



**Kompetenzorientiertes E-Assessment
für die grafische Modellierung
in der Hochschullehre**

Meike Ullrich

Kompetenzorientiertes E-Assessment für die grafische Modellierung in der Hochschullehre

von Meike Ullrich

Dissertation, Karlsruher Institut für Technologie
Fakultät für Wirtschaftswissenschaften, 2024

Tag der mündlichen Prüfung: 10. Juni 2024
Referent: Prof. Dr. Andreas Oberweis
Korreferent: Prof. Dr. Hartmut Schmeck

Impressum

Autorin: Meike Ullrich
Covergrafik: Gabriela Kapfer - <https://www.smileinitial.plus>
Stand: 17. Juni 2024



Dieses Werk – mit Ausnahme der Covergrafik – ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.de>



Die Covergrafik ist lizenziert unter einer Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz (CC BY-NC-ND 4.0 DE): <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de>

DOI: 10.5445/IR/1000171737

Kompetenzorientiertes E-Assessment für die grafische Modellierung in der Hochschullehre

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
(Dr.-Ing.)

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte
DISSERTATION

von
Meike Ullrich

Datum der mündlichen Prüfung: 10. Juni 2024
Referent: Prof. Dr. Andreas Oberweis
Korreferent: Prof. Dr. Hartmut Schmeck

Danksagung

Meine Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Angewandte Informatik (AIFB) des Karlsruher Instituts für Technologie (KIT) hat mir einen breit gefächerten Einstieg in die spannende Welt der wissenschaftlichen Forschung und der Hochschullehre ermöglicht.

Ich danke in erster Linie meinem Doktorvater Prof. Dr. Andreas Oberweis für seine kontinuierliche Unterstützung und die Gelegenheit, die beiden Bereiche Forschung und Lehre in meinem Promotionsvorhaben miteinander zu kombinieren. Ebenso bedanke ich mich bei Prof. Dr. Hartmut Schmeck für die Übernahme des Korreferats und die konstruktiven Anmerkungen zum Manuskript, sowie bei Prof. Dr. Martin Klarmann und Prof. Dr. Marc Wouters für die Vervollständigung des Prüfungskollegiums in meiner mündlichen Promotionsprüfung.

Mein Dank richtet sich ebenfalls an die zahlreichen Personen, die meinen Forschungsweg begleitet und stets an mich geglaubt haben. Ich danke allen aktuellen und ehemaligen Kolleginnen und Kollegen aus der Forschungsgruppe Betriebliche Informationssysteme für einen herausragenden Zusammenhalt im Arbeitsalltag. Namentlich möchte ich Martin Forell, Gunther Schiefer und Selina Schüler nennen, die mich stets zuverlässig und engagiert im Verbundprojekt KEA-Mod unterstützt haben. Überdies gilt mein Dank allen weiteren Personen aus dem Verbundprojekt für die produktive und inspirierende Zusammenarbeit, besonders Chantal Soyka von der Universität Paderborn und Michael Striewe von der Universität Duisburg-Essen. Vor allem meine »Doktorgeschwister« Stella Möhrle und Andreas Schoknecht haben mir in den letzten Zügen meiner Forschungsarbeit viel Kraft und Motivation gegeben – ich danke euch herzlich dafür!

Zu guter Letzt möchte ich meiner Familie meinen tiefsten Dank aussprechen. Ich danke meinen Eltern, die meinen langen Bildungsweg vorbehaltlos unterstützt haben. Von Herzen danke ich meinem Mann Patrick und meinen Kindern Lina und Tom für ihre uneingeschränkte Zuversicht. Ohne ihre Liebe, ihre Geduld und ihr Verständnis wäre die Fertigstellung dieser Dissertation nicht möglich gewesen.

Zusammenfassung

In vielen Anwendungsbereichen der Informatik spielt die grafische Modellierung eine wichtige Rolle. Grafische Modelle kommen beispielsweise bei der Geschäftsprozessmodellierung oder im Rahmen der Softwareentwicklung zum Einsatz, um komplexe Sachverhalte übersichtlich darzustellen. In der Hochschullehre kommt derzeit eine kompetenzorientierte Ausrichtung entsprechender Lehrveranstaltungen zu kurz, ebenso sind die Möglichkeiten zur technischen Unterstützung eingeschränkt. Die in dieser Arbeit behandelten Forschungsfragen sind daher einer kompetenzorientierten Ausrichtung des Prüfens auf dem Gebiet der grafischen Modellierung sowie der Entwicklung einer entsprechenden E-Assessment-Plattform gewidmet.

Im Rahmen der Arbeit wurde anhand theoriebasierter und empirischer Ansätze ein umfassendes Kompetenzmodell entwickelt, das Lernziele für zentrale Handlungsbereiche der grafischen Modellierung und überfachliche Kompetenzen beschreibt. Es wurde ein Aufgabenkatalog erstellt, der Aufgabentypen mit den im Kompetenzmodell definierten Lernzielen verknüpft. Ergänzend wurden exemplarische Bewertungsschemata und Empfehlungen für die Gestaltung lernförderlicher Feedback-Nachrichten auf Basis des Kompetenzmodells abgeleitet. Die Ergebnisse unterstützen Lehrende bei der Auswahl von Lernzielen und der Gestaltung kompetenzorientierter Prüfungen anhand passender Modellierungsaufgaben.

Zur Umsetzung kompetenzorientierter Prüfungen auf dem Gebiet der grafischen Modellierung wurde eine E-Assessment-Plattform entwickelt. Diese berücksichtigt verschiedene grafische Modellierungssprachen, individuelle Bewertungsschemata und Feedback-Empfehlungen. Zusätzlich wurden Dienste zur automatisierten Bewertung von Petri-Netzen erstellt, die Lernziele zu syntaktischen, semantischen und pragmatischen Qualitätsaspekten adressieren. Die Einsatzfähigkeit der Plattform wurde im praktischen Einsatz in Lehrveranstaltungen und Prüfungen demonstriert. Ergänzend wurden Befragungen zur Benutzungsfreundlichkeit und weiteren Aspekten durchgeführt und die Ergebnisse der Anwendung der Bewertungsdienste auf einer umfangreichen Datenbasis studentischer Petri-Netze evaluiert.

Inhaltsverzeichnis

Zusammenfassung	i
Abbildungsverzeichnis	vii
Tabellenverzeichnis	xi
Symbolverzeichnis	xii
I. Hintergrund	1
1. Einleitung	3
1.1. Hintergrund und Motivation	3
1.2. Forschungsfragen und Zielsetzung	7
1.3. Forschungsansatz	8
1.4. Projekt KEA-Mod und Folgeprojekte	14
1.5. Aufbau der Arbeit	16
2. Grundlagen	19
2.1. Hochschuldidaktik	19
2.1.1. Didaktikbegriff und Constructive Alignment	20
2.1.2. Lernziele und Kompetenzbegriff	22
2.1.3. E-Assessment und Aufgabenklassen	24
2.1.4. Bewertung und Feedback	29
2.2. Grafische Modellierung	35
2.2.1. Geschäftsprozessmodellierung	36
2.2.2. Modellbegriff	38
2.2.3. Modellierungssprachen	45
2.2.4. Petri-Netze	59

3. Verwandte Arbeiten	71
3.1. Modellierung in der Hochschullehre	72
3.1.1. Fachdidaktik und Modellierungskompetenzen	73
3.1.2. Aufgabentypen	79
3.2. Modellqualität	87
3.2.1. Qualitätsframeworks	88
3.2.2. Modellierungsrichtlinien	94
3.2.3. Qualitätsmetriken	97
3.3. Automatisierte Bewertung graphbasierter Diagramme	103
3.3.1. Allgemeine Ansätze	104
3.3.2. Ansätze aus dem Kontext Hochschullehre	113
II. Bearbeitung der Forschungsfragen	129
4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung	131
4.1. Kompetenzmodell	132
4.2. Aufgabenkatalog	136
4.3. Rubrics und Feedback	141
4.4. Diskussion	147
4.4.1. Kompetenzmodell	148
4.4.2. Aufgabenkatalog	150
4.4.3. Rubrics und Feedback	151
5. E-Assessment für das Gebiet der grafischen Modellierung	153
5.1. KEA-Mod-Plattform	153
5.1.1. Architektur	154
5.1.2. Datenmodell	158
5.1.3. Technische Realisierung	160
5.1.4. Modellierungseditor und Austauschformat	162
5.2. Prozess der automatisierten Bewertung	165
5.2.1. Bewertungsschema	166
5.2.2. Datentransferobjekte	169
5.2.3. Bewertungsinitialisierer	171
5.2.4. Bewertungsverarbeiter	172
5.2.5. Beginn der automatisierten Bewertung	173
5.3. Bewertungsdienste	174
5.3.1. Grundgerüst	176

5.3.2. Syntax-Checker	182
5.3.3. Semantik-Checker	188
5.3.4. Pragmatik-Checker	197
5.3.5. Erreichbarkeits-Checker	204
5.4. Anwendungsbeispiel	206
5.5. Evaluation	215
5.5.1. Datenbasis	216
5.5.2. Ergebnisse unter Anwendung eines Bewertungsschemas . . .	218
5.5.3. Analyse des Matching-Verfahrens im Semantik-Checker . . .	225
5.5.4. Vergleich manuelle Korrektur und Semantik-Checker	232
5.5.5. Dauer der automatisierten Bewertung	237
5.5.6. Plattform-Evaluation im Übungseinsatz	238
5.5.7. Plattform-Evaluation im Prüfungseinsatz	242
5.6. Diskussion	246
5.6.1. Modellierungsaufgaben und Modelle der Evaluation	246
5.6.2. Vergleich Bewertungsschema und Rubric	248
5.6.3. Ergebnisse der Anwendung des Bewertungsschemas	251
5.6.4. Zulassen syntaktischer Mängel im Modellierungseditor . . .	253
5.6.5. Erkenntnisse aus dem Einsatz des Semantik-Checkers	254
5.6.6. Einsatz von KI-Ansätzen im Bewertungskontext	255
5.6.7. Technische Aspekte zur Plattform	257
5.6.8. Vergleich der Plattform mit verwandten Werkzeugen	260
5.6.9. Plattform-Evaluation im Übungs- und Prüfungseinsatz	263
III. Schluss	267
6. Zusammenfassung und Ausblick	269
6.1. FF1: Hochschuldidaktische Grundlagen	269
6.2. FF2: E-Assessment-Plattform	270
6.3. Ausblick	272
Anhang	277
A. Ergänzung zur Literaturstudie	279
B. Ergänzung zur Evaluation	283
Literaturverzeichnis	289

Abbildungsverzeichnis

1.1. Übersicht über die Phasen des Forschungsprozesses	8
1.2. Übersicht über den Aufbau der Arbeit	17
2.1. Constructive Alignment	21
2.2. Lernzieltaxonomie von Anderson und Krathwohl (2001)	22
2.3. Mögliche Zeitpunkte für ein Assessment	25
2.4. Feedback-Modell	34
2.5. Einordnung konzeptueller Modelle im ARIS-Konzept	39
2.6. Beziehungen zwischen Modell und Modelldenotation	42
2.7. Übersicht über die Modellierungsschritte nach Def. 2.12	43
2.8. Semiotischer Tetraeder nach FRISCO	44
2.9. Begriffliche Einordnung in Bezug auf Diagramme und Graphen	45
2.10. Bestandteile einer Modellierungssprache	48
2.11. Beispiel ER-Diagramm	50
2.12. Übersicht über die UML-Diagrammtypen	52
2.13. Beispiel für ein UML-Klassendiagramm	52
2.14. Beispiel für ein UML-Aktivitätsdiagramm	55
2.15. Beispiel für eine Ereignisgesteuerte Prozesskette	58
2.16. Beispiel für ein BPMN-Modell	59
2.17. Grafische Darstellung des Netzes N	62
2.18. Beispiel für ein S/T-Netz	64
2.19. Beispiel für ein Petri-Netz-Geschäftsprozessmodell	68
3.1. Concept-Map zum Kurs »Business Process Modelling«	73
3.2. Beispiel zum Aufgabentyp »Modell erläutern«	83
3.3. Beispiel zum Aufgabentyp »Fehler im Modell finden«	84
3.4. Beispiel zum Aufgabentyp »Modell erstellen«	85
3.5. Beispiel zum Aufgabentyp »Modell analysieren«	86
3.6. Beispiel zum Aufgabentyp »Modell verändern«	87
3.7. Ausschnitt aus dem SEQUAL-Framework	89

Abbildungsverzeichnis

3.8. Wechselwirkungen zwischen semiotischen Qualitätsaspekten	91
3.9. Übersicht zu den Grundsätzen ordnungsmäßiger Modellierung	94
3.10. Aufbau eines Qualitätsframeworks für die Modellierung	98
3.11. Geschäftsprozessgraph zu einem gegebenen Petri-Netz N	99
3.12. Überblick Vorgehensweise Literaturstudie	115
3.13. Verteilung der Publikationen aus der Ergebnismenge über Jahre	119
4.1. Entwicklungsprozess des Kompetenzmodells	132
4.2. Struktur Kompetenzmodell für die grafische Modellierung	134
4.3. Aufgabentypen für die grafische Modellierung	138
4.4. Auszug Laut-Denk-Protokoll in Software MAXQDA	139
4.5. Beispiele für Feed-Up-Nachrichten	146
4.6. Beispiele für Feedback-Nachrichten	147
4.7. Beispiele für Feed-Forward-Nachrichten	148
5.1. Architektur der KEA-Mod-Plattform	155
5.2. Datenmodell der KEA-Mod-Plattform	159
5.3. Syntaktisch falsches Petri-Netz im Modellierungseditor	163
5.4. Konzeption Bewertungsschemata	167
5.5. Datentransferobjekte in der KEA-Mod-Plattform	169
5.6. Ein- und Ausgabe-DTO	170
5.7. Datenstruktur eines graphbasierten Diagramms	177
5.8. Überblick über die Arbeitsschritte des Semantik-Checker	190
5.9. Anmeldung an der KEA-Mod-Plattform	207
5.10. Aufgabenerstellung in der Lehrenden UI	208
5.11. Aufgabenreihen in der Studierenden UI	210
5.12. Modellierungseditor in der KEA-Mod-Plattform	211
5.13. Feedback-Anzeige für den Aufgabentyp »Modell erstellen«	212
5.14. Feedback-Anzeige für den Aufgabentyp »Modell verstehen«	213
5.15. Manuelle Revision in der Lehrenden UI	215
5.16. Aufgabe A1 mit Musterlösung	217
5.17. Ausschnitt aus der Musterlösung zu Aufgabe A2	228
5.18. Ausschnitt aus Modell ex3_21_110.kea	229
5.19. Ausschnitt aus Modell ex1_19_22.kea	231
5.20. Ausschnitt aus Modell ex1_20_149.kea	231
5.21. Ausschnitt aus Modell ex1_18_1.kea	233
5.22. Ausschnitt aus Modell ex1_16_73.kea	235
5.23. Ausschnitt aus Modell ex1_17_62.kea	236

5.24. Automatisierte Bewertung in der KEA-Mod-Plattform	237
5.25. Prüfungseinsatz der KEA-Mod-Plattform im Hörsaal	243
5.26. Bewertungsschema für Aufgaben A1-A3 in Baumstruktur	258
5.27. Bewertungsschema im ProFormA-Format	259
B.1. Aufgabe A2 mit Musterlösung	283
B.2. Aufgabe A3 mit Musterlösung	284

Tabellenverzeichnis

2.1. Digitalisierungs- bzw. Automatisierungsgrad beim Assessment	26
2.2. Beispiel eines Rubrics für den Einsatz im Deutschunterricht	32
2.3. Einsatzgebiete der Geschäftsprozessmodellierung	37
2.4. Zentrale Modellierungselemente von ER-Diagrammen	51
2.5. Zentrale Modellierungselemente von UML-Klassendiagrammen	53
2.6. Zentrale Modellierungselemente von UML-Aktivitätsdiagrammen	54
2.7. Zentrale Modellierungselemente von (e)EPK	57
2.8. Zentrale Modellierungselemente von BPMN-Modellen	60
2.9. Zentrale Modellierungselemente von Petri-Netzen	68
2.10. Ausgewählte Kontrollflussmuster der Workflow Pattern Initiative	70
3.1. Lernziele für die Modellierung nach Glinz	76
3.2. Kompetenzen zur Modellierung nach den GI-Empfehlungen	77
3.3. Lernziele für die Modellerstellung nach Bogdanova (2019)	79
3.4. Lernziele für die konzeptuelle Modellierung nach Bork (2019)	80
3.5. Aufgabentypen in Lehrbüchern zur Geschäftsprozessmodellierung	82
3.6. Quellen bezüglich der Analyse von Beschriftungen in Modellen	105
3.7. Datenextraktionsformular für die Literaturstudie	117
3.8. Outlets der Publikationen in der Literaturstudie	119
3.9. Modelltypen in den Publikationen der Literaturstudie	120
3.10. Aktueller Stand und offene Punkte gemäß der Literaturstudie	127
4.1. Kompetenzfacetten für den Kompetenzbereich MB 4	136
4.2. Kompetenzfacetten für den Aufgabentyp »Modell erstellen«	141
4.3. Optionale Kompetenzfacetten für einen Aufgabentyp	142
4.4. Rubric für den Aufgabentyp »Modell erstellen«	144
5.1. Bewertungsdienste in der KEA-Mod-Plattform	175
5.2. Feedback-Nachrichten in den Bewertungsdiensten	182
5.3. Prüfmethode syntaktischer Vorgaben für Petri-Netze	187

Tabellenverzeichnis

5.4. Feedback-Nachrichten des Semantik-Checkers	197
5.5. Übersicht über die Methoden des Pragmatik-Checkers	199
5.6. Feedback-Nachrichten des Erreichbarkeits-Checkers	205
5.7. Überblick über die Datenbasis studentischer Modelle	216
5.8. In der Evaluation verwendetes Bewertungsschema	219
5.9. Übersicht über die erzielten Punkte aller Modelle der Datenbasis . .	221
5.10. Ergebnisse der Check-Methoden im Bewertungsschema	222
5.11. Ergebnisse der Metrik-Methoden im Bewertungsschema	224
5.12. Übersicht Precision, Recall und F-Score auf der Stichprobe	227
5.13. Vorgaben für die manuelle Korrektur zum Aspekt Semantik	234
5.14. Gegenüberstellung manuelle Korrektur und Semantik-Checker	235
5.15. Ergebnisse der Fragebogenauswertung im Übungseinsatz	240
5.16. Ergebnisse der Fragebogenauswertung im Prüfungseinsatz	244
5.17. Komplexität der Musterlösungen zu den Aufgaben der Datenbasis .	248
5.18. Punkteverteilung anhand oberer Grenze und Fehlerzahl	252
A.1. In der Literaturstudie berücksichtigte Publikationen	279
B.1. Einzelergebnisse zu Precision, Recall und F-Score	284
B.2. Einzelergebnisse Vergleich manuelle Korrektur/Semantik-Checker .	286

Symbolverzeichnis

$G'(g)$	Gewichtete Summe aller einzelnen Methodenergebnisse
$G(g)$	Gewichtete Summe kombiniert mit Gruppenbildung
$R_e(g)$	Normiertes Ergebnis der Anwendung einer Methode auf einem Graphen g im Intervall $[0, 1]$
$R_a(g)$	Anwendbarkeit einer Methode auf einem Graphen g als boolescher Wert
$R_b(g)$	Direktes Methodenergebnis (Check) auf einem Graphen g als boolescher Wert
$R_d(g)$	Direktes Methodenergebnis (Metrik) auf einem Graphen g im Intervall $[0, 1]$
$R_f(g)$	Menge der Elemente im Graph g , welche das geprüfte Kriterium verletzen
$R_o(g)$	Bezugsgröße im Graphen g für die Normierung eines Methodenergebnisses
C_o	Maximal zu berücksichtigende Fehler bei der Normierung eines Methodenergebnisses ¹
C_w	Gewicht eines Methodenergebnisses
C_s	Menge von Schwellenwerten zur Umwandlung eines Methodenergebnisses
C_e	Menge von Ergebniswerten zu den Schwellenwerten C_s
C_{ok}	Maximal zu berücksichtigende Fehler bei der Normierung gruppierter Methodenergebnisse
C_t	Schwellenwert zur Erstellung positiver bzw. negativer Feedback-Nachrichten
C_n	Aktivierung der bedingten Nullifikation des Gesamtergebnisses
C_{ns}	Mindestgröße des Modells für die Bedingung zur Nullifikation
C_{max}	Maximale Anzahl für m im $n : m$ -Matching
C_{nf}	Abzug für fehlende Knoten
C_{ne}	Abzug für zusätzliche Knoten
C_{kf}	Abzug für fehlende Kanten
C_{ke}	Abzug für zusätzliche Kanten
C_f	Anzahl zu erstellender Feedback-Nachrichten im formativen Einsatz

1 Bei allen mit C beginnenden Symbolen handelt es sich um Konfigurationsparameter.

Teil I

Hintergrund

1. Einleitung

Es werden zunächst der Hintergrund und die Motivation für das in dieser Arbeit beschriebene Forschungsvorhaben vorgestellt (siehe Abschnitt 1.1). Zusätzlich sollen die Forschungsfragen und die Zielsetzung der Arbeit definiert werden (siehe Abschnitt 1.2), um daran anschließend den eigenen Forschungsansatz zu skizzieren (siehe Abschnitt 1.3). Auf die Rolle verschiedener Projekte im Kontext des Forschungsvorhabens wird in Abschnitt 1.4 eingegangen. Der Aufbau der Arbeit wird in Abschnitt 1.5 überblicksartig dargestellt.

1.1. Hintergrund und Motivation

Immer mehr Hochschulabsolventen strömen auf den Arbeitsmarkt. Nach Angaben des Statistischen Bundesamts hat sich die Zahl der bestandenen Hochschulprüfungen in den vergangenen 20 Jahren mehr als verdoppelt. Im Jahr 2022 wurden insgesamt rund 506.000 Studienabschlüsse an tertiären Bildungseinrichtungen gezählt [St23a]. Doch wie steht es um den Ausbildungserfolg? Besitzen Hochschulabsolventen das erforderliche Fachwissen und die gewünschten Kompetenzen? Die Frage nach der Qualität der Hochschulbildung gewinnt nicht zuletzt durch den 1999 angestoßenen Bologna-Prozess zunehmend an Bedeutung. Im Zentrum der Verwirklichung eines europäischen Hochschulraumes steht die Entwicklung vergleichbarer und kompatibler Hochschulabschlüsse, bei welchen »Qualifikationen im Hinblick auf Arbeitsbelastung, Niveau, Lernergebnisse und Kompetenzen« definiert werden [Eu03].

Die verstärkte Kompetenzorientierung nach Bologna steht im Einklang mit den Forderungen aus der Hochschuldidaktik und Wissenschaftsforschung, welche in den letzten Jahrzehnten einen Perspektivenwechsel von einer dozierenden- zu einer studierendenzentrierten Lehre hin propagieren. Der oft als »Shift from Teaching to Learning« bezeichnete Wandel rückt das aktive Lernen in den Mittelpunkt [Be98; BT95]. Dabei tritt die bloße Darstellung und Vermittlung von Lehrinhalten (Input- bzw. Content-Orientierung) in den Hintergrund. Einen hohen Stellenwert besitzen nun die

1. Einleitung

Ergebnisse des Lernens (Outcome-Orientierung) und damit einhergehend passende Strategien bzw. Methoden, mit denen diese erreicht werden können [Wi07].

Eng damit verbunden ist die Gestaltung und Formulierung von Lernzielen. Bereits in den 50er-Jahren entwickelte BLOOM [Bl73] eine hierarchisch aufgebaute Taxonomie, in welcher Lernziele in sechs verschiedene Niveaustufen mit zunehmender Komplexität eingeteilt werden. In der kompetenzorientierten Lehre werden nicht nur Lernziele der unteren Niveaustufen (Wissen und Verstehen) sondern auch Lernziele der höheren Niveaustufen (Anwenden, Analyse, Synthese und Bewertung) adressiert. Das Fachgutachten zur Kompetenzorientierung in Studium und Lehre von SCHAPER et al. [Sc12a, S. 32f] legt jedoch nahe, dass »zentrale Konzepte zur Gestaltung einer kompetenz- und outcome-orientierten Lehre in deutschen Hochschulen noch nicht ausreichend verbreitet sind und noch nicht in hinreichend sichtbarer Form umgesetzt werden«. In den vergangenen Jahren wurden zahlreiche Förderprogramme zur Verbesserung der Lehre initiiert, die sich den damit verbundenen Herausforderungen für die Hochschulen stellen. Beispielsweise hat der Bund zwischen 2011 und 2020 mit dem »Qualitätspakt Lehre« des Bundesministeriums für Bildung und Forschung (BMBF) rund zwei Milliarden Euro zur Verfügung gestellt, um die Betreuung der Studierenden und die Lehrqualität an Hochschulen zu verbessern. Seit dem Jahr 2016 hat das BMBF bislang vier Förderbekanntmachungen zum Themenschwerpunkt »Digitale Hochschulbildung« veröffentlicht [Bu23]. Seit 2021 fördert die Stiftung Innovation in der Hochschullehre geeignete Projekte, die Hochschulen darin unterstützen sollen, sich »auf neue gesellschaftliche Herausforderungen und Bedarfe einzustellen« [St23b].

Vor diesem Hintergrund entstehen neue Aufgabenbereiche für das akademische Lehrpersonal. Diese reichen von der Formulierung angemessener Lernziele und deren didaktischer Umsetzung in einer Veranstaltung bis hin zur Gestaltung passender kompetenzorientierter Prüfungsformate. Gleichzeitig entsteht dabei auch ein immenser Aufwand, nicht nur initial bei der Entwicklung und Einführung neuer Konzepte, sondern fortwährend bei der Korrektur von Prüfungsleistungen wie z.B. Klausuren, mit denen die Erreichung von Lernzielen höherer Niveaustufen überprüft werden soll. Denn hier kommen dementsprechend komplexere Aufgabentypen zum Einsatz, welche nicht zwingend eine eindeutige Lösung besitzen. Somit ist der triviale Abgleich mit einer Musterlösung nicht möglich und die Korrektur mit einem hohen Zeitaufwand und Maß an Konzentration verbunden. Dadurch wird die Erfüllung grundlegender Anforderungen an die Bewertung wie die konsistente und fehlerfreie Anwendung eines vorgegebenen Bewertungsschemas gefährdet, insbesondere bei

mehreren korrigierenden Personen.

Um den erhöhten Korrekturaufwand trotz steigender Studierendenzahlen zu bewältigen, bietet sich die Unterstützung durch digitale Werkzeuge an. Derzeit ist der Einsatz digitaler Prüfungen, auch als elektronische Assessments (E-Assessments) bezeichnet, an deutschen Hochschulen auf dem Vormarsch [Ha16]. Die Möglichkeiten zur automatisierten Bewertung in den gängigen Lehr-Lernsystemen wie z. B. ILIAS oder Moodle decken jedoch vorwiegend lediglich untere Niveaustufen ab (z.B. anhand von Antwort-Wahl-Aufgaben). Gemäß der Zielsetzung, eine verstärkte Kompetenzorientierung durch Berücksichtigung aller Niveaustufen zu erreichen, wird daher daran geforscht, mit E-Assessments künftig auch Prüfungsformate abbilden zu können, die höhere Kompetenzniveaus adressieren. Mit Hilfe eines solchen E-Assessment-Werkzeugs würde die angestrebte kompetenzorientierte Bewertung studentischer Leistungen in greifbare Nähe rücken. Der Wunsch nach dieser Form von Unterstützung durch Maschinen ist nicht neu. So beschreibt PRESSEY [Pr26, S. 374] im Jahr 1926 das Ziel sogenannter »Teaching Machines« wie folgt: »Lift from her [the teacher's] shoulders as much as possible of this burden and make her free for those inspirational and thought-stimulating activities which are, presumably, the real function of the teacher.« Für Lehrende hätte eine derartige Anwendung zur automatisierten Bewertung neben der Zeit- und Kostenreduktion weitere Vorteile: Die digitale Erfassung von studentischen Lösungen eröffnet die Möglichkeit der statistischen Auswertung und damit verbunden z.B. die Identifikation häufig auftretender Fehler. Dies könnte konstruktiv genutzt werden, um Rückschlüsse auf die eigene Lehre zu ziehen und diese gegebenenfalls anzupassen.

In diesem Kontext soll die hier vorgestellte Arbeit einen Beitrag für das Gebiet der grafischen Modellierung leisten. Dabei liegt der Anwendungsbereich der Geschäftsprozessmodellierung im Fokus. Die Geschäftsprozessmodellierung gehört in der Praxis neben dem Datenbankentwurf zu den häufigsten Einsatzgebieten für die konzeptuelle Modellierung [Da06] und ist curriculärer Bestandteil zahlreicher Studiengänge, insbesondere in den Disziplinen Informatik und Wirtschaftsinformatik (WI), sowie der im internationalen Raum vorherrschenden Schwesterdisziplin Information Systems (IS). Mit Hilfe der Geschäftsprozessmodellierung kann durch den Einsatz einer informalen, semi-formalen oder formalen grafischen Darstellung eine Beschreibung betrieblicher Abläufe in Form eines Geschäftsprozessmodells erstellt werden. Dafür kommen sogenannte grafische Modellierungssprachen wie z. B. die Business Process Model and Notation (BPMN), Ereignisgesteuerte Prozessketten (EPK) oder Petri-Netze zum Einsatz. Im Unternehmenskontext werden

1. Einleitung

Geschäftsprozessmodelle unter anderem neben Dokumentations- und Entwurfszwecken auch als Grundlage für die Analyse und anschließende Reorganisation der betrieblichen Abläufe eingesetzt [Ob96]. Aufgaben zur Modellbildung, d.h. zur Fähigkeit, ein passendes Modell zu einem definierten Kontext zu erstellen [Gl08], adressieren überwiegend Lernziele höherer Niveaustufen. Von Studierenden erstellte Geschäftsprozessmodelle sind jedoch oft unübersichtlich und komplex. Da keine eindeutige Musterlösung vorhanden ist und verschiedene Bewertungskriterien betrachtet werden müssen, ist die Korrektur und Bewertung von studentischen Geschäftsprozessmodellen in Klausuren für Lehrende mühsam. Insbesondere in Veranstaltungen mit hohen Teilnehmerzahlen ist daher zusätzlich in Vorlesungen und Übungen keine individuelle Rückmeldung zur Lösung von Modellierungsaufgaben möglich.

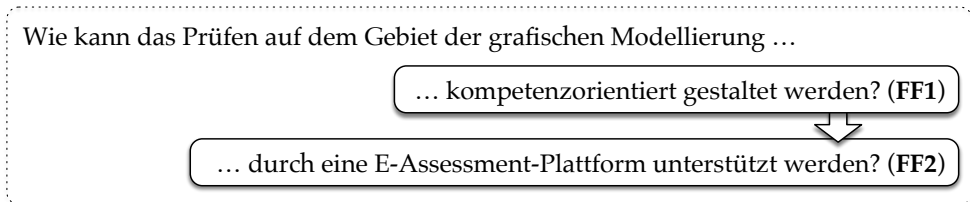
Ähnliches gilt für andere Anwendungsbereiche der grafischen Modellierung in der Informatik, wie z. B. der Softwareentwicklung oder Datenmodellierung. Dort werden beispielsweise die Modellierungssprachen Unified Model Notation (UML) oder Entity-Relationship-Model (ERM) eingesetzt. Diese Sprachen, sowie die zuvor genannten Sprachen für die Geschäftsprozessmodellierung (BPMN, EPK und Petri-Netze) haben (mindestens) zwei gemeinsame Merkmale. Erstens werden grafische Symbole verwendet, um eine zweidimensionale visuelle Darstellung zu erzeugen [Mo09]. Derartige Modelle werden häufig als Diagramm bezeichnet [LS87]. Zweitens sind diese Diagramme graphbasiert. Das bedeutet, dass sie eine Graphstruktur [Bi93] aufweisen, in der Modellelemente als Knoten dargestellt werden, die über bestimmte Kanten miteinander verbunden sind. Dadurch können z. B. Beziehungen zwischen Objekten oder der Kontrollfluss in Geschäftsprozessmodellen abgebildet werden. Der Begriff »grafische Modellierung« in dieser Arbeit bezeichnet die Modellierung graphbasierter Diagramme in den genannten Anwendungsbereichen der Informatik.

Zu E-Assessment-Werkzeugen für die Unterstützung der Analyse und Bewertung von studentischen Geschäftsprozessmodellen sind bislang nur vereinzelte Ansätze bekannt (z. B. [Th16]). Dahingegen existieren mehrere Ansätze mit dem Fokus auf graphbasierte Diagramme aus anderen Anwendungsbereichen der Informatik (z. B. [MS15a; SL15; TWS05]). All diese Ansätze kommen jedoch über ein experimentelles Stadium nicht hinaus, d.h. sie werden, wenn überhaupt, nur in Lehrveranstaltungen der jeweiligen Forschenden praktisch eingesetzt. Ebenso existieren zahlreiche wissenschaftliche Arbeiten, die sich mit Fragen zur Qualität von graphbasierten Diagrammen befassen (z. B. [KLS95; Me08a; OBS12]). Diese stehen allerdings beinahe

ausnahmslos im Zusammenhang mit dem Praxiseinsatz und berücksichtigen nicht den speziellen Kontext Hochschullehre.

1.2. Forschungsfragen und Zielsetzung

Zur Bearbeitung der zuvor beschriebenen Forschungslücke werden für das Forschungsvorhaben dieser Arbeit die beiden aufeinander aufbauenden Forschungsfragen FF1 und FF2 abgeleitet:



Das Gebiet der grafischen Modellierung umfasst neben der Geschäftsprozessmodellierung auch die Anwendungsbereiche Softwareentwicklung und Datenmodellierung. Es werden Modellierungssprachen betrachtet, mit deren Hilfe graphbasierte Diagramme erstellt werden können. Im Rahmen dieser Dissertation wird der Fokus mitunter auf die Geschäftsprozessmodellierung und insbesondere Petri-Netze gerichtet.

Bezüglich der Forschungsfrage FF1 besteht das Ziel in der Erschaffung mehrerer Artefakte. Zunächst soll ein **Kompetenzmodell** mit Lernzielen und erwünschten Kompetenzen (Kompetenzfacetten) für das Gebiet der grafischen Modellierung entwickelt werden. Daran anschließend soll ein **Aufgabenkatalog** erstellt werden, der Aufgabentypen zur Prüfung der Kompetenzfacetten aus dem Kompetenzmodell beschreibt. Zusätzlich sollen kriteriengeleitete und kompetenzorientierte **Bewertungsschemata (Rubrics)** für eine Reihe von Aufgabentypen aus dem Aufgabenkatalog sowie Empfehlungen für lernförderliche **Feedback-Nachrichten** vorgestellt werden.

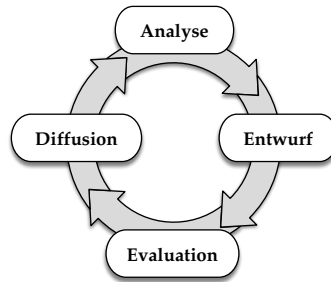
Auf der Grundlage der Ergebnisse aus FF1 sollen im Rahmen von Forschungsfrage FF2 weitere Artefakte entstehen. So soll eine **E-Assessment-Plattform** entwickelt werden, die sich in Lehrveranstaltungen auf dem Gebiet der grafischen Modellierung einsetzen lässt. Sie soll individuelle Anforderungen (z. B. Einsatzzeitpunkt oder Bewertungsschema) berücksichtigen. Mithilfe der E-Assessment-Plattform sollen

1. Einleitung

- Identifikation der Forschungslücke
- Definition der Forschungsfragen und -ziele
- In weiteren Iterationsschritten: Berücksichtigung der Erkenntnisse aus Evaluations- und Diffusionsphase

Verbreitung des Ansatzes über:

- 13 Publikationen
- 5 Konferenzworkshops
- 3 Forschungsprojekte



- Literaturrecherche und Inhaltsanalysen zur Fachdidaktik
- Entwicklung Kompetenzmodell, Aufgabenkatalog und Rubrics
- Anforderungserhebung und Literaturstudie zur automatisierten Bewertung graphbasierter Diagramme im Hochschulkontext
- Entwicklung E-Assessment-Plattform und Bewertungsdienste für Petri-Netze

- Expertenrating und Validierung des Kompetenzmodells
- Logische und kognitive Aufgabenanalyse von Modellierungsaufgaben
- Umfragen beim Einsatz der E-Assessment-Plattform in Lehrveranstaltungen und Prüfungen
- Anwendung der Bewertungsdienste auf einer Datenbasis von 2194 Petri-Netzen

Abb. 1.1.: Übersicht über die Phasen und Hauptaktivitäten des Forschungsprozesses

Aufgabentypen aus dem Aufgabenkatalog umgesetzt sowie die kompetenzorientierte und automatisierte Bewertung von graphbasierten Diagrammen ermöglicht werden. Weiterhin sollen **Bewertungsdienste für die Prüfung von Petri-Netzen** entwickelt werden. Diese sollen anhand eines Bewertungsschemas diejenigen Kompetenzfacetten adressieren, die dem Aufgabentyp »Modell erstellen« zugeordnet wurden. Dabei sollen Grenzen und Herausforderungen ausgelotet werden, insbesondere, wenn die automatisierte Bewertung mit der manuellen Bewertung durch eine korrigierende Person verglichen wird.

