untersuchen, sollten in einem zweiten Schritt die Teilnehmer der Umfrage die Probleme aus Tabelle 3-4, die während der Nutzung auftreten können, bewerten. Die Auswertung erfolgte deskriptiv mit Hilfe von Boxplots und wird im Folgenden dargelegt. Sofern mehrere Fragen auf eine *Self*-* Eigenschaft zurückzuführen sind (vgl. Tabelle 3-4), wird der Durchschnitt aus den Antworten gebildet.



Abbildung 3-4: Boxplot der Probleme mit EUS in einer dynamischen Unternehmensumwelt

Quelle: Eigene Darstellung

Im Boxplot von Abbildung 3-4 fällt auf, dass insbesondere die fehlenden Eigenschaften der *Self-Awareness* (Median von 5), *Self-Healing*, *Self-Optimizing* und *Self-Protecting* (jeweils ein Median von 4) als problematisch eingeschätzt werden. Die entsprechenden Probleme sind: eine fehlende Verifikation der Systemkorrektheit (*Self-Healing*), ein mangelnder Schutz vor Fehlern (*Self-Protecting*), die fehlende Möglichkeit zur Optimierung der Konfiguration im Falle eines Fehlers (*Self-Optimization*) und die Einschränkung der Evaluation der eigenen Aktivitäten im Falle einer Modellveränderung (*Self-Awareness*). Das Problem, dass ein EUS seine Umgebung nicht kennt, wie z. B. die operativen Systeme, mit denen es interagiert (*Context-Awareness*), wird mit einem Median von 3,5 schwach bestätigt. Ein Data Warehouse (DWH) Metadatenmanagement könnte die Informationen über die Betriebsumgebung des EUS liefern, da 15% der Teilnehmer bestätigen, dass die Daten für das EUS aus einem DWH stammen. Das Problem, dass die Vorschläge eines EUS nach einer Veränderung in der Unternehmensumwelt (z. B. neuer Lieferant) nicht mehr gültig sind,

repräsentiert ebenfalls die Eigenschaft *Self-Configuring* (vgl. Kapitel 3.3.1). Dieses Problem wird mit einem Median von 3 als nicht gravierend eingeschätzt. Möglicherweise treten die beschriebenen Änderungen selten auf oder es wird manuell schnell auf Veränderungen reagiert. Diese Vermutung liegt nahe, da in 87% der Antworten das EUS regelmäßig gewartet wird (vgl. Anhang IV).

Fünf der sechs Probleme von Entscheidungsunterstützungssystemen in einer dynamischen Unternehmensumwelt werden von den Anwendern in der Praxis als zutreffend eingestuft. Das abgeleitete Problem auf Basis der Eigenschaft *Self-Configuring* wird als neutral beurteilt. Somit kann festgehalten werden, dass die 50 befragten Anwender aus der unternehmerischen Praxis im Durchschnitt die Probleme von Entscheidungsunterstützungssystemen in einer sich verändernden Umwelt sowie einem dynamischen Kontext und somit auch die Relevanz dieser Forschungsarbeit bestätigen.

3.4 Forschungslücke und Zieldefinition

Das Prinzip der Selbst-Adaption ist ein Forschungsansatz, mit dem eine steigende Dynamik in der Umwelt geregelt werden kann. Das Potential des Prinzips ist in Kapitel 3.1 erläutert worden. Die Untersuchung des aktuellen Forschungsstandes ergab, dass die Selbst-Adaption hauptsächlich für die Kalibrierung von Parametern für eine effiziente Arbeitsweise von Heuristiken in Abhängigkeit sich verändernder Umweltbedingungen eingesetzt wird (vgl. Kapitel 3.2.2.3). Für diese Algorithmen gilt, dass die Güte der Lösung stark von den Funktionsparametern abhängt, weshalb der selbst-adaptive Ansatz in diesem Forschungsbereich weit verbreitet ist und die Effizienz steigert. Aus den Ergebnissen der Literaturrecherche geht hervor, dass das Prinzip der Selbst-Adaption jedoch nicht zur Anpassung von betriebswirtschaftlichen Modellen eingesetzt wird und im Speziellen keine strukturellen Anpassungen (vgl. Kapitel 2.3) von Entscheidungsmodellen vorgenommen werden. Die Anwendung des selbst-adaptiven Ansatzes im Bereich der Entscheidungsunterstützungssysteme stellt somit eine Forschungslücke dar. Es gilt dabei zu untersuchen, welche Komponenten notwendig sind, damit das Prinzip der Selbst-Adaption genutzt werden kann und eine Anpassung von Modellen zur Entscheidungsunterstützung an eine sich stetig verändernde Unternehmensumwelt ermöglicht wird. Darüber hinaus muss für quantitative Entscheidungsunterstützungsmodelle geprüft werden, wie neben parametrischen Änderungen strukturelle Anpassungen an der Wirkungsstruktur eines Modelles vorgenommen werden können. Das Ziel dieser Arbeit ist, dass Modelle kontinuierlich so realitätsnah wie

möglich innerhalb eines Entscheidungsunterstützungssystems repräsentiert werden, damit ein Entscheider stetig bestmöglich unterstützt werden kann.

Die praktische Relevanz dieses Forschungsvorhabens ist in einer Umfrage mit 50 Teilnehmern verifiziert worden. Die Teilnehmer bestätigten, dass es zu Problemen beim Einsatz von Entscheidungsunterstützungssystemen in einer sich verändernden Unternehmensumgebung kommen kann. Konkret wird das Erkennen und Reagieren auf Veränderungen im Kontext sowie in der Umgebung eines Entscheidungsunterstützungssystems als problematisch eingestuft, womit die praktische Relevanz des Forschungsvorhabens deutlich wird.

Zusammenfassend konnte in Kapitel 3 die geforderte Rigorosität sowie Relevanz des Forschungsvorhabens aufgezeigt werden, weshalb in den folgenden Kapiteln die Entwicklung des Artefaktes gemäß des Design Science Ansatzes erfolgt (vgl. Kapitel 1.2).

TEIL II

Entwicklung und Evaluation eines Konzeptes
für selbst-adaptive modellgetriebene
Entscheidungsunterstützungssysteme

4 Entwicklung eines Konzeptes für selbst-adaptive modellgetriebene Entscheidungsunterstützungssysteme

Im Folgenden wird eine Architektur für ein selbst-adaptives modellgetriebenes Entscheidungsunterstützungssystem vorgestellt. Hierfür werden zur Erfüllung der Rigorosität (vgl. Kapitel 1.2) die relevanten Anforderungen aus der Wissenschaft erhoben (vgl. Kapitel 4.1), um darauf aufbauend die Architektur zur Lösung der initialen Probleme (vgl. Kapitel 1.1) zu entwickeln und vorzustellen (vgl. Kapitel 4.2).

4.1 Anforderungen an das Konzept

Das Ziel dieser Arbeit ist, dass Modelle kontinuierlich so realitätsnah wie möglich innerhalb eines Entscheidungsunterstützungssystems repräsentiert werden, damit ein Entscheider stetig bestmöglich unterstützt werden kann.

Zur Erfüllung des Ziels mit Hilfe des Lösungsansatzes der Selbst-Adaption (vgl. Kapitel 3.4) ist es zunächst notwendig, Anforderungen zu definieren. Diese basieren auf den Erkenntnissen aus dem Forschungsbereich der Entscheidungsunterstützungssysteme sowie dem Bereich für selbst-adaptive Systeme und definieren das zu entwickelnde Artefakt. Bei einem Artefakt kann es sich bspw. um eine Methode, ein Modell oder ein System handeln (vgl. Hevner et al. 2004, S. 82). Bei letzterem werden in der Regel IT-Systeme auf unterschiedlichen Granularitätsebenen beschrieben und konkret Prozesse, Strukturen sowie Interaktionen definiert (vgl. Offermann et al. 2010, S. 84). In der Literatur wird die Entwicklung eines Systems am häufigsten fokussiert (vgl. Offermann et al. 2010, S. 83) und in Hinblick auf ein avisiertes selbst-adaptives Entscheidungsunterstützungssystem auch in dem vorliegenden Forschungsansatz verfolgt. Entsprechend wird im Folgenden der Begriff „Artefakt“ als Synonym für das zu entwickelnde selbst-adaptive EUS verstanden.

Aus dem Bereich der EUS sind die wichtigsten Funktionalitäten im Umgang mit einem *Modell* die Was-wäre-wenn Analysen (engl., what-if analysis), das Erstellen sowie Verwalten von unterschiedlichen Szenarien sowie die Zielwertsuche (engl., goal-seeking) (vgl. Power 2013, S. 48 f.). Für diese *Analysemöglichkeiten* ist jeweils die Interaktion mit dem Anwender notwendig, sodass eine *Benutzerschnittstelle* erforderlich ist. Neben dieser Schnittstelle sieht die Standardarchitektur für ein Entscheidungsunterstützungssystem eine

Daten- sowie Modellbank vor (vgl. Kapitel 2.2.1), die jeweils berücksichtigt werden müssen.

Zur Entwicklung eines *selbst-adaptiven Systems* stellen Lemos et al. (2013) einen Entwurfsraum vor, der die zu treffenden Entscheidungen während des Entwicklungsprozesses aufzeigt und diese in unterschiedliche Dimensionen aufteilt (vgl. Lemos et al. 2013, S. 4). Dabei dienen die fünf Dimensionen *Observation*, *Representation*, *Control*, *Identification* und *Enabling Adaptation* der Strukturierung des Entwurfsraumes (vgl. Lemos et al. 2013, S. 4). In diesem Raum werden zur funktionalen Spezifikation des zu entwickelnden Konzeptes die sechs grundlegenden *Self*-* Eigenschaften und zur prozessualen Spezifikation der *MAPE-K* Zyklus berücksichtigt (vgl. jeweils Kapitel 3.1).

In der Dimension *Observation* des Entwurfsraumes für selbst-adaptive Systeme werden nach Lemos et al. (2013) die zu überwachenden Informationen fokussiert. Die zentralen Fragen sind, welche Daten aus dem zu überwachenden System sowie der externen Umgebung erfasst werden sollen, ob diese verfügbar sind und wie häufig diese aktualisiert werden (vgl. Lemos et al. 2013, S. 5). Wird eine Entscheidung aus allen Handlungsalternativen durch den Anwender ausgewählt und in der Realität umgesetzt, benötigt das Artefakt die Rückmeldung, welche Entscheidung getroffen wurde, damit aus dem dazugehörigen Szenario die Planwerte über die zukünftige Entwicklung extrahiert werden. Dieses Szenario wird im Rahmen der modellgetriebenen Entscheidungsunterstützung durch eine was-wäre-wenn-Funktion simuliert (vgl. Power 2013, S. 48). Somit benötigt das Artefakt die simulierten Planwerte auf Basis des Modells. Gemäß des *MAPE-K* Zyklus wäre es sinnvoll, über einen Aktuator Zugriff auf das zu verwaltende System zu ermöglichen, damit Entscheidungen durch das Artefakt selbstständig durchgeführt und ein Bewusstsein über die eigenen Aktionen sowie das eigene Verhalten (*Self-Awareness*) vermittelt wird (vgl. Kapitel 3.1). Das zu verwaltende System könnte bei der Unterstützung einer Preisgestaltung der Zugriff auf einen Webshop sein, damit das Artefakt eine Preissetzung eigenständig umsetzen kann. Ergänzend dazu müssen vom Artefakt die operativen Systeme sowie ggf. externe Informationsquellen überwacht werden, damit zum einen Istwerte als Vergleichskriterium für die Planwerte und zum anderen ausreichend Informationen über die Unternehmensumwelt (*Context-Awareness*) vorhanden sind. Dieses Monitoring soll über Sensorschnittstellen des zu verwaltenden Systems erfolgen (vgl. Kapitel 3.1). Die Aktualisierungsraten der operativen Systeme und zu verwaltenden Systeme bestimmen die Intervalle zur Überwachung. Im Falle von EUS kann dies in nahezu Echtzeit erfolgen (vgl. Kapitel 2.3). Entspre-

chend dieser Bedingungen ist es sinnvoll, dass sowohl die Entscheidungen, die umgesetzt werden sollen, vom Artefakt ausgehen sowie eine dauerhafte Verbindung zu den operativen Systemen besteht, damit das selbst-adaptive System eine maximale Informationstransparenz und damit ein möglichst hohes Bewusstsein (engl. Awareness) über das eigene Handeln sowie den Kontext hat.

Die Dimension *Representation* betrachtet die Repräsentation von Adaptionszielen, Inputwerten, Wirkungen, Systemindikatoren und -zuständen (vgl. Lemos et al. 2013, S. 4). Die Leitfragen dieser Dimension lauten: Wie werden die erfassten Daten aus der *Observation* repräsentiert und wie können diese im Folgenden für eine eventuelle Anpassung genutzt werden (vgl. Lemos et al. 2013, S. 6). Es sollen die historischen Daten gesammelt werden, um Veränderungen der jeweiligen Variablen über die Zeit zu untersuchen. Bei den beobachteten Werten handelt es sich je betrachteter Variable um eine zeitliche Entwicklung, sodass für entsprechende Analysezwecke die Entwicklung als eine Zeitreihe repräsentiert werden sollte. Die Repräsentation der Daten hat nach dem *MAPE-K* Zyklus einen maßgeblichen Einfluss auf die anwendbaren Methoden in der Analyse-Phase (vgl. Kapitel 3.1). Da Entscheidungsmodelle immer problemorientiert entwickelt werden, können mehrere Modelle für unterschiedliche Entscheidungsprobleme parallel existieren (vgl. Power 2002, S. 158), weshalb das Artefakt auch mehrere Modelle berücksichtigen können sollte. Das Artefakt soll unabhängig von einer konkreten EUS-Anwendung sein und deshalb generisch für mehrere Anwendungen (bspw. IBM Cognos TM1, Quantrix oder Vensim) gelten. Hierfür ist es notwendig, die Modelle extern vorzuhalten. Die Art der Repräsentation muss unterschiedliche Variablentypen sowie eine mathematische Verknüpfung der Variablen ermöglichen (vgl. Kapitel 2.2.2). Auch die Berücksichtigung von Verzögerungseffekten ist erforderlich (vgl. Kapitel 2.2.2). Es sollen mehrere Anpassungsziele hinterlegt werden können, damit in Abhängigkeit von Benutzerinteressen oder äußeren Umweltbedingungen der Anpassungsprozess unterschiedlich gesteuert werden kann. Gemäß der Eigenschaft *Self-Optimization* sollen so unterschiedliche Anforderungsprofile erfüllt werden können (vgl. Kapitel 3.1). Ein Anpassungsziel soll sich aus einer Zielfunktion sowie einer oder mehreren Nebenbedingungen zusammensetzen.

Durch die Entscheidungen bzgl. der Leitfragen der Dimensionen *Observation* und *Representation* ist die Datengrundlage für das selbst-adaptive System geschaffen worden. Die Dimension *Control* befasst sich mit der Steuerung der Anpassung und damit mit der Lösungsfindung (vgl. Lemos et al. 2013, S. 4). Konkret betrachtet diese Dimension die Leit-

frage, welche Veränderungen vorgenommen werden müssen und wie stark diese sein müssen, um die Anpassungsziele zu erreichen (vgl. Lemos et al. 2013, S. 6). Rückkopplungsschleifen nehmen hierbei eine zentrale Rolle ein, um die Auswirkungen der Anpassungen zu erfassen (vgl. Lemos et al. 2013, S. 6). Der Adaptionsmechanismus soll ausgelöst werden, wenn zwischen den erwarteten Planwerten, die aus den Simulationsszenarien stammen, und den gemessenen Istwerten eine Abweichung festgestellt wird. Damit im zeitlichen Verlauf nicht stets nach einer Abweichung gehandelt wird (ex post), ist es vorteilhaft, wenn drohende Abweichungen im Vorfeld erkannt werden und gemäß der *Self-Healing* Eigenschaft frühzeitig darauf reagiert werden kann (vgl. Kapitel 3.1). Die Anpassungen sollen den Arbeitsaufwand eines Experten verringern und zur Komplexitätsreduktion führen, weshalb der Anpassungsprozess möglichst autonom stattfinden und entsprechend der *Self-Configuring* Eigenschaft weitestgehend automatisiert geschehen soll (vgl. Kapitel 3.1). Wenn an einem Modell Veränderungen vorgenommen werden, sollen diese protokolliert werden, um eine Evaluation der Anpassung zu ermöglichen und damit die Qualität einer Lösung zu beurteilen.

Die Dimension *Identification* betrachtet verschiedene Lösungsinstanzen, die das Adaptionsziel erfüllen können (vgl. Lemos et al. 2013, S. 4). Hierfür ist es notwendig, den Zustand des Systems jederzeit identifizieren zu können (vgl. Lemos et al. 2013, S. 6). In einem Zustand können die Systemstruktur, das Systemverhalten sowie die Einstellung der Parameter beschrieben werden (vgl. Lemos et al. 2013, S. 6). Für jeden Zeitpunkt ist es deshalb notwendig, das Modell in seiner Wirkungsstruktur sowie der Parametrisierung vollständig beschreiben zu können. Mit dieser Maßnahme soll, entsprechend der *Self-Awareness*, stets der eigene Zustand beurteilt werden können (vgl. Kapitel 3.1). Da betriebswirtschaftliche Modelle i. d. R. viele Variablen beinhalten und mehrere Dimensionen betrachten können, ist die Zahl der potentiellen Lösungen sehr hoch (vgl. Kapitel 2.3). Aufgrund dessen ist es notwendig, mehrere potentielle Lösungsinstanzen zu erstellen und diese in Abhängigkeit ihrer Genauigkeit gegeneinander zu evaluieren. Für diese Evaluation ist es gemäß der Plan-Phase aus dem *MAPE-K* Zyklus nötig, Kriterien zum Vergleich von Lösungsinstanzen zu definieren (vgl. Kapitel 3.1), damit Fehlanpassungen entsprechend der *Self-Protecting* Eigenschaften vorzeitig vermieden werden (vgl. Kapitel 3.1). In Abhängigkeit der Evaluationsergebnisse sowie unter Berücksichtigung von Präferenzen, die aus den Ziel- sowie Nebenbedingungen stammen, wird eine Lösung entsprechend der Execute-Phase im operativen System mittels eines Aktuators umgesetzt.

Die Dimension *Enabling Adaptation* fokussiert die Frage, wie die Adaption erzielt werden kann (vgl. Lemos et al. 2013, S. 5). Konkret wird in dieser Dimension festgelegt, wie der Adaptionmechanismus angesteuert wird oder wie mit Fehlern umgegangen wird (vgl. Lemos et al. 2013, S. 7). Grundlegend kann bei der Arbeitsweise des Systems zwischen einem internen und externen Ansatz unterschieden werden (vgl. Lemos et al. 2013, S. 7). Bei einem internen Ansatz sind das anzupassende System und der selbst-adaptive Mechanismus integriert (vgl. Lemos et al. 2013, S. 7). Bei einem externen Ansatz wird ein externer Controller für Anpassungen verwendet (vgl. Lemos et al. 2013, S. 7). Der Anpassungsmechanismus soll eine Unterscheidung zwischen parametrischen sowie strukturellen Veränderungen am Modell vornehmen, da es sich um einen unterschiedlich starken Eingriff in das Modell handelt (vgl. Bossel 1994, S. 241). Der Grad der Veränderung des Modells ist bei einer Parameterkalibrierung weniger stark als bei einer Anpassung der Modellstruktur. Für eine stetige Verbesserung der Vorgehensweise sollen die durchgeführten Lösungen rückwirkend bewertet werden können, womit zukünftige Anpassungen durch den Zugriff auf die *Knowledge Bank* des *MAPE-K* Zyklus verbessert werden (vgl. Kapitel 3.1). Die Anpassungen sollten durch einen externen Ansatz und mit Hilfe eines Controllers erfolgen, da unterschiedlich ausgerichtete Modelle parallel existieren können und ein generischer Ansatz verfolgt wird. In Abhängigkeit der Modellvarianten kann es auch notwendig sein, mehrere Anpassungsmethoden bereitzustellen.

Die erhobenen Anforderungen bestimmen unmittelbar den folgenden Systementwicklungsprozess (vgl. Rupp 2014, S. 16). Die finale Systemspezifikation muss alle relevanten Anforderungen abdecken (vgl. Rupp 2014, S. 28). Für die Entwicklung eines Systems gibt es gemäß des IEEE Standards 610.12-1990 deshalb Kriterien, die zur Qualitätssicherung beitragen sollen. Dazu zählen die Merkmale Korrektheit, Eindeutigkeit, Prüfbarkeit und Nachverfolgbarkeit (vgl. IEEE 1990). Demnach sollen die Anforderungen den Bedürfnissen und Zielen entsprechen (*Korrektheit*), es sollte kein Interpretationsspielraum vorhanden sein (*Eindeutigkeit*), anhand von Kriterien sollen die Anforderungsspezifikationen überprüft werden können (*Prüfbarkeit*) und der Ursprung einer Anforderung sollte festzustellen sein (*Nachverfolgbarkeit*). Um diese Vorgaben zu berücksichtigen, werden die abgeleiteten Anforderungsspezifikationen aus diesem Kapitel für die folgende Systementwicklung in Tabelle 4-1 zusammengefasst. Die Spezifikationen sind entsprechend der Dimensionen des Entwurfsraums für selbst-adaptive Systeme kategorisiert und den funktionalen sowie prozessualen Anforderungen aus der Literatur zugeordnet (vgl. Tabelle 4-1).

Bei dem zu entwickelnden selbst-adaptiven System handelt es sich um eine Erweiterung der Standardarchitektur für EUS (vgl. Kapitel 2.2.1). Aufgrund dessen werden die beschriebenen Anforderungen dieser Standardarchitektur (Analysemöglichkeiten, Benutzerschnittstelle, Daten- und Modellbank) vorausgesetzt und nicht erneut in Tabelle 4-1 aufgeführt.

Tabelle 4-1: Zusammenfassung der Anforderungsspezifikationen an das Artefakt

Quelle: Eigene Tabelle

Entwurfsdimension	Anforderungen										Anforderungsspezifikationen	
	funktionale					prozessual						
	Self-Configuring	Self-Protecting	Self-Optimizing	Self-Healing	Self-Awareness	Context-Awareness	Monitor	Analyse	Knowledge	Plan		Execute
Observation					•							Umzusetzende Entscheidungen werden durch das zu entwickelnde System durchgeführt
					•		•					Kontinuierliche Überwachung der operativen Systeme und eigenen Entscheidungen
						•	•					Kontinuierliche Überwachung externer Informationsquellen
Representation											•	Es sollen mehrere Modelle verwaltet werden können
								•				Die Repräsentation der Daten soll je Variable als Zeitreihe erfolgen
			•									Es sollen unterschiedliche Anpassungsziele und Benutzerinteressen hinterlegt werden
Control	•											Eine Anpassung wird ausgelöst, wenn eine Abweichung zw. Soll- und Istwert besteht
				•								Drohende Abweichungen sollen vor dem Eintreten erkannt werden
	•											Der Anpassungsmechanismus soll autonom und weitestgehend automatisiert erfolgen
Identification					•							Stetige Beschreibung des Modellzustandes (Wirkungsstruktur und Parametrisierung)
		•									•	Mehrere potentielle Lösungsinstanzen erstellen und diese gegeneinander evaluieren
					•						•	Eine Anpassung des Entscheidungsmodelles wird durch das System selbst durchgeführt
Enabling Adaptation										•		Der Anpassungsmechanismus unterscheidet zw. Parameter- und Strukturveränderungen
								•				Durchgeführte Anpassungen sollen gemäß der Genauigkeit rückwirkend bewertet werden

4.2 Entwicklung der Model-/System-Controller (MoSyCo) Architektur

Zur Lösung der initial festgestellten Probleme (vgl. Kapitel 1.1) wird eine Architektur für selbst-adaptive modellgetriebene Entscheidungsunterstützungssysteme vorgestellt (vgl. Kapitel 4.2.1). Die drei entworfenen Komponenten des *System-Controller* (vgl. Kapitel 4.2.2), des *Observer* (vgl. Kapitel 4.2.3) sowie des *Model-Controller* (vgl. Kapitel 4.2.4) werden detailliert beschrieben.

4.2.1 Überblick über die Architektur

Zur Erfüllung der Anforderungsspezifikationen wird, wie bereits in Kapitel 4.1 erwähnt, ein System entwickelt. Zu den gängigsten Formen zählt die Definition einer Software Architektur (vgl. Offermann et al. 2010, S. 84). Auch die vorliegende Arbeit entwickelt eine solche Architektur und beschreibt im Folgenden die grundlegende Struktur, erläutert die Prozesse und zeigt die Interaktionsmöglichkeiten auf.

Der Name der entwickelten Architektur ist von den Komponenten geprägt und lautet *Model-/System-Controller Architektur*, das Akronym lautet *MoSyCo*. Die Architektur ist in eine Modell- und Systemperspektive separiert, die miteinander verknüpft sind (vgl. Abbildung 4-1).

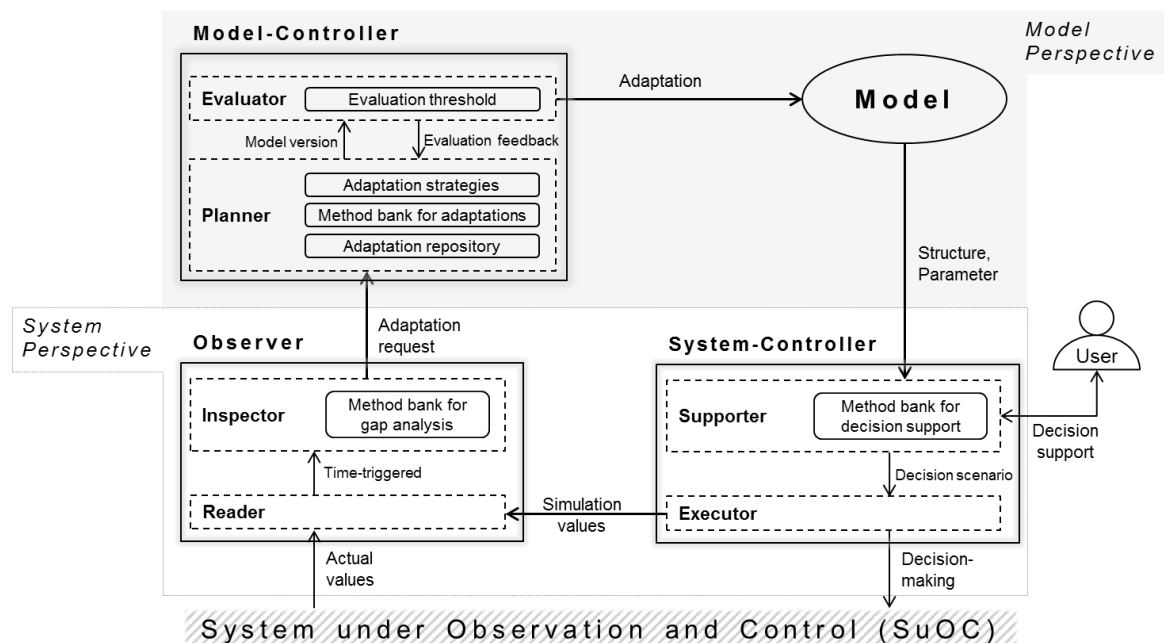


Abbildung 4-1: Model-/System-Controller (MoSyCo) Architektur

Quelle: Breuer (2018, S. 689)

In der Systemperspektive unterstützt die Komponente des *System-Controller* einen Anwender bei der Entscheidungsfindung. Diese Komponente kann Veränderungen in einem oder mehreren operativen Systemen durchführen. Diese werden als *System under Observation and Control* (kurz *SuOC*) bezeichnet. Die Komponente des *Observer* ist für die kontinuierliche Bereitstellung von Daten verantwortlich und überwacht deshalb die operativen Systeme und analysiert diese auf unplanmäßige Entwicklungen. Der *Model-Controller* und das *Model* zur Entscheidungsunterstützung sind beides Komponenten aus der Modellperspektive. Der *Model-Controller* ist für die Erfüllung des avisierten selbst-adaptiven Entscheidungsunterstützungsmodells (vgl. Kapitel 3.4) und dementsprechend für die Planung sowie Durchführung von Anpassungen verantwortlich.

Im Folgenden wird die Architektur aus Abbildung 4-1 anhand der drei Komponenten (umrahmte Kästen) beschrieben. Dabei wird auf ihre Elemente (gestrichelte Kästen) sowie die Interaktion untereinander (Pfeile), aber auch auf das Modell (Ellipse) eingegangen.

4.2.2 Entwurf des System-Controller

Die Beschreibung der *MoSyCo* Architektur aus Abbildung 4-1 beginnt mit einem Benutzer, der bei einer Entscheidung unterstützt werden soll. Jener Dienst wird durch den *System-Controller* bereitgestellt. Dieser repräsentiert u. a. die Funktionalität der Standardarchitektur für Entscheidungsunterstützungssysteme (vgl. Kapitel 4.1) und stellt eine Benutzerschnittstelle, über die ein Anwender mit dem *Supporter* Element interagieren kann, bereit. Dies ermöglicht die Eingabe oder auch Veränderung notwendiger Charakteristika einer Entscheidungssituation. Darüber hinaus stehen dem Anwender in der Methodenbank zur Entscheidungsunterstützung (engl. *Method bank for decision support*) mehrere Funktionen zur Verfügung, die jeweils in Verbindung mit dem Modell genutzt werden können. Zu diesen Methoden gehören bspw. die Zielwertsuche oder auch die Was-wäre-wenn Analyse (vgl. Kapitel 4.1). Mit Hilfe dieser Methoden kann der Anwender iterativ Zukunftsszenarien simulieren und die Auswirkungen von Entscheidungen analysieren. Über mehrere Perioden kann so der Einfluss auf die Ergebnisvariablen überprüft werden (vgl. Kapitel 2.2.2). Für eine Überprüfung ist es für den Anwender sinnvoll, dass die Ergebnisse in visueller Form aufbereitet werden, damit dieser möglichst ganzheitlich und intuitiv eine Bewertung vornehmen kann. Möchte der Anwender eine Entscheidung in der Realität umsetzen, übernimmt innerhalb des *System-Controller* das Element des *Executor* die Veränderung von Entscheidungsvariablen im operativen System und erweitert damit den Funktionsumfang der Standardarchitektur für Entscheidungsunterstützungssysteme (vgl. Kapitel 4.1). Der

Executor fungiert gemäß des *MAPE-K* Zyklus als Steuerelement (engl. Actuator) für das zu verwaltende System (vgl. Kapitel 4.1). Somit werden die Entscheidungen vom Artefakt selbst umgesetzt, wodurch dieses entsprechend der *Self-Awareness* die eigenen Aktionen protokolliert (vgl. Kapitel 4.1). Da eine Entscheidung mit Hilfe von zukunftsorientierten Analyseverfahren (Zielwertsuche oder Was-wäre-wenn Analyse) getroffen wird, kann jeder Entscheidung eine durch das Modell berechnete zukünftige Entwicklung zugeordnet werden. Bei jeder Umsetzung einer Entscheidung leitet der *Executor* diese simulierten Werte über die erwartete Entwicklung zum *Reader* Element des *Observers* weiter.

4.2.3 Entwurf des Observer

Die Komponente des *Observer* besteht aus zwei Elementen, dem *Reader* und dem *Inspector*. Die zentrale Aufgabe des *Observer* ist eine kontinuierliche Überwachung sowie Analyse des *SuOC*. Die simulierten Werte aus dem *Executor* sind hierfür eine von zwei Datenquellen. Die zweite Datenquelle ist das operative und zu überwachende System (*SuOC*), aus dem die Istwerte stammen. Der *Reader* stellt gemäß den *Sensoren* im *MAPE-K* Zyklus eine Verbindung über Schnittstellen zu verschiedenen operativen sowie externen Quellen her und sorgt für ein stetiges *Monitoring* der betrieblichen Systeme sowie der externen Unternehmensumwelt. Somit werden gemäß der Eigenschaft der *Context-Awareness* auch die Umweltbedingungen überwacht. Diese Daten repräsentieren Istwerte, die der *Reader* neben den Planwerten dauerhaft für Analysezwecke vorhält. Beide Zeitreihen werden im Element *Inspector* miteinander verglichen. Hierfür werden die Daten zeitgesteuert an den *Inspector* weitergeleitet, die Häufigkeit einer Weiterleitung hängt von den Aktualisierungsraten der operativen sowie externen Systeme ab. Dies kann für modellgetriebene EUS in nahezu Echtzeit erfolgen (vgl. Kapitel 4.1). Die Aufgabe des *Inspector* ist es, die Plan- und Istwerte zu analysieren (vgl. Analyse-Phase in Kapitel 4.1), um Abweichungen zu identifizieren. Abweichungsanalysen (engl. Gap-Analyse, vgl. Abbildung 2-2) werden auch als Soll-Ist-Vergleiche bezeichnet und sind ein wichtiger Baustein der betriebswirtschaftlichen Planung (vgl. Stelling 2008, S. 95). Durch die Analyse wird eine mögliche Differenz zwischen dem Soll- und Ist-Betrag identifiziert (vgl. Stelling 2008, S. 90). Hierfür stehen dem *Inspector* unterschiedliche Methoden zur Abweichungsanalyse (engl. *Method bank for gap analysis*) zur Verfügung. Diese sollen nicht nur vergangenheitsorientiert sein, sondern gemäß der *Self-Healing* Eigenschaft auch zukünftige Abweichungen antizipieren können. Somit soll das System nicht nur auf Abweichungen reagieren, sondern diese prognostizieren können, damit unnötige Anpassungen vermieden, die Ineffizienz inkrementeller Anpas-

sungen reduziert, vorsorgliche Anpassungen an saisonale Schwingungen ermöglicht und insgesamt die Handlungsmöglichkeiten für Anpassungen vergrößert werden (vgl. Cheng et al. 2009, S. 72). Für diesen Zweck sind in Abhängigkeit der zeitlichen Dimension zwei Arten von Methoden verfügbar: Zeitpunkt- und Zeitreihenabhängige Methoden (vgl. Backhaus et al. 2018c, S. 126). Bei zeitpunktbezogenen Berechnungen werden bspw. absolute oder relative Abweichung für einen bestimmten Zeitpunkt ermittelt (vgl. Schön 2018, S. 52). Zeitreihenbezogene Methoden ermöglichen das Entdecken und Diagnostizieren von langfristigen Entwicklungen über mehrere Zeitpunkte wie bspw. einem Trend oder einer Saisonalität (vgl. Backhaus et al. 2018d, S. 17). Auf Grundlage dieser Methodenarten können Entwicklungen, die von der geplanten Entwicklung abweichen, identifiziert werden. Wenn eine dieser Methoden im Ergebnis eine Abweichung zwischen den Ist- und modellbasierten Planwerten erkennt und diese außerhalb eines definierbaren Toleranzbereiches liegt, wird der Anpassungsmechanismus ausgelöst, indem eine Anpassungsaufforderung (engl. *adaptation request*) an den *Planner* innerhalb des *Model-Controller* übermittelt wird (vgl. Abbildung 4-1).

4.2.4 Entwurf des Model-Controller

Das Element des *Planner* empfängt die Anpassungsaufforderung und ist verantwortlich für die Entwicklung einer Lösung, die das Modell so anpasst, dass die gemessene Abweichung durch das angepasste Modell erklärt werden kann. Das fokussierte Problem ist, dass die aktuelle Entwicklung des operativen Systems durch die Simulationen des Modells nicht adäquat vorhergesagt wurde. In diesem Fall repräsentiert das Modell nicht ausreichend die Realität, sodass es an die neuen Umweltbedingungen angepasst werden muss. Die Ursache der Ungenauigkeiten des Modells kann entweder eine unzureichende Kalibrierung der Parameter oder eine falsche Wirkungsstruktur oder beides sein (vgl. Kapitel 2.3). Aufgrund dessen ist eine erneute Kalibrierung der Modellparameter und/oder eine Umstrukturierung der Ursache-Wirkungs-Beziehungen innerhalb des Modells notwendig. Für diese beiden Anpassungen benötigt der *Planner* die Informationen über das aktuelle Modell. Da das Artefakt unabhängig von einer spezifischen Anwendung zur modellgetriebenen Entscheidungsunterstützung sein soll, wird das Modell generisch vorgehalten. Für die Repräsentation eines generischen Entscheidungsunterstützungsmodells findet eine Orientierung an der Graphentheorie statt. Graphen werden in der Informatik häufig zur Modellierung eingesetzt (vgl. Meinel und Mundhenk 2015, S. 223), da diese komplexe Informationen und Sachverhalte u. a. aus dem Bereich der Wirtschaftswissenschaften für den Menschen in

einer adäquaten Art und Weise visuell darstellen können (vgl. Meinel und Mundhenk 2015, S. 236). Ein Graph besteht aus zwei Bestandteilen: Knoten und Kanten (vgl. Meinel und Mundhenk 2015, S. 224). Kanten repräsentieren eine Verbindung oder Beziehung zwischen zwei Knoten. Auf Basis der Graphentheorie sind für eine computergestützte Analyse von Graphen Datenmodelle für spezielle Datenbanken entwickelt worden (vgl. Edlich 2011, S. 207). Eine Ausprägung dieser Entwicklung ist das Property-Graph-Modell (PGM), bei dem „sowohl in den Knoten als auch in den Kanten weitere Informationen – sogenannte Properties (Eigenschaften) – gespeichert werden können“ (Edlich 2011, S. 210). Bei den Eigenschaften handelt es sich um Key/Value-Beziehungen, die anwendungsspezifisch sind und im Rahmen der Modellierung frei wählbar (vgl. Edlich 2011, S. 210 f.). Dies ermöglicht, dass der Typ eines Knotens oder einer Kante definiert werden kann, wodurch die Daten eine Semantik bekommen (vgl. Edlich 2011, S. 211). Kanten sind in einem Property-Graph-Modell stets gerichtet, d. h. es besteht eine einseitige Beziehung eines Knotens zu einem anderen (vgl. Edlich 2011, S. 209 ff.).

Zur Repräsentation eines Entscheidungsunterstützungsmodells sind Knoten für Variablen vorgesehen und eine Kante bildet die Wirkung einer Variable auf die andere ab. Mit Hilfe von Eigenschaften an den Knoten und Kanten werden u. a. die unterschiedlichen Variablentypen sowie mathematischen Operatoren repräsentiert (vgl. Kapitel 2.2.2).

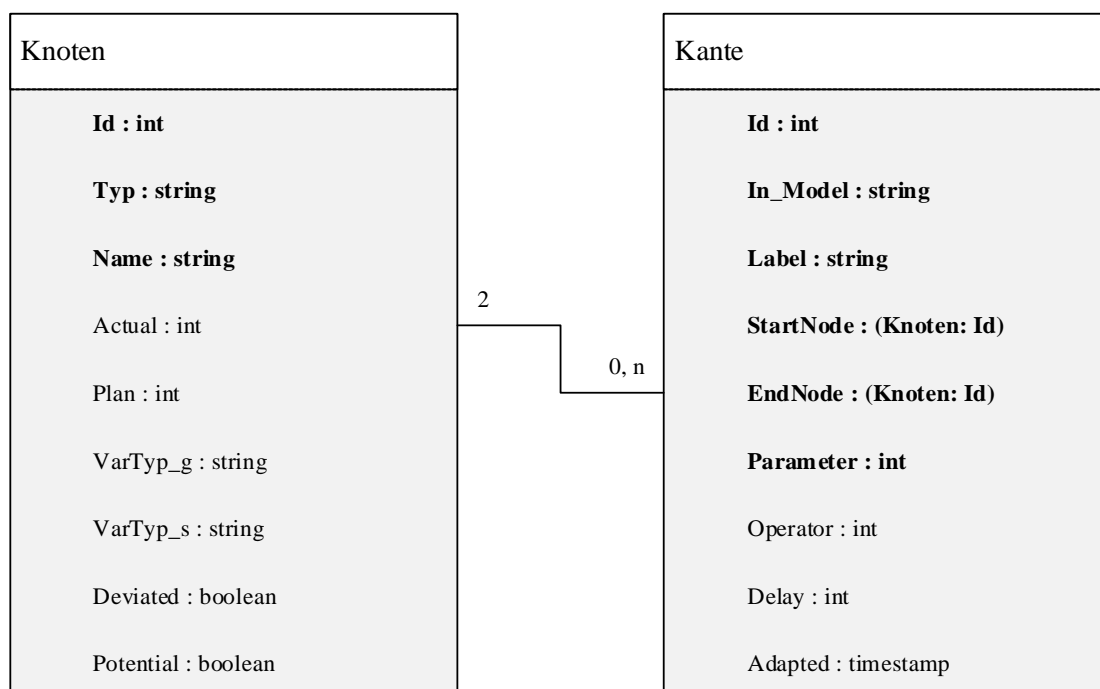


Abbildung 4-2: Datenmodell eines PGM für betriebswirtschaftliche EUS-Modelle

Quelle: Eigene Darstellung

Ein Knoten kann immer anhand einer eindeutigen `id` identifiziert werden. Als `Typ` eines Knotens sind aktuell nur `Variablen` vorgesehen. Darüber hinaus wäre es denkbar, dass auch Unternehmensziele oder -strategien als Knoten repräsentiert werden können. Jede Variable verfügt über einen `Namen`. Diese drei Eigenschaften müssen festgelegt werden, weshalb diese in der Abbildung 4-2 hervorgehoben sind. Jedem Knoten kann ein aktueller `Ist- (Actual)` sowie `Planwert` zugeordnet werden, hierbei handelt es sich jeweils um den Datentyp `Integer`. Die beiden Eigenschaften `VarTyp_g` und `VarTyp_s` spezifizieren den Variablentyp genauer. In beiden Fällen kann aus einer Liste an Ausprägungen ausgewählt werden. Der `VarTyp_g` spezifiziert eine Variable in seiner generischen Form: Hilfs-, Umgebungs-, Entscheidungs- oder Ergebnisvariable (vgl. Kapitel 2.2.2). Der `VarTyp_s` benennt den spezifischen Variablentyp, der von der Anwendung oder auch Modellierungsart zur Entscheidungsunterstützung abhängig ist. Bspw. können mit Hilfe von System Dynamics betriebswirtschaftliche Flussmodelle erstellt werden, bei denen zwischen einer Bestands-, Hilfs- und Flussvariable unterschieden wird (vgl. Bossel 2004a, S. 379 f.). Der Eigenschaft `Deviated` ist ein boolescher Wert zugeordnet, der angibt, ob aktuell eine Abweichung vorliegt. `Potential` gibt an, ob die Variable ein potentieller Kandidat ist, der bei einer Modellanpassung untersucht werden sollte. Die Identifikation, Überprüfung und Bewertung von Variablen ist ein elementarer Schritt bei der Modellerstellung (vgl. Power 2002, S. 159). Auch wenn Variablen bei der Modellerstellung als nicht relevant beurteilt wurden, sieht der vorliegende Forschungsansatz vor, dass diese Variablen trotzdem in dem Property-Graph-Modell vorgehalten werden. Dies geschieht unter der Annahme, dass die Wirkungsstruktur eines Modells sich im Laufe der Zeit ändert und dass zuvor ausgeschlossene Variablen einen Einfluss innerhalb des Modells entwickeln können. So werden im Rahmen der Modellbildung diverse Variablen auf ihre Relevanz in Abhängigkeit des Modellziels von Systemanalytikern, Systemkennern sowie Experten aus den Wissensgebieten überprüft (vgl. Bossel 2004a, S. 26). Die bei der Modellbildung als nicht-relevant beurteilten Variablen können mit Hilfe der Eigenschaft `Potential` entsprechend gekennzeichnet werden. Darüber hinaus zeichnet diese Knoten aus, dass sie nicht über eine Kante mit anderen Knoten verknüpft sind, weshalb die Kardinalität von 0 gelten kann. Eine Kante ist genau zwei Knoten zugeordnet und weist neben einer eindeutigen `id` die Eigenschaft `In_Model` auf. Diese spezifiziert, in welchem Modell eine Wirkung zwischen zwei Variablen besteht. Da Entscheidungsunterstützungsmodelle stets problemabhängig sind, können innerhalb einer betriebswirtschaftlichen Organisation mehrere Modelle mit jeweils einem

unterschiedlichen Fokus existieren (vgl. Power 2002, S. 161). In unterschiedlichen Modellen kann dieselbe Variable zur Planung genutzt werden, bspw. hat der Absatz sowohl bei der Kapital- als auch bei der Produktionsplanung einen Einfluss. Aufgrund dessen kann ein Knoten bis zu n Kanten aufweisen und entsprechend mehrfach in Modellen berücksichtigt werden. Das `Label` einer Kante definiert die Art des Einflusses. Da in einem Property-Graph-Modell alle Kanten gerichtet sind, kann zwischen einem `Start-` und `Endknoten` (engl. `Node`) differenziert werden. Die Zuordnung erfolgt durch die eindeutige `Id` eines Knotens. Der `Parameter` bestimmt die Wirkungsintensität. Der `Operator` definiert den mathematischen Transformationsschritt zwischen den beiden Variablen. Besteht eine zeitliche Verzögerung in der Wirkung, kann der Verzögerungsfaktor durch die Eigenschaft `Delay` definiert werden. Hat am Modell eine parametrische oder strukturelle Anpassung stattgefunden, d. h. sind die Eigenschaften einer bestehenden Kante verändert worden oder ist eine neue Kante zwischen zwei Knoten entstanden, wird der Zeitpunkt der Anpassung in der Eigenschaft `Adapted` protokolliert.

Ein exemplarischer Property-Graph mit dem zuvor erläuterten Datenmodell ist in Abbildung 4-3 dargestellt.

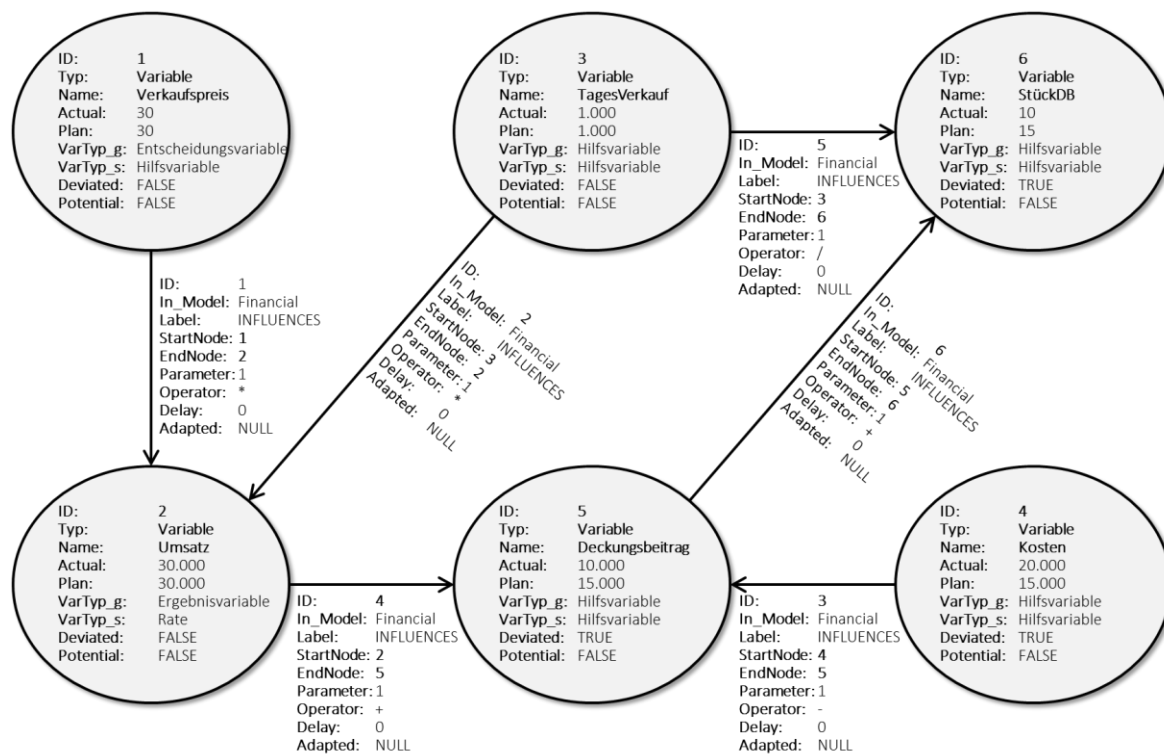


Abbildung 4-3: Property-Graph für einen exemplarischen Modellausschnitt

Quelle: Eigene Darstellung

Die Kreise repräsentieren Knoten, die jeweils mit gerichteten Kanten in Form eines Pfeiles miteinander verbunden sind. Der Knoten mit der ID 2 stellt die Variable Umsatz dar. Der Istwert (Actual) sowie der Planwert stimmen mit 30.000 überein, weshalb keine Abweichung gemessen wird und Deviated den Wert FALSE annimmt. Beim Umsatz handelt es sich um den generischen Variablentyp (VarTyp_g) einer Ergebnisvariable, anhand der eine Entscheidung beurteilt werden kann. Der Umsatz ist das multiplikative Ergebnis aus der Entscheidungsvariable Verkaufspreis und der Hilfsvariable TagesVerkauf, weshalb das Rechenzeichen für die Multiplikation (*) unter der Eigenschaft Operator in den Kanten mit der ID 1 und 2 geführt werden. Das Gewicht der beiden Faktoren ist jeweils 1, weshalb die Parameter der beiden Kanten 1 sind. Da sowohl der Verkaufspreis als auch der TagesVerkauf den Umsatz beeinflussen, weisen die Kanten das Label INFLUENCES auf. Eine Zeitverzögerung (Delay) liegt nicht vor. Eine Anpassung hat noch nicht stattgefunden (Adapted = NULL). Der Knoten mit der ID 5 repräsentiert den Deckungsbeitrag, welcher vom Planwert um 5.000 Einheiten abweicht. Der Deckungsbeitrag ist die Subtraktion aus Umsatz und Kosten, die jeweils den Parameter 1 haben. Aus der Kombination von Startknoten (StartNode) und Operator kann die Berechnungslogik für jede Variable ermittelt werden. Bei einer Subtraktion bestimmt den Subtrahenden der Operator einer Kante (vgl. Formel 4-1).

$$\text{Deckungsbeitrag} = + \text{Umsatz} - \text{Kosten}$$

Formel 4-1: Berechnungslogik einer Subtraktion im PGM (vgl. Abbildung 4-3)

Quelle: Eigene Darstellung

Im Falle einer Division, wie am Beispiel des Stückdeckungsbeitrages (kurz StückDB), definiert der jeweilige Operator einer Kante den Wert des Quotienten (vgl. Formel 4-2).

$$\text{StückDB} = \frac{+ \text{Deckungsbeitrag}}{\text{TagesVerkauf}}$$

Formel 4-2: Berechnungslogik einer Division im PGM (vgl. Abbildung 4-3)

Quelle: Eigene Darstellung

Sollte im Modell auf die Zwischenberechnung des Deckungsbeitrages verzichtet werden und sich der StückDB direkt aus Differenz von Umsatz und Kosten berechnen, müsste entsprechend der Verschachtelung eine Hilfsvariable für die Berechnung (Umsatz - Kosten) im Property-Graph-Modell eingeführt werden. Bei einer Addition sowie Multiplikation ist die Reihenfolge für die Berechnung irrelevant, sodass an den Kanten derselbe Operator

geführt werden kann (vgl. Kanten $\text{Id } 1$ und 2 in Abbildung 4-3). Die Grundrechenarten lassen sich durch das vorgestellte Datenmodell für das Property-Graph-Modell realisieren.

Muss eine Modellanpassung durchgeführt werden, da bspw. der Deckungsbeitrag vom Planwert abweicht (vgl. Abbildung 4-3), hat der *Planner* für diesen Zweck Zugriff auf die Datenbank des *Reader* Elements (vgl. Abbildung 4-1) und ist sich durch das Property-Graph-Modell des aktuell genutzten Modells bewusst, welches aufgrund der gemessenen Abweichung jedoch nicht mehr repräsentativ ist. Durch die beiden Informationsquellen soll sich das System gemäß der *Self-Awareness* des eigenen Zustandes bewusst sein. Das Ziel des *Planner* ist ein angepasstes und der Realität entsprechendes Modell. Dieses Ziel kann unter Berücksichtigung verschiedener Anpassungsstrategien (engl. *adaptation strategies*) erreicht werden (vgl. Abbildung 4-1). Die Anwendung von Regeln oder Strategien ermöglicht einen dynamischen Anpassungsprozess. In Abhängigkeit des Fokus eines Modells könnten unterschiedliche Genauigkeiten bei der Güte eines potentiell neuen Modells gefordert werden. So könnte die erwartete Güte eines Produktionsmodells höher sein als die eines Absatzmodells, da letzteres ggf. mehr von unkontrollierbaren und somit intransparenten Variablen geprägt ist. Der Typ einer abgewichenen Variable könnte auch einen Einfluss auf die Wahl einer Anpassungsmethode haben (vgl. Abbildung 2-4). Gemäß der *Self-Optimizing* Eigenschaft beeinflusst somit das Problem, das ein Modell definiert, den Lösungsprozess und dieser kann entsprechend optimiert werden. Die Anwendung von strukturentdeckenden Verfahren ist bspw. für externe Umweltvariablen geeignet, da diese nicht immer präzise gemessen werden können oder Wirkungen nicht vollständig bekannt sind (vgl. Bossel 1994, S. 37 f.). Die Methodenbank für Anpassungen (engl. *method bank for adaptations*) enthält sowohl strukturprüfende als auch strukturentdeckende Verfahren, um parametrische wie auch strukturelle Anpassungen am Modell vorzunehmen (vgl. Backhaus et al. 2018d, S. 15). Strukturprüfende Verfahren evaluieren die Zusammenhänge von Variablen, indem die kausalen Abhängigkeiten überprüft und die Parameter ggf. angepasst werden. Die Struktur bleibt in diesem Fall unverändert, lediglich die Parameter werden neu kalibriert (vgl. Backhaus et al. 2018d, S. 15). Strukturentdeckende Verfahren sind spezialisiert auf das Entdecken von unbekanntem Zusammenhängen zwischen Variablen. Durch diese Verfahren wird auch die Modellstruktur verändert, indem potentielle Beziehungen aufgedeckt werden und der Einfluss einer neuen Variable in Form des Parameters quantifiziert wird (vgl. Backhaus et al. 2018d, S. 15). Somit stellen strukturentdeckende Verfahren einen Lösungsansatz dar, mit dem die Herausforderung von strukturellen Veränderungen

an einem quantitativen Entscheidungsunterstützungsmodell (vgl. Kapitel 3.2.2.3) gelöst werden kann. Insgesamt soll durch den Einsatz von strukturprüfenden und strukturentdeckenden Verfahren das Modell so angepasst werden, dass die Entwicklung des operativen Systems durch das adaptierte Modell erklärt werden kann. Die Anpassung soll gemäß der *Self-Configuring* Eigenschaft autonom sowie automatisiert erfolgen und soll die Diskrepanz zwischen Realität und Modell reduzieren.

Sofern das quantitative Verhalten des Modells näherungsweise gleich dem Verhalten des operativen Systems ist, wird von einer Verhaltensgültigkeit gesprochen (vgl. Bossel 2004a, S. 61). Das Repository für Anpassungen (engl. *adaptation repository*) speichert jede Lösungsinstanz in Form der strukturellen sowie parametrischen Veränderungen, aber auch das Feedback des *Evaluator* (vgl. Abbildung 4-1). Der *Evaluator* prüft die Anpassungen eines potentiellen Modells. Im Rahmen der Planungsphase (vgl. *Plan* in Kapitel 4.1) können bspw. durch die Berücksichtigung unterschiedlicher Strategien mehrere Modelle entwickelt werden, weshalb der *Evaluator* im Anschluss eine wichtige Aufgabe zu erfüllen hat. Dieser muss aus einer Menge von potentiellen Modellen, auch Experimentalmodelle genannt (vgl. Sauter 2010, S. 130 f.), die beste Lösungsinstanz auswählen. Da aus dem operativen System mit fortlaufender Zeit stetig zusätzliche Istwerte gemessen werden, können diese als Evaluationskriterium verwendet werden. Das Ziel ist es, durch ein Experimentalmodell so gut wie möglich die Entwicklung der Realität zu rekonstruieren. Ist eine ausreichende Genauigkeit erreicht, kann dieses Experimentalmodell als repräsentativ gewertet und die Veränderungen dieser Lösungsinstanz können im Modell (engl. *Model*) zur Entscheidungsunterstützung in der Systemperspektive umgesetzt werden. Mit diesem Schritt wird nach dem *MAPE-K* Zyklus eine geplante Maßnahme zur Modellanpassung umgesetzt (vgl. *Execute* in Kapitel 4.1). Durch das Generieren von Experimentalmodellen kann der *Planner* die eigene Leistung evaluieren und ineffiziente Anpassungen vermeiden, wodurch die *MoSyCo* Architektur langfristig dazu in der Lage sein soll, die Leistung gemäß der *Self-Protecting* Eigenschaft iterativ zu verbessern. Für diesen Zweck werden die Ergebnisse aus der Modellevaluation als Feedback im Repository für Anpassungen gespeichert (vgl. Abbildung 4-1), damit Wissen über spezifische Anpassungen und deren Einfluss auf das System aufgebaut werden kann.

Strukturelle oder parametrische Veränderungen im *Model-Controller* führen zu einem aktualisierten Modell zur Entscheidungsunterstützung (vgl. Abbildung 4-1). In der Systemperspektive ist somit eine neue und realistischere Grundlage zur Entscheidungsunterstüt-

zung verfügbar. Sofern eine neue Entscheidung durch den *System-Controller* umgesetzt wird, leitet der *Executor* erneut die dazugehörigen simulierten Daten zum *Reader* weiter (vgl. Abbildung 4-1). Dadurch verfügt der *Inspector* über neue Vergleichswerte für die Abweichungsanalyse, womit die *MoSyCo* Architektur für eine kontinuierliche Evaluation und, sofern es die Umweltbedingungen bedürfen, für eine ständige Anpassung des Modells sorgt.

5 Ex ante Evaluation der MoSyCo Architektur

Zur Überprüfung der spezifizierten Architektur auf Vollständigkeit und Korrektheit wird im Rahmen der ex ante Evaluation Eval 2 (vgl. Kapitel 1.2) eine Umfrage mit Teilnehmern aus der Praxis durchgeführt. Das Untersuchungsdesign und die Hypothesen werden zu Beginn erläutert (vgl. Kapitel 5.1) und im Anschluss die Ergebnisse der Untersuchung vorgestellt (vgl. Kapitel 5.2).

5.1 Untersuchungsdesign und Hypothesen

Bei einer ex ante Evaluation wird das entwickelte Artefakt validiert (vgl. Sonnenberg und vom Brocke 2012, S. 392). Dieser Evaluationsschritt wird vor einer prototypischen Implementierung durchgeführt und prüft die Spezifikationen auf Korrektheit und Vollständigkeit (vgl. Sonnenberg und vom Brocke 2012, S. 395). Hierfür soll die Form und Funktionsweise eines Artefaktes in Hinblick auf das zu lösende Problem überprüft werden (vgl. Sonnenberg und vom Brocke 2012, S. 394). Die Probleme im Umgang mit EUS in einer dynamischen Unternehmensumwelt sind bereits in Kapitel 3.3 identifiziert und bestätigt worden. Die spezifizierten Komponenten sowie Funktionen des Artefaktes müssen zur Lösung dieser initial identifizierten Probleme beitragen, damit das Artefakt insgesamt vollständig und korrekt ist. Aufgrund dessen werden im Rahmen der ex ante Evaluation die Probleme in Relation zu den Funktionen der Lösungskomponenten gesetzt und der Problem-Lösung-Fit untersucht (vgl. Abbildung 5-1).

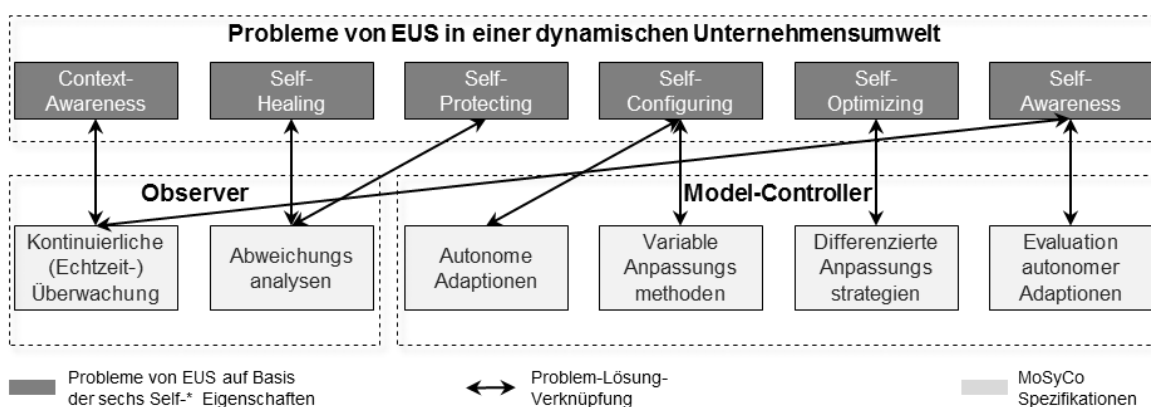


Abbildung 5-1: Verknüpfungen zw. MoSyCo Spezifikationen und initialen Problemen

Quelle: Eigene Darstellung

Eine kontinuierliche (Echtzeit-) Überwachung trägt bspw. dazu bei, dass die eigene Leistung beurteilt werden kann und somit Probleme identifiziert werden können (*Self-Awareness*), aber auch das operative Umfeld für eine *Context-Awareness* analysiert werden kann (vgl. Abbildung 5-1).

Als Evaluationsmethode empfehlen Sonnenberg und vom Brocke (2012, S. 393) u. a. eine Umfrage, weshalb der Fragebogen aus Kapitel 3.3 um die ex ante Evaluation erweitert wurde und die Ergebnisse somit auf derselben Stichprobe basieren. Die Teilnehmer bewerten Aussagen über das Potential der Spezifikationen zur Lösung der initial festgestellten Probleme, die auf den sechs *Self-** Eigenschaften basieren (vgl. Kapitel 3.3). In Tabelle 5-1 sind die Aussagen über die Problem-Lösung-Verknüpfung aufgeführt, die anhand einer fünfstufigen Likert-Skala, welche von 1 (Trifft nicht zu) bis 5 (Trifft zu) skaliert ist (vgl. Schnell 2005, S. 187 f.), bewertet werden können.