1.3. Forschungsansatz

Das Forschungsvorhaben dieser Dissertation ist eng am gestaltungsorientierten Forschungsansatz von ÖSTERLE et al. [Ös10] sowie an den von RECKER [Re13] beschriebenen Forschungsprinzipien ausgerichtet. Abb. 1.1 stellt eine grobe Übersicht über den Forschungsansatz anhand der Phasen und Hauptaktivitäten des iterativ durchgeführten Forschungsprozesses vor.

Die Identifikation der Forschungslücke sowie die Definition der Forschungsfragen und -ziele (siehe Abschnitte 1.1 und 1.2) in der Analysephase ist durch eine intensive Auseinandersetzung mit der wissenschaftlichen Literatur erfolgt (siehe Kapitel 2 und

3). In weiteren Iterationsschritten wurden in der Analysephase auch die Ergebnisse aus einer vorangegangenen Evaluations- oder Diffusionsphase ausgewertet, um z. B. Erkenntnisse aus dem Praxiseinsatz oder Umfragen sowie Rückmeldungen aus der Fachcommunity bei der Weiterentwicklung der Artefakte in der darauffolgenden Entwurfsphase zu berücksichtigen.

Das Kompetenzmodell als Artefakt zur Forschungsfrage FF1 ist in einem vierstufigen Entwicklungsprozess entstanden (siehe Abschnitt 4.1), was vier Iterationen des Forschungsprozesses entspricht. Gemäß der Empfehlungen von SCHAPER [Sc09] wurde dabei ein theoriebasierter Ansatz mit empirischen Ansätzen kombiniert. Um einen ersten Entwurf des Kompetenzmodells zu erstellen, wurde eine Literaturrecherche bezüglich Fachliteratur sowie eine Inhaltsanalyse von Modulbeschreibungen zur normativen Ausrichtung durchgeführt (theoriebasierter Ansatz). Die sukzessive Weiterentwicklung und Validierung des Kompetenzmodells sowie die Erstellung des Aufgabenkatalogs (siehe Abschnitt 4.2) ist anhand eines Expertenratings (Fragebogenstudie), einer logischen Aufgabenanalyse (Literaturrecherche und Inhaltsanalyse nach MAYRING [Ma19]) und einer kognitiven Aufgabenanalyse (Laut-Denk-Studie mit mehreren Probanden) erfolgt (empirische Ansätze). Die Entwicklung der weiteren Artefakte zu Forschungsfrage FF1 (Rubrics und Feedback-Empfehlungen) basiert größtenteils auf dem Kompetenzmodell sowie dem Aufgabenkatalog und wurde durch zusätzliche Abstimmungen mit Hochschullehrenden aus dem Gebiet der grafischen Modellierung unterstützt (siehe Abschnitt 4.3).

Der Entwurf der E-Assessment-Plattform als Artefakt zur Forschungsfrage FF2 (siehe Abschnitte 5.1, 5.2 und 5.3) stützt sich neben der Berücksichtigung der Artefakte aus FF1 auf eine Anforderungserhebung in Form von funktionalen und nicht-funktionalen Anforderungen. Zusätzlich wurde in der Entwurfsphase eine systematische Literaturstudie nach vom BROCKE et al. [BST15] durchgeführt, um einen Überblick über den Stand der Forschung zu bestehenden Ansätzen zur automatisierten Bewertung von graphbasierten Diagrammen im Hochschulkontext zu erhalten (siehe Abschnitt 3.3.2). Dadurch konnte beim Entwurf der Plattform und der Bewertungsdienste sowohl auf bewährte Ansätze zurückgegriffen als auch identifizierte Lücken gezielt adressiert werden. In der Evaluationsphase wurde die Plattform in Lehrveranstaltungen verschiedener Hochschulen praktisch eingesetzt und die Einsatzfähigkeit demonstriert. Ergänzend wurden empirische Untersuchungen in Form von Fragebogenstudien zur Benutzungsfreundlichkeit und weiterer Aspekte mit den teilnehmenden Personen durchgeführt (siehe Abschnitte 5.5.6 und 5.5.7). Weiterhin wurden die entwickelten Bewertungsdienste auf einer umfangreichen

1. Einleitung

Datenbasis mit 2.194 Petri-Netzen im Hinblick auf die Zielsetzung evaluiert (siehe Abschnitte 5.5.2, 5.5.3, 5.5.4).

Die Diffusion bzw. Verbreitung der Forschungsergebnisse ist während der Bearbeitung des Forschungsvorhabens in verschiedener Form erfolgt. Die folgenden Listen stellen eine chronologische Übersicht über 13 Publikationen und die Organisation von fünf Workshops im Kontext des Forschungsthemas dar.

Publikationen:

- *(Business Process) Models from an Educational Perspective*: **Meike Ullrich** und Andreas Schoknecht, 8th Central European Workshop on Services and their Composition (ZEUS), CEUR Workshop Proceedings, Wien, Januar 2016. Quelle: [US16]
- *Evaluation automatisierter Ansätze für die Bewertung von Modellierungsaufgaben*: Michael Fellmann, Peter Fettke, Constantin Houy, Peter Loos, Andreas Oberweis, Andreas Schoknecht, Michael Striewe, Tom Thaler und **Meike Ullrich**, 14. e-Learning Fachtagung Informatik (DeLFI), Potsdam, September 2016. Quelle: [Fe16]
- *Bewertung grafischer Prozessmodelle mit Methoden des Maschinellen Lernens - Ein Experiment*: Matthias Binder, Andreas Oberweis und **Meike Ullrich**, E-Prüfungs-Symposium (ePS), Siegen, September 2019. Quelle: [BOU19]
- *Was sollen Studierende modellieren? Erweiterung klassischer Prozessmodellierungsaufgaben um den Aspekt der Prozesserhebung*, **Meike Ullrich**, Salome E. Franzen und Andreas Oberweis, 2. Workshop zur Modellierung in der Hochschullehre (MoHoL) im Rahmen der Modellierung 2020, Wien, Februar 2020. Quelle: [UFO20]
- *Towards an Automated Assessment of Graphical (Business Process) Modelling Competences*: Michael Striewe, Constantin Houy, Jana-Rebecca Rehse, **Meike Ullrich**, Peter Fettke, Niclas Schaper und Andreas Oberweis, Workshop zum Stand, den Herausforderungen und Impulsen des Geschäftsprozessmanagements (ZuGPM) im Rahmen der GI-Jahrestagung INFORMATIK 2020, Karlsruhe, September 2020. Quelle: [St20]
- *Platform Architecture for the Diagram Assessment Domain*: **Meike Ullrich**, Martin Forell, Constantin Houy, Peter Pfeiffer, Selina Schüler, Tobias Stottrop, Brian Willems, Peter Fettke und Andreas Oberweis, Workshop Software Engineering for E-Learning Systems (SEELS) im Rahmen der International Conference on Software Engineering (SE), Braunschweig, Februar 2021. Quelle: [UI21]

- *Kompetenzorientiertes E-Assessment für die grafische, konzeptuelle Modellierung*: Michael Striewe, Martin Forell, Constantin Houy, Peter Pfeiffer, Gunther Schiefer, Selina Schüler, Chantal Soyka, Tobias Stottrop, **Meike Ullrich**, Peter Fettke, Peter Loos, Andreas Oberweis und Niclas Schaper, HMD Wirtschaftsinformatik, Ausgabe 58, S. 1350-1363, September 2021. Quelle: [St21]
- *E-Assessment-Plattform für die grafische Modellierung*: **Meike Ullrich**, Peter Pfeiffer, Gunther Schiefer, Chantal Soyka, Tobias Stottrop, Michael Striewe, Peter Fettke, Peter Loos, Andreas Oberweis und Niclas Schaper, 20. Fachtagung Bildungstechnologien (DeLFI), Karlsruhe, September 2022. Quelle: [UI22b]
- *Piloteinsatz einer E-Assessment-Plattform für die grafische Modellierung*: **Meike Ullrich**, Peter Pfeiffer, Gunther Schiefer, Chantal Soyka, Tobias Stottrop, Michael Striewe, Peter Fettke, Peter Loos, Andreas Oberweis und Niclas Schaper, 20. Fachtagung Bildungstechnologien (DeLFI), Karlsruhe, September 2022. Quelle: [UI22c]
- *Towards a Competence Model for Graphical Modeling*: Chantal Soyka, Niclas Schaper, Elena Bender, Michael Striewe und **Meike Ullrich**, ACM Transactions on Computing Education, Dezember 2022. Quelle: [So22f]
- *E-Assessment in Conceptual Modeling - A Systematic Literature Review*: **Meike Ullrich**, Constantin Houy, Tobias Stottrop, Michael Striewe, Brian Willems, Peter Fettke, Peter Loos und Andreas Oberweis, Special Issue on Teaching and Learning in Conceptual Modeling, Enterprise Modeling and Information Systems Architectures (EMISA) - International Journal of Conceptual Modeling, Februar 2023. Quelle: [UI23a]
- *Kompetenzorientiertes Prüfen in der Hochschullehre im Fachgebiet der grafischen Modellierung (KEA-Mod)*: **Meike Ullrich**, Selina Schüler, Martin Forell und Gunther Schiefer, Digitale Prüfungen - flexibel, kompetenzorientiert und gerecht: Erfahrungen und Erkenntnisse aus dem Projekt PePP: Partnerschaft für innovative E-Prüfungen. Projektverbund der baden-württembergischen Universitäten, S. 31–34, August 2023. Quelle: [UI23b]
- *Comparison of Required Competences and Exercise Material in Conceptual Modeling*: Chantal Soyka, Michael Striewe, **Meike Ullrich** und Niclas Schaper, Special Issue on Teaching and Learning in Conceptual Modeling, Enterprise Modeling and Information Systems Architectures (EMISA) - International Journal of Conceptual Modeling, September 2023. Quelle: [So23b]

1. Einleitung

Workshop-Organisation:

- *Workshop zur Modellierung in der Hochschullehre (MoHoL 2016)*¹: Michael Fellmann, Andreas Schoknecht und **Meike Ullrich**, Im Rahmen der Tagung Modellierung 2016, Karlsruhe, März 2016. Quelle: [FSU16]
- *2. Workshop zur Modellierung in der Hochschullehre (MoHoL 2020)*²: Jana-Rebecca Rehse, Michael Striewe und **Meike Ullrich**, Im Rahmen der Tagung Modellierung 2020, Wien, Februar 2020. Quelle: [RSU20]
- *Workshop Modellierungslehre digitalisieren*: Pauline Junginger, Peter Pfeiffer, Gunther Schiefer, Selina Schüler, Michael Striewe und **Meike Ullrich**, Im Rahmen der GI-Jahrestagung INFORMATIK 2021, virtuelle Durchführung, September 2021.
- *3. Workshop zur Modellierung in der Hochschullehre (MoHoL 2022)*³: **Meike Ullrich**, Peter Fettke, Peter Pfeiffer, Selina Schüler, Michael Striewe, Im Rahmen der Tagung Modellierung 2022, Hamburg, Juni 2022. Quelle: [UI22a]
- *Transferworkshop Kompetenzorientiertes E-Assessment für die grafische Modellierung*: Eric Blum, Peter Pfeiffer, Niclas Schaper, Selina Schüler, Chantal Soyka, Michael Striewe und **Meike Ullrich**, Im Rahmen der GI-Jahrestagung INFORMATIK 2022, Hamburg, September 2022.

Dabei konnten Rückmeldungen aus der Fachcommunity sowohl über die Peer-Review-Verfahren der Publikationen als auch über die Diskussionen auf den organisierten Workshops gesammelt und bei der Entwicklung der didaktischen (ad FF1) und technischen Artefakte (ad FF2) eingebracht werden.

Im Zusammenhang mit dem Forschungsvorhaben wurden verschiedene Teilziele und verwandte Aspekte auch im Rahmen betreuter Abschlussarbeiten aufgegriffen und vertieft. Sofern Ergebnisse aus diesen Abschlussarbeiten direkt in die Dissertation eingeflossen sind, wird dies an der entsprechenden Stelle explizit vermerkt.

- Irina Maryash, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Entwicklung automatisierter Ansätze für die Überprüfung der Qualität von Prozessmodellen*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2014.
- Christian Emmerich, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *eAssessment mit Unterstützung eines Computeralgebrasystems für die Mathematik*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2014.

1 <https://butler.aifb.kit.edu/MoHoL/>

2 <https://butler.aifb.kit.edu/web/mohol2020/>

3 <https://butler.aifb.kit.edu/MoHoL2022/>

- Katharina Vogel, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Entwicklung eines Bewertungsmodells für die Qualitätsbeurteilung von Prozessmodellen*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2015.
- Tobias Scherer, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *E-Assessments im Bereich der Geschäftsprozessmodellierung*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2015.
- Friedrich Volz, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Bestimmung der semantischen Qualität von Geschäftsprozessmodellen*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2016.
- Lotta Güttler, Referent: Andreas Oberweis, Betreuer: Andreas Schoknecht, Meike Ullrich: *Process Model Similarity in der Lehre*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2016.
- Christian Weise, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Konzeption und Entwicklung einer Softwarekomponente zur Unterstützung der Qualitätsbewertung von Petri-Netz Modellen*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2016.
- Matthias Binder, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Automatisierte Bewertung von Prozessmodellen mit Methoden des Maschinellen Lernens*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2018 und Publikation (Quellenangabe: [BOU19]).
- Breno Aberle, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Entwurf und Implementierung eines Petri-Netz Editors als Microservice für den Einsatz in der Hochschullehre*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2019.
- Salome Elena Franzen, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Multimediaeinsatz bei Übungsaufgaben zum Thema Geschäftsprozessmodellierung*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2019 und Publikation (Quellenangabe: [UFO20]).
- Elisa Polifke, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Entwicklung einer automatisierten Bewertung pragmatischer Qualitätsaspekte von Geschäftsprozessmodellen*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2020.

1. Einleitung

- Christian Bitterwolf, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Semantische Analyse von Petri-Netzen mit Methoden des Natural Language Processing*, Fakultät für Informatik, Karlsruher Institut für Technologie, Masterarbeit, 2021.
- David Brüning, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Analyse von graph-basierten Diagrammen mit Methoden des Natural Language Processing*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2022.
- Fabian Aisenbrey, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Generic framework for assessment services in e-assessment of graphical modeling*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2023.

1.4. Projekt KEA-Mod und Folgeprojekte

Über die zuvor aufgeführten Publikationen und Workshops hinaus ist aus dem Forschungsvorhaben zusätzlich ein Projektantrag für ein BMBF-Verbundprojekt hervorgegangen, welcher im Rahmen der dritten Förderlinie des BMBF zum Förderschwerpunkt »Digitale Hochschulbildung« eingereicht und bewilligt wurde. Im Fokus dieser Förderlinie stand die disziplin- und fachbezogene digitale Hochschulbildung [Bu18]. Es sollten digitale Lehr- und Lernkonzepte – sogenannte digitale Fachkonzepte – innerhalb einzelner Disziplinen und Fächer anwendungsnahe entwickelt, erprobt und erforscht werden, um spezifische Lehr- und Prüfungsformate innerhalb der Fächerkulturen weiterzuentwickeln. Zusätzlich sollte die nachhaltige Verankerung digitaler Fachkonzepte durch eine standortübergreifende Transferkooperation – unter anderem durch den Einbezug von Fachgesellschaften – ermöglicht werden.

Das Verbundprojekt »Kompetenzorientiertes E-Assessment für die grafische Modellierung« (KEA-Mod)⁴ wurde von November 2019 bis Ende Dezember 2022 über eine Laufzeit von drei Jahren und zwei Monaten gefördert. Es greift die beiden Forschungsfragen FF1 und FF2 aus der vorliegenden Dissertation auf, um diese in einem interdisziplinären Team bearbeiten zu können. Gleichzeitig konnte dabei der Fokus auf die grafische Modellierung im Allgemeinen sowie zusätzliche graph-basierte Modellierungssprachen ausgeweitet werden. Darüber hinaus wurde auch der Transfer zum Einsatz der im Projekt entstandenen E-Assessment-Plattform an

4 <https://keamod.gi.de>

weiteren Hochschulen und Universitäten innerhalb des Fachgebiets der grafischen Modellierung angestoßen.

Unter der Leitung des Instituts für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) am KIT mit Fokus auf die technische Entwicklung der E-Assessment-Plattform und die Modellierungssprachen Petri-Netze und Business Process Model and Notation (BPMN) waren zudem folgende Verbundpartner in weiteren Rollen am Projekt beteiligt:

- Institut für Humanwissenschaften, Arbeits- und Organisationspsychologie, Universität Paderborn (UPB): Hochschuldidaktische Fragestellungen zum Prüfen auf dem Gebiet der grafischen Modellierung und Begleitung der Evaluationsmaßnahmen der E-Assessment-Plattform
- Deutsches Forschungszentrum für künstliche Intelligenz, Saarbücken (DFKI): Technische Entwicklung der E-Assessment-Plattform und Ereignisgesteuerte Prozessketten (EPK)
- paluno - The Ruhr Institute for Software Technology, Universität Duisburg-Essen (UDE): Technische Entwicklung der E-Assessment-Plattform und verschiedene UML-Diagrammtypen
- Gesellschaft für Informatik e.V., Berlin (GI): Unterstützung bei der Wissenschaftskommunikation und dem Transfer des digitalen Fachkonzeptes zu anderen Hochschulstandorten

Im Rahmen der Förderlinie »Hochschullehre durch Digitalisierung stärken« der Stiftung Innovation in der Hochschullehre fließen die Ergebnisse aus dem KEA-Mod-Projekt seit Mitte 2021 in zwei weitere Folgeprojekte ein:

- **Partnerschaft für innovative E-Prüfungen (PePP)**⁵ - Verbundprojekt der baden-württembergischen Universitäten (darunter KIT) mit dem Ziel, den Einsatz elektronischer Prüfungen voranzutreiben und deren Potenzial zu erschließen.
- **Data Teaching Plug-in (DaTa-Pin)**⁶ - Einzelprojekt der Universität des Saarlandes zur Sicherung der Fortschritte im Bereich der Digitalisierung in der Lehre durch die Entwicklung übertragbarer Best Practice Konzepte.

⁵ <https://stiftung-hochschullehre.de/projekt/pepp/>

⁶ <https://stiftung-hochschullehre.de/projekt/data-pin/>

1. Einleitung

In beiden Folgeprojekten erfolgt die Weiterentwicklung und Erprobung der KEA-Mod-Plattform im Hinblick auf den summativen Einsatz in elektronischen Prüfungen.

Die Verfasserin dieser Dissertation hat neben der Gesamtprojektleitung des KEA-Mod-Verbundprojekts auch die Leitung bei der Softwareentwicklung der E-Assessment-Plattform übernommen. Insbesondere das Grundkonzept für die Abwicklung der automatisierten Bewertung aus ihrer Dissertation wurde in das Projekt eingebracht. Die Bearbeitung von Forschungsfrage FF1 wurde in interdisziplinärer Zusammenarbeit mit dem Verbundpartner UPB aus dem Gebiet der Hochschuldidaktik bearbeitet. Zudem wurden die im Forschungsvorhaben der Dissertation in Bezug auf Forschungsfrage FF2 entwickelten Bewertungsdienste für Petri-Netze in die KEA-Mod-Plattform eingebunden und auf dieser Basis evaluiert. Für die weitere Ausrichtung im Hinblick auf den Einsatz und die Evaluation der KEA-Mod-Plattform im Prüfungskontext wirkt die Verfasserin dieser Dissertation auch im Folgeprojekt PePP am KIT mit.

Wenn in der vorliegenden Dissertation Ergebnisse beschrieben werden, die nicht von der Verfasserin allein sondern in Zusammenarbeit mit weiteren Projektbeteiligten entwickelt wurden, wird dies stets entsprechend angemerkt. Einen Überblick hierzu bietet dabei auch Abb 1.2 im folgenden Abschnitt.

1.5. Aufbau der Arbeit

Der Aufbau dieser Arbeit orientiert sich im Wesentlichen am gewählten Forschungsansatz (siehe Abschnitt 1.3). Abb. 1.2 bietet eine Übersicht über die wesentlichen Bestandteile der Arbeit.

Teil I (Hintergrund) enthält neben dem aktuellen Kapitel 1 (Einleitung) zwei weitere Kapitel. In Kapitel 2 werden zunächst die Begriffe und Gegenstände aus den beiden Kernbereichen der Arbeit (Hochschuldidaktik und grafische Modellierung) eingeführt, die für das Verständnis der Arbeit notwendig sind. Die Auseinandersetzung mit dem Stand bestehender und verwandter Problemlösungsansätze erfolgt in Kapitel 3. Dieses bietet einen Überblick über Arbeiten aus der Literatur, die mit dem Forschungsvorhaben näher verwandt sind. Dies sind zunächst Arbeiten, die sich grundsätzlich mit dem Thema Modellierung in der Hochschullehre befassen (siehe Abschnitt 3.1) und Arbeiten zum Thema Modellqualität (siehe Abschnitt 3.2).

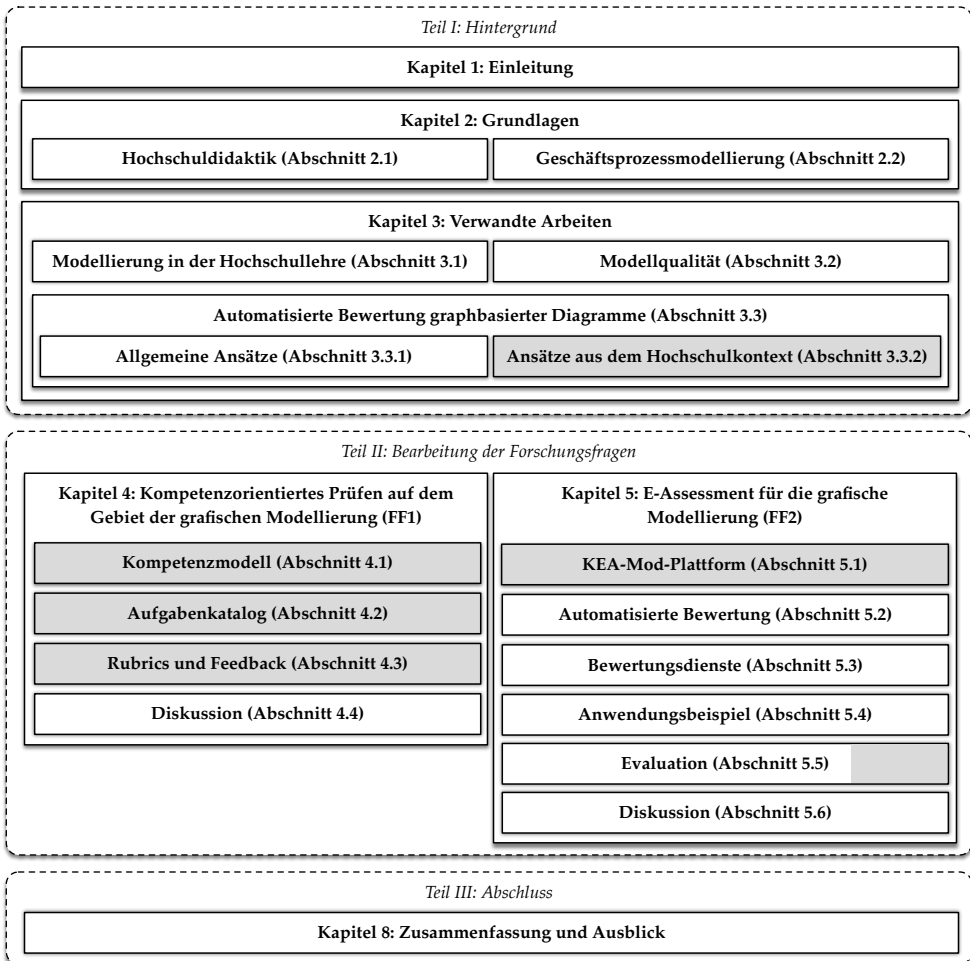


Abb. 1.2.: Übersicht über den Aufbau der Arbeit. Hervorgehoben sind Elemente, die ganz (grau) oder teilweise (teils grau) in Zusammenarbeit mit Beteiligten im Verbundprojekt KEA-Mod entstanden sind.

Anschließend werden Arbeiten betrachtet, die sich mit der automatisierten Bewertung von graphbasierten Diagrammen befassen. Dabei wird zwischen allgemeinen Ansätzen zu verschiedenen Modellaspekten (siehe Abschnitt 3.3.1) und zwischen Ansätzen aus dem Kontext Hochschullehre (siehe Abschnitt 3.3.2) unterschieden. In den Kapiteln 2 und 3 wird gleichzeitig eine weitere Verortung und Abgrenzung des Forschungsvorhabens vorgenommen.

In Teil II wird die Bearbeitung der beiden zentralen Forschungsfragen der Arbeit vor-

1. Einleitung

gestellt, wobei Kapitel 4 der Forschungsfrage FF1 und Kapitel 5 der Forschungsfrage FF2 gewidmet ist. Zur Forschungsfrage FF1 werden die entwickelten didaktischen Artefakte vorgestellt: das Kompetenzmodell (siehe Abschnitt 4.1), der Aufgabenkatalog (siehe Abschnitt 4.2) sowie die Rubrics samt Feedback-Empfehlungen (siehe Abschnitt 4.3). Die Beschreibung dieser Artefakte beinhaltet dabei sowohl den Entwurf als auch die durchgeführten Evaluations- bzw. Validierungsmaßnahmen. Anschließend erfolgt die Vorstellung der technischen Artefakte zur Forschungsfrage FF2: der KEA-Mod-Plattform (siehe Abschnitt 5.1) mit einem Fokus auf die Komponenten der automatisierten Bewertung (siehe Abschnitt 5.2) und den Bewertungsdiensten für Petri-Netze (siehe Abschnitt 5.3). Dabei werden zunächst Entwurfs- und Implementierungsaspekte in den Fokus gerückt und durch ein Anwendungsbeispiel (siehe Abschnitt 5.4) illustriert. Anschließend erfolgt die Vorstellung der Evaluationsmaßnahmen und deren Ergebnisse (siehe Abschnitt 5.5). Beide Forschungsfragen werden jeweils in einem dedizierten Abschnitt zur Diskussion kritisch reflektiert (ad FF1 siehe Abschnitt 4.4, ad FF2 siehe Abschnitt 5.6).

Die Arbeit wird in Teil III mit einer Zusammenfassung und einem Ausblick (Kapitel 6) sowie einem Anhang abgeschlossen. Dieser enthält ergänzende Angaben zur Literaturstudie (siehe Abschnitt 3.3.2) und zur Evaluation der Bewertungsdienste (siehe Abschnitte 5.5.1, 5.5.3 und 5.5.4).

2. Grundlagen

In diesem Kapitel werden bestehende wissenschaftliche Grundlagenbegriffe eingeführt. Dies dient in erster Linie dazu, im weiteren Verlauf der Arbeit auf eine möglichst einheitliche Terminologie zurückzugreifen, zugleich werden hierbei aber auch der Fokus und die Grenzen der Arbeit aufgezeigt und verdeutlicht. Auf der obersten Ebene ist das Kapitel in die zwei großen Themenbereiche des Forschungsvorhabens gegliedert. Der erste Abschnitt 2.1 befasst sich mit dem Kontext Hochschullehre und erläutert die aus dem Gebiet der Hochschuldidaktik entlehnten Begriffe und Konzepte in den Abschnitten 2.1.1 - 2.1.4. Die sogenannte grafische Modellierung ist Gegenstand des zweiten Abschnitts 2.2. Hier erfolgt zunächst eine Vorstellung des Anwendungsbereichs Geschäftsprozessmodellierung (siehe Abschnitt 2.2.1). Anschließend werden die Grundlagen zum Modellbegriff (siehe Abschnitt 2.2.2) und zu grafischen Modellierungssprachen (siehe Abschnitt 2.2.3) eingeführt. Es folgt eine vertiefte Beschreibung von Petri-Netzen (siehe Abschnitt 2.2.4), die im Fokus der in dieser Arbeit entwickelten Bewertungsdienste liegen.

2.1. Hochschuldidaktik

Bedingt durch die enge Verbindung von Forschung und Lehre an Universitäten¹ sind Forscherinnen und Forscher oftmals auch in die Hochschullehre involviert. Obwohl die Entwicklung einer speziellen Hochschuldidaktik in der Bundesrepublik Deutschland bereits in den 1960er-Jahren ihre Blütezeit verzeichnete und seit dem Ende der 1990er-Jahre einen weiteren Aufschwung erlebt [Wi13], wird kritisiert, dass beim akademischen Lehrpersonal aktuell die fachliche Expertise im Vordergrund steht und eine entsprechende pädagogische Ausbildung allenfalls »Beiwerk« sei

¹ Beispielsweise ist nach dem Landeshochschulgesetz Baden-Württemberg § 2 Absatz 1 Satz 1 LHG die Pflege und Entwicklung der Wissenschaften in Verbindung von Forschung, Lehre, Studium und Weiterbildung Aufgabe der Universitäten. Siehe Gesetz über die Hochschulen in Baden-Württemberg (Landeshochschulgesetz - LHG) in der Fassung der Bekanntmachung vom 1. Januar 2005, zuletzt geändert durch Artikel 1 des Gesetzes vom 13. März 2018 (GBl. S. 85).

2. Grundlagen

[Re15]. Auch dies belegt, dass es nach wie vor von großer Bedeutung ist, bereits bestehende hochschuldidaktische Konzepte umzusetzen und somit zu einer breiteren Anwendung zu bringen. In den folgenden Abschnitten werden für diese Arbeit relevante Eckpunkte aus dem Bereich der Hochschuldidaktik entnommen und vorgestellt: der Didaktikbegriff und das Constructive Alignment (siehe Abschnitt 2.1.1) sowie Lernziele und der Kompetenzbegriff (siehe Abschnitt 2.1.2). Zudem erfolgt eine Vorstellung des Begriffs E-Assessment (siehe Abschnitt 2.1.3) sowie der Begriffe Bewertung und Feedback (siehe Abschnitt 2.1.4).

2.1.1. Didaktikbegriff und Constructive Alignment

Die allgemeine Didaktik als Wissenschaft des Lehrens und Lernens befasst sich mit Fragestellungen rund um Lehr- und Lernprozesse, die unabhängig von spezifischen Lerninhalten sind. Der Erwerb fachbezogener Lerninhalte sowie fachlicher Kompetenzen ist dahingegen Bestandteil der sogenannten Fachdidaktiken. Eine pragmatische und dennoch vielzitierte Beschreibung für den Didaktikbegriff liefern JANK und MEYER [JM02]: »Die Didaktik kümmert sich um die Frage, wer was von wem wann mit wem wo wie womit und wozu lernen soll«. Eine differenziertere Betrachtung wird von KLAFFKI [Kl64] vorgenommen:

Definition 2.1: Didaktik und Methodik

Unter dem Begriff **Didaktik** können »alle auf die Inhaltlichkeit, auf das *Was* des Unterrichts und der Bildung gerichteten gedanklichen Bemühungen« gesammelt werden, wohingegen der Begriff **Methodik** auf das *Wie* gerichtete Fragen umfasst [Kl64, S. 9].

Da die Bestimmung von Bildungsinhalten und zu erlangenden Kompetenzen den methodischen Überlegungen zeitlich vorangestellt wird, nimmt die Methodik bei KLAFFKI [Kl64, S. 53ff] gegenüber der Didaktik eine nebengeordnete Position ein, was auch als »Primat der Didaktik« bezeichnet wird.

Nach WILDT und WILDT [WW11] hängen Bologna-Prozess, der »Shift from Teaching to Learning« sowie die Outcome- und Kompetenzorientierung in der Entwicklung der deutschen Hochschuldidaktik eng miteinander zusammen. Dabei gilt es, so die Autoren, vor diesem Spannungsfeld eine Balance zwischen Lernergebnissen (engl. *learning outcomes*), dem Lehr-/Lernprozess und entsprechenden Prüfungsformaten herzustellen. Die Integration exakt dieser drei Elemente und ihrer Wechselwirkungen

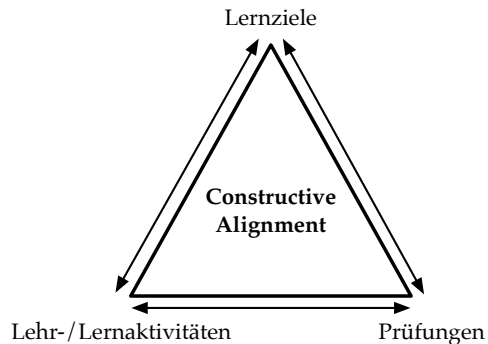


Abb. 2.1.: Constructive Alignment als »goldenes Dreieck« [Ga17a]

in einem gemeinsamen Grundkonzept wurde bereits von BIGGS [Bi96] und BIGGS und TANG [BT11] mit dem sogenannten Constructive Alignment vorgestellt (siehe Abb. 2.1). Das Konzept des Constructive Alignment ist im internationalen Hochschulraum weit verbreitet und unterstützt mit der kompetenzorientierten Gestaltung der Lehre eine zentrale Anforderung der Bologna-Reform [Sc12a]. Eine hohe Bildungsqualität entsteht demnach durch die erzielte Übereinstimmung im sogenannten »goldenen Dreieck« (siehe [Ga17a]) zwischen den folgenden drei Komponenten und ihren jeweiligen Kernfragen:

1. **Lernziele (didaktischer Aspekt):** Was sind die Lerninhalte? Was sind Lernziele bzw. die gewünschten Lernergebnisse?
2. **Lehr-/Lernaktivitäten (methodischer Aspekt):** Wie sollen die festgelegten Inhalte verinnerlicht und Lernziele erreicht werden? Welche Lehr-/Lernformen sollen dabei zum Einsatz kommen?
3. **Prüfungen:** Anhand welcher Prüfungsformate sollen die Lernergebnisse der Teilnehmer überprüft und quantifiziert werden? Welche Bewertungsschemata werden dabei angewendet?

Demnach erfordert eine konsequente Kompetenzorientierung in der Hochschullehre nicht nur die Betrachtung von didaktischen und methodischen Aspekten, vielmehr muss das Zusammenwirken von Lehr/Lern- und Prüfungsaufgaben angepasst und geeignete Prüfungsformate entwickelt werden [IW09].

Mit dem Ziel, ein E-Assessment-Tool zu entwickeln (ad FF2, siehe Abschnitt 1.2), konzentriert sich das in dieser Arbeit beschriebene Forschungsvorhaben in erster Linie auf den Aspekt der Prüfungen. Für Prüfungen gilt analog zur Methodik, dass gemäß

2. Grundlagen

WISSENSDIMENSION	PROZESSDIMENSION					
	Erinnern	Verstehen	Anwenden	Analysieren	Evaluiieren	Erschaffen
Faktenwissen						
Konzeptuelles Wissen						
Prozedurales Wissen						
Metakognitives Wissen						

Abb. 2.2.: Lernzieltaxonomie von ANDERSON und KRATHWOHL [AK01]. Weiß gefärbte Zellen stellen unzulässige Kombinationen dar.

dem »Primat der Didaktik« [KI70] die zugrundeliegenden Lernziele zuvor betrachtet werden müssen. Daher erfolgt im Rahmen dieses Forschungsvorhabens eine vorangehende Auseinandersetzung mit Lernzielen und Kompetenzen für das Gebiet der grafischen Modellierung (ad FF1, siehe Abschnitt 1.2). Der methodische Aspekt der Auswahl und Umsetzung passender Lehr-/Lernaktivitäten wird allerdings explizit ausgeklammert.

2.1.2. Lernziele und Kompetenzbegriff

Lernziele sind »die möglichst exakte Beschreibung des angestrebten Lernergebnisses« [AKS99]. MAGER [Ma72] zufolge sollen Lernziele zudem operationalisierbar sein, d.h. sie sind anhand beobachtbarer Handlungen oder greifbarer Artefakte überprüfbar.

Zur Klassifizierung von Lernzielen wird in der wissenschaftlichen Literatur häufig die Taxonomie kognitiver Lernziele von ANDERSON und KRATHWOHL [AK01] aufgegriffen, die eine Weiterentwicklung der Bloom'schen Taxonomie [Bl73] (siehe Abschnitt 1.1) darstellt. Mit dieser Taxonomie können Lernziele in einer Matrix auf einer Prozessdimension sowie auf einer Wissensdimension einsortiert werden, wie in Abb. 2.2 dargestellt. Auf der Prozessdimension stellt »Erinnern« die niedrigste und »Erschaffen« die höchste Stufe von kognitiven Prozessen dar. Die folgende Vorstellung der Stufen wird – wie auf dem Gebiet üblich – um Beispiele für typische Aufgabenformulierungen ergänzt².

² Zwar sind diese Beispiele nicht immer eindeutig einer einzelnen Stufe zuzuordnen, sie dienen jedoch als Anhaltspunkt und Unterstützung bei der Gestaltung überprüfbarer Lernziele.