Tabelle 5-1: Zu bewertende Aussagen über den Problem-Lösung-Fit

Quelle: Eigene Tabelle

Eigen-schaft	Spezi-fikation	Vari-able	Aussage
Self-Awareness	Kontinuierliche (Echtzeit-) Überwachung	BGS1 m2	Durch die kontinuierliche Überwachung der operativen Daten können Störungen und Probleme identifiziert werden.
Context-Awareness	Kontinuierliche (Echtzeit-) Überwachung	BGS1 m4	Durch die kontinuierliche Überwachung der operativen Daten ist sich das System über das operative Umfeld bewusst.
Self-Configuring	Autonome Adaption	BGS1 m5	Durch autonome Anpassungen kann sich das System automatisch an geänderte Umwelt- und Rahmenbedingungen anpassen.
Self-Configuring	Variable Anpassungsmethoden	BGS1 m6	Durch variable Anpassungsmethoden wird der Wartungsaufwand reduziert.
Self-Optimizing	Differenzierte Anpassungsstrategien	BGS1 m7	Durch unterschiedliche Anpassungsstrategien kann sich das System dynamisch auf unterschiedliche Problemstellungen einstellen.
Self-Healing	Abweichungsanalysen	BGS1 m8	Durch unterschiedliche Methoden zur Analyse von Abweichungen können ungeplante Entwicklungen vorzeitig erkannt werden.
Self-Awareness	Evaluation autonomer Adaptionen	BGS1 m9	Durch die Erfassung von durchgeführten Anpassungen kann das System rückwirkend die eigenen Maßnahmen evaluieren.
Self-Protecting	Abweichungsanalysen	BGS1 m10	Durch Methoden für zeitreihenbezogene Analysen (bspw. von Trends oder Saisonalitäten) können zukünftige Fehler antizipiert werden und das System kann sich vor Fehlern selbst schützen.

Da die ex ante Evaluation auch anwendungsorientierte Evaluationskriterien wie die Nutzbarkeit untersucht (vgl. Sonnenberg und vom Brocke 2012, S. 393), wird in Anlehnung an das Technology Acceptance Model 2 (TAM2) nach Venkatesh und Davies (2000) angenommen, dass ein höherer Problem-Lösung-Fit zu einer höheren Nutzungsintention der *MoSyCo* Architektur sowie zu einer erhöhten wahrgenommenen Nützlichkeit führt. Für diese Untersuchung ist es notwendig, die *MoSyCo* Architektur als Ganzes zu betrachten und nicht auf Komponentenebene, da eine Teilnutzung einzelner Komponenten nicht möglich ist. Für die ganzheitliche Betrachtung wird der Mittelwert über alle Antworten der acht zu bewertenden Aussagen berechnet (vgl. Tabelle 5-1). Auf Grundlage dieses Mittelwertes wird die Auswirkung des Problem-Lösung-Fit auf die Nutzungsintention sowie die wahrgenommenen Nützlichkeit untersucht (vgl. Abbildung 5-2).

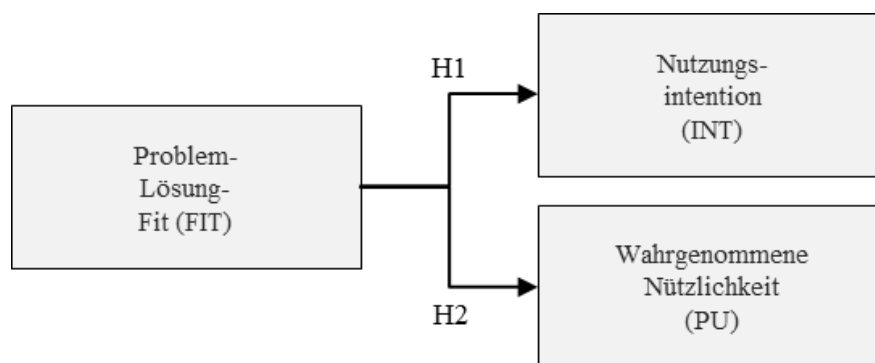


Abbildung 5-2: Ex ante Evaluation der Nutzungsintention und der wahrgenommenen Nützlichkeit

Quelle: Eigene Darstellung

Die zu untersuchenden Hypothesen lauten:

H 5-1: Je höher der Problem-Lösung-Fit ist, desto höher ist die Nutzungsintention (INT) der *MoSyCo* Architektur.

H 5-2: Je höher der Problem-Lösung-Fit ist, desto höher ist die wahrgenommene Nützlichkeit (PU) der *MoSyCo* Architektur.

Anhand dieser beiden Hypothesen sollen die Relevanz des Artefaktes für die unternehmerische Praxis und die Notwendigkeit einer prototypischen Implementierung überprüft werden. Die zu bewertenden Aussagen bzgl. der Nutzungsintention sowie der wahrgenommenen Nützlichkeit sind in Tabelle 5-2 aufgeführt. Im Vorfeld zu diesen Aussagen wurden den Umfrageteilnehmern die Spezifikationen der *MoSyCo* Architektur in Hinblick auf ein selbst-adaptives modellgetriebenes Entscheidungsunterstützungssystem erläutert, weshalb

in den Aussagen die Formulierung „solches Entscheidungsunterstützungssystem“ verwendet wird (vgl. Tabelle 5-2).

Tabelle 5-2: Aussagen zur Nutzungsintention sowie der wahrgenommenen Nützlichkeit

Quelle: Eigene Tabelle

Konstrukt	Variable	Aussage
Wahrgenommene Nützlichkeit	BGS2m1	Ein solches Entscheidungsunterstützungssystem kann meine alltägliche Arbeit erleichtern.
Nutzungsintention	BGS2m2	Ich beabsichtige, mit einem solchen Entscheidungsunterstützungssystem zu arbeiten.
Wahrgenommene Nützlichkeit	BGS2m3	Outputs aus einem solchen Entscheidungsunterstützungssystem können mir eine fundierte Entscheidungsgrundlage geben.
Nutzungsintention	BGS2m4	Ich würde ein solches Entscheidungsunterstützungssystem in Zukunft nutzen.

Zur Überprüfung der Hypothesen H5-1 und H5-2 werden Korrelationsanalysen durchgeführt. Anhand dieses Verfahrens lassen sich Zusammenhänge zwischen den Variablen FIT, INT und PU testen (vgl. Abbildung 5-2). Im Ergebnis einer Korrelationsanalyse wird auch die Stärke und die Richtung einer Wirkung angegeben. Die Auswahl eines geeigneten Verfahrens hängt vom Messniveau der Fragen ab (vgl. Eckstein 2016, S. 163). Das Signifikanzniveau der verwendeten Analyseverfahren liegt bei $\alpha = 0,05$. Die Wahrscheinlichkeit, eine Nullhypothese bei einem Signifikanztest abzulehnen, obwohl diese wahr ist, liegt demnach bei 5% (vgl. Eckstein 2019, S. 276). Die angewandten Analyseverfahren werden im Folgenden kurz erläutert.

Die *Pearson Korrelation* wird genutzt, um die Stärke und Richtung einer linearen Beziehung zwischen zwei metrischen Variablen zu bestimmen (vgl. Bourier 2018, S. 208). Der Korrelationskoeffizient r zeigt genau diese Stärke und Richtung einer Beziehung zweier Variablen an und nimmt Werte von -1, für eine negative Wirkung, bis +1, für eine positive Wirkung, an. Sind beide Variablen unkorreliert, ist der Korrelationskoeffizient null. Um eine Korrelationsanalyse nach Pearson durchführen zu können, müssen die Variablen metrisch sein sowie eine lineare Beziehung aufweisen, es darf keine Ausreißer geben und die Daten müssen normalverteilt sein (vgl. Siebertz et al. 2017, S. 381). Der Korrelationskoeffizient nach Pearson in der Grundgesamtheit soll geschätzt werden und dieser Grundge-

samtheitskorrelationskoeffizient, bezeichnet mit ρ , soll nicht null sein. Daher lassen sich die folgende Nullhypothese und Alternativhypothese für die Korrelationsanalyse nach Pearson aufstellen:

H_0 : $\rho = 0$, d. h. der Korrelationskoeffizient der Grundgesamtheit ist null

H_A : $\rho \neq 0$, d. h. der Korrelationskoeffizient der Grundgesamtheit ist nicht null

Die *Spearman's Rangkorrelation* wird alternativ zur Pearson Korrelation genutzt, da diese weniger Annahmen unterliegt (vgl. Bourier 2018, S. 218 f.). Die zu überprüfenden Variablen können entweder beide metrisch, beide ordinal oder eine metrisch und eine ordinal skaliert sein. Die Interpretation der Korrelationskoeffizienten ist gleichbedeutend wie bei der Pearson Korrelation. Um die Rangkorrelation durchzuführen, müssen die Variablen nicht linear, sondern lediglich monoton sein (vgl. Siebertz et al. 2017, S. 385 f.). Die Nullhypothese und Alternativhypothese für die Spearman's Rangkorrelation lauten:

H_0 : $\rho = 0$, d. h. der Korrelationskoeffizient ist null

H_A : $\rho \neq 0$, d. h. der Korrelationskoeffizient ist nicht null

Die im Rahmen einer Korrelationsanalyse notwendige Überprüfung von Ausreißern wird durch eine visuelle Inspektion der Boxplots vorgenommen. Die Normalverteilung der Daten wird mit Hilfe des Shapiro-Wilk-Tests überprüft (Shapiro und Wilk 1965).

Als Abschluss der ex ante Evaluation wird jede einzelne Spezifikation der *MoSyCo* Architektur in Hinblick auf eine prototypische Implementierung von den Umfrageteilnehmern bewertet, um besonders wichtige Funktionen zu identifizieren. Bei einem Prototyp handelt es sich um ein experimentelles System, welches für Testzwecke erstellt wird und deshalb auch nur einen Teil des konzipierten Artefaktes repräsentieren kann (vgl. Laudon et al. 2010, S. 936). Aus diesem Grund werden die Spezifikationen, wie der Tabelle 5-3 zu entnehmen ist, in natürlicher Sprache wiedergegeben (vgl. Rupp 2014, S. 187) und auf Basis der bereits zuvor beschriebenen fünfstufigen Likert-Skala von den Umfrageteilnehmern bewertet.

Auf Grundlage der Bewertung der Spezifikationen aus Tabelle 5-3 kann bei der prototypischen Implementierung ggf. eine Priorisierung bestimmter Funktionen erfolgen.

Tabelle 5-3: Zu bewertende Spezifikationen der MoSyCo Architektur

Quelle: Eigene Tabelle

Spezifikation	Vari- able	Aussage
Kontinuierliche (Echtzeit-) Überwachung	LK m1	Das System muss die Entwicklung von operativen Daten kontinuierlich überwachen um unplanmäßige Abweichungen unmittelbar zu erkennen.
Kontinuierliche (Echtzeit-) Überwachung	LK m2	Das System muss die operativen Daten in Echtzeit erfassen und analysieren.
Abweichungsanalysen	LK m3	Das System muss Methoden für zeitpunktbezogene Analysen von Abweichungen bereitstellen (bspw. Abweichung zum Vortag).
Abweichungsanalysen	LK m4	Das System muss Methoden für zeitreihenbezogene Analysen von Abweichungen bereitstellen (bspw. Berechnung von Saisonalitäten).
Differenzierte Anpassungsstrategien	LK m5	Das System muss sich selbst an unterschiedliche Problemstellungen des Endanwenders anpassen.
Autonome Adaption	LK m6	Das System muss sich selbst und automatisiert an neue operative Rahmenbedingungen sowie externe Umweltzustände anpassen.
Variable Anpassungsmethoden	LK m7	Das System muss unterschiedliche Methoden für automatisierte Anpassungen bereitstellen (bspw. Prüfen von bestehenden vs. Entdecken von neuen Ursache-Wirkungsbeziehungen).
Evaluation autonomer Adaptionen	LK m8	Das System muss durchgeführte Anpassungen erfassen und selbst beurteilen.
Variable Anpassungsmethoden	LK m9	Das System muss sich bei neuen erforderlichen Anpassungen an alten bereits durchgeführten Anpassungen orientieren.
Differenzierte Anpassungsstrategien	LK m10	Das System muss für unterschiedliche Regeln und Strategien eine automatisierte Anpassung bereitstellen, um dynamisch auf unterschiedliche Probleme reagieren zu können.

5.2 Ergebnisse der Untersuchung

Im Rahmen der ex ante Evaluation werden die Verknüpfungen zwischen den Lösungskomponenten sowie den initialen Problemen, die in Abhängigkeit zu den sechs *Self*-* Eigenschaften stehen, untersucht (vgl. Tabelle 5-1). Die Verteilung der Antworten wird in Abbildung 5-3 in Form eines Boxplot Diagrammes dargestellt.

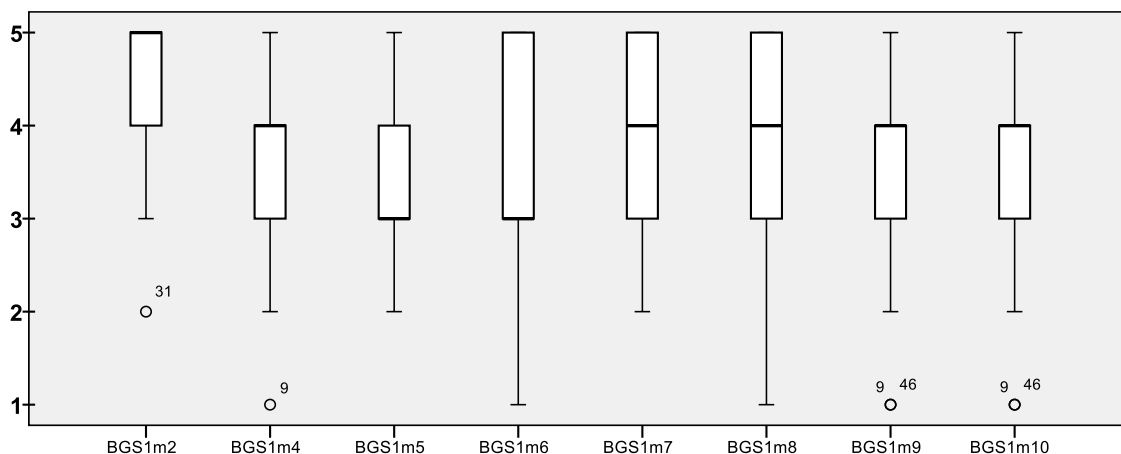


Abbildung 5-3: Boxplot der Ergebnisse zum Problem-Lösung-Fit

Quelle: Eigene Darstellung

Der Abbildung 5-3 ist zu entnehmen, dass alle acht Aussagen mit einem Median von mindestens 3 beurteilt worden sind, wodurch keine Verknüpfung als falsch wahrgenommen wird. Die Spannweite der Antworten ist jedoch bei den beiden Variablen BGS1m6 und BGS1m8 sehr hoch. Die Aussage, dass variable Anpassungsmethoden den Wartungsaufwand reduzieren (BGS1m6) wird im Mittelwert mit 3,39 bewertet, jedoch weisen manche Antworten auch das Minimum von 1 auf. Somit gibt es einzelne Ausreißer, die die Verknüpfung als nichtzutreffend bewerten. Durch die Variabilität von Anpassungsmethoden in Abhängigkeit des Problems soll eine größere Bandbreite an Problemen automatisiert gelöst werden (vgl. Kapitel 4.2.4). Das Abbilden der unterschiedlichen Problemsituationen erfordert initial jedoch einen erhöhten Arbeitsaufwand, weshalb die Variabilität der Methoden auch negativ in Bezug auf den Aufwand bewertet werden kann und somit eine mögliche Ursache für die große Spannweite der Variable BGS1m6 sein kann. Bei der Problembewertung in Kapitel 3.3 beträgt der Median des zugrundeliegenden Problems 3,5 (Variable PL1_PLm4, erhöhter Wartungsaufwand) und ist somit als nicht gravierend beurteilt worden. Da der Wartungsaufwand in der Praxis nicht außerordentlich problematisch eingeschätzt wird, könnte dies ein zusätzlicher Grund für die große Spannweite bei der Bewertung von BGS1m6 sein. Trotz dieser Spannweite kann man dem Quartilsbereich der Variable BGS1m6 entnehmen, dass über alle Antworten zusammengefasst die Problem-Lösung-Verknüpfung als nicht falsch bewertet wird. Die Spannweite der Variable BGS1m8 ist ebenfalls sehr groß (vgl. Abbildung 5-3), jedoch handelt es sich im Antwortbereich zwischen eins und zwei um wenige Ausreißer, da das 95% Konfidenzintervall des Mittelwertes zwischen 3,59 und 4,27 liegt. Der Mittelwert selbst beträgt 3,93. Somit wird bestätigt, dass unterschiedliche Methoden zur Abweichungsanalyse das Problem ungeplanter Entwick-

lungen lösen, indem diese frühzeitig erkannt werden. Der Median von BGS1m5 liegt bei 3 und der Mittelwert bei 3,41. Dass das Problem der Anpassung an geänderte Umwelt- und Rahmenbedingungen (*Self-Configuring*) durch eine autonome Adaption gelöst werden kann, wird somit schwach bestätigt. Die Variable BGS1m2, die durch eine kontinuierliche (Echtzeit-) Überwachung die *Self-Awareness* für die Identifikation von Störungen und Problemen ermöglichen soll, ist stark positiv bewertet worden. Auch die Problem-Lösung-Verknüpfungen der Variablen BGS1m4, BGS1m7, BGS1m9 sowie BGS1m10 sind mit einem Median von 4 als zutreffend bewertet worden. D. h. ein eingeschränktes Wissen über das operative Umfeld (*Context-Awareness*), die fehlende Möglichkeit zur Optimierung der Konfiguration zur Fehlerkorrektur (*Self-Optimizing*), die Einschränkung der Evaluation von vorgenommenen Modellveränderungen (*Self-Awareness*) sowie ein mangelnder Schutz vor Fehlern (*Self-Protecting*) können durch eine kontinuierliche (Echtzeit-) Überwachung, durch differenzierte Anpassungsstrategien, der Evaluation autonomer Adaptationen und mit Hilfe von zeitreihenbezogenen Abweichungsanalysen gelöst werden. Die meisten Problem-Lösung-Verknüpfungen werden als korrekt gewertet (BGS1m2, BGS1m4, BGS1m7, BGS1m8, BGS1m9, BGS1m10), wenige jedoch nur schwach bestätigt (BGS1m5, BGS1m6). In Summe wird von den 50 befragten Teilnehmern jedoch keine Verknüpfung als falsch bewertet, sodass ein Fit zwischen den initial festgestellten Problemen und den spezifizierten Lösungskomponenten festgehalten werden kann.

Auf Basis des Problem-Lösung-Fit wird gemäß dem Untersuchungsdesign die Relevanz des Artefaktes für die unternehmerische Praxis und die Notwendigkeit einer prototypischen Implementierung überprüft. Im Zuge dessen werden die Hypothesen H5-1 sowie H5-2 getestet (vgl. Kapitel 5.1).

Die Hypothese H5-1 besagt, dass je höher der Problem-Lösung-Fit bewertet ist, auch die Nutzungsintention der *MoSyCo* Architektur höher ist. Die zusammengefassten Variablen *FIT* und *INT* sollen durch die Korrelationsanalyse nach Pearson und alternativ nach Spearman überprüft werden. Für die Überprüfung der Hypothese werden drei Antworten aus der Analyse ausgeschlossen, da diese als Ausreißer identifiziert wurden (vgl. Anhang V). Es handelt sich weiter zwar um metrisch skalierte Variablen und die Daten sind linear (vgl. Scatterplot in Anhang V), jedoch sind die Daten gemäß des Shapiro-Wilk-Tests nicht normalverteilt. Somit kann die Korrelationsanalyse nach Pearson nicht durchgeführt werden und die Rangkorrelation nach Spearman wird angewandt. Die Daten werden anhand des

Scatterplot Diagramms (vgl. Anhang V) als monoton identifiziert und die Korrelationsanalyse nach Spearman wird durchgeführt.

Tabelle 5-4: Korrelationsanalyse nach Spearman für FIT und INT

Quelle: Eigene Tabelle

			FIT	INT
Spearman's rho	FIT	Correlation Coefficient	1,000	,311*
		Sig. (2-tailed)	.	,040
		N	45	44
	INT	Correlation Coefficient	,311*	1,000
		Sig. (2-tailed)	,040	.
		N	44	44

*. Correlation is significant at the 0.05 level (2-tailed).

Zur Auswertung werden 44 Antworten (= N) untersucht, um die Spearman Korrelation durchzuführen. Es ergibt sich ein Korrelationskoeffizient von $\rho = 0,311$, bei einem p-Wert von 0,04. Somit ist der Koeffizient statistisch signifikant. Die Nullhypothese kann verworfen und die Alternativhypothese angenommen werden (vgl. Kapitel 5.1). Es besteht eine positive Korrelation zwischen FIT und INT. Somit ist die Nutzungsintention der *MoSyCo* Architektur bei den Umfrageteilnehmern höher, wenn die konzipierten Komponenten der Architektur als geeignet zur Lösung der initialen Probleme (Problem-Lösung-Fit) eingeschätzt werden.

Wie bei der Korrelationsanalyse von Hypothese H5-1, werden zur Überprüfung von H5-2 die Annahmen geprüft. Das Scatterplot Diagramm zeigt eine Tendenz von Linearität (vgl. Anhang VI). Nach dem Sapiro-Wilk-Test sind die Daten nicht normalverteilt. Aufgrund dessen wird wie bei H5-1 die Korrelationsanalyse nach Spearman durchgeführt. Zur Korrelationsanalyse werden 48 Antworten (= N) mit einbezogen.

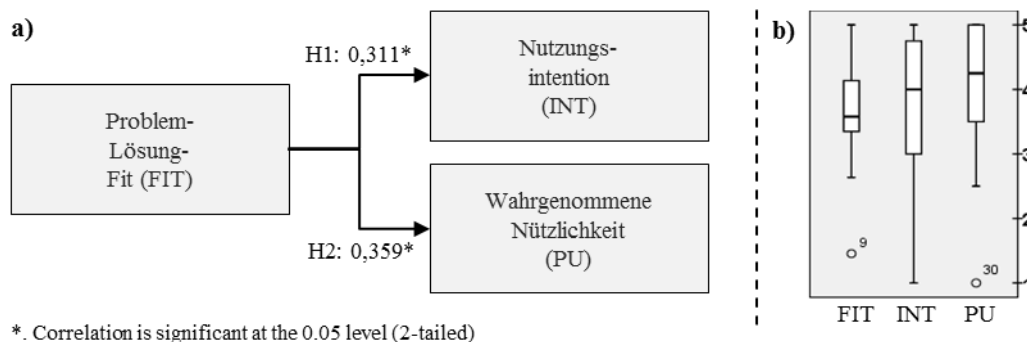
Tabelle 5-5: Korrelationsanalyse nach Spearman für FIT und PU

Quelle: Eigene Tabelle

			FIT	PU
Spearman's rho	FIT	Correlation Coefficient	1,000	,359*
		Sig. (2-tailed)	.	,012
		N	48	48
	PU	Correlation Coefficient	,359*	1,000
		Sig. (2-tailed)	,012	.
		N	48	48

*. Correlation is significant at the 0.05 level (2-tailed).

Der Korrelationskoeffizient beträgt $\rho = 0,359$ bei einem p-Wert von 0,012. Demnach ist der Korrelationskoeffizient mit einer Irrtumswahrscheinlichkeit von 5% statistisch signifikant. Neben der Nutzungsintention ist somit auch die wahrgenommene Nützlichkeit hoch, wenn der Problem-Lösung-Fit bei den Umfrageteilnehmern als hoch bewertet wird.



*. Correlation is significant at the 0.05 level (2-tailed)

Abbildung 5-4: a) Spearman Korrelationskoeffizienten b) Boxplot der Variablen FIT, INT, PU

Quelle: Eigene Darstellung

In der Abbildung 5-4a) werden die Koeffizienten der vorangegangenen Korrelationsanalyse zusammengefasst. Demnach gilt, dass wenn die Umfrageteilnehmer einen hohen Problem-Lösung-Fit wahrnehmen, auch die Nutzungsintention (INT) sowie die wahrgenommene Nützlichkeit (PU) der *MoSyCo* Architektur hoch eingeschätzt wird. Eine genaue Ausrichtung an den initialen Problemen eines Entscheidungsunterstützungssystems in einer dynamischen Unternehmensumwelt hat also eine höhere Nutzungsintention sowie wahrgenommene Nützlichkeit bei den Umfrageteilnehmern zur Folge. Der Boxplot in Abbildung 5-4b) zeigt, dass die wahrgenommene Nützlichkeit deskriptiv betrachtet mit einem Quartilsbereich von 3,5 bis 5 und einem Median von 4,25 sehr hoch ist. Die Nutzungsabsicht (INT) ist auf einem ähnlichen Niveau (Median von 4), aber die Streuung der Antworten ist größer, sodass es auch Teilnehmer gibt, die weniger bereit sind, die *MoSyCo* Architektur in der Praxis einzusetzen. Trotzdem konnte durch die Korrelationsanalyse gezeigt werden, dass die Umfrageteilnehmer, deren Probleme gezielt durch die Funktionen der *MoSyCo* Architektur verbessert werden können, eine hohe Nutzungsintension und eine hohe wahrgenommene Nützlichkeit aufweisen, womit die Relevanz für die Praxis und damit die Rechtfertigung einer prototypischen Implementierung gegeben ist.

Zur Identifikation besonders wichtiger Funktionen bewerten die Umfrageteilnehmer jede Funktion der *MoSyCo* Architektur anhand der fünfstufigen Likert-Skala (vgl. Kapitel 5.1).

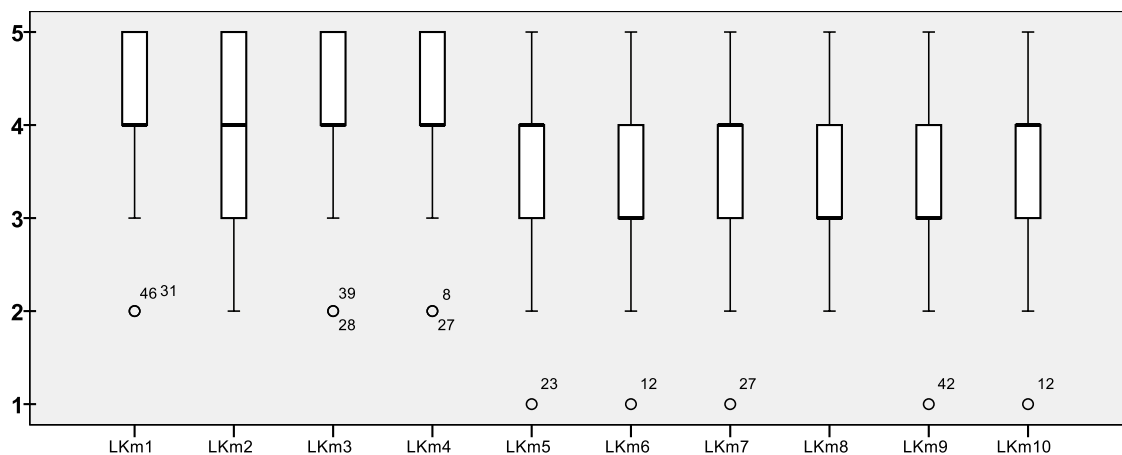


Abbildung 5-5: Boxplot zur Notwendigkeit bestimmter Funktionen der MoSyCo Architektur

Quelle: Eigene Darstellung

Die Ergebnisse werden in Form eines Boxplot in Abbildung 5-5 gezeigt. Ein sofortiges Erkennen von Abweichungen durch eine kontinuierliche Überwachung wird besonders betont (LKm1) und mit den beiden Aussagen zu den Analysemethoden LKm3 und LKm4 mit einem getrimmten Mittelwert von über vier am höchsten bewertet (vgl. Tabelle 5-6). Demnach soll es möglich sein, langfristige Entwicklungen zu analysieren, d. h. neben den zeitpunktabhängigen Methoden (LKm3) werden auch zeitreihenabhängige Methoden (LKm4) gefordert. Die Datenerfassung und -analyse muss bei der kontinuierlichen Überwachung gemäß der großen Spannweite (vgl. Abbildung 5-5) nicht für jeden in Echtzeit erfolgen (LKm2). Die Umfrageteilnehmer schätzen ein dynamisches Systemverhalten, das sich an wechselnde Endbenutzer anpasst (LKm5) und verschiedene Strategien zur Reaktion auf unterschiedliche Probleme verfolgt (LKm10). Beides hat jedoch nicht den höchsten Stellenwert und wird jeweils mit einem getrimmten Mittelwert von 3,62 bzw. 3,65 bewertet. Am wenigsten wird eine autonome Anpassung (LKm6) sowie eine Evaluation durchgeführter Anpassungen (LKm8) erwartet. Die Funktionen sind mit einem getrimmten Mittelwert von 3,33 bzw. 3,32 bewertet (vgl. Tabelle 5-6) und werden seitens der Umfrageteilnehmer im Verhältnis zu allen anderen Funktionen am wenigsten erwartet. Zur robusten automatisierten Modellanpassung werden unterschiedliche Methoden mit einem Median von 4 als hilfreich erachtet (LKm7), eine Orientierung an vergangenen Anpassungen (LKm9) wird mit einem Median von 3 als nicht unbedingt notwendig bewertet (vgl. Abbil-

dung 5-5). Tabelle 5-6 fasst die Antworten zur Notwendigkeit bestimmter Funktionen der *MoSyCo* Architektur zusammen.

Tabelle 5-6: Mittelwerte zur Notwendigkeit bestimmter MoSyCo Funktionen

Quelle: Eigene Tabelle

Spezifikation	Variable	Getrimmter Mittelwert (5%) ²
Kontinuierliche (Echtzeit-) Überwachung	LKm1	4,25
Abweichungsanalysen	LKm4	4,22
Abweichungsanalysen	LKm3	4,16
Kontinuierliche (Echtzeit-) Überwachung	LKm2	3,84
Differenzierte Anpassungsstrategien	LKm10	3,65
Differenzierte Anpassungsstrategien	LKm5	3,62
Variable Anpassungsmethoden	LKm7	3,51
Variable Anpassungsmethoden	LKm9	3,46
Autonome Adaption	LKm6	3,33
Evaluation autonomer Adaptionen	LKm8	3,32

In Tabelle 5-6 fällt auf, dass es den Umfrageteilnehmern wichtig ist, stetig über Fehler sowie Abweichungen informiert zu werden, jedoch eine automatisierte Modellanpassung nicht zwangsläufig darauf folgen muss. Der Hintergrund dessen könnte sein, dass die Frequenz einer modellbasierten Entscheidungsunterstützung in der Praxis geringer ist und deshalb eine sofortige sowie automatisierte Reaktion auf Fehler nicht unbedingt notwendig ist. Da die Umfrageteilnehmer ein modellgetriebenes EUS primär zur Planung der Produktion, der Lagerorganisation sowie der Ressourcenverwaltung (vgl. Anhang VII) einsetzen, handelt es sich zwar um Entscheidungen, die einen kurzfristigen Planungshorizont haben (vgl. Kapitel 2.1), jedoch selten mehrfach, bspw. an einem Tag, geändert werden und somit ein zeitlicher Spielraum für potentielle manuelle Fehlerkorrekturen besteht.

Die ex ante Evaluation der *MoSyCo* Architektur hat gezeigt, dass die entwickelten Komponenten sowie ihre Funktionen die Probleme von Entscheidungsunterstützungssystemen in

² Da der Median in Abbildung 5-5 ausschließlich ganzzahlige Werte annimmt, wird das getrimmte Mittel (5%) betrachtet. Hierzu werden die 5% der größten und kleinsten Stichprobenelemente ignoriert. Das getrimmte Mittel ist robuster als das arithmetische Mittel und orientiert sich an der Berechnungslogik des Median.

einer dynamischen Unternehmensumwelt lindern können und deshalb ein positiver Problem-Lösung-Fit vorhanden ist. Durch die gezielte Verbesserung der ursprünglichen Probleme besteht unter den Umfrageteilnehmern eine erhöhte Nutzungsintention sowie eine hohe wahrgenommene Nützlichkeit, womit die Konzeption der *MoSyCo* Architektur als korrekt bewertet werden kann. Durch die Nutzungsintention sowie die wahrgenommene Nützlichkeit wird die Relevanz für die Unternehmenspraxis aufgezeigt und eine prototypische Umsetzung gerechtfertigt. Im Rahmen der abschließenden Bewertung einzelner Funktionen der *MoSyCo* Architektur konnten Spezifikationen identifiziert werden, die bei der prototypischen Implementierung priorisiert werden.

TEIL III

Prototypische Implementierung
und Evaluation

6 Prototypische Implementierung der MoSyCo Architektur

Nachfolgend wird die prototypische Implementierung der *MoSyCo* Architektur für ein selbst-adaptives modellgetriebenes EUS vorgestellt. Die Implementierung stellt nur eine Variante dar, wie die Architektur umgesetzt werden kann (vgl. Kapitel 4.2). Die Umsetzung ist grundlegend von den verwendeten technischen Umgebungen abhängig, die einleitend in Zusammenhang mit den Komponenten des Prototyps vorgestellt werden (vgl. Kapitel 6.1). Im Folgenden wird die Umsetzung der Komponente des *Model* (vgl. Kapitel 6.2), des *System-Controller* (vgl. Kapitel 6.3) des *Observer* (vgl. Kapitel 6.4) und des *Model-Controller* (vgl. Kapitel 6.5) erläutert. Abschließend wird der ganzheitliche Ablauf bei einer Abweichungs- und Ursachenanalyse bei einer Nutzung des Prototyps beschrieben (vgl. Kapitel 6.6).

6.1 Überblick über die Komponenten des Prototyps

Die Systemlandschaft besteht insgesamt aus den vier Komponenten *Model*, *System-Controller*, *Observer* sowie *Model-Controller* (vgl. Abbildung 6-1). Die einzelnen Komponenten sind in unterschiedlichen Umgebungen entwickelt.

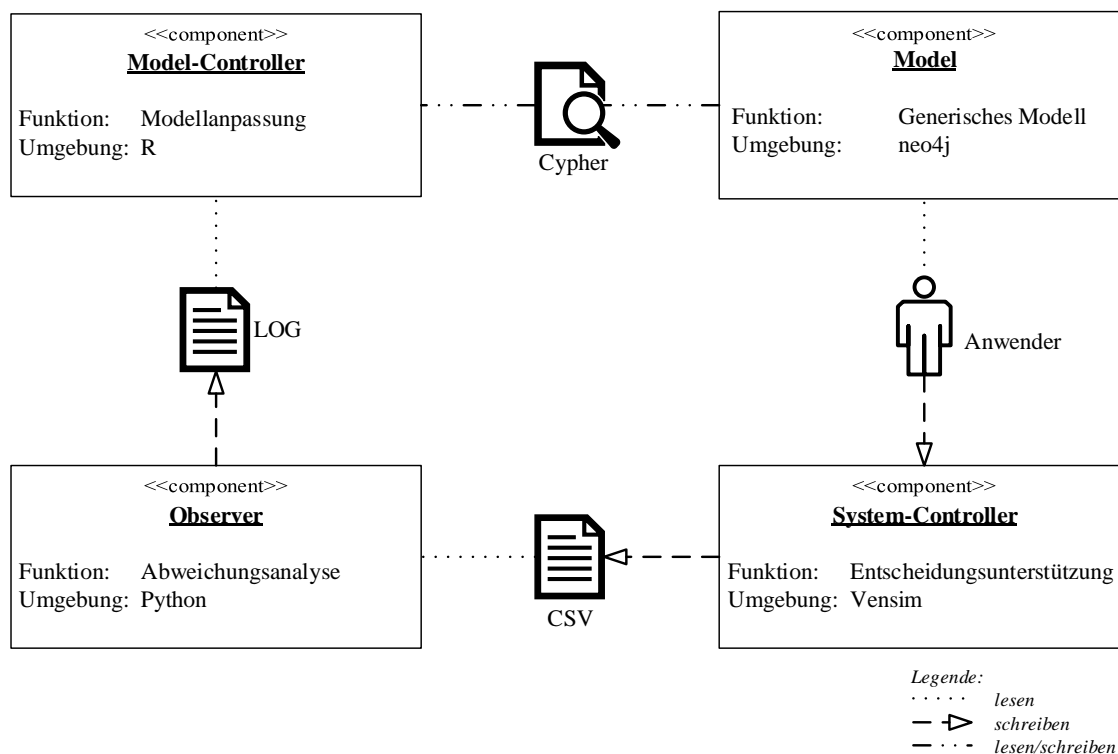


Abbildung 6-1: Übersicht der Komponenten des Prototyps

Quelle: Eigene Darstellung

Eine Entscheidungsunterstützung auf Basis eines Modells wird im *System-Controller* durch die Anwendung Vensim ermöglicht („Vensim Software“ 2018). Vensim unterstützt Analysen, bei denen verschiedene Szenarien in Abhängigkeit von Entscheidungen simuliert werden, um die Auswirkungen einer Entscheidung im Anschluss zu vergleichen („Vensim Software“ 2018). Es wird die sogenannte Personal Learning Edition (PLE) verwendet, die für die Forschung und Lehre frei verfügbar ist. Modelle werden in Vensim in einem proprietären Dateiformat abgespeichert. Der MoSyCo Prototyp arbeitet nicht originär in Vensim, stattdessen werden die Variablen und die Wirkungsstruktur extrahiert.

Das Ziel der Implementierung ist nämlich, einen möglichst generischen Ansatz für modellgetriebene Entscheidungsunterstützungssysteme bereitzustellen (vgl. Kapitel 4.1), weshalb die Komponente *Model* eine semantisch äquivalente Repräsentation des proprietären Modells in einem generischen Dateiformat darstellt. Diese Komponente basiert auf der Open-Source-Graphdatenbank neo4j („The Neo4j Graph Platform“ 2018). Durch die vielseitigen Schnittstellen sowie die funktionale Abfragesprache Cypher bietet neo4j die Möglichkeit, mit geringem Aufwand in unterschiedliche Systemlandschaften integriert zu werden („Data Integration Tools and Platform | Neo4j“ 2018).

Der *Observer* greift auf eine CSV-Datei zu, über die die Zeitreihen aus Vensim übertragen werden. Vensim bildet in dieser Datei jedes verfügbare Simulationsszenario je Variable ab. Der *Observer* ist in der Programmiersprache Python geschrieben und verarbeitet nach einem manuellen Ausführen der Komponente die Zeitreihen aus der CSV-Datei und schreibt die Ergebnisse in eine Logdatei.

Der *Model-Controller* ist in der Programmiersprache R geschrieben und arbeitet zeit- sowie ereignisgesteuert. Zeitgesteuert wird auf die Logdatei des *Observer* zugegriffen und die Daten werden auf neue Abweichungen geprüft. Im Falle einer Abweichung wird ereignisgesteuert eine Ursachenanalyse initiiert. Der *Model-Controller* verfügt über eine Schnittstelle zu neo4j. Über eine REST-API³ können via Cypher⁴ Abfragen getätigt werden. In dem vorliegenden Anwendungsfall können über diese Schnittstelle die Modellstruktur analysiert sowie Veränderungen an Parametern, die durch Eigenschaften von Kanten in der Graphdatenbank repräsentiert werden, oder an der Struktur des repräsentierten

³ Bei einer REST-API handelt es sich um eine standardisierte Schnittstelle, die bei verteilten Systemen eingesetzt werden kann und insbesondere für moderne Webservices genutzt wird (vgl. Massé 2012, S. 5 f.).

⁴ „Cypher ist wie SQL eine deklarative Abfragesprache, aber deutlich mächtiger in Bezug auf die Lesbarkeit und Repräsentation von Graphkonzepten [...]. Cypher selbst ist in Scala implementiert und nutzt die funktionalen Eigenschaften der Sprache [...]“ (Hunger 2014, S. 42).

Modells vorgenommen werden. Der Prototyp ermöglicht eine automatisierte Anpassung des generischen Modells, eine Anpassung des proprietären Modells hat manuell durch den Benutzer zu erfolgen (vgl. Abbildung 6-1). Da eine autonome Anpassung von den potentiellen Anwendern nicht priorisiert wird (vgl. LKm6 in Kapitel 5.2), wird diese Limitation als nicht bedeutend eingestuft. Durch diese Einschränkung arbeitet die Instanziierung der *MoSyCo* Architektur teilautomatisiert.

6.2 Repräsentation des Modells

Das im Rahmen des Prototyps eingesetzte Modell wird im Folgenden im proprietären Format kurz eingeführt, die Entscheidungsmöglichkeiten erläutert, auf die Struktur sowie die Dynamik des Modells eingegangen (vgl. Kapitel 6.2.1) und abschließend das generische, auf neo4j basierende Abbild vorgestellt (vgl. Kapitel 6.2.2).

6.2.1 Modellübersicht im proprietären Format

Das Entscheidungsunterstützungsmodell wurde in Anlehnung an das von Bossel (2004b, S. 24 ff.) entwickelte Modell zur Lagerhaltung und Auftragsplanung erstellt und stellenweise erweitert. Es handelt sich um ein systemdynamisches⁵ Modell, dessen Bestandteile grundlegend in Bestands-, Fluss- sowie Hilfsvariablen aufgeteilt werden können (vgl. Sterman 2000, S. 193). Letztere werden gemäß der Legende von Abbildung 6-2 weiter unterteilt in exogene (Raute) und entscheidungsrelevante (Sechseck) Variablen. Kann eine Hilfsvariablen nicht in diese Klassen eingeordnet werden, wird diese ohne eine Form dargestellt.

⁵ Die Systemdynamik ist eine Methode, die zur ganzheitlichen Analyse komplexer Systeme eingesetzt werden kann und die Analyse von Rückkopplungen über die Zeit ermöglicht. Der Ansatz zeichnet sich durch seine Interdisziplinarität aus, häufig wird er jedoch im betriebswirtschaftlichen Kontext zur modellgetriebenen Unterstützung bei komplexen Problemen in der Realität eingesetzt (vgl. Sterman 2000, S. 4 f.).

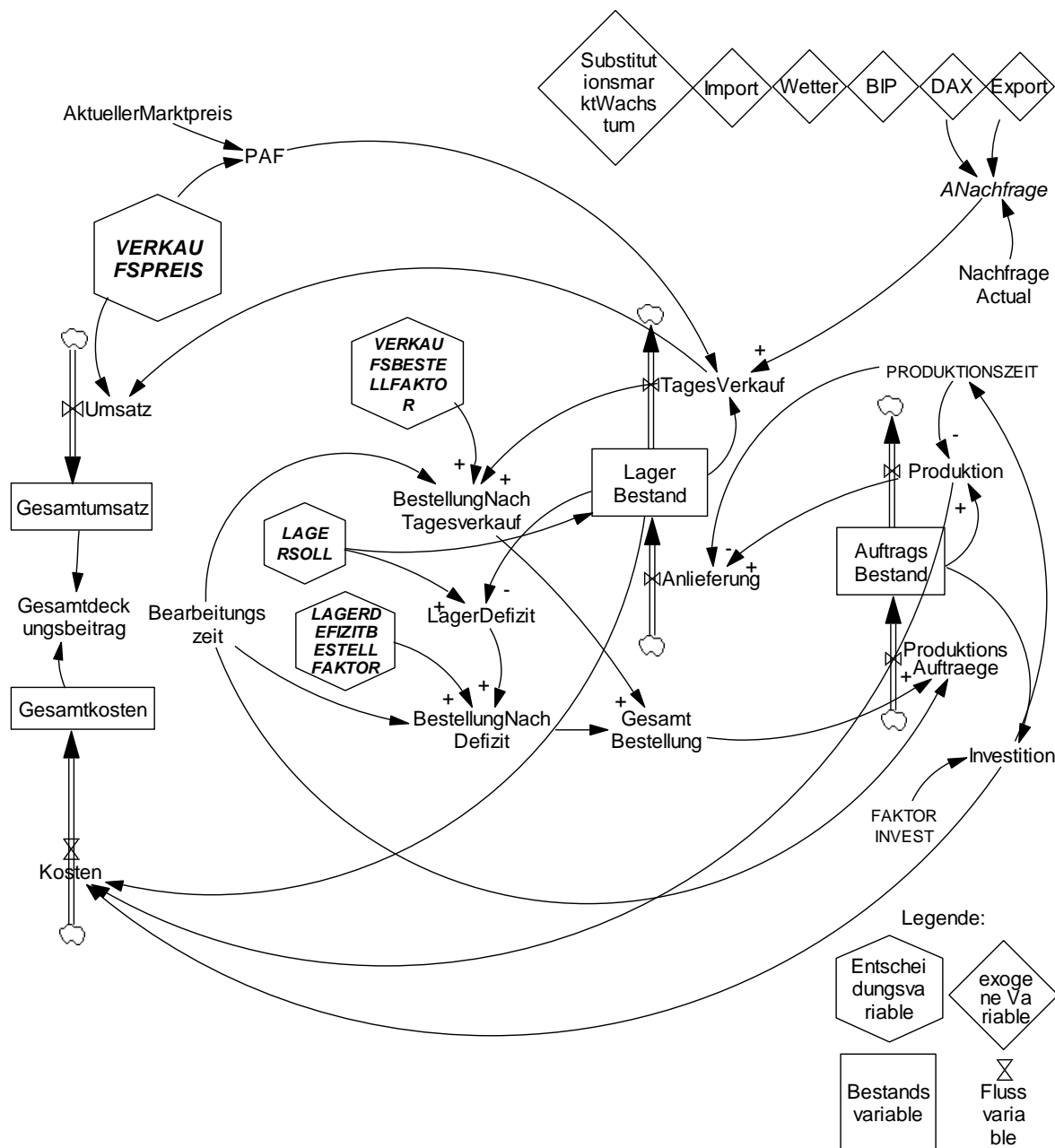


Abbildung 6-2: Systemdynamisches Entscheidungsmodell in Vensim

Quelle: Eigene Darstellung in Anlehnung Bossel (2004b, S. 24 ff.)

Das Modell aus Abbildung 6-2 betrachtet den Verkauf eines Konsumproduktes des täglichen Bedarfs, welcher langfristigen exogenen Einflussfaktoren (Kreisvariablen) unterliegt. Es werden nur Mengen verkauft, die im Lager auch in Bestand sind. Der *Tagesverkauf* verringert den *Lagerbestand* und veranlasst eine *Bestellung* neuer Waren. Um das Lager aufzufüllen, muss eine Bestellung aufgegeben werden, deren Positionen im Anschluss produziert (*Produktion*) und mit einer zeitlichen Verzögerung (*Produktionszeit*) ins Lager geliefert wird. Der *Lagerbestand* muss ausreichend groß sein, um eine eventuelle unerwartete höhere *Nachfrage* abzufangen.

Der Entscheider kann folgende Variablen bestimmen (vgl. Sechseck in Abbildung 6-2):

- *Verkaufspreis* – Zu welchem Preis wird die Ware verkauft?
- *Verkaufsbestellfaktor* – Wieviel Ware soll in Abhängigkeit der Verkaufszahlen nachbestellt werden?
- *Lagersoll* – Wie viel Ware soll im Lager vorgehalten werden?
- *Lagerdefizitbestellfaktor* – Wieviel Ware soll in Abhängigkeit des Lagerdefizits nachbestellt werden?

Verkaufs- und Lagerdefizitbestellfaktor bestimmen in Abhängigkeit von Verkaufs- oder Lagerdefizitzahlen die Warenmenge der Bestellungen. Darüber hinaus kann die Sollmenge im Lager bestimmt werden. Je höher die Sollmenge ist, desto mehr wird im Lager vorgehalten. Da jeder Lagerplatz zusätzliche Kosten verursacht, werden mit einer Steigerung der Sollmenge auch die Gesamtkosten erhöht.

In Abhängigkeit der Höhe des Auftragsbestandes zur Produktion werden Investitionen getätigt. Je mehr investiert wird, desto schneller verläuft die Produktion (Produktionszeit) und neue Ware (Anlieferung) kann somit ebenfalls schneller an das Lager geliefert werden. Die Höhe der Investitionen hängt von dem Auftragsbestand zur Produktion sowie einem Investitionsfaktor ab. Die Investitionen fließen als Investitionskosten ebenfalls in die Gesamtkosten ein.

Die Verkaufszahlen (*TagesVerkauf*) richten sich nach der *Nachfrage* und einer linearen Preisabsatzfunktion (*PAF*). Die *Nachfrage* wird von den beiden exogenen Größen *DAX* und *Export* bestimmt. Die sonstigen exogenen Variablen haben zunächst keinen Einfluss auf das Modell, weshalb von ihnen keine Pfeile ausgehen. Diese freien exogenen Variablen sind jedoch später für die Entwicklung von Modellanpassungsvorschlägen relevant. Die *PAF* hängt von dem aktuellen *Marktpreis* sowie dem eigenen *Verkaufspreis* ab (vgl. Formel 6-1). Der *Verkaufspreis* stellt eine Entscheidungsvariable dar. Je höher der *Verkaufspreis* vom *Marktpreis* prozentual abweicht, desto geringer wird der prozentuale Anteil an der Gesamtnachfrage; die Formel und die Regel zur Verteilung sehen wie folgt aus:

$$PAF = WITH\ LOOKUP \left(\frac{Verkaufspreis - Aktueller\ Marktpreis}{Aktueller\ Marktpreis} \right)$$

Formel 6-1: Berechnung der Preis-Absatz-Funktion in Vensim

Quelle: Eigene Darstellung

Der Operator `WITH LOOKUP` bewirkt, dass der berechnete Wert in der Verteilung aus Abbildung 6-3 nachgeschlagen wird.

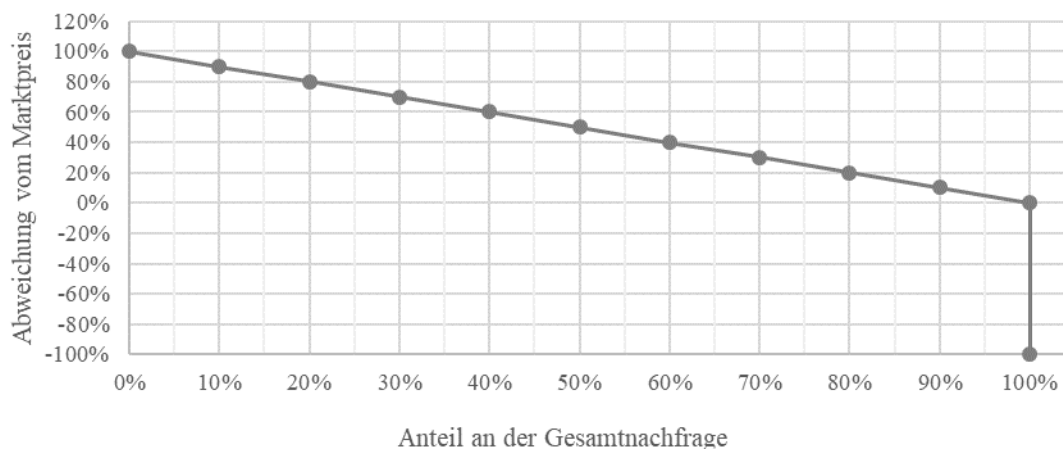


Abbildung 6-3: Anteil der Gesamtnachfrage in Abhängigkeit der Abweichung vom Marktpreis

Quelle: Eigene Darstellung

Anhand des *Verkaufspreises* und der *Verkaufsmenge* wird der *Umsatz* berechnet:

$$\text{Umsatz} = \text{TagesVerkauf} * \text{Verkaufspreis}$$

Formel 6-2: Berechnung des Umsatzes in Vensim

Quelle: Eigene Darstellung

Steigt der *Verkaufspreis*, kann gemäß der Formel 6-2 bei gleichbleibendem *Tagesverkauf* ein höher *Umsatz* erzielt werden. Der *Tagesverkauf* kann jedoch fallen, sofern der *Verkaufspreis* den *Marktpreis* stark übersteigt (vgl. Abbildung 6-3). Die Gesamtkosten und der Gesamtumsatz bestimmen gemeinsam den Gesamtdeckungsbeitrag. Alle Formeln des Entscheidungsmodells sind in Anhang VIII aufgeführt.

6.2.2 Modellübersicht im generischen Format

Die Variablen und die systemdynamische Wirkungsstruktur aus Abbildung 6-2 werden als semantisch äquivalente Repräsentation in neo4j vorgehalten, damit eine generische Variante vorhanden ist. Neo4J ist eine ACID⁶-konforme, in Java implementierte Datenbank, welche einen breiten Funktionsumfang bietet und häufig verwendet wird (vgl. Fasel 2016, S. 123). Neben einer umfangreich dokumentierten Abfragesprache (Cypher) und einer entsprechenden R-API können die Daten komfortabel per REST-Schnittstelle hinzugefügt,

⁶ ACID gewährt die Integrität von Daten in einer Datenbank und ermöglicht den Benutzern ein konfliktfreies Arbeiten. Hierfür muss eine vom Benutzer durchgeführte Transaktion stets atomar, konsistent, isoliert und dauerhaft sein (vgl. Meier 2016, S. 136).

Die Eigenschaften je Kante und Knoten können via Cypher abgefragt werden. Die Informationen können entweder in tabellarischer, textueller oder graphischer Form zurückgegeben werden. In Abbildung 6-5 ist eine Abfrage in Cypher sowie das Resultat dargestellt. Die Eigenschaften der Kante zwischen der *Nachfrage* und dem *Substitutionsmarktwachstum* sind hervorgehoben. Es wird angegeben, wann die letzte Anpassung stattgefunden hat (*AdaptationTime*), dass die Variable multiplikativ wirkt (*Operator*) und dies mit einem negativen Faktor von *-0,0158* (*Parameter*). Die obligatorische *ID* wird immer mit ausgegeben.

The screenshot displays a Cypher query in a text editor and its corresponding graph visualization. The query is as follows:

```
1 MATCH ((d:Variable)-[r:INFLUENCES]- (v:Variable))
2 WITH max(r.AdaptationDate) AS LastEdit
3 MATCH ((d:Variable)-[r:INFLUENCES]- (v:Variable))
4 WHERE r.AdaptationDate = LastEdit
5 RETURN r, v, d
```

The graph visualization shows four nodes: 'Export', 'Substitu...', 'ANachfr...', and 'DAX'. The relationships are:

- 'Export' is connected to 'ANachfr...' via an 'INFLUENCES' relationship.
- 'Substitu...' is connected to 'ANachfr...' via an 'INFLUENCES' relationship.
- 'DAX' is connected to 'ANachfr...' via an 'INFLUENCES' relationship.

The interface also shows a sidebar with icons for Graph, Rows, Text, and Code. At the bottom, the selected relationship is detailed with the following properties:

```
INFLUENCES <id>: 2155 parameter: -0.0158 AdaptationTime: 2005-12-16 operator: *
```

Abbildung 6-5: Cypher Abfrage und Eigenschaften einer Kante

Quelle: Eigene Darstellung

Ein Anwender kann das Modell anhand von Eigenschaft (Property Keys), bestimmten Kanten-typen (Relationship types) oder Typen von Knoten (Node labels) filtern. Im protypi-

schen Stadium werden ausschließlich Knoten des Typs *Variable* und Kanten des Typs *Influences* verwendet. Eine Erweiterungsmöglichkeit bei den Knoten könnten Unternehmensziele oder Anpassungsstrategien sein (vgl. Kapitel 4.2.4).

6.3 Umsetzung des System-Controller

Zur Umsetzung des *System-Controller* wird die Software Vensim genutzt (vgl. Kapitel 6.1). Vensim ist eine Anwendung, die die Methodik der Systemdynamik (engl. System Dynamics) verfolgt (vgl. Sterman 2000, S. 904) und dadurch die Modellierung, Simulation, Analyse und Gestaltung von dynamisch-komplexen Sachverhalten ermöglicht (vgl. Sterman 2000, S. 41). Vensim stellt eine interaktive Softwareumgebung bereit, die die Entwicklung, Exploration, Analyse und Optimierung von Modellen ermöglicht (vgl. Eberlein und Peterson 1992, S. 216). Der Fokus der Anwendung liegt auf der Simulation und der Analyse zukünftiger Zustände (vgl. Eberlein und Peterson 1992, S. 216). Um Entscheidungen und Wirkungen besser nachvollziehen zu können, stellt Vensim Funktionalitäten zur Analyse von Wirkungsketten, die in einer Baumstruktur dargestellt werden, bereit (vgl. Eberlein und Peterson 1992, S. 217). Zur Entscheidungsunterstützung können sogenannte Läufe, die jeweils unterschiedliche Szenarien in Form von Simulationen repräsentieren, verglichen werden. Mit Hilfe dieser Läufe können die Auswirkungen einer veränderten Entscheidung oder Annahme analysiert werden. So kann beispielsweise die Auswirkung einer Erhöhung des *Verkaufspreises* auf den erwarteten *Tagesverkauf* im Modell aus Abbildung 6-2 analysiert werden (vgl. Abbildung 6-6).

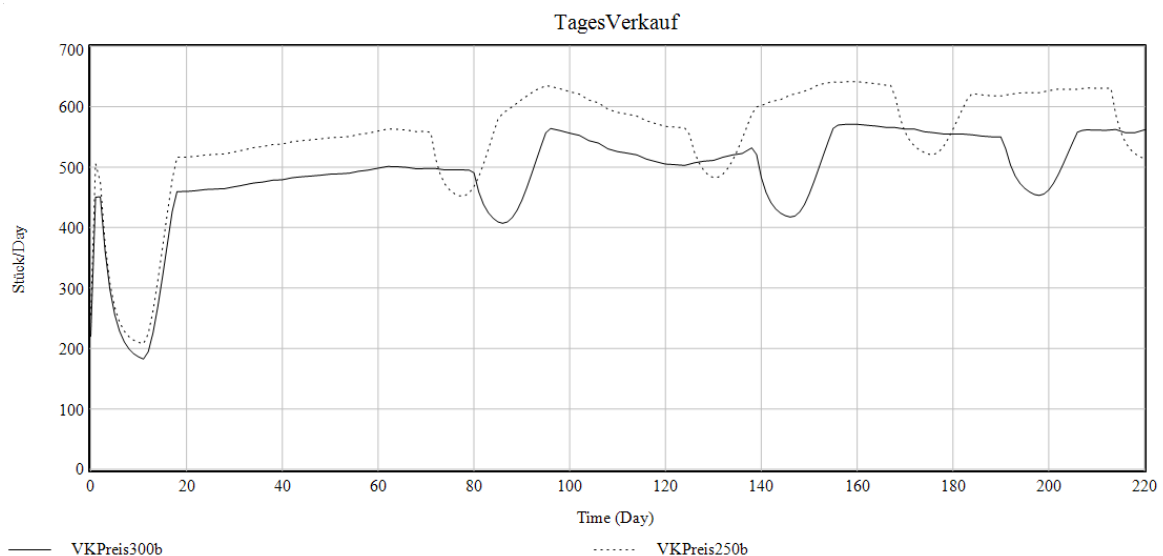


Abbildung 6-6: exemplarische Gegenüberstellung zweier Läufe in Vensim

Quelle: Eigene Darstellung

Die durchgezogene Linie in Abbildung 6-6 repräsentiert ein Szenario des *Tagesverkaufs*, bei dem der *Verkaufspreis* um 50 Geldeinheiten auf 300 erhöht wurde (*VKPreis300b*). Der erwartete *Tagesverkauf* ist mit wenigen Ausnahmen auf einem geringeren Niveau im Vergleich zu der gestrichelten Linie, bei der ein *Verkaufspreis* von 250 Geldeinheiten angenommen wird (*VKPreis250b*).

Über die Funktion der Ursachenanalyse können ausgehend von einer Variable rückwärtsgerichtet die beeinflussenden Wirkungen analysiert werden. In Abbildung 6-7 kann exemplarisch die Wirkungen auf den *Tagesverkauf* analysiert werden. Der Einbruch des *Tagesverkaufs* zu Beginn kann durch die rückwärtsgerichtete Ursachenanalyse auf einen geringen *Lagerbestand* zurückgeführt werden, der im selben Zeitraum einen Tiefpunkt aufweist (vgl. Abbildung 6-7). Mit einem Anstieg des *Lagerbestandes* steigt auch der *Tagesverkauf* wieder (vgl. Abbildung 6-7).

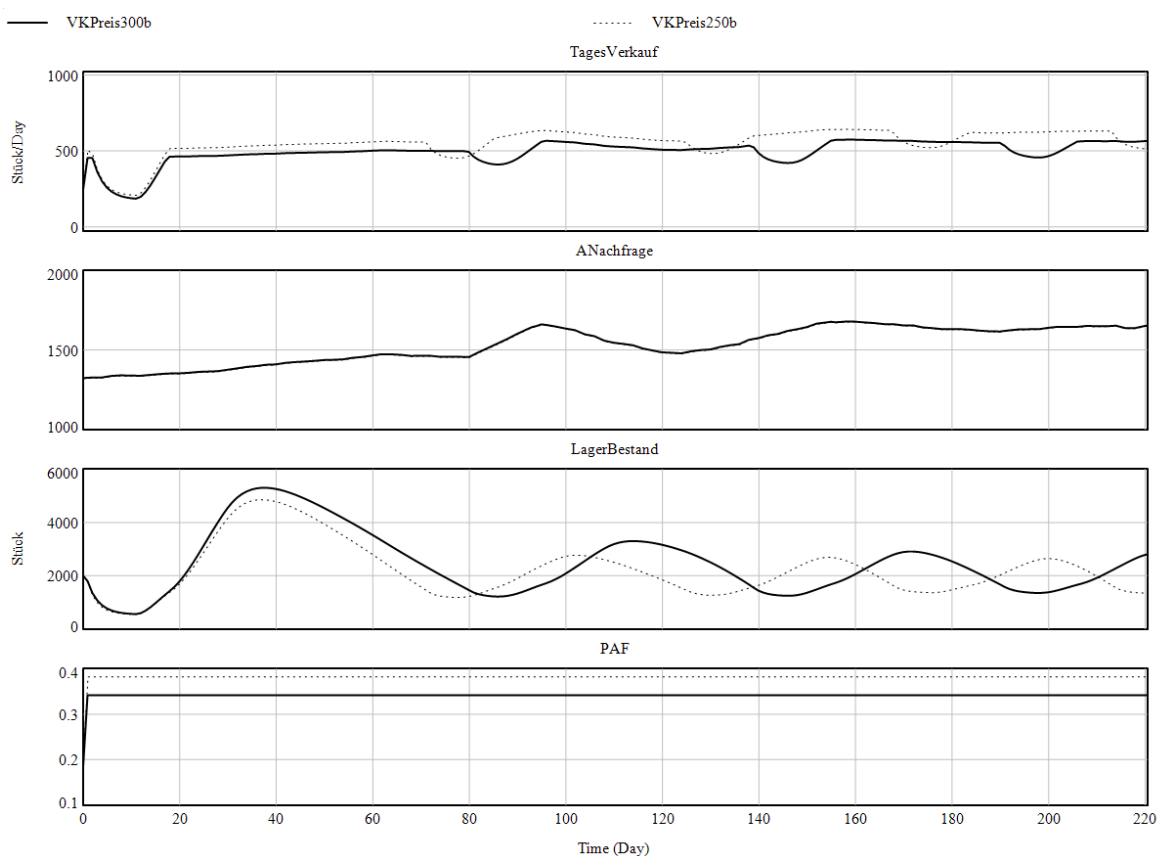


Abbildung 6-7: exemplarische Ursachenanalyse für den Tagesverkauf

Quelle: Eigene Darstellung

Unter anderem durch diese Werkzeuge können Entscheidungen mit Hilfe von Vensim unterstützt werden (vgl. Kapitel 2.2.1). Die Vensim Umgebung repräsentiert den *System-Controller* in der *MoSyCo* Architektur (vgl. Kapitel 4.2.2). Eine automatisierte Umsetzung

einer Entscheidung im operativen System (*SuOC*) wird im prototypischen Stadium nicht vorgenommen. Soll bspw. die Entscheidung über einen *Verkaufspreis* von 300 Geldeinheiten in einem Webshop umgesetzt werden, übernimmt der Anwender diese Aufgabe. Vensim simuliert für jede im Modell enthaltene Variable einen Wert pro Zeitpunkt über den gesamten Planungshorizont, sodass für eine Analyse von Abweichungen die simulationsbasierten Planwerte des Szenarios *VKPreis300b* aus Vensim genutzt werden können (vgl. Abbildung 6-7). Um Programmieraufwand im Rahmen der prototypischen Implementierung zu verringern, ist es zweckdienlich, ein einziges System für die Erzeugung von Plan- und Istwerten einzusetzen und somit die Anzahl an Schnittstellen zu reduzieren.

6.4 Umsetzung des Observer

Der *Observer* ist konzeptionell für die Überwachung des operativen Systems verantwortlich und führt Abweichungsanalysen auf Basis der Plan- und Istwerte durch (vgl. Kapitel 4.2.3). Im prototypischen Stadium greift der *Observer* hierfür auf CSV-Dateien aus Vensim zu und erstellt im Ergebnis eine Log-Datei für den *Model-Controller* (vgl. Abbildung 6-1). Die zur Abweichungsanalyse angewandte Programmiersprache sowie die verwendeten Bibliotheken werden zu Beginn dieses Kapitels erläutert (vgl. Kapitel 6.4.1). Im Anschluss werden die Verzeichnisstruktur und Modulübersicht (vgl. Kapitel 6.4.2) sowie die verschiedenen Ausgabemöglichkeiten (vgl. Kapitel 6.4.3) vorgestellt. Abschließend wird die Umsetzung der wichtigsten Elemente beschrieben: der *Reader* (vgl. Kapitel 6.4.4), der *Inspector* (vgl. Kapitel 6.4.5) und der *Plotter* (vgl. Kapitel 6.4.6).

6.4.1 Angewandte Programmiersprache und verwendete Bibliotheken

Die Komponente des *Observer* wurde in der Programmiersprache Python (Version 3.6) entwickelt. Python ist eine universelle, interpretierte und dynamische Sprache, die sowohl Objektorientierung als auch funktionale Programmierung unterstützt (vgl. Nelli 2018, S. 19 f.). Python ist eine reife und beliebte Programmiersprache, weshalb eine Vielzahl von hilfreichen und stabilen Bibliotheken⁷ zur Verfügung steht. Insbesondere in den Bereichen der Natur- und Ingenieurwissenschaften, sowie der Mathematik ist Python stark vertreten (vgl. Millman und Aivazis 2011, S. 9). Dabei zählen Bibliotheken wie *Pandas*, *Numpy* und *Matplotlib* zu den Standardwerkzeugen und werden neben *PyQt5* sowie *Facebook Prophet* im Rahmen dieser prototypischen Implementierung genutzt.

⁷ Bibliotheken (oder auch Module genannt) erweitern den Funktionsumfang und verbessern damit die Entwicklungsumgebung Python (vgl. Nelli 2018, S. 21).

Pandas (McKinney 2010) liefert dem Benutzer ein- bis mehrdimensionale Datenstrukturen (insbesondere die zweidimensionalen DataFrames) zur effizienten Datenverarbeitung. Diese ermöglichen schnelle Operationen und Berechnungen auf großen Datensätzen. Der *Observer* nutzt diese Datenstrukturen, um Plan- und Ist-Daten effizient zu speichern, aufzubereiten und zu verwalten. Mit *Numpy* (van der Walt et al. 2011) werden DataFrames analysiert und verglichen. *Matplotlib* (Hunter 2007) ist eine Bibliothek für zweidimensionale grafische Darstellungen und wird in der GUI des *Observer* zum Zeichnen von Graphen verwendet. *PyQt5* (Hess und Summerfield 2013) ermöglicht eine leistungsfähige und plattformunabhängige Visualisierung von Daten. *Facebook Prophet* (Taylor und Letham 2018), im Folgenden *Prophet* genannt, ist eine Bibliothek zum Erstellen von Prognosen auf Basis von Zeitreihen. *Prophet* stellt ein flexibel einsetzbares Werkzeug dar, mit dem auch betriebswirtschaftliche Zeitreihen, die komplexe Merkmale aufweisen können, robust prognostiziert werden können (vgl. Taylor und Letham 2018, S. 37 f.). Insbesondere Merkmale wie Trends, Impulse, saisonale Schwankungen und Feiertage kann *Prophet* sehr gut aus einer Zeitreihe extrahieren (vgl. Taylor und Letham 2018, S. 42 f.) und eignet sich deshalb für die Analyse von betriebswirtschaftlichen Zeitreihen.

Mit Hilfe von *Prophet* soll der *Observer* von einem reaktiven System zu einem vorausschauenden System aufgewertet werden (vgl. Kapitel 4.2.3). Somit soll nicht erst auf Abweichungen reagiert werden, wenn das Modell bereits an Präzision verloren hat, sondern bereits im Vorfeld soll eine drohende Abweichung erkannt werden. Ein Modell basiert primär auf Expertenwissen sowie quantitativen Analysen (vgl. Kapitel 2.3) und simuliert zukünftige Entwicklungen. Um vorausschauend handeln zu können, wird zur Abweichungsanalyse neben dem modellgetriebenen auch ein datengetriebener Ansatz verfolgt. Dieser datengetriebene Ansatz wird durch *Prophet* ermöglicht. *Prophet* analysiert die Zeitreihen mit den Ist-Werten aus dem operativen System und erstellt Entwicklungsprognosen, denen die in der Zukunft liegenden simulationsbasierten Planwerte aus dem Modell gegenübergestellt werden können. Die Entwicklungsprognosen von *Prophet* werden im Rahmen der *Observer* Komponente als Indikator für drohende Abweichungen eingesetzt und stellt den Lösungsansatz für ein vorausschauendes System dar.

6.4.2 Verzeichnisstruktur und Modulübersicht

Die Dateiverzeichnisstruktur der *Observer* Komponente wird in Abbildung 6-8 dargestellt. In die *observer.log* auf oberster Ebene protokolliert der Prototyp die Ergebnisse. Der Ordner „data“ enthält die Plan- und Istwerte aus dem *Vensim*. Der Kern der Applikation ist das

Observer Package. Es enthält die verschiedenen entwickelten Module, die die Klassen der Komponenten enthalten.

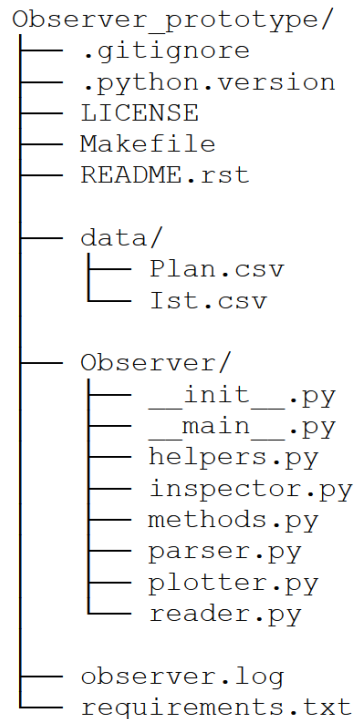


Abbildung 6-8: Dateiverzeichnisstruktur der Observer Komponente

Quelle: Eigene Darstellung

Jedes Element der *Observer*-Komponente aus dem Artefakt (vgl. Kapitel 4.2.3) wurde in einer korrespondierenden Klasse implementiert. Das *Observer*-Modul besteht aus den Dateien „`__init__.py`“ und „`__main__.py`“. In diesen ist die *Observer*-Klasse definiert, welche die übergeordnete Programmlogik repräsentiert. Ausgehend davon werden die wichtigsten Komponenten gestartet: der *Reader*, der *Inspector* und, sofern aktiviert, auch der *Plotter* (vgl. Abbildung 6-9).

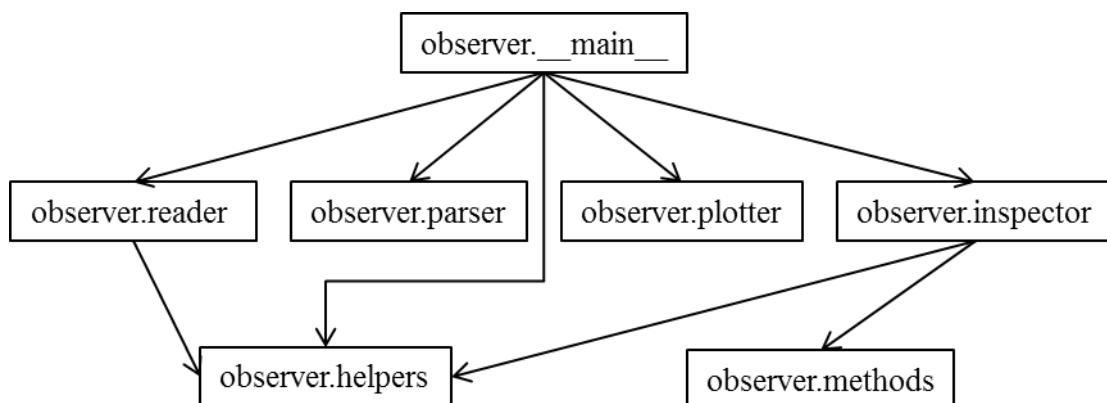


Abbildung 6-9: Modulübersicht der Observer Komponente

Quelle: Eigene Darstellung

Die Klassen dieser Komponenten sind in den gleichnamigen Modulen „*reader.py*“, „*inspector.py*“ und „*plotter.py*“ implementiert. Die *Observer* Komponente wird von der Kommandozeile aus gestartet (vgl. Listing 6-1).

```
python -m mosyco [-h] [-v | -q] [-s SYSTEMS [SYSTEMS ...]]
[-m MODELS [MODELS ...]] [-t THRESHOLD] [--gui] [--logfile]
```

Listing 6-1: Kommandozeilenparameter zum Start der Observer Komponente

Quelle: Eigene Darstellung

Die Optionen, die der Anwender des Programmes hat, sind in eckigen Klammern dargestellt. Tabelle 6-1 erläutert die möglichen Parameter.

Tabelle 6-1: Erläuterung der Kommandozeilenparameter

Quelle: Eigene Tabelle

Parameter	Erläuterung
-h ; --help	Hilfetext anzeigen
-v ; --verbose	Detaillierte Log-Ausgabe aktivieren
-q ; --quiet	Log-Ausgaben unterdrücken
-s ; --systems [Variable 1, Variable 2, ...]	Liste der Variablennamen der Ist-Daten
-m ; --models [Variable 1, Variable 2, ...]	Liste der Variablennamen der Plan-Daten
-t ; --threshold [Schwellwert]	Schwellwert, wann eine Abweichung vorliegt
--gui	GUI-Modus aktivieren
--logfile	Log-Ausgaben in eine Datei schreiben

Der *Parser* (vgl. Abbildung 6-9) ist für das korrekte Einlesen der oben definierten Kommandozeilenparameter (vgl. Tabelle 6-1) verantwortlich. Das Modul „*methods.py*“ stellt die Methoden zur Abweichungsanalyse bereit und „*helpers.py*“ beinhaltet diverse Hilfsfunktionen.

Das Aktivitätsdiagramm in Abbildung 6-10 fasst die Arbeitsweise der *Observer* Komponente zusammen.

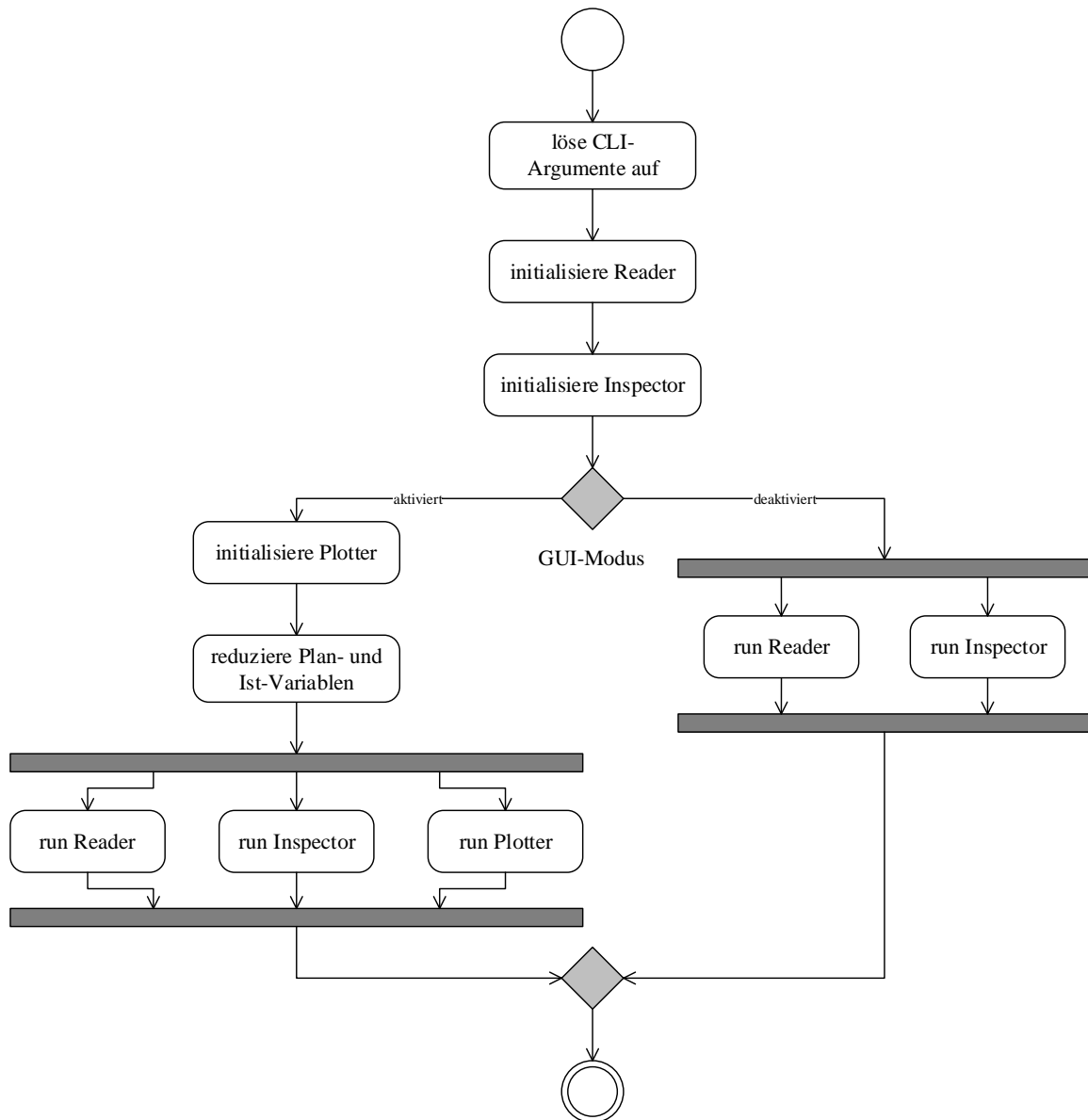


Abbildung 6-10: Observer Aktivitätsdiagramm, High-Level

Quelle: Eigene Darstellung

Zum Start werden die Kommandozeilenparameter (CLI-Argumente) eingelesen und die Komponenten initialisiert. Der *Plotter* wird nur gestartet, wenn der GUI-Modus durch den entsprechenden Parameter aktiviert wird. Der *Reader*, *Inspector* und ggf. auch *Plotter* laufen parallel zueinander. So ist es möglich, dass der *Reader* bereits neue Messpunkte einliest, während der *Inspector* noch Daten analysiert. Gleichmaßen ist es im GUI-Modus möglich, dass der *Plotter* noch Prognosen einzeichnet und der *Inspector* bereits mit der Analyse neuer Daten beginnt.

6.4.3 Ausgabemöglichkeiten

Grundlegend kann der Anwender über zwei Ausgabemöglichkeiten über die Ereignisse des Prototyps informiert werden: über die Kommandozeile sowie die GUI. Über die Kommandozeile erhält der Anwender über das standardmäßige Logging-Modul in Python Informationen über die Arbeitsweise (vgl. Listing 6-2).

```
observer: Starting observer...
observer: Running in DEBUG Mode
observer.reader: Initialized reader...
observer.inspector: Using threshold: ±3.0%
observer.reader: Reader has started sending data to queue...
observer.inspector: Starting Inspector...
```

Listing 6-2: Auszug des Loggings der Observer Komponente

Quelle: Eigene Darstellung

Sofern der *Observer* mit detaillierten Log-Ausgaben (`--verbose`; vgl. Tabelle 6-1) aufgerufen wird, werden jegliche Informationen im Rahmen der Abweichungsmessung protokolliert:

- Der Zeitpunkt, an dem die Abweichung eintritt bzw. potentiell eintreten wird,
- die Namen der betroffenen Variablen,
- den Ist-Wert aus dem operativen System oder der zukünftig prognostizierte Wert,
- den Plan-Wert aus dem Modell sowie
- die absolute und/oder relative Abweichung der Plan- und Istwerte.

Sofern beim Programmstart der Parameter `--logfile` übergeben wird, werden die oben genannten Informationen neben der Kommandozeilenausgabe auch in die Log-Datei geschrieben.

Eine visuelle Darstellung durch eine GUI erleichtert das Verständnis der Arbeitsweise des Prototyps, hat aber auch für die Entwicklung, Evaluation und das Debugging des Prototyps Vorteile. Mittels des Parameters `--gui` beim Programmstart (vgl. Tabelle 6-1) wird die grafische Schnittstelle aktiviert (vgl. Abbildung 6-11).

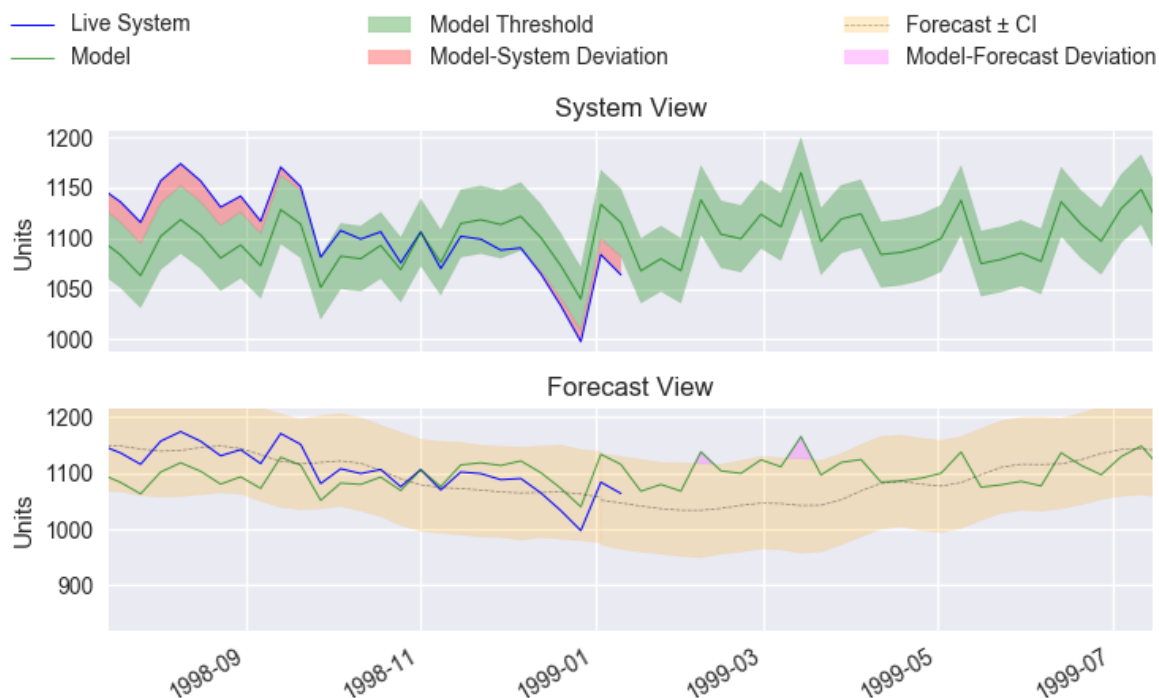


Abbildung 6-11: GUI der Observer Komponente

Quelle: Eigene Darstellung

Zur besseren Übersicht werden im GUI-Modus zwei Graphen angezeigt: Die Systemansicht (engl. System View) und die Prognoseansicht (engl. Forecast View).

Die *Systemansicht* gibt im oberen Diagramm Auskunft über die Abweichungsanalyse zwischen dem zu überwachenden operativen System (*SuOC*, vgl. Kapitel 4.2.1) und dem Modell. Die Werte aus dem *SuOC* werden in Form einer blauen Linie dargestellt und die Modellwerte mit einer grünen Linie. Der Toleranzbereich, in dem Abweichungen vom Modell akzeptiert werden, wird durch die grüne Fläche repräsentiert. Das vertikale Intervall der grünen Fläche wird durch den Schwellwert (vgl. Tabelle 6-1) bestimmt. Abweichungen zwischen den modellbasierten Planwerten und den operativen Istwerten werden durch die rote Fläche hervorgehoben. Dem Nutzer werden die Abweichungen zwischen den Modell- und Ist-Werten so direkt kenntlich gemacht. In Abbildung 6-11 ist ersichtlich, dass zu Beginn des dargestellten Zeitintervalls bis ungefähr zum Zeitpunkt 1998-10 eine Abweichung vorliegt. Ab ungefähr 1998-12 bis zum zuletzt verarbeiteten Zeitpunkt des *Plotter*, der in der Mitte der X-Achse zentriert wird, wird erneut eine Abweichung gemessen.

Die *Prognoseansicht* im unteren Diagramm (vgl. Abbildung 6-11) stellt die Prognose auf Basis von Facebook *Prophet* dar (grau gestrichelte Linie) sowie die aus der Systemansicht bekannten Plan- und Ist-Werte. Das Konfidenzintervall der Prognose ist durch die orangefarbene Fläche um die grau gestrichelte Linie herum dargestellt. Abweichungen zwischen

den Modellwerten und der Prognose durch Facebook *Prophet* werden violett eingefärbt. So sagt die Entwicklungsprognose voraus, dass zu den Zeitpunkten 1999-02 und 1999-04 eine Abweichung auftreten könnte.

Beide Ansichten teilen sich die X-Achse, auf der die Zeit fortgeschrieben wird. Somit kann der Anwender immer eine halbe Periode in die Vergangenheit sowie in die Zukunft blicken. Im Prototyp wird ein Jahr als Periode interpretiert. Die Y-Achsen der beiden Diagramme verändern sich dynamisch in Abhängigkeit der Höhe der zu visualisierenden Werte.

Die System-, Modell- und Prognosedaten werden vor dem Einzeichnen in das jeweilige Diagramm wöchentlich gemittelt. Dadurch wird jeweils eine geglättete Linie dargestellt, die für den Anwender besser zu interpretieren ist. Möchte der Anwender die Graphen genauer analysieren, hat er die Möglichkeit, die fortlaufende Visualisierung neuer Messdaten durch das Drücken der Leertaste zu pausieren. Durch erneutes Drücken der Leertaste wird die Animation wieder fortgeführt. Der Prototyp der *Observer* Komponente arbeitet im GUI-Modus genauso wie im Konsolen-Modus, mit einer Einschränkung: Die Visualisierung im GUI-Modus kann immer nur eine Variable darstellen, da mehrere Variablen zu unübersichtlich sein würden. Sollten beim Programmaufruf (vgl. Tabelle 6-1) trotzdem mehrere Variablen übergeben werden, so wird nur die erste Variable über die GUI visualisiert.

6.4.4 Umsetzung des Reader

Da es sich um einen Prototyp handelt, werden die Daten aus dem operativen System sowie dem Modell über CSV-Dateien eingelesen. In einer zahlungspflichtigen Version der Simulationssoftware Vensim können auch ODBC Schnittstellen genutzt werden, um die Daten ohne eine Zwischenspeicherung zu übertragen. Für den operativen Betrieb einer selbst-adaptiven Systemlandschaft ist eine solche Schnittstelle unabdingbar.

Aufgrund der lokalen Zwischenspeicherung ist der Implementierungsaufwand des *Reader* vergleichsweise gering. Der *Reader* bekommt einen eigenen Thread auf der CPU zugewiesen. Zuerst wird die CSV-Datei im Lesemodus geöffnet und lädt die Inhalte in den Arbeitsspeicher. Im Anschluss werden die Plan- und Ist-Werte aus der CSV-Datei als Key-Value-Paare in Abhängigkeit eines Zeitstempels in eine Warteschlange geschickt. Diese Warteschlange wird *Reader Queue* genannt und verbindet den *Reader* mit dem *Inspector*. Durch den Einsatz der Warteschlange wäre im operativen Betrieb die von den potentiellen An-

wendern priorisierte kontinuierliche Überwachung (vgl. LKm1 in Kapitel 5.2) möglich, da über Schnittstellen Istwerte stetig erfasst werden können und zur weiteren Analyse in der *Queue* vorgehalten werden. Auch wenn die nachgelagerte Analyse aufgrund eines Ressourcenengpasses verzögert ist, können durch den Einsatz der *Reader Queue* die Istwerte aus einem oder mehreren operativen Systemen kontinuierlich erfasst werden.

Sind aus der im Rahmen des Prototyps genutzten CSV-Datei keine neuen Zeilen mehr einzulesen, übermittelt der *Reader* ein None-Objekt, das dem *Inspector* vermittelt, dass keine weiteren Daten mehr kommen. Der Ablauf des *Reader* ist in Abbildung 6-12 zusammengefasst.

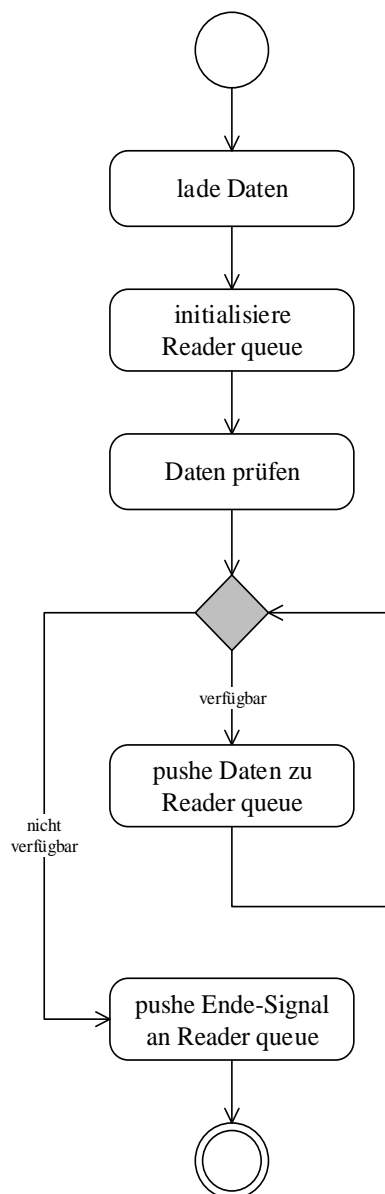


Abbildung 6-12: Observer Aktivitätsdiagramm, Reader Element

Quelle: Eigene Darstellung

6.4.5 Umsetzung des Inspector

Wie auch der *Reader* wird dem *Inspector* ein separater Thread zugewiesen und initialisiert. Es werden die Datenstrukturen vorbereitet, in denen die Plan-, Ist- sowie Prognosewerte gespeichert werden. Der *Inspector* hat Zugriff auf den gesamten Zeitverlauf, damit jegliche Analysevarianten ermöglicht werden.

Der *Inspector* ruft in einer Schleife so lange neue Daten aus der *Reader Queue* ab, bis dieser das `None`-Objekt übermittelt bekommt und daraufhin keine neuen Daten mehr folgen (vgl. Abbildung 6-13). Jeden neu eingetroffenen Ist-Wert vergleicht der *Inspector* mit den entsprechenden Plan-Werten aus dem Modell. Dazu greift dieser auf die Methodenbank für Abweichungsanalysen zurück (vgl. Abbildung 4-1), die im Prototyp durch das Modul „`methods.py`“ repräsentiert wird. Standardmäßig wird die prozentuale Abweichung zwischen den zwei Werten berechnet. Überschreitet die Abweichung den Schwellwert, gibt der *Inspector* über das Logging-Modul eine entsprechende Meldung aus. Diese Art der Abweichungsanalyse deckt die von den potentiellen Anwendern priorisierte zeitpunktabhängige Methode ab (vgl. LK_{m3} in Kapitel 5.2).

Am Ende einer definierbaren Periode ruft der *Inspector* die *Prophet* Bibliothek aus der Methodenbank auf, um eine Prognose für die Folgeperiode zu erstellen. Hierzu wird jedes Mal ein neues *Prophet*-Objekt mit den zu analysierenden historischen Istwerten instanziiert. Die Analyse dieser Zeitreihe deckt den gewünschten zeitreihenabhängigen Ansatz ab (vgl. LK_{m4} in Kapitel 5.2). Die *Prophet*-Objekte werden als Forecaster bezeichnet. Mit einem Forecaster wird eine neue Vorhersage erstellt, die auf der Zeitreihe der Ist-Werte basiert. Aufgrund dessen handelt es sich um eine reine Entwicklungsprognose (vgl. Kapitel 2.1). Da im Prototyp ein Jahr als Periode interpretiert wird (vgl. Kapitel 6.4.3), wird somit jährlich eine neue Entwicklungsprognose erstellt. Das Intervall ist jedoch konfigurierbar. Die Prognosewerte werden in einem separaten DataFrame gespeichert. Das Konfidenzintervall der Vorhersagen liegt auf dem Niveau von 95%. *Prophet* weist die Trend- sowie Saisonkomponente der Prognose separat aus, damit diese Informationen für weitere Analysen genutzt werden können. Nachdem die Prognose erstellt wurde, werden diese mit den Plan-Werten aus dem Modell verglichen. Dazu wird mit Hilfe von *Pandas* geprüft, in welchen Zeiträumen die Plan-Werte des Modells innerhalb des Konfidenzintervalls der Prognose liegen. Liegt ein Plan-Wert außerhalb des Konfidenzintervalls der Prognose, wird wieder über das Logging-Modul eine entsprechende Ausgabe getätigt. Der *Inspector* arbeitet so lange, bis keine neuen Ist-Daten mehr verfügbar sind.

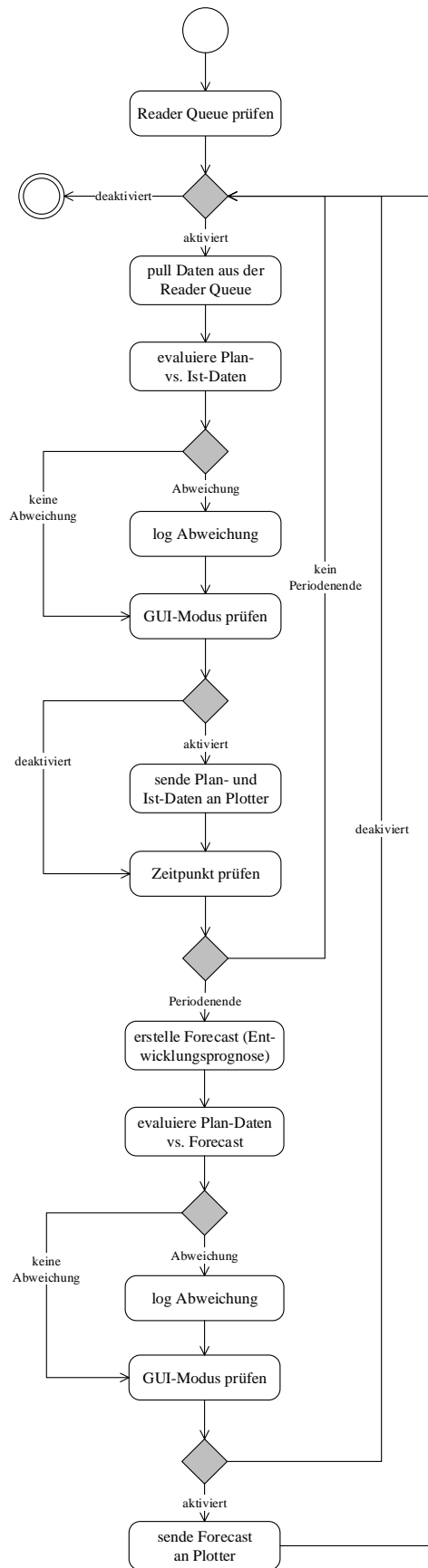


Abbildung 6-13: Observer Aktivitätsdiagramm, Inspector

Quelle: Eigene Darstellung

6.4.6 Umsetzung des Plotter

Das *Plotter*-Modul ist für die grafische Darstellung verantwortlich und wird deshalb nur initialisiert, wenn der Prototyp im GUI-Modus gestartet wird. Mit aktiviertem GUI-Modus ändert sich der Ablauf der *Observer* Komponente geringfügig: Da die Programmschleife der grafischen Darstellung im Hauptthread arbeiten muss, wird im GUI-Modus erst der *Plotter* initialisiert, bevor der *Reader* und der *Inspector* gestartet werden (vgl. Abbildung 6-10). Bei der Initialisierung werden das Hauptfenster, die Legende und die Achsen der Graphen vorbereitet. Darüber hinaus wird die *Plotter Queue* initialisiert, die eine Kommunikation mit dem *Inspector* ermöglicht. Im Anschluss werden ausgehend vom *Plotter* der *Reader* und der *Inspector* in einem separaten Thread gestartet. Im Rahmen der Abweichungsanalyse leitet der *Inspector* die relevanten Daten über die Ist- und Planwerte über die *Plotter Queue* an den *Plotter* weiter (vgl. Abbildung 6-14).

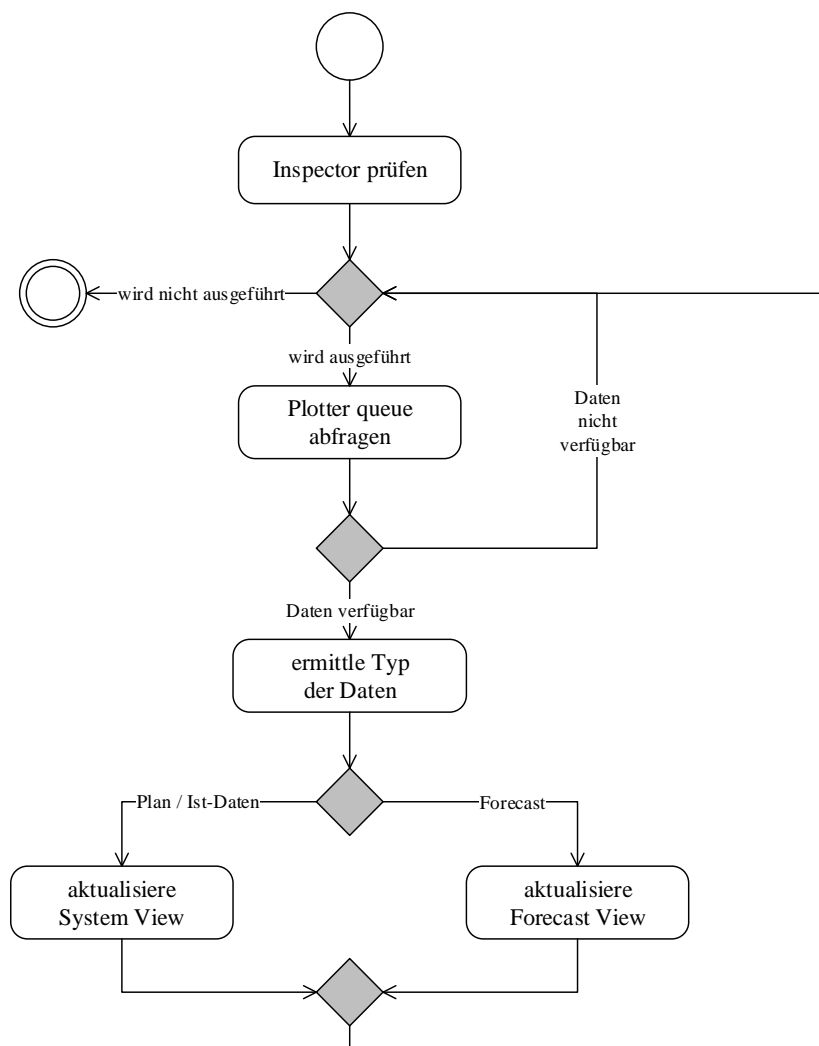


Abbildung 6-14: Observer Aktivitätsdiagramm, Plotter

Quelle: Eigene Darstellung

Der *Plotter* ruft die Daten aus der Warteschlange ab und prüft die Art des Objektes (vgl. Abbildung 6-14). Wenn das Objekt eine Liste von Ist-Daten enthält, aktualisiert dieser damit die Graphen in der Systemansicht (vgl. Kapitel 6.4.3). Wenn das Objekt ein DataFrame ist, der Prognosewerte enthält, dann wird eine neue Prognose in der Prognosesicht eingezeichnet.

6.5 Umsetzung des Model-Controller

Mit Blick auf die Umsetzung des *Model-Controller* werden zu Beginn die angewandte Programmiersprache sowie die verwendeten Pakete erläutert (vgl. Kapitel 6.5.1). Im Anschluss wird jeweils die Umsetzung der Prüfung auf neue Abweichungen (vgl. Kapitel 6.5.2), der Ursachenanalyse (vgl. Kapitel 6.5.3), der Evaluation von Modellanpassungsvorschlägen (vgl. Kapitel 6.5.4) sowie der Durchführung von Modellanpassungen (vgl. Kapitel 6.5.5) beschrieben.

6.5.1 Angewandte Programmiersprache und verwendete Pakete

Der *Model-Controller* wurde mit der Programmiersprache R entwickelt. Bei R handelt es sich um eine Programmiersprache, die gemeinsam mit Python zu den beliebtesten und weit verbreiteten Sprachen gehört (vgl. Sauer 2019b, S. 15). Insbesondere die Anzahl an Paketen, die eine Erweiterung des Funktionsumfangs von R darstellen, ist sehr hoch und die Anzahl der Neuerscheinungen pro Monat steigt exponentiell (vgl. Sauer 2019b, S. 15). R erweist sich insbesondere für statistische Datenanalysen als geeignet (vgl. Sauer 2019b, S. 14 f.) und ist deshalb zur Entwicklung von Modellanpassungsvorschlägen anwendbar. Bei R handelt es sich um einen Interpreter, bei dem einzelne Zeilen ausgeführt werden können (vgl. Sauer 2019b, S. 13). Da die Auswahl von Methoden zur Anpassung eines betriebswirtschaftlichen Modells aufgrund der dynamischen Unternehmensumwelt explorativ ist, erweist sich eine interpretierte Programmiersprache mit umfangreichen Erweiterungsmöglichkeiten für das *Prototyping* als vorteilhaft, da ein Austausch von Methoden vereinfacht möglich ist. Es wird ein experimentelles *Prototyping* verfolgt, bei dem die Alternativen der informationstechnischen Realisierung fokussiert werden (vgl. Leimeister und Stahlknecht 2015, S. 269). Für ein effizientes experimentelles *Prototyping* ist es praktisch, wenn durch den Interpreter einzelne Zeilen ausgeführt werden können.

Für das Ziel der Modellanpassung wird im Rahmen des Prototyps auf mehrere Pakete zurückgegriffen. Das *caret* Paket bietet eine Vielzahl an Methoden zur Modellbildung, bspw.

die lineare Regression (vgl. Sauer 2019b, S. 16). Es werden auch Evaluationsmethoden wie die Kreuzvalidierung (vgl. Kuhn 2008, S. 6) bereitgestellt. Das Paket *BSTS* (Bayesian Structural Time Series) kann ebenfalls zur Modellbildung genutzt werden und setzt maschinelles Lernen ein, um eine Merkmalsauswahl, die Ableitung kausaler Wirkungen und Zeitreihenprognosen zu ermöglichen (vgl. Scott und Varian 2014, S. 4 ff.). Zum effizienten Arbeiten mit DataFrames in R wird *tidyverse* genutzt, welches *dplyr* zur Aufbereitung der Daten nutzt und *ggplot2* zur Visualisierung der Daten (vgl. Sauer 2019a, S. 76). Da es sich bei der Datengrundlage um Zeitreihen handelt, wird zusätzlich das Paket *zoo* genutzt, das ähnlich wie *tidyverse* auf eine effiziente Arbeitsweise mit dem Fokus auf Zeitreihen ausgelegt ist. *Zoo* ermöglicht es, Berechnungen durchzuführen, Unregelmäßigkeiten in der Zeitreihe zu identifizieren, aber auch mit Matrizen oder Faktoren umzugehen (vgl. Zeileis und Grothendieck 2005, S. 2). Das R-*Tcl/Tk* (Tool Command Language) wird verwendet, um eine regelmäßige, zeitgesteuerte Überprüfung auf neue Abweichungen durchzuführen (vgl. Dalgaard 2001, S. 28). Die Schnittstelle zwischen R und dem generischen Modell in *neo4j* wird durch das Paket *RNeo4j* bereitgestellt (White 2016).

6.5.2 Zeitgesteuerte Prüfung neuer Abweichungen

Der Programmcode des *Model-Controller* muss initial einmal ausgeführt werden. Die Auswertung der Log-Daten des *Observer* (vgl. Kapitel 4.2.4) wird im Anschluss wiederholt zeitgesteuert vorgenommen. Das Aktivitätsdiagramm zu diesem Vorgang ist in Abbildung 6-15 dargestellt und fasst darüber hinaus die Arbeitsweise des *Model-Controller* zusammen.

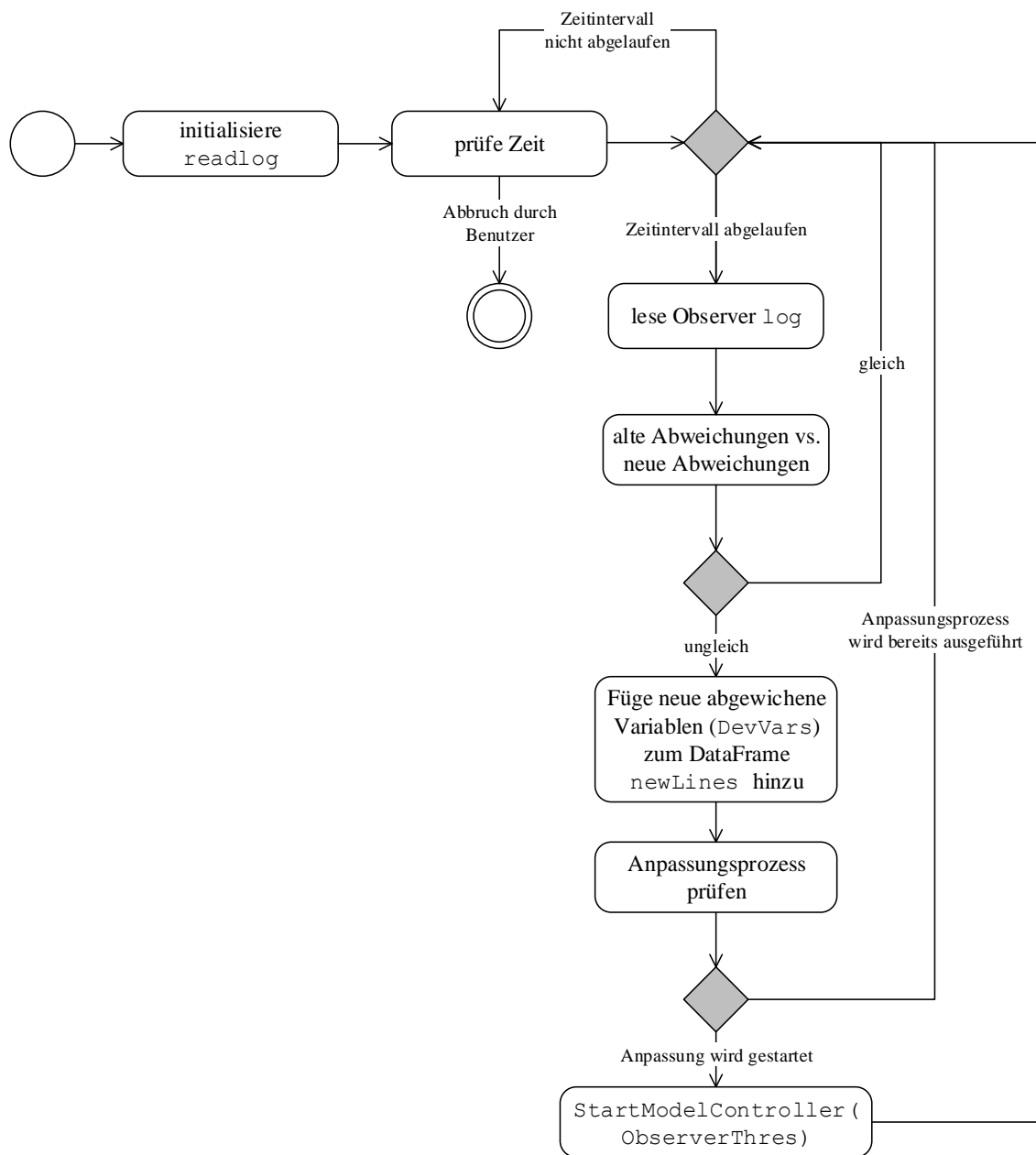


Abbildung 6-15: Model-Controller Aktivitätsdiagramm, readlog Funktion

Quelle: Eigene Darstellung

Nach einer Initialisierung erfolgt ein zeitgesteuertes Einlesen der Logdatei des *Observer*. Standardmäßig geschieht dies alle 10 Sekunden und wird durch das Paket Tcl/Tk ermöglicht (vgl. Kapitel 6.5.1), die `readlog` wiederholt ausführt. Die Inhalte der Logdatei sind protokollierte Abweichungen zwischen Plan- und Istwerten. Nach dem Einlesen wird geprüft, ob im Vergleich zur vorangegangenen Einleseprozess neue Abweichungen hinzugefügt wurden. Ist dies der Fall, werden die jeweiligen abgewichenen Variablen (`DevVars`) zum `DataFrame newLines` hinzugefügt. Die Liste dient als zentrales Objekt für alle Funk-

tionen des *Model-Controller* und enthält Informationen über den Abweichungszeitpunkt als Datum, den Variablennamen als Zeichenkette, einer logischen Variable zum Status, ob eine Abweichung gemessen wurde, den Ist- sowie Planwert, die absolute sowie relative Abweichung (jeweils als Gleitkommazahl) und über eine boolesche bzw. logische Variable zum Status der Anpassung (vgl. Tabelle 6-2). Dieser Status gibt Auskunft darüber, ob eine Anpassung bereits stattgefunden hat und ob diese erfolgreich war.

Tabelle 6-2: Spaltennamen des DataFrame `newLines` inkl. Variablentypen

Quelle: Eigene Tabelle

TIME	VARIABLE	DEVIATED	ACTUAL	PLAN	ABS_DEV	REL_DEV	ADAPTED
<date>	<character>	<logical>	<double>	<double>	<double>	<double>	<logical>

Sofern eine neue Abweichung vorhanden ist und demnach die Prüfung der Ungleichheit zwischen den aktuellen Abweichungen und denen der vorangegangenen Version `TRUE` ist, wird die Abweichung dem DataFrame `newLines` hinzugefügt. Sofern keine Modellanpassung stattfindet, also der Status von `BSTsisRunning` `FALSE` ist, wird durch das Ereignis der Abweichung die Funktion `StartModelController` ausgeführt und der Parameter `ObserverThres` übergeben, der über den relativen Grenzwert zur Bewertung einer Abweichung informiert. Da es sich um eine selbst-adaptive Systemlandschaft handelt, läuft die zeitgesteuerte Schleife zum Einlesen der `Observer Log` dauerhaft, bis diese vom Benutzer abgebrochen wird.

6.5.3 Ereignisgesteuerte Ursachenanalyse

Die Funktion `StartModelController` wird ereignisgesteuert bei einer Abweichung ausgeführt, wenn eine Anpassungsaufforderung besteht. Die Namen der abgewichenen Variablen (`DevVars`) werden in diesem Fall der Liste `DevVarList` hinzugefügt (vgl. Abbildung 6-16). Durch die zeitgesteuerte Prüfung auf Abweichungen sowie die ereignisgesteuerte Ursachenanalyse ist die Komponente des *Model-Controller* für den operativen Einsatz in einer selbst-adaptiven Systemlandschaft geeignet.

Für die Variablen aus der Liste `DevVarList` werden jeweils die unabhängigen Variablen, d. h. die beeinflussenden Variablen, in `neo4j` identifiziert. Für diesen Arbeitsschritt wird das Paket *RNeo4j* eingesetzt (vgl. Kapitel 6.5.1). Im Anschluss werden sowohl für die abhängigen als auch die unabhängigen Variablen die Istwerte bis zum Entscheidungszeitpunkt

gefiltert. Mit diesen Zeitreihen (`mydata`), dem relativen Grenzwert zur Bewertung einer Abweichung (`ObserverThres`) und der Information, welche Variablen für eine Strukturanpassung in Betracht kommen sollen (`VarFilter`), wird die Ursachenanalyse und damit `ApplyBSTS` gestartet (vgl. Abbildung 6-16).

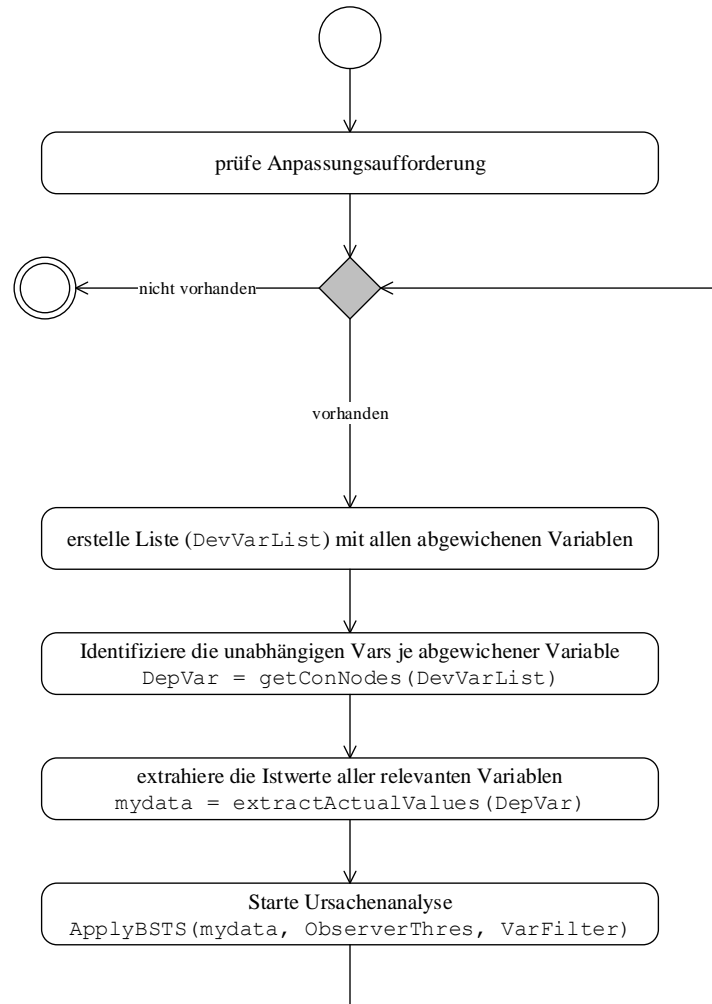


Abbildung 6-16: Model-Controller Aktivitätsdiagramm, `StartModelController` Funktion

Quelle: Eigene Darstellung

Die Funktion `ApplyBSTS` ist für die eigentliche Anpassung des Modells verantwortlich und erstellt unterschiedliche Anpassungsvorschläge. Daraus resultieren unterschiedliche Modelle, die in Hinblick auf ihre Güte evaluiert werden, um das am besten geeignete Modell in `neo4j` umzusetzen. Das Aktivitätsdiagramm der Funktion `ApplyBSTS` ist in Abbildung 6-17 zu sehen. Die Funktion arbeitet ebenfalls ereignisgesteuert in Abhängigkeit der Funktion `StartModelController`. Es folgt eine Schleife über alle abgewichenen Variablen (`DevVarList`).

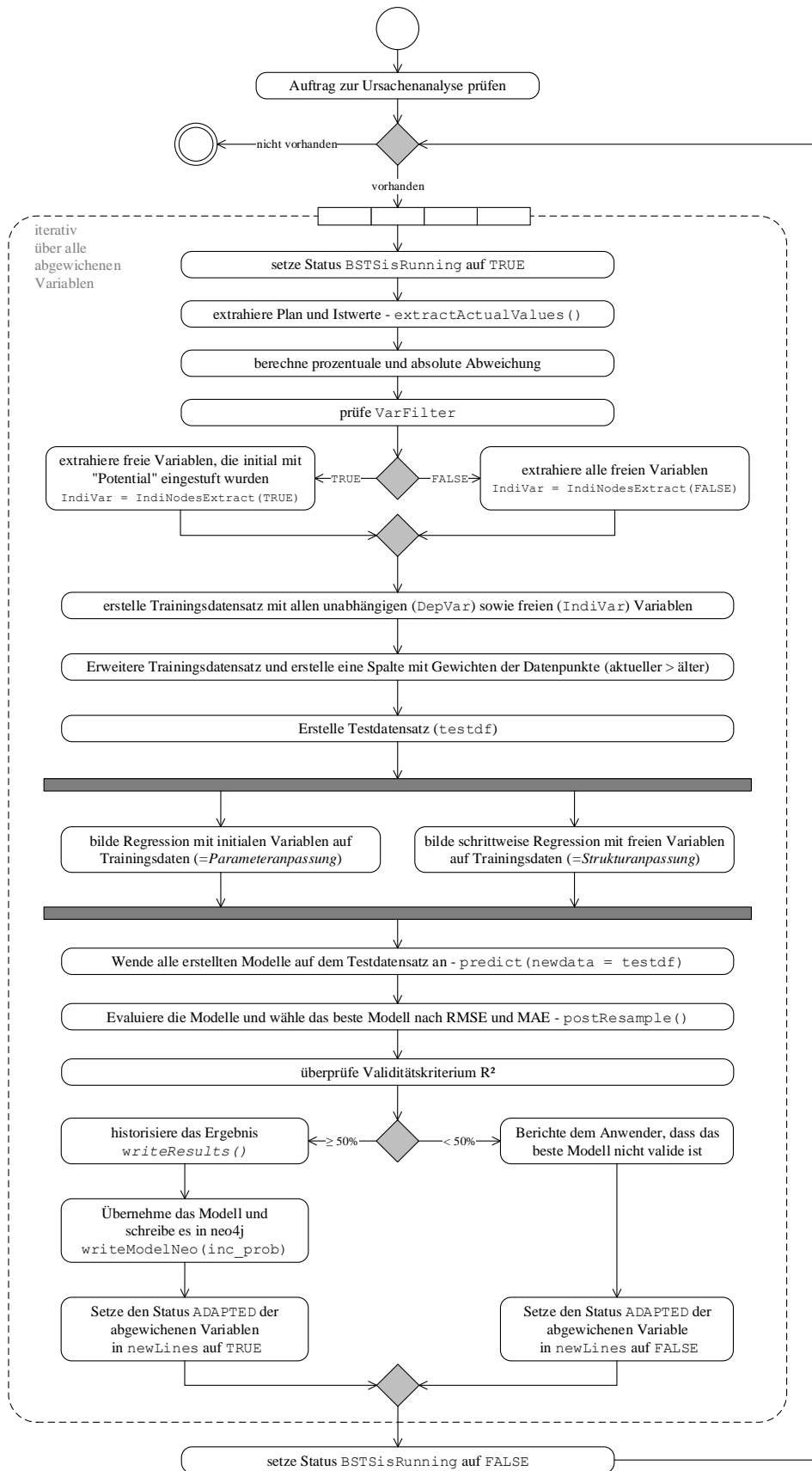


Abbildung 6-17: Model-Controller Aktivitätsdiagramm, ApplyBSTS Funktion

Quelle: Eigene Darstellung

Zu Beginn einer jeden Anpassung wird `BSTsisRunning` auf `TRUE` gesetzt, um eine erneute Ausführung der Funktion zu verhindern (vgl. Abbildung 6-15). Es werden die Plan- sowie Istwerte je abhängiger Variable gefiltert, um anschließend die prozentuale sowie absolute Abweichung zu berechnen. Anschließend werden die Variablen identifiziert, die für eine Strukturanpassung in Betracht gezogen werden.

Das Argument `VarFilter` bestimmt den Prozess für Strukturanpassungen. Bei `FALSE` werden alle frei verfügbaren Variablen für eine schrittweise Regression eingesetzt. Es handelt sich um Variablen, die bis zu diesem Zeitpunkt noch keinen Einfluss im Modell haben, aber deren Daten verfügbar sind (vgl. freie Variablen in Abbildung 6-17). Sofern `VarFilter` mit `TRUE` übergeben wird, werden ausschließlich sogenannte „potential Variables“ (vgl. Kapitel 4.2.4) für Strukturanpassungen berücksichtigt, die eine Teilmenge der frei verfügbaren Variablen sind. Hierbei handelt es sich um Variablen, die zum Zeitpunkt der Modellerstellung keinen Einfluss haben, jedoch bei zukünftigen Modellanpassungen eine Rolle spielen könnten und relevanter als die anderen freien Variablen sind. Diesen Status bestimmt der Modell-Verantwortliche. Die in Betracht kommenden Variablen für Strukturanpassungen werden in dem Objekt `IndiVar` abgespeichert. Nach der Identifikation der Variablen, die bei einer Strukturanpassung berücksichtigt werden, wird ein Trainingsdatensatz mit Zeitreihen auf Basis des `zoo` Pakets erstellt (vgl. Kapitel 6.5.1). Dieser beinhaltet die vollständigen Zeitreihen jener freien Variablen für Strukturanpassungen sowie der abhängigen und unabhängigen Variablen. Dieser Datensatz wird auf die Zeitpunkte gefiltert, bei denen die relative Abweichung der abhängigen Variable oberhalb eines definierten Grenzwertes liegt. Standardmäßig beträgt dieser Wert 2,5%. Das Filtern wird durch die gleichnamige Funktion aus dem `tidyverse` Paket ermöglicht (vgl. Kapitel 6.5.1). Im Anschluss wird ein neues Attribut des Trainingsdatensatzes generiert, das zur Gewichtung der Datenpunkte verwendet wird. Aktuellere Datenpunkte sollen ein höheres Gewicht haben als ältere. Diesem liegt die Prämisse zugrunde, dass sich eine Störung aufgrund der dynamischen Unternehmensumwelt (vgl. Kapitel 2.3) über die Zeit verändern kann und dadurch die aktuelleren Zeitpunkte einen höheren Erklärungsbeitrag im Rahmen einer Modellanpassung haben können. Darüber hinaus kann der Trainingsdatensatz wenig Datenpunkte enthalten, sofern eine geringe Anzahl an Abweichungen gemessen wurde. Unter diesen Rahmenbedingungen ist eine Gewichtung der Datenpunkte für eine anschließende Regression vorteilhaft, da die Güte einer Schätzung insbesondere bei einem kleinen Datensatz durch eine Gewichtung verbessert werden kann (vgl. Huber et al. 2007, S. 10). Durch

eine Gewichtung der Datenpunkte werden Fallgewichte übergeben, die eine Wahrscheinlichkeit zur Berücksichtigung dieser Punkte beim Schätzen einer neuen Formel repräsentieren (vgl. Elliott 2009, S. 1). Im Rahmen des Prototyps wird zur Gewichtung folgende Exponentialfunktion verwendet:

$$Weights = K * \exp\left(-\frac{P * Anpassungszeitpunkt}{Beobachtungszeitpunkt}\right)$$

Formel 6-3: Formel zur Gewichtung der Datenpunkte

Quelle: Eigene Darstellung

Unter dem Anpassungszeitpunkt wird der letzte Zeitpunkt des Trainingsdatensatzes verstanden, der den Zeitpunkt, zu dem die Anpassung durchgeführt wird, repräsentiert. Der Beobachtungszeitpunkt ist dynamisch und entspricht dem Zeitpunkt der Beobachtung. Abbildung 6-18 präsentiert ein Beispiel mit 16 Datenpunkten im Trainingsdatensatz. In diesem Fall ist der Anpassungszeitpunkt 16 und die Beobachtungszeitpunkte erstrecken sich von 1 bis 16. Über den Parameter P lässt sich steuern, wie viel stärker der Anpassungszeitpunkt (16) im Verhältnis zu den Beobachtungszeitpunkten (1-16) gewichtet wird. Je höher P gewählt wird, desto steiler verläuft die Exponentialfunktion in Annäherung an den Anpassungszeitpunkt. Über den Parameter K lässt sich die absolute Höhe der Gewichte bestimmen, das Verhältnis zwischen den einzelnen Zeitpunkten bleibt unverändert. Standardmäßig ist der Parameter K = 10 und P = 4.