Erinnern: Abrufen von relevanten Informationen (wortwörtlich) aus dem Gedächtnis (»Benennen Sie ...«, »Geben Sie ... wieder«)

Verstehen: Bedeutung von Informationen erkennen und diese (mit eigenen Worten) wiedergeben (»Fassen Sie ... zusammen«, »Nennen Sie ein Beispiel für ...«)

Anwenden: Durchführen eines bestimmten Verfahrens zur Lösung von Problemen eines bekannten Typs (»Berechnen Sie ...«, »Erstellen Sie ...«)

Analysieren: Ein Problem in einzelne Teile zerlegen und Beziehungen dieser Teile zueinander bestimmen (»Stellen Sie ... gegenüber«, »Schlussfolgern Sie ...«)

Evaluieren: Kriteriengeleitete Beurteilung von Sachverhalten (»Werten Sie ... aus«, »Geben Sie eine begründete Entscheidung für ...«)

Erschaffen: Zusammensetzen von mehreren Elementen zu einer neuen größeren Einheit oder einer neuen Struktur (»Entwerfen Sie ...«, »Entwickeln Sie ...«)

Die niedrigste Stufe auf der Wissensdimension bildet das Faktenwissen (isolierte »bits of information« [AK01], z. B. Begriffe der Fachterminologie), welches in der nächsthöheren Stufe des konzeptuellen Wissens um das Wissen über die Bedeutung in einem größeren Kontext ergänzt wird (z. B. Kategorien, Theorien und Prinzipien). In der nächsten Stufe folgt das Wissen um bestimmte Handlungsabläufe (z. B. Fertigkeiten, Techniken), das als prozedurales Wissen bezeichnet wird. Das metakognitive Wissen (z. B. Selbstreflektion, Strategieentwicklung) stellt die höchste Stufe der Wissensdimension dar. Anhand der Taxonomiematrix wird deutlich, dass mit zunehmender Stufe auf der Prozessdimension auch die Stufe auf der Wissensdimension ansteigt. Beispielsweise ist für Lernziele der Stufe »Erinnern« lediglich der Abruf von Faktenwissen erforderlich, wohingegen zur Erfüllung von Lernzielen der Stufe »Anwenden« sowohl auf Faktenwissen als auch auf konzeptuelles und prozedurales Wissen zurückgegriffen werden muss.

Bei der Auswahl und Formulierung von Lernzielen sollen im Sinne der Kompetenzorientierung stets die zu erlangenden Kompetenzen als Ausgangslage dienen. Diesbezüglich soll nun der Begriff »Kompetenz« genauer untersucht werden. Dieser erfreut sich zwar sehr großer Beliebtheit, es existiert jedoch kein einheitliches Begriffsverständnis. ILLER und WICK [IW09] beschreiben ein Basiskonzept für den Kompetenzbegriff, welches an die verbreitete Definition von WEINERT [We01] angelehnt wurde und größtenteils konsensfähig ist:

2. Grundlagen

Definition 2.2: Kompetenz

»Kompetenz ist [...] eine individuelle oder kollektive Voraussetzung zur Lösung komplexer Aufgaben, sie basiert auf kognitiven und (je nach Autor/in) motivationalen, volitionalen sowie sozialen Ressourcen, die in einem Lernprozess angeeignet werden müssen, um sie für die erfolgreiche Bewältigung von Umwelthanforderungen einsetzen zu können« [IW09, S. 195].

Eine bestimmte Kompetenz zu besitzen, bedeutet demnach, dass eine Person dazu fähig ist (bzw. die entsprechende Voraussetzung besitzt), bestimmte Probleme oder Aufgaben zu lösen. Dazu greift die (kompetente) Person auf verschiedene Ressourcen zurück. Bei der (kompetenten) Problemlösung stehen kognitive Ressourcen wie Kenntnisse und Fertigkeiten (Denk- oder Bewegungsmuster) zwar im Vordergrund. Jedoch spielen auch psychologische Faktoren wie Motivation, Selbststeuerung, Entscheidungsfähigkeit (motivationale und volitionale Ressourcen) eine wichtige Rolle. Schließt die Problemlösung auch die Kommunikation mit anderen Personen ein, so muss weiterhin auf soziale Ressourcen (z.B. Einfühlungsvermögen oder Kritikfähigkeit) zurückgegriffen werden.

Da sich eine Kompetenz als reine Voraussetzung zur Lösung einer bestimmten Aufgabe nicht direkt beobachten lässt, ist es notwendig, Handlungen zu beschreiben, deren Ausführung die zu erfassende Kompetenz voraussetzt. Die Kompetenzmessung erfolgt dann indirekt als Messung der beobachtbaren Handlung, der Performanz [SHB13], oder der durch diese Handlung geschaffenen Artefakte. Das in dieser Arbeit entwickelte E-Assessment-Werkzeug legt den Fokus auf die Prüfung von Kompetenzen bezüglich kognitiver Ressourcen anhand von graphbasierten Diagrammen. Im Kompetenzmodell werden jedoch auch motivationale, volitionale und soziale Ressourcen berücksichtigt.

2.1.3. E-Assessment und Aufgabenklassen

Der aus dem Englischen stammende Begriff »Assessment« (auf deutsch etwa Bewertung, Beurteilung oder Einschätzung) wird für Prüfungen benutzt, die Fähigkeiten und Leistungsvermögen von Personen messen und bewerten. Assessments finden nicht nur in einem Lehr-/Lernkontext Anwendung, sondern auch z. B. bei Bewerbungsverfahren in Unternehmen.

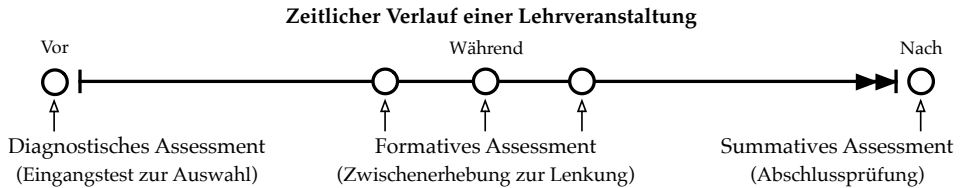


Abb. 2.3.: Mögliche Zeitpunkte für ein Assessment, in Anlehnung an [RM10, S. 14]

Definition 2.3: E-Assessment

Der Begriff E-Assessment umfasst das »Spektrum der auf den neuen (elektronischen) Informations- und Kommunikationstechnologien basierenden Verfahren der lernzielbezogenen Bestimmung, Beurteilung, Bewertung, Dokumentation und Rückmeldung der jeweiligen Lernvoraussetzungen, des aktuellen Lernstandes oder der erreichten Lernergebnisse/-leistungen vor, während [...] oder nach Abschluss [...] einer spezifischen Lehr-/Lernperiode« [BS07, S. 5].

Diese Definition³ für den Begriff E-Assessment ist speziell auf den Einsatzkontext Lehre ausgerichtet. Sie beschreibt auch die unterschiedlichen möglichen Zeitpunkte für ein Assessment⁴, die in Abb. 2.3 dargestellt werden. Für das in dieser Arbeit entwickelte E-Assessment-Werkzeug wurden in erster Linie formative und summative Einsatzzeitpunkte betrachtet. Die Anwendung als diagnostisches Werkzeug ist aber grundsätzlich ebenfalls möglich.

Für eine Klassifikation der Werkzeuge auf dem Gebiet des E-Assessments gibt es in der Literatur zahlreiche Vorschläge (für eine Übersicht siehe z. B. [RM10, S.15-17]). Frühe Arbeiten konzentrieren sich dabei besonders auf die Darstellung der technischen Umsetzung von E-Assessment-Systemen. Beispielsweise definieren BULL und MCKENNA [BM04] die Klassen »Stand-alone«, »Closed network/intranet«, »Internet« für den Bereich des sogenannten computergestützten Assessments (engl. *Computer-Assisted/Aided Assessment, CAA*), welches sie als »the use of computers in the assessment of student learning« [BM04, S. XIV] definieren. CONOLE und WARBURTON [CW05] greifen diese Klassifikation auf und erweitern sie um eine Unterscheidung zwischen dem computerbasierten Assessment (engl. *Computer-Based*

³ Die Definition stammt aus einem unveröffentlichten Manuskript von Egon Bloh aus dem Jahr 2006 und wird von vielen Quellen, u.a. von BRAHM und SEUFERT [BS07] zitiert.

⁴ In dieser Arbeit wird der Begriff Assessment als übergeordneter Begriff aufgefasst, der sowohl traditionelle (z. B. analoge) Beurteilungsverfahren als auch das E-Assessment einschließt.

2. Grundlagen

Tab. 2.1.: Digitalisierungs- bzw. Automatisierungsgrad beim Assessment

Komponenten des Assessments			
Eingabe und Erfassung der Lösung (Digitalisierungsgrad der Lösung)		Bewertung der Lösung (Automatisierungsgrad der Bewertung)	
<i>vollständig analog</i>	Handschriftliche Lösungen, die z. B. im Rahmen einer Übung oder Klausur erstellt werden	<i>vollständig manuell</i>	Manuelle Korrektur der Lösungen und Notengebung durch Lehrpersonal
<i>anteilig analog bzw. digital</i>	Konvertierung handschriftlicher Lösungen anhand eines Scans, ggf. Anwendung von Verfahren zur Text- oder Mustererkennung	<i>anteilig manuell bzw. automatisiert</i>	Umfasst manuelle und automatisierte Arbeitsschritte (z. B. automatisierter Ausschluss offensichtlich falscher Lösungen vor der manuellen Bewertung)
<i>vollständig digital</i>	Direkte Eingabe und Erfassung der Lösungen durch Studierende an einem digitalen Endgerät (z. B. Computer, Tablet, ...)	<i>vollständig automatisiert</i>	Automatisierte Bewertung aller Aspekte der eingereichten Lösungen (z. B. Klausur im Antwort-Wahl-Verfahren)

Assessment, CBA), bei dem sowohl Eingabe als auch Bewertung mit Hilfe von Computern erfolgen und dem weiter gefassten CAA, bei dem entweder Eingabe oder Bewertung computerunterstützt durchgeführt werden. In diesem Sinne werden in dieser Arbeit die Eingabe und Erfassung sowie die Bewertung studentischer Lösungen als die zwei grundsätzlichen Komponenten eines Assessments aufgefasst. Um einen entsprechenden Überblick über die verschiedenen Ausprägungen in Bezug auf den Digitalisierungs- bzw. Automatisierungsgrad⁵ geben zu können, werden in Tab. 2.1 einige Beispiele aufgeführt. Anhand dieser Tabelle lassen sich ebenso verschiedene Kombinationsmöglichkeiten ableiten, um die folgenden Punkte zu verdeutlichen:

- Anhand einer vollständig analogen Lösung kann weder eine anteilige noch eine vollständige Automatisierung der Bewertung realisiert werden. Lediglich eine vollständig manuelle Bewertung ist möglich. Da in diesem Fall keinerlei Computer- oder IT-Unterstützung stattfindet, handelt es sich **nicht um ein E-Assessment**. Erst eine nachträgliche Digitalisierung einer analogen Lösung oder die direkte digitale Eingabe von Lösungen ermöglicht die Anwendung einer anteiligen oder vollständig automatisierten Bewertung.

⁵ Auch FERSTL und SINZ [FS13, S. 55] unterscheiden bezüglich des Automatisierungsgrads von Aufgaben in Geschäftsprozessen drei Stufen: nicht-automatisiert, teilautomatisiert und vollautomatisiert, je nachdem ob personelle (hier manuell) oder maschinelle (hier automatisiert) Aufgabenträger an der Durchführung einer Aufgabe beteiligt sind. Analog dazu wurde in dieser Arbeit der Digitalisierungsgrad ebenfalls anhand von drei Stufen festgelegt.

- Die Kombination einer vollständig digitalen Lösung mit der vollständig automatisierten Bewertung bildet ein **vollautomatisiertes E-Assessment**. Dieses kann dem Bereich der computerbasierten Bewertung (CBA) zugeordnet werden.
- Alle übrigen Kombinationsmöglichkeiten, bei denen *anteilig oder vollständig digitale Lösungen* oder eine *anteilig oder vollständig automatisierte Bewertung* einfließen, bilden ein **semi-automatisiertes E-Assessment**. Dies entspricht auch dem zuvor vorgestellten Bereich der computerunterstützten Bewertung (CAA).

Dem in dieser Arbeit entwickelten E-Assessment-Werkzeug liegt das Ziel zugrunde, die vollständig digitale Eingabe und Erfassung studentischer Lösungen umzusetzen und auf diesen eine vollständig automatisierte Bewertung durchzuführen. Es handelt sich also um ein vollautomatisiertes E-Assessment. Die Ergebnisse der automatisierten Bewertung können bei entsprechender Konfiguration vor Veröffentlichung manuell überprüft und angepasst werden. In diesem Fall ergibt sich ein semi-automatisiertes E-Assessment. Eine Zuordnung des E-Assessment-Werkzeugs ist sowohl zum Bereich CBA als auch zum Bereich CAA möglich.

Neben der didaktisch-methodischen Planung und der technischen Realisierung eines E-Assessment-Werkzeugs sind in der Praxis weiterhin organisatorische und rechtliche Herausforderungen zu meistern (siehe z. B. [Ha16]), insbesondere wenn das E-Assessment-Werkzeug summativ zur Leistungsfeststellung eingesetzt wird. Nicht immer stehen an Hochschulen ausreichend Plätze für die direkte digitale Erfassung (z. B. mit Computern, Tablets oder anderen Endgeräten) zur Verfügung um ein summatives E-Assessment zeitgleich mit einer großen Kohorte an Studierenden durchführen zu können. Auch die Identitätsfeststellung sowie die Einhaltung des Datenschutzes müssen gewährleistet werden. Ebenso gilt es, Maßnahmen zur Vermeidung von Täuschungsversuchen zu entwickeln. Diese Herausforderungen liegen nicht im Fokus dieser Dissertation, wurden aber in Folgeprojekten berücksichtigt (siehe Abschnitte 1.4, 5.5.7, 6.3).

Anhand welcher Handlungen oder Artefakte ein Assessment auf den aktuellen Lernstand bzw. die Lernergebnisse zugreift, wird wesentlich durch das Prüfungsformat bzw. die eingesetzten Aufgabentypen bestimmt. Für das Gebiet des E-Assessments ist das breite Spektrum an klassischen Prüfungsformaten neben der schriftlichen Prüfung (Mündliche Prüfung, Referat, Seminararbeit, Laborprotokoll, Portfolio, ...) größtenteils durch fehlende technische Werkzeuge zur (semi-)automatisierten

2. Grundlagen

Bewertung zum aktuellen Zeitpunkt stark eingeschränkt⁶. Auch die üblichen Aufgabentypen in schriftlichen Prüfungen sind in dieser Hinsicht begrenzt. Dies lässt sich anhand der Klassifikation von Aufgaben in die beiden Klassen konvergenter und divergenter Aufgaben (siehe z. B. [Gr09; Mc02]) verdeutlichen:

- **Konvergente Aufgaben** (auch geschlossene Aufgaben) haben nur eine einzige, präzise definierte, richtige Lösung. Typische Beispiele dafür sind Antwort-Wahl-Aufgaben, Sortieraufgaben (Rang- bzw. Reihenfolge bestimmen) oder Zuordnungsaufgaben (z. B. Elemente zweier Listen zueinander zuordnen).
- **Divergente Aufgaben** besitzen einen höheren Freiheitsgrad und damit einhergehend mehrere verschiedene richtige Lösungen oder unterschiedliche Lösungswege. Dazu gehören z. B. Freitextaufgaben, Rechen- und Beweis- und Programmieraufgaben sowie der in dieser Arbeit behandelte Aufgabentyp »Modell erstellen« (siehe Abschnitt 3.1.2).

Konvergente Aufgaben lassen sich zwar ohne weiteres vollständig automatisiert bewerten, es lassen sich mit ihnen jedoch zumeist lediglich Lernziele der unteren Niveaustufen gemäß der in Abschnitt 2.1.2 vorgestellten Lernzieltaxonomie von ANDERSON und KRATHWOHL [AK01] überprüfen (siehe z. B. [St11, S. 21])⁷. Divergente Aufgaben erreichen zwar höhere Niveaustufen, gleichzeitig steigt dadurch aber auch die Komplexität der Bewertung [Mc02]. Das gilt sowohl für die manuelle Bewertung als auch für Verfahren zur automatisierten Bewertung. Letztere sind aus diesem Grund für divergente Aufgaben seltener anzutreffen. In vielen Fachdisziplinen sind daher Forschungsbemühungen zu beobachten, die sich auf domänenspezifische E-Assessment-Werkzeuge zur automatisierten Bewertung divergenter Aufgaben konzentrieren. Im Folgenden werden einige Beispiele aufgeführt:

- Freitextaufgaben sind die vorherrschende Form divergenter Aufgaben. Hierbei wird gefordert, einen längeren zusammenhängenden Text zu einer gegebenen Problemstellung zu verfassen. Beispielsweise existieren Ansätze für die automatisierte Bewertung von Essays (engl. *Automated Essay Scoring*, AES) durch die kombinierte Anwendung von Methoden der Computerlinguistik und Klassifikationsverfahren aus dem Gebiet des maschinellen Lernens (siehe

⁶ Derzeit liefert die Forschung zu Large Language Models (LLM) aus dem Gebiet des maschinellen Lernens vielversprechende Ergebnisse (siehe Abschnitt 5.6.6), so dass sich dies in den kommenden Jahren ändern könnte.

⁷ Gegebenenfalls kann dem durch eine entsprechende Gestaltung entgegengewirkt werden. So finden sich auch anderslautende Ansichten in der Literatur, z. B. der Artikel »Multiple Choice Questions Not Considered Harmful« von WOODFORD und BANCROFT [WB05].

z. B. PAGE [Pa68] als Pionier des Ansatzes oder eine aktuellere Arbeit, wie z. B. [BV20]).

- Für die Bewertung von Aufgaben zu mathematischen Berechnungen existieren Ansätze zur Automatisierung, die ein Computeralgebrasystem (CAS) einbinden. Dadurch wird ermöglicht, dass nicht nur das Endergebnis überprüft, sondern die eingereichte Lösung schrittweise validiert wird. Somit können auch teilweise korrekte Lösungen identifiziert und Teilpunkte vergeben werden (siehe z. B. [Sa13]).
- Auf dem Gebiet der Chemie erfordert sowohl die Eingabe als auch die automatisierte Bewertung von Strukturformeln und Reaktionsgleichungen eine domänenspezifische Anpassung bestehender E-Assessment-Systeme. Hierzu wurde z. B. in [PS19] ein Ansatz vorgestellt.
- Auch in der Physik wird eine grafische Darstellung zur Anwendung des Schnittprinzips in der Mechanik eingesetzt. In [Ru20] wird eine Schnittstelle für die automatisierte Bewertung von gezeichneten Freikörperbildern (engl. *free body diagram*) entworfen.
- Wird im Rahmen überschaubarer Programmieraufgaben ein Quelltext eingereicht, so kann dieser automatisiert auf statische und dynamische Eigenschaften überprüft werden, z. B. mit Testverfahren aus dem Gebiet der Softwareentwicklung. Auf dem heutigen Stand existiert eine Reihe von Systemen, die speziell für den Hochschulkontext entwickelt wurden und eine Bandbreite an unterschiedlichen Programmiersprachen abdecken (z. B. Autolab der Carnegie Mellon University⁸, Praktomat am KIT [BHS17] oder JACK beim Verbundpartner an der Universität Duisburg-Essen [SGB08]).

In diese (weitaus unvollständige) Liste von Forschungsbestrebungen reiht sich auch die vorliegende Arbeit mit dem Gebiet der grafischen Modellierung ein. Die verwandten Ansätze auf diesem Feld werden im folgenden Kapitel in Abschnitt 3.3.2 vorgestellt.

2.1.4. Bewertung und Feedback

Die im vorangehenden Abschnitt vorgestellte Definition für E-Assessment (Def. 2.3) beinhaltet bereits den Begriff »Bewertung«, jedoch ohne diesen genauer auszufüh-

⁸ <http://autolabproject.com>

2. Grundlagen

ren. In dieser Arbeit soll dieser Begriff auf der Basis des breiter gefassten Begriffs »Urteilen« wie folgt definiert werden:

Definition 2.4: Urteilen (Bewertung)

»Beim Urteilen wird einem Urteilsobjekt ein Wert auf einer Urteilsdimension zugeordnet. Urteile können auf verschiedenen Urteilsdimensionen abgegeben werden.« Die Urteilsdimension Bewertung »[...] kann man sich als Skala mit Endpunkten wie z.B. »gut - schlecht«, »zielbehindernd - zielbefördernd« oder »schwerwiegend - unbedenklich« vorstellen« [BFP11, S. 2].

BETSCH et al. [BFP11] nennen weitere Beispiele für Urteilsdimensionen neben der Bewertung: Wahrscheinlichkeit, Zeitdauer, räumliche Größe, Vertrautheit oder Glaubwürdigkeit, die je nach Urteilszweck zum Einsatz kommen. Urteile bilden demnach eine Grundvoraussetzung für alltägliche Denkprozesse wie Entscheiden und Problemlösen [BFP11, S. 2].

Übertragen auf die Terminologie dieser Arbeit im Kontext Hochschullehre wird die Bewertung als Vorgang (Urteilen auf der Urteilsdimension Bewertung) aufgefasst, bei welchem ein Bewertungsergebnis (Urteil) zu einer studentischen Leistung (Urteilsobjekt) erstellt wird. Dieses Bewertungsergebnis soll ausdrücken, inwieweit die studentische Leistung die festgelegten Lernziele (siehe auch Abschnitt 2.1.2) erfüllt. Die Bewertung wird im Hochschulkontext als Instrument zur Leistungsbeurteilung eingesetzt. Dabei drücken Bewertungsergebnisse nicht nur aus, an welcher Stelle eine individuelle Leistung auf einer Skala von »gut - schlecht« einzuordnen ist, sondern stützen letztendlich die Entscheidung, ob ein Studienabschluss bestanden oder nicht bestanden ist.

Bewertungsergebnisse können dabei in unterschiedlichen Formen und für verschiedene Zwecke eingesetzt und auf höheren Ebenen aggregiert werden, wie die folgenden Beispiele zeigen:

- Bewertung einer mündlichen Prüfung zur Erreichung einer Klausurzulassung mit dem Ergebnis »bestanden« oder »nicht bestanden«
- Bewertung einer Übungsaufgabe zur Rückmeldung des aktuellen Leistungsstands an Studierende anhand der erreichten Punktzahl gemessen an der maximal erreichbaren Punktzahl für diese Aufgabe, z. B. 8 von 10 Punkten.
- Bewertung einer Klausur anhand der erreichten Punktzahl über alle Aufgaben, gemessen an der maximal erreichbaren Punktzahl über alle Aufgaben, z. B. 76

von 90 Punkten. Ein derartiges Ergebnis kann anhand eines festgelegten Schlüssels in ein Ergebnis auf einer Skala umgerechnet werden, z. B. die im deutschen Schulwesen gebräuchliche Notenskala von 1 (sehr gut) bis 6 (ungenügend)⁹.

- Bewertung des Studienabschlusses anhand der nach Leistungspunkten gewichteten Durchschnittsnote über alle belegten Module, Pflichtveranstaltungen und die Abschlussarbeit.

Ein Bewertungsergebnis muss dabei immer eine relative Interpretation ermöglichen. Das bedeutet, es muss erkennbar sein, an welcher Position auf der Urteilsdimension (z. B. 3 auf einer Skala von 1 bis 6 oder 0.5 im Intervall $[0, 1]$, was 50 % entspricht) oder in welchem Abstand zum bestmöglichen Ergebnis (z. B. 5 von 10 Punkten) das festgestellte Ergebnis anzusiedeln ist. Die verwendete Skala muss mindestens ordinalskaliert sein, damit Vergleichsoperationen (größer oder kleiner) durchgeführt werden können, zudem muss festgelegt sein, welche Bedeutung das obere und untere Ende der Skala besitzen. Beispielsweise hat sich im deutschen Schulwesen in der gymnasialen Oberstufe auch die Punkteskala von 0-15 etabliert. Bei dieser entspricht der Wert 0 dem Ergebnis »ungenügend« und der Wert 15 dem Ergebnis »sehr gut«¹⁰.

Aus den bisherigen Ausführungen ergibt sich, dass beispielsweise die Bestimmung der Anzahl von Fehlern in einem Diktat (z. B. 5 Fehler) alleine noch nicht zu einem Bewertungsergebnis nach den o.g. Anforderungen führt. Auf Grundlage zu berücksichtigender Bezugsgrößen, wie z. B. dem Umfang und der Schwierigkeit des Diktats, muss eine Umrechnung anhand eines festgelegten Schlüssels stattfinden, so dass die Anzahl an Fehlern in ein entsprechendes Bewertungsergebnis übertragen werden kann (z. B. 0-1 Fehler: Note 1, 2-4 Fehler: Note 2, 5-8 Fehler: Note 3, 9-15 Fehler: Note 4, >15 Fehler: Note 5).

Besonders zur Unterstützung der Bewertung von komplexeren und kompetenzorientierten Aufgaben wird im Hochschulkontext die Verwendung von kriteriengeleiteten Bewertungsschemata (engl. *Rubrics*) empfohlen [Bo21; Sc21; SL13]. Dabei werden auf einer Dimension verschiedene Bewertungskriterien festgelegt, die es jeweils zu

⁹ Siehe z. B. die Verordnung des Kultusministeriums Baden-Württemberg über die Notenbildung (Notenbildungsverordnung, NVO) vom 5. Mai 1983, Stand: Letzte berücksichtigte Änderung: § 9 geändert durch Artikel 14 der Verordnung vom 18. Juni 2020 (GBl. S. 577, 589).

¹⁰ Siehe z. B. die Verordnung des Kultusministeriums Baden-Württemberg über die Jahrgangsstufen sowie die Abiturprüfung an Gymnasien der Normalform und Gymnasien in Aufbauform (Abiturverordnung Gymnasien der Normalform - AGVO) vom 19. Oktober 2018, Stand: Letzte berücksichtigte Änderung: mehrfach geändert durch Artikel 11 der Verordnung vom 18. Juni 2020 (GBl. S. 577, 585).

2. Grundlagen

Tab. 2.2.: Beispiel eines Rubrics zu einer Erläuterungsaufgabe mit zusätzlicher Gewichtung der Bewertungskriterien

Bewertungskriterien	Grad der Erfüllung		
	Stufe 1 (nicht erfüllt)	Stufe 2 (teilweise erfüllt)	Stufe 3 (vollständig erfüllt)
Kriterium 1 (10 %) Rechtschreibung	Mehr als 10 Rechtschreibfehler	2-9 Rechtschreibfehler	0-1 Rechtschreibfehler
Kriterium 2 (10 %) Grammatik und Stil	Mehr als 8 Grammatik- oder Stilfehler	2-7 Grammatik- oder Stilfehler	maximal 1 Grammatik- oder Stilfehler
Kriterium 3 (80 %) Themenerläuterung	mehr als einen Aspekt ausgelassen	einen Aspekt ausgelassen	alle Aspekte erläutert

prüfen gilt. Der Grad der Erfüllung (Urteilsdimension in Def. 2.4) der Bewertungskriterien wird auf einer weiteren Dimension zusätzlich in verschiedene Stufen unterteilt. Rubrics können sowohl mit aufgabenspezifischen als auch aufgabenübergreifenden Bewertungskriterien ausgestaltet werden. Aufgabenspezifische Bewertungskriterien beziehen sich auf individuelle Aspekte der Aufgabe, wie z. B. das Vorhandensein bestimmter Inhalte in einem Text. Aufgabenübergreifende Bewertungskriterien können für verschiedene Aufgaben verwendet werden, beispielsweise zur Rechtschreibung und Grammatik.

Tab. 2.2 zeigt ein Beispiel für ein Rubric, das z.B. für die Bewertung einer Erläuterungsaufgabe im Deutschunterricht zum Einsatz kommen könnte. Für eine zur Aufgabe erstellte Einreichung werden alle im Rubric eingesetzten Bewertungskriterien unabhängig voneinander geprüft und die passende Stufe anhand der Beschreibung in den jeweiligen Spalten ermittelt. Anhand des Rubrics ist es möglich, die erzielten Stufen der einzelnen Bewertungskriterien zu einem Gesamtergebnis zu aggregieren. Dazu können die Bewertungskriterien auch zusätzlich individuell gewichtet werden, wie das Beispiel zeigt. Desto mehr Stufen in der Dimension des Erfüllungsgrads eingesetzt werden, desto differenzierter fällt die Bewertung aus. Im Sinne eines kompetenzorientierten Assessments orientieren sich die in einem Rubric verwendeten Bewertungskriterien nach Möglichkeit an den gewünschten bzw. zu prüfenden Kompetenzen. Im Beispiel in Tab. 2.2 gehören dazu neben der Beherrschung der Rechtschreibregeln, Grammatik- und Stilregeln auch Kenntnisse zu einem bestimmten Thema und die Fähigkeit, diese Kenntnisse in Textform zu erläutern.

Wesentliche Anforderungen an eine Bewertung sind Objektivität und Transparenz.

Laut JONSSON und SVINGBY [JS07] trägt der Einsatz von Rubrics zur Steigerung der Objektivität bei, da dabei die Bewertungskriterien und die Merkmale für den jeweiligen Erfüllungsgrad definiert werden und somit weniger Abweichungen durch eine subjektive Interpretation von Kriterien zu erwarten sind. In diesem Zusammenhang werden bei einer Evaluation von Rubrics häufig die Interrater-Reliabilität und die Intrarater-Reliabilität untersucht. Die Interrater-Reliabilität gibt an, wie zuverlässig verschiedene Personen, die ein gemeinsames Rubric einsetzen, damit zum gleichen Ergebnis gelangen. Die Intrarater-Reliabilität bezieht sich auf den Einsatz eines Rubrics durch eine Person. Hier kann es dazu kommen, dass eine einzelne Person zu unterschiedlichen Zeitpunkten bei der Betrachtung derselben studentischen Leistung zu abweichenden Ergebnissen gelangt. Durch den Einsatz von Rubrics können sowohl Intrarater- als auch Interrater-Reliabilität gesteigert werden [JS07]. Werden Rubrics zusammen mit dem Bewertungsergebnis veröffentlicht, wird nicht nur das Bewertungsverfahren transparent gemacht, sondern auch der Lernprozess unterstützt, da anhand der Bewertungskriterien die kompetenzbezogenen Anforderungen vermittelt werden [Sc21].

Ebenso bietet der Einsatz von Rubrics eine Grundlage für die Gestaltung informativer und lernförderlicher Rückmeldungen [JS07]. Rückmeldungen, im weiteren Verlauf der Arbeit als Feedback (siehe Def. 2.5) bezeichnet, spielen besonders im Kontext formativer Assessments eine bedeutende Rolle im Lernprozess.

Definition 2.5: Feedback

Feedback ist eine Information zu bestimmten Aspekten der eigenen Leistung oder des eigenen Verständnisses, die von einem anderen Agenten (z. B. einem Lehrer, einem anderen Lernenden oder einem automatisierten Bewertungsverfahren) bereitgestellt wird. (Übersetzt und angelehnt an [HT07].)

Nach dieser Definition handelt es sich strenggenommen auch bei einem Bewertungsergebnis in Form einer Gesamtpunktzahl, einer Note oder der simplen Rückmeldung ob richtig oder falsch, um Feedback. Damit Feedback sich allerdings förderlich auf den Lernprozess auswirkt, sollte ein darüber hinausgehendes, sogenanntes »elaboriertes Feedback« zum Einsatz kommen (siehe z. B. [CCG87]). Ebenso sollten bei der Gestaltung von Feedback-Nachrichten didaktische Empfehlungen berücksichtigt werden (z. B. [HT07; Na14]).

HATTIE und TIMPERLEY [HT07] beschreiben ein Feedback-Modell für lernförderli-

2. Grundlagen

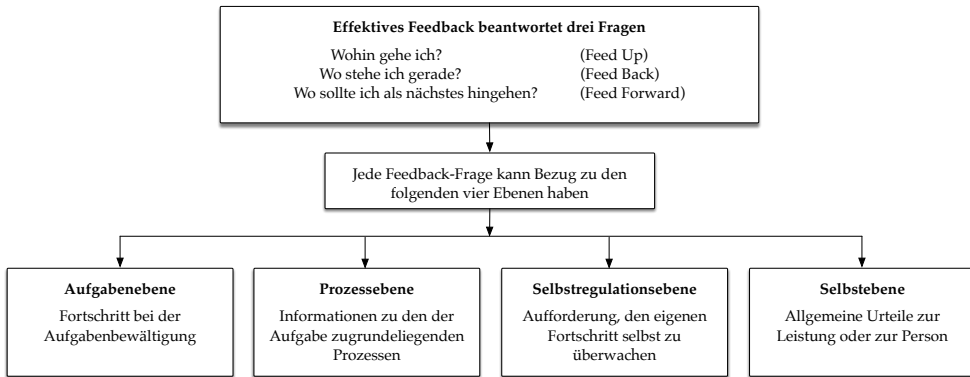


Abb. 2.4.: Feedback-Modell für effektives Feedback, übersetzt und modifiziert nach [HT07, S. 87]

ches¹¹ Feedback, welches in Abb. 2.4 zusammengefasst dargestellt wird. Demnach sollte Feedback den Lernenden stets Antworten zu drei Kernfragen liefern: »Wohin gehe ich?« (Feed Up) bezieht sich auf die Vermittlung der grundlegenden Ziele und der zugrundeliegenden Bewertungskriterien und Kompetenzen, die hinter einer Aufgabe stehen. »Wo stehe ich gerade?« (Feed Back) legt offen, wo Lernende in Bezug auf diese Ziele zum aktuellen Zeitpunkt stehen. »Wo sollte ich als nächstes hingehen?« (Feed Forward) liefert Vorschläge, welche Schritte oder Strategien zur Zielerreichung durchgeführt werden könnten. Bei diesen drei Fragen können darüber hinaus Bezüge zu vier unterschiedlichen Ebenen hergestellt werden. Auf der Aufgabenebene wird Lernenden aufgabenspezifisches Feedback gegeben, welches den Fortschritt bei der Aufgabenbewältigung dokumentiert oder z. B. das Verständnis der Aufgabenstellung verbessern soll. Informationen zu allgemeinen Prozessen, Strategien oder Bewertungskriterien die im Kontext der Aufgabenstellung relevant, jedoch nicht aufgabenspezifisch sind, werden der Prozessebene zugeordnet. Auf der Selbstregulationsebene liefert Feedback keine inhaltlichen Hinweise, sondern regt zur selbstständigen Auseinandersetzung mit der Aufgabenstellung an (z. B. »Versuch es noch einmal!«). Feedback auf der Selbstebene besteht aus einem sehr allgemeinen Urteil und liefert keine konkreten Bezugspunkte zur Aufgabe (z. B. »Gut gemacht!«). Es sollte nur sparsam eingesetzt werden, da in Studien kaum ein positiver Effekt nachgewiesen werden konnte und es sogar demotivierend wirken kann [HT07].

Im Rahmen dieser Arbeit werden zur Untersuchung studentischer Leistungen anhand eines E-Assessment-Werkzeugs kompetenzorientierte Aufgaben zur grafischen

11 In der zitierten Quelle wird der als Synonym interpretierte Begriff »effective feedback« verwendet.

Modellierung gestellt und bearbeitet. Durch das Werkzeug und ergänzende Bewertungsdienste wird dabei folgendes abgedeckt:

- Bewertung der studentischen Leistung durch die Berechnung eines entsprechenden Bewertungsergebnisses
- Generierung von Feedback-Nachrichten zur studentischen Leistung, die sich auf die eingesetzten Bewertungskriterien beziehen
- Darstellung von Hinweisen bezüglich Feed Up und Feed Forward auf Basis des Bewertungsergebnisses

2.2. Grafische Modellierung

Die Modellierung wird in zahlreichen unterschiedlichen Disziplinen zu verschiedenen Zwecken eingesetzt. Wie bereits in der Einleitung (Abschnitt 1.1) beschrieben, kommen bei der grafischen Modellierung zur Darstellung eines Modells grafische Symbole zum Einsatz. In der Informatik und verwandten Disziplinen¹² finden sich verschiedene Anwendungsbereiche für die grafische Modellierung, beispielsweise die Softwareentwicklung, Daten- und Geschäftsprozessmodellierung. Während sich die Forschungsfragen dieser Arbeit (siehe Abschnitt 1.2) primär mit der grafischen Modellierung in diesen Anwendungsbereichen im Allgemeinen befassen, so hat die Verfasserin dieser Dissertation während der Zusammenarbeit im KEA-Modellprojektverbund (siehe Abschnitt 1.4) und bei der Entwicklung von Bewertungsdiensten einen Fokus auf den Anwendungsbereich der Geschäftsprozessmodellierung gelegt. Abschnitt 2.2.1 bietet daher einen Überblick über Begriffe und Einsatzgebiete der Geschäftsprozessmodellierung. Im anschließenden Abschnitt 2.2.2 wird der Modellbegriff erörtert, wobei auch Bezug auf den Begriff des »konzeptuellen Modells« aus der Literatur genommen wird. In Abschnitt 2.2.3 wird zunächst eine theoretische Betrachtung von Modellierungssprachen durchgeführt. Dabei erfolgt auch die Einordnung der Begriffe »grafische Modellierung« und »graphbasiertes Diagramm«, ebenso werden mit dem ER-Modell, UML, EPK und BPMN einige Beispiele für grafische Modellierungssprachen aus den betrachteten Anwendungsbereichen vorgestellt. Da sich die in dieser Arbeit entwickelten Bewertungsdienste auf die

¹² Mit verwandten Disziplinen sind neben WI und IS (siehe Abschnitt 1.1) auch weitere bereichsübergreifende Disziplinen gemeint, die sich aus verschiedenen Fachgebieten zusammensetzen, z. B. die sogenannten »Bindestrich-Informatik«-Fächer wie Bio-, Umwelt- oder Medizin-Informatik.