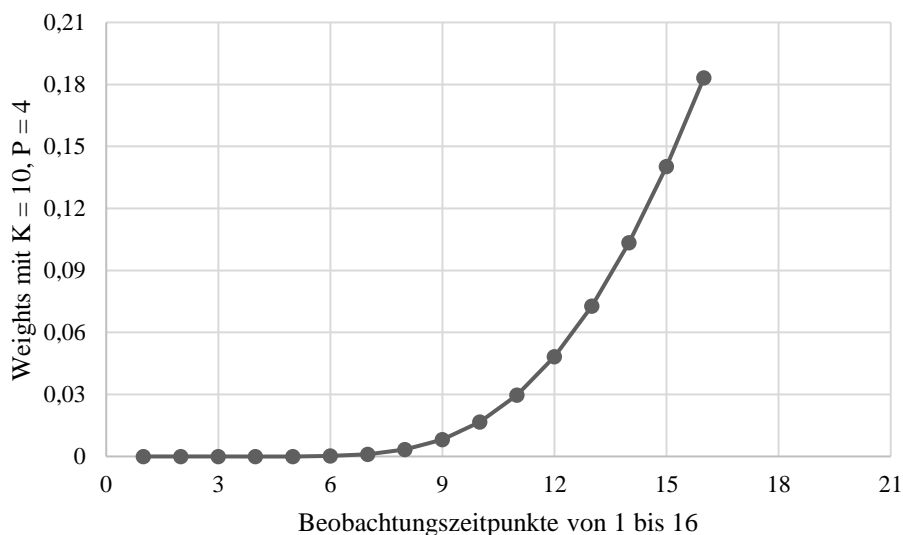


Abbildung 6-18: Exponentielle Funktion der Gewichtung

Quelle: Eigene Darstellung

Mit der Gewichtung der Datenpunkte ist der Trainingsdatensatz finalisiert und im Anschluss wird ein Testdatensatz erstellt (vgl. Abbildung 6-17). Dieser ist notwendig, um nicht nur die Anpassungsgüte des Modells bestimmen zu können, sondern auch die Prognosegüte (vgl. Küsters 2012, S. 428f). Insbesondere bei mehreren konkurrierenden Modellen wird dieser Ansatz empfohlen, um die Prognosequalität retrospektiv anhand eines unabhängigen Datensatzes zu beurteilen (vgl. Küsters 2012, S. 428f). Die Zeitreihe des Testdatensatzes beginnt mit dem Folgezeitpunkt des Anpassungszeitpunktes. Die Länge des Testdatensatzes hängt von der Anzahl verfügbarer Istwerte nach dem Anpassungszeitpunkt ab. Die Struktur des Testdatensatzes ist bis auf die Gewichtung identisch mit der des Trainingsdatensatzes.

6.5.4 Methodik zur Entwicklung und Evaluation von Modellanpassungsvorschlägen

Nachdem der Test- und Trainingsdatensatz erstellt wurde, werden die Anpassungsvorschläge erstellt. Hierfür wird eine lineare Regression zur Schätzung der Formeln verwendet (vgl. Abbildung 6-17). Es werden mehrere Anpassungsvorschläge und damit Regressionsgleichungen parallel erstellt, die in Parameter- und Strukturanpassungen separiert werden können. Mit diesem Schritt werden unterschiedliche Hypothesen eines Ursache-Wirkungs-Modells erstellt, die anschließend in einer linearen Regressionsbeziehung bestimmt werden (vgl. Backhaus et al. 2018b, S. 62). Zur Anpassung könnten auch generalisierte lineare Modelle (GLM) genutzt werden oder Modelle auf Basis von *BSTS* (vgl. Kapitel 6.5.1). Insbesondere das von Google entwickelte *BSTS* überzeugt durch robuste Prognosen, es ist jedoch auf Absatzmärkte und die Wirkung von Marketingkampagnen spezialisiert (vgl. Brodersen et al. 2015). Für Entscheidungsmodelle dieser Domäne wird der Einsatz von *BSTS* empfohlen. Aufgrund des breiten Anwendungsspektrums der linearen Regression wird diese Anpassungsmethode im Rahmen des Prototyps bevorzugt genutzt. Darüber hinaus fehlt *BSTS* die von der MoSyCo Architektur geforderte Möglichkeit, einem Datenpunkt eine Gewichtung zuzuweisen (vgl. Kapitel 6.5.3).

Bei der Parameteranpassung bleiben die initialen abhängigen und unabhängigen Variablen für das Ursache-Wirkungs-Modell bestehen. Bei einer Strukturanpassung werden die freien Variablen (Ergebnis nach `VarFilter`) verwendet, um unterschiedliche Ursache-Wirkungs-Hypothesen, sogenannte Experimentalmodelle, zu erstellen. Die Menge an Experimentalmodellen hängt von der Anzahl freier Variablen ab. Es bestehen 2^n Kombinationsmöglichkeiten, wobei n die Anzahl freier Variablen repräsentiert. Gibt es beispielsweise zwei initial

unabhängige Variablen (A & B) und zwei freie Variablen (C & D), ergeben sich insgesamt $2^n = 4$ Experimentalmodelle (vgl. Tabelle 6-3).

Tabelle 6-3: Mögliche Experimentalmodelle (A, B: unabhängige Variablen; C, D: freie Variablen)

Quelle: Eigene Tabelle

Anpassungsvariante	Experimentalmodelle
Parameteranpassung	$y = A + B$
Strukturanpassung	$y = A + B + C$
Strukturanpassung	$y = A + B + D$
Strukturanpassung	$y = A + B + C + D$

Nachdem die verschiedenen Experimentalmodelle formuliert sind, muss geschätzt werden, wie stark eine oder mehrere unabhängige Variablen auf die abhängige und zu schätzende Variable wirken (vgl. Backhaus et al. 2018b, S. 63ff).

$$\widehat{y}_n = \beta_0 + \beta_1 x_{1n} + \beta_2 x_{2n} + \dots + \beta_j x_{jn}$$

Formel 6-4: Multiple lineare Regressionsgleichung

Quelle: In Anlehnung an Backhaus et al. (2018b, S. 72)

In Formel 6-4 ist \widehat{y}_n die zu schätzende Variable zum Zeitpunkt n. Diese berechnet sich aus einer Konstante β_0 sowie dem Produkt aus den Regressionskoeffizienten β_j und den Werten der multiplen unabhängigen Variablen x_j zum Zeitpunkt n. Die Parameter β_0 bis β_j werden so bestimmt, dass die Summe der quadrierten Residuen minimal ist und somit eine optimale Regressionsgleichung bestimmt wird (vgl. Backhaus et al. 2018b, S. 70). Diese Art der Regressionsschätzung wird auch Methode der kleinsten Quadrate genannt.

$$\varepsilon = \sum_{n=1}^N e_n^2 = \sum_{n=1}^N (y_n - \widehat{y}_n)^2 \rightarrow \min$$

Formel 6-5: Fehlerterm einer linearen Regressionsgleichung

Quelle: In Anlehnung an Backhaus et al. (2018b, S. 70)

Der in Formel 6-5 dargestellte Fehlerterm ε entspricht der Summe aller quadrierten Werte der Residualgrößen e_n . Die Residuen berechnen sich aus dem Beobachtungswert der abhängigen Variable Y zum Zeitpunkt n (y_n) und dem ermittelten Schätzwert, der auf Basis der Regressionsgleichung (\widehat{y}) zum selben Zeitpunkt bestimmt wird.

Einflussgrößen, die nicht in der vorgegebenen Regressionsgleichung enthalten sind, aber einen Erklärungsbeitrag hätten, erhöhen die Residuen (vgl. Backhaus et al. 2018b, S. 69). Die Methode der kleinsten Quadrate arbeitet unter der Prämisse, dass jeder Datenpunkt denselben Erklärungsbeitrag für das Schätzen der Regressionsgeraden hat (vgl. Kuhlenkasper und Handl 2017, S. 223), d. h. die Datenpunkte sind alle gleichgewichtet. Bei einer gewichteten Regression kann bestimmten Datenpunkten ein höheres Gewicht zugewiesen werden, sofern diese bspw. einen höheren Erklärungsbeitrag zur Regressionsschätzung haben bzw. dadurch die Residuen verringert werden (vgl. Strutz 2016, S. 48). Die Zielfunktion der Kleinsten-Quadrate-Methode wird hierfür wie in Formel 6-6 dargestellt um ein Gewicht w für jeden Zeitpunkt n erweitert (vgl. Strutz 2016, S. 26).

$$\varepsilon = \sum_{n=1}^N e_n^2 = \sum_{n=1}^N \mathbf{w}_n * (y_n - \widehat{y}_n)^2 \rightarrow \min$$

Formel 6-6: Fehlerterm einer gewichteten linearen Regressionsgleichung

Quelle: In Anlehnung an Strutz (2016, S. 26)

Bei der im Vorfeld definierten Zielfunktion zur Regressionsschätzung handelt es sich um einen Sonderfall der gewichteten Regression, bei dem alle Datenpunkte gleichgewichtet sind und w_n stets 1 ist. Mit Hilfe der Gewichtung können Verzerrungen beim Schätzen der Parameter bei kleinen Stichproben und ungleichen Wahrscheinlichkeiten zur Berücksichtigung von Datenpunkten verhindert werden (vgl. Elliott 2009, S. 1).

Die Schätzungen der Regressionsgleichungen finden auf Basis des Trainingsdatensatzes statt. Es wird eine gewichtete Regression auf Basis des dafür vorgesehenen Attributes im Trainingsdatensatz durchgeführt. Jede geschätzte Regressionsgleichung wird angewandt (`predict`) und zur Berechnung der abhängigen Variable genutzt (vgl. Abbildung 6-17). Dies geschieht auf Basis des Testdatensatzes.

Im Anschluss erfolgt die Evaluation der Anpassungsvorschläge und Gütekriterien wie R^2 , RMSE und MAE werden ermittelt. Hierfür wird die Funktion `postResample` (vgl. Abbildung 6-17) des `caret` Paketes verwendet (vgl. Kapitel 6.5.1).

Das Bestimmtheitsmaß R^2 gibt Auskunft darüber, wie gut die Regressionsgleichung den Zusammenhang zwischen unabhängiger und abhängiger Variable wiedergibt, indem die statistisch erklärte Varianz ins Verhältnis zur beobachteten Varianz gesetzt wird (vgl. Urban und Mayerl 2018, S. 54).

$$R^2 = \frac{\sum_{n=1}^N (\hat{Y}_n - \bar{y})^2}{\sum_{n=1}^N (Y_n - \bar{y})^2}$$

Formel 6-7: Bestimmtheitsmaß R^2 einer linearen Regression

Quelle: In Anlehnung an Urban und Mayerl (2018, S. 70)

In Formel 6-7 beschreibt der Zähler die Distanz zwischen dem geschätzten Wert \hat{y} sowie dem Mittelwert \bar{y} und der Nenner die Distanz zwischen dem empirisch gemessenen Wert y_n und dem Mittelwert (vgl. Urban und Mayerl 2018, S. 51). Da die Varianz für das gesamte Regressionsmodell beschrieben werden soll, wird die Summe der Abweichungsquadrate betrachtet (vgl. Urban und Mayerl 2018, S. 52). Das Gütemaß R^2 liegt zwischen 0 und 1, wobei ein Wert von 1 bedeutet, dass die Beobachtungswerte aus dem Testdatensatz (y) exakt mit den Schätzwerten \hat{y} übereinstimmen (vgl. Urban und Mayerl 2018, S. 54). Demnach ist ein höheres R^2 erstrebenswert.

Im Vergleich zum Bestimmtheitsmaß wird beim Root Mean Square Error (RMSE; auf Deutsch: Wurzel der mittleren Fehlerquadratsumme) nicht die Modellgüte beurteilt, sondern die Prognosegüte (vgl. Küsters 2012, S. 431f). Die Kennzahl gibt an, wie stark die prognostizierten Werte (\hat{y}_n) im Durchschnitt über die Anzahl an Beobachtungen (N) von den tatsächlichen Beobachtungswerten (y_n) aus dem Testdatensatz abweichen (vgl. Küsters 2012, S. 434). Wie der Formel 6-8 zu entnehmen ist, berechnet sich der RMSE aus der Quadratwurzel des durchschnittlichen Prognosefehlers, um positive wie auch negative Abweichungen zu erfassen. Je höher der RMSE ist, desto schlechter ist die Prognosegüte des Modells.

$$RMSE = \sqrt{\frac{\sum_{n=1}^N (\hat{Y}_n - y_n)^2}{N}}$$

Formel 6-8: Wurzel der mittleren Fehlerquadratsumme einer linearen Regression

Quelle: In Anlehnung an Küsters (2012, S. 432 ff.)

Der RMSE kann zu einer überproportionalen Bewertung von Ausreißern neigen (vgl. Küsters 2012, S. 434). Im Vergleich dazu bewertet der Mean Absolute Error (MAE, auf Deutsch: mittlerer absoluter Fehler) jede Abweichung zur Beurteilung der Güte eines Prognosemodells proportional (vgl. Küsters 2012, S. 434). Im Vergleich zum RMSE wird gemäß Formel 6-9 die absolute Abweichung bewertet. Je höher der Wert des MAE ist, desto schlechter ist die Prognosegüte des Modells.

$$MAE = \frac{1}{N} \sum_{t=1}^N |\hat{y}_t - y_t|$$

Formel 6-9: Mittlerer absoluter Fehler einer linearen Regression

Quelle: In Anlehnung an Küsters (2012, S. 434)

Im Rahmen der `ApplyBSTS` Funktion (vgl. Abbildung 6-17) werden die jeweiligen Modelle nach dem RMSE bewertet, es wird das Modell mit dem geringsten Prognosefehler gewählt. Als ergänzendes Kriterium könnte der proportional bewertende MAE berücksichtigt werden und mit dem Ergebnis des überproportional bewertenden RMSE verglichen werden (vgl. Küsters 2012, S. 434). Eine Regel für den kombinierten Einsatz des MAE und RMSE ist im aktuellen Entwicklungsstadium nicht hinterlegt, da es weitere Exploration zur Abwägung der beiden Maße bedarf. Die Prognosegüte wird im Prototyp in erster Instanz anhand des RMSE bewertet. In einer zweiten Instanz wird das Bestimmtheitsmaß R^2 geprüft, dieses muss oberhalb eines definierbaren Grenzwertes liegen. Standardmäßig wird ein Wert von 0,5 erwartet, sodass durch das Modell mehr als 50% der Varianz erklärt werden soll (vgl. Abbildung 6-17). Durch das Bestimmtheitsmaß wird die Anpassungsgüte (innerhalb der Zeitreihe zur Regressionsschätzung) und durch den RMSE die Prognosegüte (außerhalb der Zeitreihe zur Regressionsschätzung) evaluiert. Somit werden zwei unterschiedliche Evaluationsansätze verfolgt, um eine möglichst hohe empirische Evidenz zu erreichen (vgl. Küsters 2012, S. 427f). Auf Grundlage der aufgeführten Gütekriterien werden die Experimentalmodelle, die die verschiedenen Modellanpassungsmöglichkeiten beinhalten, gegeneinander evaluiert und das beste Modell wird ausgewählt.

Sollte dieses Modell ein zu geringes Bestimmtheitsmaß aufweisen, wird der Anwender darüber informiert, dass keines der Modelle als valide eingestuft wurde (vgl. Abbildung 6-17). In diesem Fall wird der Status `ADAPTED` in dem DataFrame `newLines` (vgl. Tabelle 6-2) auf `FALSE` gesetzt, da für die Abweichung zu diesem Zeitpunkt kein valider Modellanpassungsvorschlag gefunden werden konnte und somit keine Anpassung durchgeführt wird.

6.5.5 Durchführung von Modellanpassungen

Im positiven Fall, dass die Güte des Modells sowie der Prognose ausreichend ist, werden die Informationen aus der Regressionsschätzung über die Residuen, die Koeffizienten, deren Standardfehler sowie t- und p-Werte, aber auch das Bestimmtheitsmaß, der RMSE und MAE in einer Liste für jeden Anpassungszeitpunkt historisiert. Somit besteht die Möglich-

keit, durchgeführte Modellanpassungen im Nachhinein zu analysieren, aber auch bei der Entwicklung neuer Anpassungsvorschläge auf vergangene Fälle zurückzugreifen.

Im Anschluss soll das generische Modell in neo4j durch die Funktion `writeModelNeo` angepasst werden (vgl. Abbildung 6-17). Initial prüft die Funktion, ob das aktuelle Ergebnis der Regression sich von dem des letzten Anpassungszeitpunktes unterscheidet. Zusätzlich wird geprüft, ob das Signifikanzniveau der geschätzten Koeffizienten unter einem definierbaren Grenzwert liegt. Dieser wird mit dem Argument `inc_prob` übergeben. Zur Prüfung wird der p-Wert der F-Statistik verwendet, jener muss standardmäßig unterhalb von 0,05 liegen. Beim p-Wert handelt es sich um ein empirisches Signifikanzniveau, das Auskunft über die Irrtumswahrscheinlichkeit je unabhängiger Variable gibt (vgl. Backhaus et al. 2018b, S. 83). Mit der Prüfung des p-Wertes wird die Signifikanz der Koeffizienten bewertet, womit eine dritte Instanz zur Modellevaluation hinzugezogen wird. Diese betrachtet das Modell nicht als Ganzes, wie es das Bestimmtheitsmaß oder RMSE bzw. MAE tun, sondern die Zusammensetzung des Modells auf Variablenebene. Bei einem Grenzwert von 0,05 muss die Irrtumswahrscheinlichkeit jeder einzelnen unabhängigen Variable kleiner als 5% sein.

Liegen neue Regressionsergebnisse vor und sind die Koeffizienten signifikant, werden bei einer Parameteranpassung die Parameter der Kanten zwischen den unabhängigen und abhängigen Variablen mit der aktualisierten Wirkung überschrieben. Darüber hinaus wird der Zeitpunkt der Anpassung als Metadatum mit übergeben. Bei einer Strukturveränderung hingegen werden die neuen Kanten zwischen den unabhängigen und abhängigen Variablen hergestellt und der Wert des Parameters sowie des Anpassungszeitpunktes erfasst. Abschließend wird in dem DataFrame `newLines` (vgl. Tabelle 6-2) nach einer erfolgreichen Anpassung der Status des Attributes `ADAPTED` auf `TRUE` gesetzt.

Ist die Ursachenanalyse für jede abgewichene Variable durchgeführt worden und liegen keine neuen Anpassungsaufträge mehr vor, wird der Status von `BSTsisRunning` auf `FALSE` gesetzt.

6.6 Ganzheitliche Ablaufbeschreibung

Der technische Ablauf des *MoSyCo* Prototyps wird nachfolgend vorgestellt. Dadurch soll ein Verständnis über die Prozesse vermittelt sowie das Zusammenspiel der vorgestellten Komponenten auf Basis der entsprechenden Methodenaufrufe erläutert werden

Zur Erläuterung der Zusammenhänge der Elemente des *Observer* wird ein Sequenzdiagramm verwendet (vgl. Abbildung 6-19). Im Sequenzdiagramm sind die Hauptkomponenten für die Abweichungsanalyse weiß hinterlegt und die GUI-Komponenten sind grau gekennzeichnet. Die Benutzerschnittstelle besteht aus der Kommandozeile und der optionalen GUI. Über beides kann ein Anwender mit dem Prototyp interagieren. Wie bereits in den vorangegangenen Kapiteln erwähnt, arbeiten *Reader*, *Inspector* und *Plotter* parallel zueinander. Eine Darstellung dieser Nebenläufigkeit wäre im UML Sequenzdiagramm jedoch zu unübersichtlich geworden, weshalb darauf verzichtet wird. Stattdessen wird der Austausch der Daten zwischen den Elementen fokussiert, wodurch der Datenfluss, aber auch die Verteilung der Verantwortlichkeit aufgezeigt wird. Beim Start des *Reader* werden vom Anwender die zu überwachenden Variablen `vars` übergeben. Die Datenquelle ist eine CSV-Datei aus dem *System-Controller* (vgl. Kapitel 6.1), aus der jeder verfügbare Datenpunkt zu Beginn eingelesen wird und an die *Reader Queue* weitergeleitet wird (vgl. Abbildung 6-19). Im Anschluss fragt der *Inspector* mittels `receive()` die Plan- und Istwerte aus der *Reader Queue* ab. Durch das Sequenzdiagramm werden die Rollen der Warteschlangen (engl. *queue*) in der *Observer* Komponente hervorgehoben. Über die Warteschlangen wird der Informationsaustausch zwischen den Elementen *Reader*, *Inspector* und *Plotter* koordiniert, um das übergeordnete Ziel einer Systemüberwachung zur Abweichungsanalyse zu erfüllen. Mittels `eval_actual()` wird die Abweichungsanalyse gestartet, bei der der zu analysierende Zeitpunkt (`date`) sowie die Plan- und Istwerte übergeben werden. Insgesamt vernachlässigt Abbildung 6-19 durch die Nichtdarstellung der Parallelität lediglich, dass die einzelnen Komponenten nach Abschluss der Bearbeitung eines Datenpunktes direkt mit der Verarbeitung des nächsten Datenpunktes beginnen können. So verhält es sich bspw. auch bei der Abweichungsanalyse des *Inspector*. Im Rahmen der Abweichungsanalyse kann der *Inspector* auf absolute und relative Methoden (`methods`) zur Abweichungsanalyse zurückgreifen. Der Benutzer übergibt beim Start des *Observer* einen Grenzwert (vgl. Kapitel 6.4.2), der bei der Bewertung einer Abweichung berücksichtigt wird. Überschreitet die Abweichung den tolerierbaren Grenzwert, so wird die Überschreitung in der Log-Datei protokolliert sowie dem Benutzer über die Kommandozeile ausgegeben. Sofern der GUI-Modus aktiviert ist, wird die nicht tolerierbare Abweichung innerhalb der *System View* (vgl. Abbildung 6-11) durch `plot_model_actual_deviation()` rot eingezeichnet. Im selben Diagramm wird der Verlauf der Planwerte durch die Methode `plot_model()` und äquivalent dazu der Verlauf der Istwerte durch `plot_actual()` visualisiert. Die Planwerte

sind stetig eingezeichnet, die Istwerte werden in Abhängigkeit der Messung bzw. der Verfügbarkeit gezeichnet.

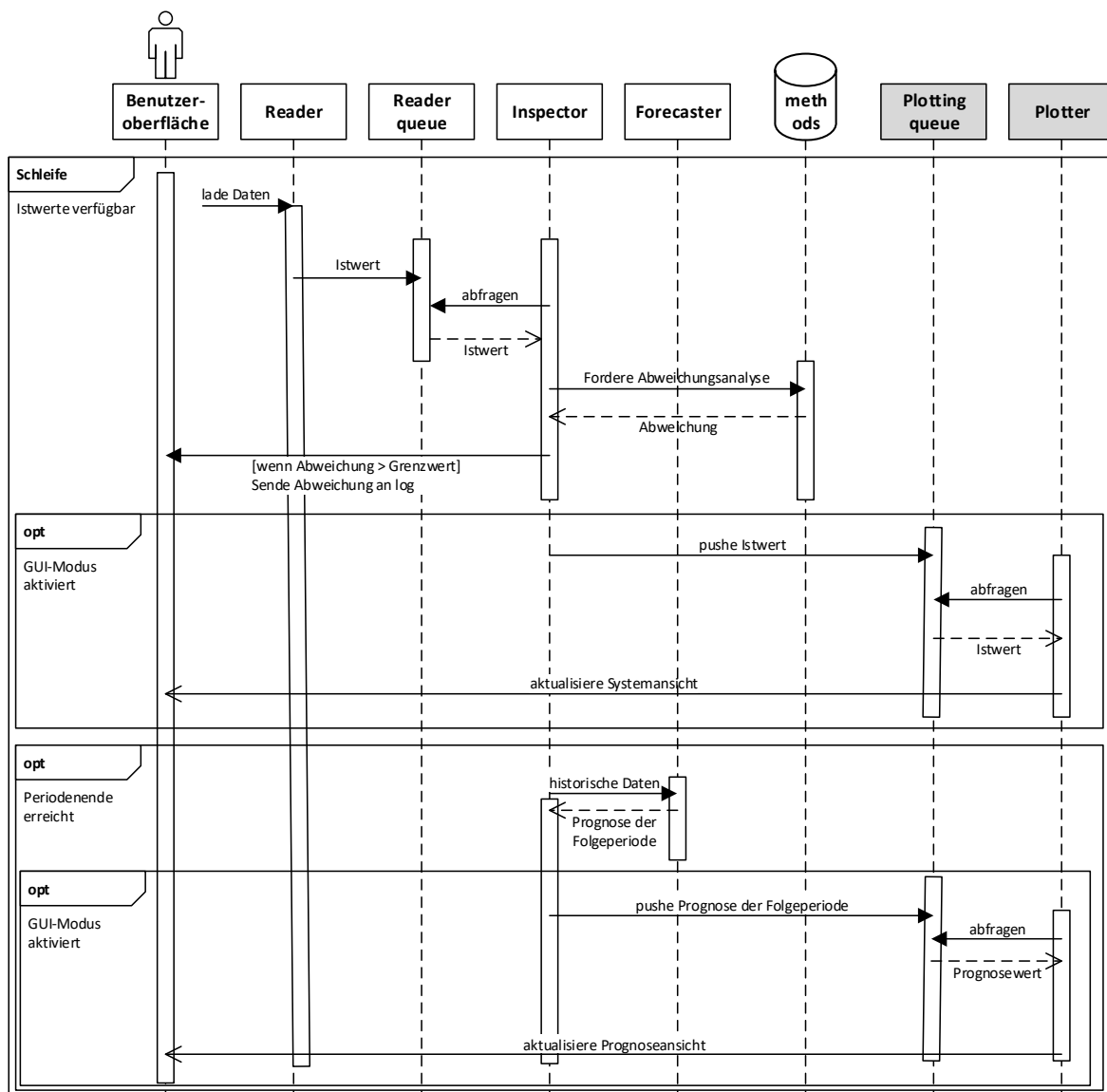


Abbildung 6-19: Observer Sequenzdiagramm

Quelle: Eigene Darstellung

In Abbildung 6-19 stellt das Periodenende ein weiteres optionales Fragment neben dem GUI-Modus dar. Sobald eine Periode vergangen ist, fordert der *Inspector* eine Prognose der nächsten Periode vom *Forecaster* an. Eine Periode ist standardmäßig als ein zwölfmonatiger Zeitraum definiert (vgl. Kapitel 6.4.5). Die Prognose wird mit der Methode `forecast_period()` aufgerufen und es werden die historischen Daten über die vergangene Periode (`period`) sowie der Name der zu prognostizierenden Variable (`var`) übergeben.

Der *Forecaster* ruft mit `predict()` die Methodenbank auf und übergibt *Prophet* (vgl. Kapitel 6.4.1) zur Zeitreihenanalyse die entsprechenden Datenpunkte. Der *Inspector* erhält die Prognose und kann mit der Methode `eval_future()` eine Abweichungsanalyse durchführen. Dabei werden die Planwerte mit den Prognosewerten über den Prognosehorizont von 12 Monaten verglichen. Äquivalent zur Abweichungsanalyse zwischen Ist- und Planwerten wird eine Abweichung als solche beurteilt, wenn diese den definierten Grenzwert überschreitet. Ist der GUI-Modus aktiviert, erfolgt periodenweise die Aktualisierung der *Forecast View* mittels der Methode `plot_forecast()`. Die Komponente des *Observer* arbeitet so lange bis keine weiteren Istwerte mehr vorhanden sind (vgl. Abbildung 6-19). Jegliche Abweichungen, die im Rahmen der Analysen identifiziert werden, werden unmittelbar in der Log-Datei protokolliert und damit dem *Model-Controller* zur Verfügung gestellt. Der Inhalt dieser Datei entspricht mit Ausnahme des letzten Attributes (*Adapted*) dem Aufbau des DataFrame `newLines` (vgl. Tabelle 6-2).

Die Zusammenhänge der einzelnen Elemente innerhalb des *Model-Controller* werden durch ein Sequenzdiagramm in Abbildung 6-20 erläutert. Die Funktion `readlog` fügt die jeweils abgewichenen Variablen (`DevVars`) dem DataFrame `newLines` hinzu. Diese Liste repräsentiert im prototypischen Stadium das *Adaptation Repository* (vgl. Kapitel 4.2.4). Die Funktion `ApplyBSTS` wird ausgeführt, welche iterativ für alle abgewichenen Variablen einen Modellanpassungsvorschlag entwickelt. Je abgewichener Variable werden die unabhängigen (`getConnectedNodes`) und die frei verfügbaren (`IndiNodeExtract`) Variablen in *neo4j* identifiziert. In Abhängigkeit, ob nur sog. „potential Variables“ berücksichtigt werden sollen (`VarFilter = TRUE`), wird die Menge an freien Variablen eingegrenzt (vgl. Kapitel 6.5.3). Für die abgewichene, die abhängigen sowie die freien Variablen werden jeweils die Istwerte aus der *CSV*-Datei extrahiert (`extractActualValues`). Dabei handelt es sich um die aus *Vensim* stammende *CSV*-Datei über alle im Modell enthaltenen Variablen mit den Plan- sowie Istwerten. Die Istwerte werden als *zoo* Zeitreihe (vgl. Kapitel 6.5.1) abgespeichert und in einen Trainings- und Testdatensatz aufgeteilt (vgl. Kapitel 6.5.3).

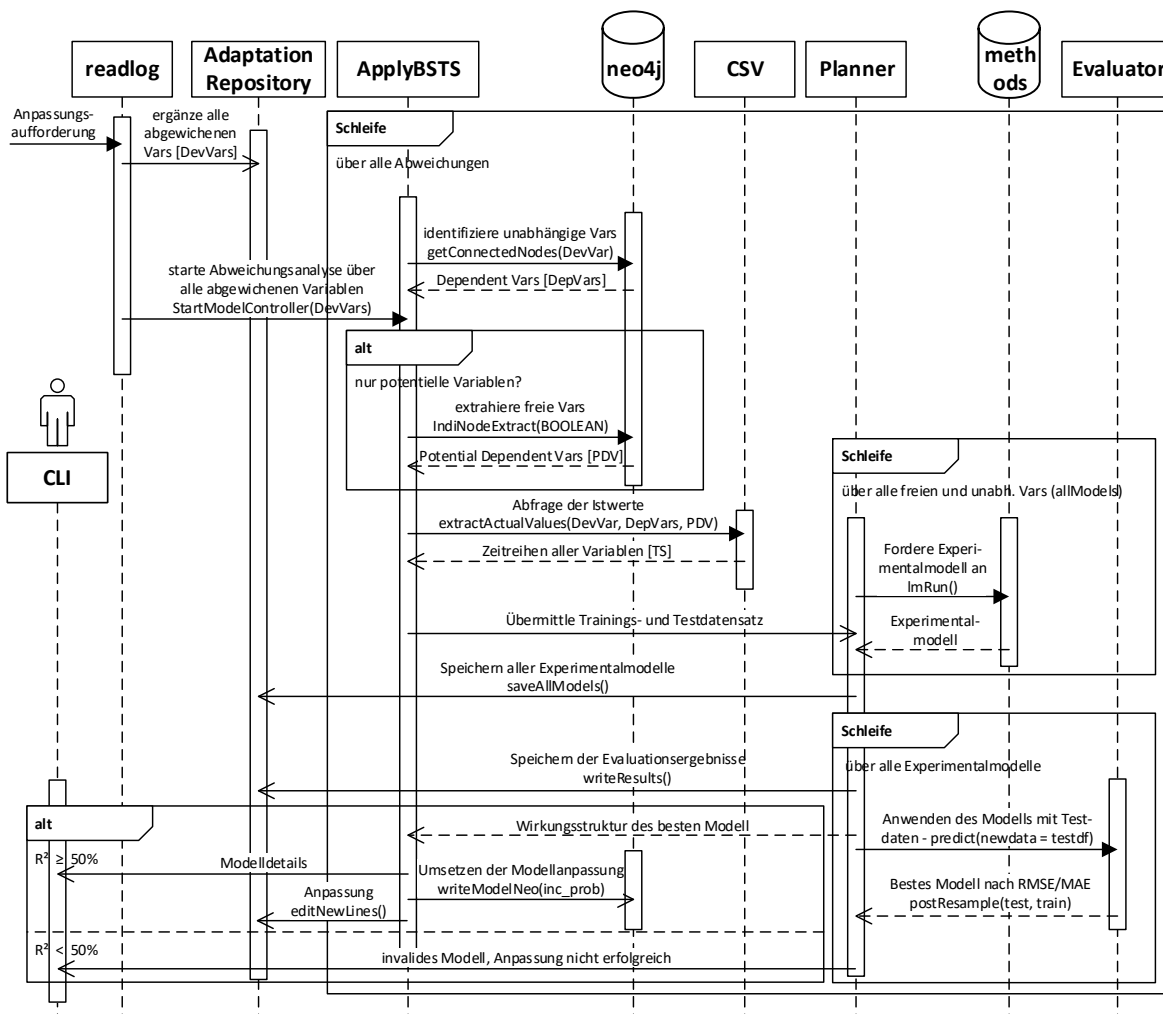


Abbildung 6-20: Model-Controller Sequenzdiagramm

Quelle: Eigene Darstellung

Der *Planner* entwickelt mit Methoden aus der Methodenbank verschiedene Experimentalmodelle. Die unterschiedlichen Strukturen der Experimentalmodelle werden in der Liste `allModels` gespeichert (vgl. Abbildung 6-20). Zur Schätzung der Wirkungsintensität wird eine gewichtete lineare Regression angewandt (vgl. Kapitel 6.5.4). Die Methode `lmRun()` führt über alle in der Liste `allModels` enthaltenen Strukturen die Regression zur Schätzung der vollständigen Gleichung aus (vgl. Kapitel 6.5.4). Alternativ steht hierfür das *BSTS* Paket (vgl. Kapitel 6.5.1) zur Verfügung. Je abgewichener Variable werden mehrere Experimentalmodelle entwickelt, die sowohl Parameter- als auch verschiedene Strukturanpassungen enthalten (vgl. Tabelle 6-3), weshalb in einem nächsten Schritt der *Evaluator* das Modell mit der besten Güte auswählt. Hierfür werden alle Experimentalmodelle mit dem Testdatensatz angewandt und mit Hilfe der Funktion `postResample()` hinsichtlich der Prognosegüte evaluiert. `postResample` bekommt sowohl den Test- als auch den Trainings-

datensatz übergeben und berechnet u. a. den RMSE und das R^2 (vgl. Kapitel 6.5.4). Die Ergebnisse der Evaluation werden über die Methode `writeResults()` abgespeichert. Das R^2 stellt die Referenz für das alternative Systemverhalten innerhalb des `alt`-Fragmentes des UML Sequenzdiagrammes dar (vgl. Abbildung 6-20). Kann das beste Modell weniger als 50% der Varianz (R^2) erklären, wird der Anwender über das invalide Modell informiert. Sofern das beste Modell ein R^2 von größer gleich 50% aufweist, wird die Modellanpassung eingeleitet. Die Anpassungen werden in *neo4j* durch die Methode `writeModelNeo()` übernommen, sofern die p-Werte der einzelnen Regressoren den definierbaren Grenzwert (`inc_prob`) einhalten. Ist die Anpassung erfolgreich gewesen, wird der `DataFrame` `newLines` durch die Methode `editNewLines()` geändert, indem in der jeweiligen Zeile der abgewichenen Variable der Wert des Attributes `ADAPTED` (vgl. Tabelle 6-2) auf `TRUE` gesetzt wird, im negativen Fall entsprechend auf `FALSE`. Der Anwender wird über die Kommandozeile (CLI) über umgesetzte Anpassungen informiert. Ein exemplarischer Output über die Kommandozeile ist in Abbildung 6-21 dargestellt.

```

[1] "Der MoSyCo Prototyp läuft und beginnt mit einer Abweichungsanalyse, sobald die Plan- von den Istwerten abweichen. Zur Ursachenanalyse wird eine lineare Regression durchgeführt. Bei potentiellen Strukturänderungen wird eine schrittweise Regression angewandt, die freie Variablen auf ihren Erklärungsbeitrag prüft. Regressoren, die einen p-wert von unter 0.05 aufweisen, werden als valide gewertet und im Modell berücksichtigt."
[2] "Die abgewichene Variable lautet: ANachfrage"
[3] "ANachfrage ist 1614 mal vom Planwert abgewichen, bei einer Toleranz von 5 Prozent."
[4] "Die letzte Abweichung fand zum Vensim-Zeitpunkt 1700 statt, zusätzliche Infos über die Abweichung:"
      TIME VARIABLE DEVIATED ACTUAL PLAN ABS_DEV REL_DEV ADAPTED
1 2009-08-27 ANachfrage TRUE 2431.89 1331.56 1100.33 0.4524588 NA
[5] "zur Regressionsbildung wurden 50 Zeitpunkte verwendet"
[6] "Die Abweichung ist auf eine strukturveränderung zurückzuführen, die aktualisierten Variablen sowie Koeffizienten lauten:"
      Estimate Std. Error t.value Pr...t...
Export      14.9955744 0.002598026146 5771.911 3.765703e-139
DAX          0.1499844 0.000005033751 29795.745 1.182036e-172
Substitutionsmarktwachstum 2.0019759 0.001049538597 1907.482 1.500070e-116
[7] "Das aktualisierte Modell zum Datum 2009-08-27 bzw. Vensim-Zeitpunkt 1700 ist hier verfügbar: http://scilly:7474/browser/"
[8] "_____>

```

Abbildung 6-21: R Kommandozeilenoutput nach erfolgreicher Modellanpassung

Quelle: Eigene Darstellung

In Zeile 1 wird der Anwender anfänglich über das Ziel und die Vorgehensweise des *MoSyCo* Prototyps unterrichtet. Wird eine Abweichung festgestellt, werden der Name der abgewichenen Variable (Zeile 2), wie häufig eine Abweichung gemessen wurde (Zeile 3) und Informationen über die Höhe der Abweichung ausgegeben (Zeile 4). Ab Zeile 5 werden Details zum am besten evaluierten Modell ausgegeben, welches in *neo4j* umgesetzt wurde. In Zeile 5 wird der Anwender über die Anzahl berücksichtigter Datenpunkte zur Bildung der gewichteten Regression informiert. Zeile 6 enthält das aktualisierte Modell in Form der Regressoren (Spalte 1), den Koeffizienten bzw. Parametern (Spalte 2), den Standardfehlern (Spalte 3), den t- (Spalte 4) sowie den p-Werten (Spalte 5). Der Zeitpunkt, zu dem die Modellanpassung stattgefunden hat, sowie ein Link zur grafischen Darstellung des aktualisier-

ten Modells in *neo4j* wird abschließend in Zeile 7 ausgegeben. Die äquivalente Darstellung der Anpassung in dem generischen *Model* in *neo4j* ist in Abbildung 6-22 visualisiert.

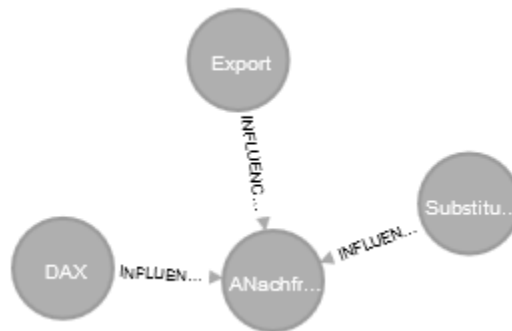


Abbildung 6-22: Neo4j Output nach erfolgreicher Modellanpassung

Quelle: Eigene Darstellung

Das *Substitutionsmarktwachstum* wirkt zusätzlich zu den beiden bekannten unabhängigen Variablen *DAX* und *Export*. Details zu der Anpassung wie der Anpassungszeitpunkt oder die Wirkungsintensität auf die *Nachfrage* werden in den Eigenschaften der Kante vorgehalten (vgl. Abbildung 6-5).

7 Ex post Evaluation des MoSyCo Prototyp

Im Rahmen der ex post Evaluation Eval 3 wird der entwickelte Prototyp bewertet und dabei geprüft, ob dieser seinen Zweck erfüllt und welches Verbesserungspotential vorhanden ist (vgl. Kapitel 1.2). Zu Beginn des Kapitels werden die Evaluationsziele sowie die Hypothesen aufgestellt (vgl. Kapitel 7.1). Anschließend werden das Untersuchungsdesign (vgl. Kapitel 7.2) und die Durchführung der Evaluation (vgl. Kapitel 7.3) erläutert. Abschließend werden die Ergebnisse vorgestellt sowie interpretiert (vgl. Kapitel 7.4) und die Limitationen der Untersuchung diskutiert (vgl. Kapitel 7.5).

7.1 Evaluationsziel und Hypothesen

Für den Beginn einer Evaluation wird empfohlen, die Evaluationsziele und zentralen Fragestellungen zu bestimmen (vgl. Stockmann und Meyer 2010, S. 161). Das primäre Ziel ist es, Erkenntnisse über die Gebrauchstauglichkeit (engl. usability) des *MoSyCo* Prototyps zu gewinnen. Die zur Beurteilung eingesetzten Kriterien werden in Kapitel 7.1.1 hergeleitet. Darüber hinaus sollen Aussagen über das Vertrauen in die Ergebnisse des Prototyps sowie die selbst-adaptive Arbeitsweise getroffen werden (vgl. Kapitel 7.1.2).

7.1.1 Usability

Sonnenberg und vom Brocke (2012, S. 393) nennen Kriterien zur Untersuchung eines Artefaktes im Rahmen der ex post Evaluation, zu denen unter anderem die Effektivität und Effizienz gehören. Es soll geprüft werden, wie und wie gut ein Prototyp bei einer Nutzung funktioniert und somit die Gebrauchstauglichkeit (engl. usability) getestet werden (vgl. Sonnenberg und vom Brocke 2012, S. 395). Aus der Definition der Usability kann ein weiteres Kriterium für die Evaluation abgeleitet werden. Die DIN EN ISO 9241-11 definiert Usability als „*das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen*“ (DIN EN ISO 9241-11 1999, S. 94). Für eine zufriedenstellende Nutzung steht die Unterstützung des Anwenders bei der Zielerreichung in dem jeweiligen Kontext im Vordergrund. Zur Beurteilung der Qualität der Unterstützung kann die Effektivität sowie Effizienz berücksichtigt werden und die Zufriedenheit des Nutzers stellt ein subjektives Maß über die Unterstützungsqualität dar. Im Folgenden werden Hypothesen zur Untersuchung der *Effektivität, Effizienz und Zufriedenheit* im Rahmen der Nutzung

des *MoSyCo* Prototyps aufgestellt. Damit die Leistung eines Prototyps besser bewertet werden kann, ist es sinnvoll, zwei funktional ähnliche Systeme miteinander zu vergleichen (vgl. Peffers et al. 2012, S. 405). Dieses Alternativsystem sollte zur Erfüllung der Aufgaben äquivalente Werkzeuge bereitstellen. Aus diesem Grund sollen im Rahmen der ex post Evaluation auch Probanden teilnehmen, die ein funktional vergleichbares System zur Erfüllung der selben Aufgaben nutzen.

Die *Effektivität* beschreibt die Vollständigkeit und Genauigkeit, mit der ein bestimmtes Ziel erreicht wird (vgl. Sarodnick und Brau 2011, S. 37). Auf den *MoSyCo* Prototyp projiziert bedeutet dies, dass die korrekten Ursachen einer Abweichung identifiziert werden und adäquate Modellanpassungsvorschläge ausgegeben werden. Andernfalls ist die Grundlage für eine valide Entscheidungsunterstützung nicht gegeben. Zur Überprüfung der Effektivität soll die Zielsetzung klar formuliert sein, damit die Daten eindeutig erhoben werden können und Fehlinterpretationen vermieden werden (vgl. Döring und Bortz 2016b, S. 983). Der Erfüllungsgrad einer Aufgabe spiegelt häufig die Effektivität wider (vgl. Hornbæk 2006, S. 82). Sofern ein Nutzer mit der Unterstützung eines Systems die zu erfüllenden Aufgaben erfolgreich umsetzen kann, ist ein System effektiv. Für die Evaluation des *MoSyCo* Prototyps bedeutet dies, dass die Anpassungsvorschläge für das Modell korrekt sind. Als weiteres Kriterium zur Beurteilung des Erfüllungsgrades einer Aufgabe kann die Genauigkeit des Systems untersucht werden. Im Falle von wiederholt auftretenden Abweichungen kann die Menge korrekter Modellanpassungsvorschläge den falschen gegenübergestellt werden. Zusammengefasst lässt sich die folgende Hypothese ableiten:

H 7-1: Mit Hilfe des Prototyps können Abweichungs- und Ursachenanalysen effektiver als mit einem funktional vergleichbaren System umgesetzt werden.

Die *Effizienz* ist das zweite Kriterium zur Messung der Usability. Gemäß der DIN ISO EN 9241-11 setzt die Effizienz den notwendigen Aufwand mit der Genauigkeit und der Vollständigkeit einer Zielerreichung in Relation (DIN EN ISO 9241-11 1999, S. 94). Der Aufwand wird häufig als die Zeit operationalisiert, die für das Erledigen einer Aufgabe benötigt wird (vgl. Hornbæk 2006, S. 84 f. Sarodnick und Brau 2011, S. 37). Der *MoSyCo* Prototyp ist demnach effizient, wenn dieser weniger Zeit für eine Modellanpassung benötigt als ein System mit vergleichbarem Funktionsumfang. Da der Prototyp teilautomatisiert Abweichungsanalysen durchführt (vgl. Kapitel 6.4) und Modellanpassungsvorschläge au-

tomatisiert erarbeitet sowie umgesetzt (vgl. Kapitel 6.5), ist zu erwarten, dass der Prototyp effizienter arbeitet:

H 7-2: Mit Hilfe des Prototyps können Abweichungs- und Ursachenanalysen effizienter als mit einem funktional vergleichbaren System umgesetzt werden.

Bei den beiden Kriterien der Effektivität und der Effizienz handelt es sich um objektive Leistungsmerkmale, die den Erfüllungsgrad einer Aufgabe beschreiben. Bei der *Zufriedenheit* handelt es sich um ein subjektives Kriterium, welches sich bei den verschiedenen Teilnehmern unterscheiden kann. Gemäß der DIN ISO EN 9241-11 ist die Zufriedenheit definiert als „die Freiheit von Beeinträchtigungen und eine allgemein positive Einstellung gegenüber der Systemnutzung“ (DIN EN ISO 9241-11 1999, S. 94). Die Einstellung gegenüber der Systemnutzung ist insofern wichtig, da diese mit der Akzeptanz des Systems zusammenhängt (vgl. Davis et al. 1989, S. 985). Nach Davis et al. (1989, S. 985) führen eine wahrgenommene Nützlichkeit und eine wahrgenommene Leichtigkeit der Benutzung zu einer positiven Einstellung gegenüber dem System. Die wahrgenommene Nützlichkeit bezieht sich auf die subjektive Beurteilung, inwiefern das System dem Nutzer bei der Durchführung einer Aufgabe einen Mehrwert oder eine Leistungsverbesserung erbringt. Bei der wahrgenommenen Leichtigkeit handelt es sich um den vom Nutzer wahrgenommenen Aufwand, der benötigt wird, um das System zu nutzen (vgl. Davis et al. 1989, S. 985). Beide Faktoren haben einen direkten Einfluss auf die Nutzungsabsicht eines Systems (vgl. Davis et al. 1989, S. 985). Da der Prototyp sowohl eine teilautomatisierte Abweichungs- als auch eine automatisierte Ursachenanalyse durchführt, ist zu erwarten, dass ein Anwender einen Mehrwert bei der Nutzung des Prototyps wahrnimmt. Aufgrund der überwiegend automatisierten Prozesse des Prototyps fällt der Bedienungsaufwand gering aus. Es ist insofern zu erwarten, dass der Anwender eine hohe Leichtigkeit bei der Systemnutzung wahrnimmt. Zusammenfassend wird folgende Hypothese aufgestellt:

H 7-3: Nutzer des Prototyps weisen eine positive Einstellung gegenüber dem *MoSyCo* Prototyp auf.

7.1.2 Vertrauen in selbst-adaptive Systeme

Das Vertrauen in automatisierte Systeme ist wichtig, denn es beeinflusst die Adoption einer Technologie (vgl. McKnight 2005, S. 330). Dies hängt zum einen von dem Selbstbewusstsein einer Person und zum anderen von dem Verlass auf die Leistungsfähigkeit eines Sys-

tems ab (vgl. Lee und Moray 1994, S. 178; McKnight 2005, S. 330). Bereits bei der ersten Nutzung eines Systems können die Zuverlässigkeit, Qualität der Leistung und eine effektive Unterstützung das Vertrauen in eine Technologie erhöhen (vgl. McKnight 2005, S. 330). Auf Grundlage dieser Abhängigkeiten wird im Rahmen der Evaluation davon ausgegangen, dass durch die Funktionserfüllung des Prototyps folgendes gilt:

H 7-4: Das Vertrauen in automatisierte Systeme ist nach der Nutzung des Prototyps erhöht.

Neben dem generellen Vertrauen in automatisierte Systeme soll das Vertrauen in die Richtigkeit der Modellanpassungen überprüft werden. Die Anpassung eines betriebswirtschaftlichen Modells zur Entscheidungsunterstützung kann aufgrund der Multi-Dimensionalität (vgl. Kapitel 2.3) sehr komplex sein, weshalb diese Aufgabe einen negativen Einfluss auf das Selbstbewusstsein einer Person haben kann (vgl. Lee und Moray 1994, S. 178), die eine Abweichungsanalyse und Ursachenidentifikation durchzuführen hat. Lee und Moray haben im Rahmen ihrer Adoptionsforschung festgestellt, dass ein automatisiertes System bevorzugt wird, wenn das Vertrauen in das System das eigene Selbstbewusstsein übersteigt (vgl. Lee und Moray 1994, S. 178). Da die Abweichungsanalyse sowie Ursachenidentifikation sehr komplex sein können und im Vergleich zu Systemen mit ähnlichem Funktionsumfang der Prototyp zur automatisierten Erfüllung dieser Aufgaben entwickelt ist, wird von Folgendem ausgegangen:

H 7-5: Das Vertrauen über die Richtigkeit der Modellanpassungen vom Prototyp ist höher als bei einem funktional vergleichbaren System.

7.2 Untersuchungsdesign

Das Evaluationsziel und die aufgestellten Hypothesen bestimmen die Fragestellungen sowie die Datenerhebungsmethode (vgl. Döring und Bortz 2016d, S. 182 ff.). Im Rahmen der Evaluation des *MoSyCo* Prototyps wird ein Mixed-Methods-Ansatz verfolgt. Dieser Ansatz wird zu Beginn beschrieben (vgl. Kapitel 7.2.1), im Anschluss wird der fragebogengestützte Usability-Test als Erhebungsmethode vorgestellt (vgl. Kapitel 7.2.2). Abschließend wird die konzipierte Aufgabe (vgl. Kapitel 7.2.3) erläutert, die im Rahmen des Usability-Tests von den Probanden aus der Stichprobe (vgl. Kapitel 7.2.4) gelöst werden soll.

7.2.1 Mixed-Methods-Ansatz

Forschungsmethoden können grundlegend in die Kategorien quantitativ und qualitativ eingeteilt werden, wobei beide Ansätze Vor- und Nachteile sowie unterschiedliche Vorgehensweisen und Erhebungsmethoden aufweisen (vgl. Bryman 2015, S. 37). So werden beispielsweise im Rahmen der quantitativen Forschung Daten durch standardisierte und strukturierte Instrumente wie Fragebögen erhoben (vgl. Bryman 2015, S. 37 f.). Beim qualitativen Ansatz ist das Vorgehen hingegen weniger strukturiert und enthält bspw. mehr offene Fragen. Die Daten hierfür können durch Gruppendiskussionen oder Beobachtungen erhoben werden (vgl. Bryman 2015, S. 38). Beim Mixed-Methods-Ansatz werden die beiden Forschungsmethoden miteinander kombiniert, indem innerhalb einer Studie gleichzeitig eine qualitative als auch eine quantitative Teilstudie durchgeführt werden (vgl. Döring und Bortz 2016d, S. 184).

Der Mixed-Methods-Ansatz wird im Rahmen der ex post Evaluation des *MoSyCo* Prototyps verfolgt. Um die Effektivität, Effizienz und Zufriedenheit messen zu können, werden quantitative Daten benötigt. Beispiele hierfür sind die zur Lösung einer Aufgabe benötigte Zeit oder die Zufriedenheit der Nutzer mit dem Prototyp (vgl. Moser 2012, S. 224). Bei der quantitativen Forschungsmethode müssen die zu beurteilenden Merkmale im Vorfeld definiert sein. Beim TAM werden bspw. die vordefinierten Konstrukte häufig mit Hilfe eines Fragebogens untersucht (vgl. Davis et al. 1989, S. 990). Da es sich beim Evaluationsobjekt jedoch um einen Prototyp handelt, der nicht vollständig entwickelt ist, können die zu untersuchenden Merkmale im Vorfeld nicht vollumfänglich definiert werden. Aus diesem Grund wird zusätzlich ein qualitativer Ansatz verfolgt, über den im Vorfeld unbekannte Merkmale entdeckt werden sollen. Bspw. sollen nicht absehbare Probleme bei der Nutzung des Prototyps aufgedeckt werden. Auf diese Weise können Verbesserungsvorschläge abgeleitet werden, die für eine Weiterentwicklung notwendig sind. Darüber hinaus bietet der qualitative Ansatz die Möglichkeit, durch offene Fragen die deskriptiven Ergebnisse aus der quantitativen Teilstudie besser verstehen zu können (Kuckartz et al. 2008, S. 12).

7.2.2 Fragebogengestützter Usability Test

Ein Usability-Test soll die Schwachstellen eines Produktes aufdecken, um daraus Verbesserungsvorschläge ableiten zu können (vgl. Kapitel 7.1.1). Grundsätzlich findet ein Usability-Test in einem Labor statt, damit die Umgebung möglichst kontrolliert und frei von externen Störfaktoren ist (vgl. Döring und Bortz 2016d, S. 206; Sarodnick und Brau 2011, S. 168 f.). Für einen solchen Test sollen realistische Aufgaben gestellt werden, die die Pro-

banden mit Hilfe eines Produktes bzw. Systems zu lösen haben. Im Rahmen der ex post Evaluation werden die Testpersonen darum gebeten, ihre Gedanken während der Bearbeitung der Aufgabe laut zu äußern. Die Methode des lauten Denkens ermöglicht es, die Vorgehensweise und die Handlungen der Probanden besser zu verstehen (vgl. Bryman 2015, S. 228 f. Sarodnick und Brau 2011, S. 170). Während des Experimentes werden die Probanden beobachtet und die Äußerungen, Gesten sowie Auffälligkeiten protokolliert. Sobald das Experiment abgeschlossen ist, findet eine Nachbesprechung statt. Dabei wird auf die protokollierten Auffälligkeiten während der Lösung der Aufgaben eingegangen, bspw. warum eine Verzögerung bei der Bearbeitung aufgetreten ist. Darüber hinaus besteht die Möglichkeit, offene Fragen zu stellen.

Neben dem teilstrukturierten Interview wurde ein strukturierter Fragebogen eingesetzt. Die Umfrage ist auf Basis des Online-Umfragetools *LimeSurvey* (vgl. Engard 2009, S. 272) erstellt worden. Der Fragebogen ist zweigeteilt. Teil A des Fragebogens fokussiert die Vorkenntnisse der einzelnen Umfrageteilnehmer sowie deren Vertrauen in automatisierte Systeme. Der erste Teil des Fragebogens wird vor dem eigentlichen Experiment beantwortet. Die Fragen aus Teil B zielen auf den wahrgenommenen Nutzen und die Zufriedenheit mit dem System während des Experimentes ab. Da ein Teil der Hypothesen aus Kapitel 7.1 auf einen Vergleich zwischen zwei Systemen ausgelegt ist, ist eine Filterfrage vorgesehen, die das verwendete System vor den Fragen aus Teil B bestimmt. Tabelle 7-1 enthält den Fragenkatalog des Usability-Tests. Die meisten Fragen können einer der fünf Hypothesen aus Kapitel 7.1 zugeordnet werden. Einige Aussagen zielen auf die Vorkenntnisse der Teilnehmer ab, weshalb diese Fragen nicht direkt einer Hypothese zugeordnet werden können.

Tabelle 7-1: Fragenkatalog Usability-Test

Quelle: Eigene Tabelle

Hypo- these	Teil	Variab- le	Aussage
/	A	VV5_ SQ001	Ich verfüge über theoretische Erfahrungen bei der Analyse von Daten in einem modellbasierten System.
/	A	VV5_ SQ002	Ich verfüge über praktische Erfahrungen bei der Analyse von Daten in einem modellbasierten System.
/	A	VV6_ SQ001	Ich schätze meine analytischen Fähigkeiten gut ein.
/	A	VV6_ SQ002	Ich verlasse mich auf meine analytischen Fähigkeiten.
/	A	VV6_ SQ003	Ich vertraue Entscheidungen, die ich mit der Unterstützung eines Modells treffe.
7-4	A	EM_ SQ001	Ich vertraue automatisierten (selbstständig handelnden) computergestützten Systemen.
/	A	EM_ SQ003	Ich sehe / kenne den Nutzen modellgetriebener Systeme.
7-3	B	MA1_ SQ001	Es war einfach, mit dem System die Abweichungen und deren Ursachen zu finden.
7-2	B	MA1_ SQ002	Durch das System konnte ich die Abweichungen und deren Ursachen im Modell schnell finden.
7-3	B	MA1_ SQ003*	Es war komplex, die Ursachen von den Abweichungen zu identifizieren.
7-3	B	MA1_ SQ004	Die Durchführung von Anpassungen funktionierte einwandfrei.
7-3	B	MA1_ SQ005*	Es hat mich insgesamt viel Zeit gekostet, um die Anpassungen im Modell durchzuführen.
7-3	B	MA1_ SQ006	Beim Finden von Abweichungen im Modell bin ich mit der Leistung des Systems zufrieden.
7-3	B	MA1_ SQ007	Beim Erstellen von Anpassungsvorschlägen für das Modell bin ich zufrieden mit der Leistung.
7-3	B	MA1_ SQ008	Die automatische Modellanpassung erleichtert meine Arbeit.
7-5	B	MA1_ SQ009	Ich bin mir sicher, dass die Modellanpassungen korrekt sind.
7-5	B	VA1_ SQ001	Ich verlasse mich darauf, dass die Änderungsvorschläge vom System richtig sind.
7-5	B	VA1_ SQ002	Es hat mich Zeit gekostet, da ich die Richtigkeit der Änderungsvorschläge nochmal prüfen musste.
7-3	B	VA1_ SQ003	Ich würde das System zur Identifikation von Ursachen bei demselben Problem wieder nutzen.
7-5	B	VA1_ SQ004	Das System soll mehr Informationen dazu liefern, wie es auf die neuen Parameter gekommen ist.
7-5	B	VA1_ SQ005	Im Falle einer Fehleinschätzung / Fehlanpassung des Modells würde ich dem System nicht mehr vertrauen.
7-4	B	PD3_ SQ004	Ich vertraue automatisierten (selbstständig handelnden) computergestützten Systemen.

7.2.3 Aufgabe im Rahmen des Usability-Tests

Die Teilnehmer des Experimentes werden in zwei Gruppen eingeteilt, welche sich in dem genutzten System unterscheiden. Die Aufgabe wird dieselbe sein, jedoch wird die Experimentalgruppe (EG) den *MoSyCo* Prototyp nutzen und die Kontrollgruppe (KG) ein System mit einem ähnlichen Funktionsumfang (vgl. Kapitel 7.1). Da die Hauptaufgaben des Prototyps die Analyse von Abweichungen und das Erstellen von Modellanpassungsvorschlägen sind, müssen diese Funktionalitäten durch das Alternativsystem ermöglicht werden. Für eine Abweichungsanalyse müssen Differenzen berechnet werden und bei einer Toleranzüberschreitung hervorgehoben werden. Der *MoSyCo* Prototyp nutzt vorrangig eine lineare Regression zur Entwicklung von Modellanpassungsvorschlägen (vgl. Kapitel 6.5). Diese Funktionen deckt das Leistungsspektrum von Microsoft Excel ab (vgl. Turban et al. 2011, S. 149 ff.). Excel ist ein Tabellenkalkulationsprogramm, das für Abweichungsanalysen entsprechende Rechenoperatoren zur Verfügung stellt (vgl. Turban et al. 2011, S. 150). Lineare Regressionsanalysen können standardmäßig auf Basis des Funktionsumfangs von Microsoft Excel erstellt werden; durch Erweiterungsmöglichkeiten lassen sich bspw. auch nichtlineare Optimierungen durchführen oder künstliche neuronale Netze erstellen (vgl. Turban et al. 2011, S. 149). Aufgrund der Integration in die Office-Pakete von Microsoft und der sehr vielfältigen Anwendungsgebiete ist das Tabellenkalkulationsprogramm sehr weit verbreitet (vgl. Benker 2014, S. 1 f.) und zählt zu den beliebtesten Anwendungen für Endbenutzer (vgl. Turban et al. 2011, S. 149). Somit ist Microsoft Excel als Alternativsystem geeignet, da es den erforderlichen Funktionsumfang bietet und die Teilnehmer aufgrund der weiten Verbreitung grundlegende Anwendungskenntnisse aufweisen können. Der Prototyp und Excel werden auf einem Computer mit Microsoft Windows 7, einem Intel Core i7-4600M Prozessor und 16 GB RAM ausgeführt.

Die Teilnehmer des Experimentes arbeiten mit dem Entscheidungsmodell zur Lager- und Absatzplanung aus Kapitel 6.2. Die Ausgangssituation ist, dass folgende Entscheidungen getroffen werden:

- Verkaufspreis: 300 GE
- Verkaufsbestellfaktor: 0,8
- Lagersoll: 2000
- Lagerdefizitbestellfaktor: 0,5

Auf Basis dieser Entscheidungsvariablen und der in dem Modell getroffenen Annahmen (vgl. Anhang VIII) ist der Verlauf der *Nachfrage* aus Abbildung 7-1 zu erwarten.

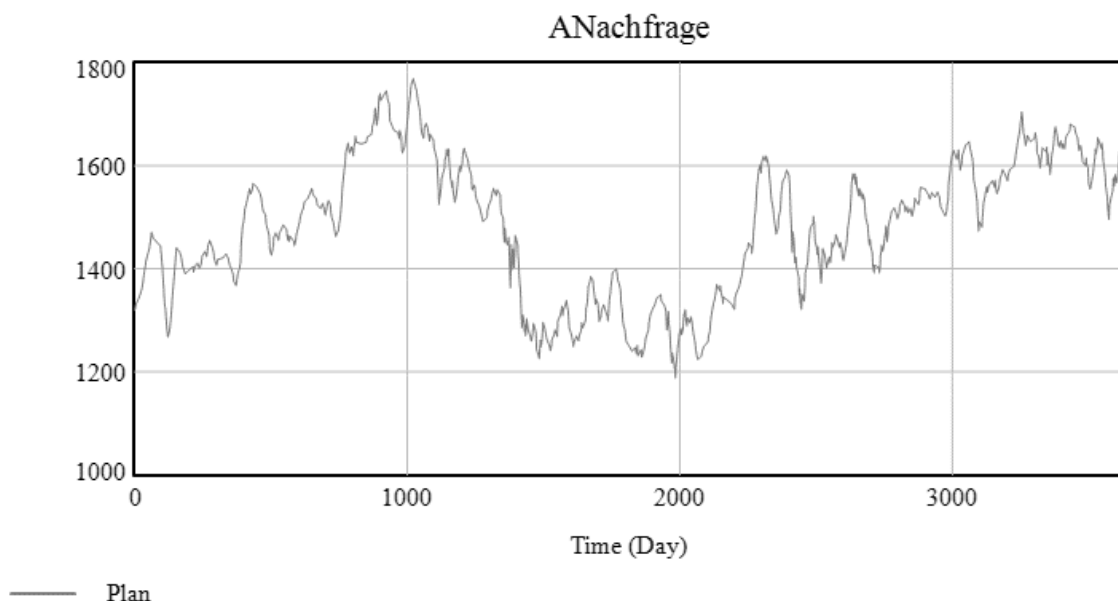


Abbildung 7-1: Erwartete Nachfrage zum Entscheidungszeitpunkt 0

Quelle: Eigene Darstellung

Die Entwicklung der *Nachfrage* basiert auf den exogenen Einflussfaktoren *DAX* und *Export* (vgl. Formel 7-1) und wirkt auf den *Tagesverkauf*, der wiederum indirekt auf die Lagerkapazitäten, die Preise, das Bestellverhalten und die Produktionsmengen Auswirkungen hat (vgl. Abbildung 6-2).

$$Nachfrage = 0,1 * DAX + 10 * Export$$

Formel 7-1: Berechnung der Nachfrage in Vensim zum Zeitpunkt 0

Quelle: Eigene Darstellung

Somit haben die Annahmen über die Berechnung der *Nachfrage* einen Einfluss auf erfolgsrelevante Faktoren wie den Umsatz und die Kosten (vgl. Abbildung 6-2), weshalb insbesondere die Überwachung der *Nachfrage* von hoher Relevanz ist. Die Aufgabe innerhalb des Experimentes ist es, die Entwicklung der *Nachfrage* zu prüfen und bei einer nicht tolerierbaren Abweichung von den Planwerten eine Ursachenanalyse im Modell durchzuführen.

Für eine Abweichungsanalyse ist es notwendig, Istwerte zum Vergleich zur Verfügung zu haben. Da es sich um ein kontrolliertes Experiment handelt, werden die Istwerte ebenfalls modellgetrieben in einem Realitätsmodell erstellt. Das Entscheidungsmodell basiert auf

Annahmen über die Wirkungen der *Nachfrage* in der Realität. Da das Grundproblem modellgetriebener Entscheidungsunterstützungssysteme ist, dass die Modellannahmen im Laufe der Zeit nicht mehr gültig sein können (vgl. Kapitel 2.3), ist es für das Experiment notwendig, einen zeitlichen Verlauf abbilden zu können. Realitäts- und Entscheidungsmodell sind miteinander verknüpft (vgl. Abbildung 7-2).

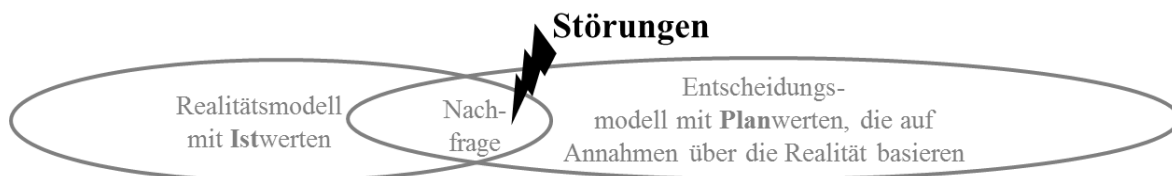


Abbildung 7-2: Abhängigkeit zwischen Realitäts- und Entscheidungsmodell

Quelle: Eigene Darstellung

Die Wirkungsstruktur der *Nachfrage* wird im Entscheidungsmodell angenommen, verändert sich im Rahmen des Experimentes jedoch durch Einflüsse aus dem Realitätsmodell. Die Probanden des Experimentes haben keinen Zugriff auf das Realitätsmodell. Die Teilnehmer haben jedoch die Möglichkeit, die Realität voranschreiten zu lassen und somit Istwerte für die Abweichungsanalyse zu erzeugen. Der zeitliche Horizont beider Modelle beträgt 10 Jahre bzw. 3650 Tage.

Die Aufgabe der Teilnehmer an der *ex post Evaluation* ist es, pro Jahr einen zufälligen Zeitpunkt auszuwählen und im Anschluss eine Abweichungsanalyse und ggf. Ursachenidentifikation durchzuführen. Für die Auswahl eines Zeitpunktes stehen den Teilnehmern zehn Schieberegler, einer pro Jahr, zur Verfügung (vgl. Abbildung 7-3).

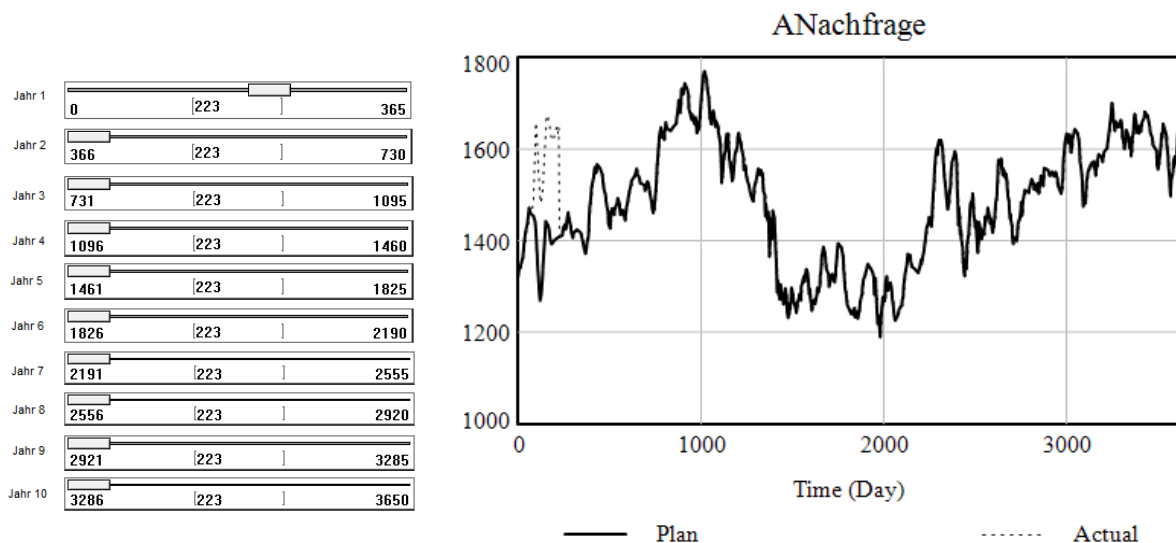


Abbildung 7-3: Entwicklung der Nachfrage zum Zeitpunkt 223 in Vensim

Quelle: Eigene Darstellung

In Abbildung 7-3 ist über die Schieberegler der Zeitpunkt 223 ausgewählt worden. Somit werden im Realitätsmodell bis zu diesen Zeitpunkt Istwerte generiert, welche im Entscheidungsmodell verfügbar sind. Die Istwerte werden in Abbildung 7-3 durch die gestrichelte Linie dargestellt. Es fällt auf, dass ab dem Zeitpunkt 80 eine Abweichung zwischen den Planwerten (durchgezogene Linie) und den Istwerten vorhanden ist (vor dem Zeitpunkt 80 überlagert die Linie der Planwerte die der Istwerte). Die Abweichung ist auf die erste im Realitätsmodell implementierte Störung (vgl. Tabelle 7-2) zurückzuführen. Bei der Berechnung der Nachfrage erhöht sich die Wirkung des DAX von 0,1 auf 0,15 (vgl. Formel 7-2; Tabelle 7-2).

$$Nachfrage = 0,15 * DAX + 10 * Export$$

Formel 7-2: Berechnung der Nachfrage in Vensim zum Zeitpunkt 95

Quelle: Eigene Darstellung

Alle Störungen, die auf die Nachfrage wirken, sind in Tabelle 7-2 aufgeführt. Jede Wirkungsänderung im Realitätsmodell ist mit der RAMP-Funktion modelliert, die im Intervall zwischen einem Start- und Endzeitpunkt (vgl. Spalte 3 in Tabelle 7-2) einen Anstieg oder eine Senkung der Wirkung verteilt („Vensim Help“ 2018). Im Rahmen der ersten Störung wird der Anstieg der Wirkung von 0,05 Einheiten auf die 15 Zeitpunkte zwischen 80 und 95 verteilt.

Tabelle 7-2: Störungen zur Berechnung der Nachfrage im Realitätsmodell

Quelle: Eigene Tabelle

Störung	Variable	Zeitpunkt	Wirkung
1	DAX	80	0,1
		95	0,15
2	Export	800	10
		900	15
3	Substitutionsmarktwachstum	1470	0
		1490	2
4	Export	2200	15
		2201	22
	Substitutionsmarktwachstum	2200	2
		2201	0,6
5	Wetter	2970	0
		2990	8,5
	Substitutionsmarktwachstum	2970	0,6
		2990	-0,8

Diese Störungen sind in dem nicht für die Experimentteilnehmer verfügbaren Realitätsmodell enthalten. Dieses Modell ist in Abbildung 7-4 dargestellt und basiert auf den Ansätzen für ein Handels- und Preissetzungsmodell nach Sterman (2000, S. 811 ff.). Es soll mit seiner Wirkungsstruktur die dynamische Unternehmensumwelt in der Realität repräsentieren (vgl. Anhang IX).

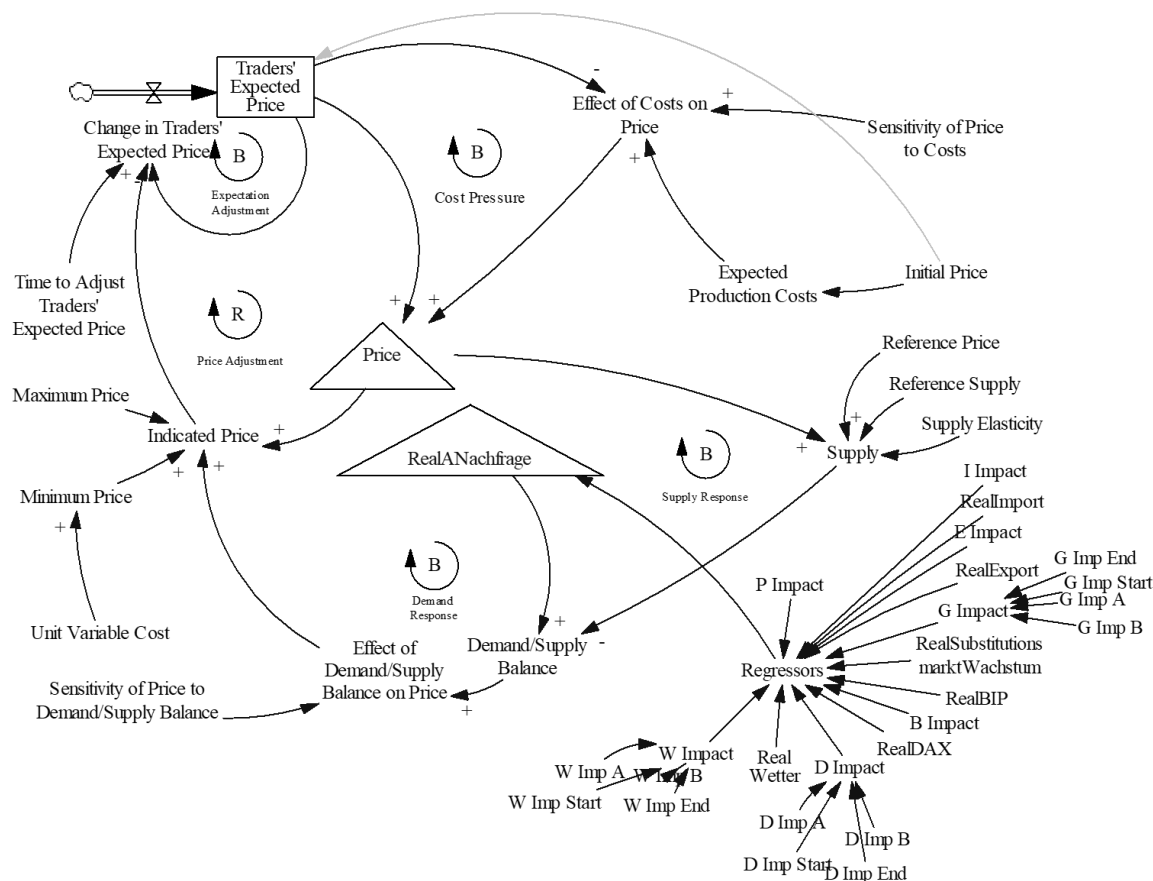


Abbildung 7-4: Realitätsmodell in Vensim

Quelle: Eigene Darstellung in Anlehnung an Sterman (2000, S. 811 ff.)

Der erwartete Handelspreis (*Traders' Expected Price*) kann bspw. von einer UVP abhängen. Jener bildet eine Grundlage für den zu berechnenden Preis (engl. *Price*). Dieser beeinflusst und wird beeinflusst von den Produktionskosten, dem Angebot sowie der *Nachfrage* (hier *RealANachfrage* genannt). Die Variablen *Price* und *RealANachfrage*⁸ sind als Dreiecke dargestellt und bilden die Schnittstelle zum Entscheidungsmodell (vgl. Abbildung 7-2). Das Entscheidungsmodell versucht, diese beiden Größen durch Annahmen zu bestimmen, deckt jedoch die im Laufe der Zeit eintretenden Störungen aus Tabelle 7-2 nicht ab. Die *Nachfrage* im Realitätsmodell kann von Regressoren wie bspw. dem Wachstum

⁸ Die Variablen mit dem Präfix *Real* verweisen auf Variablen aus dem Realitätsmodell.

eines *Substitutionsmarktes*, dem *BIP*, dem *DAX* oder auch vom *Wetter* abhängen (vgl. Abbildung 6-2).

7.2.4 Stichprobe und Einführungsveranstaltung

Häufig wird ein größerer Stichprobenumfang präferiert, denn eine Studie wird mit steigendem Stichprobenumfang aussagekräftiger (vgl. Döring und Bortz 2016d, S. 294). Dennoch können nach Kuzel auch mit einer geringeren Anzahl an Teilnehmern repräsentative Ergebnisse erreicht werden, sofern es sich um eine homogene Gruppe handelt (vgl. Kuzel 1992, S. 41). Guest et al. (2006, S. 74) haben diese Aussage empirisch untersucht und festgestellt, dass innerhalb einer homogenen Gruppe bereits nach 12 Teilnehmern eine Informationssättigung eintritt. Diese Sättigung ist erreicht, wenn durch zusätzliche Untersuchungsteilnehmer keine weiteren Erkenntnisse gewonnen werden können (vgl. Guest et al. 2006, S. 59).

An dem Experiment nehmen 13 Probanden teil, die zufällig in die beiden Gruppen eingeteilt wurden. Sieben Teilnehmer wurden der Experimentalgruppe und sechs Teilnehmer der Kontrollgruppe zugeteilt. Bei der ex post Evaluation handelt es sich um ein quasi-Experiment. Bei einem quasi-Experiment wird keine Randomisierung⁹ bei der Personenauswahl vorgenommen (vgl. Bryman 2015, S. 60). Ein quasi-Experiment weist eine geringere interne Validität auf, da die Vergleichsgruppen sich hinsichtlich personengebundener Störvariablen unterscheiden können. Durch die fehlende Randomisierung sind die abhängigen Variablen möglicherweise nicht eindeutig auf den Einfluss einer unabhängigen Variable zurückzuführen sind (vgl. Döring und Bortz 2016c, S. 99). Um dem entgegenzuwirken, kann ein *Matching* der Stichproben vorgenommen werden, damit die interne Validität erhöht wird (vgl. Saunders et al. 2012, S. 175 f.). Hierfür sollen die Ausprägungen der Störvariablen der Teilnehmer in der KG ähnlich zu denen in der EG sein. Die Stichprobe der ex post Evaluation umfasst Studierende der Universität Osnabrück. Die Repräsentativität und externe Validität werden bei Studien mit Studierenden teilweise in Frage gestellt (vgl. Kohnke 2015, S. 195 f.). Studierende stellen eine eigene homogene Gruppe dar und seien nicht repräsentativ für andere Gruppen, womit die Übertragbarkeit auf andere Kontexte oder Personengruppen hinterfragt wird (vgl. Kohnke 2015, S. 195 f.). Somit ist es nach Kuzel (1992) und Guest et al. (2006) zum einen erstrebenswert eine möglichst homo-

⁹ Eine Randomisierung bei der Personenauswahl ist gegeben, wenn eine Person durch eine zufällige Auswahl an dem Experiment teilnimmt (vgl. Bryman 2015, S. 60).

gene Gruppe zu definieren, zum anderen birgt es nach Kohnke (2015) jedoch die Gefahr, nicht repräsentativ für andere Fälle zu sein.

Die Rekrutierung von Teilnehmern für ein Experiment ist oftmals zeitlich beschränkt und die Ressourcen sind stark eingeschränkt (vgl. Bryman 2015, S. 188). Auch die ex post Evaluation im Rahmen dieses Forschungsvorhabens ist von diesen Restriktionen betroffen, weshalb Studierende mit ähnlichen Voraussetzungen für eine Untersuchung ausgewählt werden (vgl. Sarodnick und Brau 2011, S. 213). Für die ex post Evaluation stellen das Alter, der Bildungsstand und die bisherigen Erfahrungen bei der Analyse von Daten mit einem Modell die relevanten *personengebundenen* Merkmale dar. Die am Experiment teilnehmenden Studierenden sind im Alter zwischen 22 und 30 Jahren. Sie sind Masterstudierende im Studiengang der Betriebswirtschaftslehre mit generalistischer Ausrichtung oder dem Schwerpunkt Accounting und weisen einschlägige Erfahrungen im Bereich der Datenanalyse auf. Die Aussage EM_SQ003 (vgl. Tabelle 7-1) wird von den Teilnehmern mit einem Mittelwert von 4 bestätigt. Sie kennen also den Nutzen modellgetriebener Systeme.

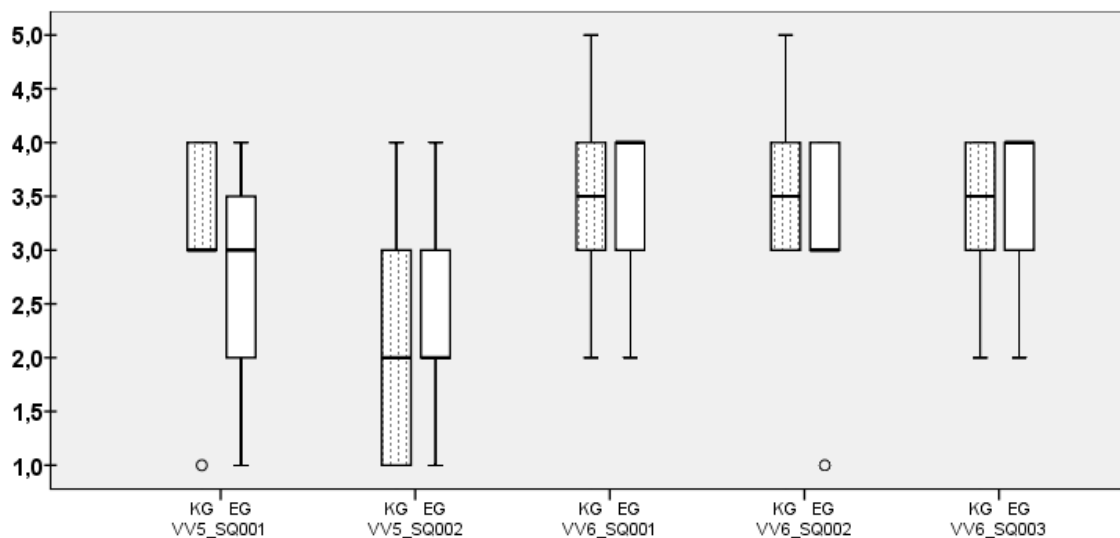


Abbildung 7-5: Boxplot über die Vorkenntnisse und Erfahrungen der Teilnehmer

Quelle: Eigene Darstellung

Abbildung 7-5 ist zu entnehmen, dass die Teilnehmer durchschnittlich hohe theoretische Vorkenntnisse über modellgetriebene EUS haben (VV5_SQ001) und praktische Erfahrungen diesbezüglich nur eingeschränkt vorhanden sind (VV5_SQ002). Die Mediane der Antworten der KG und der EG liegen auf demselben Niveau, die Streuung ist bei den theoretischen Vorkenntnissen der EG größer, wird jedoch durch die geringere Streuung bei den praktischen Erfahrungen ausgeglichen. Sowohl in der KG als auch in der EG verlassen sich die Studierenden aufgrund ihrer betriebswirtschaftlichen Ausbildung auf ihre analytischen

Fähigkeiten (VV6_SQ001) und schätzen diese mittel bis hoch ein (VV6_SQ002). Alle Teilnehmer haben das Fallstudienseminar Applied Analytics besucht, in dem durch den Einsatz verschiedener Technologien Datenanalysen zur Beantwortung betriebswirtschaftlicher Fragestellungen durchgeführt und auch Modelle zur Vorhersage entwickelt wurden. Somit kennen die Teilnehmer den Entwicklungsprozess eines Modells, was dazu führen kann, dass sie ein grundlegendes Vertrauen in modellbasierte Entscheidungen aufweisen (VV6_SQ003). Mit den sich ausgleichenden theoretischen und praktischen Vorkenntnissen sowie den fast gleich ausgeprägten Antworten aus der Fragenkategorie VV6 kann die Gruppe hinsichtlich der ausgewählten *personengebundenen* Merkmale als homogen eingestuft werden, sodass ein kleiner Stichprobenumfang ausreichend ist.

Zur Sicherstellung der Vorkenntnisse über EUS sowie betriebswirtschaftliche Modelle und dadurch zur weiteren Förderung einer homogenen Gruppe ist vor dem Experiment eine Einführungsveranstaltung durchgeführt worden. Dabei wurde den Studierenden neben den Grundlagen eines EUS sowie eines Entscheidungsmodells das im Rahmen der Evaluation betrachtete Modell vorgestellt (vgl. Abbildung 6-2). Vensim, die Umgebung zur Realisierung der Funktionalitäten des *System-Controller* (Kapitel 6.3), wurde den Studierenden exemplarisch vorgeführt. Zentraler Bestandteil der Veranstaltung war es, die Rolle der Studierenden und die zu erfüllenden Aufgaben zu definieren. Hierfür wurden exemplarisch Istwerte auf Basis des Realitätsmodells erzeugt (vgl. Kapitel 7.2.3) und eine Abweichungsanalyse sowie eine Ursachenidentifikation vorgeführt. Die Abweichungs- und Ursachenanalyse wurde sowohl mit dem Prototyp als auch mit Excel demonstriert. Der Fokus lag insbesondere auf der Vermittlung von Methoden, die in beiden Gruppen zum Einsatz kommen. Korrelationsanalysen zur Überprüfung einer Beziehung zwischen Variablen sowie die lineare Regression zur Schätzung einer Modellgleichung sind konkret betrachtet worden. Neben den Voraussetzungen und der Anwendung sind die verschiedenen Gütekriterien zur Bewertung der Analyseergebnisse erläutert worden. Im Rahmen der Einführungsveranstaltung sind die Teilnehmer durch die Erläuterung des Modells fachlich und durch die Vermittlung der Analysemöglichkeiten methodisch auf das Experiment vorbereitet worden. Zum Abschluss sind die Studierenden zufällig der Experimental- oder Kontrollgruppe zugeteilt worden.

7.3 Untersuchungsdurchführung

Das Experiment zur ex post Evaluation des *MoSyCo* Prototyps hat in Einzelterminen an vier Tagen stattgefunden. Bevor die Teilnehmer mit der Aufgabe der Abweichungsanalyse sowie ggf. der Ursachenidentifikation begonnen haben, mussten sie jeweils Teil A des Fragebogens (vgl. Tabelle 7-1) ausfüllen. Im Anschluss startete der Usability-Test und die Zeit, die zur Lösung der Aufgabe benötigt wurde, ist erfasst worden.

Während des Usability-Tests ist die Vorgehensweise bis zur Ursachenanalyse bei beiden Gruppen identisch gewesen. Zu Beginn hat der Proband mit Hilfe des Schiebereglers einen zufälligen Zeitpunkt im ersten Jahr ausgewählt (vgl. Abbildung 7-3). Bis zu dem ausgewählten Zeitpunkt konnten die Plan- und Istwerte aufgerufen werden. Im Anschluss unterschieden sich die Vorgehensweisen der beiden Gruppen. Die *EG* hat die Plan- und Istwerte aller im Modell enthaltenen Variablen als CSV-Datei lokal auf dem Computer in einem bestimmten Pfad abgespeichert (vgl. Abbildung 6-1). Mit dem Befehl aus Listing 7-1 ist über die Kommandozeile der Observer gestartet worden, der wiederum die lokal gespeicherte CSV-Datei eingelesen und die Abweichungsanalyse begonnen hat.

```
python -m mosyco -v -logfile
```

Listing 7-1: Anweisung zum Ausführen des Observer

Quelle: Eigene Darstellung

Im Falle einer Abweichung führte der *Model-Controller* automatisiert die Ursachenidentifikation durch und informierte den Probanden über die Ergebnisse der Anpassungen über eine Ausgabe in der Kommandozeile (vgl. Abbildung 6-21) oder über die Visualisierungsmöglichkeiten von neo4j (vgl. Abbildung 6-4). Mit jeder Ausführung des Prototyps ist die Zeit vom Start der Abweichungsanalyse bis zum Output eines Ergebnisses gemessen und dokumentiert worden.

Bei der *KG* lief die Ursachenanalyse aufgrund des alternativen Systems Excel anders ab. Die Daten aus der CSV-Datei wurden in ein Tabellenblatt übertragen. Die Daten wurden im Anschluss aufbereitet, wobei die Probanden den Zeitraum identifizieren mussten, in dem eine Abweichung vorgelegen hat. Der Zeitraum konnte entweder durch eine visuelle Analyse in Vensim (vgl. Abbildung 7-3) oder auf Basis von Differenzen in Excel identifiziert werden. Es gibt keine Vorschrift, wie eine Testperson bei der Lösung einer Aufgabe vorzugehen hat, jedoch wurden im Rahmen der Einführungsveranstaltung die Möglichkeiten aufgezeigt (vgl. Kapitel 7.2.4). Die Studierenden müssen bestimmen, welcher Zeitraum

analysiert wird, welche Variablen auf einen potentiellen Einfluss untersucht werden und welche Methode zur Schätzung einer Modellgleichung eingesetzt wird. Wie auch bei der Nutzung des Prototyps ist für jede Abweichungsanalyse bis zum endgültigen Anpassungsvorschlag die Zeit gemessen und protokolliert worden.

Anschließend haben die Probanden die Ergebnisse aus der Abweichungs- und Ursachenanalyse in Vensim verwertet, indem nach einer Abweichung eine manuelle Anpassung am proprietären Modell vorgenommen wurde (vgl. Abbildung 6-1). Die Vorgehensweise bei dieser Teilaufgabe ist bei beiden Gruppen identisch. Das Modell in Vensim konnte entweder durch eine Parameter- oder Strukturanpassung verändert werden. Nach einer erfolgten Anpassung des Modells müssen die Planwerte aktualisiert werden, da die Annahmen des Modells sich geändert haben und sich somit auch die zukünftige Entwicklung ändert. Nachdem die Planwerte aktualisiert wurden, wiederholt der Proband den beschriebenen Vorgang und wählt einen Zeitpunkt für das Folgejahr aus, um im Anschluss erneut eine Abweichungs- und ggf. Ursachenanalyse durchzuführen.

Während des Experimentes sind die Teilnehmer von zwei Personen beobachtet worden. Es sind Beobachtungsprotokolle über die Herangehensweisen und Äußerungen der Studierenden, über die Dauer einzelner Teilaufgaben sowie mögliche Auffälligkeiten geführt worden. Dabei kann es sich sowohl um menschliche als auch um technische Auffälligkeiten handeln.

Das Experiment ist für eine Vergleichbarkeit zeitlich begrenzt, sodass die Teilnehmer jeder Gruppe eine Stunde Zeit haben, um alle zehn Jahre iterativ zu durchlaufen. Das Experiment ist beendet, sobald die Abweichungs- und Ursachenanalyse des zehnten Jahres abgeschlossen oder die einstündige Experimentdauer abgelaufen ist. Abschließend haben die Probanden Teil B des Fragebogens beantwortet (vgl. Tabelle 7-1).

Für die KG war das Experiment nach der Beantwortung des zweiten Fragebogenteils beendet. Die Probanden aus der EG sind nach der Umfrage zusätzlich im Rahmen eines Interviews befragt worden. Den Probanden wurden halb offene Fragen und zusätzlich ggf. situationsabhängige Fragen gestellt. Die Interviews sind transkribiert und ausgewertet worden (vgl. Anhang X).

7.4 Untersuchungsergebnisse

Nach erfolgreicher Durchführung der Untersuchung werden in diesem Kapitel die Ergebnisse analysiert. Die zur Datenauswertung genutzten Methoden werden zu Beginn vorgestellt (vgl. Kapitel 7.4.1). Darauf aufbauend werden die Ergebnisse in Abhängigkeit der Hypothesen detailliert beschrieben (vgl. Kapitel 7.4.2) und abschließend in Kapitel 7.4.3 zur Identifikation von Problemen und Verbesserungsmöglichkeiten interpretiert.

7.4.1 Vorgehen bei der Datenauswertung

Da im Rahmen der Evaluation sowohl qualitative als auch quantitative Daten erhoben wurden, werden unterschiedliche Methoden zur Auswertung benötigt.

Die *qualitative* Teilstudie umfasst die Beobachtungen während des Usability-Tests sowie das teilstrukturierte Interview. Die Beobachtungen werden von zwei Personen unabhängig voneinander geführt. Dadurch können Beobachtungsfehler in Form von Interpretations- und Erinnerungsfehlern vermieden werden (vgl. Döring und Bortz 2016a, S. 331), da fehlende Informationen durch die zweite Person ergänzt werden können. Außerdem kann das Risiko einer Fehlinterpretation durch eine hohe Subjektivität bei mehreren beobachtenden Personen verringert werden. Durch einen Abgleich der Beobachtungsprotokolle kann ein objektives Verständnis der Situationen im Usability-Test erfasst werden, damit möglichst vollständig das Nutzerverhalten repräsentiert werden kann. Ergänzend zu den Beobachtungen werden Interviews geführt, welche nachträglich transkribiert werden. Der Fokus der Interviews liegt auf thematisch-inhaltlichen Informationen, weshalb die Inhalte nur wörtlich transkribiert werden und zusätzliche Informationen wie die Betonung oder die Stimmlage vernachlässigt werden (vgl. Mayring 2016, S. 89 ff.).

Die *quantitative* Teilstudie umfasst den Fragebogen und die Protokolle über die Zeiten zur Aufgabenerfüllung. Die Fragen und Aussagen in der Umfrage wurden anhand einer fünfstufigen Likert-Skala bewertet, welche von 1 (Trifft nicht zu) bis 5 (Trifft zu) skaliert ist (vgl. Schnell 2005, S. 187 f.). Auf Basis der Antworten können deskriptive Statistiken über die Lage- und Streuungsmaße sowie die Zusammenhänge erhoben werden. Dieselben Statistiken können für die Analyse der Bearbeitungsdauer durchgeführt werden.

7.4.2 Interpretation der Ergebnisse

In diesem Kapitel werden die fünf aufgestellten Hypothesen aus Kapitel 7.1 untersucht. Zur Untersuchung der Effektivität (vgl. Kapitel 7.4.2.1), der Effizienz (vgl. Kapitel 7.4.2.2)

sowie der Sicherheit über die Richtigkeit der Modellanpassungen (vgl. Kapitel 7.4.2.5) werden die Ergebnisse der beiden Versuchsgruppen gegenübergestellt. Die Voraussetzung zur Untersuchung der Hypothesen über die Zufriedenheit der Modellanpassungen des Prototyps und das Vertrauen in automatisierte Systeme setzt die Nutzung des Prototyps voraus. Aus diesem Grund werden für die Hypothesen H7-3 (vgl. Kapitel 7.4.2.3) und H7-4 (vgl. Kapitel 7.4.2.4) lediglich die Teilnehmer aus der EG betrachtet.

7.4.2.1 Hypothese H7-1: Effektivität

Die Hypothese H7-1 besagt, dass ein Anwender mit Hilfe des Prototyps die Abweichungs- und Ursachenanalyse effektiver als mit dem Alternativsystem Excel durchführen kann. Die Effektivität wird anhand der Kriterien *Erfüllungsgrad der Aufgaben* und *Genauigkeit der Ergebnisse* bewertet. Da es sich um ein kontrolliertes Experiment handelt, bei dem die Störfaktoren im Realitätsmodell vordefiniert sind (vgl. Tabelle 7-2), können die ermittelten Lösungen der EG sowie KG mit der Musterlösung verglichen werden.

Der *Erfüllungsgrad einer Aufgabe* hängt davon ab, wie viele Teilnehmer der EG und der KG innerhalb der vorgegebenen Zeit die gesamte Aufgabe abschließen konnten. Die Aufgabe im Usability-Test gilt als erfüllt, wenn ein Teilnehmer jede Abweichungs- und ggf. Ursachenanalyse bis zum zehnten Jahres innerhalb der vorgegebenen Zeit von einer Stunde abgeschlossen hat. In der EG haben sieben von sieben (100%) die Aufgabe erfolgreich abgeschlossen. In der KG hat keiner (0%) der sechs Probanden die Aufgabe innerhalb einer Stunde vollständig bis zum zehnten Jahr erfüllen können. Somit ist der Erfüllungsgrad der Aufgabe in der Gruppe, die den *MoSyCo* Prototyp nutzt, deutlich höher im Vergleich zur Kontrollgruppe.

Die *Genauigkeit der Ergebnisse* wird als zweites Bewertungskriterium für die Effektivität verwendet. Hierfür werden die Anpassungsvorschläge aus beiden Gruppen mit der Musterlösung verglichen. Somit kann eine Fehlerquote ermittelt werden. Als Referenz zum Vergleich mit der Musterlösung wird das Ergebnis nach der Ursachenanalyse verwendet, d. h. die Ergebnisse der Regression aus Excel oder des *MoSyCo* Prototyps und nicht das durch den Anwender angepasste Modell in Vensim. Beim Übertrag der Regressionsergebnisse ist es nämlich teilweise zu Fehlern gekommen, womit nicht mehr die Effektivität des Prototyps oder die von Excel untersucht werden würde. Die Aufgabe im Rahmen des Usability-Tests war es, in jedem der zehn Jahre eine Abweichungs- und ggf. Ursachenanalyse durchzuführen, womit der Prototyp bei sieben Probanden insgesamt 70 Mal ausgeführt werden

sollte. Faktisch geschah dies jedoch nur 69 Mal, da ein Teilnehmer durch eine eigenständig in Vensim durchgeführte Analyse keine Abweichung identifiziert hat und somit den Prototyp nicht ausgeführt hat. Da die Identifikation von Abweichungen automatisiert erfolgen soll, ist dieses Vorgehen nicht intendiert.

Bei 69 Ausführungen hat der Prototyp den Probanden aus der Experimentalgruppe insgesamt 43 Modellanpassungen vorgeschlagen, da jeweils eine signifikante Abweichung identifiziert worden ist. In den anderen Fällen lag keine Abweichung vor oder diese war innerhalb eines Toleranzbereiches. Im Rahmen des Experimentes wurde eine Abweichung von 5% toleriert. Die schrittweise Regression prüft für die fokussierte *Nachfrage* die freien Variablen *Substitutionsmarktwachstum*, *Import*, *Wetter* und *BIP* (vgl. Abbildung 6-2) auf ihren Erklärungsbeitrag. Der Filter auf nur potentielle Variablen soll von den Teilnehmern nicht verwendet werden (vgl. Kapitel 6.5.3). Die Kontrollgruppe mit Excel hat ebenfalls die Information, dass die vier freien Variablen mögliche neue Regressoren für eine Strukturanpassung darstellen. Bei 43 Anpassungsvorschlägen und zwei Variablen (*DAX* und *Export*) für Parameteranpassungen (vgl. Formel 7-1) sowie die vier Variablen für mögliche Strukturanpassungen ergeben sich 258 zu überprüfende Variablen. Verglichen wird die Wirkung jeder Variable auf die *Nachfrage*. Sofern eine Variable keinen Einfluss hat, wird eine Wirkung von null angenommen. Insgesamt stimmen in der EG 12 Werte nicht mit der Musterlösung überein. Die Genauigkeit des Prototyps liegt bei $(258-12) / 258 = 95,3\%$. Die zwölf Fehler lassen sich auf zwei unterschiedliche Ursachen zurückführen. Die Fehler können zum einen auf den verhaltensnachahmenden Ansatz zurückgeführt werden, bei dem das Verhalten eines Systems nachgeahmt wird, ohne die Wirkungsstruktur zu kennen (vgl. Bossel 2004a, S. 52 f.). Für bestimmte Zeitpunkte kann eine von der Musterlösung abweichende Wirkungsstruktur das gleiche Verhalten aufweisen, diese stellt jedoch keine langfristig valide Lösung dar (vgl. Bossel 2004a, S. 56). In diesem Fall kann das Modellverhalten kurzfristig dem der Realität entsprechen, die Wirkungsstruktur des Modelles wird jedoch für die Interpretation als falsch gewertet. Zum anderen kann die Wahl des Zeitpunktes einen Einfluss auf die Güte einer Modellanpassung haben. Manche Störungen entwickeln sich bspw. über 100 Zeitpunkte (vgl. Tabelle 7-2), womit auch die Wirkung über die 100 Zeitpunkte nie gleich ist. In Abbildung 7-6 wird eine solche Entwicklung präsentiert.



Abbildung 7-6: Wirkungsänderung des Export innerhalb Störung 2 (vgl. Tabelle 7-2)

Quelle: Eigene Darstellung

Soll bspw. zum Zeitpunkt 900 eine Regressionsgleichung geschätzt werden (vgl. Abbildung 7-6), stößt die gewählte lineare Regression und damit der Prototyp an methodische Grenzen. Dieser Fall ist während des Experimentes eingetreten und für 100 unterschiedliche Wirkungen muss ein repräsentativer Wert gefunden werden. Ein ähnlicher, wenn auch entgegengesetzter, Fall ist während des Experimentes ebenfalls eingetreten. Die Testperson wählte den Zeitpunkt 1489 aus, wobei die Störung erst ab 1470 wirkte und die Veränderung sich bis zum Zeitpunkt 1490 erstreckt (vgl. Tabelle 7-2). Aufgrund der geringen Anzahl an Werten (19 Datenpunkte) und wiederholt aufgrund der sich verändernden Wirkung, konnte der Prototyp kein valides Modell ermitteln, das ein Bestimmtheitsmaß von über 50% aufweisen konnte. Die Testperson wurde über das invalide Modell und damit über den nicht erfolgreichen Anpassungsversuch informiert (vgl. Kapitel 6.5.5).

Die Anzahl der Modellanpassungen bei der *Kontrollgruppe* variiert im Vergleich zur Experimentalgruppe aufgrund der zeitlichen Begrenzung und der Häufigkeit von Fehlanpassungen. Insgesamt sind 22 Ursachenanalysen mit Excel zur Modellanpassung von den sechs Probanden in der KG durchgeführt worden, womit 132 Variablen verglichen werden müssen. Bei 40 Variablen stimmen die Werte nicht mit denen aus der Musterlösung überein. Daraus ergibt sich eine Genauigkeit von $(132-40) / 132 = 69,7\%$. Die fehlerhaften Werte sind auf die Auswahl des zu analysierenden Zeitraums und der unabhängigen Variablen zurückzuführen. Die Wahl der unabhängigen Variablen hat einen starken Einfluss auf die Güte einer Regression (vgl. Fahrmeir et al. 2016b, S. 437 f.) und kann bei Fehlern dazu führen, dass ein Modell weder verhaltens- noch strukturgetreu ist (vgl. Bossel 2004a, S. 52 ff.). Äquivalent zur EG kann ein verhaltensgetreues Modell kurzfristig korrekt sein, es wird jedoch aufgrund der inkorrekten Wirkungsstruktur und damit aufgrund der mangel-

den langfristigen Repräsentativität als falsch bewertet. Innerhalb der Kontrollgruppe fiel auf, dass häufig Strukturanpassungen von den Probanden durchgeführt wurden, obwohl eine Änderung der Parameter ausgereicht hätte. Konkret sind bei 9 der 22 Ursachenanalysen Veränderungen an der Modellstruktur vorgenommen worden. Lediglich eine der Anpassungen ist notwendig und korrekt gewesen. Im Vergleich zu den Ergebnissen des Prototyps wird deutlich, dass die Lösungsgüte stark von der Art der Anpassung abhängt. Die Probanden der EG haben 14 Strukturanpassungen vorgenommen, von denen lediglich eine falsch gewesen ist. Die Genauigkeit der Ergebnisse bei Strukturanpassungen ist innerhalb der EG wesentlich besser.

Anhand der ermittelten Genauigkeit von 95,3% des Prototyps und von 69,7% des Alternativsystems Excel wird deutlich, dass der *MoSyCo* Prototyp zuverlässigere Modellanpassungsvorschläge ausgibt. Jedoch unterscheiden sich die von den Probanden ausgewählten Zeitpunkte, da gemäß der Aufgabenstellung der Zeitpunkt innerhalb eines Jahres frei gewählt werden darf. Wie bereits erläutert, kann die Wahl des Zeitpunktes einen maßgeblichen Einfluss auf die Güte eines Regressionsergebnisses haben. Aufgrund dessen werden die 22 Ursachenanalysen aus der KG mit dem Prototyp nachträglich wiederholt. Zu jedem dieser 22 Zeitpunkte, die von den Teilnehmern der KG ausgewählt wurden, wird der Prototyp ausgeführt. Da zu einem Zeitpunkt die *Nachfrage* um weniger als 5% abweicht, gibt der Prototyp lediglich 21 Anpassungsvorschläge aus. Daraus ergeben sich insgesamt 126 zu untersuchende Variablen. Drei Werte stimmen nicht mit der Musterlösung überein, was einer Genauigkeit von 97,6% entspricht. Einer dieser Fehler ist auf eine inkorrekte Anpassung der Struktur zurückzuführen. Auch bei dem direkten Vergleich zu den selben Anpassungszeitpunkten sind die Ergebnisse des *MoSyCo* Prototyp im Vergleich zum Alternativsystem Excel zuverlässiger und die Genauigkeit ist höher.

Durch den Vergleich der EG und der KG wird deutlich, dass die *Genauigkeit der Ergebnisse* aber auch der *Grad der Erfüllung der Aufgabe* seitens des Prototyps höher ist. Auf Grundlage dessen kann die Hypothese H7-1 akzeptiert werden: Mit Hilfe des *MoSyCo* Prototyps können Abweichungs- und Ursachenanalysen effektiver umgesetzt werden als mit dem Alternativsystem Excel.

7.4.2.2 Hypothese H7-2: Effizienz

Die Hypothese H7-2 besagt, dass ein Anwender mit dem Prototyp Abweichungsanalysen zwischen den Plan- und Istwerten sowie ggf. eine Ursachenanalyse effizienter als mit dem

Alternativsystem Excel durchführen kann. Die Zeit, die je Abweichungs- und Ursachenanalyse benötigt wird, bildet die Grundlage zum Vergleich der beiden Gruppen. Hierbei handelt es sich um ein objektives Maß, anhand dessen die Geschwindigkeit bei der Aufgabenlösung gemessen wird. Darüber hinaus wird die subjektive Wahrnehmung der Teilnehmer herangezogen, um abschließend die Hypothese H7-2 zu beurteilen.

Die Zeit zur Erfüllung ist je Teilnehmer gemessen worden, maximal sind 60 Minuten möglich. Alle zehn Jahre konnten von der EG zwischen 35 und 57 Minuten bearbeitet werden. Im Mittel haben die Teilnehmer in der EG 48 Minuten benötigt. Im Vergleich dazu hat keiner der Teilnehmer aus der KG die Aufgaben innerhalb der vorgegebenen Zeit erfüllen können (vgl. Kapitel 7.4.2.1), somit arbeitet die EG mit dem Prototyp effizienter als die KG. Dieser Effekt ist maßgeblich auf den teilautomatisierten Prozess des Prototyps zurückzuführen.

Die Störungen sind im Rahmen des Experimentes so konzipiert, dass die Komplexität mit jeder zusätzlichen Störung ansteigt. Somit kann es dazu führen, dass bei späteren Ursachenanalysen aufgrund der gestiegenen Komplexität auch die Bearbeitungsdauer steigt. Aufgrund dessen ist es sinnvoll, die benötigte Zeit je Störung und Ursachenanalyse zu untersuchen. Bei der EG wurde die Ausführungszeit des Prototyps gemessen. Die Zeit stieg pro Jahr an. Im ersten Jahr ist bspw. nach acht Sekunden ein Ergebnis ausgegeben worden, während im zehnten Jahr bereits 80 Sekunden benötigt wurden. Die Ursache für die steigende Ausführungszeit pro Jahr ist, dass der *Observer* mit jeder Ausführung beim Zeitpunkt $t = 0$ die Analyse beginnt. Für einen operativen Einsatz müsste der Zeitraum, der in der Vergangenheit liegt, nicht nochmals analysiert werden. Im Durchschnitt hat eine Ursachenanalyse bei den Teilnehmern in der EG zwischen 33 und 42 Sekunden gedauert. Der Median liegt bei 35 Sekunden. Bei den Teilnehmern der KG ist die Zeit nach der Datenübertragung in Excel bis zur letzten Regressionsanalyse, sofern mehrere durchgeführt wurden, gemessen worden. Bei der KG schwankt die durchschnittliche Zeit im Vergleich zur EG wesentlich stärker. Das Intervall liegt zwischen 4 und 14 Minuten. Der Gruppenmittelwert liegt bei ca. 8 Minuten und der Median bei 7 Minuten und 20 Sekunden. Die große Bandbreite resultiert durch unterschiedliche Vorgehensweisen seitens der Teilnehmer. Manche Probanden haben bspw. eine Korrelationsanalyse zur Identifikation relevanter Faktoren zur Bestimmung der *Nachfrage* durchgeführt. Die korrelierten Variablen sind jedoch nur Indikatoren für die Auswahl der unabhängigen Variablen bei einer Regressions-schätzung, sodass manche Teilnehmer auf diesen Schritt verzichtet haben. Stattdessen sind

die unabhängigen Variablen nach Gefühl ausgewählt worden. Teilweise sind mehrere Regressionsanalysen nacheinander ausgeführt worden, um diverse unabhängige Variablen zu prüfen. Dieses Vorgehen ähnelt dem Erstellen der unterschiedlichen Experimentalmodelle des Prototyps. Das manuelle Erstellen der Regression benötigt jedoch mehr Zeit. Darüber hinaus beansprucht die Kontrolle, Interpretation und Analyse der Ergebnisse viel Zeit bei den Teilnehmern in der KG. Diese Erkenntnis konnte primär durch die Methode des lauten Denkens gewonnen werden. Die große Bandbreite der benötigten Zeit resultiert also aus unterschiedlichen Vorgehensweisen bei der Abweichungs- und Ursachenanalyse sowie aus detaillierten Ergebniskontrollen.

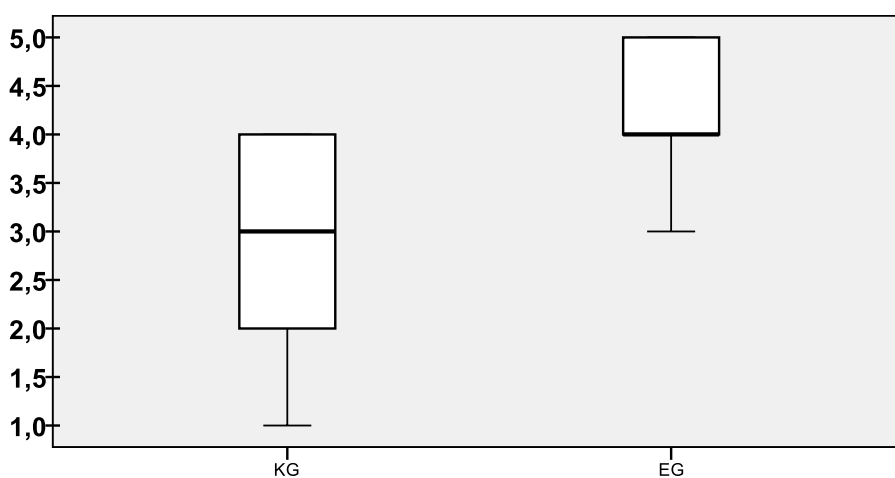


Abbildung 7-7: Boxplot zur Variable MA1_SQ002 gruppiert nach EG / KG

Quelle: Eigene Darstellung

Neben den objektiven Kriterien über die benötigte Zeit zur Aufgabenerfüllung sind die Probanden nach der Effizienz (MA1_SQ002) gefragt worden. Dem Boxplot aus Abbildung 7-7 ist zu entnehmen, dass der Median bei der EG bei 4 liegt, und der der KG bei 3. Die Teilnehmer der EG nehmen durch den Prototyp eine größere Unterstützung bei der Schnelligkeit der Aufgabenerfüllung wahr. Auch die Streuung der Antworten ist geringer im Vergleich zur KG (vgl. Abbildung 7-7).

Anhand des objektiven Kriteriums der benötigten Zeit zur Aufgabenerfüllung sowie des subjektiven Kriteriums der wahrgenommenen Schnelligkeit kann die effizientere Arbeitsweise des Prototyps bestätigt werden und somit auch die Hypothese H7-2. Mithilfe des Prototyps können Abweichungs- und Ursachenanalysen effizienter als mit dem Alternativsystem Excel durchgeführt werden.

7.4.2.3 Hypothese H7-3: Zufriedenheit

Zur Beurteilung der Zufriedenheit mit den jeweiligen Systemen sind zwei Aussagen formuliert worden. Dabei wird die Zufriedenheit mit der Leistung zum einen bei der Identifikation von Abweichungen (MA1_SQ006) und zum anderen bei dem Erstellen von Anpassungsvorschlägen (MA1_SQ007) innerhalb der EG untersucht. Die beiden Prozesse hängen zwar unmittelbar miteinander zusammen, jedoch repräsentieren diese zwei unterschiedliche Aufgaben. Das Erstellen von Anpassungsvorschlägen bezieht sich nicht nur auf die Berechnung richtiger Werte, sondern auch auf die Darstellung der Ergebnisse.

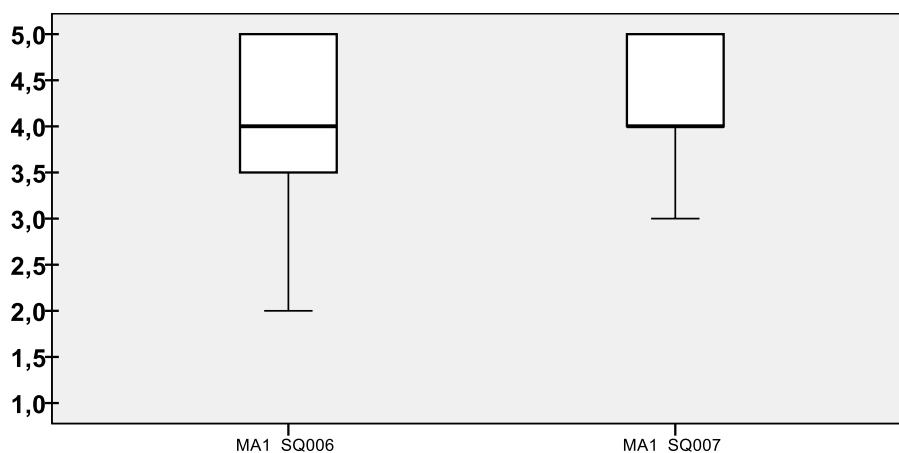


Abbildung 7-8: Boxplot der Variablen MA1_SQ006 und MA1_SQ007 in der EG

Quelle: Eigene Darstellung

Die Zufriedenheit mit der Leistung des Prototyps beim Analysieren von Abweichungen sowie beim Erstellen von Anpassungsvorschlägen wurde jeweils mit einem Median von 4,0 bewertet (vgl. Abbildung 7-8). Die Teilnehmer sind im Durchschnitt mit der Leistung des Prototyps zufrieden. Lediglich ein Proband ist mit der Leistung der Abweichungsanalyse weniger zufrieden und hat die Aussage mit 2 und demnach als eher unzutreffend bewertet. Da der *Observer* zur Abweichungsanalyse nach jedem Jahr erneut ausgeführt werden muss, könnte dies eine Erklärung für die Unzufriedenheit sein. Eine andere mögliche Erklärung könnte sein, dass der *MoSyCo* Prototyp während einer Abweichungs- und Ursachenanalyse abgestürzt ist. In diesem Fall ist das berechnete Modell nicht ausreichend valide gewesen, da das R^2 bei unter 50% lag. Aufgrund eines Fehlers im Quellcode ist der Prototyp hierbei abgestürzt. Auch wenn die restlichen Abweichungs- und Ursachenanalysen bei jenem Probanden einwandfrei funktionierten, kann dieser Misserfolg eine Auswirkung auf die wahrgenommene Leistung gehabt haben. Nach dem Abschluss des gesamten Experimentes ist der Fehler im Quellcode korrigiert worden.

Als Ergänzung zu den Ergebnissen aus der Umfrage sind die Probanden nach ihrer Meinung bezüglich der gelieferten Informationen und der Darstellung der Ergebnisse befragt worden. Die Präferenzen über die Form der Darstellung von Ergebnissen unterscheidet sich zwischen den Probanden. Einerseits bevorzugen einige Teilnehmer die graphische Darstellung in neo4j (vgl. Abbildung 6-22) aufgrund ihrer Einfachheit und Übersichtlichkeit. Andererseits halten andere Teilnehmer die tabellarische Darstellung (vgl. Abbildung 6-21) für besser geeignet, da sie alle Informationen in einer Tabelle ablesen können. Das Format der Outputs stimmt größtenteils mit den Outputs gängiger Statistiksoftware wie SPSS, EViews und Stata überein. Dabei handelt es sich um Programme, die den Probanden aus dem Studium bekannt sind. Aus diesem Grund könnten sie dieses Format gewöhnt sein und deshalb bevorzugen. Neben dem Format ist darüber hinaus interessant, ob die in der Ausgabe gelieferten Informationen für den Anwender ausreichend und nachvollziehbar sind. Der Anwender soll alle relevanten Informationen zur Regressionsanalyse und zu den Anpassungsvorschlägen erhalten. Durch das Interview konnte festgestellt werden, dass die Probanden mit der Darstellungsform des Outputs zufrieden sind und die darin enthaltenen Informationen nachvollzogen werden können (vgl. Anhang X: PE1, Z. 28; PE2, Z. 125; PE3, Z. 43 f.; PE4, Z. 34; PE5, Z. 79; PE6, Z. 47; PE7, Z. 29 f.).

Darüber hinaus sind die wahrgenommene Nützlichkeit sowie die wahrgenommene Leichtigkeit untersucht worden, da sie gemeinsam zu einer positiven Einstellung gegenüber der Systemnutzung führen können (vgl. Kapitel 7.1.1). Diesbezüglich sind die Probanden befragt worden, ob sie das System für das gleiche Problem wieder nutzen würden (VA1_SQ003). Der Median zu dieser Aussage liegt bei 4,0. Somit ist eine Nutzungsabsicht bei den Probanden vorhanden und sie sind positiv gegenüber dem *MoSyCo* Prototyp eingestellt, sodass sie ihn erneut für eine Abweichungs- und Ursachenanalyse nutzen würden. Für die Teilnehmer erbringt der Prototyp zur Aufgabenerfüllung einen Mehrwert und stellt eine Unterstützung dar. In den Interviews wird deutlich, dass die Probanden mit der Bedienung zufrieden sind.

Zur Untersuchung der Hypothese H7-3 sind zusammengefasst die Ergebnisse aus den Interviews und der Umfrage betrachtet worden. Nach der Auswertung beider Evaluationsteile sind die Teilnehmer im Durchschnitt mit der Leistung des Prototyps zur Identifikation von Abweichungen und zur Erstellung von Anpassungsvorschlägen zufrieden. Die Hypothese H7-3 kann somit bestätigt werden. Basierend auf diesem Ergebnis soll im Rahmen der vorliegenden Arbeit untersucht werden, welche Faktoren für eine Zufriedenheit mit der Leis-

tung des Prototyps relevant sind. Hierfür wird eine Korrelationsanalyse auf Basis der Antworten in der EG durchgeführt und es werden entsprechend die Korrelationskoeffizienten ρ untersucht. Die Interpretation orientiert sich an den etablierten Grenzen: Ein $|\rho|$ von kleiner 0,5 entspricht einer schwachen Korrelation, ein absoluter Wert zwischen 0,5 und 0,8 entspricht einem mittelstarken Zusammenhang und ein $|\rho|$ von größer 0,8 einer starken Korrelation (vgl. Fahrmeir et al. 2016a, S. 130).

Die Zufriedenheit mit der Leistung der Anpassungsvorschläge des Prototyps (MA1_SQ007) ist in der ersten Spalte der Tabelle 7-3 abgetragen, sodass der Einfluss der anderen Faktoren in den jeweiligen Zeilen der ersten Spalte geprüft werden kann.

Tabelle 7-3: Korrelationsmatrix über die Zufriedenheit der Leistung des Prototyps

Quelle: Eigene Tabelle

	MA1_SQ007	MA1_SQ001	MA1_SQ002	MA1_SQ003	MA1_SQ004	MA1_SQ005	MA1_SQ006	MA1_SQ008	MA1_SQ009
MA1_SQ007	1								
MA1_SQ001	0,88	1							
MA1_SQ002	1,00	0,88	1						
MA1_SQ003	-0,68	-0,78	-0,68	1					
MA1_SQ004	0,88	0,94	0,88	-0,71	1				
MA1_SQ005	-0,47	-0,62	-0,47	0,71	-0,42	1			
MA1_SQ006	0,76	0,58	0,76	-0,14	0,54	-0,27	1		
MA1_SQ008	0,04	0,00	0,04	0,31	-0,11	0,11	0,55	1	
MA1_SQ009	0,33	0,18	0,33	0,12	0,05	-0,05	0,76	0,92	1

Legende: | ρ | > 0,8 0,5 ≤ | ρ | < 0,8 | ρ | < 0,5

Der größte Zusammenhang besteht mit der Variable MA1_SQ002, wonach die Zufriedenheit dann hoch ist, wenn auch die Aussage über die wahrgenommene Schnelligkeit der Abweichungs- und Ursachenanalyse hoch bewertet wurde. Demnach wirkt sich eine effiziente Unterstützung des Anwenders positiv auf dessen Zufriedenheit mit der Leistung aus. Ähnlich positiv korreliert sind die Variablen MA1_SQ001 und MA1_SQ004 mit jeweils einem Koeffizienten von 0,88. Die Einfachheit der Nutzung (MA1_SQ001) sowie die problemlose Funktionsweise erhöhen demnach die Zufriedenheit der Anwender mit den Anpassungen. Die Variable MA1_SQ003 weist einen negativen Korrelationskoeffizienten auf, dessen Wirkung als moderat bewertet werden kann. Je weniger komplex die Teilnehmer die Identifikation von Ursachen wahrnehmen, desto höher ist die Zufriedenheit mit der Analyse des Prototyps. Die letzte Variable, die eine mittelhohe Korrelation mit der Zufriedenheit aufweist, ist die Zufriedenheit mit der Abweichungsanalyse (MA1_SQ006). Wenn

die Zufriedenheit mit der Leistung der Abweichungsanalyse hoch ist, ist ebenfalls die Zufriedenheit mit den Anpassungsvorschlägen hoch. Potentielle Gründe für eine schlechter wahrgenommene Zufriedenheit mit der Abweichungsanalyse sind bereits in Kapitel 7.4.2.2 diskutiert worden. Der Zeitaufwand zur Modellanpassung (MA1_SQ005), die Arbeitserleichterung (MA1_SQ008) sowie Sicherheit über korrekte Modellanpassungen haben keinen signifikanten Einfluss auf die Zufriedenheit. Insbesondere die Arbeitserleichterung durch eine automatische Modellanpassung ist nahezu unkorreliert, da die Anpassung im proprietären Modell vom Probanden selbst durchgeführt werden muss und dementsprechend keine Wirkung auf die Zufriedenheit gemessen werden kann.