2. Grundlagen

Geschäftsprozessmodellierung mit Petri-Netzen beziehen, ist deren Beschreibung ein eigener Abschnitt (2.2.4) gewidmet.

2.2.1. Geschäftsprozessmodellierung

Die Geschäftsprozessmodellierung wird oft als Teilgebiet des Geschäftsprozessmanagements (engl. *Business Process Management*, BPM) aufgefasst (siehe z. B. [We19]). In den 1980er- und 1990er-Jahren begann die Ablösung von einer vorwiegend funktionsorientierten Sichtweise hin zu einer prozessorientierten Ausrichtung von Unternehmen¹³ (siehe z. B. [Ga83; HC93]). Demzufolge treten Geschäftsprozesse in den Mittelpunkt von Verwaltungs- und Managementbestrebungen.

Definition 2.6: Geschäftsprozess

Ein **Geschäftsprozess**^a ist »eine Menge von manuellen, teil-automatisierten oder automatisierten Aktivitäten, die in einem Betrieb nach bestimmten Regeln auf ein bestimmtes Ziel hin ausgeführt werden. Die Aktivitäten hängen über betroffene Personen, Maschinen, Dokumente, Betriebsmittel u.ä. miteinander zusammen.« [Ob96, S. 14f].

^a In der Originalquelle [Ob96] wird anstelle des Begriffs »Geschäftsprozess« der synonyme Begriff »betrieblicher Ablauf« verwendet.

Die Definition 2.6 nach OBERWEIS [Ob96] macht deutlich, dass ein Geschäftsprozess über zahlreiche Teilkomponenten in seiner Gesamtheit nicht unmittelbar für die Betrachtung zugänglich ist. Geschäftsprozessmodelle, die einen Geschäftsprozess umfassend darstellen, erleichtern die zielgerichtete Auseinandersetzung mit dem zugrundeliegenden Geschäftsprozess.

In der Literatur werden zahlreiche Einsatzgebiete für die Geschäftsprozessmodellierung beschrieben. Diese können nach van der AALST und STAHL [AS11, S. 32f] anhand von zwei verschiedenen Dimensionen klassifiziert werden. In der ersten Dimension erfolgt eine inhaltliche Ausrichtung eines Geschäftsprozessmodells entweder in Richtung des betrachteten Geschäftsprozesses auf der Organisationsebene oder in die Richtung eines Informationssystems. In der zweiten Dimension lassen

¹³ In dieser Arbeit wird stets vom unspezifischen Oberbegriff der Organisation ausgegangen, womit erwerbswirtschaftlich ausgerichtete Unternehmen, Ein-Mann-Betriebe, gemeinnützige Organisationen und öffentliche Einrichtungen gleichermaßen eingeschlossen werden. Nichtsdestotrotz findet im fortlaufenden Text eine Übernahme von den in der Literatur häufig auftretenden Begriffen wie Geschäft oder Betrieb aus den jeweiligen Originalquellen statt, um die Fachterminologie nicht zu verwässern.

Tab. 2.3.: Einsatzgebiete der Geschäftsprozessmodellierung entlang zweier Dimensionen, in Anlehnung an [AS11, S. 32f]

	Entwurf/Gestaltung	Analyse
Informationssystem	<ul style="list-style-type: none"> • Spezifikation • Implementation • Konfiguration 	<ul style="list-style-type: none"> • Verifikation • Simulation
Geschäftsprozess	<ul style="list-style-type: none"> • Business Process Reengineering • Continuous Process Improvement 	<ul style="list-style-type: none"> • Gamification • Simulation

sich Entwurfs- bzw. Gestaltungs- und Analyseziele voneinander abgrenzen. In Tab. 2.3 sind Beispiele für klassische Einsatzgebiete anhand der beiden Dimensionen aufgeführt. Im Rahmen der Informationssystemgestaltung können Geschäftsprozessmodelle bei der Spezifikation, Implementation und Konfiguration zum Einsatz kommen. Dies betrifft beispielsweise das Anforderungsmanagement im Zusammenhang mit einem Requirement Engineering-Ansatz (Spezifikation), die softwaretechnische Umsetzung von Prozessbestandteilen (Implementation) sowie die Auswahl von Standardsoftware oder die Ausführung in Workflow-Managementsystemen¹⁴ (Konfiguration). In Bezug auf die Gestaltung von Geschäftsprozessen findet eine Anwendung von Methoden aus dem Geschäftsprozessmanagement wie z.B. dem Business Process Reengineering [HC93] oder dem Continuous Process Improvement [Ha91] statt. Das Ziel dieser Methoden ist es, die Qualität der bestehenden Prozesse durch eine mehr oder weniger starke Umgestaltung zu verbessern. Um die Korrektheit von Prozessmodellen zu analysieren, steht eine Reihe an Werkzeugen für die formale Verifikation zur Verfügung (siehe z. B. die Auflistung in [SWL17]). Zu Analyse Zwecken können Prozessmodelle zudem unabhängig von ihrer inhaltlichen Ausrichtung als Grundlage für eine Simulation eingesetzt werden (siehe z. B. [Ob96, S. 20], [BKR00, S. 53], [AS11, S. 32f]), um Durchlaufzeiten und Engpässe zu ermitteln. In aktuellen Gamification-Ansätzen auf dem Gebiet der Prozessmodellierung sollen sich Mitarbeiter eines Unternehmens auf spielerische Weise mit den eigenen Prozessen auseinandersetzen, um auf diese Weise Verbesserungspotential zu ermitteln [Pf17]. Zusätzlich zu den vorgestellten Gestaltungs- und Analyse Zwecken spielen Prozessmodelle in der Praxis auch eine Rolle bei der Dokumentation von Unternehmensabläufen, um diese für Management und Mitarbeiter transparent zu

¹⁴ Ein Workflow-Managementsystem ist eine Softwareanwendung mit deren Hilfe Workflows definiert, verwaltet und ausgeführt werden können [Ho95, S. 6]. In jüngerer Zeit hat sich daneben auch der Begriff prozessorientiertes Informationssystem (engl. *Process-aware Information System* (PAIS)) etabliert [DAH05]. Siehe auch den Abschnitt 2.2.4 zu Petri-Netzen für eine Erläuterung des Begriffs Workflow.

2. Grundlagen

kommunizieren und Wissen weiterzugeben [Ob96, S. 20]. Zudem ist die Dokumentation in Form von Prozessmodellen auch im Rahmen der Zertifizierung (z. B. nach ISO 9000) erforderlich [SN00].

2.2.2. Modellbegriff

Die umgangssprachliche Bedeutung des Begriffs Modell als vereinfachtes Abbild der Wirklichkeit greift als Grundlage für eine Definition im Rahmen dieser Arbeit zu kurz. Vielmehr sind für eine Annäherung an die zentralen Artefakte der hier betrachteten grafischen Modellierung eine Abgrenzung von alternativen Begriffsverwendungen sowie die Betrachtung immanenter Aspekte vonnöten. In seinem grundlegenden Werk »Allgemeine Modelltheorie« hat STACHOWIAK [St73] die im allgemeinen Sprachgebrauch auftretenden Ausprägungen des Modellbegriffs im Wesentlichen auf drei gemeinsame Hauptmerkmale zusammengeführt, die in den folgenden Definitionen 2.7, 2.8 und 2.9 vorgestellt werden.

Definition 2.7: Abbildungsmerkmal

»Modelle sind stets Modelle von *etwas*, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können.« [St73, S. 131]

Definition 2.8: Verkürzungsmerkmal

»Modelle erfassen im allgemeinen *nicht alle* Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant scheinen.« [St73, S. 132f]

Definition 2.9: Pragmatisches Merkmal

»Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion a) für *bestimmte* [...] *Subjekte*, b) *innerhalb bestimmter Zeitintervalle* und c) unter Einschränkung auf *bestimmte gedankliche oder tatsächliche Operationen*.« [St73, S. 133]

Anhand dieser Basisdefinitionen wird deutlich, wie weit der Modellbegriff gefasst werden kann. Beispiele dafür sind die unterschiedlichen Arten von Modellen, die in

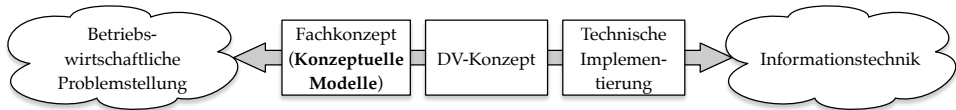


Abb. 2.5.: Einordnung konzeptueller Modelle im ARIS-Konzept. Eigene modifizierte Darstellung in Anlehnung an [Sc98, S. VI]

verschiedenen Disziplinen eingesetzt werden, z.B. mathematische Modelle, die in Formeln beschrieben sind, theoretische Konstrukte und Paradigmen als Modelle zur Erklärung bestimmter Sachverhalte oder Phänomene in den Sozial- und Wirtschaftswissenschaften sowie Modelle in der Chemie oder Physik, die zur Veranschaulichung von kleinen Teilchen und deren Reaktionsweisen dienen. In der wissenschaftlichen Literatur zur grafischen Modellierung in der Informatik taucht häufig der Begriff des konzeptuellen Modells auf. Die kontroverse Auseinandersetzung mit diesem Begriff (siehe z. B. die Arbeiten von THOMAS [Th05] oder THALHEIM [Th09; Th18]) legt nahe, dass dessen zentrale Konzepte selbst innerhalb der Fachdisziplinen einerseits unzureichend definiert und andererseits uneinheitlich verwendet werden. Somit erfolgt an dieser Stelle eine Einordnung des Begriffs, wie er im Rahmen dieser Arbeit aufgefasst wird.

Der Begriff des konzeptuellen Modells¹⁵ oder der konzeptuellen Modellierung (engl. *conceptual modeling*) wurde ursprünglich von CHEN [Ch76] geprägt, dem Begründer der Entity-Relationship (ER)-Modellierung (siehe auch Abschnitt 2.2.3) aus dem Gebiet der Datenmodellierung. Für eine Einordnung in einen größeren Gesamtzusammenhang ist das ARIS-Konzept von SCHEER [Sc98] geeignet. Dieses bildet einen Rahmen zur Vermittlung zwischen einer betriebswirtschaftlichen Problemstellung und deren informationstechnischer Umsetzung entlang verschiedener Stufen (siehe Abb. 2.5). In der ersten Stufe des Fachkonzepts werden betriebliche Fragestellungen berücksichtigt, die in der zweiten Stufe in ein DV-Konzept übertragen werden und in der dritten Stufe in einer Implementierung münden. Nach dieser Einteilung lassen sich konzeptuelle Modelle (z. B. ER-Diagramm) auf der obersten Stufe des Fachkonzepts ansiedeln. Sie dienen mitunter als Grundlage für die Überführung in ein DV-Konzept (z. B. Datenbankschema) und die anschließende technische Implementierung (z. B. in einem Datenbank-Managementsystem). Somit wird eine Brücke

¹⁵ Nach HESSE und MAYR [HM08] ist der Begriff konzeptuelles (d.h. ein Konzept aufweisendes) Modell einzusetzen, der Begriff konzeptionelles (d.h. ein Konzept betreffendes) Modell wird meist fälschlich als Synonym verwendet. Die synonyme Verwendung dieser Begriffe »ohne erkennbare Bedeutungsverschiebung« wird auch von FRANK [Fr94, S. 75] beobachtet. Aus Gründen der Einheitlichkeit wird im weiteren Verlauf stets der Begriff konzeptuelles Modell bzw. konzeptuelle Modellierung verwendet.

2. Grundlagen

zwischen Management oder Endanwendern auf der einen und Systementwicklern bzw. IT-Experten auf der anderen Seite geschaffen. Diese Brückenfunktion wird häufig als zentrales Anliegen der konzeptuellen Modellierung aufgeführt (siehe z. B. [Du13; Sc98]).

Die beschriebene Einordnung von konzeptuellen Modellen auf der Ebene des Fachkonzepts deckt sich mit weiteren Arbeiten. THALHEIM [Th12] bezeichnet konzeptuelle Modelle als »intermediate artifact for system construction«, laut BECKER und PFEIFFER [BP06, S. 6] beschäftigt sich die konzeptuelle Modellierung mit der »Darstellung eines Realweltausschnitts und ist im Kontext der Softwareentwicklung in die Fachkonzeptphase eingebunden«. FRANK [Fr94, S. 13, S. 211] beschreibt konzeptuelle Modelle als Hilfsmittel zur »anschaulichen Beschreibung der Unternehmensrealität« unter »Abstraktion von technischen Gegebenheiten (wie Hardware oder Programmiersprachen«. Im Hinblick auf diese Ausführungen lassen sich für ein konzeptuelles Modell alle drei Hauptmerkmale eines Modells nach STACHOWIAK [St73] ausmachen und konkretisieren:

- **Abbildungsmerkmal** (gemäß Def. 2.7): Der Abbildungsgegenstand konzeptueller Modelle ist ein bestehender oder geplanter Wirklichkeitsausschnitt (Ist- bzw. Soll-Zustand). Dabei wird zumeist Bezug auf die Entwicklung oder Konfiguration eines Software- bzw. Informationssystems genommen.
- **Verkürzungsmerkmal** (gemäß Def. 2.8): Bei der Erstellung eines konzeptuellen Modells werden nur für den konkreten Verwendungszweck relevante Aspekte abgebildet. Beispielsweise ergibt sich aus der Einordnung auf der Ebene des Fachkonzepts die Vernachlässigung implementationsrelevanter Sachverhalte.
- **Pragmatisches Merkmal** (gemäß Def. 2.9): Ein konzeptuelles Modell sollte hinsichtlich seiner Funktion bzw. seines Verwendungszwecks gestaltet werden. Der Einsatz zur Kommunikation auf der Ebene des Fachkonzepts impliziert z. B. die Verwendung einer grafischen Notation für die Darstellung oder eine möglichst gute Verständlichkeit des Modells innerhalb des intendierten Nutzerkreises.

Grafische Modelle als Artefakte der grafischen Modellierung können als Unterklasse konzeptueller Modelle aufgefasst werden, die eine grafische Repräsentation aufweisen. Ergänzend können Unterklassen gemäß dem betreffenden Anwendungsbereich gebildet werden. Bei Geschäftsprozessmodellen handelt es sich um eine Unterklasse konzeptueller Modelle, die eine prozessorientierte Sicht ermöglichen (siehe z. B.

[Fr14; Th06]). Eine weitere Unterklasse bilden konzeptuelle Datenmodelle wie ER-Diagramme [Ch76]. Eine Einschränkung auf konzeptuelle Modelle und der damit einhergehende Fokus auf die Ebene des Fachkonzepts ist im Hinblick auf die technische Umsetzung der E-Assessment-Plattform bezüglich der Forschungsfrage FF2 dieser Arbeit nicht nötig. Jedoch wird die wissenschaftliche Literatur zur konzeptuellen Modellierung im Rahmen von Forschungsfrage FF1 bei der Entwicklung des Kompetenzmodells und des Aufgabenkatalogs mitberücksichtigt.

Um die relevanten Aspekte entlang des Modellbegriffs im Rahmen dieser Arbeit präzise und differenziert behandeln zu können, soll das sogenannte FRISCO-Framework¹⁶ (A **F**ramework of **I**nformation **S**ystem **C**oncepts) [Fa98] als Basis für die folgenden Begriffsdefinitionen dienen. So wird im FRISCO-Framework zunächst einmal zwischen einem Modell (Def. 2.10) als interner, d.h. gedanklicher Konzeption¹⁷ selbst sowie seiner externen Repräsentation in Form einer sogenannten Modelldenotation (Def. 2.11) unterschieden¹⁸.

Definition 2.10: Modell

Ein **Modell** ist eine zweckgerichtet abstrahierte, klare, präzise und eindeutige Konzeption. (Übersetzt nach [Fa98, S. 55].)

Definition 2.11: Modelldenotation

Eine **Modelldenotation** ist eine präzise und eindeutige Repräsentation eines Modells in einer angemessenen formalen oder semi-formalen Modellierungssprache. (Übersetzt nach [Fa98, S. 55].)

Somit kann ein bestimmtes Modell in unterschiedlichen Sprachen auf unterschiedliche Weise beschrieben werden, eine bestimmte Modelldenotation wird jedoch in genau einer Sprache ausgedrückt [Fa98, S. 56]¹⁹. Die Modellierungssprache selbst

¹⁶ Bereits im Jahr 1987 wurde eine Arbeitsgruppe der IFIP (International Federation for Information Processing) ins Leben gerufen, welche nach drei themenbezogenen Konferenzen und einem Zwischenbericht schließlich 11 Jahre später das FRISCO-Framework veröffentlichte. Das Ziel war es, einfache und präzise Definitionen sowie eine umfassende Terminologie auf dem Gebiet der Informationssysteme zu etablieren.

¹⁷ In [Fa98, S. 47] wird die Konzeption als gedankliche Auffassung einer bestimmten Domäne definiert, siehe auch die spätere Beschreibung anhand des semiotischen Tetraeders in Abb. 2.8.

¹⁸ Im Forschungskontext dieser Arbeit wird der Begriff Modell in aller Regel synonym zum Begriff Modelldenotation verwendet, wie auch in [Fa98, S. 55] festgestellt. Das bedeutet zugleich, dass diese Unterscheidung zwischen interner und externer Ausprägung zumeist nicht getroffen wird.

¹⁹ Es sei anzumerken, dass diese Definition einer Modelldenotation eine sehr optimistische ist. In der späteren Auseinandersetzung mit dem Begriff der Modellqualität (siehe Abschnitt 3.2) wird deutlich, dass weder die Merkmale »präzise und eindeutig« noch die Wahl einer angemessenen Sprache für die

2. Grundlagen

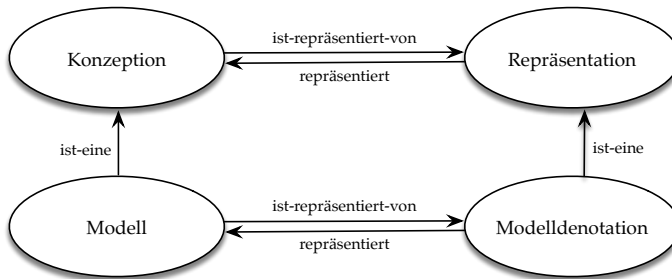


Abb. 2.6.: Beziehungen zwischen Modell, Modelldenotation, Konzeption und Repräsentation in Anlehnung an [Fa98, S. 56]

kann anhand eines sogenannten Meta-Modells spezifiziert werden [Fa98, S. 58]²⁰.

In Abb. 2.6 werden die Beziehungen zwischen den zentralen Begriffen aus den beiden Definitionen 2.10 und 2.11 dargestellt.

Weiterhin wird unter dem Begriff Modellierung gemeinhin der Vorgang verstanden, mit welchem man von einer Konzeption bezüglich realweltlicher Sachverhalte zu einer konkreten Repräsentation gelangt. Im FRISCO-Framework wird die Modellierung (Def. 2.12) auf Grundlage der vorangehenden Definitionen als Modellierungsprozess durch eine modellierende Person (Def. 2.13) definiert.

Definition 2.12: Modellierung

Der Prozess der **Modellierung** ist eine Sequenz bestehend aus dem **Wahrnehmen** einer Domäne, gefolgt vom **Begreifen** dieser Wahrnehmung in Form von einem Modell, gefolgt vom **Abbilden** dieses Modells in Form einer Modelldenotation. (Übersetzt nach [Fa98, S. 56].)

Definition 2.13: Modellierende Person

Bei der **modellierenden Person** handelt es sich um einen menschlichen Akteur, der den Modellierungsprozess durchführt. (Übersetzt nach [Fa98, S. 56].)

Abb. 2.7 veranschaulicht die Sequenz der drei Schritte (i)-(iii), die eine modellierende Person gemäß Definition 2.12 während der Modellierung ausführt. Mit Bezug auf

Repräsentation eines Modells selbstverständlich sind.

²⁰ In der Literatur wird die Modellierungssprache mitunter als Modell bezeichnet, z. B. das ER-Modell von CHEN [Ch76].



Abb. 2.7.: Übersicht über die einzelnen Schritte der Modellierung gemäß Definition 2.12. Interne Elemente (grau) entsprechen hierbei dem kognitiven Anteil, welcher sich im Kopf der modellierenden Person abspielt. Externe Elemente (weiß) sind die eingangs betrachtete Domäne sowie die Modelldenotation als finales Ergebnis.

die Problemstellung dieser Dissertation ist diese verkettete Darstellung nützlich, um auf die verschiedenen grundsätzlich möglichen Ursachen einer fehlerhaften Modelldenotation einzugehen:

- (i) Eine gestörte Wahrnehmung der Domäne liegt vor, wenn der betrachtete Realitätsausschnitt durch die menschlichen Sinne nicht korrekt erfasst wird, z. B. wenn die modellierende Person in Interviews zur Prozessaufnahme Dinge akustisch falsch versteht oder eine schriftlich vorliegende Dokumentation nur schwer entziffern kann.
- (ii) Um die Wahrnehmung in ein korrektes Modell zu überführen, ist die korrekte Interpretation des Wahrgenommenen erforderlich. Fehler treten hierbei z. B. dann auf, wenn die modellierende Person nicht über das nötige Hintergrundwissen zur betrachteten Domäne verfügt oder Konflikte zwischen widersprüchlichen Schilderungen nicht berücksichtigt und entsprechend auflöst.
- (iii) Zur Erstellung einer Modelldenotation wird das Modell mit Hilfe einer formalen oder semi-formalen Sprache übersetzt. Diesbezüglich ist zum einen die Auswahl einer geeigneten Sprache entscheidend, zum anderen der Grad zu welchem der Modellierer die gewählte Sprache beherrscht. Verfügt z. B. die gewählte Sprache nicht über die nötigen Sprachkonstrukte, um alle Inhalte des Modells auszudrücken, ist die resultierende Modelldenotation zwangsläufig unvollständig und einer modellierenden Person mit wenig Erfahrung können bei der Erstellung der Modelldenotation Fehler z. B. aufgrund Verletzungen der sprachlichen Syntaxregeln unterlaufen.

Die Beziehungen zwischen den Begriffen Konzeption, Repräsentation und Domäne werden im FRISCO-Framework [Fa98, S. 51] mit Hilfe des semiotischen Tetraeders aus den Sprachwissenschaften klassifiziert (siehe Abb. 2.8). Die Semiotik als Wissenschaft der Zeichen befasst sich mit sprachlichen oder nichtsprachlichen Zeichensystemen unterschiedlicher Natur, z. B. Schrift oder Musik. Ursprünge des semiotischen

2. Grundlagen

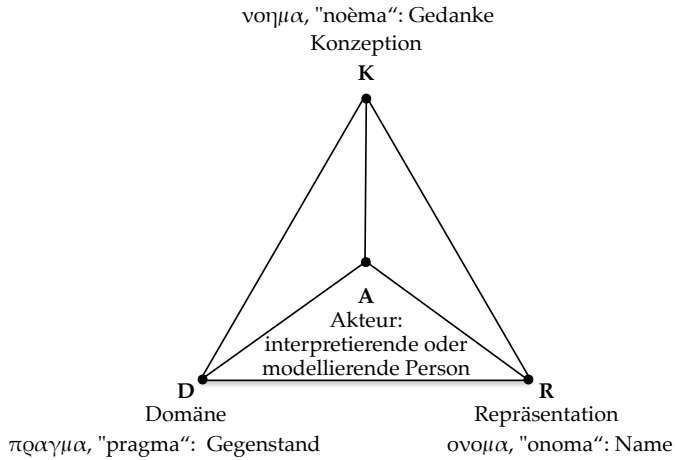


Abb. 2.8.: Semiotischer Tetraeder nach FRISCO, in Anlehnung an [Fa98, S. 51]

Dreiecks finden sich bereits bei Aristoteles (384-322 v. Chr.), der zwischen Gegenstand, dem Gedanken an diesen und dem dafür verwendeten Namen differenzierte. Charles Sanders Peirce (1839-1914), der als Begründer der modernen Semiotik gilt, erweiterte dieses Konzept um einen menschlichen Akteur, die interpretierende Person, der über eine Repräsentation zu einer Konzeption und der damit verbundenen Domäne gelangt. Im FRISCO-Framework wird analog zur interpretierenden Person die modellierende Person eingeführt, die eine Repräsentation erstellt. Diese vermittelt ebenfalls – jedoch andersgerichtet – über die drei Bereiche hinweg. Die Unterscheidung zwischen diesen beiden Rollen ist im Kontext der konzeptuellen – und somit auch der grafischen – Modellierung besonders relevant, da viele Anwendungsszenarien in den betrachteten Anwendungsbereichen eine entsprechende Trennung vorsehen: Eine Person mit Expertenwissen liefert Informationen über die Domäne an eine modellierende Person, welche diese Informationen in eine Modelldenotation übersetzt. Die fertige Modelldenotation wiederum steht dann je nach Anwendungszweck einem erweiterten Personenkreis zur Verfügung, z. B. einer Mitarbeiterin im Unternehmen, die einen Prozess wie dokumentiert ausführen soll. Um hierbei die korrekte Interpretation der Modelldenotation zu unterstützen, gilt z.B. die Verständlichkeit als wesentliches Qualitätskriterium für die konzeptuelle Modellierung (siehe auch die weitere Ausführung in Abschnitt 3.2).

Der nun folgende Abschnitt, der sich vertieft mit dem hier als Repräsentation eingeführten Begriff befasst, berührt weiterhin das Feld der Semiotik.

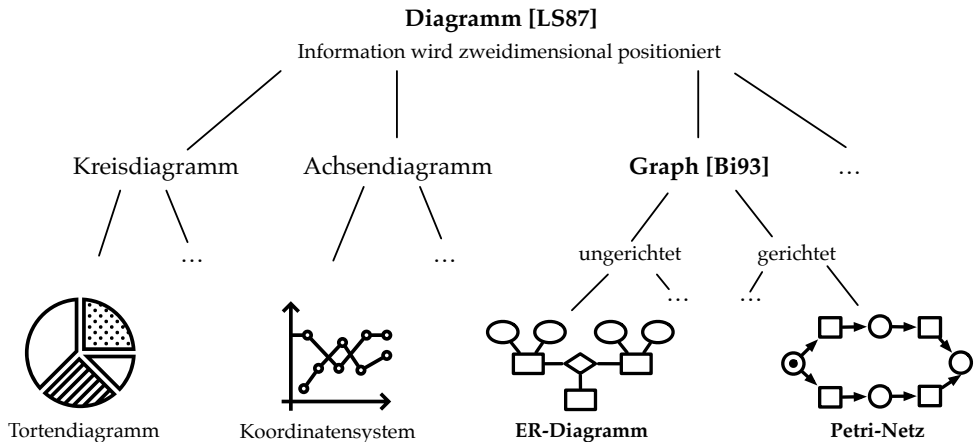


Abb. 2.9.: Einordnung eines ER-Diagramms und eines Petri-Netzes in Bezug zu den Begriffen Diagramm und Graph. Es handelt sich demnach um graphbasierte, grafische Modelle, kurz: graphbasierte Diagramme.

2.2.3. Modellierungssprachen

In dem hier bedeutsamen Kontext ist eine Modellierungssprache eine künstliche und (semi)-formale Sprache, die zum Zweck der Modellierung eingeführt wird. In der Terminologie des vorangegangenen Abschnittes dient eine Modellierungssprache also zur Repräsentation eines Modellierungsgegenstands einer bestimmten Domäne in Form einer Modelldenotation. In dieser Arbeit werden sogenannte grafische Modellierungssprachen betrachtet, bei denen sich die Modelldenotation aus grafischen Symbolen zusammensetzt²¹. In der wissenschaftlichen Literatur auf dem Gebiet der grafischen Modellierung spielen auch die Begriffe Diagramm und Graph eine Rolle. An dieser Stelle soll daher eine kurze Einführung und anschließende Einordnung der betrachteten grafischen Modelle (bzw. Modelldenotationen) mit Bezug zu diesen beiden Begriffen erfolgen, wie es auch in Abb. 2.9 dargestellt wird.

Definition 2.14: Diagramm

Ein Diagramm ist eine Datenstruktur, in der Information zweidimensional positioniert wird. (Übersetzt und angelehnt an [LS87].)

²¹ Für nicht-grafische Sprachen dienen nicht nur Programmiersprachen als Beispiel. Auch bei Modellierungssprachen gibt es verschiedene nicht-grafische Sprachen, wie z. B. die XML-basierte WS-Business Process Execution Language (BPEL), bei welcher der Schwerpunkt auf der Ausführbarkeit durch ein Informationssystem liegt.

2. Grundlagen

Gemäß dieser Definition nach LARKIN und SIMON [LS87] kann ein Diagramm – im Gegensatz zu eindimensionalen (linearen) textuellen Repräsentationsformen – explizit topologische und geometrische Beziehungen zwischen den dargestellten Elementen wiedergeben. Mit Diagrammen lässt sich eine Reihe ganz unterschiedlicher Konzepte repräsentieren, z. B. die größengerechte Darstellung von Anteilen anhand eines Kreisdiagramms oder der Verlauf von Messwerten über die Zeit in einem Koordinatensystem bzw. Achsendiagramm (siehe Abb. 2.9). Auch bei den in dieser Arbeit behandelten Modelldenotationen bzw. grafischen Modellen werden Elemente grafisch in einer zweidimensionalen Struktur abgebildet. Daraus folgt, dass es sich bei ihnen um eine Teilmenge von Diagrammen handelt. Gelegentlich wird der Begriff Diagramm auch explizit für die Artefakte der grafischen Modellierung verwendet, beispielsweise bei ER-Diagrammen oder UML-Klassendiagrammen.

Als erste Arbeit auf dem Gebiet der Graphen wird das Königsberger Brückenproblem von Euler (1736) gezählt, heute werden Graphen in vielen verschiedenen Disziplinen eingesetzt. Das mathematische Fundament bildet die Graphentheorie.

Definition 2.15: Graph

Ein Graph ist ein Tripel (V, E, I) , für welches gilt:

- (i) V ist die Menge von Knoten;
- (ii) E ist die Menge von Kanten;
- (iii) $I \subseteq (V \times E)$ ist eine Inzidenzrelation.

(Übersetzt und angelehnt an [Bi93, S. 3f].)

Nach dieser Definition von BIGGS [Bi93] lassen sich mit Graphen grundlegend Beziehungen zwischen Elementen beschreiben. Wird die Menge von Kanten der Inzidenzrelation nach als geordnetes Paar aufgefasst (d.h. $(a, b) \neq (b, a)$), so handelt es sich um einen gerichteten Graphen. Bei der grafischen Repräsentation von Graphen werden deren Elemente typischerweise ebenfalls zweidimensional positioniert²². Somit bilden Graphen wiederum eine Teilmenge von Diagrammen.

In Abb. 2.9 werden zwei Modelldenotationen der in dieser Arbeit betrachteten grafischen Modellierungssprachen zu den Begriffen Diagramm und Graph in Bezug gesetzt. Bei beiden Modelldenotationen handelt es sich um Diagramme, da sie anhand einer zweidimensionalen Darstellung repräsentiert werden. Weiterhin weist das

²² Graphen müssen nicht zwingend grafisch dargestellt werden. Sie lassen sich auch rein formal (textuell) beschreiben (siehe z. B. die Angabe der Petri-Netz-Mengen S , T und F in Abschnitt 2.2.4). Eine derartige Beschreibung entspricht *nicht* der hier verwendeten Definition eines Diagramms.

ER-Diagramm die Struktur eines ungerichteten Graphen auf, wohingegen die Repräsentation der zeitlichen Aufeinanderfolge von Aktivitäten in einem Geschäftsprozess durch das Petri-Netz auf Basis eines gerichteten Graphen erfolgt. In beiden Fällen sind die zuvor beschriebenen Eigenschaften eines Diagramms bzw. eines Graphen erfüllt. Es handelt sich bei ihnen demnach um graphbasierte, grafische Modelle, kurz: graphbasierte Diagramme.

Im Kontext der grafischen Modellierung setzt sich eine (semi-)formale Modellierungssprache aus drei verschiedenen Bestandteilen zusammen: Syntax, Semantik und Notation²³. Dabei werden neben Modellierungselementen und deren zulässiger Komposition (Syntax) auch deren Bedeutung und Entsprechung im Kontext des Modellierungsgegenstands (Semantik) und deren Darstellungsweise (Notation) festgelegt. Der Modellierungsgegenstand muss dabei nicht zwingend real existieren (Modelldenotation als Abbild des Gegenstands), er kann auch für die Entwicklung vorgesehen sein (Modelldenotation als Vorbild für den Gegenstand). Abb. 2.10 – anhand der gestrichelten Linie in eine obere und eine untere Hälfte geteilt – stellt einen Überblick über die Beziehungen zwischen einer Modellierungssprache und ihren drei Bestandteilen dar. Die obere Hälfte zeigt einen Ausschnitt der Darstellung von KARAGIANNIS und KÜHN [KK02]. Da die in dieser Arbeit betrachteten grafischen Modellierungssprachen ausschließlich graphbasiert sind (gemäß Def. 2.15), wurde der Bestandteil Syntax in der unteren Hälfte zweckorientiert erweitert. Dies ermöglicht eine differenzierte Betrachtung der Modellierungselemente.

Den Ausgangspunkt für die folgenden Beschreibungen der ersten beiden Bestandteile Syntax und Semantik bilden die Arbeiten von HAREL und RUMPE [HR00; HR04], die sich mit der ordnungsgemäßen Definition von Modellierungssprachen für Systeme und Software auseinandergesetzt haben. Weiterhin fließen die Ausführungen von FRANK und PRASSE [FP97] und KARAGIANNIS und KÜHN [KK02] zur Beschreibung des Bestandteils Notation in Bezug auf grafische Modellierungssprachen mit ein.

Syntax: Die Syntax einer Modellierungssprache setzt sich aus einer Menge verfügbarer Modellierungselemente A (Alphabet), sowie einer Menge von Regeln \mathcal{R} zu deren zulässigen Anordnung zusammen. Nach der in Abb. 2.10 dargestellten Erweiterung wird die Menge A in drei Teilmengen zerlegt: $A = V \cup E \cup S$.

²³ In Arbeiten auf den Gebieten des Software Engineering und der IS-Forschung (siehe z. B. [FL03; FP97; Wi97]) wird mitunter auf die Notation verzichtet und stattdessen zwischen abstrakter und konkreter Syntax unterschieden. Die abstrakte Syntax entspricht nach dieser Auffassung dem ersten, hier als Syntax bezeichneten, die konkrete Syntax dem dritten, hier als Notation bezeichneten Bestandteil. HAREL und RUMPE [HR00; HR04] fassen die Sprachelemente der Syntax grundsätzlich als konkrete Symbolmenge auf und setzen die Syntax somit gänzlich mit der Notation gleich.

2. Grundlagen

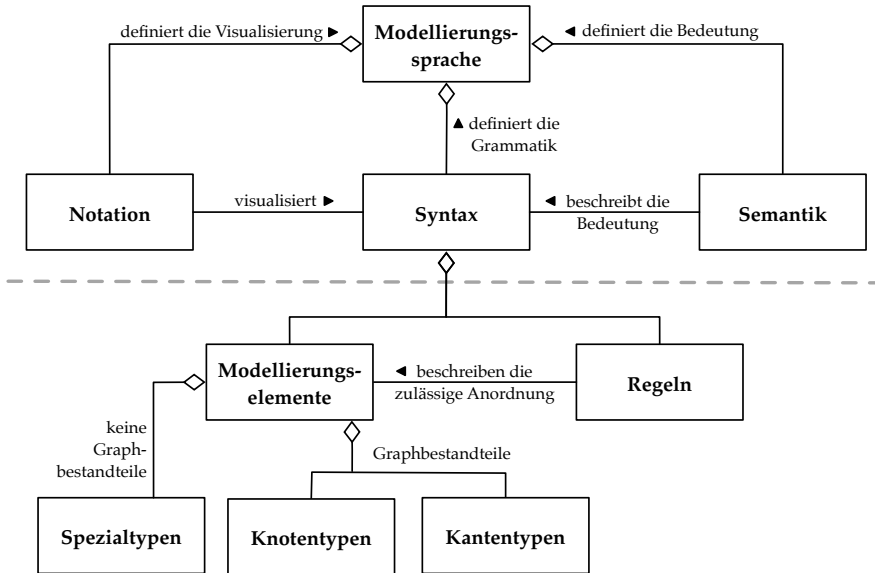


Abb. 2.10.: Aufbau und Bestandteile einer Modellierungssprache. Obere Hälfte in Anlehnung an [KK02], untere Hälfte eigene Erweiterung zur differenzierten Betrachtung von graphbasierten Modellierungssprachen.

Dadurch wird zwischen Knotentypen V sowie Kantentypen E als Graphbestandteilen und sogenannten Spezialtypen S unterschieden²⁴. Für die formale Beschreibung der Syntax können z. B. Grammatiken zum Einsatz kommen.

Semantik: Um die Bedeutung der Modellierungselemente in A festzulegen, werden diese anhand einer Abbildungsfunktion $sem : A \rightarrow S$ auf die sogenannte semantische Domäne S abgebildet. Eine Beschreibung kann formal durch Semantiken (gemeint sind Methoden der Informatik) erfolgen, häufig wird dafür aber auch (informal) die natürliche Sprache eingesetzt.