7.4.2.4 Hypothese H7-4: Vertrauen in automatisierte Systeme

Die Hypothese H7-4 untersucht das Vertrauen in automatisierte Systeme, wie es der *MoSyCo* Prototyp darstellen soll. Die Hypothese besagt, dass das Vertrauen in automatisierte Systeme nach der Nutzung des *MoSyCo* Prototyps erhöht ist. Zur Überprüfung der Hypothese wird ausschließlich die EG betrachtet, da nur diese mit einem automatisierten System gearbeitet hat. Die Teilnehmer haben im Rahmen der Umfrage die Aussage „*Ich vertraue automatisierten (selbstständig handelnden) computergestützten Systemen*“ anhand der fünfstufigen Likert-Skala zweimal bewertet. Im Teil A (EM_SQ001) vor dem Experiment und im Teil B (PD3_SQ004) nach dem Experiment.

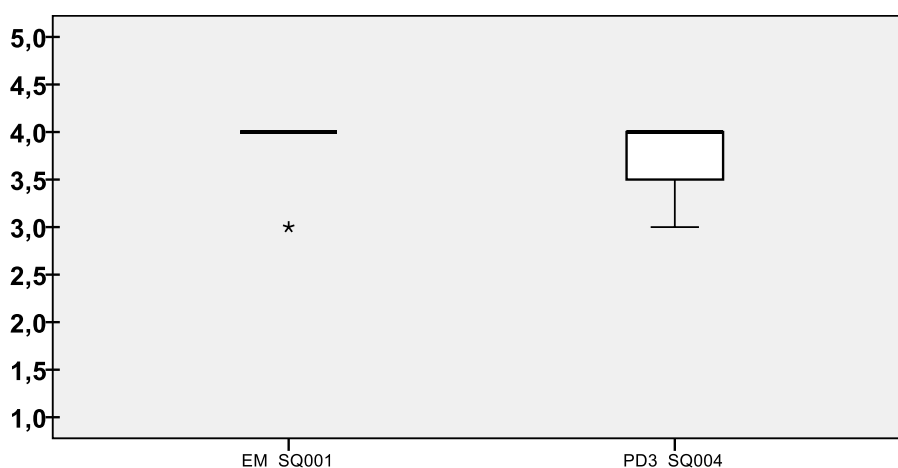


Abbildung 7-9: Boxplot der Variablen EM_SQ001 und PD3_SQ004

Quelle: Eigene Darstellung

Der Median beider Aussagen liegt bei 4,0. Dennoch ist in Abbildung 7-9 zu erkennen, dass sich die Antworten unterscheiden. So haben im ersten Teil der Umfrage sechs von sieben Teilnehmern die Aussage mit 4 (trifft eher zu) bewertet und ein Teilnehmer mit 3 (teils

teils). Aufgrund dieser Verteilung stellt der Wert 4 gleichzeitig das 25%-, 50%- sowie 75%-Quantil dar. Nach dem Experiment sinkt das 25%-Quantil auf 3,5, da ein weiterer Teilnehmer die Aussage mit 3 bewertet hat. Somit ist das Vertrauen der Teilnehmer innerhalb der EG in automatisierte Systeme leicht gesunken. Im Rahmen der Nachbesprechung sind mögliche Gründe über ein geringeres Vertrauen diskutiert worden. Die Testperson konnte zwar den Zweck und die Funktion des Prototyps nachvollziehen, allerdings ist der Person die Berechnung von neuen Modellgleichung nicht transparent genug (vgl. Anhang X: PE4, Z. 63). Der Output nach einer Modellanpassung verfügt zwar über Informationen zur Regressionsbildung (vgl. Abbildung 6-21). Dazu zählen bspw. wie viele Datenpunkte zur Regression berücksichtigt wurden oder welche Schwellwerte zur Bewertung der Validität gelten (vgl. Kapitel 6.6). Jedoch könnten in Abhängigkeit eines gewünschten Detailgrades bspw. Informationen über die verworfenen Experimentalmodelle ausgegeben werden, um so die Auswahl des besten Modells transparent zu begründen. Das Ausgabeformat müsste im Vorfeld in Abhängigkeit der Präferenzen des Anwenders abgefragt werden, da im Rahmen des Experimentes nur eine Person eine Intransparenz wahrgenommen hat und die restlichen Probanden die Menge der Informationen als gut bewertet haben.

Abschließend kann aufgrund des im Verlaufe des Experimentes gesunkenen Vertrauens die Hypothese H7-4 nicht bestätigt werden. Das Vertrauen in automatisierte Systeme erhöht sich nach der Nutzung des *MoSyCo* Prototyps nicht.

7.4.2.5 Hypothese H7-5: Sicherheit über die Richtigkeit von Modellanpassungen

Die Hypothese H7-5 betrachtet das subjektive Urteil der Probanden über die Korrektheit von Modellanpassungen. Die Hypothese H7-5 besagt, dass die Sicherheit über die Richtigkeit von Modellanpassungen bei der EG höher ist als bei der KG. Zur Bewertung der Richtigkeit von Modellanpassungen stehen den Gruppen dieselben Gütekriterien einer Regression zur Verfügung: u. a. die t- sowie p-Werte und das R^2 -Maß. Alle Teilnehmer des Experimentes haben die Aussage „*Ich bin mir sicher, dass die Modellanpassungen korrekt sind*“ (MA1_SQ009) anhand der fünfstufigen Likert-Skala bewertet. In Abbildung 7-10 sind die Boxplots in Abhängigkeit der Gruppe dargestellt. Der Median der KG liegt bei 2,5 und in der EG bei 4. Somit sind sich die Teilnehmer in der EG im Durchschnitt sicherer, dass die Modellanpassungen richtig sind.

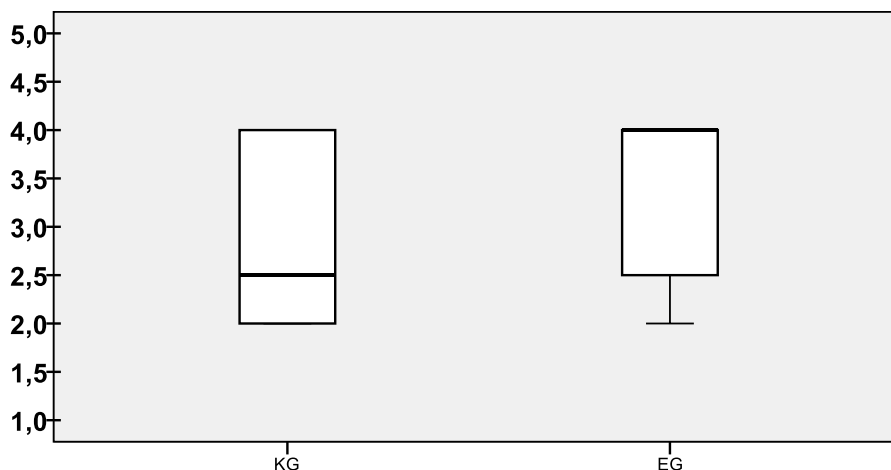


Abbildung 7-10: Boxplot der Variable MA1_SQ009 gruppiert nach EG / KG

Quelle: Eigene Darstellung

Eine mögliche Erklärung für das niedrigere Niveau der wahrgenommenen Sicherheit in der KG ist, dass innerhalb dieser Gruppe Abweichungen häufiger unmittelbar in der Folgeperiode nach einer Modellanpassung aufgetreten sind, obwohl keine neue Störung eingetreten ist. Demnach entsprechen die durchgeführten Anpassungen im Entscheidungsmodell nicht den Annahmen im Realitätsmodell. Durch die Methode des lauten Denkens konnte nachvollzogen werden, dass die Teilnehmer in der KG die inkorrekte Modellanpassung aufgrund der weiter bestehenden Abweichungen registriert haben. Jedoch konnten sie nicht den Fehler identifizieren, der für die Fehlanpassung verantwortlich ist und wussten demnach auch nicht, was bei der nächsten Ursachenanalyse verbessert werden kann. Die Unsicherheit über die Richtigkeit der Modellanpassungen blieb im Rahmen des Experimentes bestehen und infolge dessen ist der Median über alle Antworten von MA1_SQ001 auf dem niedrigen Niveau von 2,5.

Die Sicherheit über korrekte Modellanpassungen ist bei der EG höher als bei der KG (vgl. Abbildung 7-10). Trotzdem ist die Streuung innerhalb der Antworten der EG relativ groß und das 25%-Quantil liegt bei 2,5. Somit gibt es auch Probanden, die sich der Korrektheit der Ergebnisse des Prototyps nicht sicher sind. Aufgrund dessen bestätigen die Teilnehmer der EG die Aussage VA1_SQ004 mit einem Mittelwert von 3,7. Demnach fordern manche Teilnehmer mehr Informationen zum Anpassungsprozess (vgl. Tabelle 7-1). Trotzdem konnte im Rahmen des Experimentes aufgrund der Beobachtungen festgestellt werden, dass die Teilnehmer in der EG in den meisten Fällen die vorgeschlagenen Werte des *MoSyCo* Prototyps ohne Überprüfung auf Korrektheit zur Anpassung des proprietären Modells in Vensim übernommen haben. Ein grundlegendes Vertrauen muss bei den Ergebnissen

vorhanden sein, sodass die Werte ohne weitere Prüfung übernommen wurden. Bestätigt wird diese Annahme durch die Aussage VA1_SQ002, bei der die Teilnehmer angeben, dass die Überprüfung der Richtigkeit von Änderungsvorschlägen nicht viel Zeit gekostet hat (vgl. Abbildung 7-11).

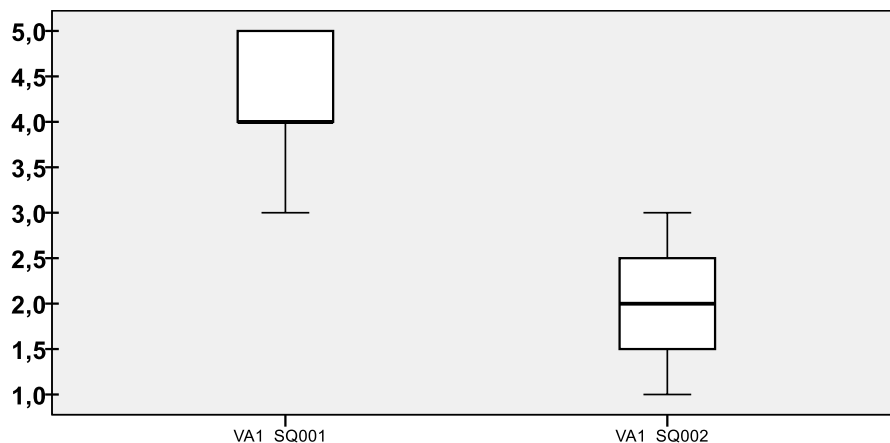


Abbildung 7-11: Boxplot der Variablen VA1_SQ001 und VA1_SQ002

Quelle: Eigene Darstellung

Im Rahmen des Interviews wurden die Probanden darüber hinaus gefragt, inwieweit sie glauben, dass es Fehlanpassungen vom Prototyp gegeben hat. Einerseits glauben die Teilnehmer nicht, dass die berechneten Ergebnisse des Prototyps fehlerhaft sind (vgl. Anhang X: PE1, Z. 36; PE2, Z. 132; PE3, Z. 61; PE6, Z. 124). Sie bestätigen, dass sie zur Überprüfung der Richtigkeit die Gütekriterien wie t- oder p-Werte analysieren können (vgl. Anhang X: PE1, Z. 45; PE2, Z. 114 f.; PE6, Z. 124). Ferner bewerten sie die Modellanpassung als gut, da nach einer Anpassung nicht erneut eine Abweichung festgestellt wurde. Andererseits können die Probanden nicht ausschließen, dass Fehler eingetreten sind. Dies liegt daran, dass lediglich im Falle einer Anpassung eine Ausgabe über die Kommandozeile erfolgt. Wird keine Abweichung gemessen, erfolgt keine Ausgabe, um eine Informationsüberflutung zu vermeiden. Durch die Antworten auf die Aussage VA1_SQ005 (vgl. Tabelle 7-1) wird mit einem Median von drei deutlich, dass auch im Falle einer Fehlanpassung am Modell das Vertrauen in den *MoSyCo* Prototyp nur teilweise eingeschränkt würde. Insgesamt scheinen die Menge und die Gestaltung von Informationen ausreichend zu sein, da sich die Teilnehmer der EG mit einem Median von 4 auf die Änderungsvorschläge verlassen (VA1_SQ001) und auch die Beobachtungen bestätigt haben, dass die Vorschläge des *MoSyCo* Prototyps im proprietären Vensim-Modell direkt umgesetzt wurden.

Zusammengefasst sind die Probanden in der KG aufgrund der häufig festgestellten Abweichungen verunsichert über die Qualität ihrer Anpassungen. Im Gegenteil dazu haben die Teilnehmer der EG die Anpassungsvorschläge des Prototyps ohne umfangreiche Prüfung umgesetzt. Deshalb ist die Überprüfung von Ergebnissen wenig zeitaufwändig (VA1_SQ002) und die EG nimmt im Durchschnitt eine höhere Sicherheit bei der Richtigkeit von Modellanpassungen im Vergleich zur KG wahr (vgl. Abbildung 7-10). Auf dieser Grundlage kann die Hypothese H7-5 akzeptiert werden, wenn auch die Informationsauswahl sowie -gestaltung aufgrund der Antworten zu VA1_SQ004 genauer untersucht werden sollten, um die methodische Transparenz und somit das Vertrauen in die Arbeitsweise zu erhöhen.

7.4.3 Zusammenfassung der Ergebnisse und Identifikation von Problemen und Verbesserungsmöglichkeiten

Im Rahmen der ex post Evaluation ist der implementierte Prototyp fokussiert worden. Die Untersuchung betrachtete die Anwendung des Prototyps zur Lösung einer Aufgabenstellung und führt einen Vergleich mit dem etablierten Alternativsystem Excel durch (vgl. Kapitel 7.2.3). Die Ergebnisse aller Hypothesentests sind in Tabelle 7-4 zusammengefasst.

Tabelle 7-4: Übersicht der Ergebnisse je Hypothese

Quelle: Eigene Tabelle

Nr.	Hypothese	Ergebnis
7-1	Mit Hilfe des Prototyps können Abweichungs- und Ursachenanalysen effektiver als mit einem funktional vergleichbaren System umgesetzt werden.	✓
7-2	Mit Hilfe des Prototyps können Abweichungs- und Ursachenanalysen effizienter als mit einem funktional vergleichbaren System umgesetzt werden.	✓
7-3	Nutzer des Prototyps weisen eine positive Einstellung gegenüber dem MoSyCo Prototyp auf.	✓
7-4	Das Vertrauen in automatisierte Systeme ist nach der Nutzung des Prototyps erhöht.	✗
7-5	Das Vertrauen über die Richtigkeit der Modellanpassungen vom Prototyp ist höher als bei einem funktional vergleichbaren System.	✓

Lediglich der Hypothesentest zur Hypothese H7-4 konnte nicht bestätigt werden. Die Ergebnisse sind insgesamt vielversprechend und zeigen das Potential des implementierten Prototyps auf. Durch die Überprüfung der Effektivität ist jedoch aufgefallen, dass die ein-

gesetzte Methode der linearen Regression bei sich stetig verändernden Parametern sowie wenigen Datenpunkten an ihre Grenzen stößt (vgl. Kapitel 7.4.2.1) und ebenfalls Ansätze wie die dynamische lineare Regression in der Methodenbank vorgehalten werden sollten. Dieser Ansatz zeichnet sich durch eine zeitabhängige Schätzung von Koeffizienten bspw. durch den Einsatz vom Kalman Filter aus (vgl. Petris et al. 2009, S. 31). Trotzdem ist die Genauigkeit der Ergebnisse des Prototyps insbesondere bei Strukturanpassungen deutlich höher als die innerhalb der KG (vgl. Kapitel 7.4.2.1), womit die Stärke des Prototyps bei der Erfüllung strukturell komplexer Aufgaben bestätigt wird (vgl. Kapitel 2.3). Durch die Interviews ist das Verbesserungspotential für die Ausgabe von Modellanpassungsvorschlägen identifiziert worden (vgl. Kapitel 7.4.2.3). Teilweise sind Strukturänderungen in der tabellarischen Form visuell von den Probanden nicht erfasst worden, sie haben sich auf die bereits bekannten Variablen fokussiert. Ein Lösungsansatz hierfür könnte sein, dass über die Kommandozeile ausschließlich Veränderungen dargestellt werden und nicht alle Einflussfaktoren, unter denen ggf. auch unveränderte Faktoren sind. Alternativ könnte das Ausgabeformat von neo4j verstärkt genutzt werden, bei dem Strukturänderungen durch neue Knoten eher auffallen. Darüber hinaus bietet die Darstellungsform von Knoten und Kanten die Möglichkeit, dass mehrere abgewichene Variablen sowie deren Regressoren zeitgleich dargestellt werden. Angelehnt an das Exception Reporting könnten Veränderungen durch farbliche Markierungen der Knoten und Kanten hervorgehoben werden (vgl. Gluchowski et al. 2008, S. 79). Eine tabellarische Ausgabe ist in diesem Fall zwar auch möglich, jedoch neigt dieses Format bei zunehmender Anzahl an Zeilen und Spalten dazu, unübersichtlich zu werden. Zum Thema des Ausgabeformats ist durch die Methode des lauten Denkens aufgefallen, dass die Probanden teilweise unsicher gewesen sind, ob noch eine Ausgabe und somit ein Anpassungsvorschlag folgt. Sofern keine Abweichungen gemessen werden, erfolgt auch keine Ausgabe (vgl. Kapitel 7.4.2.5). Die Unsicherheit wurde verstärkt, da das rechenintensive Forecasting (vgl. Kapitel 6.4.5) zu langen Ausführungszeiten des *Observer* führt (vgl. Kapitel 7.4.2.2). Dies kann durch eine kontinuierliche Abweichungsanalyse gelöst werden, bei der der *Observer* nicht jedes Mal gestartet werden muss. Durch die implementierten Warteschlangen ist der Prototyp für diese Arbeitsweise vorbereitet. Die beiden Probleme bzgl. des Ausgabeformats können umgangen werden, wenn der Prototyp wie konzipiert autonom und vollständig automatisiert arbeitet (vgl. Kapitel 4.2.4). Den Teilnehmern wurde im Rahmen des Interviews die autonome und vollautomatisierte Alternative vorgestellt. Sie haben diese Arbeitsweise grundsätzlich bevorzugt (vgl. Anhang X: PE1, Z. 135; PE2, Z. 183; PE3, Z. 111; PE4, Z. 88; PE5, Z. 124; PE6, Z.

165; PE7, Z. 155). Einerseits ist es eine Arbeitserleichterung für den Anwender und versehentliche Fehlanpassungen, die bspw. beim Übertrag zwischen generischem und proprietärem Modell auftreten können, können verhindert werden. Andererseits haben die Probanden die Bedenken geäußert, dass durch die Automatisierung der Überblick und die Kontrolle über die vorgenommenen Änderungen verloren gehen. Diesbezüglich haben sich die Probanden ein Reporting als Feedback über die durchgeführten Anpassungen gewünscht. Um das Kontrollproblem zu mindern, wird im Rahmen der Interviews vorgeschlagen, dass dem Anwender eine Möglichkeit gewährt werden soll, durch die er in den Anpassungsprozess eingreifen kann, um unerwünschte Anpassungen zu vermeiden (vgl. Anhang X: PE1, Z. 143; PE2, Z. 126 f.; PE3, Z. 128; PE6, Z. 166). Diese Eingriffsmöglichkeit sollte jedoch in Abhängigkeit der Expertise und des Status des Anwenders durch einen Authentisierungsprozess reguliert werden, um mögliche Fehleingriffe zu verhindern.

7.5 Limitationen der Untersuchung

Insgesamt konnten bis auf eine Ausnahme die Hypothesen bestätigt und das Potential des Prototyps aufgezeigt werden (vgl. Kapitel 7.4.3). Es gibt allerdings Limitationen bei der Untersuchung. Im Folgenden sollen die vier Limitationen hinsichtlich *Aufgabenstellung*, *Software*, *Datenbasis* und *Probanden* diskutiert werden.

Bei der *Aufgabenstellung* ist der Umfang der zu überwachenden Variablen limitiert. Im Rahmen des Experimentes soll eine abhängige Variable überwacht werden, die maximal von sechs unabhängigen Variablen bestimmt werden kann. Das Modell ist wesentlich größer und würde somit das Potential für eine komplexere Aufgabenstellung bieten, jedoch müssen im Hinblick auf die KG die zu erledigenden Aufgaben annähernd praktikabel sein. Bereits bei der limitierten Variante konnten die Teilnehmer der KG die Aufgaben nicht vollständig lösen. Auch die Anzahl zu verwaltender Modelle könnte für weitere Evaluationsvorhaben gesteigert werden. Im Rahmen des Untersuchungsdesigns könnte das Merkmal der Komplexität in den Hypothesen mitberücksichtigt werden. Da der Schwierigkeitsgrad je Störung steigt, könnten die Faktoren Effektivität sowie Effizienz zusätzlich in Abhängigkeit der Aufgabe und der Komplexität analysiert werden. Bei der Untersuchung der Hypothesen könnten zusätzlich zu den deskriptiven Faktoren auch ein Strukturgleichungsmodell berücksichtigt werden, das in den zu prüfenden Wirkungen ggf. den Schwierigkeitsgrad von Aufgaben berücksichtigt.

Die genutzte *Software* des Alternativsystems zum Vergleich mit dem Prototyp ist eine weitere Limitation. Es ist das weit verbreitete System Excel genutzt worden, mit dem sowohl potentielle Struktur- als auch Parameteranpassungen (vgl. Kapitel 7.4.2.1) berechnet werden können. Jedoch ist der Aufbereitungsaufwand sehr hoch, was sich deutlich in der Effizienz widerspiegelt (vgl. Kapitel 7.4.2.2). Eine mögliche Erklärung liegt in der Tatsache begründet, dass Excel im Gegensatz zum Prototyp nicht auf die Analyse von Abweichungen und der Identifikation von Ursachen durch eine Regressionsbildung spezialisiert ist. Dies sollte bei der Interpretation der Ergebnisse hinsichtlich Effektivität und Effizienz berücksichtigt werden.

Die dritte Limitation betrifft die zugrundeliegende *Datenbasis*. Die Untersuchung basiert auf fiktiven Daten. Die Modelle stammen nicht aus einem realen Unternehmen und auch die Szenarien für die Störungen sind fiktiv. Die verwendeten Modelle finden jedoch in der Wissenschaft und Lehre Anwendung, weshalb durch die stetige Anwendung eine Kontrolle und ggf. Verbesserung erfolgt (vgl. Kapitel 6.2). Trotzdem sollte die fiktive Datenbasis bei der Interpretation der Ergebnisse berücksichtigt werden.

Die letzte Limitation stellt die Gruppe der *Probanden* dar, da es sich um Studierende handelt und somit nicht um echte Experten aus der Praxis. Allerdings wurden ausschließlich Masterstudierende ausgewählt, die unmittelbar vor dem Einstieg in das Berufsleben stehen und einem potentiellen Anwender ähnlicher sind als Studienanfänger. Darüber hinaus haben die Teilnehmer im Rahmen ihrer Studienlaufbahn Erfahrungen mit Modellen sammeln können. Die methodischen Grundlagen sind ihnen ebenfalls vermittelt worden. Bei der Auswahl der Studierenden ist demnach versucht worden, eine bestmögliche Grundlage für eine realitätsnahe Evaluation zu schaffen, jedoch handelt es sich bei den Probanden um eine mögliche Limitation.

8 Schlussbetrachtung

Die Forschungsergebnisse der vorliegenden Arbeit werden im Folgenden zusammenfassend vorgestellt (vgl. Kapitel 8.1). Abschließend wird ein Ausblick auf potentielle weiterführende Forschungsvorhaben gegeben (vgl. Kapitel 8.2).

8.1 Zusammenfassung der Forschungsergebnisse

Das Ziel der Arbeit ist, Modelle innerhalb eines Entscheidungsunterstützungssystem mit Hilfe der Selbst-Adaption kontinuierlich realitätsnah zu repräsentieren, damit ein Entscheider stetig bestmöglich unterstützt werden kann (vgl. Kapitel 1.1). Zur Erreichung dieses Ziels sind unterschiedliche Forschungsschritte geplant sowie durchgeführt worden (vgl. Kapitel 1.2). Die Erkenntnisse aus den jeweiligen Schritten werden im Folgenden zusammengefasst.

In Kapitel 2 wird das Nutzenpotential von Entscheidungsunterstützungssystemen aufgezeigt. Insbesondere bei der betriebswirtschaftlichen Planung können durch Simulationen verschiedene zukünftige Szenarien evaluiert werden, um adäquate Entscheidungsunterstützung zu liefern. Die Grundlage zur Simulation eines Szenarios bildet ein Modell, welches versucht, Wirkungszusammenhänge aus der Realität abzubilden. Ein Problem betriebswirtschaftlicher Planungsmodelle ist eine steigende Dynamik in der Unternehmensumwelt, die die Validität eines Modells negativ beeinflussen kann und somit eine Fehlerquelle für die adäquate Entscheidungsunterstützung darstellen kann.

Kapitel 3 stellt mit dem Prinzip der Selbst-Adaption einen potentiellen Lösungsansatz zur Sicherstellung der Modellvalidität in dynamischer Umwelt vor. Mit Hilfe einer Literaturrecherche werden vorhandene Forschungsarbeiten identifiziert, die die Selbst-Adaption zur Anpassung an eine dynamische Umwelt einsetzen. Die Ergebnisse werden Konzeptzentriert anhand der jeweils eingesetzten Verfahren dargestellt. Der Fokus der identifizierten Forschungsarbeiten liegt primär auf der Kalibrierung Algorithmen-spezifischer Parameter in Abhängigkeit der Umweltbedingungen. Keine der Forschungsarbeiten betrachtet als Anwendungsdomäne der Selbst-Adaption das Themengebiet der modellgetriebenen Entscheidungsunterstützungssysteme. Im Rahmen dieser Domäne ist es notwendig, neben parametrischen Anpassungen auch Strukturveränderungen an einem quantitativen Entscheidungsmodell durchzuführen. Diese Notwendigkeit sowie der problematische Einsatz

von modellgetriebenen EUS in einer dynamischen Unternehmensumwelt sind in einer Umfrage von Praktikern bestätigt worden, womit die Relevanz der Forschung aufgezeigt wurde.

Im folgenden Kapitel 4 wurden die in der Wissenschaft etablierten Ansätze zur Entwicklung eines selbst-adaptiven Systems untersucht und darauf aufbauend ein Lösungskonzept für ein selbst-adaptives modellgetriebenes EUS entwickelt (vgl. Kapitel 1.2). Die funktionalen Anforderungen orientieren sich an den sechs *Self-** Eigenschaften und die prozessualen Anforderungen an dem *MAPE-K* Zyklus. Auf Basis der entwickelten Anforderungen ist die *Model-/System-Controller Architektur* konzipiert worden, die aus vier zentralen Komponenten besteht: Dem Modell als Abbild der Realität, dem System-Controller zur Anwendung des Modells sowie zur Unterstützung des Anwenders, dem Observer zur Überwachung der Modellvalidität und dem Modell-Controller zur Erstellung von Modellanpassungsvorschlägen im Fall einer Abweichung. Der Observer überwacht das Modell und prüft kontinuierlich, ob eine *Modellaktualisierung* notwendig ist (vgl. Kapitel 1.1). Der Model-Controller sorgt durch ein dynamisches Vorgehen beim Erstellen von Experimentalmodellen für eine Reduktion der *strukturellen Komplexität* und durch eine vielschichtige Evaluation auf Modell- und Variablenebene für einen Abbau der *funktionalen Komplexität* (vgl. Kapitel 1.1).

Kapitel 5 betrachtet im Rahmen der Eval 2 Aktivität der ex ante Evaluation (vgl. Kapitel 1.2) die Spezifikationen der entwickelten *MoSyCo* Architektur und prüft diese auf Korrektheit und Vollständigkeit. Zur Sicherstellung der Relevanz der Forschung erfolgt dieser Schritt im Rahmen einer Umfrage von potentiellen Anwendern der *MoSyCo* Architektur aus der Praxis. Im Ergebnis wird die Korrektheit der Spezifikation bestätigt, indem der Beitrag einzelner Funktionen zur Lösung der initial festgestellten Probleme geprüft wurde. Abschließend haben die potentiellen Anwender eine Priorisierung der Funktionalitäten vorgenommen, die im Rahmen der prototypischen Implementierung berücksichtigt wurde.

Zur Instanziierung der konzipierten Architektur wurde eine prototypische Implementierung vorgenommen und diese in Kapitel 6 dokumentiert. Unter Berücksichtigung aktueller Entwicklungsumgebungen wie Python und R sind zur Zeitreihenanalyse bei der Abweichungs- und Ursachenanalyse innovative Ansätze (bspw. *Prophet*) genutzt worden. Der Prototyp evaluiert im Falle einer Abweichung verschiedene Modellanpassungsvorschläge und setzt den Vorschlag mit der höchsten Güte automatisiert in einem generischen Modellabbild um. Durch dieses automatisierte Vorgehen wird der *Aufwand* der manuellen Modellwartung

(vgl. Kapitel 1.1) stark reduziert. Der Anwender hat die Möglichkeit, die Ergebnisse der Abweichungsanalyse und Ursachenidentifikation über graphische Benutzerschnittstellen zu verfolgen.

In Kapitel 7 wird die Gebrauchstauglichkeit des Prototyps und das Vertrauen in automatisierte Systeme überprüft. Der Prototyp weist eine deutlich höhere Effektivität und Effizienz bei der Abweichungs- und Ursachenanalyse im Vergleich zum Alternativsystem Excel auf. Auch das Vertrauen in die Richtigkeit der Ergebnisse des automatisierten Anpassungsprozesses des *Model-Controller* konnte bestätigt werden. Durch das durchgeführte Laborexperiment ist aufgezeigt worden, dass trotz des prototypischen Stadiums der Implementierung der *MoSyCo* Architektur die Gebrauchstauglichkeit für verschiedene Nutzungsszenarien gegeben ist.

Das Gesamtziel der Arbeit konnte durch die erfolgreiche Durchführung der einzelnen Forschungsschritte erreicht werden. Neben der Vollständigkeit und Korrektheit der konzipierten *MoSyCo* Architektur konnte die Gebrauchstauglichkeit der prototypischen Instanziierung belegt werden. Die Ergebnisse der vorliegenden Arbeit können als Basis für weitere Forschungsvorhaben im Rahmen selbst-adaptiver modellgetriebener Entscheidungsunterstützungssysteme genutzt werden.

8.2 Ausblick auf weiterführende Forschungsarbeiten

Potentielle weiterführende Forschungsarbeiten können im Hinblick auf die unterschiedlichen Forschungsschritte kategorisiert werden. Im Bereich des *Konzeptes*, des *Prototyps* sowie der *Evaluation des Prototyps* sind Ansatzpunkte vorhanden.

Der Forschungsbedarf im Hinblick auf das *Konzept* kann in Abhängigkeit der Komponenten betrachtet werden. Der *Observer* sieht bereits vor, dass neben der reaktiven Abweichungsanalyse von Plan- und Istwerten auch vorausschauend Plan- und Prognosewerte analysiert werden. Die Schlussfolgerungen bei potentiellen Abweichungen zwischen Plan- und Prognosewerten können erweitert werden. So wäre es sinnvoll, wenn bei einer in der Zukunft drohenden Abweichung der Grenzwert für die Abweichungsanalyse zwischen Plan- und Istwert reduziert wird. In diesem Fall würde die reaktive Abweichungsanalyse von den Ergebnissen der vorausschauenden Analyse profitieren, indem eine Modellanpassung aufgrund des geringeren Toleranzbereiches für Abweichungen früher initiiert wird. Durch diese Bewertung der Prognosewerte könnte das Risiko, eine Entscheidung auf Basis

eines invaliden Modells zu tätigen, weiter reduziert werden. Auch der *Model-Controller* kann hinsichtlich des Erstellens von Anpassungsvorschlägen erweitert werden. Neben dem bereits vorhandenen Ansatz, Anpassungsvorschläge datenbasiert zu berechnen, könnte zusätzlich ein fallbasierter Ansatz verfolgt werden. Demnach müssten Modellanpassungsvorschläge nicht jedes Mal neu berechnet werden, sondern auf Grundlage vergangener Anpassungen könnten die bestehenden Erkenntnisse der Ursachenanalyse auf den neu zu analysierenden Fall übertragen werden. Das fallbasierte Schließen würde hierfür einen geeigneten methodischen Ansatz darstellen (vgl. Aamodt und Plaza 1994). Ein zusätzlicher Fortschritt könnte erzielt werden, wenn zur Ursachenanalyse eine generische Wissensbasis genutzt wird, in der allgemeingültige Wirkungsbeziehungen innerhalb einer Branche vorgehalten werden (vgl. Benjamins 2017). So stellt der Ansatz eine Erweiterung der *MoSyCo* Architektur auf Basis der eigenen unternehmensspezifischen Daten dar und der wissensbasierte Ansatz kann das Konzept auf Basis unternehmensübergreifenden Wissens über Wirkungsbeziehungen erweitern. Die wissensbasierte Erweiterungsmöglichkeit würde die Einführung der *MoSyCo* Architektur in einem neuen Unternehmen erleichtern, da branchenspezifische Wirkungszusammenhänge aus Entscheidungsmodellen in der Wissensbasis bereits vorhanden wären. Sowohl der fallbasierte als auch der wissensbasierte Ansatz können im Vergleich zur datenbasierten Alternative jedoch einen erhöhten Pflegeaufwand nach sich ziehen, da die Fallmerkmale der Fallbasis oder die branchenspezifischen Wirkungszusammenhänge in der Wissensbasis über die Zeit repräsentativ bleiben müssen.

Die im *Prototyp* jeweils enthaltene Methodenbank zur Abweichungs- und Ursachenanalyse könnte erweitert werden. Der *Observer* könnte neben der absoluten und relativen Abweichungsanalyse zusätzliche Fehlermaße berücksichtigen. *Scikit-learn* ist bspw. eine Bibliothek für maschinelles Lernen (vgl. Pedregosa et al. 2011), die es ermöglicht, aufeinanderfolgende Plan- und Istwerte anhand eines gemittelten quadratischen Fehlers auf signifikante Abweichungen zu analysieren. Im Rahmen der Abweichungsanalyse zwischen Plan- und Prognosewerten bieten sich ebenfalls alternative Methoden wie das ARIMA-Modell an, welches u. a. in der Bibliothek *Statsmodels* verfügbar ist (vgl. McKinney et al. 2011). Darüber hinaus sind in dieser Bibliothek Methoden vorhanden, mit denen sich Trends oder Saisonalitäten in Zeitreihen analysieren lassen. Die Instanziierung des *Model-Controller* könnte für eine verbesserte Ursachenanalyse um eine Methode erweitert werden, die eine zeitabhängige Schätzung von Koeffizienten vorsieht, wie es bspw. durch den Einsatz des Kalman Filters geschieht (vgl. Petris et al. 2009, S. 31). Darüber hinaus könnten auch An-

sätze des maschinellen Lernens wie bspw. künstliche neuronale Netze auf ihre Tauglichkeit untersucht werden.

Weiterer Forschungsbedarf im Hinblick auf die *Evaluation des Prototyps* richtet sich primär auf den praktischen Einsatz im Unternehmen. Für einen vollständigen Forschungsprozess nach Sonnenberg und vom Brocke ist es ohnehin notwendig, den Prototyp im Unternehmensumfeld einzusetzen und im Rahmen der vierten Evaluation bspw. durch ein Feldexperiment zu untersuchen (vgl. Sonnenberg und vom Brocke 2012, S. 395 f.). Die Anwendung des Prototyps im Unternehmen sowie die vierte Evaluationsaktivität sind jedoch nicht Bestandteil der vorliegenden Forschungsarbeit (vgl. Kapitel 1.2) und stellen deshalb weiteren Forschungsbedarf dar. Insbesondere der Umgang mit Daten, die in Echtzeit gemessen werden, könnte aus Sicht der Leistungsfähigkeit des Prototyps neue Erkenntnisse über Engpässe erbringen. Neben diesem technischen Aspekt könnte aus verhaltenensorientierter Sicht das Vertrauen in automatisiert handelnde Systeme weiter untersucht werden.

Anhang

Anhang I Fragebogen



Teil NV: Nutzung und Verbreitung von Entscheidungsunterstützungssystemen

Ich möchte Sie nun bitten, einige Fragen zur Nutzung von Entscheidungsunterstützungssystemen in Ihrem Unternehmen zu beantworten.

Entscheidungsunterstützungssysteme (EUS, im Englischen Decision Support Systems) haben das Ziel, einen Entscheider auf operativer als auch strategischer Ebene mit der Sammlung, Aufbereitung und Analyse von relevanten Daten bei der Entscheidungsfindung in unterschiedlichen Situationen zu unterstützen. Diese Systeme müssen nicht eigenständig sein, sondern die Funktionen eines EUS können in andere Systeme (bspw. Enterprise-Resource-Planning-Systeme (ERP), Customer-Relationship-Management (CRM) etc.) integriert sein. Beispielhafte Funktionen von Entscheidungsunterstützungssystemen sind u.a.:

Optimierung von Variablen innerhalb vorgegebener Grenzwerte (bspw. Minimierung von Kosten, Maximierung des Gewinns)
 Aggregation von Daten in unterschiedliche Kennzahlen Simulation bestimmter Szenarien als Prognose für die Zukunft
 Zielwertbestimmung unter verschiedenen Restriktionen Verwaltung und Aufarbeitung einer großen Datenbasis Förderung von
 Zusammenarbeit in Arbeitsgruppen durch Netzwerk- und Kommunikationstechnologien

NV1. Sind Ihnen Entscheidungsunterstützungssysteme bekannt?

Ich habe davon noch nie gehört.

Ich habe davon gehört, aber kann es nicht zuordnen.

Ich kenne Entscheidungsunterstützungssysteme.

NV2. Werden in Ihrem Unternehmen Entscheidungsunterstützungssysteme eingesetzt, die bspw. die o.g. Funktionen bereitstellen?

Ja

Nein

NV3. Haben Sie in der Vergangenheit die oben genannten (o.ä.) Funktionen eines EUS genutzt, um damit Entscheidungen zu treffen?

Ja

Nein

Teil EEUS: Einsatz von Entscheidungsunterstützungssystemen

Bitte beantworten Sie nun einige Fragen zu Ihrem eingesetzten Entscheidungsunterstützungssystem (EUS).

EEUS1. In welchen Bereichen werden Entscheidungsunterstützungssysteme in Ihrem Unternehmen eingesetzt?

Beschaffung

Controlling

Fertigung

Finanzwesen

Logistik

Marketing



EEUS4. Würden Sie die von Ihnen (bzw. Ihren Mitarbeitern) genutzten Entscheidungsunterstützungssysteme weiterempfehlen?

Ja

Nein

Sonstiges

Sonstiges

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

EEUS5. Welche Geschäftsprozesse erfolgen durch die Assistenz von Entscheidungsunterstützungssystemen in Ihrem Unternehmen?

Auftragsplanung

Lagerplanung

Produktionsplanung

Liquiditätsplanung

Kredit- und Darlehensvergabe

Transport- und Fahrtenplanung

Preisplanung

Ressourcenplanung

Arbeitseinsatzplanung

Sonstiges

Sonstiges

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

EEUS6. Werden die Entscheidungsunterstützungssysteme in Ihrem Unternehmen regelmäßig aktualisiert, gewartet und Instand gehalten?

Ja

Nein

EEUS7. Wer ist für die Wartung und Instandhaltung der Systeme in Ihrem Unternehmen verantwortlich?

Interne IT-Abteilung

Fachabteilung

Externer IT-Dienstleister



Sonstiges

Sonstiges

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

EEUS8. Woher stammen die Daten, mit welchen das Entscheidungsunterstützungssystem gespeist wird?

- Enterprise-Resource-Planning (ERP) System
- Customer-Relationship-Management (CRM) System
- Warenwirtschaftssystem (WWS)
- Produktionsplanungs- und -steuerungssysteme (PPS-Systeme)
- Supply-Chain-Management (SCM) System
- Transportation-Management-System (TMS)
- Campus-Management-System (CMS)
- Data Warehouse (DWH)
- Excel
- Sonstiges

Sonstiges

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

EEUS9. Beurteilen Sie die nachfolgenden Aussagen für Ihr eingesetztes Entscheidungsunterstützungssystem.

	Trifft nicht zu	Trifft über nicht zu	Teils-teils	Trifft über zu	Trifft zu
Der Einsatz des EUS steigert die Effizienz des Unternehmens durch eine verbesserte Entscheidungsgrundlage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Der Output des EUS ist korrekt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich kann mich auf den Output des EUS verlassen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Nutzung des EUS ist risikoreich, da möglicherweise inadäquate Entscheidungen auf Grundlage falscher Daten getroffen werden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

EEUS10. Welche Funktionen werden durch das Entscheidungsunterstützungssystem in Ihrem Unternehmen abgedeckt?

- Selektion/Auswahl der relevanten Daten aus einer großen Datenbasis (bspw. aus anderen Systemen oder einer Datenbank)
- Aggregation und Verdichtung von Daten in bestimmte Kennzahlen
- Bestimmung von statistischen Zusammenhängen zwischen Variablen oder Kennzahlen (bspw. die Parameter einer Wahrscheinlichkeitsverteilung)



Simulationen von Entwicklungen in der Zukunft zur Analyse von Entscheidungsalternativen	<input type="checkbox"/>
Auflösen von Simultanitätsgleichungen unter Berücksichtigung von Abhängigkeiten (bspw. die Umlage von Gemeinkosten in der Kostenstellenrechnung)	<input type="checkbox"/>
Optimierung von Variablen oder Kennzahlen unter Berücksichtigung von Nebenbedingungen (bspw. Minimierung von Kosten oder Maximierung des Gewinns)	<input type="checkbox"/>
<p>Teil KEUS: Kein Einsatz von Entscheidungsunterstützungssystemen</p> <p>Bitte beantworten Sie die folgenden Fragen, da Sie bzw. Ihr Unternehmen kein Entscheidungsunterstützungssystem nutzen bzw. nutzt.</p> <p>KEUS1.Warum verzichtet Ihr Unternehmen auf den Einsatz von Entscheidungsunterstützungssystemen?</p> <div style="display: flex; justify-content: space-between; align-items: flex-start; margin-top: 10px;"> <div style="width: 80%;"> <p style="margin-bottom: 10px;">Zu hohe Kosten für Einführung und Wartung</p> <p style="margin-bottom: 10px;">Zu hoher Arbeitsaufwand in der Einführung und Wartung</p> <p style="margin-bottom: 10px;">Keine Akzeptanz durch Mitarbeiter</p> <p style="margin-bottom: 10px;">Zu hoher Aufwand in Training und Schulung der Mitarbeiter</p> <p style="margin-bottom: 10px;">Übernahme von EUS dauert zu lange</p> <p style="margin-bottom: 10px;">Kein Nutzen</p> <p style="margin-bottom: 10px;">Sonstiges</p> <p style="margin-bottom: 10px;">Sonstiges</p> <div style="border: 1px solid black; height: 1.2em; width: 500px;"></div> </div> <div style="width: 10%; text-align: center;"> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> </div> </div>	
<p>KEUS2.Bestehen Pläne für die Einführung von Entscheidungsunterstützungssystemen in Ihrem Unternehmen?</p> <div style="display: flex; justify-content: flex-end; margin-top: 10px;"> <div style="width: 70%; text-align: right;"> <p style="margin-bottom: 10px;">Ja</p> <p style="margin-bottom: 10px;">Nein</p> <p style="margin-bottom: 10px;">Nicht bekannt</p> </div> <div style="width: 20%; text-align: center;"> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> </div> </div>	



	Trifft nicht zu	Trifft aber nicht zu	Teils-teils	Trifft über zu	Trifft zu	Keine Antwort
Outputs aus diesem System geben mir eine fundierte Entscheidungsgrundlage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich würde ein solches Entscheidungsunterstützungssystem in Zukunft nutzen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Teil UI: Unternehmensdaten

Abschließend möchte ich Sie bitten, Ihr Unternehmen hinsichtlich Größe und Branche einzuordnen.

Sollten Sie Interesse an den Ergebnissen dieser Umfrage haben, so haben Sie die Möglichkeit, die unten stehenden Felder zur Kontaktaufnahme auszufüllen.

UI1. Wie viele Mitarbeiter arbeiten in Ihrem Unternehmen?

- Weniger als 10
- 10-49
- 50-249
- 250-999
- 1000-5000
- Mehr als 5000

UI2. In welcher Branche ist Ihr Unternehmen tätig?

- Land- und Forstwirtschaft, Fischerei
- Bergbau und Gewinnung von Steinen und Erden
- Verarbeitendes Gewerbe
- Energieversorgung
- Wasserversorgung; Abwasser- und Abfallentsorgung und Beseitigung von Umweltverschmutzungen
- Baugewerbe
- Handel; Instandhaltung und Reparatur von Fahrzeugen
- Verkehr und Lagerei
- Gastgewerbe
- Information und Kommunikation
- Erbringung von Finanz- und Versicherungsdienstleistungen
- Grundstücks- und Wohnungswesen
- Öffentliche Verwaltung, Verteidigung; Sozialversicherung
- Erziehung und Unterricht



Gesundheits- und Sozialwesen

Kunst, Unterhaltung und Erholung

Sonstiges

Sonstiges

Grid of 20 empty boxes for text input.

UI 3. Wenn Sie Interesse an den Ergebnissen dieser Umfrage haben, füllen Sie bitte die folgenden Felder aus.

Name des Unternehmens

Ansprechpartner

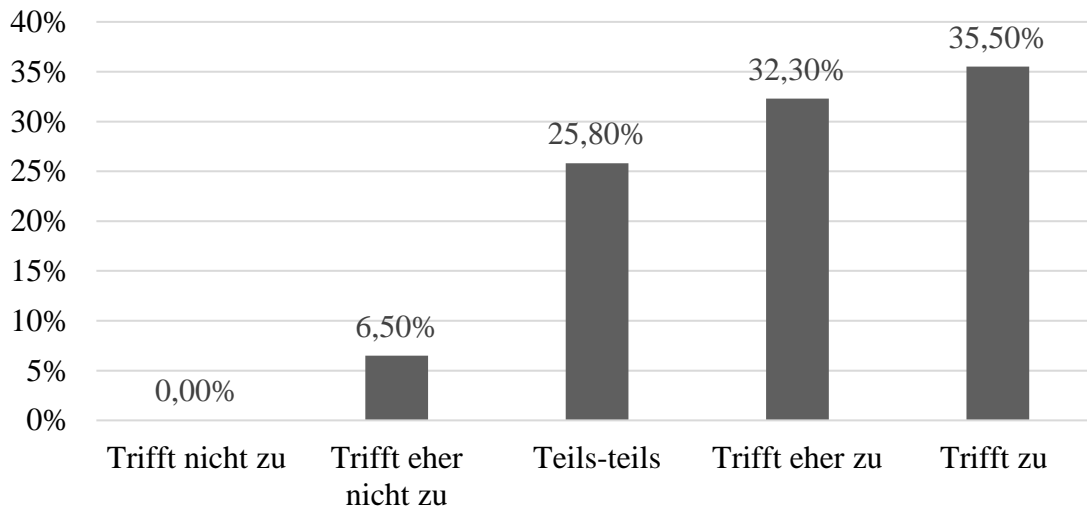
E-Mail-Adresse

Telefonnummer

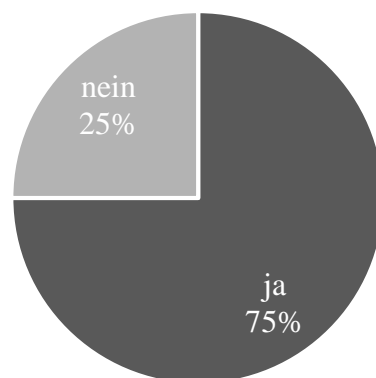
Vielen Dank für Ihre Teilnahme!

Anhang II Effizienz und Weiterempfehlungsrate

Aussage: „Der Einsatz des EUS steigert die Effizienz des Unternehmens durch eine verbesserte Entscheidungsgrundlage.“

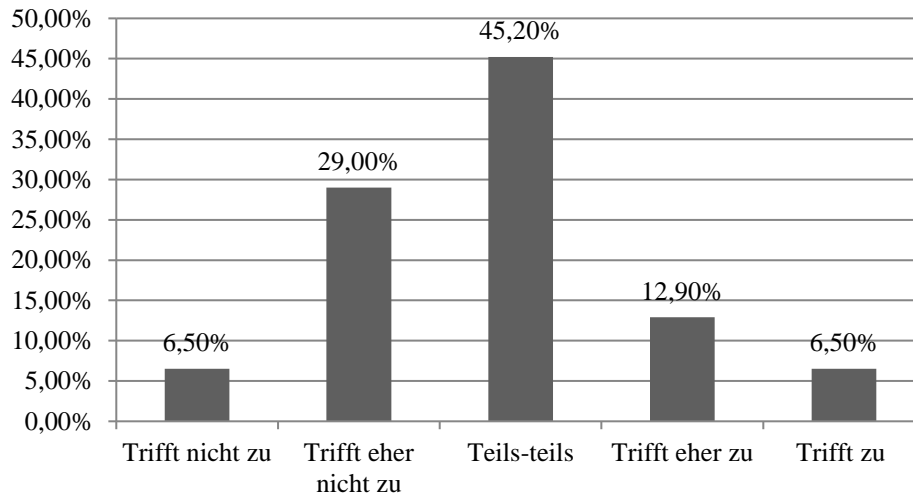


Frage: „Würden Sie die von Ihnen genutzten EUS weiterempfehlen?“



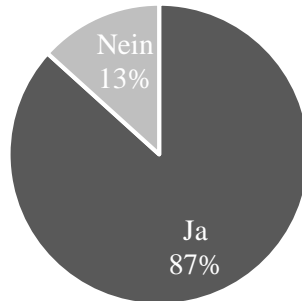
Anhang III Risiko eines falschen Ergebnisses

Aussage: „Die Nutzung des EUS ist risikoreich da möglicherweise inadäquate Entscheidungen auf Grundlage falscher Daten getroffen werden.“

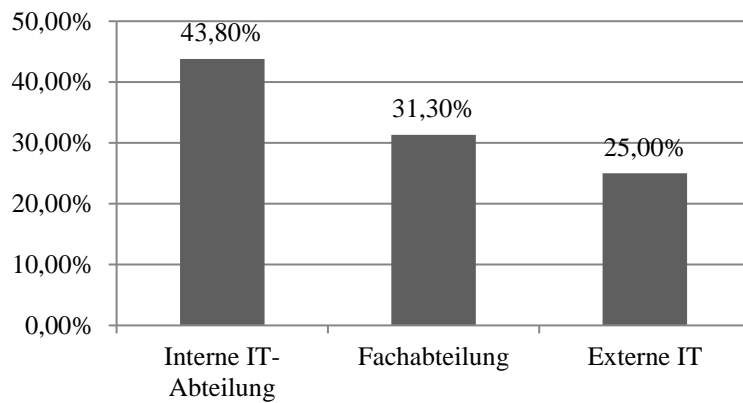
Risiko des Outputs

Anhang IV Wartung des EUS

Frage: „Wird das EUS regelmäßig gewartet?“

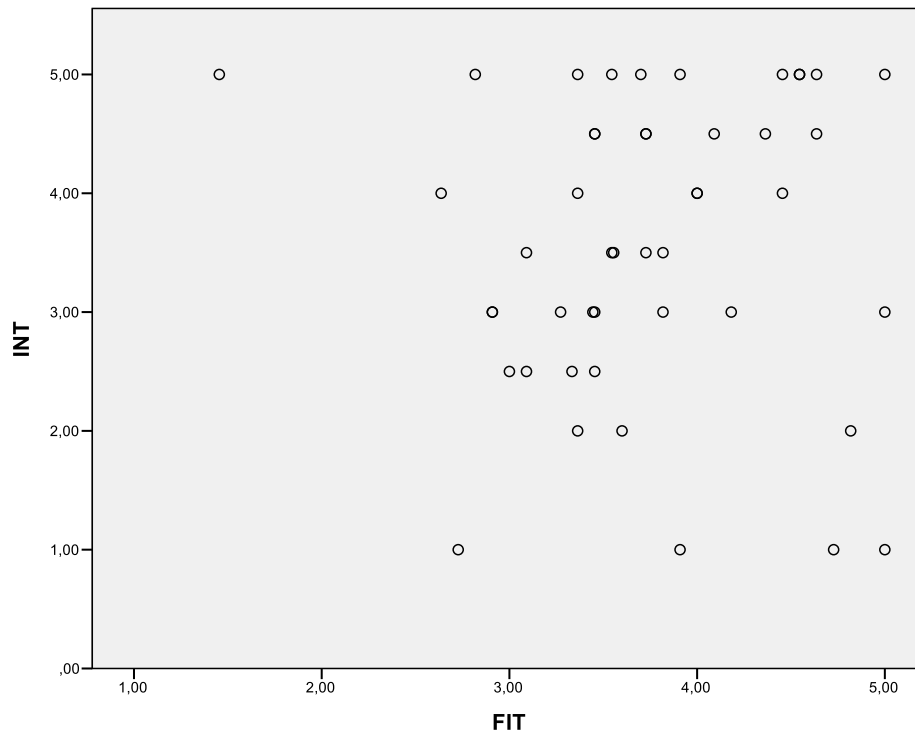


Frage: „Wer ist für die Wartung verantwortlich?“

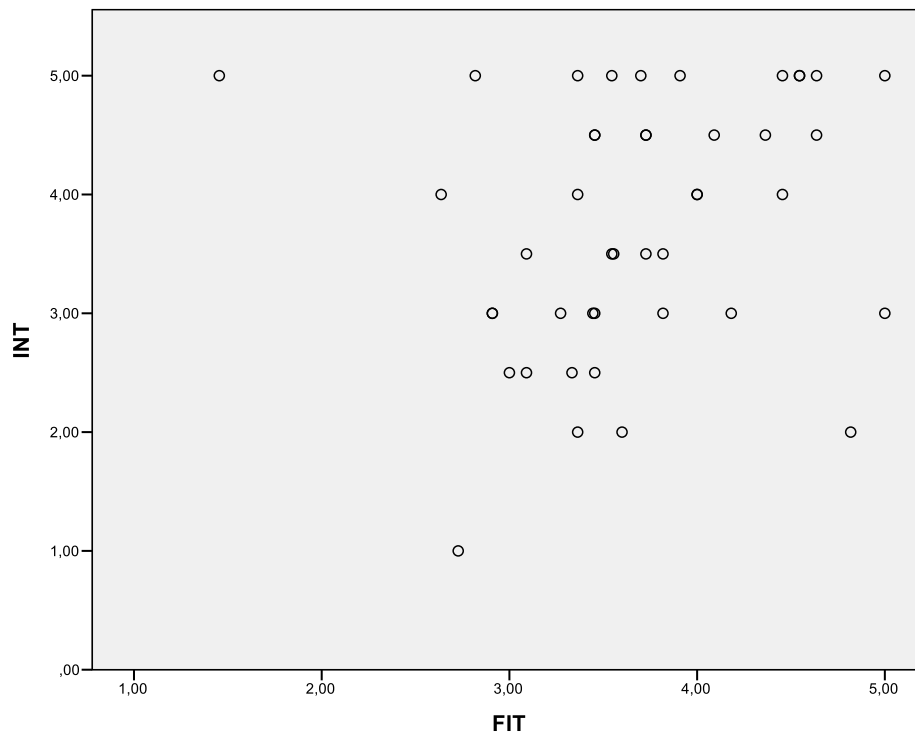
Wartungsverantwortlichkeit

Anhang V Überprüfung der Annahmen für Hypothese H5-1

- Scatterplot Diagramm der Variablen „FIT“ und „INT“ mit Ausreißer



- Scatterplot Diagramm der Variablen „FIT“ und „INT“ ohne 3 Ausreißer



- Shapiro-Wilk-Test auf Normalverteilung

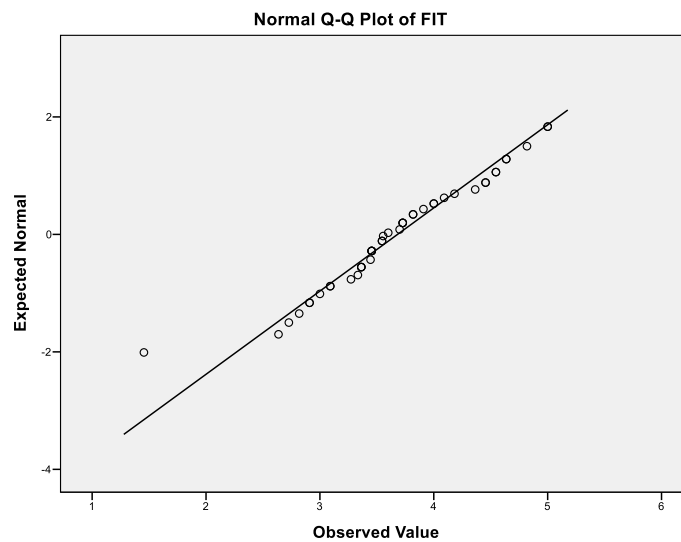
Tests of Normality

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
FIT	,088	44	,200 *	,963	44	,168
INT	,167	44	,003	,911	44	,002

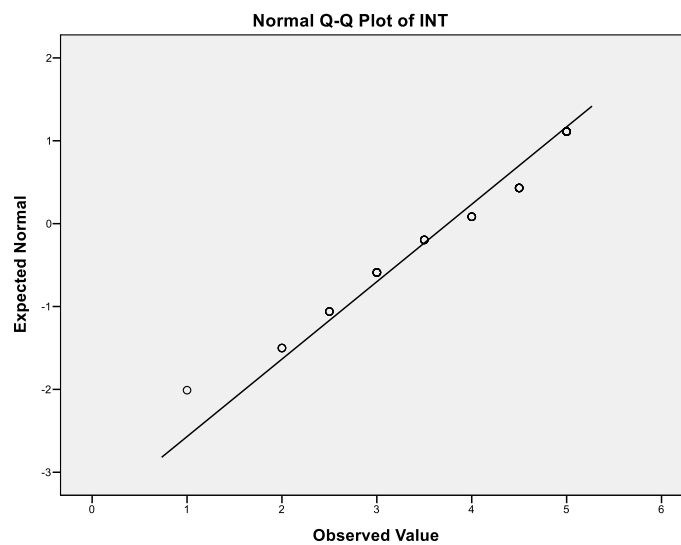
*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

- Der Fit ist normalverteilt, da der p-Wert $> 0,05$ ist. Q-Q Plot:

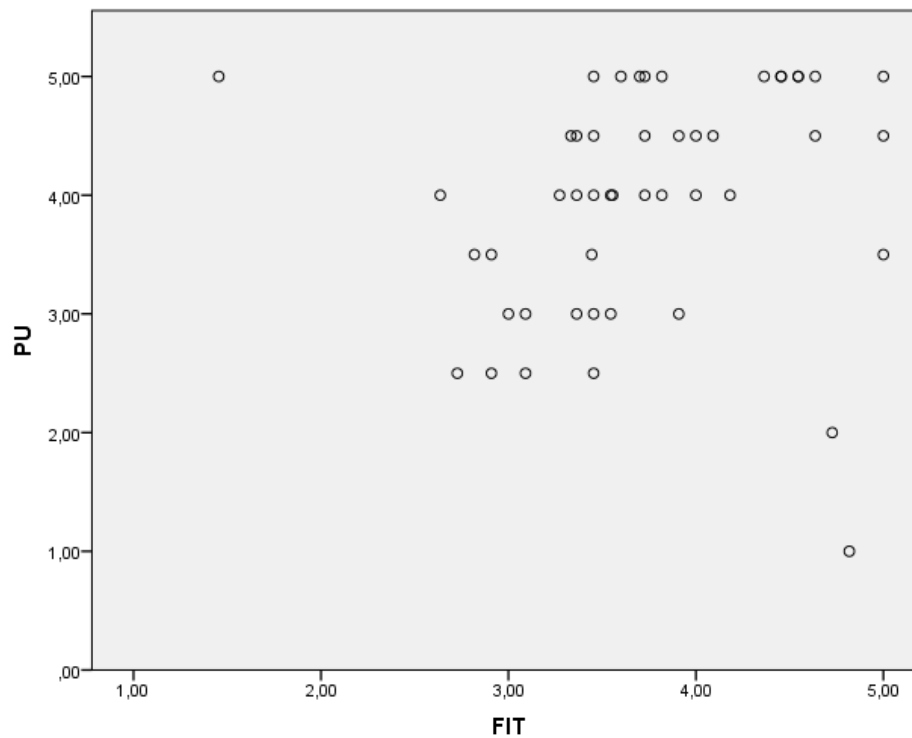


- Die Intention ist nicht normalverteilt, da der p-Wert $< 0,05$ ist. Q-Q Plot:



Anhang VI Überprüfung der Annahmen für Hypothese H5-2

- Scatterplot Diagramm der Variablen „FIT“ und „PU“



- Shapiro-Wilk-Test

Tests of Normality

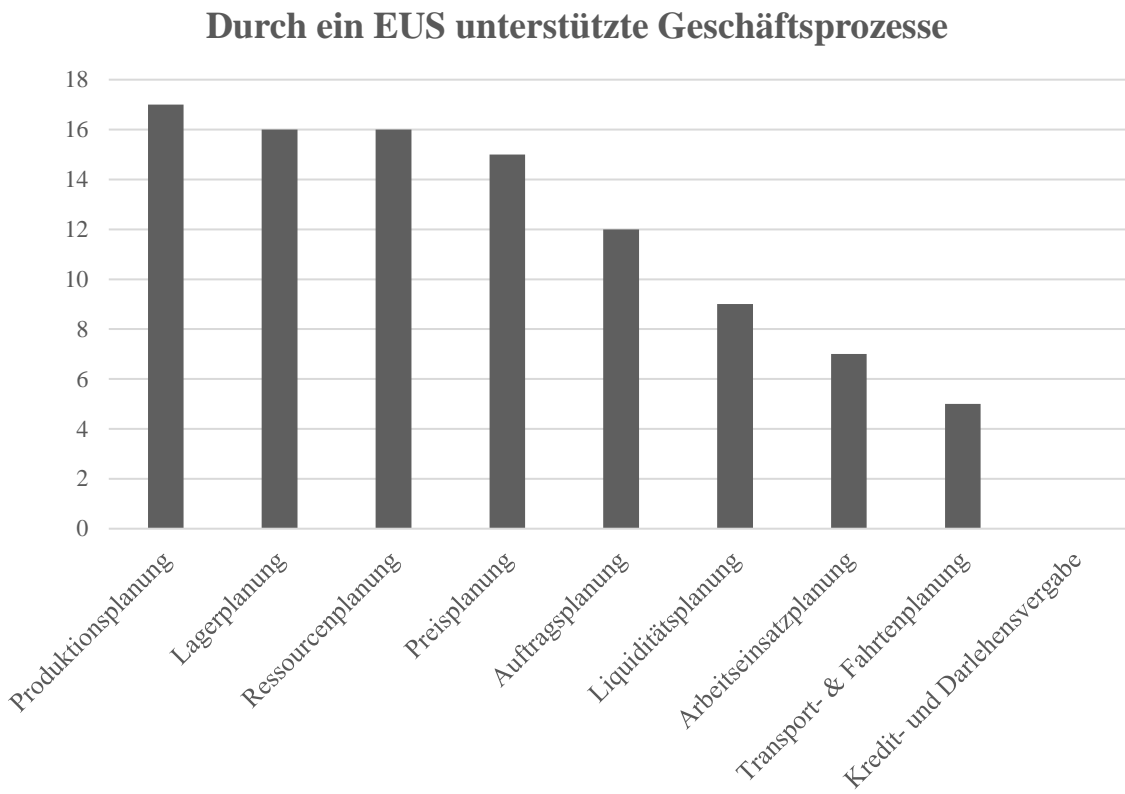
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
FIT	,085	48	,200*	,960	48	,101
PU	,180	48	,001	,888	48	,000

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

- Fit ist normalverteilt, da p-Wert > 0,05
- Nutzen ist nicht normalverteilt, da p-Wert < 0,05

Anhang VII Unterstützte Geschäftsprozesse



Anhang VIII Formeln des Entscheidungsmodells

Die Variablen mit dem Präfix Real verweisen auf das Realitätsmodell aus Anhang IX.