Notation: Für die Zwecke der Repräsentation wird anhand der Notation das konkrete Aussehen der Modellierungselemente in A anhand einer Abbildungsfunktion festgelegt: $not : A \rightarrow \mathcal{N}$. Im Kontext von \mathcal{N} sind theoretisch alle Darstellungsformen zulässig, die für die interpretierenden Personen zugänglich sind²⁵.

²⁴ Spezialtypen sind bei einigen Modellierungssprachen notwendig, um Modellierungselemente zu erfassen, die nicht Bestandteil eines Graphen sind, aber in der Modellierungssprache eingesetzt werden, um das Vorliegen individueller Eigenschaften oder eine Gruppierung von Modellelementen darzustellen (z. B. UML-Aktivitäten oder BPMN-Pools).

²⁵ Außerhalb des Kontexts grafischer Modellierungssprachen müssen bei der Definition der Notation nicht zwingend grafische Symbole (z. B. Formen oder Icons) zum Einsatz kommen, auch die Festlegung

In einer Modelldenotation konkret zum Einsatz kommende Modellierungselemente einer Modellierungssprache werden **Modellelemente** genannt. Um anhand einer Modelldenotation eine Repräsentation des Modellierungsgegenstands zu erzeugen, werden die in der Modelldenotation enthaltenen Modellelemente in aller Regel zusätzlich mit einer natürlichsprachlichen Beschriftung²⁶ versehen. Anhand der Beschriftung kann die jeweilige Entsprechung im Kontext des Modellierungsgegenstands definiert und bei mehrfachem Auftreten von Modellelementen desselben Typs eines Modellierungselements eine Unterscheidung ermöglicht werden (siehe auch [Mo09]). Formal lässt sich dies über eine Beschriftungsfunktion $label : E \rightarrow C$ beschreiben, anhand welcher die Elemente E in einer Modelldenotation auf eine Zeichenkette $C \subseteq \Sigma^*$ ²⁷ abgebildet werden können.

Die Ausdrucksmächtigkeit einer Modellierungssprache wird durch die Festlegung von Syntax und Semantik bestimmt, wohingegen die – grundsätzlich austauschbare – Notation auf diese keinerlei Einfluss besitzt. Die im Kontext einer Modellierungssprache festgelegte Notation wird auch als **primäre Notation** bezeichnet. Beispielsweise erfordert die Notation von ER-Diagrammen die Darstellung von Entitätstypen als Rechteck und Beziehungstypen als Raute (siehe auch den folgenden Unterabschnitt). Davon lässt sich die sogenannte **sekundäre Notation** abgrenzen, die neben der Form weitere visuelle Eigenschaften wie z.B. die Größe, Farbe oder Textur berücksichtigt. Nach PETRE [Pe06] umfasst die sekundäre Notation genau jene Aspekte, die nicht formaler Bestandteil einer Notation sind, aber dennoch interpretiert werden können. Auch die meist nicht durch Regeln bestimmte räumliche Anordnung der Modellelemente in einer Modelldenotation fällt in diesen Bereich.

In den folgenden Unterabschnitten werden bedeutsame Vertreter grafischer Modellierungssprachen der in dieser Arbeit gewählten Anwendungsbereiche vorgestellt. Dies sind Entity-Relationship (ER)-Diagramme für die konzeptuelle Datenmodellierung sowie die Unified Modeling Language (UML) im Kontext der Softwareentwicklung. Zwei weitere Sprachen stammen aus dem Anwendungsbereich der Geschäftsprozessmodellierung: Ereignisgesteuerte Prozessketten (EPK) sowie Business Process Model and Notation (BPMN). Für die Darstellung dieser Modellierungssprachen dient die in diesem Abschnitt entwickelte formale Beschreibung als Grundlage.

von Begriffen (z. B. Befehle einer Programmiersprache) ist möglich.

²⁶ In anderen Arbeiten wird die Beschriftung auch als Inschrift [Re10] oder Anschrift [Ba90] bezeichnet, im englischen aber teilweise auch im deutschen Sprachraum ist die Bezeichnung *label* üblich.

²⁷ Die Kleenesche Hülle Σ^* über einem Alphabet Σ ist definiert als die Menge aller Zeichenketten, die durch beliebige Konkatenation von Zeichen des Alphabets Σ gebildet werden können. Beispielsweise kann als Alphabet Σ für die Zeichenkette C der globale UTF-8-Zeichensatz zum Einsatz kommen, so dass damit natürlichsprachliche Beschriftungen ermöglicht werden.

2. Grundlagen

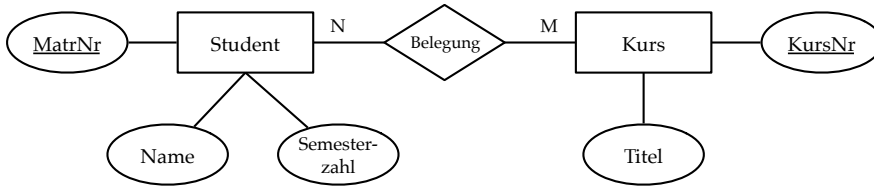


Abb. 2.11.: Beispiel für ein ER-Diagramm, welches die Beziehungen zwischen Studenten und Kursen beschreibt.

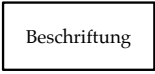

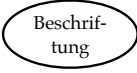
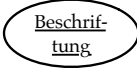

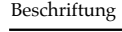
Entity-Relationship-Modell (ER-Modell)

CHEN [Ch76] ist Begründer des Entity-Relationship-Modells für den konzeptuellen Datenbankentwurf. Die grafische Darstellung in Form von ER-Diagrammen (auch ERD genannt) ist weit verbreitet [Da06]. Nach CHEN und KNÖLL [CK91, S. 35f]²⁸ werden bei der ER-Modellierung zwei wesentliche Strukturierungskonzepte betrachtet: Entitäten und Beziehungen. Eine Entität (oder Objekt) ist eine eindeutig identifizierbare Sache aus der realen Welt, die durch Beziehungen mit anderen Entitäten in Verbindung steht. Über die sogenannten Entitäts- bzw. Beziehungstypen²⁹ werden Eigenschaften, wie zugehörige Attribute oder Beziehungskomplexitäten (z. B. eins-zu-viele - 1:N, viele-zu-viele - N:M) definiert. Im Wesentlichen besteht ein ER-Diagramm aus der Darstellung von Entitätstypen (Rechteck) und Beziehungstypen (Raute), die über Kanten miteinander verbunden sind. Die Zuordnung von Attributen (Oval) erfolgt ebenfalls anhand von Kanten. Somit handelt es sich bei ER-Diagrammen um ungerichtete Graphen. Über die vergangenen Jahrzehnte hinweg wurden für die grafische Darstellung von ER-Diagrammen variierende Notationen entwickelt. Darüber hinaus existieren zahlreiche Erweiterungen der ER-Modellierung, welche zusätzliche Sprachkonzepte einführen, wie z. B. mengenwertige Attribute, Generalisierungs- oder Aggregationskonzepte (beschrieben z. B. in [La05, S. 113f]). Im Rahmen dieser Arbeit genügt jedoch die Betrachtung der zentralen Modellierungselemente der klassischen ER-Modellierung nach CHEN [Ch76]. Diese sind in Tab. 2.4 zusammengefasst. Das Beispiel in Abb. 2.11 zeigt die zwei Entitätstypen *Student* und *Kurs*, über welchen der Beziehungstyp *Belegung* definiert ist. Sie besitzen jeweils ein Schlüsselattribut *MatrNr* bzw. *KursNr* zur eindeutigen Identifizierung, sowie weitere Attribute, wie z. B. *Name* oder *Titel*. Anhand der

²⁸ Lizenzierte deutsche Übersetzung von Peter P.S. Chen: »The Entity-Relationship-Approach to Logical Database Design« (1977), erschienen bei Q.E.D. Information Sciences, Wellesley, MA, U.S.A.

²⁹ (Konkrete) Entitäten bzw. Beziehungen können zu einem (abstrakten) Entitäts- bzw. Beziehungstyp klassifiziert werden.

Tab. 2.4.: Entity-Relationship-Modellierung (ER): Zentrale Modellierungselemente

Knotentyp $v \in V$	Semantik $sem(v)$	Notation $not(v)$
Entitätstyp	Wohlunterscheidbares physisch oder gedanklich existierendes Objekt	
Beziehungstyp	Beschreibt die Art der Beziehung zwischen zwei zugeordneten Entitätstypen	
Attribut	Interessierende Eigenschaft des zugeordneten Entitäts- oder Beziehungstyps	
Schlüsselattribut	Eindeutige Identifikation der zugeordneten Entität über dieses Attribut möglich	
Kantentyp $e \in E$	Semantik $sem(e)$	Notation $not(e)$
Unbeschriftete Kante	Zuordnung von zwei Elementtypen	
Beschriftete Kante	Definiert die Häufigkeit des Auftretens der zugeordneten Entitätstypen über einen Beziehungstyp	

Kantenbeschriftungen zwischen Beziehungs- und Entitätstypen werden die Beziehungskomplexitäten (auch Kardinalitäten genannt) beschrieben. Im Beispiel wird durch die Beziehungskomplexität N:M zwischen *Student* und *Kurs* ausgedrückt, dass ein Student mehrere verschiedene Kurse belegen kann. Ebenso kann ein Kurs durch mehrere verschiedene Studierende belegt werden.

Unified Modeling Language (UML)

Die Unified Modeling Language (UML) entstand Mitte der 1990er-Jahre im Zusammenhang mit dem Aufkommen der objektorientierten Softwareentwicklung. Im Jahr 1997 wurde die Sprache von der Object Management Group (OMG) als Standard angenommen und liegt seit 2017 in der aktuellen Spezifikation (Version 2.5.1) vor [Ob17]. Wie in Abb. 2.12 dargestellt, können mit UML verschiedenartige Diagramme erstellt werden, die sich auf der obersten Ebene zwei unterschiedlichen Klassen zuordnen lassen, nämlich Strukturdiagrammen und Verhaltensdiagrammen. Strukturdiagramme repräsentieren zeitunabhängig die statische Struktur von Objekten in einem System. Dahingegen können Verhaltensdiagramme das dynamische – d.h.

2. Grundlagen

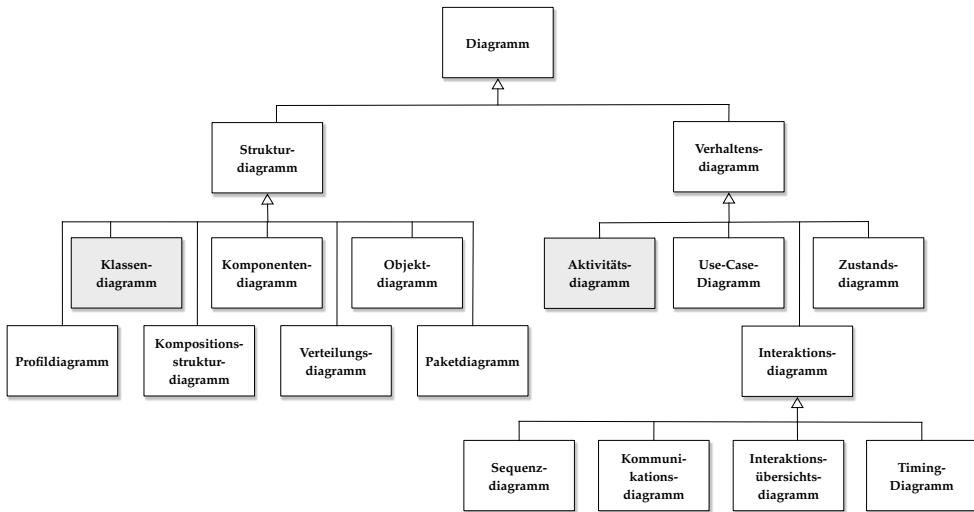


Abb. 2.12.: Übersicht über die UML-Diagrammtypen [Ob17, S. 685]

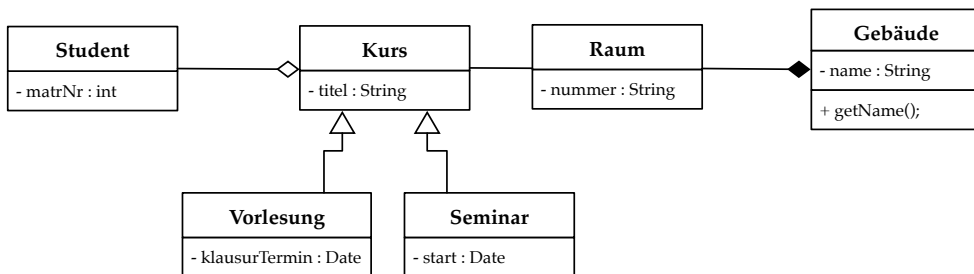


Abb. 2.13.: Beispiel für ein UML-Klassendiagramm zur Kursverwaltung im Hochschulkontext

über die Zeit veränderte – Verhalten von Objekten in einem System wiedergeben. Von den insgesamt 14 UML-Diagrammtypen soll an dieser Stelle jeweils ein Vertreter der beiden Klassen herausgegriffen und grob betrachtet werden: das Klassen- und das Aktivitätsdiagramm.

UML-Klassendiagramme (UML-KD) aus der Klasse der Strukturdiagramme dienen als Hilfsmittel, um im Rahmen der objektorientierten Softwareentwicklung die eingesetzten Klassen, Schnittstellen und ihre verschiedenartigen Beziehungen zueinander (z. B. Abstraktion und Vererbung) grafisch darzustellen. Aufgrund der Vielzahl an unterschiedlichen Beziehungsarten sind in der Übersichtstabelle 2.5 nur die im folgenden Beispiel in Abb. 2.13 verwendeten Modellierungselemente aufgeführt.

Tab. 2.5.: UML-Klassendiagramm: Zentrale Modellierungselemente

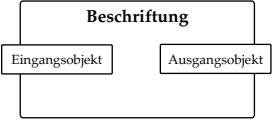
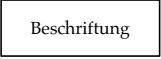
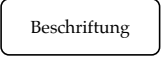



Knotentyp $v \in V$	Semantik $sem(v)$	Notation $not(v)$			
Klasse	Wohlunterscheidbares physisch oder gedanklich existierendes Objekt mit optionaler Angabe von Attributen und Methoden	<table border="1"> <tr> <td>Beschriftung</td> </tr> <tr> <td>Attributliste</td> </tr> <tr> <td>Methodenliste</td> </tr> </table>	Beschriftung	Attributliste	Methodenliste
Beschriftung					
Attributliste					
Methodenliste					
Kantentyp $e \in E$	Semantik $sem(e)$	Notation $not(e)$			
Assoziationskante	Unspezifizierte Beziehung zwischen zwei Klassen	————			
Aggregationskante	Oberklasse besitzt eine Liste von Objekten der Unterklasse (Kante zeigt zur Oberklasse)	————◇			
Kompositionskante	Oberklasse setzt sich aus Objekten der Unterklasse zusammen (Kante zeigt zur Oberklasse)	————◆			
Vererbungskante	Ober- und Unterklasse stehen in einer Vererbungshierarchie zueinander (Kante zeigt zur Oberklasse)	————▷			

Die Klassen `Kurs` und `Raum` sind durch eine unspezifizierte Assoziation miteinander verknüpft. Dahingegen sind die Assoziationen zwischen `Student` und `Kurs` (Aggregation) und `Raum` und `Gebäude` (Komposition) weitergehend spezifiziert. Studenten können zwar einem `Kurs` zugeordnet sein, ihre Existenz ist aber nicht an diesen gebunden, d.h. wird ein `Kurs` entfernt, bleiben die zugehörigen Studenten weiterhin bestehen. Anders verhält es sich bei der Zuordnung von `Raum` zu `Gebäude`. Ein `Raum` gehört immer zu einem bestimmten `Gebäude` und kann ohne dieses nicht existieren, d.h. wird ein `Gebäude` entfernt (sei es in dem durch das Modell dargestellte System oder in der Realität), sind auch die dazugehörigen Räume nicht mehr vorhanden. Das Vererbungskonzept wird durch die Darstellung der Klasse `Kurs` illustriert. Diese vererbt ihre Eigenschaften den beiden Klassen `Vorlesung` und `Seminar`, die diese um ihre jeweiligen speziellen Eigenschaften ergänzen. Im UML-Klassendiagramm können für Klassen zusätzliche Eigenschaften wie Attribute zusammen mit ihrem Datentyp (z. B. das Attribut `matrNr` vom Typ `int` für die Klasse `Student`) und Methoden (z. B. `getName()` für die Klasse `Gebäude`) festgelegt werden. Wie ER-Diagramme können UML-Klassendiagramme durch die Angabe von Beziehungskomplexitäten ergänzt werden.

UML-Aktivitätsdiagramme (UML-AD) aus der Klasse der Verhaltensdiagramme können neben der intendierten Beschreibung von Softwaresystemverhalten auch für die Beschreibung von Geschäftsprozesse eingesetzt werden. Die UML2.0-Spezifikation

2. Grundlagen

Tab. 2.6.: UML-Aktivitätsdiagramm: Zentrale Modellierungselemente

Spezialtyp $s \in S$	Semantik $sem(s)$	Notation $not(s)$
Aktivität	Übergeordnete Aktivität, deren Verhalten beschrieben wird (Verhaltensbeschreibung innerhalb des Rahmens, auf der Rahmenkante liegen optionale Ein- und Ausgabeobjekte der Aktivität)	
Knotentyp $v \in V$	Semantik $sem(v)$	Notation $not(v)$
Ein-/Ausgabe-knoten	Information/Datei oder reales Objekt, das als Ein- oder Ausgabe für Aktionen dient (Objektknoten)	
Aktion	Fundamentale Einheit, die Verhalten beschreibt (Ausführbarer Knoten)	
XOR-Split/Join	Exklusive Verzweigung bzw. Zusammenführung des Kontrollflusses (Kontrollknoten)	
AND-Split/Join	Nebenläufige Aufspaltung bzw. Synchronisation des Kontrollflusses (Kontrollknoten)	
Kantentyp $e \in E$	Semantik $sem(e)$	Notation $not(e)$
Kante	Kontrollfluss in Pfeilrichtung	

[Ob17, S. 373] definiert ein Aktivitätsdiagramm als einen Graphen, dessen Knoten in drei Klassen eingeteilt werden: Ausführbare Knoten, die untergeordnete Schritte der beschriebenen Gesamtaktivität benennen, Objektknoten, die Ein- und Ausgabe sowie den Datenfluss spezifizieren, sowie sogenannte Kontrollknoten, die den Kontrollfluss³⁰ entlang der ausführbaren Knoten bestimmen. Aufgrund des großen Umfangs und zahlreicher Erweiterungen sind in Tab. 2.6 nur die im Beispiel in Abb. 2.14 verwendeten Modellierungselemente aufgeführt. Das Beispiel stellt dar, wie die Klausurzulassung für einen bestimmten Kurs erreicht werden kann. Dazu müssen nach erfolgter Kursanmeldung eine Gruppenarbeit und ein Test absolviert werden. Für die Gruppenarbeit ist eine vorangehende Gruppenbildung erforderlich. Der Test wird bei Abwesenheit online, bei Anwesenheit vor Ort durchgeführt. Sind Gruppenarbeit und Test absolviert worden, wird die Klausurzulassung als *erreicht* im System vermerkt.

³⁰ Verschiedene Kontrollfluss-Muster für Geschäftsprozesse werden z. B. von der Workflow Patterns Initiative [Ru06] beschrieben und in Abschnitt 2.2.4 dieser Arbeit vorgestellt.

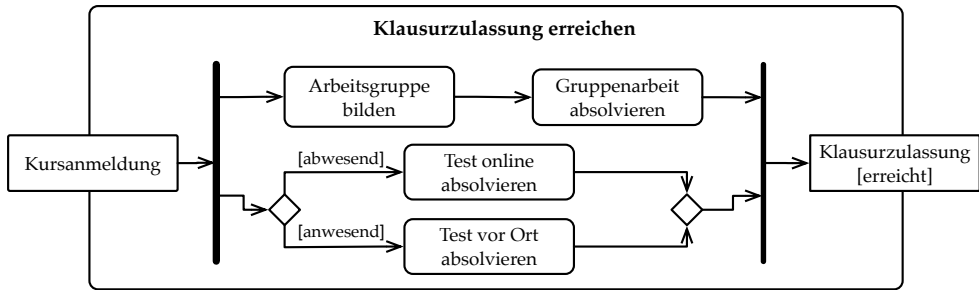


Abb. 2.14.: Beispiel für ein UML-Aktivitätsdiagramm zur Beschreibung des Erreichens einer Klausurzulassung

Ereignisgesteuerte Prozesskette (EPK)

Der Modellierungsansatz der Ereignisgesteuerten Prozesskette (EPK) [KNS92] für die Geschäftsprozessmodellierung entstand Anfang der 1990er-Jahre an der Universität des Saarlandes. EPK-Modelle sind in der Praxis unter anderem aufgrund der engen Zusammenarbeit mit dem großen Softwareunternehmen SAP (siehe z. B. [KLC99]) – insbesondere im deutschsprachigen Raum – weit verbreitet [Fe09]. Auch in der Forschung wird die EPK intensiv thematisiert, bislang existiert allerdings keine Standardisierung durch eine entsprechende Organisation [Fe13], obgleich verschiedene Ansätze für die formale Spezifikation von Syntax und Semantik vorgeschlagen wurden (z. B. [Ki06; NR02; Ri99]). EPK werden durch eine Abfolge von Ereignissen (passive Elemente) und Funktionen (aktive Elemente) beschrieben. Dabei legen Verknüpfungsoperatoren den Kontrollfluss fest. Ein EPK-Modell kann zusätzlich um den Ressourcenaspekt (d.h. Organisationseinheiten, Informations- und Sachobjekte) ergänzt werden (erweiterte EPK, auch eEPK genannt). Für die Verbindungen werden sowohl gerichtete als auch ungerichtete Kanten verwendet, so dass eine (e)EPK nicht zwingend einen gerichteten Graphen darstellt. Für die EPK-Modellierung wurden verschiedene syntaktische Regeln aufgestellt, von denen die wesentlichen im Folgenden aufgeführt werden³¹:

- (i) Es existiert mindestens ein Start- und ein Endereignis.
- (ii) Ereignisse lösen Funktionen aus und Funktionen müssen von Ereignissen ausgelöst werden. Das einzige Ereignis, auf welches keine Funktion folgt, ist das Endereignis³².

³¹ Eine Übersicht zu den syntaktischen Regeln der EPK liefert die Arbeit von FELLMANN et al. [Fe13].

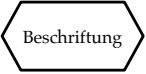
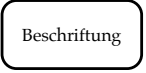

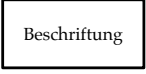



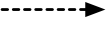


³² Daraus folgt, dass EPK immer mit Ereignissen beginnen und enden.

2. Grundlagen

- (iii) Informationsobjekte und Organisationseinheiten werden mit Funktionen verbunden.
- (iv) Ein Verknüpfungsoperator verbindet mehrere Funktionen und Ereignisse miteinander. Dabei gilt, dass Verknüpfungsoperatoren jeweils genau eine eingehende Kante und mehrere ausgehende Kanten oder vice versa besitzen dürfen.
- (v) Da Ereignisse keine Entscheidungskompetenz besitzen, darf auf ein Ereignis kein XOR- oder OR-Split folgen. Alle anderen Kombinationen von Ereignissen, Funktionen und entsprechenden Verknüpfungsoperatoren sind zulässig.

Die zentralen Modellierungselemente der EPK sind in Tab. 2.7 dargestellt. Obgleich dafür keine verbindliche Vorschrift existiert, werden EPK in der Regel von oben nach unten modelliert (Verlauf des Kontrollflusses), wie auch das Beispiel in Abb. 2.15. Hier wird eine eEPK für die Abwicklung eines Antrags auf Ausstellung einer KIT-Card dargestellt. Für die Erstellung der KIT-Card müssen die Studiengebühren bezahlt werden und ein Passfoto vorhanden sein. Dieses kann entweder direkt aufgenommen oder aber aus einer Passfoto-Datenbank kopiert werden. Informations- bzw. Datenobjekte, die in diesem Prozess auftauchen, sind die Passfoto-Datenbank, die von der Funktion `Passfoto prüfen` gelesen wird und der KIT-Card-Drucker, an welchem die KIT-Card von der Funktion `KIT-Card erstellen` ausgegeben wird. Im Beispiel tauchen zwei Organisationseinheiten auf: Studierende müssen die Funktion `Studiengebühren zahlen ausführen`, die KIT-Bibliothek ist für die Ausführung der Funktion `KIT-Card erstellen` verantwortlich. Desweiteren erfüllt das Beispiel alle zuvor aufgeführten syntaktischen Regeln für EPK.

Tab. 2.7.: Ereignisgesteuerte Prozesskette (EPK): Zentrale Modellierungselemente

Knotentyp $v \in V$	Semantik $sem(v)$	Notation $not(v)$
Ereignis	Passives Element, welches eine zeitpunktbezogene Zustandsbeschreibung enthält	
Funktion	Aktives Element, welches die Durchführung eines betrieblichen Vorgangs benennt	
Organisationseinheit	Für die Ausführung einer Funktion verantwortliche Organisationseinheit (z. B. Abteilung, Rolle)	
Informationsobjekt	Realer oder abstrakter Gegenstand, der in einer Funktion eine Rolle spielt	
OR-Split/Join	Verzweigung bzw. Zusammenführung des Kontrollflusses anhand logischem Oder	
XOR-Split/Join	Exklusive Verzweigung bzw. Zusammenführung des Kontrollflusses	
AND-Split/Join	Nebenläufige Verzweigung bzw. Zusammenführung des Kontrollflusses	
Kantentyp $e \in E$	Semantik $sem(e)$	Notation $not(e)$
Kontrollflusskante	Beschreibt den Kontrollfluss in Pfeilrichtung	
Informationsflusskante	Beschreibt den Informationsfluss in Pfeilrichtung	
Rollenkante	Zuordnung von Organisationseinheit zu Funktion	

Business Process Model And Notation (BPMN)

Wie auch die UML, wird die Business Process Model And Notation (BPMN) als Standard von der OMG herausgegeben. Gemäß der Spezifikation [Ob14] steht hinter der BPMN das Ziel, eine Notation zur Verfügung zu stellen, die für verschiedene Anwendergruppen – vom technischen Entwickler hin zum Geschäftsmanager – leicht verständlich ist. Sie wurde auf der Grundlage bestehender Modellierungssprachen entwickelt und enthält in der aktuellen Version 2.0.2 116 Modellierungselemente³³. Mit Hilfe der BPMN können Kollaborations-, Prozess- und Choreographie-Diagramme

³³ In ihrem vielzitierten Artikel »How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation« stellten zur MUEHLEN und RECKER [MR08] bereits für die Version 1.2 der BPMN mit 55 Modellierungselementen fest, dass in 120 analysierten BPMN-Modellen aus der Praxis lediglich durchschnittlich 9 Modellierungselemente verwendet wurden.

2. Grundlagen

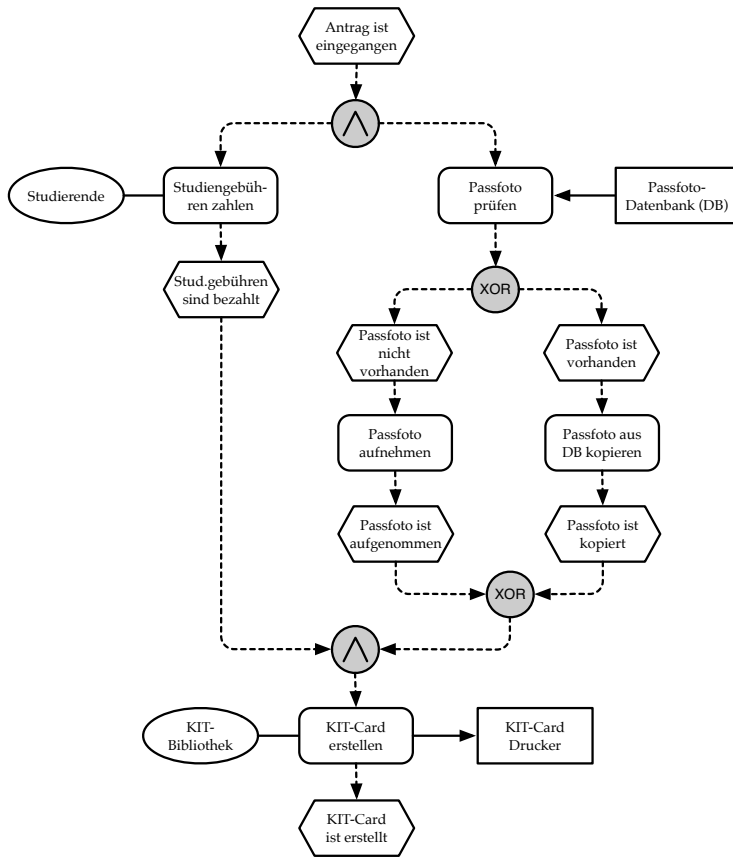


Abb. 2.15.: Beispiel für eine EPK zur Abwicklung eines Antrags auf Ausstellung einer KIT-Card

erstellt werden, die jeweils unterschiedliche Blickwinkel auf Geschäftsprozesse ermöglichen. Um die Verantwortlichkeiten für die Ausführung unterschiedlicher Aktivitäten und den Nachrichtenfluss zwischen den am Prozess beteiligten Akteuren besser darstellen zu können, greift die BPMN auf sogenannte Swimlanes zurück. Die übergeordneten Pools können hierbei in verschiedene Lanes eingeteilt sein, um auf diese Weise die beteiligten Rollen und Systeme einer Organisation abzubilden. Tab. 2.8 stellt mit 10 verschiedenen Modellierungselementen, die im Beispiel in Abb. 2.16 verwendet werden, nur einen Bruchteil der Gesamtmenge an Modellierungselementen aus der BPMN-Spezifikation vor. Das Beispiel verwendet zwei Pools *Studierende* und *Lehrstuhl* zur Darstellung der beteiligten Akteure. Anhand zweier Lanes werden für den Lehrstuhl die Rollen *Mitarbeitende* und *Sekretar*

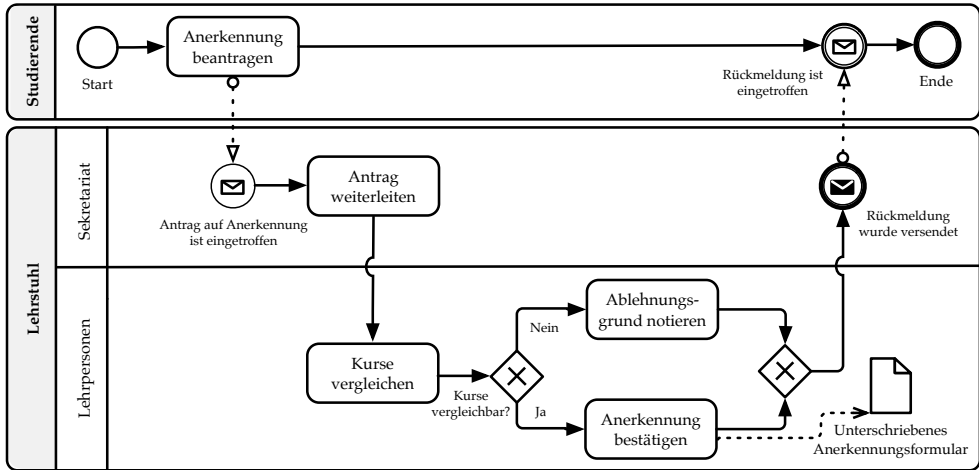


Abb. 2.16.: Beispiel für ein BPMN-Modell zum Ablauf einer Kursanerkennung










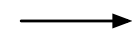
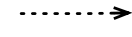
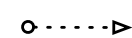
riat eingeführt. Eine studierende Person beginnt den Prozess mit der Beantragung der Anerkennung, diese geht beim Sekretariat ein und wird an die verantwortliche Lehrperson weitergeleitet. Diese vergleicht die im Antrag genannten Kurse und entscheidet über die Vergleichbarkeit. Sind die beiden Kurse nicht vergleichbar, wird ein Ablehnungsgrund notiert, anderenfalls wird die Anerkennung bestätigt und das unterschriebene Anerkennungsformular generiert. Anschließend versendet das Sekretariat die entsprechende Rückmeldung an die studierende Person.

2.2.4. Petri-Netze

Der Ausgangspunkt für die heutigen Formen von Petri-Netzen wurde 1962 in der Dissertation »Kommunikation mit Automaten« von PETRI [Pe62] entwickelt. Heutzutage werden Petri-Netze in vielen verschiedenen Anwendungsbereichen, darunter die Geschäftsprozessmodellierung, eingesetzt. Im Gegensatz zu den zuvor beschriebenen Sprachen für die Verhaltens- bzw. Ablaufbeschreibung (UML-AD, EPK, BPMN) besitzen Petri-Netze keine explizite Darstellung des Kontrollflusses durch spezielle Knotentypen. Vielmehr ist das dynamische Ablaufverhalten durch eine Schaltregel präzise festgelegt. Aufgrund ihrer formal eindeutig definierten Ablaufsemantik bieten Petri-Netze eine große Bandbreite an Analysemöglichkeiten, z. B. die Simulation.

2. Grundlagen

Tab. 2.8.: BPMN: Zentrale Modellierungselemente

Spezialtyp $s \in S$	Semantik $sem(s)$	Notation $not(s)$
Pool/Lane	Legt die Verantwortlichkeiten für die Ausführung der Aktivitäten im jeweiligen Bereich fest	
Knotentyp $v \in V$	Semantik $sem(v)$	Notation $not(v)$
Aktivität	Arbeitsschritt innerhalb eines Prozesses	
Datenobjekt	Information/Datei, die in einem Arbeitsschritt verwendet oder erzeugt wird	
Ereignis (eintretend)	Start eines Prozesses, erfordert Reaktion	
Endereignis (auslösend)	Ende eines Prozesses, erfordert keine Reaktion	
Ereignis (Typ Nachricht)	Start eines Prozesses und Empfangen einer Nachricht, erfordert Reaktion	
Zwischenereignis (Typ Nachricht)	Empfangen einer Nachricht innerhalb des Prozesses, erfordert Reaktion	
Endereignis (Typ Nachricht)	Versenden einer Nachricht und Ende des Prozesses, erfordert keine Reaktion	
XOR-Split/Join	Exklusive Verzweigung- bzw. Zusammenführung des Kontrollflusses	
Kantentyp $e \in E$	Semantik $sem(e)$	Notation $not(e)$
Sequenzfluss	Beschreibt den Kontrollfluss	
Datenassoziation	Verbindet Datenobjekte mit Aktivitäten	
Nachrichtenfluss	Beschreibt den Nachrichtenfluss (Kreis: Sender, Pfeil: Empfänger)	

Auf dem Gebiet der Petri-Netze³⁴ wurden zahlreiche unterschiedliche Netztypen entwickelt, ebenso wurde eine große Bandbreite an formalen Eigenschaften definiert.

³⁴ Der Begriff Petri-Netz wird fortan als Oberbegriff für die verschiedenen Netztypen verwendet.

In diesem Abschnitt werden ausgewählte Netztypen und für die entwickelten Bewertungsdienste relevante formale Eigenschaften eingeführt. Die Darstellung in den folgenden Definitionen orientiert sich an den Arbeiten von PETRI [Pe01; Pe80], den verbreiteten Lehrbüchern von BAUMGARTEN [Ba90] und REISIG [Re10] sowie an wissenschaftlicher Literatur aus dem Fachgebiet (z. B. [Aa98; Mu89; Ob96; Pe81]).

Definition 2.16: Netz

Ein **Netz** ist ein Tripel (S, T, F) , für welches gilt:

- (i) S ist die Menge von Stellen;
- (ii) T ist die Menge von Transitionen;
- (iii) $F \subseteq (S \times T) \cup (T \times S)$ ist eine Flussrelation;
- (iv) $S \cap T = \emptyset$;
- (v) $S \cup T \neq \emptyset$.

(In Anlehnung an [Ba90; Ob96])

Gemäß Def. 2.16 werden mit (i) und (ii) zunächst die zwei Knotentypen Stelle und Transition festgelegt. In der semantischen Domäne repräsentieren Stellen passive und Transitionen aktive Systemelemente [Pe80]. Aufgrund (iii) können gerichtete Kanten von Elementen aus S zu Elementen aus T und vice versa eingeführt werden³⁵. Zusätzlich sind die Mengen S und T disjunkt (wegen (iv)) sowie in Vereinigung nichtleer (wegen (v)). Aus (iii) und (iv) folgt darüber hinaus, dass es sich bei einem Netz um einen bipartiten Graphen handelt³⁶. Konventionell werden bei der grafischen Darstellung von Petri-Netzen Stellen durch Kreise, Transitionen durch Rechtecke und Kanten durch Pfeile repräsentiert. Abb. 2.17 zeigt ein beispielhaftes Netz N , welches durch die folgenden Mengen S, T, F angegeben ist:

- $S = \{s1, s2, s3, s4, s5, s6\}$
- $T = \{t1, t2, t3, t4, t5\}$
- $F = \{(s1, t1), (t1, s2), (t1, s3), (s2, t2), (s3, t3), (s3, t4), (t2, s4), (t3, s5), (t4, s5), (s4, t5), (s5, t5), (t5, s6)\}$

Eine häufig genutzte Grundlage für die formale Betrachtung von Netzen bildet das Konzept des Vor- und Nachbereichs von Netzelementen.