- (01) AktuellerMarktpreis = A FUNCTION OF()
 AktuellerMarktpreis=
 Price
 Units: **undefined**
- (02) ANachfrage = A FUNCTION OF(DAX,Export,NachfrageActual,Time)
 ANachfrage=
 IF THEN ELSE(Time<=AA Entscheidungszeitpunkt, NachfrageActual, (
 +DAX*0.1
 +Export*10
))
 Units: **undefined**
- (03) Anlieferung=
 DELAY FIXED(Produktion, PRODUKTIONSZEIT,Produktion)
 Units: Stück/Day
- (04) AuftragsBestand= INTEG (
 +ProduktionsAuftraege-Produktion,
 1)
 Units: Stück
- (05) Bearbeitungszeit=
 7
 Units: **undefined**
- (06) BestellungNachDefizit=
 DELAY FIXED(LAGERDEFIZITBESTELLFAKTOR*LagerDefizit, Bear-
 beitungszeit, LagerDefizit
 * LAGERDEFIZITBESTELLFAKTOR)
 Units: Stück/Day
 Das Reporting zum LagerDefizit erfolgt alle x-Tage (= Bearbeitungszeit) und erst dann kann bestellt werden.
- (07) BestellungNachTagesverkauf=
 DELAY FIXED(VERKAUFSBESTELLFAKTOR*TagesVerkauf, Bearbei-
 tungszeit, TagesVerkauf
 *VERKAUFSBESTELLFAKTOR)
 Units: Stück/Day
 Das Reporting zum TagesVerkauf erfolgt alle x-Tage (= Bearbeitungszeit) und erst dann kann bestellt werden.
- (08) BIP = A FUNCTION OF()
 BIP=
 RealBIP
 Units: **undefined**

- (09) DAX = A FUNCTION OF()
 DAX=
 RealDAX
 Units: **undefined**
- (10) Export = A FUNCTION OF()
 Export=
 RealExport
 Units: **undefined**
- (11) FAKTORINVEST=
 2
 Units: Dimensionless
- (12) FINAL TIME = 3650
 Units: Day
 The final time for the simulation.
- (13) Flussvariable=A FUNCTION OF(Flussvariable)
 Flussvariable=
 Units: **undefined**
- (14) GesamtBestellung=
 IF THEN ELSE(BestellungNachDefizit+BestellungNachTagesverkauf<0, 0,
 BestellungNachDefizit
 +BestellungNachTagesverkauf)
 Units: Stück/Day
- (15) Gesamtdeckungsbeitrag=
 Gesamtumsatz-Gesamtkosten
 Units: **undefined**
- (16) Gesamtkosten= INTEG (
 Kosten,
 0)
 Units: **undefined**
- (17) Gesamtumsatz= INTEG (
 Umsatz,
 0)
 Units: **undefined**
- (18) Import = A FUNCTION OF()
 Import=
 RealImport
 Units: **undefined**
- (19) INITIAL TIME = 0
 Units: Day
 The initial time for the simulation.

- (20) Investition=
 $\text{AuftragsBestand}/100 \cdot \text{FAKTORINVEST}$
 Units: Dimensionless
 Pro 100 Aufträge werden die Investitionen kalkuliert
- (21) Kosten=
 $(\text{Investition}) + (\text{LagerBestand} \cdot 0.5) + (\text{Produktion} \cdot 75)$
 Units: **undefined**
 Investition + 50 Cent je Ware im Lager + 71€ je produzierter
 Einheit.
- (22) LagerBestand= INTEG (
 (+Anlieferung
 -TagesVerkauf),
 LAGERSOLL)
 Units: Stück
- (23) LagerDefizit=
 $\text{MAX}(0, \text{LAGERSOLL} - \text{LagerBestand})$
 Units: Stück
- (24) LAGERDEFIZITBESTELLFAKTOR=
 0.5
 Units: (Stück/Day)/Stück
 Bestellung proportional zum Lagerdefizit (0 bis 2)
- (25) LAGERSOLL=
 2000
 Units: Stück
- (26) NachfrageActual = A FUNCTION OF(Time)
 NachfrageActual=
 IF THEN ELSE(Time<=AA Entscheidungszeitpunkt, RealNachfrage, :NA:
)
 Units: **undefined**
- (27) PAF = WITH LOOKUP (
 $\text{VERKAUFSPREIS}/(\text{AktuellerMarktpreis} + \text{VERKAUFSPREIS}),$
 $([(-1,0)-(1,1)],(0,1),(0,1),(0.1,0.9),(0.2,0.8),(0.3,0.7),(0.4,0.6),(0.5,$
 $,0.5),(0.6,0.4),(0.7,0.3),(0.8,0.2),(0.9,0.1),(1,0))$
 Units: **undefined**
 Abweichung des eigenen Preises vom Marktpreis bestimmt den
 Prozentualen Anteil an der Gesamtnachfrage
- (28) Produktion=
 $\text{AuftragsBestand}/\text{PRODUKTIONSZEIT}$
 Units: Stück/Day
- (29) ProduktionsAuftraege=
 $\text{DELAY1}(\text{GesamtBestellung}, \text{Bearbeitungszeit})$
 Units: Stück/Day

- (30) PRODUKTIONSZEIT=
MIN(14, 10/Investition)
Units: Day
- (31) SAVEPER =
TIME STEP
Units: Day [0,?]
The frequency with which output is stored.
- (32) SubstitutionsmarktWachstum = A FUNCTION OF()
SubstitutionsmarktWachstum=
RealSubstitutionsmarktWachstum
Units: **undefined**
- (33) TagesVerkauf=
PAF * IF THEN ELSE((LagerBestand<(ANachfrage)), LagerBestand,
ANachfrage
)
Units: Stück/Day
- (34) TIME STEP = 1
Units: Day [0,?]
The time step for the simulation.
- (35) Umsatz=
TagesVerkauf*VERKAUFSPREIS
Units: **undefined**
- (36) VERKAUFSBESTELLFAKTOR=
0.8
Units: Dmnl [0,2]
Bestellung proportional zum Verkauf (0 bis 2)
- (37) VERKAUFSPREIS=
300
Units: **undefined**
- (38) Wetter = A FUNCTION OF()
Wetter=
RealWetter
Units: **undefined**

Anhang IX Formeln des Realitätsmodells

- (01) B Impact=
0
Units: **undefined**
- (02) Change in Traders' Expected Price=
(Indicated Price - Traders' Expected Price)/Time to Adjust Traders' Expected Price
Units: \$/Unit/Year
Traders' beliefs about the underlying equilibrium price adjust in response to the gap between the indicated price and the current belief. Expected underlying price adjusts via first-order adaptive expectations to the actual price, constrained to be greater than a minimum level.
- (03) D Imp A=
0.1
Units: **undefined**
- (04) D Imp B=
0.15
Units: **undefined**
- (05) D Imp End=
95
Units: **undefined**
- (06) D Imp Start=
80
Units: **undefined**
- (07) D Impact=
D Imp A+RAMP(((D Imp B-D Imp A) / (D Imp End-D Imp Start)), D Imp Start,
D Imp End)
Units: **undefined**
- (08) "Demand/Supply Balance" = A FUNCTION OF(Supply)
"Demand/Supply Balance"=
RealNachfrage/Supply
Units: Dimensionless
The balance between demand and supply, expressed as a dimensionless ratio.
- (09) E Impact=
1+RAMP(((1.5 - 1) / (900-800)), 800, 900)+RAMP(((2 - 1.5) / (2201-2200)), 2200, 2201)
Units: **undefined**
RAMP((((D Imp B) - (D Imp A)) / ((D Imp End) - (D Imp Start))),

D Imp Start, D Imp End)

- (10) Effect of Costs on Price=
 $1 + \text{Sensitivity of Price to Costs} \cdot ((\text{Expected Production Costs} / \text{Traders' Expected Price}) - 1)$
 Units: Dimensionless
 Price responds to the gap between traders' beliefs about the underlying equilibrium price and their beliefs about the costs of production. When expected costs rise above the expected price, prices tend to rise, and vice versa.
- (11) "Effect of Demand/Supply Balance on Price"=
 $\text{"Demand/Supply Balance"}^{\text{"Sensitivity of Price to Demand/Supply Balance"}}$
 Units: Dimensionless
 The effect of the demand/supply balance on price is a power function of the demand/supply balance. The higher the sensitivity of price to the demand/supply balance, the greater the change in price induced by any imbalance.
- (12) Expected Production Costs=
 $\text{Initial Price} \cdot (1 + (\text{STEP}(0,1) - \text{STEP}(0,400)))$
 Units: \$/Unit
 Expected Production Costs represent beliefs among market participants about the unit costs of production (variable and fixed, including normal profit margins). Expected Costs therefore represent beliefs about what a 'fair' price would be, or the long-run equilibrium price. Expected costs adjust to the actual costs with a delay representing the time required for gain information and adjust beliefs about costs. Exogenous for partial *model* test.
- (13) FINAL TIME = 3650
 Units: Day
 The final time for the simulation.
- (14) G Imp A=
 0
 Units: **undefined**
- (15) G Imp B=
 1.2
 Units: **undefined**
- (16) G Imp End=
 1490
 Units: **undefined**
- (17) G Imp Start=
 1470

Units: **undefined**

- (18) $G \text{ Impact} =$
 $G \text{ Imp A} + \text{RAMP}(((G \text{ Imp B} - G \text{ Imp A}) / (G \text{ Imp End} - G \text{ Imp Start})), G \text{ Imp Start},$
 $G \text{ Imp End}) + \text{RAMP}(((0.6) - (1.2)) / (2201 - 2200)), 2200, 2201) + \text{RAMP}((($
 $(-0.8) - (0.6)) / (2990 - 2970)), 2970, 2990)$
 Units: **undefined**

- (19) $I \text{ Impact} =$
 0
 Units: **undefined**
 $\text{RAMP}(((D \text{ Imp B}) - (D \text{ Imp A})) / ((D \text{ Imp End}) - (D \text{ Imp Start}))),$
 $D \text{ Imp Start}, D \text{ Imp End})$

- (20) Indicated Price = A FUNCTION OF("Effect of Demand/Supply Balance on Price"
 ,Maximum Price,Minimum Price)
 Indicated Price=
 $\text{MIN}(\text{Maximum Price}, \text{MAX}(\text{Minimum Price}, \text{Price} * \text{"Effect of De-}$
 $\text{mand/Supply Balance on Price"}$
 $))$
 Units: \$/Unit
 Beliefs about the underlying equilibrium price are never less
 than a minimum price level capturing the variable costs of goods.

- (21) Initial Price=
 60
 Units: \$/Unit
 The initial price of the commodity. Used to initialize unit
 costs and price expectations so the system begins in equilibrium.

- (22) INITIAL TIME = 0
 Units: Day
 The initial time for the simulation.

- (23) Maximum Price=
 250
 Units: **undefined**
 Trader's and market researchers do not believe prices can raise
 above that price.

- (24) Minimum Price=
 Unit Variable Cost
 Units: \$/Unit
 Trader's do not believe prices can fall below the variable cost
 per unit of production.

- (25) $P \text{ Impact} =$
 0
 Units: **undefined**

- (26) Reference Price=
100
Units: **undefined**
- (27) Reference Supply=
125
Units: Units/Period
Reference supply is 100 units/period.
- (28) Regressors = A FUNCTION OF(B Impact,D Impact,E Impact,G Impact,I Impact ,P Impact,W Impact)
Regressors=
(0 + STEP(B Impact, 1460))*RealBIP
+D Impact*RealDAX
+G Impact*RealSubstitutionsmarktWachstum
+P Impact*Price
+I Impact*RealImport
+ E Impact*RealExport * 10
+ W Impact*RealWetter
Units: **undefined**
- (29) SAVEPER =
TIME STEP
Units: Day [0,?]
The frequency with which output is stored.
- (30) Sensitivity of Price to Costs=
0.5
Units: Dimensionless
Controls the response of price to discrepancies between the expected price and the expected cost of production.
- (31) "Sensitivity of Price to Demand/Supply Balance"=
1
Units: Dimensionless
Determines the response of the indicated price to imbalances between demand and supply.
- (32) Supply = A FUNCTION OF(Reference Price,Reference Supply,Supply Elasticity)
Supply=
Reference Supply * (Price/Reference Price)^Supply Elasticity
Units: Units/Period
Supply is given by a constant elasticity supply curve. Supply equals the reference supply at the reference price.
- (33) Supply Elasticity=
0.5
Units: Dimensionless
The elasticity of supply (the fractional change in supply for a given fractional change in price)

- (34) TIME STEP = 1
Units: Day [0,?]
The time step for the simulation.
- (35) Time to Adjust Traders' Expected Price=
1
Units: Years
Trader's belief about the underlying equilibrium price adjust to actual prices over this period.
- (36) Traders' Expected Price= INTEG (
Change in Traders' Expected Price,
Initial Price)
Units: \$/Unit
The price market makers and traders believe would clear the market if demand and supply were in balance, and no other pressures to change price existed.
- (37) Unit Variable Cost=
60
Units: \$/Unit
Unit Variable Costs of production are exogenous. A variety of test inputs allow users to try different patterns, including a step, pulse, sine wave, and random noise.
- (38) W Imp A=
0
Units: **undefined**
- (39) W Imp B=
5
Units: **undefined**
- (40) W Imp End=
2990
Units: **undefined**
- (41) W Imp Start=
2970
Units: **undefined**
- (42) W Impact=
W Imp A+RAMP(((W Imp B-W Imp A) / (W Imp End-W Imp Start)), W
Imp Start,
W Imp End)
Units: **undefined**

1 Anhang X Transkription der Interviews

2 Transkription von Interview PE1

3 Interviewer: Marcel-Philippe Breuer

4 Transkribiert von Chung Ni Jamie Wong

5

6 Abkürzungen:

7 I = Interviewer

8 B = Befragter

9

10

11 Beginn:

12

13 I: In dem Browserfenster mit dem Neo4j, wo die Kreise und Pfeile sind,
14 konntest du es nachvollziehen, was für Elemente da sind und was sie
15 besagen? (0:00:16)

16

17 B: Also ich finde sie sind selbsterklärend und sie zeigen die Bezie-
18 hungen.

19

20 I: Und in dem rechten Fenster (zeigte auf dem R-Output), wie konntest
21 du denn mit dem Output umgehen? Konntest du es nachvollziehen? Ha-
22 ben dir Informationen gefehlt oder sind auch welche überflüssig?
23 (0:00:42)

24

25 B: Also vielleicht kann man die Infos ausgeben, dass sich eben nichts
26 verändert hat und für mich war es erstmal schwer zu verstehen, es
27 kommt jetzt nichts und das heißt, es hat sich nichts verändert. An-
28 sonsten ist es ganz übersichtlich dargestellt und es wird alles er-
29 läutert. Die einzelnen Variablen stehen da und wie sie einfließen,
30 Standardabweichung, t-Werte, ja.

31

32 I: Glaubst du, dass es Fehlanpassung gegeben hat, also dass, was dir
33 rechts der Prototyp ausgespuckt hat, mal ein Fehler dabei war?
34 (0:01:34)

35

36 B: Glaube ich nicht.

37

38 I: Warum? (0:01:40)

39

40 B: Ähm habe mich jetzt einfach mal auf System verlassen. Also die
41 Standardabweichungen waren ja wirklich sehr klein.

42

43 I: Aber wie hättest du es prüfen können? (0:01:53)

44

45 B: Nur anhand der Werte, die hier stehen.

- 46
- 47 I: Und im Vensim? Hättest du damit Möglichkeiten gehabt, um das Ergeb-
48 nis zu prüfen? (0:02:02)
- 49
- 50 B: Ich habe es ja theoretisch so blind übernommen und es im Vensim
51 eingetragen.
- 52
- 53 I: Du hast dir schon die Entwicklung nochmal angeschaut. Hast du ver-
54 sucht nachzuvollziehen, ob die rote und die blaue Linie überein-
55 stimmen? (0:02:24)
- 56
- 57 B: Ja stimmt. Also da hat es ja gepasst, also von daher habe ich mich
58 auch darauf verlassen, dass die Werte richtig sind.
- 59
- 60 I: Also hast du jetzt keine Fehlanpassung festgestellt, dass da ein
61 Fehler drin gewesen ist, das was ausgespuckt wurde von dem Proto-
62 typ? (0:02:29)
- 63
- 64 B: Ne, habe so nicht gesehen.
- 65
- 66 I: Konntest du nachvollziehen, was der Prototyp macht? (0:02:53)
- 67
- 68 B: Hier? (Auf R-Output gezeigt)
- 69
- 70 I: Sowohl als auch, also alle drei Fenster. Alle drei sind ja im Prin-
71 zip der Prototyp. (Auf Neo4j, Observer und R-Output gezeigt)
72 (0:03:02)
- 73
- 74 B: Also das (auf Observer gezeigt) am wenigstens. Ja ich denke, dass
75 die Daten, die hier ausspuckt, übernimmt es dann in Neo4j. Das ist
76 ein bisschen schöner dargestellt finde ich, ein bisschen einfacher
77 zu interpretieren. Wenn man eben auf die einzelnen Variablen gehen
78 muss und der Wert wird angezeigt, ist natürlich schöner als dieser
79 R-Output weil der sieht ja immer bisschen viel aus.
- 80
- 81 I: Aber irgendwann, ab dem dritten Fall, hast du dir den Neo4j nicht
82 mehr angeschaut. Also dass die Zusammenhänge, wie die Variablen
83 wirken, dass hast du in Neo4j wahrgenommen, weil sie ja da ver-
84 knüpft sind. Aber die Stärke der Wirkung hast du denn aus R über-
85 nommen, also die Werte, wie stark der Parameter ist, weil da ist
86 eben tabellarisch dargestellt und dadurch auch für den Menschen
87 glaube ich auch besser zu erfassen. (0:04:23)
- 88
- 89 B: Ja.
- 90
- 91 I: Was für eine Methode hat der Prototyp da angewandt, um zum Ergebnis
92 zu kommen? Weißt du es? Oder hast du es gelesen? Also rechts wie
93 kommt er auf die Ergebnisse da? (0:04:43)
- 94

- 95 B: Ja er macht so ein Soll-Ist-Vergleich und dann diese Strukturverän-
96 derung wenn eine Abweichung vorkommt und dadurch dass eben eine
97 neue Variable eingeflossen ist.
98
- 99 I: Und wie kommt er auf dieses Ergebnis, dass jetzt eine neue Variable
100 fließt? Was ist es für eine Methode, die der Prototyp hier anwen-
101 det? (0:05:14)
102
- 103 B: Eine Regressionsanalyse?
104
- 105 I: Wie du eben gesagt hast, er macht eine Abweichungsanalyse und man
106 kann auch in dem Prototyp definieren, wann eine Abweichung als sol-
107 che gewertet wird, also standardmäßig ist 5% gesetzt. Es stand auch
108 in diesem Output irgendwann mal. Würdest du dir auch mehr Einstell-
109 möglichkeiten wünschen, um die Modelle anzupassen und Soll-Ist-
110 Vergleich zu machen? (0:05:55)
111
- 112 B: Ich denke, dass mit dem 5% ist man eigentlich in einem ganz guten
113 Rahmen. Wenn man es noch genauer will, muss man dann denke ich mal
114 den 1% Level nehmen.
115
- 116 I: Das kann man auch machen. Also den Grenzwert kann man einstellen,
117 wann eine Abweichung als solche gewertet wird. Meine letzte Frage.
118 Momentan schreibt er dir die Ergebnisse dort in Neo4j, also mit den
119 Bubbles und Verknüpfungen und das ist ja quasi nochmal eine zweite
120 Präsentation von dem Vensim-Modell und du musstest ja übertragen in
121 Vensim. Also die Ergebnisse von Neo4j nehmen und dann in Vensim
122 eintragen. Stell dir vor, der Prototyp würde automatisiert arbei-
123 ten. Das heißt, er würde die Ergebnisse direkt in Vensim übernehmen
124 und die Formeln würden sofort angepasst werden so wie es eben im
125 Prototyp errechnet, wie es eben sein soll, und würde das auch
126 eben sofort machen, wenn dann eine Abweichung festgestellt
127 wird. Also man bräuchte auch nicht mehr vorher alles markieren,
128 wegspeichern, sondern das würde sofort umgesetzt werden. Es würde
129 sofort ein neuer Wert reinkommen. Er würde analysieren, ist eine
130 Abweichung da. Wenn ja, würde er rechnen und die Ergebnisse weg-
131 schreiben. Das heißt im Prinzip wäre das *Model* dauerhaft an der Re-
132 alität angepasst. Wie würdest du so ein Fall bewerten? Ist es gut
133 und würde dir was fehlen? (0:07:34)
134
- 135 B: Ja es ist natürlich eine Arbeitserleichterung in dem Sinne. Man
136 müsste nicht mehr ständig hin und her klicken. Andererseits wenn
137 man es selbst macht und hat man ein bisschen Überblick, welche Va-
138 riablen fließen jetzt ein, was hat sich da geändert und gegebenen-
139 falls auch mal selbst nachdenken und schauen, ob es wirklich plau-
140 sibel ist. Es könnte ja immer Fehler darunter laufen, deswegen wä-
141 ren es gut, dass man es stichprobenartig mit reinguckt und sich
142 nicht voll blind darauf verlassen. An sich gut, klar, wenn es lau-
143 fend direkt anpasst und man soll trotzdem ein bisschen Kontrolle
144 haben.
145
- 146 I: Um so eine Kontrolle zu ermöglichen, welche Informationen würdest
147 du denn haben wollen? Was würde dir einfallen oder würde dir was
148 einfallen? Also wenn er automatisiert arbeitet, man muss es ja
149 nicht stichprobenartig kontrollieren. Man könnte auch zeitweise In-

150 formationen vom Prototyp bekommen, dass wieder was angepasst wurde
151 zum Beispiel. Also wenn es automatisiert geschieht, was für In-
152 formationen wären dafür sinnvoll, dass du dann sagst, okay ich
153 brauche weniger zu kontrollieren, was denn der Prototyp selbst
154 macht, sondern ich vertrauen ihm damit. Also was für Informationen
155 bräuchtest du dann für Status-Update? (0:09:04)

156

157 B: Also eben was sich da geändert hat, welche Variablen betroffen
158 sind, vielleicht welche die neu dazugekommen sind und was ich hier
159 in einem Fall auch schon mal hatte, dass es von Positivem auf das
160 Negative gegangen ist. Das ist ja in dem Sinne schon wichtige In-
161 formationen.

162

163 I: Okay vielen Dank. (0:09:45)

164

1 Transkription von Interview PE2

2 Interviewer: Marcel-Philippe Breuer

3 Transkribiert von Chung Ni Jamie Wong

4

5

6 Beginn:

7

8 I: Du hast jetzt häufiger, also du hast festgestellt, dass häufig das
9 gleich Ergebnis dir mitgeteilt wurde, wo du dich gefragt hast, wa-
10 rum sagt er mir das? Hast du dir das gefragt? (0:00:17)

11

12 B: Ja.

13

14 I: Hast du eine Antwort darauf? (0:00:19)

15

16 B: Warum er mir immer das gleich Ergebnis liefert? Ich denke, weil die
17 richtig waren, dass man da nichts anpassen muss.

18

19 I: Ok. Also du hast es so interpretiert, der gibt dir die gleichen
20 Werte zurück wie vorher auch, weil es sich nichts verändert hat.
21 Aber was hat er um die Ergebnisse rum noch geschrieben? (0:00:52)

22

23 B: Also dass er so und so viele Abweichungen gefunden hat und die t-
24 Werte.

25

26 I: Aber wie ist es vereinbart mit deiner These von eben, dass er als
27 Bestätigung dieses Modells, also diese Werte nochmal wiedergibt,
28 wenn er vorher sagt dass er Abweichungen gefunden hat? (0:01:20)

29

30 B: Also man kann aus dem Output nichts auslesen, was noch angepasst
31 werden kann. So kann ich jetzt nicht anpassen, wenn es immer das
32 Gleiche ist. Dann kann ich mir ja schlecht was ausdenken.

33

34 I: Ja, also was gefehlt hat, du hast am Anfang das Substitutionsmarkt-
35 wachstum vernachlässigt. Du hast es erst hinzugenommen als dann
36 auch das Wetter hinzugekommen ist. Dann hast du gemerkt: okay, auf
37 einmal stehen vier Variablen, die Einfluss auf die Nachfrage haben.
38 Erst zu dem Zeitpunkt hast du dann diese Strukturveränderung durch-
39 geführt. Das heißt, Variable A, die haben auch einen Einfluss auf
40 die Nachfrage. Und vorher war es eben so, dass Substitutionsmarkt-
41 wachstum schon einen Einfluss hatte und da hast du eben auf DAX und
42 Export konzentriert. (0:02:13)

43

44 B: Das ist mir später aufgefallen.

45

46 I: Also es ist interessiert, warum es dir dann nicht aufgefallen ist.
47 (0:02:20)

48

- 49 B: Weil man immer nur das Gleiche gemacht hat und dann war man einfach
50 so drin.
51
- 52 I: Das heißt, du hast am Anfang Parameteranpassungen gemacht und dann
53 hast du gedacht, das muss ich jetzt auch weitermachen. (0:02:36)
54
- 55 B: Ich habe gar nicht gemerkt, dass da ein dritter Parameter gekommen
56 ist, beziehungsweise erst später.
57
- 58 I: Und dann ist es dir eben aufgefallen, als das Wetter, die vierte
59 Variable, neu hinzugekommen ist. (0:02:50)
60
- 61 B: Ja genau.
62
- 63 I: Deswegen hat der Prototyp dir immer die gleichen Vorschläge ge-
64 macht. Er hat gemessen, dass da eine Abweichung ist, weil das Sub-
65 stitutionsmarktwachstum hat gefehlt. Es ist nicht mit in die Plan-
66 werte eingeflossen, weil du die Formeln vorher nicht angepasst
67 hast. Deswegen hat er häufiger mal dir immer wieder das Gleiche ge-
68 sagt. Du hast eben nur auf DAX und Export geachtet. Das ist der
69 Hintergrund. Um da entgegenzuwirken, könnte man beispielsweise die
70 Variablen nochmal separieren, dass man als Tabelle mit allen Vari-
71 ablen, sondern nur das, was neu dazugekommen ist, also hier das
72 Substitutionsmarktwachstum. Das könnte zum Beispiel so eine Situa-
73 tion verhindern. Wie siehst du es oder was meinst du, was hätte man
74 machen können, damit es nicht passiert? Also dass man das Substitu-
75 tionsmarktwachstum nicht überliest?(0:04:13)
76
- 77 B: Schwierig. Ich glaube, wo es direkt aufgefallen wäre bei Neo4j,
78 wenn ich es jedes Mal durchgeführt hätte, weil da sieht ja man di-
79 rekt, okay da sind vier Dinge, die jetzt doch beeinflussen. Aber
80 ich habe es nicht durchgeführt.
81
- 82 I: Aber das hast du ja zwei oder drei Mal gemacht und ich habe dich
83 extra gefragt, was siehst du da? Da hast du erklärt, bist zum DAX
84 gegangen, zum Export gegangen und du hast dir die Pfeile angeschaut
85 und erklärt, was das ist. Ich dachte, ok da kommt noch eine dritte
86 aber ich wollte auch nicht zu direkt nachfragen. Das wäre zu offen-
87 sichtlich gewesen. (0:04:55)
88
- 89 B: Ja stimmt.
90
- 91 I: Auch wenn du die Variablen übersehen hast im Neo4j, den Output hast
92 du verstanden? Also dass die und die Variablen wirken, dass er die
93 Verbindungen darstellen an den Pfeilen. Fehlen dir da noch Informa-
94 tionen oder hättest du gerne, noch weitere Informationen gezeigt
95 bekommen? (0:05:43)
96
- 97 B: Also ich glaube für Leute, die so einen Output schon kennen aus dem
98 Studium, ist es einfach zu lesen. Aber wenn jemand gar keinen Hin-
99 tergrund hat, dann würde ich es auch komplett nicht verstehen.

100

101 I: Meinst du jetzt den rechten Output, den tabellarischen? (0:06:06)

102

103 B: Ja hier müsste man wissen, dass der Parameter quasi ein Beitrags-
104 faktor ist.

105

106 I: Aber konntest die Tabelle rechts auch nachvollziehen? (0:06:30)

107

108 B: Ja.

109

110 I: Ok, gut. Fehlen dir noch Information rechts oder sind da zu viele
111 Information oder wie würdest du generell die Information bewerten?
112 (0:06:43)

113

114 B: Genutzt habe ich eigentlich nur den Estimate und den t-value. Aber
115 standard error und den prob-Wert, ich vermute, dass es das ist, der
116 „pr.“, hatte ich in dem Moment nicht gebraucht.

117

118 I: Und oberhalb und unterhalb der Tabelle, also die textuellen Sachen
119 quasi? (0:07:00)

120

121 B: Ja ist glaube ich wichtig, dass er ansagt, wie viele Abweichungen
122 er gefunden hat, welche Variable nach Abweichung er untersucht,
123 also die Nachfrage hier, ist auch wichtig. Und dass er ansagt, zu
124 welchem Zeitpunkt er zum letzten Mal kontrolliert hat. Ich wüsste
125 nicht, was hier noch fehlt.

126

127 I: Meinst du, dass der Prototyp auch mal Fehlanpassung durchgeführt
128 hat? Ist es dir was aufgefallen oder glaubst du, dass diese Anpas-
129 sungsvorschläge, die er gemacht hat, nicht korrekt gewesen sind?
130 (0:07:45)

131

132 B: Ist mir so nicht aufgefallen, ne.

133

134 I: Hast du nachvollziehen können, was der Prototyp macht?
135 (0:07:55)

136

137 B: Er prüft ja eigentlich die eine Variable, die Nachfrage. Das ist
138 dann die exogene Variable und fügt dann die Variablen hinzu, wo er
139 glaubt, dass sie die Nachfrage beeinflussen und dann prüft er, ob
140 sie einen signifikanten Einfluss haben und schmeißt er die aus dem
141 Modell oder nicht. Und dann sagt er nur noch die Gewichtung der
142 einzelnen Variablen auf die exogene Variable.

143

144 I: Und methodisch macht er das mit? (0:08:31)

145

146 B: Eine Regression.

147

- 148 I: Man kann dort auch Grenzwerte definieren bei dem Prototyp, also
149 wann er eine Abweichung als solche bewertet. Standardmäßig ist es
150 dort 5% eingestellt und steht auch auf dem einführenden Text. Wür-
151 dest du dir auch andere Einstellungsmöglichkeiten wünschen oder
152 fällt dir da was ein, was du gerne verändern können würdest?
153 (0:09:01)
- 154
- 155 B: Also ich glaube standardmäßiger 5% ist auch schon sehr genau. Wenn
156 man noch genauer haben möchte, kann man auch auf 1% runtergehen.
- 157
- 158 I: Aber würdest du andere Sachen einstellen können möchtest? Also
159 nicht nur Grenzwerte definieren, sondern auch andere Sachen, die
160 dir der Prototyp mitgeben könnte, die für den Anpassungsprozess
161 sinnvoll sein könnte? (0:09:33)
- 162
- 163 B: Nee ich glaube nicht.
- 164
- 165 I: Momentan ist es ja so, dass der Prototyp seine Ergebnisse in Neo4j
166 abschreibt, also mit den Bubbles und die Verbindungen dazu. Von
167 dort musst du ja die Ergebnisse entnehmen und in Vensim in dem Mo-
168 dell dann selbst anpassen. Stell dir vor, der Prototyp würde auto-
169 matisiert arbeiten. Das heißt, er nimmt die Ergebnisse, was er be-
170 rechnet hat in der Regression und verändert das sofort im Vensim,
171 also automatisiert, dass du dann gar nicht selbst machen musst,
172 dass du von dem Prozess nichts mitbekommst. Jedes Mal, also auch
173 dieses Abspeichern, aus der Tabelle und abspeichern, auch nicht ma-
174 chen musst. Da sind Schnittstellen definiert. Der kriegt sofort,
175 wenn neuer Tageswert da ist, kann vergleichen, ob Abweichungen da
176 sind und Regression durchführen, was ein besseres Modell wäre. Und
177 wenn er was gefunden hat, wird auch sofort weggeschrieben. Das
178 heißt, im Prinzip versucht er dauerhaft an der Realität anzupassen,
179 immer automatisiert. Wie würdest du so generell so etwas finden? So
180 autonomes und automatisiertes System, das dauerhaft das Modell an-
181 passt. (0:11:01)
- 182
- 183 B: Finde ich eine gute Idee. Was ich aber berücksichtigen würde ist,
184 dass man da schon darauf zugreifen kann, dass man dann die Tabellen
185 sieht oder einsehen kann, wenn beispielsweise einem irgendwie auf-
186 fällt, dass die Werte extrem nach oben, die Nachfrage extrem nach
187 oben geht, dass man dann in die Tabelle reingehen und analysieren
188 kann, woran lag es jetzt. Aber nicht dass ich mich darum kümmern
189 muss, dass die Werte richtig eingetippt werden müssen. Das würde
190 ich schon gut finden, wenn es von alleine geht, also automatisiert.
- 191
- 192 I: Ich würde es als Nachvollziehbarkeit definieren, was du jetzt ge-
193 nannt hast (0:11:47)
- 194
- 195 B: Ja genau.
- 196
- 197 I: Das heißt, da sind auch Informationen oder Rückmeldung, die du er-
198 warten würdest, dass du das Ganze dann auch akzeptieren würdest, so
199 ein automatisiertes System. Also dass du Rückmeldung bekommst, fol-
200 gende Änderung haben stattgefunden aufgrund von. Also im Prinzip
201 ähnlich wie rechts eigentlich ne. Die Tabelle nicht so schön, aber

202 die Art der Informationen wären das, was man verwenden könnte als
203 Bericht, wenn man das automatisieren könnte. (0:12:20)

204

205 B: Ja genau.

206

207 I: Vielen Dank. (0:12:22)

208

1 Transkription von Interview PE3

2 Interviewer: Marcel-Philippe Breuer

3 Transkribiert von Chung Ni Jamie Wong

4

5

6 Beginn:

7

8 I: In dem linken Fenster hier hast du ja Kreise gehabt und Verbindun-
9 gen dazwischen. Konntest du nachvollziehen, was da angezeigt wird?
10 (0:00:14)

11

12 B: Ja das konnte ich nachvollziehen. Das hat er im Endeffekt auch das
13 wiedergespiegelt, wie der vorangegangene Teil im Output zu sehen
14 war. Von daher war es selbsterklärend.

15

16 I: Was ist der Unterschied zwischen den beiden Outputs, zwischen
17 rechts und links? (0:00:30)

18

19 B: Im rechten haben wir ja einen zahlenbasierten Output und im linken
20 einen Verbindung ausgerichteten Output, der keine Größen angibt.

21

22 I: Also welche würdest du eher präferieren? (0:00:54)

23

24 B: Ich würden den rechten präferieren, also den mit Zahlen. Im Linken
25 würde glaube ich im großen Vorteil, wenn man die Zahlen mit einbin-
26 den könnte.

27

28 I: Die sind ja zu sehen. Aber ist umständlicher als rechts. Also es
29 ist mir auch aufgefallen, dass du gewissenhaft den Output gelesen
30 hast. Du hast den Text auch durchgelesen und nicht nur die Tabelle
31 angeschaut, nicht nur die Koeffizienten sondern auch den Text
32 durchgelesen. Hast du den Output rechts, also die Tabelle nachvoll-
33 ziehen können? Hast du es verstanden? (0:01:34)

34

35 B: Das habe ich im Großen und Ganzen nachvollziehen können. Natürlich
36 im Wesentlichen habe ich in dem späteren Teil eher den unteren Teil
37 konzentriert. Aber die Aussagen, die getätigt werden, sind auch
38 nachvollziehbar gewesen.

39

40 I: Haben dir Informationen gefehlt? Oder würdest du dir was wünschen?
41 Oder ist etwas unklar? (0:02:00)

42

43 B: Also für den jetzt gerade durchgeführten Zweck nicht. Ich bin damit
44 zufrieden.

45

46 I: Hat dir nicht etwas gefehlt, dass wenn es keine Änderung gibt, dass
47 eine Meldung kommt, von wegen dass keine Änderung ist?

48

- 49 B: Also im ersten habe ich mich auch da aufgehängt, aber es lag ein-
50 fach nur daran, dass ich den Satz danach geschrieben war nicht ge-
51 lesen hatte. Später dann nicht mehr. Aber in der Tat, wenn da ein
52 Satz stehen würde: keine relevante Abweichung, es wäre wahrschein-
53 lich gerade dann gut. Zwischenzeitlich habe ich dich auch ein, zwei
54 Mal gefragt, braucht das jetzt einfach mal länger oder so. Wenn man
55 Feedback hätte, sozusagen, die Berechnungen sind abgeschlossen, es
56 wäre mit Sicherheit nicht verkehrt.
- 57
- 58 I: Glaubst du, dass es Fehlanpassung gegeben hat von dem Prototyp?
59 Also dass dir etwas falsches vorgeschlagen wurde? (0:03:06)
- 60
- 61 B: Glaube nicht.
- 62
- 63 I: Kannst du das denn beurteilen irgendwie? (0:03:15)
- 64
- 65 B: (...) Ich wüsste ehrlich gesagt nicht wie. Ich habe mir ja am An-
66 fang die Outputs wieder angeguckt, also zumindest in der ersten
67 Hälfte. Aber wie aussagekräftig sie dann sind im Zusammenhang un-
68 tereinander ist auch die Frage. Also wie ich es genau beurteilen
69 könnte, wüsste ich jetzt nicht.
- 70
- 71 I: Man kann ja rechts bei dem Prototyp immer Grenzwerte definieren,
72 also wann eine Abweichung als solche gewertet wird. Es ist momentan
73 auf 5& eingestellt. Würdest du dir auch andere Einstellmöglichkeiten
74 wünschen, dass man noch andere Sachen definieren kann, die man
75 dem Prototyp mitgibt oder konfigurieren möchte? Fällt dir da was
76 ein? (0:04:18)
- 77
- 78 B: (...) Vielleicht könnte man noch ein bisschen die Entwicklung ange-
79 ben, also wie es im Vergleich zum Mal davor war direkt. Also klar,
80 dass es jetzt so übersichtlich gemacht, dass man mit einem Blick
81 auch sehen kann. Aber wenn zum Beispiel noch eine Zeile wäre, hier
82 ist eine Abweichung, der Unterschied hier ist in dem Fall -1 oder
83 so, das vielleicht noch. Aber sonst was Anpassung angeht, wüsste
84 jetzt nicht was man sonst noch so machen könnte außer eben den Ran-
85 ge, ab wann eine Abweichung gilt.
- 86
- 87 I: Ist auch nachvollziehbar. Momentan ist ja quasi redundante Informa-
88 tion wenn jetzt bei drei Variablen sich nur eine verändert hat,
89 werden trotzdem alle drei Werte wieder dargestellt. Also das könnte
90 man zum Beispiel in einer Zeile/ (0:05:40)
- 91
- 92 B: Genau, also wenn man hier am Ende hätte sagen könnte, hier sind die
93 Veränderungen. Dann sieht man die Veränderungen, man würde wahr-
94 scheinlich dann auch etwas schneller arbeiten.
- 95
- 96 I: Momentan ist es ja so, dass der Prototyp die Ergebnisse in Neo4j
97 schreibt mit den Pfeilen und den Blasen und von dort musst du jedes
98 Mal die Ergebnisse entnehmen und in Vensim übertragen und dort an-
99 wenden. Stell dir vor, der Prototyp würde automatisiert machen. Das
100 heißt, er würde automatisiert die Ergebnisse in Vensim schreiben
101 und das Modell wird dort sofort angepasst werden. Aber auch dieser

102 Prozess, den du vorher gemacht hast, dass du die Tabelle genommen
103 hast, weggespeichert hast und es ausgeführt hast, das würde automa-
104 tisiert. Das heißt, er würde jeden Tag, wenn ein neuer Wert rein-
105 kommt, würde er das erfassen, würde prüfen, ob eine Abweichung vor-
106 handen ist und würden dann gegebenenfalls das Modell anpassen und
107 auch direkt die im Modell umsetzen. Das heißt, das würde dann kon-
108 tinuierlich angepasst werden und geprüft werden. Wie würdest du das
109 dann bewerten oder einschätzen, finden? (0:06:54)

110

111 B: Ich finde das, würde es dann erstmal als gut einschätzen. Ich glau-
112 be wichtig dabei ist, dass man in stichprobenartig sich ein Zyklus
113 auch mal aussuchen kann, der dann komplett dargestellt wird, eben
114 auch gerade dieser Schritt mit dem Anpassen der Funktion, dass man
115 da mal gucken kann, ok, ich habe hier meinen Output und er hat es
116 genau richtig angepasst. Aber an sich kann ich mir das sehr gut
117 vorstellen, dass es automatisiert geschieht. Dadurch das ist ja
118 auch ein/. Ja also ich hatte eben das Gefühl, für mich war es sehr
119 schnell nachvollziehbar. Also im Prinzip System ist was glaube ich
120 ganz gut, wenn man überwachen kann, dadurch dass es schnell zu se-
121 hen ist.

122

123 I: Aber du würdest schon so eine Art Bericht wäre ganz gut, was pas-
124 siert ist. (0:07:53)

125

126 B: Ja vielleicht dass man irgendwie die größeren Abweichungen mal mit-
127 geteilt bekommt, also besonders auffällige Abweichungen da sind.
128 Also ein paar Stichproben über den Tag verteilt mitgeteilt bekom-
129 men, dass man im Live sehen kann, das sind meine Werte, hat er die
130 richtig eingepflegt. Man könnte auch ein zweites Programm haben,
131 das ein Backup geben kann, da kommen auch die gleichen Werte als
132 Vergleichswert.

133

134 I: Das heißt, ein bisschen fehlt dir da auch, die Bewertung der Ergeb-
135 nisse, also wie gut und wie valide sie sind. Das fehlt momentan ein
136 bisschen oder? (0:09:38)

137

138 B: Also in dem momentan hat es mich gerade natürlich nicht gestört.
139 Wenn ich mir es jetzt überlege, dass ich mich darauf verlassen wür-
140 de, weil ich damit eben was richtig auswerten möchte. Das finde ich
141 dann, für meinen Auftrag wichtig ist, würde ich auch auf jeden Fall
142 gerne eine Rückfallsicherung haben, dass es auf einer anderen Art
143 und Weise nachgeprüft werden.

144

145 I: Also das passiert dort auch. Zumindest bräuchte man im Vorfeld die
146 Information darüber, was gemacht wird. Also einleitend gab es da
147 ein Text. Er hat da versucht darzustellen, was dieser Prototyp
148 macht, was berechnet wird. Aber halt nicht in diesem Detailgrad. Da
149 gibt es schon Auswahlprozesse und verschiedenen Kriterien, nach de-
150 nen etwas ausgewertet wird. Das heißt, es muss nicht neu hinzuge-
151 fügt werden, sondern transparenter gemacht werden und mitgeteilt
152 wird. (0:10:54)

153

154 B: Ja genau.

155

156 I: Vielen Dank. (0:10:57)
157

1 Transkription von Interview PE4

2 Interviewer: Marcel-Philippe Breuer

3 Transkribiert von Chung Ni Jamie Wong

4

5

6 Beginn:

7

8 I: Konntest du Informationen im Neo4j nachvollziehen? Das hast du näm-
9 lich gar nicht genutzt. Du hast rein textbasiert gearbeitet, also
10 rechts unten der Output. (0:00:15)

11

12 B: Ja.

13

14 I: Hat dir das gereicht? Oder was war der Grund dafür? (0:00:18)

15

16 B: Kann ich mir das nochmal angucken? Stimmt, den Schritt habe ich
17 einfach komplett vergessen und nicht daran gedacht, dass man auch
18 was damit machen können müssen.

19

20 I: Müssen nicht. Hat ja scheinbar auch so funktioniert. (0:00:37)

21

22 B: Es hat auch so funktioniert. Also ich habe einfach nicht daran ge-
23 dacht, dass ich es auch machen muss oder kann.

24

25 I: Aber den Output rechts konntest du auch nachvollziehen, also den
26 textuellen beziehungsweise den tabellarischen Output? (0:00:49)

27

28 B: Ja. Es gab ja am Anfang eine kleine Anweisung von euch, als eine
29 neue Variable dazu kam, muss man diese auch einfügen. Dann war mir
30 das System klar.

31

32 I: Haben dir da Information gefehlt auf der rechte Seite? (0:01:01)

33

34 B: Nein.

35

36 I: Also glaube du, dass es Fehlanpassung gegeben hat, also das, was
37 dir der Prototyp vorgeschlagen hat, falsch gewesen ist? (0:01:13)

38

39 B: Kann ich gar nicht einschätzen. Gar keine Ahnung.

40

41 I: Wie hättest du es prüfen können? (00:01:21)

42

43 B: Weiß ich nicht. (...) Vielleicht wenn ich mir dasselbe, also den-
44 selben Tag in demselben Jahr eingeben hätte, nochmal denselben Pro-
45 zess gemacht hätte?

46

47 I: Ja genau, dann hättest du es sehen können. Hast du nachvollziehen
48 können, was der Prototyp macht? (0:01:47)

49

50 B: Das Grundschema denke ich ja.

51

52 I: Und zwar? (0:01:57)

53

54 B: Er prüft halt die Planvorgaben, was das System sagt, und die Ist-
55 Werte. Mit den Formeln prüft er, ob die Planvorgaben tatsächlich
56 erreicht wurden. Und irgendwie hat das Programm auch noch gemerkt,
57 die Variablen müssen so und so viel zählen, damit man auf diesen
58 Soll- oder Ist-Wert rankommt.

59

60 I: Und wie hat der Prototyp das gemacht? Was für eine Methode?
61 (0:02:47)

62

63 B: Keine Ahnung.

64

65 I: Man kann dort in dem Prototyp auch einen Grenzwert definieren, also
66 wann eine Abweichung auch als solches definiert wird. Standardmäßig
67 ist es auf 5% gesetzt. Das heißt, wenn es mehr als 5% abweicht,
68 dann wird als Abweichung, also Ausreißer gewertet. Würdest du dir
69 auch andere Einstellmöglichkeiten wünschen neben dem Grenzwert, den
70 man bestimmen kann. Wüsstest du, was fehlen würde? (0:03:24)

71

72 B: Ich wüsste jetzt nicht, was das System noch kann und was man noch
73 ändern könnte.

74

75 I: Momentan ist es so, dass der Prototyp die Ergebnisse rechts, die
76 musst du manuell in Vensim einpflegen. Das heißt, das ist ein manu-
77 eller Prozess. Stell dir vor, der Prototyp würde es automatisiert
78 machen. Das heißt, er würde direkt Veränderungen im Vensim, in dem
79 Programm, mit dem du die Entscheidungen triffst, im Modell verän-
80 dern. Du bräuchtest nicht mehr die Tabelle anzeigen zu lassen, zu
81 speichern und rechts oben, den *Observer* auszuführen. Das bräuchtest
82 du alles nicht machen. Er würde auch jeden Tag neue Ist-Werte be-
83 kommen und einen Vergleich machen. Wenn eine Abweichung vorhanden
84 ist, auch Anpassungsvorschläge prüfen und es valide ist, dann auch
85 direkt ausgeführt. Wie würdest du es finden, wenn so was voll auto-
86 matisiert stattfinden würde? (0:04:33)

87

88 B: Gut. Es wird etwas erleichtern.

89

90 I: Hättest du irgendwie Bedenken oder bräuchtest du weitere Informati-
91 onen? (0:04:45)

92

93 B: Also ich denke mal, wenn es automatisch machen würde, würde schon
94 ein Fenster kommen: Ja hier die und die Zahlen wurden mit folgenden
95 Zahlen übernommen. Dann wüsste ich es ja und bin zufrieden. Wenn es

- 96 nicht stehen würde, also quasi dass es passiert ohne dass ich es
97 sehe, welche Zahlen übernommen wurden, dann würde ich es schon
98 fragwürdig finden.
99
- 100 I: Vertraust du den Werten, die da rechts rausgekommen sind? Also den
101 Paramtern (0:05:16)
102
- 103 B: Habe ich eine andere Wahl? Ich habe da keinen Blick für.
104
- 105 I: Also würdest du mehr Transparenz wollen? Was würde dir denn fehlen,
106 dass was da gemacht wurde, ist eine Regression. Das müsste dir
107 schon mal im Laufe des Studiums über den Weg gelaufen sein. Da gibt
108 ja bestimmte Gütekriterien für, wie gut eine solche Regression ist.
109 Also hier sind p-Werte angegeben, aber es gibt ja noch weitere In-
110 dikatoren. Würde dir was bringen, wenn da mehr Zahlen stehen wür-
111 den, also würdest du dem System mehr vertrauen oder? (0:05:55)
112
- 113 B: Also ich bin wahrscheinlich nicht der Ziellanwender, der letztend-
114 lich für dieses Programm gedacht wäre und ich bin auch nicht der
115 extreme Statistiker oder Mathematiker. Deswegen reichen mir die
116 Werte hier. Jemand, der Profi in dem Fach ist oder jemand, der Sta-
117 tistiker ist, würde dann wahrscheinlich noch andere Gütekriterien
118 haben wollen. Aber für mich hat es jetzt gereicht.
119
- 120 I: Also würdest du die Wahl der Kriterien oder da gewählt werden, vom
121 Anwender abhängig machen oder der Gruppe der Anwender? (0:06:29)
122
- 123 B: Ja ich denke schon. Um zu vereinfachen, sage ich jetzt Mal ein
124 Sachbearbeiter, für den würde auch nicht so relevant sein, welcher
125 t-Test verwendet wurde.
126
- 127 I: Vielen Dank für deine Teilnahme. (0:06:46)

1 Transkription von Interview PE5

2 Interviewer: Marcel-Philippe Breuer

3 Transkribiert von Chung Ni Jamie Wong

4

5

6 Beginn:

7

8 I: Es kam ja bei dir einmal vor, dass er gesagt hat, dass das Modell
9 sei nicht valide. Kannst du es nochmal bewerten oder beschreiben?
10 (0:00:17)

11

12 B: Also ich habe gedacht durch diese Meldung, dass ich was falsch an-
13 geklickt habe.

14

15 I: Nein war nicht der Fall. (0:00:26)

16

17 B: Ok. (...) Also wie ich es wahrgenommen habe?

18

19 I: Ja oder wie würdest du es interpretieren oder was da passiert ist.
20 Also was schätzt du, was ist da passiert und führt dazu, dass diese
21 Meldung gekommen ist? (0:00:49)

22

23 B: (...) Irgendwas muss anscheinend nicht gepasst haben. Ich kann mir
24 das selber nicht erklären. Also da fühle ich mich zu unsicher in
25 diesem Bereich, dass ich jetzt irgendwie genau sagen könnte, also
26 als ich das gesehen habe. Eigentlich verlasse ich mich schon auf so
27 was, auf so Outputs, aber als ich es gesehen habe, hätte ich jetzt
28 praktisch am liebsten von vorne angefangen und dann den Fehler be-
29 heben oder so.

30

31 I: Also du hättest neue Analyse angestoßen? (0:01:39)

32

33 B: Ja genau.

34

35 I: Also er hätte ja auch die Möglichkeit, dir das Modell, was er dann
36 erstellt hat, vorzuschlagen. (0:02:02)

37

38 B: Oder wenn man das dann gesehen hätte, wo genau das Problem ist oder
39 wieso er dann sagt, dass das Modell nicht valide ist, also eine Be-
40 gründung wieso.

41

42 I: Also es gibt ja zwei Möglichkeiten, entweder kriegst du Modell aus-
43 gegeben, wo man nicht weiß, ob es stimmt, also nicht valide ist,
44 oder eben erstmal sagen, nein ich gebe dir keins aus, das hat er ja
45 gemacht, weil die Signifikanz und das R-Quadrat zu niedrig ist,
46 beispielsweise. Was für einen der beiden Fälle würdest du bevorzu-
47 gen? (0:02:38)

48

- 49 B: Kein Modell.
50
- 51 I: Es war ja auch teilweise der Fall, dass schon eine Ausgabe gekommen
52 ist rechts und dann nochmal was gekommen ist mit geänderten Zahlen.
53 Wie würdest du es bewerten? Hat es dich gestört oder wie hast du es
54 wahrgenommen? (0:03:09)
55
- 56 B: Ne. Ich habe hier ja den Befehl durchgeführt dann kam hier der Out-
57 put raus und dann habe ich immer nur geguckt auf diese verschiede-
58 nen Variablen. Und dadurch, dass ich gedacht habe, ich kann nur ei-
59 ne ändern, als das Wetter hinzukam, dann habe ich nur das Wetter
60 hinzugefügt. Es war natürlich ein bisschen blöd.
61
- 62 I: Ist ja alles in Ordnung. Es gab ja noch dieser Output im Neo4j.
63 Hast du es nachvollziehen können, was dargestellt wird? (0:04:02)
64
- 65 B: Ja doch, also durch die Erklärung hat man gesehen, wie die ver-
66 schiedenen Variablen auf die Nachfrage auswirken und in welchen Ma-
67 ße dann praktisch und zu welcher Zeit.
68
- 69 I: Und den Output rechts, den tabellarischen? Den konntest du auch
70 nachvollziehen, was da angezeigt wurde? (0:04:23)
71
- 72 B: Ja ganz grob aber nur.
73
- 74 I: Wie ist es im Vergleich zu dem Neo4j? (0:04:31)
75
- 76 B: Das (auf Neo4j gezeigt) finde ich einfacher. Also so was (auf R-
77 Output gezeigt) schreckt natürlich erstmal ab. Aber man muss sich
78 natürlich erstmal genauer angucken. Weil hier ist es ja einfacher
79 dargestellt ne.
80
- 81 I: Glaubst du, dass es Fehlanpassung gegeben hat? Also das, was dir
82 das System vorgeschlagen hat, auch falsch gewesen ist? (0:05:13)
83
- 84 B: Also ich muss sagen, ich habe dem jetzt einfach vertraut und dann
85 geschaut, wie ich es am besten ist, dass es keine Abweichung mehr
86 gibt. Es kann bestimmt sein, dass ein Fehler drin war, aber ich
87 wüsste jetzt nicht, wie ich welchen Maßstab nehmen könnte, um dies
88 zu bewerten.
89
- 90 I: Glaubst du, dass es selbstständig auch prüft? (0:05:52)
91
- 92 B: Ja denke ich schon, weil es ist ja immer/ Also zum Beispiel in ei-
93 nem Jahr war es so, dass zwei Variablen gleich waren und eine hat
94 sich verändert. Und dann habe ich natürlich die, die sich verändert
95 hat, da übertragen.
96

- 97 I: Aber die Frage war ja, ob du glaubst, dass da Fehlanpassung gab,
98 also das was dir vorgeschlagen wurde, falsch gewesen ist. Also
99 meinst du, dass er auch prüft, was er dir ausgibt? Dass bevor er
100 dir das Ergebnis ausgibt, dass er die auch im Vorfeld prüft? Ob das
101 eben gültig ist? Ob das stimmt? (0:06:34)
102
- 103 B: Ja ich denke schon. Also hätte es doch gar keinen Sinn, wenn er es
104 nicht macht oder?
105
- 106 I: Ja das stimmt, es gab ja auch diesen einen Fall. (0:06:50)
107
- 108 B: Genau. Also eigentlich sagt ja dieser Error-Fall, dass es auf jeden
109 Fall der Fall sein muss, dass das Modell, das Ergebnis im Vorfeld
110 geprüft wird. Ansonsten wäre ja keine Abweichung da oder so.
111
- 112 I: Momentan ist es ja so, dass der Prototyp dir das Ergebnis im Neo4j
113 anzeigt und von dort musst du es dir ja immer manuell übernehmen
114 nach links zum Vensim. Also die Vorschläge, die er dir gibt, musst
115 du selber eintippen. Stell dir vor, der Prototyp würde das halt au-
116 tomatisiert machen. Das heißt, er könnte dann das Modell links
117 selbst verändern. Das musst nicht mehr du machen, auch nicht mehr
118 die ganzen Daten wegspeichern und das Kommando ausführen. Das würde
119 dann quasi auch automatisiert passieren. Das heißt, er versucht,
120 ständig das Modell anzupassen in Abhängigkeit der neuen Ist-Werte,
121 die da reinkommen, um das Modell möglichst realitätsnah zu halten.
122 Wie würdest du es finden? (0:07:57)
123
- 124 B: Viel einfacher. Dann hätte man ja nicht zwei Arbeitsplätze und ir-
125 gendwie sich anschauen muss. Also wenn eins ist.
126
- 127 I: Wenn er dann solche Anpassungen machen würde, automatisiert, wür-
128 dest du auch Feedback erwarten? Oder würdest du sagen, ok, ich ver-
129 traue dem System. Lass es einfach machen und es ist egal, was da
130 passiert und ob er was macht und wann. Hauptsache es ist irgendwie
131 aktuell. (0:08:32)
132
- 133 B: Ne, also ich würde dann schon erwarten, dass irgendwie Feedbackmel-
134 dung kommt, weil nur durch so was kann man nachvollziehen, wie so
135 was entstanden ist.
136
- 137 I: Also diese Nachvollziehbarkeit würdest du schon dir wünschen.
138 (0:08:46)
139
- 140 B: Ja also ich finde, gerade wenn man einem Program vertraut und nur
141 welche Zahl da hat, man muss schon irgendwie da nachvollziehen, wie
142 das zustande gekommen ist, also wie der Weg der Zahl.
143
- 144 I: Hast du jetzt nachvollziehen können, wie er auf die Zahlen kommt?
145 Rechts, also wie berechnet er so eine Variable und die Koeffizien-
146 ten, den Wert?(0:09:17)
147

- 148 B: Also wie genau der Parameter von Jahr 6,7 von Export sich geändert
149 hat?
- 150
- 151 I: Wie berechnet er den Parameter? (0:09:31)
- 152
- 153 B: Kann ich jetzt nicht nachvollziehen, beispielsweise.
- 154
- 155 I: Und das kannst du auch nicht aus dem Output herausbekommen?
156 (0:09:40)
- 157
- 158 B: Ja doch. Wenn ich jetzt die Sachen vergleiche.
- 159
- 160 I: Ne, also die Frage war ja, weißt du, WIE er das Ergebnis berechnet.
161 Also was für eine Methode dahinter steckt. (0:09:54)
- 162
- 163 B: (...) Ne, weiß ich gerade nicht.
- 164
- 165 I: Vertraust du den Werten, die da rauskommen? Also die Zahlen, die er
166 dir vorschlägt als Wirkung von den Variablen. (0:10:13)
- 167
- 168 B: Ja eigentlich schon. Also ich finde es jetzt schwer zu sagen, dass
169 ich was vertraue, weil ich habe zu wenig Hintergrund an Informati-
170 on, um jetzt irgendwie sagen zu können, dass irgendwie was falsch
171 wäre.
- 172
- 173 I: Eigentlich steht da ja auch, dass eine Regression durchgeführt
174 wird. Zur Regressionsbildung wurde zum Beispiel 50 Zeitpunkte ver-
175 wendet. Das sagt aus, dass die Regression als Methode angewandt
176 wird. (0:10:53)
- 177
- 178 B: Aso, das war die Methode?
- 179
- 180 I: Ja und das war auch die Information, die an den Anwender mitgeteilt
181 werden soll, wie bin ich auf das Ergebnis gekommen, ok ich habe ei-
182 ne Regression durchgeführt. (0:11:08)
- 183
- 184 B: Ja ok.
- 185
- 186 I: Vielen Dank für deine Teilnahme. (0:11:13)
- 187

1 Transkription von Interview PE6

2 Interviewer: Marcel-Philippe Breuer

3 Transkribiert von Chung Ni Jamie Wong

4

5

6 Beginn:

7

8 I: Du hast jetzt diesen Neo4j-Output wenig bis gar nicht benutzt. Es
9 hat ja trotzdem funktioniert scheinbar, aber wie würdest du es denn
10 interpretieren oder was meinst du, was es aussagt? (0:00:26)

11

12 B: Also um ehrlich zu sein, ich habe so verstanden, dass ich nur hier
13 was anpassen muss, deswegen habe ich es auch gar nicht versucht.

14

15 I: Ja, deswegen die Frage, wie interpretierst du es denn, das war dort
16 dargestellt ist. (0:00:38)

17

18 B: Ich glaube das ist genau das Gleiche wie hier, einfach eine andere
19 Möglichkeit ist, darzustellen. Man könnte auch hier quasi andere
20 Variablen mit einpflegen.