³⁵ Da das kartesische Produkt $S \times T := \{(s, t) \mid s \in S, t \in T\}$ der Menge aller geordneten Paare mit erstem Element aus S und zweitem Element aus T entspricht, sind die Kanten stets gerichtet.

³⁶ Ein Graph $G = (V, E)$ ist bipartit, wenn V sich in zwei disjunkte Teilmengen X und Y zerlegen lässt und zusätzlich für jede Kante $\{x, y\} \in E$ entweder $x \in X \wedge y \in Y$ oder $x \in Y \wedge y \in X$ gilt, d.h. keine Kante zwischen Knoten innerhalb einer der beiden Teilmengen verläuft.

2. Grundlagen

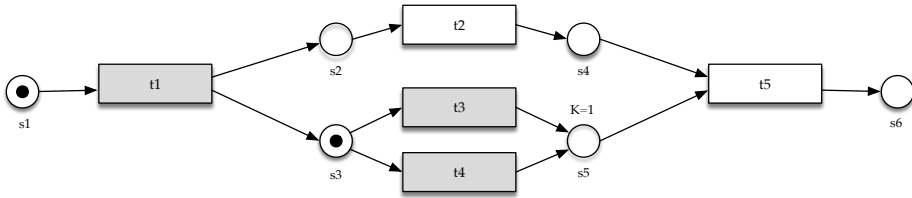


Abb. 2.17.: Beispiel für die grafische Darstellung des Netzes N

Definition 2.17: Vor- und Nachbereich

Für jedes Element $x \in S \cup T$ eines Netzes (S, T, F) gilt:

$\bullet x := \{y \mid (y, x) \in F\}$ ist der **Vorbereich** von x ;

$x \bullet := \{y \mid (x, y) \in F\}$ ist der **Nachbereich** von x .

(Übersetzt und angelehnt an [Pe01].)

Für das zuvor angegebene Netz N (siehe Abb. 2.17) ist beispielsweise $\bullet t_5 = \{s_4, s_5\}$ der Vorbereich und $t_5 \bullet = \{s_6\}$ der Nachbereich der Transition t_5 .

Auf Basis der Topologie eines Netzes können ausschließlich statische Netzeigenschaften definiert werden. Außerdem lassen sich Netztypen angeben, die über rein graphentheoretische Bedingungen definiert sind, wie z. B.. Zustandsmaschinen oder Free-Choice-Netze [Ba90, S. 71-74]. Def. 2.18 liefert ein Beispiel für eine statische Netzeigenschaft.

Definition 2.18: Stark zusammenhängend

Ein Netz (S, T, F) ist genau dann **stark zusammenhängend**, wenn für alle Elementpaare $x, y \in S \cup T$ mit $x \neq y$ eine Kantenfolge $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n) \in F$ mit Anfangselement $v_1 = x$ und Endelement $v_n = y$ existiert. (Siehe [Aa98; Ba90].)

Nach dieser Definition ist das zuvor angegebene Netz N nicht stark zusammenhängend. Für den Beweis genügt die Angabe eines beliebigen Elementpaares, für das keine entsprechende Kantenfolge gefunden werden kann, z. B. t_5, s_2 .

Die Betrachtung dynamischer Netzeigenschaften wird erst durch die Erweiterung der Netz-Definition um das Konzept der Marken sowie der Einführung einer Schaltregel ermöglicht. In dieser Arbeit wird dazu der Netztyp der Stellen/Transitionen-Netze betrachtet, die zu den sogenannten markierten Petri-Netzen zählen.

Definition 2.19: Stellen/Transitionen-Netz

Ein **Stellen/Transitionen-Netz** (S/T-Netz) ist ein 6-Tupel (S, T, F, K, W, M_0) , für welches gilt:

- (i) (S, T, F) ist ein Netz,
- (ii) $K : S \rightarrow \mathbb{N} \cup \{\infty\}$ weist jeder Stelle eine Aufnahmekapazität für Marken zu,
- (iii) $W : F \rightarrow \mathbb{N}$ weist jeder Kante eine Gewichtung zu,
- (iv) $M_0 : S \rightarrow \mathbb{N}_0$, wobei $\forall s \in S : M_0(s) \leq K(s)$, weist jeder Stelle eine Anzahl von Marken zu (Startmarkierung).

(Angelehnt an [Ba90; Ob96].)

Bei der grafischen Darstellung von S/T-Netzen werden die Aufnahmekapazität für Marken sowie die Gewichtung von Kanten anhand von Beschriftungen abgebildet. Bei leerer Beschriftung wird eine unbeschränkte Aufnahmekapazität (∞) bzw. eine Kantengewichtung von 1 angenommen. Die Anzahl an Marken in einer Stelle wird durch Punkte dargestellt. Für S/T-Netze gilt die folgende Schaltregel in Def. 2.20:

Definition 2.20: Schaltregel für S/T-Netze

Für ein gegebenes **S/T-Netz** (S, T, F, K, W, M_0) , gilt:

- Eine Transition $t \in T$ heißt **aktiviert** unter einer Markierung $M : S \rightarrow \mathbb{N}_0$, wenn:
 - (i) $\forall s \in \bullet t : M(s) \geq W(s, t)$,
 - (ii) $\forall s \in t \bullet : M(s) \leq K(s) - W(t, s)$.
- Wenn eine Transition t unter M aktiviert ist, dann kann t **schalten**. Das Schalten von t unter M überführt M in die Folgemarkierung M' , wobei gilt:

$$M'(s) = \begin{cases} M(s) - W(s, t), & \text{falls } s \in \bullet t \text{ und } s \notin t \bullet, \\ M(s) + W(t, s), & \text{falls } s \in t \bullet \text{ und } s \notin \bullet t, \\ M(s) - W(s, t) + W(t, s), & \text{falls } s \in t \bullet \text{ und } s \in \bullet t, \\ M(s) & \text{sonst.} \end{cases}$$

(Angelehnt an [Ba90; Ob96].)

Um das zuvor angegebene Netz N in ein S/T-Netz N' zu überführen, seien K, W und M_0 beispielsweise wie folgt festgelegt:

2. Grundlagen

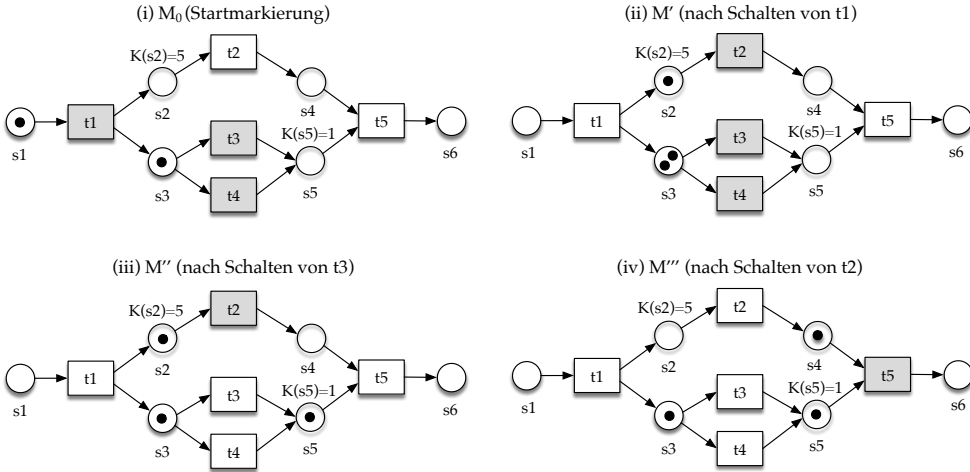


Abb. 2.18.: Grafische Darstellung und beispielhaftes Schaltverhalten des S/T-Netzes N' . Jeweils aktivierte Transitionen sind zur Verdeutlichung grau eingefärbt.

- $K(s_1) = K(s_3) = K(s_4) = K(s_6) = \infty, K(s_2) = 5, K(s_5) = 1$
- $W : F \rightarrow \{1\}$
- $M_0 = \{(s_1, 1), (s_2, 0), (s_3, 1), (s_4, 0), (s_5, 0), (s_6, 0)\}$

Dieses S/T-Netz N' ist in Abb. 2.18 dargestellt. Unter M_0 sind gemäß der Schaltregel aus Definition 2.20 die Transitionen t_1 , t_3 und t_4 aktiviert. Nach dem anfänglichen Schalten von Transition t_1 ergibt sich zunächst die Folgemarkierung $M' = \{(s_1, 0), (s_2, 1), (s_3, 2), (s_4, 0), (s_5, 0), (s_6, 0)\}$, unter der die Transitionen t_2 , t_3 und t_4 aktiviert sind. Dieser Schaltvorgang kann auch in der Form $M_0 \xrightarrow{t_1} M'$ notiert werden. Durch das Schalten der Transition t_3 (oder alternativ t_4) ergibt sich die Folgemarkierung $M'' = \{(s_1, 0), (s_2, 1), (s_3, 1), (s_4, 0), (s_5, 1), (s_6, 0)\}$ unter der nur noch t_2 aktiviert ist. Für die beiden Transitionen t_3 und t_4 ist zwar die Bedingung (i) gemäß Def. 2.20 für die Aktivierung erfüllt, allerdings aufgrund von $K(s_5) = 1$ nicht die Bedingung (ii). Die Folgemarkierung $M''' = \{(s_1, 0), (s_2, 0), (s_3, 1), (s_4, 1), (s_5, 1), (s_6, 0)\}$ wiederum folgt aus dem anschließenden Schalten der Transition t_2 .

Durch die Formalisierung der Schaltregel ist die Kontrollflusssemantik für Petri-Netze präzise definiert. Das wiederholte Schalten von Transitionen in einem Petri-Netz, um mögliche Schaltfolgen ausgehend von einer Startmarkierung zu ermitteln, wird auch als Markenspiel bezeichnet. Durch das systematische Prüfen von Schaltmöglichkeiten kann eine Erreichbarkeitsanalyse zur Ermittlung der Menge aller

möglichen Folgemarkierungen durchgeführt werden (siehe z. B. [Ba90, S. 84f] und Abschnitt 5.3.5). Auch die Simulation von Petri-Netzen basiert auf dem wiederholten Ausführen des Markenspiels. Die Umsetzung von Simulationswerkzeugen auf Basis von Petri-Netzen ist bereits in frühen Arbeiten für zahlreiche Anwendungsbereiche beschrieben worden (z. B. für Computersysteme [BPR72] oder für Geschäftsprozesse [PS93]).

Im Folgenden werden einige Definitionen für dynamische Eigenschaften vorgestellt und auf die hier betrachteten S/T-Netze angepasst, um an späterer Stelle darauf zurückgreifen zu können.

Definition 2.21: Erreichbar

Eine Markierung M_n heißt **erreichbar** unter einer Markierung M_1 (geschrieben: $M_1 \xrightarrow{*} M_n$) genau dann, wenn es eine Schaltfolge $\sigma = t_1, t_2, \dots, t_{n-1}$ gibt so dass $M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} \dots \xrightarrow{t_{n-1}} M_n$ gilt. Dabei ist auch die leere Schaltfolge erlaubt, also z.B. $M_1 \xrightarrow{*} M_1$. (Übersetzt nach [Aa96].)

Definition 2.22: Lebendig

Ein S/T-Netz (S, T, F, K, W, M_0) ist **lebendig** genau dann, wenn es für jede unter M_0 erreichbare Markierung M' und jede Transition $t \in T$ eine unter M' erreichbare Markierung M'' gibt, die t aktiviert. (Übersetzt nach [Aa96].)

Definition 2.23: Beschränkt, sicher

Ein S/T-Netz (S, T, F, K, W, M_0) ist **beschränkt** genau dann, wenn für jede unter M_0 erreichbare Markierung M' und jede Stelle $s \in S$ die Anzahl an Marken in s beschränkt ist. Das S/T-Netz wird **sicher** (oder 1-beschränkt) genannt, wenn die Anzahl an Marken in jeder Stelle ≤ 1 ist. (Übersetzt nach [Aa96].)

Im Zusammenhang mit der Modellierung von Geschäftsprozessen mit Petri-Netzen spielen insbesondere Workflow-Netze eine bedeutende Rolle. Die Workflow Management Coalition (WMC) definiert einen Workflow als eine vollständig oder teilweise IT-gestützte Ausführung eines Geschäftsprozesses [Ho95, S. 6]³⁷. Nach van der AALST [Aa96; Aa98] sind Workflows fallbasiert, d.h. sie beziehen sich auf

³⁷ Insbesondere in der früheren Literatur wird der Begriff Workflow auch synonym zum Begriff Geschäftsprozess verwendet [AH04].

2. Grundlagen

die Abwicklung von Fällen des zugrundeliegenden Prozesses. Durch die Modellierung eines Workflows mit einem sogenannten Workflow-Netz wird beschrieben, welche Aktivitäten in welcher Reihenfolge ausgeführt werden müssen, um einen Fall abzuwickeln. Eine Instanz eines Workflow-Netzes repräsentiert dabei die individuelle Durchführung des entsprechenden Workflows für einen spezifischen Fall. Ein Workflow-Netz sei wie folgt definiert:

Definition 2.24: Workflow-Netz

Ein Netz $N = (S, T, F)$ ist ein **Workflow-Netz** genau dann, wenn gilt:

- (i) N hat genau^a zwei spezielle Stellen i und o , für die gilt:
 - $\bullet i = \emptyset$: Die Stelle i ist eine Eingabestelle (Quelle),
 - $o \bullet = \emptyset$: Die Stelle o ist eine Ausgabestelle (Senke).
- (ii) Wenn eine Transition t^* zu N hinzugefügt wird, welche die Stellen o und i miteinander verbindet ($\bullet t^* = \{o\}$ und $t^* \bullet = \{i\}$), ist das daraus resultierende Netz stark zusammenhängend.

Übersetzt und angelehnt an [Aa96].

^a In [Aa96] heißt es »has two special places«. Da aus dieser Formulierung nicht präzise hervorgeht, ob damit genau zwei oder mindestens zwei Stellen gemeint sind, wurde in dieser Definition der Zusatz »genau« ergänzt.

Gemäß Def. 2.24 bilden Workflow-Netze eine Teilklasse von Netzen (siehe Def. 2.16), welche die notwendigen Eigenschaften eines Workflows aufweisen. Eine Marke in der Quelle i eines Workflow-Netzes entspricht einem Fall, der abgewickelt werden muss, eine Marke in der Senke o entspricht einem Fall, der vollständig abgewickelt wurde. In einem Workflow-Netz werden Aktivitäten durch Transitionen, Bedingungen durch Stellen und der Zustand des Falls durch die Markierung abgebildet. Durch die Bedingung (ii) aus Def. 2.24 wird die Anforderung erfüllt, dass alle Transitionen und Stellen im Netz theoretisch zur Abwicklung von Fällen beitragen können. Beide Bedingungen (i) und (ii) können durch die Analyse der statischen Netzstruktur überprüft werden.

Über die sogenannte Soundness-Eigenschaft von Workflow-Netzen [Aa96; Aa97; Aa98] kann zusätzlich auch das dynamische Verhalten auf weitere nützliche Anforderungen hin untersucht werden. Diese Anforderungen besagen grob, dass (a) ein Workflow-Netz immer korrekt terminieren muss und dass (b) alle Transitionen

aktivierbar sein müssen. Dies wird anhand Definition 2.25³⁸ verdeutlicht. Dabei gilt, dass M_i diejenige Markierung bezeichnet, in welcher ausschließlich die Quelle i des Workflow-Netzes mit genau einer Marke belegt ist und alle anderen Stellen unmarkiert sind (gilt analog für M_o und die Senke o des Workflow-Netzes).

Definition 2.25: Sound

Ein Workflow-Netz $WN = (S, T, F, M_0)$ mit $M_0 = M_i$ ist **sound** genau dann, wenn gilt:

- (i) Für jede Markierung M , die von M_i aus erreichbar ist, existiert eine Schaltfolge von M nach M_o , d.h.: $\forall M : (M_i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} M_o)$.
- (ii) Die einzige Markierung, die von M_i aus erreichbar ist und bei der sich mindestens eine Marke in der Ausgabestelle o befindet, ist M_o , d.h.: $\forall M : (M_i \xrightarrow{*} M \wedge M \geq M_o) \Rightarrow (M = M_o)$.^a
- (iii) Unter M_i besitzt WN keine toten Transitionen, d.h.: $\forall t \in T \exists M, M' : M_i \xrightarrow{*} M \xrightarrow{t} M'$.

Übersetzt und angelehnt an [Aa96; Aa97; Aa98].


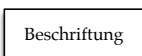


^a Der Vergleichsoperator \geq bezieht sich beim Vergleich zweier Markierungen auf die jeweilige Anzahl an Marken im Netz.

In Tab. 2.9 sind die in diesem Abschnitt eingeführten Modellierungselemente für Petri-Netze noch einmal zusammengefasst dargestellt. Das Petri-Netz in Abb. 2.19 ist aus struktureller Sicht äquivalent zu den Petri-Netzen, die in den vorangegangenen Abbildungen 2.17 und 2.18 vorgestellt wurden. Anhand der Beschriftungen ermöglicht dieses Petri-Netz allerdings die Abbildung eines Prozesses, der die Eigenschaften eines Workflows (fallbasiert und (teil)-automatisiert) erfüllt. Es folgt den Definitionen für S/T-Netze (Def. 2.19) und Workflow-Netze (Def. 2.24). Die Marken im Petri-Netz drücken den aktuellen Zustand des Falls aus, nämlich dass die beiden Bedingungen Mahnung ist erstellt und Gebühren sind eingegangen erfüllt sind. Unter Berücksichtigung der Schaltregel (Def. 2.20) kann der Ablauf, bzw. der Kon-

³⁸ In [Aa96; Aa97; Aa98] wird die Soundness-Eigenschaft auf unmarkierten Workflow-Netzen definiert. Auch in jüngeren Arbeiten von VAN DER AALST wird das Markenkonzept zwar für Netze (gemäß Def. 2.16) eingeführt, jedoch nicht für Workflow-Netze (siehe z. B. [Aa16, S. 60-66]). Die in dieser Arbeit angegebene Definition folgt daher dem Beispiel von RICHTER-VON HAGEN und STUCKY [RS04, S. 115] und definiert die Soundness als Eigenschaft markierter Workflow-Netze durch Erweiterung um die Angabe einer Startmarkierung $M_0 = M_i$. Auf die Angabe einer eigenen Schaltregel für das markierte Workflow-Netz wird verzichtet. Grundsätzlich lässt sich z. B. auch die Schaltregel für S/T-Netze aus Def. 2.20 anwenden, wenn zusätzlich Kantengewichtung K und Aufnahmekapazität W (siehe Def. 2.19) festgelegt werden. Diese Attribute bleiben bei Workflow-Netzen jedoch in der Regel unberücksichtigt.

2. Grundlagen

Tab. 2.9.: Petri-Netze: Zentrale Modellierungselemente

Knotentyp $v \in V$	Semantik $sem(v)$	Notation $not(v)$
Stelle	Bedingung oder Zustand (passives Systemelement)	 Beschriftung
Transition	Aktivität (aktives Systemelement)	 Beschriftung
Marke	Drückt aus, ob die Bedingung einer Stelle erfüllt ist, bzw. ob ein bestimmter Zustand besteht	
Kantentyp $e \in E$	Semantik $sem(e)$	Notation $not(e)$
Gerichtete Kante	Bestimmt die Flussrichtung	

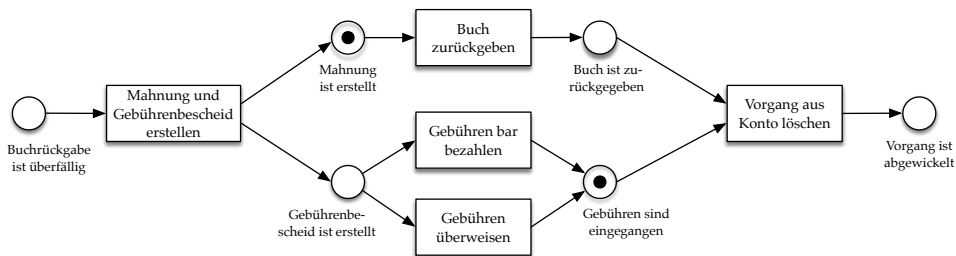


Abb. 2.19.: Beispiel-Petri-Netz zur Abwicklung einer überfälligen Buchrückgabe

trollfluss im Prozess ermittelt werden, um z. B. zu erkennen, welche Aktivitäten für den betrachteten Fall als nächstes auszuführen sind. Die einzige Transition, die auf Grundlage der aktuell vorliegenden Markierung aktiviert ist, entspricht der Aktivität Buch zurückgeben.

Im Rahmen der Workflow Patterns Initiative wurde eine umfassende Sammlung von 43 fundamentalen Workflow-Kontrollflussmustern (engl. *Workflow control-flow pattern*, WCP) veröffentlicht [Ru06], die nicht nur bei der Modellierung von Workflows sondern auch allgemein bei der Modellierung von Geschäftsprozessen unabhängig von der verwendeten Modellierungssprache zum Einsatz kommen können^{39,40}. Die folgende Tabelle stellt eine Übersicht vor, bei der einige ausgewählte Kontroll-

³⁹ Die Kontrollflussmuster sind darüber hinaus auch dazu geeignet, die Ausdrucksmächtigkeit von Geschäftsprozessmodellierungssprachen in Bezug auf ebendiese Muster zu untersuchen, siehe dazu auch die durchgeführten Analysen im Anhang von [Ru06, S. 92-134].

⁴⁰ Auch von BÖRGER [Bö07] und BÖRGER und THALHEIM [BT08] wurde eine Sammlung von 13 elementaren Kontrollflussmustern anhand abstrakter Zustandsmaschinen formal präzise beschrieben.

flussmuster in Form von Petri-Netzen dargestellt werden⁴¹. Die Kontrollflussmuster WCP-1 bis WCP-5 sind sogenannte Basis-Kontrollflussmuster, bei WCP-21 handelt es sich um eines von drei Iterationsmustern, die in der Sammlung veröffentlicht wurden. Die weiteren Muster lassen sich beispielsweise in Klassen erweiterter Verzweigungs- und Synchronisationsmuster, Abbruchmuster, Trigger-Muster und zustandsbasierte Muster einteilen. Auch das Beispiel in Abb. 2.19 weist einige Kontrollflussmuster auf, die in Tab. 2.10 vorgestellt wurden. So finden die Rückgabe des Buches und die Bezahlung der Gebühren nebenläufig statt, das heißt gleichzeitig oder in unspezifischer Reihenfolge (WCP-2; WCP-3). Die Bezahlung der Gebühren kann entweder bar oder per Überweisung erfolgen (WCP-4; WCP-5).

Während die in diesem Abschnitt vorgestellten – sogenannten elementaren – Netztypen zwar die Darstellung der Kontrollflussperspektive abdecken, bleiben andere Aspekte unberücksichtigt. So bieten elementare Netztypen keine Konzepte für die Beschreibung von Daten oder des zeitlichen Verhaltens. Aus diesem Grund wurden Erweiterungen entwickelt, die als höhere Petri-Netze (engl. *high level Petri nets*) bezeichnet werden. Beispielsweise führen Prädikate/Transitionen-Netze [GL79] oder gefärbte Petri-Netze [Je81] das Konzept typisierter bzw. unterscheidbarer Marken ein und erweitern das Schaltverhalten um den Zugriff auf Attribute dieser Marken. Bei der Verwendung von zeitbehafteten [Ra74] oder stochastischen Petri-Netzen [Mo82] können konstante bzw. exponentialverteilte Werte zum zeitlichen Verhalten, wie z.B. Schaltdauer oder Wartezeiten festgelegt werden, was insbesondere für die Simulation von Bedeutung ist. Große und komplexe Petri-Netze können durch Einführung einer Hierarchie über verschiedene Abstraktionsebenen übersichtlicher gestaltet werden. Dazu werden Teilnetze eingeführt und mittels den Netztransformationen der Vergrößerung bzw. Verfeinerung aus dem Gesamtnetz aus- bzw. eingeblendet (siehe z. B. [Ba90; Re10]). Letzteres, sowie die aufgeführten höheren Petri-Netze stehen im Zusammenhang mit den Lerninhalten für die Geschäftsprozessmodellierung nicht im Vordergrund (siehe Abschnitt 3.1) und werden daher in dieser Arbeit nicht vertieft behandelt. Sie bieten jedoch einen Ausblick auf künftige Weiterentwicklungsmöglichkeiten im Kontext der hier behandelten Forschungsfragen.

⁴¹ In [Ru06] werden die Kontrollflussmuster als gefärbte Petri-Netze (*coloured Petri nets* (CPN)) abgebildet, da im Rahmen dieser Arbeit jedoch ein Ansatz für elementare Petri-Netze entwickelt wird, wurde eine Übertragung der Muster in Petri-Netze vorgenommen.

2. Grundlagen

Tab. 2.10.: Ausgewählte Kontrollflussmuster der Workflow Pattern Initiative [Ru06], eigene Darstellung der Beispiele.

Kontrollflussmuster	Beschreibung	Beispiel
Sequenz (WCP-1)	Aktivität A2 kann nach Abschluss der vorangehenden Aktivität A1 unmittelbar ausgeführt werden.	
Parallele Verzweigung (WCP-2)	Ab Aktivität A1 verzweigt der Kontrollfluss in zwei (oder mehr) parallele Pfade, so dass die Aktivitäten A2 und A3 nebenläufig ausgeführt werden können (AND-Split).	
Synchronisation (WCP-3)	Zwei (oder mehr) parallele Pfade werden nach Abschluss der Aktivitäten A1 und A2 wieder zu einem Pfad zusammengeführt, so dass Aktivität A3 ausgeführt werden kann (AND-Join).	
Exklusive Auswahl (WCP-4)	Ab Aktivität A1 verzweigt der Kontrollfluss in zwei (oder mehr) parallele Pfade von denen genau einer ausgewählt werden muss, so dass entweder Aktivität A2 oder Aktivität A3 ausgeführt werden kann (XOR-Split).	
Einfache Zusammenführung (WCP-5)	Zwei (oder mehr) parallele Pfade werden nach Abschluss einer der Aktivitäten A1 oder A2 wieder zu einem Pfad zusammengeführt, so dass Aktivität A3 ausgeführt werden kann (XOR-Join).	
Strukturierte Schleife (WCP-21)	Die Aktivitäten A1 und A3 können wiederholt nacheinander ausgeführt werden bevor Aktivität A2 ausgeführt wird (Dabei kann A1 als Pre- und A3 als Post-Test auf Vorliegen einer Bedingung für die Ausführung der jeweils anderen Aktivität interpretiert werden).	

3. Verwandte Arbeiten

Im Folgenden werden Arbeiten aus der wissenschaftlichen Literatur vorgestellt, die mit dem Forschungsvorhaben in einem engeren Zusammenhang stehen. Dabei werden nicht nur Arbeiten aus dem Anwendungsbereich der Geschäftsprozessmodellierung betrachtet, sondern auch allgemeinere Arbeiten, z. B. aus dem Gebiet der konzeptuellen Modellierung oder zu anderen grafischen Modellierungssprachen (siehe Abschnitt 2.2.3). Gewissermaßen handelt es sich bei den Inhalten dieses Kapitels ebenso um Grundlagen wie beim vorherigen Kapitel 2, allerdings besitzen sie einen stärkeren Bezug zum Forschungsvorhaben. Sie werden daher in einer sehr breiten Auslegung des Begriffs als »verwandte Arbeiten« bezeichnet.

Während in der zweiten Hälfte von Kapitel 2 zwischen den Begriffen Modell (als gedankliche Konzeption) und Modelldenotation (als dessen externe Repräsentation) unterschieden wurde (siehe Def. 2.10 und 2.11), so wird zur vereinfachten Lesbarkeit ab nun der Begriff Modell wieder so eingesetzt, wie es in der wissenschaftlichen Literatur gängig ist – d.h. bei einem Modell handelt es sich um eine Modelldenotation.

Im ersten Abschnitt 3.1 wird der Anwendungskontext Hochschullehre im Zusammenhang mit dem Thema Modellierung genauer beleuchtet. Der zweite Abschnitt 3.2 behandelt konzeptionelle Arbeiten, die sich mit der Modellqualität im Allgemeinen befassen, wie z.B. Qualitätsframeworks, Richtlinien und Qualitätsmetriken. Anschließend werden im dritten Abschnitt 3.3 Arbeiten betrachtet, die in Bezug auf die technische Realisierung von Bewertungsdiensten von Relevanz sind. Dazu zählen automatisierte Verfahren zur allgemeinen Analyse von Modellen (Abschnitt 3.3.1) und insbesondere Ansätze, die sich ebenso wie die vorliegende Arbeit mit der automatisierten Bewertung von Modellen im Kontext Hochschullehre befassen (Abschnitt 3.3.2). Gleichzeitig wird in diesem Kapitel an passenden Stellen die bestehende und mit dem Forschungsvorhaben adressierte Forschungslücke aufgezeigt.

3.1. Modellierung in der Hochschullehre

Die internationale Fachgesellschaft »Association for Computing Machinery« (ACM) veröffentlicht seit 1960 regelmäßig Empfehlungen für die Curriculumsgestaltung IT-naher Studiengänge wie z. B. Computer Science (CS), Information Systems (IS), Information Technology (IT) und Software Engineering (SE) [As20]. Auch die deutsche Fachgesellschaft »Gesellschaft für Informatik e.V.« (GI) gibt derartige Empfehlungen heraus, z. B. für die Studiengänge Informatik (INF) [Ge16a] und Wirtschaftsinformatik (WI) [Ge16b]. Eine Analyse der genannten Empfehlungen zeigt, dass Module oder Kurse mit Bezug zur Modellierung¹ ausnahmslos in allen jeweils aktuellsten Fassungen vertreten sind. Die Bandbreite der Anwendungsbereiche reicht dabei von der Datenmodellierung (CS, IS, IT, SE, INF, WI) über die Softwareentwicklung (CS, IT, SE, INF, WI) hin zur Geschäftsprozessmodellierung (IS, SE, WI)². Diese Ergebnisse belegen die Relevanz des Themas Modellierung in der Hochschullehre für informatikbezogene Studiengänge aus Sicht dieser beiden Fachgesellschaften.

Die Behandlung des Themas Modellierung in Lehrveranstaltungen wird an deutschsprachigen Universitäten ganz unterschiedlich realisiert [Gl08, S. 248]. Einerseits variiert der Umfang, in dem Modellierungsthemen behandelt werden, andererseits ist auch die Positionierung uneinheitlich. So gibt es teilweise explizite Grundlagenveranstaltungen zur Modellierung, teilweise wird die Modellierung aber auch im Kontext individueller Fachvorlesungen behandelt (z. B. Datenmodellierung als Bestandteil einer Vorlesung zum Thema Datenbanken). GLINZ [Gl08] befürwortet die Behandlung des Themas Modellierung in einer eigenständigen und obligatorischen Einführungsvorlesung zusammen mit der späteren Vertiefung in Fachvorlesungen (siehe Thesen 11 und 12 in [Gl08, S. 429]). Das im Forschungsvorhaben entwickelte E-Assessment-Werkzeug kann grundsätzlich in allen Veranstaltungen mit Bezug zur grafischen Modellierung eingesetzt werden, unabhängig davon, ob es sich um eine einschlägige Modellierungsveranstaltung oder um eine Fachvorlesung zum Geschäftsprozessmanagement handelt.

Im folgenden Abschnitt 3.1.1 werden fachdidaktische Arbeiten sowie Ansätze mit Bezug auf Modellierungskompetenzen vorgestellt. Diese sollen in die Entwicklung des Kompetenzmodells (ad Forschungsfrage FF1, siehe Abschnitt 1.2) einfließen.

¹ Darüber hinaus enthalten die Empfehlungen der ACM zum Begriff Modellierung auch Treffer wie z. B. »mathematical modeling« oder »modeling and simulation«, die sich allerdings nicht auf die in dieser Arbeit betrachtete grafische Modellierung beziehen und daher nicht berücksichtigt wurden.

² Die Kürzel in den Klammern hinter den Anwendungsgebieten geben die Vertretung in den Empfehlungen für die jeweiligen Studiengänge an.

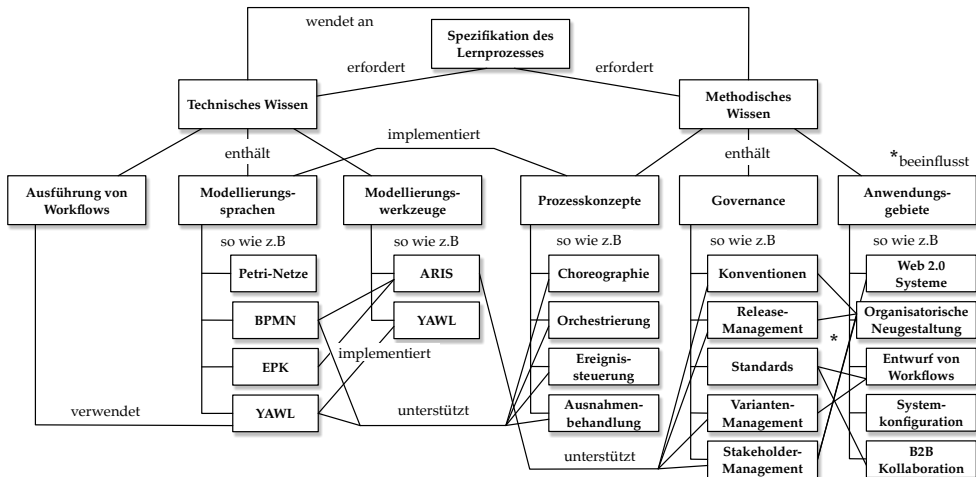


Abb. 3.1.: Concept-Map zum Kurs »Business Process Modelling«, nach [RR09, S. 4]

Der anschließende Abschnitt 3.1.2 stellt verschiedene Aufgabentypen vor, die in der Hochschullehre auf dem Gebiet der Geschäftsprozessmodellierung eingesetzt werden.

3.1.1. Fachdidaktik und Modellierungskompetenzen

Nur wenige wissenschaftliche Arbeiten setzen sich mit dem Thema Geschäftsprozessmodellierung in der Hochschullehre aus fachdidaktischer Sicht auseinander³. Es existiert zwar eine Reihe von Vorschlägen zur Curriculumentwicklung für das Feld des Geschäftsprozessmanagements (z. B. [An10; Ba07; Ba10; BST15; Re12; Se10]). In diesen wird das Teilgebiet Geschäftsprozessmodellierung explizit erwähnt, jedoch nicht detailliert ausgeführt. Oft wird auf den Erfahrungsbericht von RECKER und ROSEMAN [RR09] über einen Kurs zur Geschäftsprozessmodellierung verwiesen⁴. Dort werden Kursinhalte und Zusammenhänge zwischen diesen in Form einer Concept-Map vorgestellt (siehe Abb. 3.1). Anhand dieser Darstellung wird deutlich, dass neben dem technischen Wissen über Modellierungssprachen und

³ Häufiger finden sich Beschreibungen innovativer methodischer Ansätze wie z. B. Gamification [PV18] oder Social BPM Labs [Ca13a; Ca13b]. Bei letzteren hat die Verfasserin dieser Dissertation im Rahmen eines früheren Projekts »Lernen – Forschen – Anwenden« mitgewirkt. In diesem Forschungsvorhaben werden methodische Aspekte allerdings explizit ausgeklammert.

⁴ Zitiert von 59 Quellen laut Google Scholar <http://scholar.google.com>. Abfrage am 24.06.2020.

3. Verwandte Arbeiten

Werkzeuge auch ein grundsätzliches methodisches Wissen bei der Geschäftsprozessmodellierung eine wichtige Rolle einnimmt. RECKER und ROSEMANN [RR09, S. 9] beschreiben dies so: »[...] we recognize a need to complement the technical skills usually taught [...] with methodological knowledge pertaining to the establishment of process capabilities, i.e., the ability to identify, and critically reflect upon, relevant ›process‹ concepts such as event management, exception handling, and choreography and orchestration of inter-related business activities«.

Da sich ansonsten in der Literatur keine weiteren entsprechenden Arbeiten finden, die sich speziell auf die Geschäftsprozessmodellierung beziehen, wurden auch generelle Arbeiten zur Modellierung berücksichtigt. Diese werden im Folgenden in chronologischer Reihenfolge vorgestellt.

Die älteste Arbeit stammt aus dem Jahr 1967. MORRIS [Mo67] betrachtet Modellierung als Kunst, die auf den intuitiven Fähigkeiten eines erfahrenen Modellierers aufbaut, und widmet sich der folgenden Frage: »If one grants that modeling is and, for greatest effectiveness, probably ought to be, an intuitive process for the experienced, then the interesting question becomes the pedagogical problem of how to develop this intuition« [Mo67, S. 707]. In seinem Ansatz zur Entwicklung von Modellierungsfähigkeiten stellt er eine Reihe von Hypothesen auf, die sich auf den Prozess der Modellierung beziehen, z. B. die schrittweise Erweiterung eines einfachen Modells, die Berücksichtigung bekannter Analogien oder die Betrachtung einer spezifischen Instanz des Problems. Die Hypothesen sind abstrakt gehalten und anleitender Natur (z. B. »establish some symbols« oder »write down the obvious«). Bei der Entwicklung von Modellierungsfähigkeiten kommt es laut MORRIS [Mo67, S. 716] vor allem auf die Betrachtung des Modellierungsprozesses an sich an, denn: »the teaching of models is not equivalent to the teaching of modeling«.

POWELL [Po95a; Po95b] sieht Modellierung ebenso wie MORRIS [Mo67] als Kunst und stuft fundamentale Modellierungsfähigkeiten als wichtiger ein als das Wissen über konkrete Modellierungsmethoden bzw. -sprachen. Er beschreibt sechs Modellierungsprinzipien – Heuristiken genannt – die er Studierenden vermitteln möchte. Zu diesen zählen Konzepte wie z. B. Dekomposition, Prototyping, Visualisierung und Parametrisierung.

FREDERIKS und van der WEIDE [FW06] befassen sich mit den erforderlichen Kompetenzen zur Informationsmodellierung. Dazu betrachten sie die am Modellierungsprozess beteiligten Rollen des Domänenexperten und des Systemanalysten (modellierende Person). Da im Rahmen dieser Arbeit der Fokus auf Modellierungskompetenzen

liegt, werden im Folgenden die acht Fähigkeiten A1-A8 des Systemanalysten (A) vorgestellt, die in [FW06, S. 14f] beschrieben werden⁵.

- A1:** »Tabula rasa base skill«: Damit alle relevanten Aspekte (wiederkehrende Handlungsabläufe und Ausnahmen) der Domäne berücksichtigt werden, ist die anfängliche Unvoreingenommenheit des Systemanalysten von großer Bedeutung.
- A2:** »Consistency base skill«: Es ist notwendig, dass der Systemanalyst die Menge der Domänenaussagen⁶ auf ihre Konsistenz hin überprüfen kann, um z. B. Widersprüche aufzudecken.
- A3:** »Grammar base skill«: Diese Fähigkeit bezieht sich darauf, dass ein Systemanalyst dazu in der Lage sein muss, anhand der Domänenaussagen eine Menge an syntaktischen Kategorien zu identifizieren.
- A4:** »Abstraction base skill«: Die Menge syntaktischer Kategorien muss vom Systemanalysten auf Übereinstimmungen überprüft werden, um daraufhin zugrundeliegende Satzstrukturen ableiten zu können.
- A5:** »Modeling base skill«: Die vorliegenden abstrakten Satzstrukturen muss der Systemanalyst durch die Anwendung einer ausgewählten Modellierungstechnik in ein konzeptuelles Modell überführen (oder mit diesem abgleichen) können.
- A6:** »Generation base skill«: Da das Modell möglicherweise auch neue Domänenaussagen ermöglicht, müssen Systemanalysten diese generieren können (um sie vom Domänenexperten bestätigen oder widerlegen zu lassen).
- A7:** »Fundamental base skill«: Der Systemanalyst muss durch abstraktes Denken dazu in der Lage sein, die Qualität des Modells in Bezug auf die gestellten Anforderungen zu lenken.
- A8:** »Impersonation base skill«: Die Fähigkeit des Systemanalysten, sich in die Welt des Domänenexperten hineinversetzen zu können, ist nicht notwendig, wirkt sich aber positiv auf das Ergebnis des Modellierungsprozesses aus.

Anhand dieser Liste an Kompetenzen wird deutlich, dass im Verlauf des Modellierungsprozesses neben der reinen Modellbildung (im Sinne von A5 - »Modeling base

⁵ Dabei beschreiben insbesondere die Fähigkeiten A1-A7 aufeinander aufbauende Schritte des Modellierungsprozesses. Eine entsprechende Übersichtsgrafik ist in [FW06, S. 12] zu finden.

⁶ In [FW06] werden die Domänenaussagen als sogenannte »sample sentences« bezeichnet, die ein Domänenexperte zur Beschreibung der Domäne (»universe of discourse«) in natürlicher Sprache erstellt.

3. Verwandte Arbeiten

Tab. 3.1.: Zentrale Lernziele für die Modellierung aus [Gl08, S. 427]

<p>These 2. Die Studierenden sollen die grundlegenden, gebietsübergreifenden Phänomene der Modellierung lernen (Theorie, Grundsätze der Modellbildung, Definition und Manipulation von Modellen, Metamodelle).</p> <p>These 3. Die Studierenden sollen die Konzepte der Modellierung von Strukturen, Abläufen, Verhalten und Interaktion in Informatikproblemen verstehen und anwenden.</p> <p>These 4. Die Studierenden sollen lernen, welches die typischen Artefakte sind, die aus Modellen bestehen oder in denen Modelle eine zentrale Rolle spielen, sowie wann und wo diese Artefakte typisch entstehen und verwendet werden.</p> <p>These 5. Die Studierenden sollen in allen Teildisziplinen der Informatik, mit denen sie im Studium in Kontakt kommen, ihr Wissen über Modellierung auf Probleme der jeweiligen Teildisziplin anwenden.</p>	<p>These 6. Die Studierenden sollen sowohl das Lesen und Verstehen bestehender Modelle als auch die Bildung von Modellen lernen.</p> <p>These 7. Die Studierenden sollen die Möglichkeiten und Grenzen modellbasierter Verfahren kennen lernen.</p> <p>These 8. Die Studierenden sollen geeignete Modellierungssprachen in dem Umfang erlernen, der sie zur Anwendung des Konzeptwissens über Modellierung befähigt.</p> <p>These 9. Das Erlernen des Umgangs mit Modellierungswerkzeugen ist kein notwendiger Bestandteil der Lehre in Modellierung. Dies gilt primär für die universitäre Lehre, wo die Entwicklung konzeptioneller Fähigkeiten im Vordergrund steht.</p> <p>These 10. Als Voraussetzung für das Lernen von Modellierung benötigen die Studierenden ein Basiswissen in theoretischer Informatik, beispielsweise über Graphen, Automaten und Sprachen.</p>
--	--

skill«) viele weitere Fähigkeiten eine wichtige Rolle spielen. Die Untersuchung der Konsequenzen, die sich anhand dieser Kompetenzen für die Lehre ergeben, erfordert weitergehende Forschung. Dabei gilt es zu klären, wie diese Kompetenzen gemessen und gelehrt (bzw. erlernt) werden können [FW06, S. 17].

GLINZ [Gl08] formuliert zwölf zentrale Thesen für die Lehre der Modellierung in informatikbezogenen Studiengängen und bezeichnet die Thesen 2 bis 10 als zentrale Lernziele. Diese werden in Tab. 3.1 wiedergegeben und beziehen sich in erster Linie auf Inhalte, Voraussetzungen und den Anwendungskontext der Modellierung im Studium. Während der Modellierungsprozess an sich zwar in These 6 erwähnt (»Studierende sollen [...] die Bildung von Modellen lernen«) und das Erlernen von Modellierungssprachen gemäß These 8 gefordert wird, findet keine detailliertere Ausführung dieser beiden Punkte statt, die in Bezug auf die Forschungsfragen im Rahmen dieser Arbeit von Bedeutung sein könnte.

Erst in jüngerer Zeit wurden Arbeiten veröffentlicht, die sich didaktisch fundiert mit Lernzielen für das Gebiet der Modellierung befassen. In den Empfehlungen der GI [Ge16a, S. 29f] für Informatik-Studiengänge werden die Kompetenzen zur Modellierung anhand eines Kompetenzmodells vorgestellt. Die verwendete Darstellung (siehe Tab. 3.2) ist an die in Abschnitt 2.1.2 eingeführte Taxonomie von ANDERSON

Tab. 3.2.: Kognitive Kompetenzen zur Modellierung aus den GI-Empfehlungen für Informatikstudiengänge [Ge16a, S. 29f]

Prozessdimension	Stufe 1 Verstehen	Stufe 2 Anwenden	Stufe 3 Analysieren
Geringe Kontextualisierung und Komplexität	Formale Beschreibungen von Aufgaben und Systemeigenschaften in grundlegenden Kalkülen, wie z.B. strukturierten Mengen, Termen und Algebren, Aussagen- und Prädikatenlogik sowie Graphen, interpretieren und erläutern. Von wichtigen spezialisierten Modellierungskalkülen, wie z. B. endliche Automaten, kontextfreie Grammatiken und Petri-Netze sowie den Sprachen der UML, die Beschreibungsformen und die Bedeutung interpretieren und erläutern.	Aufgaben und Systemeigenschaften auf ihren konzeptionellen Kern abstrahieren und durch ein Modell in informeller oder formaler Notation präzise und vollständig beschreiben. Verschiedene Sichten auf ein System mit passenden Modellen darstellen.	Formale Beschreibungen mittel- großer Aufgabenstellungen analysieren und die spezifizierten Eigenschaften ermitteln. Überprüfen, ob ein Modellkandidat vorgegebene Anforderungen erfüllt.
Starke Kontextualisierung und hohe Komplexität		Stufe 2a Übertragen	Stufe 3a Bewerten
		Eine vorgegebene Aufgabenstellung größerer Komplexität mit passenden Kalkülen auf angemessenem Abstraktionsgrad zielgerichtet formal beschreiben. Dabei professionelle Werkzeuge einsetzen und Ergebnisse bewerten. Allgemeine, abstrakte Fragestellungen und Zielsetzungen in konkrete Spezifikationen und Analysefragen umsetzen.	Praxisrelevante Modellierungskalküle und -werkzeuge auf ihre Eignung für eine konkrete Anwendung prüfen, beurteilen und auswählen.

und KRATHWOHL [AK01] angelehnt⁷. Dabei wurde die Wissensdimension durch die Charakterisierung von Komplexität und Anwendungskontext ersetzt, woraus sich

⁷ Eine genaue Beschreibung und Begründung für die Reduktion der Anderson-Krathwohl-Matrix ist in [Ge16a, S. 67f] zu finden. Darüber hinaus besitzt die Tabelle im Original noch eine Spalte für die Stufe 4 – Erzeugen. Diese Spalte ist allerdings im Kompetenzmodell zur Modellierung ohne Inhalt und wurde daher bei der Wiedergabe in dieser Arbeit aus Platzgründen entfernt.

3. Verwandte Arbeiten

für die Kompetenzen der Stufen 2 und 3 eine Feinabstufung in die Stufen 2a bzw. 3a ergibt. Die Erstellung von Modellen – verortet in Stufe 2: Anwenden – kann sich demnach hinsichtlich ihrer Kontextualisierung und ihrer Komplexität (z. B. kleine Beispiele vs. große betriebliche Projekte) unterscheiden. Im Kompetenzmodell der GI werden einige der bereits zuvor beschriebenen Konzepte aufgegriffen, z. B. die notwendige Abstraktion auf einen »konzeptionellen Kern« (analog z. B. zum »abstraction base skill« in [FW06]) oder die Beschreibung eines Modells in Form einer Notation (vgl. Modell vs. Modelldenotation im FRISCO-Framework [Fa98]). Auch beinhaltet die Formulierung der Kompetenz zur Erstellung von Modellen (siehe Tab. 3.2, Stufe 2: Anwenden) detailliertere Angaben als die These 6 in [GI08] (siehe Tab. 3.1).

Sowohl BOGDANOVA und SNOECK [BS17; BS19] als auch BORK [Bo19] greifen ebenso wie die GI-Empfehlungen [Ge16a; Ge16b] die Lernzieltaxonomie von ANDERSON und KRATHWOHL [AK01] auf, um diese für das Gebiet der konzeptuellen Modellierung anzupassen. In [BS17; BS19] werden Lernziele der Wissensdimension für die meisten Prozesstufen der Anderson-Krathwohl-Matrix mit Inhalten für die konzeptuelle Datenmodellierung verknüpft. Somit kann jedes Lernziel einem Inhaltsbereich, und einer Ebene auf der Wissens- beziehungsweise Prozessdimension zugeordnet werden. Tab. 3.3 zeigt alle Lernziele für den Inhaltsbereich »Modelle«, der auf der Wissensdimension zur Ebene des prozeduralen Wissens gehört. Weitere 12 Tabellen für zusätzliche Inhaltsbereiche und Ebenen der Wissensdimension sind in [BS19, S. 98ff] aufgeführt. Zusätzlich werden zwei Übersichten vorgestellt, die aufeinander aufbauende Konzepte und Kursinhalte hierarchisch klassifizieren, um den Lernprozess entsprechend gestalten zu können (siehe [BS19, S. 95]). Diese Ergebnisse können zwar einen Beitrag dazu leisten, Lehrende bei der Anpassung bestehender Kurse oder der Erstellung neuer Kurse zu unterstützen, allerdings sind sie auf das Gebiet der Datenmodellierung limitiert.

In der Arbeit von BORK [Bo19] werden die Gebiete konzeptuelle Modellierung (allgemein) sowie Metamodellierung (speziell) betrachtet und entsprechende Lernziele auf der Anderson-Krathwohl-Matrix definiert. Tab. 3.4 stellt die Lernziele für das Gebiet der konzeptuellen Modellierung anhand der Stufen der kognitiven Prozessdimension vor. Eine Tabelle analog für das Gebiet der Metamodellierung ist in [Bo19] zu finden. In der Arbeit wird eine Lehrveranstaltung zur Metamodellierung anhand der Taxonomie evaluiert und dabei bestehende Lücken aufgezeigt.

Aus den in diesem Abschnitt vorgestellten Arbeiten lassen sich mindestens zwei wegweisende Erkenntnisse ableiten. Erstens besteht weitgehender Konsens dahingehend,

Tab. 3.3.: Beispiele für Lernziele mit Bezug auf das Erstellen eines Modells (Inhaltsbereich: Modelle, Ebene: Prozedurales Wissen), [BS19, S. 99].

Schlüsselwörter:	Modellierungsrichtlinien, einfaches Modell, komplexes Modell, Anforderungen
Prozessstufe:	Studierende sollten dazu in der Lage sein...
Erinnern	...die Schritte zum Erstellen eines Modells wiederzugeben
Verstehen	...die Schritte zum Erstellen eines Modells zu diskutieren und die Bedeutung jedes Schritts zu belegen
Anwenden	...die ersten drei (n) Schritte zur Erstellung eines Modells auszuführen, bei gegebenen detaillierten Anforderungen
Analysieren	...einen vollständig gelösten Fall zu untersuchen und dabei die Schritte der Modellierung hervorzuheben, die während der Lösung durchgeführt wurden
Evaluieren	...zu prüfen, ob andere Studierende beim Modellieren die Modellierungsrichtlinien eingehalten haben, um die Anschaulichkeit der Richtlinien zu evaluieren
Erschaffen	... Modellierungsrichtlinien auf Basis eigener Erfahrungen zu verbessern

dass Modellierungskompetenzen und methodisches Wissen zur Modellierung in der Lehre einen höheren Stellenwert besitzen sollten. In früheren Arbeiten (z. B. [GI08; RR09]) wird der Fokus zumeist stark auf Inhalte und technisches Wissen (Modellierungsmethoden, -sprachen, konkrete Modelle) gelegt, erst jüngere Arbeiten (z. B. [Bo19; BS17; BS19; Ge16a; Ge16b]) greifen didaktische Instrumente wie die Anderson-Krathwohl-Matrix [AK01] auf und rücken Lernziele und zu entwickelnde Kompetenzen stärker ins Blickfeld. Zweitens macht die Betrachtung von Fähigkeiten und Kompetenzen zur Modellierung deutlich, dass es bei der Gestaltung entsprechender Lehrveranstaltungen eine Reihe von Aspekten zu berücksichtigen gilt. Beispielsweise kann ein »consistency base skill« [FW06] nur dann entwickelt werden, wenn entsprechende Modellierungsaufgaben mit widersprüchlichen Domänenaussagen in der Lehre eingesetzt werden. Ein Abgleich eigener Lehrveranstaltungen mit den Lernzieltaxonomien von BORK [Bo19] und BOGDANOVA und SNOECK [BS19] unterstützt Lehrende dabei, derartige Lücken zu identifizieren. Allerdings sind die Lernziele in den hier vorgestellten Taxonomien entweder sehr abstrakt [Ge16a; Ge16b] oder sie beschränken sich auf einen spezifischen Anwendungsbereich der Modellierung [Bo19; BS19].

3.1.2. Aufgabentypen

Ein Aufgabentyp fasst Aufgaben mit gleicher Struktur aber unterschiedlichen Inhalten zu einer gemeinsamen Klasse zusammen. Bisher existieren nur wenige Arbeiten,

3. Verwandte Arbeiten

Tab. 3.4.: Zuordnung von Lernzielen auf dem Gebiet der konzeptuellen Modellierung zu den Prozessdimensionsstufen, leicht gekürzt nach [Bo19, S. 7705].

Dimension	Konzeptuelle Modellierung
Erinnern	Das Ziel ist es, Kenntnisse über Syntax, Semantik und Notation einer Modellierungssprache sowie über [...] eine Modellierungsmethode zu besitzen.
Verstehen	Um ein konzeptuelles Modell zu verstehen, müssen die Lernenden die Bedeutung der einzelnen Elemente eines Modells sowie die Bedeutung der Beziehungen zwischen diesen Elementen kennen. Der Fokus dieser Prozessstufe liegt auf den semantischen Aspekten einer Modellierungssprache.
Anwenden	Lernende können eine Modellierungsmethode auf eine bestimmte Art und Weise anwenden, die einen bestimmten Zweck erfüllt oder ein vorliegendes Problem löst. Die Lernenden können unter mehreren gegebenen Modellen auswählen, welches für einen bestimmten Fall am besten geeignet ist, oder sie können ein vordefiniertes Modell verwenden, um Code zu generieren (z.B. SQL-Code aus ER-Modellen generieren), oder es nach einer vordefinierten Prozedur in ein anderes Modell umwandeln, indem sie eine Reihe von Modelltransformationen anwenden.
Analysieren	Die Analyse eines konzeptuellen Modells kann sowohl auf syntaktischer als auch auf semantischer Ebene erfolgen, d.h. die Analyse der Verwendung bestimmter syntaktischer Elemente oder die semantische Analyse größerer Modelle, gegebenenfalls in verknüpfte Untermodelle zerlegt. Daher liegt der Fokus dieser Prozessstufe auf der Syntax und Semantik einer Modellierungssprache.
Evaluieren	Auf dieser Ebene werden ebenfalls die Syntax und Semantik eines Modells untersucht, jedoch mit dem Ziel, die Analyseergebnisse auf gegebene Kriterien zu beziehen. Die syntaktische Evaluation eines konzeptuellen Modells kann z. B. durch Verwendung von Kardinalitäten und den erlaubten Beziehungen zwischen Klassen durchgeführt werden. Eine semantische Evaluation kann untersuchen, ob ein gegebenes Modell bestimmte Aspekte der Realität [...] korrekt und umfassend darstellt oder ob zwei oder mehrere Modelle zueinander konsistent sind. Für eine umfassende Evaluation der Modellqualität kann auf die Qualitätskriterien von [KLS95] verwiesen werden.
Erschaffen	Diese Prozessstufe betrifft die Erstellung eines neuen konzeptuellen Modells. Daher müssen alle Aspekte einer Modellierungsmethode berücksichtigt werden. Die Lernenden müssen in der Lage sein, die richtigen syntaktischen Elemente auszuwählen, die zur Semantik des zu modellierenden Sachverhalts passen, die richtige Notation zu verwenden und sie mit anderen Elementen mittels der korrekten Beziehungen zu verbinden. Bei komplexeren Modellierungsmethoden kann die Erstellung auch das Befolgen einer vordefinierten Prozedur [...] beinhalten.

die sich mit Aufgabentypen speziell auf dem Gebiet der grafischen Modellierung befassen.

BRINDA [Br04] leitet sogenannte Aufgabenklassen als Schablonen für die Erstellung neuer Aufgaben auf dem Gebiet der objektorientierten Modellierung durch die Analyse von 320 Aufgaben ab. Dabei wurden 31 grundlegende und 28 komplexe Aufgabenklassen extrahiert. Ebenso werden 12 Aufgabentypen vorgestellt und mit der Lernzieltaxonomie von [Bl73] kombiniert. Diese Aufgabentypen sind allerdings

sehr abstrakt und ohne erkennbaren Bezug zur Modellierung gehalten, z. B. »Beschreibung«, »Spezifikation«, »Zuordnung«, »Analyse«, »Vergleich«, »Identifikation«, »Diskussion«, »Konstruktion« [Br04, S. 123].

Eine weitere Sammlung von Aufgabentypen, ebenfalls speziell für das Gebiet der objektorientierten Modellierung wurde von BRANDSTEIDL et al. [BSK09] vorgestellt. Hier werden vier abstraktere Aufgabentypen (»Multiple-Choice-Aufgaben«, »Fehler finden«, »Verstehen von Sequenzen« und »Modellieren von Diagrammen«) eingeführt, denen jeweils einige Varianten und insgesamt 13 Beispiele zugeordnet werden. BOGDANOVA und SNOECK [BS17] stellen 12 häufige Aufgabentypen auf dem Gebiet der Unternehmensmodellierung vor. Diese sind allerdings teilweise sehr spezifisch und explizit auf Klassendiagramme ausgerichtet, z. B. »Schlagen Sie auf Basis des Anforderungsdokuments Attribute für die verschiedenen Klassen vor« [BS17, S. 7].

Im Rahmen dieses Abschnitts soll nun eine Übersicht über grundlegende Aufgabentypen auf dem Gebiet der Geschäftsprozessmodellierung erstellt und deren Verteilung erfasst werden⁸. Zu diesem Zweck wurden anhand des Katalogs der Universitätsbibliothek des KIT vier Lehrbücher zu den Suchbegriffen Geschäftsprozessmanagement und Geschäftsprozessmodellierung ermittelt, die Übungsaufgaben beinhalten⁹:

- van der AALST, W. M. P. und STAHL, C.: Modeling Business Processes. A Petri Net-Oriented Approach. The MIT Press, 2011 [AS11]
- DUMAS, M.; LA ROSA, M.; MENDLING, J. und REIJERS, H. A.: Fundamentals of Business Process Management. Springer, 2013 [Du13] (Kapitel 3 »Essential Process Modeling« und Kapitel 4 »Advanced Process Modeling«, S. 63-152)
- DRESCHER, A.; KOSCHMIDER, A. und OBERWEIS, A.: Modellierung und Analyse von Geschäftsprozessen: Grundlagen und Übungsaufgaben mit Lösungen. De Gruyter Studium, 2017 [DKO17]
- GADATSCH, A.: Grundkurs Geschäftsprozess-Management. 9. Aufl., Springer, 2020 [Ga20] (Kapitel 5 »Modellierung und Analyse von Prozessen«, S. 87-150)

Aus diesen Lehrbüchern wurden alle Übungsaufgaben herausgesucht, die sich auf Modelle (ausschließlich in Form von Modelldenotationen) beziehen. Das heißt, es wird entweder ein Modell in der Aufgabenstellung angegeben, oder die Aufgabe

⁸ Die hier vorgestellte Untersuchung und ihre Ergebnisse wurden später in [So23b] veröffentlicht.

⁹ Tatsächlich geben nur wenige Lehrbücher Übungsaufgaben an. AUBERTIN et al. [Au12] untersuchen den Stand der Lehrbuchliteratur zum Geschäftsprozessmanagement und stellen fest, dass von 9 Lehrbüchern nur das Buch von GADATSCH [Ga20] (in der 6. Auflage von 2010) Übungsaufgaben enthält.

3. Verwandte Arbeiten

Tab. 3.5.: Verteilung von Aufgabentypen in Lehrbüchern zur Geschäftsprozessmodellierung

Lehrbuch/Aufgabentyp	Modell erläutern	Fehler im Modell finden	Modell erstellen	Modell analysieren	Modell verändern
VDAALST und STAHL [AS11]	7	0	24	17	8
DUMAS et al. [Du13]	5	2	26	3	8
DRESCHER et al. [DKO17]	17	20	47	27	6
GADATSCH [Ga20]	3	0	2	0	0
Summe (Anteil)	32 (14%)	22 (10%)	99 (45%)	47 (21%)	22 (10%)

erfordert die Angabe eines Modells im Rahmen der Lösung. Somit wurden beispielsweise reine Wissensabfragen wie z. B. »Beschreiben Sie die Grundelemente der EPK« [DKO17, S. 51] von der Betrachtung ausgeschlossen. Die so ermittelten Übungsaufgaben konnten anhand fünf verschiedener Klassen gruppiert werden, die jeweils einen bestimmten Aufgabentyp repräsentieren. Tab. 3.5 gibt Aufschluss über die Verteilung dieser Aufgabentypen über die betrachteten Lehrbücher hinweg. In den folgenden Absätzen werden die fünf Aufgabentypen aus Tab. 3.5 charakterisiert und anhand von jeweils einem repräsentativen Beispiel vorgestellt¹⁰.

Modell erläutern Bei Aufgaben dieses Typs ist ein Modell gegeben, dessen Aussagen in der Regel als Freitext in natürlicher Sprache beschrieben werden sollen (siehe Abb. 3.2). Ist das Ergebnis der Aufgabe ein solcher Freitext, handelt es sich um eine divergente Aufgabe (siehe Abschnitt 2.1.3). Allerdings ist es auch möglich, diesen Aufgabentyp in Form einer konvergenten Aufgabe zu stellen, z. B. als Antwort-Wahl-Aufgabe, bei der zum Modell passende Aussagen ausgewählt werden müssen.

Fehler im Modell finden Bei diesem Aufgabentyp wird gefordert, zu einem gegebenen Modell eine festgelegte Art von Fehlern zu identifizieren. Die Fehler sollen dabei z. B. im Modell markiert werden (siehe Abb. 3.3), oft soll zusätzlich eine kurze Beschreibung mit Begründung erfolgen. Meist handelt es sich bei den zu identifizierenden Fehlern um Verstöße gegen syntaktische Regeln der Modellierungssprache,

¹⁰ Für das Beispiel des in dieser Arbeit zentralen Aufgabentyps »Modell erstellen« wurde zusätzlich ein Lösungsvorschlag angegeben, bezüglich möglicher Lösungen für die anderen Aufgabentypen sei der interessierte Leser auf [DKO17] verwiesen.

Aufgabe: Beschreiben Sie den folgenden in BPMN modellierten Geschäftsprozess in natürlicher Sprache.

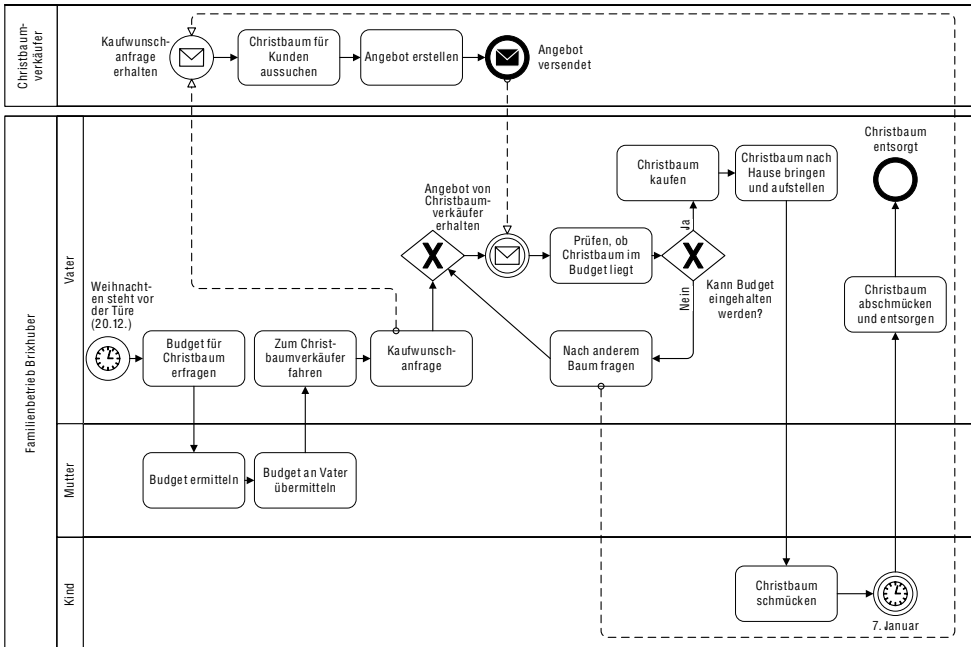


Abb. 3.2.: Beispiel zum Aufgabentyp »Modell erläutern« aus [DKO17, S. 70]

in der das Modell erstellt wurde. Ist die Menge der Markierungsmöglichkeiten dabei geschlossen, so handelt es sich um eine konvergente Aufgabe. Bezüglich der Begründungen gilt analog zum Aufgabentyp »Modell erläutern«, dass eine Umsetzung in Form einer konvergenten Aufgabe nur möglich ist, wenn z. B. verschiedene Fehlerbeschreibungen im Rahmen einer Antwort-Wahl-Aufgabe zur Auswahl stehen.

Modell erstellen Hier ist typischerweise die textuelle Beschreibung eines Modellierungsgegenstands in natürlicher Sprache gegeben, der durch die Anwendung einer festgelegten Modellierungssprache in ein Modell überführt werden soll (siehe Abb. 3.4). Diese textuelle Beschreibung wird im weiteren Verlauf dieser Arbeit auch als Szenario bezeichnet. Da es z. B. für die Anordnung und Beschriftung der Modellelemente oder für die gewählte Abstraktionsebene bei der Modellierung sehr viele unterschiedliche Möglichkeiten gibt, existiert nicht nur eine einzige richtige Lösung für diesen divergenten Aufgabentyp mit hohem Freiheitsgrad. Eine Reduktion des

3. Verwandte Arbeiten

Aufgabe: Die nachfolgende erweiterte Ereignisgesteuerte Prozesskette enthält Modellierungsfehler. Markieren Sie die Fehler im Diagramm.

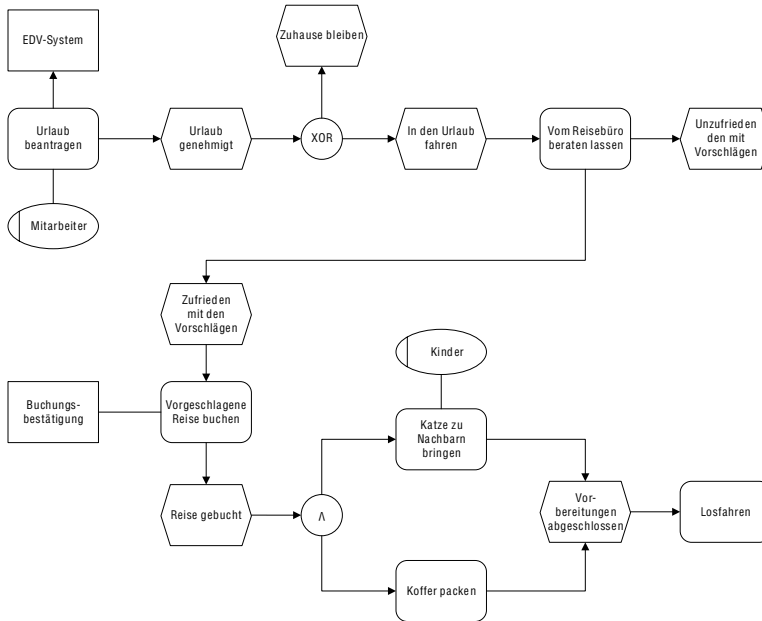


Abb. 3.3.: Beispiel zum Aufgabentyp »Fehler im Modell finden« aus [DKO17, S. 62]

Freiheitsgrads z. B. durch die Vorgabe bestimmter Beschriftungen oder durch eine sehr präzise Beschreibung des Szenarios ist möglich. Allerdings sinken dabei auch Anspruch und Komplexität im Kontext der erforderlichen Kompetenzen. Im Zusammenhang mit Aufgaben des Typs »Modell erstellen« dient häufig eine sogenannte Musterlösung als Vorschlag für ein mögliches korrektes Modell. Davon abweichende Modelle können aber bei Erfüllen aller durch das gewählte Bewertungsschema festgelegten Anforderungen ebenfalls die volle Punktzahl erreichen.

Modell analysieren Unter diesem Aufgabentyp wurden Aufgaben zusammengefasst, bei denen ein gegebenes Modell in Bezug auf festgelegte Eigenschaften analysiert oder in Bezug auf bestimmte Kriterien bewertet werden soll. Auch die Anwendung von Analysemethoden (z. B. Erstellung eines Überdeckungs- oder Erreichbarkeitsgraphen für ein Petri-Netz) gehört zu diesem Aufgabentyp. Erwartungsgemäß finden sich die meisten Aufgaben dieser Art sowie auch das entsprechende

Aufgabe: Gegeben sei die folgende Ablaufbeschreibung:

›Studierende eines bestimmten Studiengangs müssen sich im Rahmen ihres Studiums in den beiden Fächern Modellierung von Geschäftsprozessen (MvG) und Workflow-Management (WFM) prüfen lassen, wobei die Reihenfolge der Prüfungen keine Rolle spielt. Vorher lernen Studierende für die Fächer. Die Prüfungen werden mit bestanden oder mit nicht-bestanden bewertet. Einem Studierenden wird ein Zeugnis ausgestellt, falls beide Prüfungen bestanden wurden. Ansonsten muss die entsprechende Prüfung bzw. die beiden Prüfungen erneut geschrieben werden.«

Modellieren Sie den Ablauf als Petri-Netz und geben Sie eine geeignete Startmarkierung an.

Musterlösung:

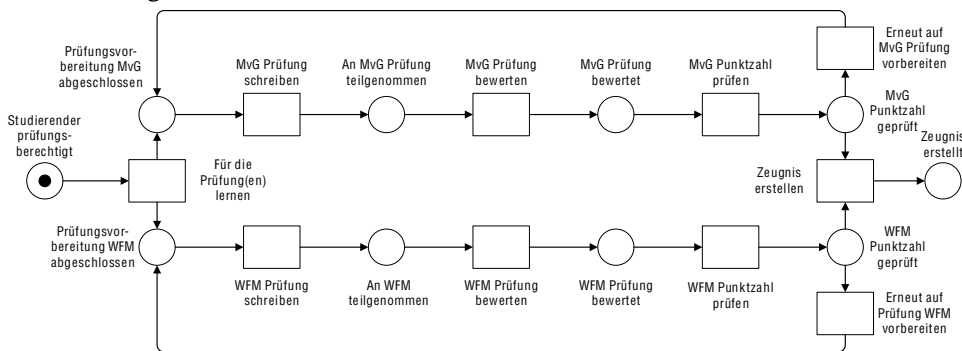


Abb. 3.4.: Beispiel zum Aufgabentyp »Modell erstellen« aus [DKO17, S. 97, S. 171]

Beispiel in Abb. 3.5 im Kontext von Petri-Netzen¹¹, für welche zahlreiche formale Eigenschaften und Analysemethoden definiert wurden (siehe Abschnitt 2.2.4). Aufgaben zur Überprüfung von Eigenschaften oder Kriterien können zwar grundsätzlich als konvergente Aufgabe umgesetzt werden, solange sich die Lösung eindeutig bestimmen lässt. Häufig soll aber zusätzlich eine Begründung in natürlicher Sprache erfolgen. Für Aufgaben zur Anwendung von Analysemethoden gilt, dass bei der Bewertung des Ergebnisses (z. B. eines Überdeckungsgraphen) auch der Lösungsweg bzw. anteilig richtige Lösungen berücksichtigt werden müssen - es handelt sich dann wie auch bei Aufgaben zum »Modell erstellen« um divergente Aufgaben - obwohl hier im Gegensatz zu diesen eine eindeutige Lösung existiert.

¹¹ Nach der Auszählung in Tab. 3.5 stammen $\approx 94\%$ der Aufgaben diesen Typs aus den beiden Lehrbüchern, die sich umfassender mit Petri-Netzen auseinandersetzen ([AS11; DKO17]).

3. Verwandte Arbeiten

Aufgabe: Überprüfen Sie, welche der folgenden Aussagen auf das gegebene Netz zutreffen:

- Das Netz ist schwach und stark zusammenhängend.
- Das Netz erfüllt die free-choice Eigenschaft.
- Das Netz ist ein Workflow-Netz.
- Das Netz kann in s-Komponenten zerlegt werden, so dass jede Stelle in einer s-Komponente enthalten ist.