21

22 I: Also der Hintergrund ist, der Prototyp hat nicht die Möglichkeit,
23 direkt im Vensim die Änderungen vorzunehmen. Deswegen wird quasi
24 dort, im Neo4j, das Ganze eins zu eins präsentiert. Man kann eben
25 auch hier, das ganze Modell anzeigen lassen. Alle Variablen, die
26 hier drin sind (Vensim), werden auch hier (Neo4j) angezeigt. Das
27 ist der Grundgedanke, dass man nicht nur einen textuellen Output
28 hat, sondern auch eben das Ganze auch nochmal in seiner Abhängig-
29 keit sieht. Man sieht, dass die Nachfrage von Substitutionsmarkt-
30 wachstum und DAX beeinflusst wird, ja, aber so ist es hier doch an-
31 deres visualisiert, teilweise auch intuitiver. Du warst jetzt je-
32 mand, der eher datenbasiert gearbeitet hat aber eben war auch schon
33 jemand da, der hat visuell mehr gearbeitet hat und eigentlich fast
34 nur mit diesem Output hier (Neo4j) gearbeitet hat und weniger damit
35 (R). Gut, die Ergebnisse sind im Prinzip ähnlich. Wie würdest du
36 die vergleichen, wenn du dir jetzt die Ergebnisse hier anschaust?
37 (0:01:49)

38

39 B: Hier (Neo4j) fehlen mir quasi die quantitativen Werte. Ich sehe
40 welche Variablen in diesem Modell eingehen, aber ich sehe nicht wie
41 stark der Einfluss ist. Bei dem anderen sehe ich alles in Einem.

42

43 I: Hier siehst du schon die Parameter. Das geht schon. Aber es ist
44 schon eine komprimiertere Darstellung, also diese ganze Werte, die
45 Validität, beispielsweise die p- und t-Werte. (0:02:34)

46

47 B: Also zur Präsentation würde ich eher das (Neo4j) nehmen.

48

49 I: Der hintere Gedanke ist auch, dass eben eine Abweichung dann auch
50 in die Tiefe gehen kann. Also wenn jetzt die Nachfrage abweicht,

51 dass der Tagesverkauf abweicht, dass der Lagerbestand abweicht und
52 das eben textuelle diese Abhängigkeiten darzustellen, ist halt eben
53 schwierig. Deswegen gibt es dieser Neo4j. Für diesen Fall haben wir
54 immer nur die Nachfrage. (...) Was du noch gesagt hattest, welche
55 Zeitpunkte werden betrachtet, also du würdest gerne die Informatio-
56 nen haben, wie dieser Zeitraum bestimmt wurde zur Bildung der Re-
57 gression oder wie hast du damit gemeint? (0:03:21)

58

59 B: Ich hatte hier vor allem mal einen Satz: „Zur Regressionsbildung
60 wurden zwei verschiedene Zeitpunkte verwendet.“ Um nachzuvollzie-
61 hen, ob der wirklich nur von dem vorletzten zum letzten Zeitpunkt
62 gewählt hat und ob der wirklich nur der Zeitraum quasi dann neu
63 verwendet wird für die Regressionsbildung. So zum Beispiel, wäre es
64 interessant, wenn er zwischen altem und neuem des Jahres, also zum
65 Beispiel zwischen dem Ende des Vorjahres und dem Anfang dieses Jah-
66 res genommen, wenn nur 20 Tage, ob wirklich nur maximal 20 Tage zur
67 Regressionsbildung verwendet werden könnten. Das war eigentlich die
68 Frage.

69

70 I: Wäre da aber auch schon noch ein Hinweis sinnvoll, wie dieser Zeit-
71 raum auch ausgewählt wird, weil für dich auch nicht nachvollziehbar
72 war, wie dieser 42 zustanden kam. Also der Hintergrund ist, letzt-
73 endlich werden nur die Datenpunkte zur Bildung der Regression, wo
74 überhaupt eine Abweichung stattgefunden hat, weil da ist eine Stö-
75 rung, also irgendwas hat da stattgefunden und eine Störung ist ein-
76 getreten, deswegen werden halt nur die Zeitpunkte mit einer Abwei-
77 chung genommen. Aber es ist nachvollziehbar, dass diese Information
78 auch dienlich wäre. (...) Hast du sonst den Output dort im R, also
79 den textuellen, hast du denn nachvollziehen können? (0:04:46)

80

81 B: Abweichung ab 5% ist vorgegeben?

82

83 I: Ja, kann man auch ändern. Also das ist eine Einstellmöglichkeit. Du
84 kannst auch sagen, ich möchte ab 1% Anpassung durchführen lassen.

85

86 B: Also heißt es, letzten Endes alles was unter 5% abgewichen ist,
87 taucht jetzt hier nicht auf?

88

89 I: Ja. Aber es ist einstellbar. Dieser Grenzwert, wann eine Ursa-
90 chenanalyse angestoßen wird, das ist einstellbar. (0:05:16)

91

92 B: Und wenn man hier sieht, die Nachfrage hat 42 Mal vom Plan abgewi-
93 chen. Da werden wirklich nur die Tage betrachtet von dem vorletzten
94 und letzten Zeitpunkt, die ich ausgewählt habe.

95

96 I: Genau. Haben dir andere Informationen gefehlt? Dort im textuellen
97 Output. Den Zeitraum hast du gesagt, wie der bestimmt wird, wäre
98 noch eine Info, die da einfließen könnte. Sonst noch etwas?
99 (0:05:47)

100

101 B: Für mich war es nur manchmal/ Ich habe halt ein bisschen gewartet,
102 ob jetzt was kommt oder nicht. Manchmal wusste ich nicht, ob er

- 103 jetzt fertig geladen hat. Wahrscheinlich das hätte man irgendwo
104 oben gesehen ne? Ob es fertig geladen ist?
- 105
- 106 I: Aber er hätte auch Rückmeldung geben können, es gab keine Abwei-
107 chung. (0:06:07)
- 108
- 109 B: Genau.
- 110
- 111 I: Ja das hatten wir auch schon mal. Man hätte auch einfach Gleiche
112 wiedergeben können ohne Veränderung. (0:06:29)
- 113
- 114 B: Ja genau. Es ist ja quasi gut, dass es nicht aktualisiert ist, weil
115 es ja die gleichen Parameter hat. Aber es ist trotzdem eine Rück-
116 meldung, dass es zu dem Zeitpunkt aktualisiert ist ohne Verände-
117 rung.
- 118
- 119 I: Glaubst du, dass es auch mal Fehlanpassungen gegeben hat? Also dass
120 das System was Falsches vorgeschlagen hat? (0:06:50)
- 121
- 122 B: (...) Man kann es nicht ausschließen, aber muss ich die Technik
123 dahinter verstehen oder die Programmierung verstehen. Aber ich wür-
124 de jetzt so natürlich erstmal vertrauen. Also t-Werte haben ja ei-
125 gentlich immer gepasst.
- 126
- 127 I: Das ist dann ein Kriterium, wie dann die Validität bestimmt werden
128 kann. Konntest du denn nachvollziehen, was genau der Prototyp ge-
129 macht hat? Also was für eine Methode steckt dahinter? Wie kommt er
130 auf die Ergebnisse? (0:07:32)
- 131
- 132 B: Ist es eine (...) Regressionsanalyse?
- 133
- 134 I: Ja genau. Da gibt es auch mehr Gütekriterien, die dann noch ausge-
135 wiesen werden können. Würdest du dir auch mehr wünschen, also dass
136 es jetzt auf Koeffizientenebene. Also jeder Koeffizient hat quasi
137 diesen p-Wert und t-Wert ausgegeben. Man könnte auch Werte für das
138 gesamte Modell wie R-Quadrat ausgegeben können. Würdest du auch so-
139 was präferieren oder? (0:08:12)
- 140
- 141 B: Da habe ich zu wenig Grundwissen, um zu sagen, ob das ausreicht
142 oder nicht ausreicht.
- 143
- 144 I: Also es gibt im Prototyp auch Mechanismen, mit denen eben dieses
145 gesamte Modell bewertet wird und verworfen wird, wenn es zu
146 schlecht ist. Würdest du dir am Anfang noch mehr Informationen wün-
147 schen? Also es gab da einen einführenden Text, da stand halt nur,
148 ab welchen p-Wert es verworfen wird oder nicht auf Koeffizienten-
149 ebenen. (0:08:53)
- 150
- 151 B: Ne, dafür sind die Informationen ausreichend.

152

153 I: Momentan ist es ja so, dass die Ergebnisse, die du textuell bekom-
154 men hast, musst du jedes Mal manuell dort nochmal im Vensim einge-
155 ben und die Zeit fortlaufen lassen. Du musst eben auch die Ergeb-
156 nisse von dort wegspeichern und im *Observer* ausführen. Stell dir
157 mal vor, es würde alles wegfallen. Jedes Mal, wenn es einen aktuel-
158 len Wert gibt, wird es auch automatisiert abgegriffen werden und er
159 könnte dann auch, wenn eben Abweichungen da sind, automatisiert
160 links im Vensim anpassen. Also er versucht halt dann, dauerhaft das
161 Modell anzupassen und auch dauerhaft realitätsnah zu sein, weil das
162 ist ja das Ziel. Wie würdest du es finden, wenn so etwas vollauto-
163 matisiert stattfinden würde? (0:09:43)

164

165 B: Würde natürlich vereinfachen. Letzten Endes gerade am Anfang muss
166 natürlich auch Stichproben machen, um zu gucken, ob alles korrekt
167 ist. Aber letztendlich wäre das ein vollautomatisierter Prozess,
168 dann muss man nicht mehr die Daten eintragen. Es würde dann die
169 komplette Arbeit ersparen.

170

171 I: Stichwort Vertrauen, hast du da irgendwelche/ (0:10:06)

172

173 B: Na gut, wenn man an einer Maschine arbeitet, muss man generell ein
174 Grundvertrauen für so was haben. Mit Stichproben können das Ver-
175 trauen bestätigt werden, dass man einfach am Anfang ein paar Stich-
176 proben macht, ob die Werte auch eins zu eins übernommen wurden und
177 so. Wenn das passt, dann fortführen. Wenn nicht, dann letzten Endes
178 immer noch manuell machen.

179

180 I: Wenn so eine automatisierte Anpassung stattgefunden hat, willst du
181 es im Prinzip auch gar nicht wissen, sondern du bist zufrieden,
182 wenn im Hintergrund Prüfungen stattfinden und anpassen. Oder wür-
183 dest du dir auch einen Input oder ein Berichtähnliches wünschen?
184 (0:10:51)

185

186 B: Gut, für diejenigen, die damit arbeiten wollen mit dem Modell, ist
187 es natürlich auch interessant, welche Variablen jetzt einen Ein-
188 fluss haben oder nicht.

189

190 I: Gut, dass sehen sie ja auch im Vensim ne, es wird ja automatisiert
191 angepasst. Das heißt, das Ergebnis würde ja immer dargestellt wer-
192 den, aber weniger die Hintergrundinformationen, wie er dahin gekom-
193 men ist, vielleicht wie es vorher gewesen ist oder was sich verän-
194 dert hat. (0:11:14)

195

196 B: Ja. Für mich ist jetzt weniger, aber für Manche ist es bestimmt
197 interessant, dass er eine Veränderung sieht. Um zu sehen, kann ja
198 zum Beispiel, wenn das jeweilige Jahr einfach in einer roten Farbe
199 ist. Also das Jahr, wo eine Farbe ist, eine Veränderung stattgefunden
200 hat. Bei mir waren sieben Veränderungen, dass es sieben Mal rot
201 ist. Dann könnte man einfach hier reinklicken und sagen, ja das
202 hier wurde abgeändert und die Variable ist dazu gekommen. Dass man
203 es alles auf einem Blick hat. Für diejenigen, die es interessiert,
204 können sich noch weiter informieren lassen.

205

206 I: Also müsste man schon ein bisschen auch in Abhängigkeit des Anwen-
207 ders machen. (0:12:00)

208

209 B: Ja. Also es ist ein Vorschlag. Für normal würde es reichen, quasi
210 auf dem ersten Blick, dass man hier sieht, das ist eine Abänderung.
211 Für die, die das wollen, können mit einem Klick weiter informieren.

212

213 I: Ok. Vielen Dank. (0:12:19)

214

1 Transkription von Interview PE7

2 Interviewer: Marcel-Philippe Breuer

3 Transkribiert von Chung Ni Jamie Wong

4

5

6 Beginn:

7

8 I: Du hast die meiste Zeit jetzt hier mit dem textuellen Output gear-
9 beitet. Es gibt noch so einen Output hier. Der kannst du dir auch
10 anzeigen lassen. Verstehst du jetzt auf der Schnelle auch, was hier
11 dargestellt wird? Auch wenn du vorher nicht hiermit gearbeitet
12 hast? (0:00:23)

13

14 B: Also wenn ich das hier sehe, auf jeden Fall. Ich fand, es ist ziem-
15 lich eindeutig, was angepasst werden müsste und woran es genau
16 liegt. Das hat man auch daran gesehen, dass es einfach direkt funk-
17 tioniert hat. Wenn ich was angepasst habe, dann muss ich danach
18 nicht mehr was anpassen.

19

20 I: Wie würdest den Output links mit den Pfeilen und Kreisen interpre-
21 tieren, was das ist? (0:00:52)

22

23 B: Es wird auf jeden Fall angezeigt, welche Einflussfaktoren auf die
24 Nachfrage wirken. Ah hier werden die Parameter auch angezeigt.

25

26 I: Also wie würdest du das vergleichen mit dem textuellen Output?
27 (0:01:18)

28

29 B: Ich finde das übersichtlicher, weil hier muss man auch erst überall
30 rübergehen und hier hat man alles auf einem Blick.

31

32 I: Hat das auch Vorteile links? Oder würdest du sagen, ne brauche ich
33 nicht? (0:01:29)

34

35 B: Ich persönlich bräuchte es nicht. Ich glaube es könnte entscheidend
36 sein, wenn es ein bisschen komplexer noch ist.

37

38 I: Also wenn mehr Variablen sind? (0:01:43)

39

40 B: Genau, dass man es auf einem Blick so hat, was überhaupt darauf
41 Einfluss hat und dann hat man den textuellen ja noch.

42

43 I: Konntest du den textuellen Output rechts nachvollziehen, was da
44 ausgegeben wird? (0:02:04)

45

- 46 B: Also das Einzige, worauf ich jetzt richtig geguckt habe, waren ei-
47 gentlich immer nur diese Bewertungen und das sind ja eigentlich
48 schon die Anpassungen gewesen. So auf den Standardfehler und auf
49 die t-Statistik habe ich jetzt nicht so geschaut.
50
- 51 I: Würden dir noch Informationen dort fehlen? Oder sind das schon zu
52 viele? (0:02:41)
53
- 54 B: Vielleicht bisschen mehr Informationen darüber, wie die zustande
55 gekommen sind. Was jetzt genau die ausschlaggebenden Faktoren dafür
56 waren. Oder was sich verändert hat, dass das Substitutionsmarkt-
57 wachstum jetzt dazu kommt, oder das Wetter.
58
- 59 I: Also meinst du, dass man eher rein die Veränderung darstellen soll-
60 te? Also wenn vorher DAX und Export schon da waren, dass man dann
61 nur sagt, das Substitutionsmarktwachstum ist neu hinzugekommen und
62 nicht mehr die alten Faktoren mit auflistet, weil sie sich nicht
63 verändert haben? (0:03:26)
64
- 65 B: Ne, die würde ich glaube ich schon so lassen. Und was mich auch
66 manchmal irritiert hat, dass er gar nichts anzeigt, wenn keine An-
67 passung geben muss. Dann würde ich eher sagen, dass da irgendwie
68 steht, dass da keine Anpassung vorgenommen werden müssen oder ein-
69 fach/ Ja so was würde ich machen. Vielleicht auch bei den Variab-
70 len, wenn jetzt Export gleichbleibt und DAX gleich, dass es irgend-
71 wie gekennzeichnet ist, dass ich nur sehe, was ich jetzt anpassen
72 muss.
73
- 74 I: Also was sich wirklich verändert hat im Vergleich zum Modell oder
75 zum Zeitpunkt davor. (0:04:08)
76
- 77 B: Ja.
78
- 79 I: Konntest du denn nachvollziehen, was der Prototyp macht? Also weißt
80 du was er für eine Methode verwendet hat, um auf dieses Ergebnis da
81 zu kommen? (0:04:20)
82
- 83 B: Eine Regressionsanalyse?
84
- 85 I: Ja. Hast du denn auch nachvollziehen können, wie er die Regression
86 gebildet hat? Was er da im Hintergrund gemacht hat konkret?
87 (0:04:39)
88
- 89 B: Ne.
90
- 91 I: Würdest du die Information brauchen? Also würdest du gerne mehr
92 Information dazu haben, wie konkret die Regression gebildet wurde,
93 also zum Beispiel welche Datenpunkte, welche Zeitpunkte konkret. Da
94 steht jetzt im Output, wie viele Datenpunkte, also Zeitpunkte, für

- 95 die Regression verwendet wurden. Hier steht: „Zur Regressionsbil-
96 dung wurden 48 Zeitpunkte verwendet.“ (0:05:15)
- 97
- 98 B: Ja ich glaube es ist eigentlich schon interessant, wenn man weiß,
99 welche es waren, also welche verwendet wurden. Ich weiß nicht, also
100 spontan würde ich jetzt nicht sagen, dass mir das direkt gefehlt
101 hat. Aber man weiß, also ich, habe jetzt nicht so viel mit der The-
102 matik zu tun, wie so Modelle arbeiten im Hintergrund. Und ich weiß
103 jetzt eigentlich nicht wirklich, was er gemacht hat. Ne.
- 104
- 105 I: Also eine einführende Information wäre dann sinnvoll, wie er arbei-
106 tet. Gut, es gab so einen einführenden Text am Anfang, aber da
107 stand da nur, ab welchem P-Wert ein Regressor angenommen wird oder
108 verworfen wird. Ich glaube schon, dass du es gelesen hast. Das ist
109 eigentlich schon ein Hinweis darauf, wie jetzt eine Auswahl getrof-
110 fen wird in dem Prototyp, aber in der Art könnte eigentlich auch
111 noch mehr Informationen sein, wann ein Modell überhaupt als valide
112 angesehen wird, weil er im Hintergrund mehrere Modelle erstellt und
113 es wird eines aus diesen mehreren Modellen ausgewählt nach bestimm-
114 ten Kriterien. Wäre es denn für dich sinnvoll, so Information zu
115 bekommen, damit dieser Prozess, der da stattfindet, auch nachvoll-
116 ziehbar ist? (0:07:17)
- 117
- 118 B: Ja aber ich würde auch nicht zu viel dann dahin schreiben, weil
119 irgendwann/. Ich meine, dafür ist ja auch ein bisschen das Modell,
120 dass es da arbeitet und gibt Ergebnisse aus, mit denen man arbeiten
121 kann. Ich glaube grundsätzlich ist es schon gut, wenn man weiß, was
122 da stattfindet, aber nicht bei jedem Vorgang.
- 123
- 124 I: Also bräuchte man eine Initial gelegte Dokumentation. Glaubst du,
125 dass er auch mal Fehlanpassung gegeben hat? Also dort die Anpas-
126 sungsvorschläge, wie du da siehst, dass es mal falsch gewesen ist?
127 (0:07:59)
- 128
- 129 B: Ich glaube bei mir war es immer so, wenn ich was angepasst habe,
130 dass ich öfter mal danach nichts anpassen musste.
- 131
- 132 I: Wie bewertest du es? (0:08:13)
- 133
- 134 B: Ja bewerte es positiv. Ich glaube aber schon, dass es bestimmt mal
135 passieren kann. Aus meiner Erfahrung muss ich jetzt sagen, dass ich
136 ja nie was nachanpassen musste, also dass es immer geklappt hat,
137 was mir vorgeschlagen wurde.
- 138
- 139 I: Momentan ist es ja so, dass du die Ergebnisse, die textuelle raus-
140 kommen, übernehmen musst und manuell in Vensim eintragen musst. Das
141 Gleich ist, dass du auch jedes Mal die Werte anzeigen lassen und
142 wegspeichern und in dem schwarzen Observer ausführen musst, damit
143 die Abweichungsanalyse gestartet wird. Stell dir mal vor, dass es
144 nicht notwendig ist, also dass es automatisiert ist. Der Prototyp
145 würde mitbekommen, dass es neue Werte gibt und vergleicht die mit
146 den Planwerten. Wenn Abweichungen da sind, wird eben auch die Ana-
147 lyse durchführt, die Ursachenanalyse, und dann auch in das Modell

- 148 direkt geschrieben werden kann. Das heißt, deine manuelle Arbeit
149 würde wegfallen und es macht das automatisiert. Das heißt, das Mo-
150 dell würde auch dauerhaft kontrolliert und gegebenenfalls auch an-
151 gepasst. Auch das Gleiche, was du eben gemacht hast, eine Parame-
152 ter- oder Strukturanpassung. Wie würdest du so einen vollautomati-
153 sierten Prozess bewerten oder finden? (0:09:46)
- 154
- 155 B: Ja, an sich ist natürlich schon praktisch und auch sehr zeitspa-
156 rend. Aber was ich auf jeden Fall noch haben wollen würde, dass ich
157 immer sehe oder dass es mir irgendwie angezeigt wird, dass was an-
158 gepasst wurde. Also nicht, dass es so im Hintergrund läuft und ich
159 bekomme nicht mit, dass das Modell verändert wird oder Strukturan-
160 passungen vorgenommen werden oder die Parameter. Also das fände ich
161 schon wichtig, dass man es mitbekommt, dass angepasst wird. Aber an
162 sich sehe ich es positiv.
- 163
- 164 I: Vertraust du diesen Werte, die da auch ausgespuckt werden?
165 (0:10:28)
- 166
- 167 B: Grundsätzlich ja. Also ich glaube, das hat jeder bei so automati-
168 sierten Systemen leichter Zweifel.
- 169
- 170 I: Also meinst du, wenn ein Mensch diese Analyse machen würde, ist sie
171 da besser? Würdest du mehr vertrauen? (0:10:51)
- 172
- 173 B: Ne. Also ich glaube schon, dass Menschen mehr Fehler machen. Also
174 ich sage mal so, ich vertraue dem System mehr als wenn das jetzt
175 zehn oder zwanzig Menschen machen würden.
- 176
- 177 I: Vielen Dank. (0:11:11)

Literaturverzeichnis

- Aamodt, A. und Plaza, E. (1994). Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1), S. 39–59.
- Adam, D. (1996). *Planung und Entscheidung: Modelle, Ziele, Methoden* (4., vollst. überarb. und wesentlich erw. Aufl.). Wiesbaden: Gabler.
- Aggarwal, C. C. (2018). Text Preparation and Similarity Computation. In: C. C. Aggarwal (Hrsg.), *Machine Learning for Text* (S. 17–30). Cham: Springer International Publishing.
- Alter, S. (1980). *Decision support systems: current practice and continuing challenges*. Reading, Mass. [u.a.]: Addison-Wesley.
- Ameduri, S., Diodati, G. und Concilio, A. (2009). SMA Embedded Panel Optimized Through a Genetic Approach. *Journal of Intelligent Material Systems & Structures*, 20(13), S. 1529–1540.
- Andersson, J., Lemos, R. de, Malek, S. und Weyns, D. (2009). Reflecting on self-adaptive software systems. In: *2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems* (S. 38–47).
- Babae, R. und Babamir, S. M. (2013). Runtime verification of service-oriented systems: a well-rounded survey. *International Journal of Web and Grid Services*, 9(3), S. 213–267.
- Bäck, T. (1996). Evolution strategies: An alternative evolutionary algorithm. In: J. Alliot, E. Lutton, E. Ronald, M. Schoenauer, & D. Snyers (Hrsg.), *Artificial Evolution. AE 1995* (S. 1–20). Berlin, Heidelberg: Springer.
- Backhaus, K., Erichson, B., Plinke, W. und Weiber, R. (2018a). Clusteranalyse. In: *Multivariate Analysemethoden* (S. 435–496). Berlin, Heidelberg: Springer Gabler.
- Backhaus, K., Erichson, B., Plinke, W. und Weiber, R. (2018b). Regressionsanalyse. In: *Multivariate Analysemethoden* (S. 57–124). Berlin, Heidelberg: Springer Gabler.

- Backhaus, K., Erichson, B., Plinke, W. und Weiber, R. (2018c). Zeitreihenanalyse. In: *Multivariate Analysemethoden* (S. 125–161). Berlin, Heidelberg: Springer Gabler.
- Backhaus, K., Erichson, B., Plinke, W. und Weiber, R. (2018d). Zur Verwendung dieses Buches. In: *Multivariate Analysemethoden* (S. 3–53). Berlin, Heidelberg: Springer Gabler.
- Beheshti, Z. und Shamsuddin, S. M. (2013). A review of population-based meta-heuristic algorithm. *International Journal of Advances in Soft Computing and Its Applications*, (5), S. 1–35.
- Benjamins, A. (2017). *Automatisierte Analyse von Wirkungsbeziehungen zur situativen Entscheidungsunterstützung: Konzeption, prototypische Implementierung und Evaluation eines wissensbasierten Systems* (Dissertation). Universität Osnabrück, Osnabrück.
- Benker, H. (2014). Das Tabellenkalkulationsprogramm EXCEL. In: H. Benker (Hrsg.), *EXCEL in der Wirtschaftsmathematik: Anwendung von Tabellenkalkulationsprogrammen für Studenten, Dozenten und Praktiker* (S. 1–2). Wiesbaden: Springer Vieweg.
- Bergmann, R. und Bungert, M. (2011). *Strategische Unternehmensführung: Perspektiven, Konzepte, Strategien*. Berlin [u.a.]: Physica-Verl.
- Bhaskar, H., Mihaylova, L. und Achim, A. (2010). Video Foreground Detection Based on Symmetric Alpha-Stable Mixture Models. *IEEE Transactions on Circuits & Systems for Video Technology*, 20(8), S. 1133–1138.
- Borkar, P. und Sarode, M. V. (2018). Modality of teaching learning based optimization algorithm to reduce the consistency ratio of the pair-wise comparison matrix in analytical hierarchy processing. *Evolving Systems*, 9(2), S. 169–180.
- Bossel, H. (1994). *Modeling and Simulation*. Wellesley: A K Peters.
- Bossel, H. (2004a). *Systeme, Dynamik, Simulation: Modellbildung, Analyse und Simulation komplexer Systeme*. Norderstedt: Books on Demand.
- Bossel, H. (2004b). *Wirtschaft, Gesellschaft und Entwicklung*. Norderstedt: Books on Demand.

- Bourier, G. (2018). Zusammenhang zwischen zwei Merkmalen. In: G. Bourier (Hrsg.), *Beschreibende Statistik: Praxisorientierte Einführung - Mit Aufgaben und Lösungen* (S. 195–231). Wiesbaden: Springer Gabler.
- Breuer, M.-P. (2016). An architecture for self-adaptive model-based DSS illustrated by a reverse logistics scenario. In: S. Liu, B. Delibašić, I. Linden, & F. Oderanti (Hrsg.), *Proceedings of the 2016 International Conference on Decision Support System Technology - ICDSST 2016* (S. 54). Plymouth: EWG-DSS.
- Breuer, M.-P. (2018). Design and Ex ante Evaluation of an Architecture for Self-adaptive Model-Based DSS. In: Á. Rocha, H. Adeli, L. P. Reis, & S. Costanzo (Hrsg.), *Trends and Advances in Information Systems and Technologies - WorldCIST 2018* (Bd. 746, S. 686–696). Cham: Springer International Publishing.
- Brockhoff, K. (1977). *Prognoseverfahren für die Unternehmensplanung*. Wiesbaden: Gabler.
- Brodersen, K. H., Gallusser, F., Koehler, J., Remy, N. und Scott, S. L. (2015). Inferring causal impact using Bayesian structural time-series models. *Annals of Applied Statistics*, 9, S. 247–274.
- Brun, Y., Desmarais, R., Geihs, K., Litoiu, M., Lopes, A., Shaw, M. und Smit, M. (2013). A Design Space for Self-Adaptive Systems. In: R. de Lemos, H. Giese, H. A. Müller, & M. Shaw (Hrsg.), *Software Engineering for Self-Adaptive Systems II* (S. 33–50). Berlin, Heidelberg: Springer.
- Brun, Y., Marzo Serugendo, G., Gacek, C., Giese, H., Kienle, H., Litoiu, M., Müller, H., Pezzè, M. und Shaw, M. (2009). Engineering Self-Adaptive Systems through Feedback Loops. In: B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, & J. Magee (Hrsg.), *Software Engineering for Self-Adaptive Systems* (S. 48–70). Berlin, Heidelberg: Springer.
- Bryman, A. (2015). *Business research methods* (Fourth edition). Oxford: Oxford University Press.
- Chen, H., Lv, F., Lin, T. und Chen, S. (2009). Closed-Loop Control of Robotic Arc Welding System with Full-penetration Monitoring. *Journal of Intelligent & Robotic Systems*, 56(5), S. 565–578.

- Chen, Z. und Ellis, T. (2014). A self-adaptive Gaussian mixture model. *Computer Vision and Image Understanding*, 122, S. 35–46.
- Cheng, S.-W., Poladian, V. V., Garlan, D. und Schmerl, B. (2009). Improving Architecture-Based Self-Adaptation through Resource Prediction. In: B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, & J. Magee (Hrsg.), *Software Engineering for Self-Adaptive Systems* (S. 71–88). Berlin, Heidelberg: Springer.
- Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, 1(1), S. 104–126.
- Dalgaard, P. (2001). A primer on the R-Tcl/Tk package. *R News*, 1(3), S. 27–31.
- Data Integration Tools and Platform | Neo4j. (2018). Abgerufen am 8. August 2018 von <https://neo4j.com/data-integration/>.
- Davis, F. D., Bagozzi, R. P. und Warshaw, P. R. (1989). User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science*, 35(8), S. 982–1003.
- Del Moral, P. (1996). Nonlinear filtering: Interacting particle resolution. *Markov Processes and Related Fields*, 2(4), S. 555–580.
- DIN EN ISO 9241-11. (1999). *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze*. Berlin: Beuth.
- Döring, N. und Bortz, J. (2016a). Datenerhebung. In: N. Döring & J. Bortz (Hrsg.), *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* (S. 321–577). Berlin, Heidelberg: Springer.
- Döring, N. und Bortz, J. (2016b). Evaluationsforschung. In: N. Döring & J. Bortz (Hrsg.), *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* (S. 975–1036). Berlin, Heidelberg: Springer.
- Döring, N. und Bortz, J. (2016c). Qualitätskriterien in der empirischen Sozialforschung. In: N. Döring & J. Bortz (Hrsg.), *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* (S. 81–119). Berlin, Heidelberg: Springer.

- Döring, N. und Bortz, J. (2016d). Untersuchungsdesign. In: N. Döring & J. Bortz (Hrsg.), *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* (S. 181–220). Berlin, Heidelberg: Springer.
- Dörner, D. (1996). *Die Logik des Mißlingens: strategisches Denken in komplexen Situationen*. Reinbek: Rowohlt.
- Drazan, P. J. (1995). Model based decision support systems. In: H. H. Adelsberger, J. Lažanský, & V. Mařík (Hrsg.), *Information Management in Computer Integrated Manufacturing: A Comprehensive Guide to State-of-the-Art CIM Solutions* (S. 418–426). Berlin, Heidelberg: Springer.
- Eberlein, R. L. und Peterson, D. W. (1992). Understanding models with VensimTM. *European Journal of Operational Research*, 59(1), S. 216–219.
- Eckstein, P. P. (2016). Zusammenhangsanalyse. In: P. P. Eckstein (Hrsg.), *Angewandte Statistik mit SPSS: Praktische Einführung für Wirtschaftswissenschaftler* (S. 163–196). Wiesbaden: Springer Gabler.
- Eckstein, P. P. (2019). Statistische Induktion. In: P. P. Eckstein (Hrsg.), *Statistik für Wirtschaftswissenschaftler: Eine realdatenbasierte Einführung mit SPSS* (S. 235–324). Wiesbaden: Springer Gabler.
- Edlich, S. (2011). *NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken* (2., aktualisierte und erw. Aufl.). München: Hanser.
- Elliott, M. R. (2009). Model Averaging Methods for Weight Trimming in Generalized Linear Regression Models. *Journal of official statistics*, 25(1), S. 1–20.
- Engard, N. C. (2009). LimeSurvey <http://limesurvey.org>. *Public Services Quarterly*, 5(4), S. 272–273.
- Ertel, W. (2016). Neuronale Netze. In: W. Ertel (Hrsg.), *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung* (S. 265–311). Wiesbaden: Springer Vieweg.
- Fahrmeir, L., Heumann, C., Künstler, R., Pigeot, I. und Tutz, G. (2016a). Multivariate Deskription und Exploration. In: L. Fahrmeir, C. Heumann, R. Künstler, I. Pigeot, & G. Tutz (Hrsg.), *Statistik: Der Weg zur Datenanalyse* (S. 105–164). Berlin, Heidelberg: Springer Spektrum.

- Fahrmeir, L., Heumann, C., Künstler, R., Pigeot, I. und Tutz, G. (2016b). Regressionsanalyse. In: L. Fahrmeir, C. Heumann, R. Künstler, I. Pigeot, & G. Tutz (Hrsg.), *Statistik: Der Weg zur Datenanalyse* (S. 437–475). Berlin, Heidelberg: Springer Spektrum.
- Fasel, D. (2016). Übersicht über NoSQL-Technologien und -Datenbanken. In: D. Fasel & A. Meier (Hrsg.), *Big Data: Grundlagen, Systeme und Nutzungspotenziale* (S. 109–137). Wiesbaden: Springer Vieweg.
- Fayyad, U., Piatetsky-Shapiro, G. und Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), S. 37–54.
- Filieri, A., Ghezzi, C. und Tamburrelli, G. (2012). A formal approach to adaptive software: continuous assurance of non-functional requirements. *Formal Aspects of Computing*, 24(2), S. 163–186.
- Ford, B. J., Xu, H. und Valova, I. (2013). A Real-Time Self-Adaptive Classifier for Identifying Suspicious Bidders in Online Auctions. *Computer Journal*, 56(5), S. 646–663.
- Garcia-Diaz, V., Pelayo Garcia-Bustelo, B. C. und Cueva Lovelle, J. M. (2012). MDCI: Model-driven continuous integration. *Journal of Ambient Intelligence and Smart Environments*, 4(5), S. 479–481.
- Gerdes, I., Klawonn, F. und Kruse, R. (2004). *Evolutionäre Algorithmen: genetische Algorithmen - Strategien und Optimierungsverfahren - Beispielanwendungen* (1. Aufl.). Wiesbaden: Vieweg+Teubner.
- Gluchowski, P., Dittmar, C. und Gabriel, R. (2008). *Management Support Systeme und Business Intelligence: computergestützte Informationssysteme für Fach- und Führungskräfte* (2. vollst. überarb.). Berlin: Springer.
- Goldman, R. P., Musliner, D. J. und Krebsbach, K. D. (2001). Managing online self-adaptation in real-time environments. In: R. Laddaga, P. Robertson, & H. Shrobe (Hrsg.), *Self-Adaptive Software: Applications* (Bd. 2614, S. 6–23).
- Gordon, G. W. T. und Bentley, J. P. (2002). On Evolvable Hardware. In: S. J. Ovaska & L. M. Sztandera (Hrsg.), *Soft Computing in Industrial Electronics* (S. 279–323). Heidelberg: Physica-Verlag HD.

- Görz, G., Schneeberger, J. und Schmid, U. (2014). *Handbuch der künstlichen Intelligenz* (5., überarb. und aktualisierte Aufl.). München: Oldenbourg.
- Götze, U. (1991). *Szenario-Technik in der strategischen Unternehmensplanung*. Wiesbaden: Dt. Univ.-Verl.
- Gregor, S. und Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Q.*, 37(2), S. 337–356.
- Guest, G., Bunce, A. und Johnson, L. (2006). How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods*, 18(1), S. 59–82.
- Hanke, J. E. und Reitsch, A. G. (1981). *Business forecasting*. Boston: Allyn and Bacon.
- Hao-li, C., Zhong-ke, S. und Qing-hua, F. (2006). The study of the detection and tracking of moving pedestrian using monocular-vision. In: V. N. Alexandrov, G. D. VanAlbada, P. M. A. Sloot, & J. Dongarra (Hrsg.), *Computational Science - Iccs 2006, Pt 4, Proceedings* (Bd. 3994, S. 878–885).
- Happe, M., Luebbers, E. und Platzner, M. (2013). A self-adaptive heterogeneous multi-core architecture for embedded real-time video object tracking. *Journal of Real-Time Image Processing*, 8(1), S. 95–110.
- Herrera, M., Izquierdo, J., Montalvo, I., Garcia-Armengol, J. und Roig, J. V. (2009). Identification of surgical practice patterns using evolutionary cluster analysis. *Mathematical and Computer Modelling*, 50(5–6), S. 705–712.
- Hess, D. und Summerfield, M. (2013). PyQt Whitepaper. Riverbank Computing Ltd. Abgerufen am 23. März 2019 von <https://www.riverbankcomputing.com/static/Docs/PyQt4-whitepaper/pyqt-whitepaper-a4.pdf>.
- Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2), S. 87–92.
- Hevner, A. R., March, S. T., Park, J. und Ram, S. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28(1), S. 75–105.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Michigan, Ann Arbor: University of Michigan Press.

- Hoorfar, A. (2007). Evolutionary Programming in Electromagnetic Optimization: A Review. *IEEE Transactions on Antennas & Propagation*, 55(3), S. 523–537.
- Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64(2), S. 79–102.
- Huber, F., Herrmann, A., Meyer, F., Vogel, J. und Vollhardt, K. (Hrsg.). (2007). Grundlagen zur Schätzung von komplexen Strukturgleichungen unter Verwendung von PLS. In: *Kausalmodellierung mit Partial Least Squares: Eine anwendungsorientierte Einführung* (S. 3–15). Wiesbaden: Gabler.
- Hunger, M. (2014). *Neo4j 2.0: eine Graphdatenbank für alle*. Frankfurt am Main: Entwickler.press.
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), S. 90–95.
- IEEE. (1990). *IEEE Standard 610.12-1990*. New York: The Institute of Electrical and Electronics Engineers.
- Jacob, R., Heinz, A. und Décieux, J. P. (2013). *Umfrage: Einführung in die Methoden der Umfrageforschung* (3., überarb. Aufl.). München: Oldenbourg.
- Ji, X., Wei, Z. und Feng, Y. (2006). Effective vehicle detection technique for traffic surveillance systems. *Journal of Visual Communication and Image Representation*, 17(3), S. 647–658.
- Jockisch, M. und Rosendahl, J. (2009). Klassifikation von Modellen. In: G. Bandow & H. H. Holzmüller (Hrsg.), „*Das ist gar kein Modell!*“: *Unterschiedliche Modelle und Modellierungen in Betriebswirtschaftslehre und Ingenieurwissenschaften* (S. 23–52). Wiesbaden: Gabler.
- Josefiok, M., Krahn, T. und Sauer, J. (2015). A Survey on Expert Systems for Diagnosis Support in the Field of Neurology. In: R. Neves-Silva, L. C. Jain, & R. J. Howlett (Hrsg.), *Intelligent Decision Technologies* (S. 291–300). Cham: Springer International Publishing.
- Keen, P. G. W. und Scott Morton, M. S. (1978). *Decision support systems: an organizational perspective*. Reading, Mass. [u.a.]: Addison-Wesley.

- Kennedy, J. und Eberhart, R. (1995). Particle Swarm Optimization. In: *Proceedings of IEEE International Conference on Neural Networks* (Bd. 4, S. 1942–1948). Perth, Australia: IEEE.
- Kephart, J. O. und Chess, D. M. (2003). The Vision of Autonomic Computing. *Computer*, 36(1), S. 41–50.
- Keymeulen, D., Iwata, M., Kuniyoshi, Y. und Higuchi, T. (1998). Online evolution for a self-adapting robotic navigation system using evolvable hardware. *Artificial Life*, 4(4), S. 359–393.
- Kirchhof, R. (2013). *Ganzheitliches Komplexitätsmanagement: Grundlagen und Methodik des Umgangs mit Komplexität im Unternehmen* (1. Aufl.). Wiesbaden: Springer.
- Klein, H. (1991). Anti-tachycardia stimulation in supraventricular and ventricular tachycardia with and without cardioverter defibrillator. *Herz*, 16(3), S. 182–198.
- Kohnke, O. (2015). B Theoretischer Teil. In: O. Kohnke (Hrsg.), *Anwenderakzeptanz unternehmensweiter Standardsoftware: Theorie, Einflussfaktoren und Handlungsempfehlungen* (S. 35–234). Wiesbaden: Springer.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9), S. 1464–1480.
- Kovacevic, D., Mladenovic, N., Petrovic, B. und Milosevic, P. (2014). DE-VNS: Self-adaptive Differential Evolution with crossover neighborhood search for continuous global optimization. *Computers & Operations Research*, 52, S. 157–169.
- Krallmann, H. und Rieger, B. (2001). Entscheidungsunterstützendes System (EUS). In: P. Mertens, A. Back, J. Becker, W. König, H. Krallmann, B. Rieger, A.-W. Scheer, D. Seibt, P. Stahlknecht, H. Strunz, R. Thome, & H. Wedekind (Hrsg.), *Lexikon der Wirtschaftsinformatik* (4., vollst. neu bearb. u. erw. Aufl., S. 149–150). Berlin [u.a.]: Springer.
- Kuckartz, U., Dresing, T., Rädiker, S. und Stefer, C. (2008). *Qualitative Evaluation: der Einstieg in die Praxis* (2., aktualisierte Auflage). Wiesbaden: VS für Sozialwissenschaften.
- Kuhlenkasper, T. und Handl, A. (2017). Lineare Regression. In: *Multivariate Analysemethoden* (S. 219–247). Berlin, Heidelberg: Springer Spektrum.

- Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 028(i05).
- Küsters, U. (2012). Evaluation, Kombination und Auswahl betriebswirtschaftlicher Prognoseverfahren. In: *Prognoserechnung* (S. 423–467). Physica, Heidelberg.
- Kuzel, A. J. (1992). Sampling in qualitative inquiry. In: *Doing qualitative research* (S. 31–44). Thousand Oaks, CA, US: Sage Publications, Inc.
- Laddaga, R. (1999). Guest Editor's Introduction: Creating Robust Software through Self-Adaptation. *IEEE Intelligent Systems*, 14(3), S. 26–29.
- Lalanda, P., McCann, J. A. und Diaconescu, A. (2013). *Autonomic Computing*. London: Springer.
- Laudon, K. C., Laudon, J. P. und Schoder, D. (2010). *Wirtschaftsinformatik: eine Einführung* (2., aktualisierte Aufl.). München [u.a.]: Pearson Studium.
- Lee, J. D. und Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-Computer Studies*, 40(1), S. 153–184.
- Lee, S., Kim, G. und Kim, S. (2011). Self-adaptive and dynamic clustering for online anomaly detection. *Expert Systems with Applications*, 38(12), S. 14891–14898.
- Leimeister, J. M. und Stahlknecht, P. (2015). *Einführung in die Wirtschaftsinformatik* (12., vollst. neu überarb. u. ak. Aufl.). Berlin, Heidelberg: Springer Gabler.
- Lemos, R. de, Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N. M., Vogel, T., Weyns, D., Baresi, L., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Desmarais, R., Dustdar, S., Engels, G., Geihs, K., Göschka, K. M., Gorla, A., Grassi, V., Inverardi, P., Karsai, G., Kramer, J., Lopes, A., Magee, J., Malek, S., Mankovskii, S., Mirandola, R., Mylopoulos, J., Nierstrasz, O., Pezzè, M., Prehofer, C., Schäfer, W., Schlichting, R., Smith, D. B., Sousa, J. P., Tahvildari, L., Wong, K. und Wuttke, J. (2013). Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. In: R. de Lemos, H. Giese, H. A. Müller, & M. Shaw (Hrsg.), *Software Engineering for Self-Adaptive Systems II* (S. 1–32). Berlin, Heidelberg: Springer.
- Liang, T. P. (1986). Critical Success Factors of Decision Support Systems: An Experimental Study. *ACM SIGMIS Database*, 17(2), S. 3–16.

- Liu, S., Duffy, A. H. B., Whitfield, R. I. und Boyle, I. M. (2009). Integration of decision support systems to improve decision support performance. *Knowledge and Information Systems*, 22(3), S. 261–286.
- Macías-Escrivá, F. D., Haber, R., del Toro, R. und Hernandez, V. (2013). Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Systems with Applications*, 40(18), S. 7267–7279.
- Makridakis, S. G. und Wheelwright, S. C. (1989). *Forecasting methods for management* (5. Aufl.). New York [u.a.]: Wiley.
- Massé, M. (2012). *REST API design rulebook*. Sebastopol: O'Reilly.
- Mayring, P. (2016). *Einführung in die qualitative Sozialforschung: eine Anleitung zu qualitativem Denken* (6., überarbeitete Auflage). Weinheim: Beltz.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. In: S. van der Walt & J. Millman (Hrsg.), *Proceedings of the 9th Python in Science Conference* (S. 51–56).
- McKinney, W., Perktold, J. und Seabold, S. (2011). Time series analysis in python with statsmodels. In: *Proceedings of the 10th Python in Science Conference - SCIPY 2011* (S. 96–102).
- McKnight, D. H. (2005). Trust in information technology. In: G. B. Davis (Hrsg.), *The Blackwell Encyclopedia of Management* (2. Aufl., Bd. 7, S. 329–331). Malden: Wiley-Blackwell.
- Meier, A. (2016). *SQL- & NoSQL-Datenbanken* (8., überarbeitete und erweiterte Auflage 2016). Berlin, Heidelberg: Springer Vieweg.
- Meinel, C. und Mundhenk, M. (2015). Graphen und Bäume. In: C. Meinel & M. Mundhenk (Hrsg.), *Mathematische Grundlagen der Informatik: Mathematisches Denken und Beweisen Eine Einführung* (S. 223–252). Wiesbaden: Springer Vieweg.
- Metzger, M. (2000). A comparative evaluation of DRE integration algorithms for real-time simulation of biologically activated sludge process. *Simulation Practice and Theory*, 7(7), S. 629–643.

- Meyyappan, L., Saygin, C. und Dagli, C. H. (2007). Real-time routing in flexible flow shops: a self-adaptive swarm-based control model. *International Journal of Production Research*, 45(21), S. 5157–5172.
- Meyyappan, L., Soylemezoglu, A., Saygin, C. und Dagli, C. H. (2008). A wasp-based control model for real-time routing of parts in a flexible manufacturing system. *International Journal of Computer Integrated Manufacturing*, 21(3), S. 259–268.
- Millman, K. J. und Aivazis, M. (2011). Python for Scientists and Engineers. *Computing in Science & Engineering*, 13(2), S. 9–12.
- Mittelbach, H. (2015). *Statistik*. Berlin, Boston: Oldenbourg Wissenschaftsverlag.
- Moser, C. (2012). Usability Testing. In: C. Moser (Hrsg.), *User Experience Design: Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern* (S. 219–242). Berlin, Heidelberg: Springer.
- Musliner, D. J., Goldman, R. P., Pelican, M. J. und Krebsbach, K. D. (1999). Self-Adaptive Software for Hard Real-Time Environments. *IEEE Intelligent Systems & Their Applications*, 14(4), S. 23.
- Nelli, F. (2018). Introduction to the Python World. In: F. Nelli (Hrsg.), *Python Data Analytics: With Pandas, NumPy, and Matplotlib* (S. 19–48). Berkeley, CA: Apress.
- Neri, F. und Tirronen, V. (2010). Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33(1–2), S. 61–106.
- Newell, A. und Simon, H. A. (1972). *Human problem solving* (2. print.). Englewood Cliffs, NJ: Prentice-Hall.
- Niknam, T., Azizipanah-Abarghooee, R. und Roosta, A. (2012). Reserve Constrained Dynamic Economic Dispatch: A New Fast Self-Adaptive Modified Firefly Algorithm. *IEEE Systems Journal*, 6(4), S. 635–646.
- Niknam, T., Narimani, M. R., Azizipanah-Abarghooee, R. und Bahmani-Firouzi, B. (2013). Multiobjective Optimal Reactive Power Dispatch and Voltage Control: A New Opposition-Based Self-Adaptive Modified Gravitational Search Algorithm. *IEEE Systems Journal*, 7(4), S. 742–753.

- Oden, J. T. und Prudhomme, S. (2011). Control of modeling error in calibration and validation processes for predictive stochastic models. *International Journal for Numerical Methods in Engineering*, 87(1–5), S. 262–272.
- Offermann, P., Blom, S., Schönherr, M. und Bub, U. (2010). Artifact Types in Information Systems Design Science – A Literature Review. In: R. Winter, J. L. Zhao, & S. Aier (Hrsg.), *Global Perspectives on Design Science Research* (S. 77–92). Berlin, Heidelberg: Springer.
- Oreizy, P., Gorlick, M. M., Taylor, R. N., Heimhigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D. S. und Wolf, A. L. (1999). An architecture-based approach to self-adaptive software. *IEEE Intelligent Systems and their Applications*, 14(3), S. 54–62.
- Pan, J., Wang, J., Cao, S. und Luo, X. (2011a). Interactive sign language synthesis based on adaptive display resolution visibility for ubiquitous devices. *Computer Animation and Virtual Worlds*, 22(2–3), S. 213–220.
- Pan, Q.-K., Suganthan, P. N., Wang, L., Gao, L. und Mallipeddi, R. (2011b). A differential evolution algorithm with self-adapting strategy and control parameters. *Computers & Operations Research*, 38(1), S. 394–408.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. und Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct), S. 2825–2830.
- Peffer, K., Rothenberger, M., Tuunanen, T. und Vaezi, R. (2012). Design Science Research Evaluation. In: K. Peffer, M. Rothenberger, & B. Kuechler (Hrsg.), *Design Science Research in Information Systems. Advances in Theory and Practice* (S. 398–410). Berlin, Heidelberg: Springer.
- Petris, G., Petrone, S. und Campagnoli, P. (2009). Dynamic linear models. In: P. Campagnoli, S. Petrone, & G. Petris (Hrsg.), *Dynamic Linear Models with R* (S. 31–84). New York: Springer.
- Pick, R. A. und Weatherholt, N. (2012). A Review On Evaluation And Benefits Of Decision Support Systems. *Review of Business Information Systems*, 17(1), S. 7–20.

- Pieroth, G. (2013). *Systematische Prognosefehler in der Unternehmensplanung: eine ökonomisch-psychologische Analyse*. Wiesbaden: Springer Gabler.
- Power, D. J. (2002). *Decision support systems: concepts and resources for managers*. Westport: Quorum Books.
- Power, D. J. (2013). *Decision support, analytics, and business intelligence* (2. ed). New York: Business Expert Press.
- Qin, A. K., Huang, V. L. und Suganthan, P. N. (2009). Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 13(2), S. 398–417.
- Quitadamo, R. und Zambonelli, F. (2008). Autonomic communication services: a new challenge for software agents. *Autonomous Agents and Multi-Agent Systems*, 17(3), S. 457–475.
- Ramirez-Quintana, J. A. und Chacon-Murguia, M. I. (2015). Self-adaptive SUM-CNN neural system for dynamic object detection in normal and complex scenarios. *Pattern Recognition*, 48(4), S. 1137–1149.
- Randolph, J. (2009). A Guide to Writing the Dissertation Literature Review. *Practical Assessment, Research, & Evaluation*, 14(13).
- Rao, R. V., Savsani, V. J. und Balic, J. (2012). Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. *Engineering Optimization*, 44(12), S. 1447–1462.
- Rao, R. V., Savsani, V. J. und Vakharia, D. P. (2011). Teaching-learning-based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems. *Comput. Aided Des.*, 43(3), S. 303–315.
- Rashedi, E., Nezamabadi-pour, H. und Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Inf. Sci.*, 179(13), S. 2232–2248.
- Rechenberg, I. und Eigen, M. (1976). Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. *ARSP: Archiv für Rechts- und Sozialphilosophie*, 62(2), S. 298–300.
- Rumelhart, D. E., Hinton, G. E. und Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), S. 533–536.

- Rupp, C. (2014). *Requirements-Engineering und -Management: aus der Praxis von klassisch bis agil* (6., aktualisierte und erweiterte Auflage). München: Hanser.
- Salehie, M. und Tahvildari, L. (2009). Self-adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems*, 4(2), S. 14:1–14:42.
- Sarodnick, F. und Brau, H. (2011). *Methoden der Usability Evaluation: wissenschaftliche Grundlagen und praktische Anwendung* (2., überarb. und aktualisierte Aufl). Bern: Huber.
- Sauer, S. (2019a). Datenjudo. In: S. Sauer (Hrsg.), *Moderne Datenanalyse mit R: Daten einlesen, aufbereiten, visualisieren, modellieren und kommunizieren* (S. 75–101). Wiesbaden: Springer Gabler.
- Sauer, S. (2019b). Hallo, R. In: S. Sauer (Hrsg.), *Moderne Datenanalyse mit R: Daten einlesen, aufbereiten, visualisieren, modellieren und kommunizieren* (S. 13–19). Wiesbaden: Springer Gabler.
- Saunders, M., Lewis, P. und Thornhill, A. (2012). *Research methods for business students* (6. Aufl.). Harlow [u.a.]: Pearson.
- Sauter, V. L. (2010). *Decision support systems for business intelligence* (2. Aufl.). New Jersey: Wiley.
- Sbarciog, M., Saraiva, I. und Vande Wouwer, A. (2013). Accelerating animal cell growth in perfusion mode by multivariable control: simulation studies. *Bioprocess and Biosystems Engineering*, 36(5), S. 517–530.
- Schmalen, H. und Pechtl, H. (2013). *Grundlagen und Probleme der Betriebswirtschaft* (15., überarb. und erw. Aufl). Stuttgart: Schäffer-Poeschel.
- Schnell, R. (Hrsg.). (2005). *Wahrnehmung - Kognition - Ästhetik: Neurobiologie und Medienwissenschaften* (1. Aufl.). Bielefeld: transcript Verlag.
- Schön, D. (2018). Fachliche inhaltliche Ausgestaltung. In: D. Schön (Hrsg.), *Planung und Reporting im BI-gestützten Controlling: Grundlagen, Business Intelligence, Mobile BI und Big-Data-Analytics* (S. 25–224). Wiesbaden: Springer Gabler.

- Schreider, S. Y., Young, P. C. und Jakeman, A. J. (2001). An application of the Kalman filtering technique for streamflow forecasting in the Upper Murray Basin. *Mathematical and Computer Modelling*, 33(6–7), S. 733–743.
- Schryen, G. (2015). Writing Qualitative IS Literature Reviews - Guidelines for Synthesis, Interpretation, and Guidance of Research. *Communications of the Association for Information Systems*, 37(1).
- Scott, S. L. und Varian, H. R. (2014). Predicting the present with Bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1–2), S. 4–23.
- Shapiro, S. S. und Wilk, M. B. (1965). An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, 52(3/4), S. 591–611.
- Siebertz, K., Bebber, D. van und Hochkirchen, T. (2017). Korrelationsanalyse. In: K. Siebertz, D. van Bebber, & T. Hochkirchen (Hrsg.), *Statistische Versuchsplanung: Design of Experiments (DoE)* (S. 381–394). Berlin, Heidelberg: Springer.
- Simon, H. A. (1977). *The new science of management decision*. Englewood Cliffs, N.J: Prentice-Hall.
- Sonnenberg, C. und vom Brocke, J. (2012). Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In: K. Peffers, M. Rothenberger, & B. Kuechler (Hrsg.), *Design Science Research in Information Systems. Advances in Theory and Practice. DESRIST 2012* (S. 381–397). Berlin, Heidelberg: Springer.
- Sprague, R. H. und Carlson, E. D. (1982). *Building effective decision support systems*. New Jersey: Prentice-Hall.
- Srinivasa, K. G., Sharath, S., Venugopal, K. R. und Patnaik, L. A. (2005). Selective dissemination of XML documents using GAs and SVM. In: *Computational Intelligence and Security, Pt 1, Proceedings* (Bd. 3801, S. 562–567).
- Stelling, J. N. (2008). *Kostenmanagement und Controlling*. München: Oldenbourg.
- Sterman, J. D. (2000). *Business dynamics: systems thinking and modeling for a complex world*. Boston, Mass.: Irwin/McGraw-Hill.

- Sterritt, R. (2005). Autonomic computing. *Innovations in Systems and Software Engineering, 1*(1), S. 79–88.
- Stockmann, R. und Meyer, W. (2010). *Evaluation: eine Einführung*. Opladen [u.a.]: Budrich.
- Stölzle, W. (2007). Planung. In: H.-U. Küpper, A. Pfingsten, & R. Köhler (Hrsg.), *Handwörterbuch der Betriebswirtschaft* (Bd. 6., vollst. neu gestaltete Aufl., S. 1382–1393). Stuttgart: Schäfer-Poeschel.
- Storn, R. und Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization, 11*(4), S. 341–359.
- Strutz, T. (2016). *Data Fitting and Uncertainty: A practical introduction to weighted least squares and beyond* (2. Aufl.). Wiesbaden: Springer Vieweg.
- Taylor, S. J. und Letham, B. (2018). Forecasting at Scale. *The American Statistician, 72*(1), S. 37–45.
- Teo, J. (2006). Exploring dynamic self-adaptive populations in differential evolution. *Journal Soft Computing, 10*(8), S. 673–686.
- The Neo4j Graph Platform. (2018). Abgerufen am 8. August 2018 von <https://neo4j.com/product/>.
- Treyer, O. A. G. (2010). *Business Forecasting: Anwendungsorientierte Theorie quantitativer Prognoseverfahren* (1. Aufl.). Bern [u.a.]: Haupt.
- Turban, E., Sharda, R. und Delen, D. (2011). *Decision support and business intelligence systems* (9. Aufl.). Upper Saddle River, NJ [u.a.]: Pearson.
- Turban, E., Sharda, R. und Delen, D. (2014). *Decision support and business intelligence systems* (9. Aufl.). Harlow: Pearson.
- Urban, D. und Mayerl, J. (2018). Das Grundmodell der Regressionsanalyse. In: *Angewandte Regressionsanalyse: Theorie, Technik und Praxis* (S. 19–106). Wiesbaden: Springer VS.
- van der Walt, S., Colbert, S. C. und Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering, 13*(2), S. 22–30.

- Van Dyke Parunak, H. (1997). „Go to the ant“: Engineering principles from natural multi-agent systems. *Annals of Operations Research*, 75(0), S. 69–101.
- Venkatesh, V. und Davis, F. D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46(2), S. 186–204.
- Vensim Help. (2018). [Dokumentation]. Abgerufen am 19. Februar 2019 von https://www.vensim.com/documentation/index.html?fn_ramp.htm.
- Vensim Software. (2018). [Software Description]. Abgerufen am 8. August 2018 von <https://vensim.com/vensim-software/>.
- Vogel, J. (2015). *Prognose von Zeitreihen: Eine Einführung für Wirtschaftswissenschaftler*. Wiesbaden: Springer Gabler.
- Walker, W. E., Harremoes, P., Rotmans, J., Sluijs, J. P. van der, Asselt, M. B. A. van, Jansen, P. und Krauss, M. P. K. von. (2003). Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support. *Integrated Assessment*, 4(1), S. 5–17.
- Wang, C.-M. und Huang, Y.-F. (2010). Self-adaptive harmony search algorithm for optimization. *Expert Systems with Applications*, 37(4), S. 2826–2837.
- Wang, C.-W., Hunter, A., Gravill, N. und Matusiewicz, S. (2014). Unconstrained Video Monitoring of Breathing Behavior and Application to Diagnosis of Sleep Apnea. *IEEE Transactions on Biomedical Engineering*, 61(2), S. 396–404.
- Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B. und Tian, Q. (2011). Self-adaptive learning based particle swarm optimization. *Information Sciences*, 181(20), S. 4515–4538.
- Weber, J. (2005). *Alle entscheidenden Konzepte, Steuerungssysteme und Instrumente* (1. Aufl.). Weinheim: Wiley-VCH-Verl.
- Weicker, K. (2007). *Evolutionäre Algorithmen* (2., überarb. und erw. Aufl.). Wiesbaden: Vieweg+Teubner.
- Weicker, K. (2015). *Evolutionäre Algorithmen* (3., überarb. und erw. Aufl.). Wiesbaden: Springer Vieweg.
- White, N. (2016, April 5). Neo4j Driver for R. Abgerufen am 13. Februar 2019 von <https://www.rdocumentation.org/packages/RNeo4j/versions/1.6.4>.

- Wild, J. (1980). Einführung. In: J. Wild (Hrsg.), *Grundlagen der Unternehmensplanung* (S. 12–31). Wiesbaden: VS Verlag für Sozialwissenschaften.
- Wong, W. K., Guo, Z. X. und Leung, S. Y. S. (2014). Intelligent multi-objective decision-making model with RFID technology for production planning. *International Journal of Production Economics*, 147, S. 647–658.
- Xu, L. und Jordan, M. I. (1996). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8(1), S. 129–151.
- Xue, D., Demkowicz, L. und Hao, L. (2007). A 3-D hp finite/infinite element method to calculate power deposition in the human head. *IEEE Transactions on Biomedical Engineering*, 54(4), S. 734–741.
- Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. In: O. Watanabe & T. Zeugmann (Hrsg.), *Stochastic Algorithms: Foundations and Applications* (S. 169–178). Berlin, Heidelberg: Springer.
- Zeileis, A. und Grothendieck, G. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software*, 14(6), S. 1–27.
- Zhao, S.-Z., Suganthan, P. N. und Das, S. (2011). Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. *Soft Computing*, 15(11), S. 2175–2185.
- Zhao, Y., Zhou, W. L., Huang, J. M., Yu, S. und Lanham, E. J. (2003). Self-adaptive clock synchronization for computational grid. *Journal of Computer Science and Technology*, 18(4), S. 434–441.
- Zhong, Y. und Zhang, L. (2012). Remote Sensing Image Subpixel Mapping Based on Adaptive Differential Evolution. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 42(5), S. 1306–1329.
- Zhu, J., Yan, X. und Zhao, W. (2013). Chemical process dynamic optimization based on the differential evolution algorithm with an adaptive scheduling mutation strategy. *Engineering Optimization*, 45(10), S. 1205–1221.