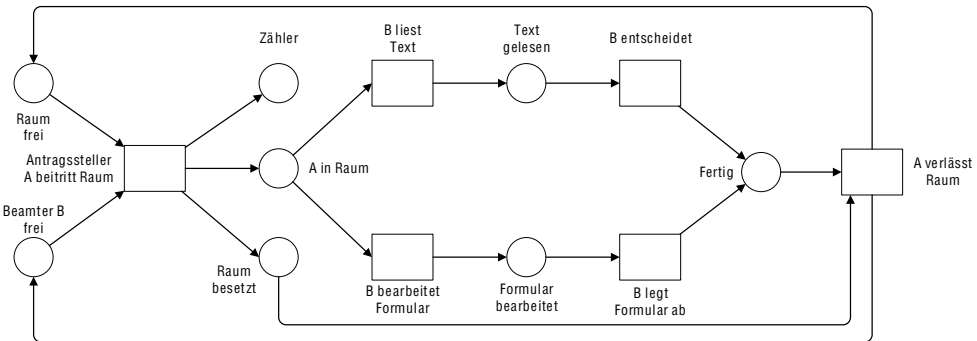
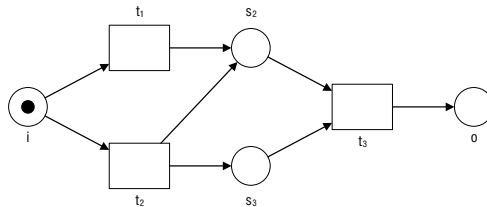


Abb. 3.5.: Beispiel zum Aufgabentyp »Modell analysieren« aus [DKO17, S. 232f]

Modell verändern Aufgaben, die in dieser Klasse zusammengefasst wurden, besitzen die gemeinsame Eigenschaft, dass ein gegebenes Modell auf eine bestimmte Art und Weise verändert werden soll (»Reparieren Sie ...«, »Erweitern Sie ...«, »Passen Sie das Modell an ...«). Dadurch sollen z. B. zusätzliche Informationen zum Sachverhalt durch das Modell ausgedrückt oder die Erfüllung bestimmter Eigenschaften hergestellt werden. Häufig wird nur die Erweiterung, also das Hinzufügen von Modellelementen zum bestehenden Modell gefordert (siehe Abb. 3.6), das Entfernen oder die Anpassung bestehender Modellelemente kann aber auch nötig sein. Bei Aufgaben diesen Typs ist das Ergebnis wiederum ein Modell. Der Grad der Divergenz dieser Aufgaben hängt stark davon ab, welchen Umfang die durch die Aufgabenstellung geforderten Änderungen besitzen und welcher Freiheitsgrad bei der Umsetzung vorliegt.

In dieser Arbeit steht der Aufgabentyp »Modell erstellen« im Fokus. Dieser ist gleichzeitig der am häufigsten auftretende Aufgabentyp auf dem Gebiet der Geschäftsprozessmodellierung (siehe Tab. 3.5). Dies ist wenig überraschend, da die Modellbildung definitionsgemäß wesentlicher Bestandteil der Modellierung ist (siehe Def. 2.12). Die Fähigkeit bzw. Kompetenz zur Modellbildung wird auch in den

Aufgabe: Gegeben sei das folgende markierte Workflow-Netz:



Erweitern Sie dieses markierte Workflow-Netz so, dass es die Eigenschaft sound erfüllt. Sie dürfen dabei keine Stellen, Transitionen oder Kanten entfernen.

Abb. 3.6.: Beispiel zum Aufgabentyp »Modell verändern« aus [DKO17, S. 236]

zuvor betrachteten fachdidaktischen Arbeiten thematisiert (z. B. [Ge16a; Gl08]). Allerdings birgt die Automatisierung der Bewertung für Aufgaben zum Typ »Modell erstellen« aus wissenschaftlicher Sicht die größten Herausforderungen, da es sich dabei grundsätzlich um divergente Aufgaben handelt. Diesen Herausforderungen möchte sich das Forschungsvorhaben stellen. Für Aufgaben zu den anderen Typen ist weitestgehend auch eine Umsetzung als konvergente Aufgabe möglich, so dass die Automatisierung der Bewertung hier vergleichsweise unkompliziert ist. Sie werden daher im Rahmen der in Forschungsfrage FF2 entwickelten E-Assessment-Plattform nicht berücksichtigt, bieten aber einen wertvollen Ausgangspunkt für die zukünftige Erweiterung.

Offen ist zudem die Zuordnung von Aufgabentypen zu konkreten Lernzielen bzw. Kompetenzen. Im Zusammenhang mit der Entwicklung eines Bewertungsschemas für den Aufgabentyp »Modell erstellen« gilt es zu klären, welche Lernziele sich anhand des Ergebnisartefakts – des studentischen Modells – grundsätzlich überprüfen lassen. Dieser Aspekt wird in der vorliegenden Arbeit im Kontext von Forschungsfrage FF1 adressiert (siehe Abschnitt 4.2).

3.2. Modellqualität

Die Norm IEC 2371 definiert Qualität als den Grad der »Übereinstimmung zwischen den festgestellten Eigenschaften und den vorher festgelegten Anforderungen an eine Betrachtungseinheit« [Ti16, S. 563]. Welche Anforderungen für die jeweilige Betrachtungseinheit gelten, ist immer bezogen auf einen bestimmten Anwendungszweck

3. Verwandte Arbeiten

oder eine bestimmte Zielgruppe zu ermitteln und daher nicht eindeutig. Auch für ein Modell gilt BECKER et al. [BPV12, S. 32] zufolge, dass es nicht entweder richtig oder falsch ist, sondern dahingegen »mehr oder weniger zweckmäßig und [...] damit in den Augen des Modellnutzers eine höhere oder weniger hohe Qualität« aufweist. Neben der anwendungsbezogenen Zweckmäßigkeit eines Modells (siehe z. B. die Beispiele für Einsatzzwecke von Geschäftsprozessmodellen in Abschnitt 2.2.1) gibt es jedoch eine ganze Reihe weiterer Aspekte, die aus Qualitätsgesichtspunkten von Bedeutung sind. Eine Vielzahl an Forschenden auf dem Gebiet der Modellierung befasst sich daher mit Überlegungen zu relevanten Aspekten der Modellqualität, die in Form von sogenannten Qualitätsframeworks und -metriken sowie Modellierungsrichtlinien zur Erstellung qualitativ hochwertiger Modelle präsentiert werden. In den folgenden drei Abschnitten werden prominente Arbeiten aus diesen Bereichen vorgestellt.

3.2.1. Qualitätsframeworks

Im Kontext der Modellierung sollen Qualitätsframeworks ein Mittel zur Strukturierung und Bestimmung der Modellqualität darstellen [Mo05a]. Derartige Ansätze können OVERHAGE et al. [OBS12] zufolge den sogenannten »analytischen Qualitätssicherungsansätzen« zugeordnet werden¹². Im Zentrum analytischer Qualitätssicherungsansätze stehen Maßnahmen, die dazu dienen, das Qualitätsniveau eines Untersuchungsgegenstands zu messen, beispielsweise anhand von Kennzahlssystemen [OBS12, S. 218].

Zu den einflussreichsten Ansätzen mit dem Ziel, die Modellqualität konzeptueller Modelle zu bestimmen, zählt das »Semiotic Quality Framework« (SEQUAL-Framework). Es wurde ursprünglich von LINDLAND ET AL. [KLS95; LSS94] entworfen und von KROGSTIE ET AL. [KJ02; Kr12; KSJ06] weiterentwickelt. Es greift in seiner ursprünglichen Version zunächst die fundamentale Unterscheidung der Semiotik zwischen den drei Aspekten Syntaktik, Semantik und Pragmatik auf, die auf MORRIS [Mo88]¹³ zurückgeht [LSS94]. In [KLS95] werden zusätzliche semiotische Aspekte aus dem FRISCO-Framework [Fa98]¹⁴ integriert. Abb. 3.7 zeigt einen Ausschnitt über zentrale Elemente des umfangreichen SEQUAL-Frameworks nach [KJ02; KSJ06]. Im

12 Das Konzept der generellen Einteilung von Qualitätmaßnahmen in konstruktive und analytische Maßnahmen wird z. B. von BALZERT [Ba08] beschrieben

13 Die Quelle [Mo88] referenziert die deutsche Übersetzung des Werks von MORRIS [Mo88], das ursprünglich 1938 unter dem Namen »Foundations of the Theory of Signs« erschien.

14 In [KLS95] wird als Quelle eine »personal communication with the FRISCO task group« angegeben,

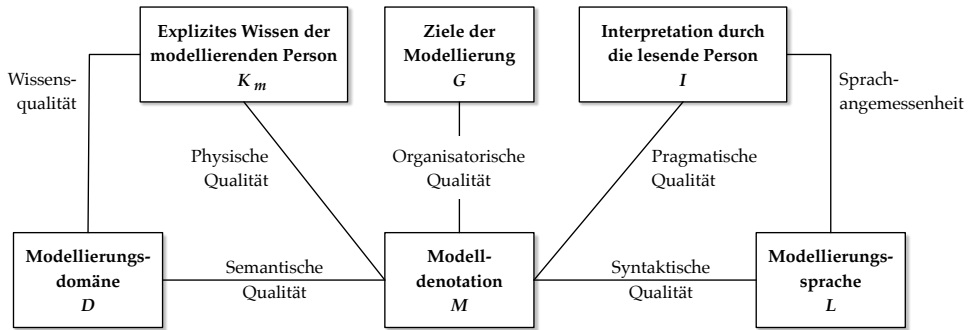


Abb. 3.7.: Ausschnitt aus dem SEQUAL-Framework, in Anlehnung an [KSJ06]

Zentrum steht die Modelldenotation¹⁵ M , die mit weiteren Elementen in Verbindung gebracht wird, um auf dieser Basis jeweils verschiedene Qualitätsaspekte zu beleuchten:

Syntaktische Qualität: Anhand des Vergleichs der Modelldenotation M mit der Syntax der verwendeten Modellierungssprache L wird die syntaktische Qualität bestimmt. Darunter fällt die Überprüfung der Verwendung der festgelegten Modellierungselemente und der Einhaltung der syntaktischen Regeln für die Anordnung dieser Elemente (wie in Abschnitt 2.2.3 definiert)¹⁶.

Semantische Qualität: Die Modellierungsdomäne D stellt den zu modellierenden Wirklichkeitsausschnitt dar. Mit Bezug auf die Domäne D kann die semantische Qualität anhand der **Validität** und **Vollständigkeit** der in der Modelldenotation M enthaltenen Aussagen bestimmt werden: Eine Modelldenotation M ist valide, wenn die enthaltenen Aussagen in Bezug auf die Domäne D korrekt (und relevant) sind. Eine Modelldenotation M ist vollständig, wenn sie alle Aussagen enthält, die durch die Domäne D gegeben werden.

Pragmatische Qualität Die pragmatische Qualität beschreibt die Übereinstimmung zwischen einer Modelldenotation M und deren Interpretation I durch die le-

das FRISCO-Framework wurde tatsächlich erst drei Jahre später veröffentlicht.

¹⁵ In [KJ02; KSJ06] wird der Begriff »Model externalization« verwendet. Da diese Formulierung impliziert, dass hiermit die externe Repräsentation eines Modells analog zum FRISCO-Framework [Fa98] gemeint ist, wurde der entsprechende Begriff Modelldenotation (gemäß Def. 2.11 auf S. 41) für die Übersetzung gewählt und in diesem Abschnitt weiterverwendet.

¹⁶ Unter anderem in [KLS95] werden (abstrakte) formale Definitionen für einige Qualitätsaspekte des SEQUAL-Frameworks angegeben. Z.B. wird die syntaktische Korrektheit definiert als $M \setminus L = \emptyset$, was bedeutet, dass alle Aussagen in der Modelldenotation M in der Sprache L enthalten (d.h. in Bezug auf diese syntaktisch korrekt) sind.

3. Verwandte Arbeiten

sende Person (bzw. die intendierte Zielgruppe). Die korrekte Interpretation der enthaltenen Aussagen wird einerseits durch die Modellierungskennntnisse der lesenden Person und andererseits durch die Eigenschaften der Modelldenotation M selbst beeinflusst. Zur Verbesserung der pragmatischen Qualität können beispielsweise die Lesbarkeit (z. B. durch ein übersichtliches Layout) und die Verständlichkeit (z. B. durch eindeutige Beschriftungen) der Modelldenotation M angehoben werden.

Organisatorische Qualität Die eingangs erwähnte Zweckmäßigkeit von Modellen wird durch die organisatorische Qualität¹⁷ abgebildet. Eine Modelldenotation M , die mit den festgelegten Zielen G der Modellierung übereinstimmt, erfüllt diesen Qualitätsaspekt. Beispielsweise sollte der gewählte Abstraktionsgrad der Modelldenotation M dem Anwendungszweck entsprechen (z. B. niedriger Abstraktionsgrad für eine lückenlose Dokumentation).

Physische Qualität Die physische Qualität drückt aus, inwieweit das explizite Wissen K_m des Modellierers über die Domäne D externalisiert wurde (z. B. analog auf einem Blatt Papier oder in digitaler Form) und der Zielgruppe in einer persistenten und verfügbaren Modelldenotation M zur Verfügung steht¹⁸.

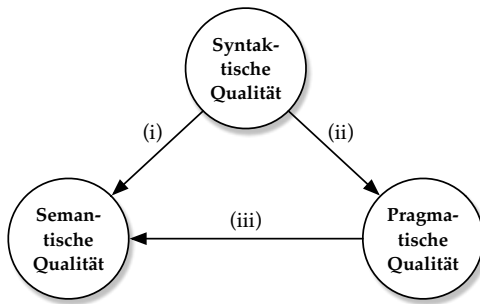
Durch Verbindungen zwischen den anderen Elementen (und nicht mit der Modelldenotation M) können weiterhin Qualitätsaspekte berücksichtigt werden, die zwar im Zusammenhang mit der Modellierung stehen, sich aber nicht auf ein konkretes Modell beziehen. Beispielsweise beschreibt die Wissensqualität den Kenntnisstand des Modellierers anhand dessen expliziten Wissens K_m über die Modellierungsdomäne D . Ob eine Modellierungssprache L angemessen ist, um die korrekte Interpretation I durch den Leser zu unterstützen, wird mittels der Sprachangemessenheit beschrieben¹⁹.

MOODY et al. [Mo02; Mo03] evaluieren die Anwendung des SEQUAL-Frameworks zur Untersuchung der Modellqualität von Prozess- und Informationsmodellen innerhalb der IS-Disziplin. Demnach stellen die syntaktische, semantische und pragmatische Qualität notwendige und angemessene Qualitätsaspekte dar, lediglich

¹⁷ Die organisatorische Qualität (siehe z. B. [KJ02; KSJ06]) wird in späteren Arbeiten (siehe z. B. [Kr12; Kr16]) als deontische Qualität bezeichnet.

¹⁸ Die Definition der physischen Qualität über die Verbindung der Modelldenotation M mit dem expliziten Wissen des Modellierers K_m entspricht der Darstellung in [KJ02; KSJ06], in [KLS95] wird die physische Qualität über die Verbindung von der Modelldenotation M zu sich selbst definiert.

¹⁹ In [KLS95] wird die Sprachangemessenheit (als sog. Sprachqualität) zusätzlich über die Verbindung zwischen der Modellierungssprache L und der Domäne D mit $D \setminus L = \emptyset$ definiert. Dies bedeutet, dass eine Modellierungssprache L dann angemessen ist, wenn ihre Ausdrucksmächtigkeit es erlaubt, alle in der Domäne D enthaltenen Aussagen korrekt wiederzugeben.



(i) Ein Modell, das syntaktische Fehler aufweist, enthält uneindeutige Aussagen.

(ii) Ein Modell, das syntaktische Fehler aufweist, ist schwer zu interpretieren.

(iii) Ein Modell, das schwer zu verstehen ist, kann nur schwer mit der Modellierungsdomäne verglichen werden

Abb. 3.8.: Wechselwirkungen zwischen syntaktischer, semantischer und pragmatischer Qualität, in Anlehnung an [Mo02, S. 387f]

bei der Anwendung des Frameworks werden Inkonsistenzen beobachtet. Dies ist dem Umstand geschuldet, dass das SEQUAL-Framework einen hohen Abstraktionsgrad aufweist und die Qualitätsaspekte nur vage beschrieben werden. In [Mo02; Mo03] werden darüber hinaus Wechselwirkungen zwischen den drei wesentlichen Qualitätsaspekten aufgezeigt, wie in Abb. 3.8 dargestellt. Diese Wechselwirkungen, insbesondere (i), stellen auch für die in dieser Arbeit entwickelten Bewertungsdienste eine Herausforderung dar. Ein Fehler auf der syntaktischen Ebene löst zwangsläufig einen Fehler auf der semantischen Ebene aus, der bei der Bewertung von Modellen im Kontext Hochschullehre gegebenenfalls als Folgefehler aufgefasst werden sollte. Allerdings ergeben sich dabei Interpretationsschwierigkeiten, wie auch in [OBS12, S. 225] festgestellt: »Beim Studium eines syntaktisch fehlerhaften Modells ist neben den Regelverstößen vom Modellnutzer jeweils außerdem zu ermitteln, was der Modellierer eigentlich darstellen wollte«.

Das 3QM-Framework (Qualitätsmerkmale, -metriken und -messverfahren) zur Bestimmung der Modellqualität von Geschäftsprozessmodellen von OVERHAGE et al. [OBS12] adressiert die von MOODY et al. [Mo02; Mo03] beschriebenen Schwachpunkte des SEQUAL-Frameworks. Es übernimmt aus diesem die drei wesentlichen semiotischen Qualitätsaspekte und orientiert sich weiterhin an den Vorschlägen von MOODY [Mo05a] für die hierarchische Gestaltung von Qualitätsframeworks für die konzeptuelle Modellierung. Als Ergebnis werden für die Hauptqualitätsmerkmale Syntax, Semantik und Pragmatik insgesamt 9 Untermerkmale und 35 Metriken mit entsprechenden Messverfahren festgelegt. Im Folgenden wird ein kurzer Überblick über die Qualitätsmerkmale des 3QM vorgestellt. Eine vollständige Angabe der dazugehörigen Metriken und Messverfahren ist in [OBS12, S. 226] zu finden.

3. Verwandte Arbeiten

Syntaktik Für das Qualitätsmerkmal Syntaktik definiert das 3QM-Framework die drei Untermerkmale Wortsyntax, Satzsyntax und Textsyntax²⁰. Anhand der **Wortsyntax** wird überprüft, ob im Modell ausschließlich Wörter verwendet werden, die im Lexikon der Modellierungssprache enthalten sind²¹. Die **Satzsyntax** untersucht, ob direkte Verbindungen zwischen zwei Wörtern nach den Regeln der Grammatik zulässig sind, wohingegen die **Textsyntax** auch indirekte Verbindungen berücksichtigt. Letzteres betrifft z. B. die Verwendung kompatibler Konnektoren bei der Aufspaltung bzw. Zusammenführung getrennter Ausführungspfade in Prozessmodellen.

Semantik Für das Qualitätsmerkmal Semantik werden die vier Untermerkmale Vollständigkeit, Korrektheit, Relevanz und Flexibilität eingeführt. Die **Vollständigkeit** ist (wie auch im SEQUAL-Framework) wie folgt definiert: Ein Modell ist vollständig, wenn es alle für das Verständnis des Wirklichkeitsausschnitts notwendigen Inhalte darstellt. Mithilfe der **Korrektheit** wird abgebildet, zu welchem Grad die Bedeutung der Modellelemente mit dem darzustellenden Wirklichkeitsausschnitt übereinstimmt, die **Relevanz** bildet ab, inwieweit die verwendeten Modellelemente tatsächlich für die Darstellung des Wirklichkeitsausschnitts notwendig sind (Korrektheit und Relevanz entsprechen zusammengenommen der Validität im SEQUAL-Framework). Speziell für die Geschäftsprozessmodellierung wird das Untermerkmal der **Flexibilität** eingeführt. Es bezieht sich darauf, ob Aktivitäten oder ganze Ausführungspfade, die voneinander unabhängig sind, in nebenläufiger Form modelliert werden, so dass die Ausführung flexibel erfolgen kann.

Pragmatik Das Qualitätsmerkmal Pragmatik besitzt zwei Untermerkmale: Eindeutigkeit und Verständlichkeit. Bei der **Eindeutigkeit** geht es um die Eingängigkeit der Modellinhalte. Damit ist gemeint, dass der Leser die Bedeutung des Modells möglichst eindeutig interpretieren kann und dabei keine Widersprüche auftreten. Die **Verständlichkeit** bezieht sich auf die Eingängigkeit des Sprachgebrauchs bei der Formulierung, d.h. z. B. inwieweit die Beschriftung von Modellelementen durch Synonyme variiert wird.

Für alle Untermerkmale werden Metriken und Messverfahren angegeben, die zu ermittelnde Maßzahlen und deren Zusammensetzung zur Messung der Ausprägung

²⁰ Diese wurden ebenfalls aus dem Gebiet der Semiotik entlehnt. So klassifiziert MORRIS [Mo88] die Morphologie (Wortsyntax) in die Unterkategorien der Satz- und Textsyntax [OBS12, S. 221].

²¹ Gemäß der in Abschnitt 2.2.3 festgelegten Terminologie sind mit Wörtern Modellelemente gemeint, das Lexikon der Modellierungssprache entspricht dem Alphabet *A* der Syntax.

des jeweiligen Untermerkmals definieren. Beispielsweise wird zum Qualitätsmerkmal Wortsyntax eine Metrik beschrieben, die den »originalen Verstoß« gegen die Regeln einer Modellierungssprache betrachtet. Das Messverfahren für diese Metrik gibt die Formel » $R=1-A/(\text{Anzahl insgesamt modellierter Sprachkonstrukte})$ « an, wobei A die »Anzahl erstmals nicht korrekt modellierter Sprachkonstrukte« bezeichnet [OBS12, S. 226]. Durch die Unterscheidung zwischen »originalen und wiederholten Verstößen« [OBS12, S. 225] kann eine unterschiedliche Gewichtung bei der Bewertung dieser Fehler erfolgen, was auch im Zusammenhang mit der Entwicklung von Bewertungsstrategien zur Bestimmung von Punktzahlen für den Kontext Hochschullehre von Bedeutung ist. Für die Gewichtung aller Qualitätsmerkmale wird in [OBS12] ein konkreter Vorschlag angegeben, so dass das 3QM-Framework sich direkt einsetzen lässt, um die Gesamtqualität eines Modells zu quantifizieren.

Dedizierte Qualitätsframeworks oder Bewertungsschemata für das Gebiet der Modellierung im Hochschulkontext existieren bislang nicht. Die hier vorgestellten Qualitätsframeworks liefern wichtige Anhaltspunkte für die Ableitung und Ausrichtung von Modellierungskompetenzen und für die Gestaltung von Bewertungsschemata. Allerdings sind dabei Anpassungen mit Bezug auf den Kontext Hochschullehre des Forschungsvorhabens notwendig. Hier gilt es insbesondere, das Erreichen der adressierten Lernziele zu überprüfen, so dass zunächst eine entsprechende Abbildung der beschriebenen Qualitätsaspekte auf Lernziele entwickelt werden muss. Ebenso basieren von Studierenden erstellte Modelle in der Regel auf einem fiktiven Sachverhalt, der anhand des Szenarios einer Modellierungsaufgabe vom Typ »Modell erstellen« präsentiert wird. So müssten beispielsweise die im SEQUAL-Framework eingeführten Ziele der Modellierung G in diesen fiktiven Sachverhalt einfließen, um Aussagen über die organisatorische Qualität treffen zu können. Darüber hinaus finden die Elemente des SEQUAL-Frameworks im vorliegenden Anwendungskontext teilweise keine passende Entsprechung. So ist z. B. davon auszugehen, dass bei Vorliegen einer textuellen Beschreibung der Modellierungsdomäne D , diese mit dem expliziten Wissen der modellierenden Person K_m gleichgesetzt werden kann. Die Anwendung des 3QM-Frameworks auf studentische Modelle ist zwar grundsätzlich möglich, allerdings muss untersucht werden, ob die für den Aufgabentyp »Modell erstellen« notwendigen Kompetenzen damit angemessen adressiert werden. Auch ist unklar, inwieweit sich die dort eingesetzten Metriken zur Bestimmung der Modellqualität im Rahmen von Forschungsfrage FF2 anhand von Bewertungsdiensten automatisiert umsetzen lassen.

3. Verwandte Arbeiten

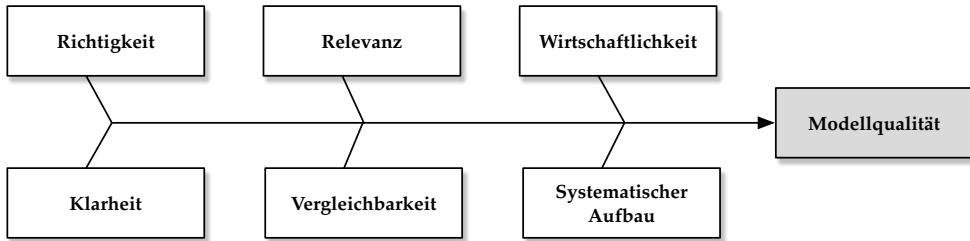


Abb. 3.9.: Übersicht über die Qualitätsaspekte der »Grundsätze ordnungsmäßiger Modellierung« (GoM), in Anlehnung an [BRU00, S. 32]

3.2.2. Modellierungsrichtlinien

Im Gegensatz zu den zuvor beschriebenen Qualitätsframeworks, die sich gemäß eines »analytischen Qualitätssicherungsansatzes« (siehe Beginn des letzten Abschnitts 3.2.1) auf die Zusammensetzung und Bestimmung der Modellqualität konzentrieren, liegt der Fokus bei Modellierungsrichtlinien in erster Linie auf dem Modellierungsprozess. Mit Hilfe von Modellierungsrichtlinien soll die modellierende Person während der Modellierung dabei unterstützt werden, qualitativ hochwertige Modelle zu erstellen. Demnach können die beiden in diesem Abschnitt beschriebenen Ansätze den sogenannten »konstruktiven Qualitätssicherungsansätzen« zugeordnet werden [OBS12]. Eine einflussreiche Arbeit aus dieser Richtung stellen die »Grundsätze ordnungsmäßiger Modellierung« (GoM) von BECKER et al. [BPV12] und BECKER et al. [BRS95] dar. Diese sind auch in die »Guidelines of Business Process Modeling« [BRU00] eingeflossen. Abb. 3.9 zeigt eine Übersicht über die Qualitätsaspekte der GoM. Dabei stellen die drei oberen Aspekte Richtigkeit, Relevanz und Wirtschaftlichkeit notwendige Kriterien dar, die drei unteren Aspekte besitzen dahingegen optionalen Charakter. Die Grundsätze der GoM werden nach [BPV12; BRS95; BRU00] wie folgt definiert und zusammengefasst:

Richtigkeit Ein Modell soll einerseits in Bezug auf das Metamodell der Modellierungssprache konsistent und vollständig sein (entspricht dem syntaktischen Qualitätsaspekt im SEQUAL-Framework). Zusätzlich sollen Struktur und Verhalten eines Modells mit der Wirklichkeit übereinstimmen (entspricht teilweise dem semantischen Qualitätsaspekt im SEQUAL-Framework).

Relevanz Das Prinzip der Relevanz legt drei Bedingungen fest: Die Auswahl eines geeigneten Objektsystems für die Darstellung des Wirklichkeitsausschnitts, die Verwendung einer passenden Modellierungstechnik sowie die Entwicklung

eines minimalen Modells. Ein minimales Modell enthält keine Elemente, deren Entfernung nicht mit Informationsverlust für die lesende Person einhergeht.

Wirtschaftlichkeit Um ein ausgewogenes Kosten-Nutzen-Verhältnis bei der Erstellung eines Modells zu erzielen, sollen die anderen Prinzipien der GoM gegebenenfalls eingeschränkt umgesetzt werden (z. B. eine Verringerung der Richtigkeit oder Klarheit). Um die Kosten durch die Erstellung eines Modells zu reduzieren, kann eine Unterstützung durch den Einsatz von Referenzmodellen oder Werkzeugen mit Modellierungsunterstützung erfolgen, auch die Wiederverwendung bestehender Modelle ist denkbar.

Klarheit Ein Modell soll von der lesenden Person verstanden werden, es ist nicht ausreichend, wenn es von der modellierenden Person als verständlich betrachtet wird (entspricht somit dem pragmatischen Qualitätsaspekt im SEQUAL-Framework). Dieses Prinzip bezieht sich im Wesentlichen auf die Eigenschaften des Modell-Layouts (z. B. Kantenüberschneidungen, Visualisierung).

Vergleichbarkeit Innerhalb eines Modellierungsprojekts sollen bestehende Konventionen (z. B. für die Wahl der Beschriftungen) über mehrere Modelle hinweg konsistent eingesetzt werden.

Systematischer Aufbau Die Beziehungen zwischen Modellen, die verschiedene Sichten darstellen, sollen präzise definiert werden (z. B. über ein Metamodell, welches alle relevanten Sichten integriert).

Ebenso wie beim SEQUAL-Framework, werden in den GoM keine konkreten Angaben zur Messung der jeweiligen Qualitätsaspekte gemacht. Somit bleiben sie – genau wie das SEQUAL-Framework – für die unmittelbare Anwendung ohne Expertenwissen auf dem Gebiet der Modellierung zu abstrakt [MRA10, S. 127].

Aus dieser Motivation heraus werden mit den »Seven Process Modeling Guidelines« (7PMG) von MENDLING et al. [MRA10] sieben konkret anwendbare Richtlinien für die Geschäftsprozessmodellierung vorgeschlagen. Diese sollen dazu führen, dass Modelle möglichst verständlich gestaltet werden und infolgedessen eine möglichst geringe Anzahl syntaktischer Fehler aufweisen. Die sieben Richtlinien G1-G7 lauten wie folgt:

G1: *Benutze so wenige Elemente im Modell wie möglich* - Größere Modelle sind schwerer zu verstehen und gehen mit einer höheren Fehlerwahrscheinlichkeit einher.

3. Verwandte Arbeiten

- G2:** *Minimiere die ein- und ausgehenden Kanten pro Element* - Bei einer hohen Anzahl von ein- und ausgehenden Kanten sinkt die Modellverständlichkeit, ebenso steigt die Fehlerwahrscheinlichkeit.
- G3:** *Verwende ein Start- und ein Endereignis* - Dies schafft größere Klarheit für den Leser und ist ebenso Voraussetzung für die Anwendung des Modells in einigen Workflow-Engines.
- G4:** *Modelliere so strukturiert wie möglich* - Strukturierte Modelle besitzen für jede Verzweigung des Kontrollflusses auch eine entsprechende Zusammenführung vom gleichen Typ. Für strukturierte Modelle wurde nachgewiesen, dass sie leichter zu verstehen sind und eine geringere Fehlerwahrscheinlichkeit besitzen.
- G5:** *Vermeide OR-Join/Splits* - Insbesondere die Ambiguität von OR-Joins führt zu einem erschwerten Verständnis des Modells.
- G6:** *Verwende Beschriftungen im Verb-Objekt-Stil²²* - Der Verb-Objekt-Stil ist eindeutiger und zweckdienlicher als andere Beschriftungsstile.
- G7:** *Zerlege das Modell, wenn es mehr als 50 Elemente enthält²³* - Die Aufteilung auf mehrere Modelle (z. B. durch Einführung einer Hierarchie) führt zu einer verbesserten Übersichtlichkeit (Begründung erfolgt analog zu G1).

Die Herleitung und Begründung der Richtlinien basiert auf empirischen Untersuchungen der Autoren auf dem Gebiet der Forschung zur Fehlerwahrscheinlichkeit von Modellen (u.a. [Me08b; MNA07; MRC07]). Dieses Forschungsgebiet liefert Hinweise dazu, welche Faktoren in Bezug auf die pragmatische Qualität (gemäß SEQUAL-Framework, siehe Abschnitt 3.2.1) eine Rolle spielen. Deren individuelle Auswertung greift allerdings im Rahmen dieser Arbeit zu weit und könnte während der Anschlussforschung aufgegriffen werden. Weiterhin wird beispielsweise von MORENO-MONTES DE OCA und SNOECK [MS15b] eine deutlich umfangreichere Liste von 27 Richtlinien für die Geschäftsprozessmodellierung vorgestellt. Diese wurden auf der Basis einer Übersichtsarbeit [Mo15] zur Modellqualität und zu bestehenden Richtlinien zusammengefasst.

Die Vermittlung von Modellierungsrichtlinien im Lehrkontext kann einen Beitrag dazu leisten, Studierende dazu zu befähigen, qualitativ hochwertige Modelle zu erstellen. Wird die Anwendung von Modellierungsrichtlinien darüber hinaus an Lernziele geknüpft, ist eine anschließende Überprüfung notwendig. Dabei gilt es zu

²² Dabei ist zu beachten, dass sich die 7PMG auf Beschriftungen in englischer Sprache beziehen.

²³ In einer späteren Arbeit wird die Anzahl auf 31 Elemente reduziert [Me12].

bewerten, inwieweit die geforderten Modellierungsrichtlinien von Studierenden bei der Modellierung tatsächlich umgesetzt werden. Vor diesem Hintergrund stellt die geringe Präzision einiger Richtlinien (welche in der Natur genereller Handlungsempfehlungen begründet liegt) eine Herausforderung dar. Diese muss zunächst bewältigt werden, um ein den Richtlinien entsprechendes objektives und zuverlässiges Bewertungsverfahren entwickeln zu können. So werden beispielsweise für den Grundsatz der Klarheit aus den GoM kaum konkrete Beispiele aufgeführt. Auch für die deutlich konkreteren Richtlinien der 7PMG ist z. B. anhand der Richtlinien G1 und G2 unklar, welche Optimalwerte für die Anzahl an Elementen bzw. der ein- und ausgehenden Kanten pro Element im konkreten Anwendungsfall gelten.

3.2.3. Qualitätsmetriken

Die Auseinandersetzung mit der Literatur in den beiden vorangehenden Abschnitten zeigt deutlich, dass die Modellqualität anteilig durch verschiedene Aspekte beeinflusst wird. Da diese Aspekte überwiegend (mit Ausnahme vom 3QM-Framework [OBS12]) lediglich auf einer abstrakten Ebene definiert wurden, sind sie für eine Messung und darauf aufbauende Bewertung nicht ohne weiteres zugänglich. Eine entsprechende Operationalisierung kann durch den Einsatz von Metriken erreicht werden. Der Zusammenhang zwischen Modellqualität und Metriken soll anhand des Vorschlags von MOODY [Mo05a] für die Gestaltung von Qualitätsframeworks für die konzeptuelle Modellierung veranschaulicht werden. Dabei setzt sich die Modellqualität in einem hierarchischen Aufbau (siehe Abb. 3.10)²⁴ auf der obersten Ebene aus Charakteristika (entsprechend Aspekten), untergeordneten Subcharakteristika (Teilaspekten) und auf der untersten Ebene jeweils konkreten Metriken zusammen. Für Metriken muss dabei ein Messverfahren angegeben werden, das eine Quantifizierung anhand bestimmter Modelleigenschaften ermöglicht.

Zur formalen Beschreibung von Qualitätsmetriken für Geschäftsprozessmodelle soll zunächst der Untersuchungsgegenstand, d.h. das Geschäftsprozessmodell formalisiert werden. Dazu eignet sich das Konzept der Geschäftsprozessgraphen, die unabhängig von der verwendeten Sprache die grundlegende Modellstruktur beschreiben.

24 Als Vorbild dient dabei das ISO/IEC 9126 Modell für Softwarequalität.

3. Verwandte Arbeiten

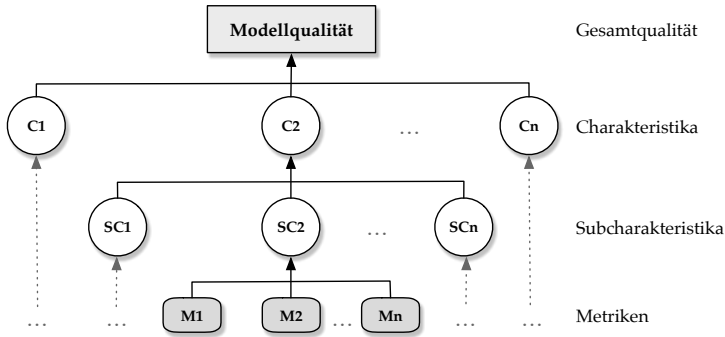


Abb. 3.10.: Vorschlag für den Aufbau eines Qualitätsframeworks zur konzeptuellen Modellierung, in Anlehnung an [Mo05a]

Definition 3.1: Geschäftsprozessgraph

Sei T eine Menge von Typen und Ω eine Menge textueller Beschriftungen. Ein **Geschäftsprozessgraph** (GPG) ist ein Tupel $(N, E, \tau, \lambda, \alpha)$, für welches gilt:

- (i) N ist eine endliche Menge von Knoten;
- (ii) $E : N \times N$ ist eine endliche Menge von Kanten;
- (iii) $\tau : (N \cup E) \rightarrow T$ bildet Knoten und Kanten auf Typen ab;
- (iv) $\lambda : (N \cup E) \rightarrow \Omega$ bildet Knoten und Kanten auf textuelle Beschriftungen ab;
- (v) $\alpha : (N \cup E) \rightarrow (T \rightarrow \Omega)$ bildet Knoten und Kanten auf Attribute ab, wobei ein Attribut eine Kombination aus einem Typ und einer Beschriftung darstellt.
- (vi) $\beta : (N \cup E) \rightarrow (\Omega \rightarrow \Omega)$ bildet Knoten und Kanten auf Schlüssel-Wert-Paare in Form von textuellen Beschriftungen ab.

(Übersetzt und um (vi) ergänzt nach DIJKMAN et al. [Di11, S. 499]).

Sollen auch Aspekte untersucht werden, die die sekundäre Notation (siehe Abschnitt 2.2.3) betreffen, so müssen die relevanten Merkmale (wie z. B. Position, Farbe oder Größe) ebenfalls formalisiert werden. Dies kann anhand der Abbildungsfunktion β realisiert werden, wie das Beispiel in Abb. 3.11 zeigt. Hier wird der Geschäftsprozessgraph zum abgebildeten Petri-Netz N dargestellt und exemplarisch die Eigenschaften der beiden Knoten $n1$ und $n4$ sowie der Kante $e2$ aufgelistet. Für alle Knoten ist die jeweilige Position anhand der xy-Koordinaten als Schlüssel-Wert-Paar vermerkt.

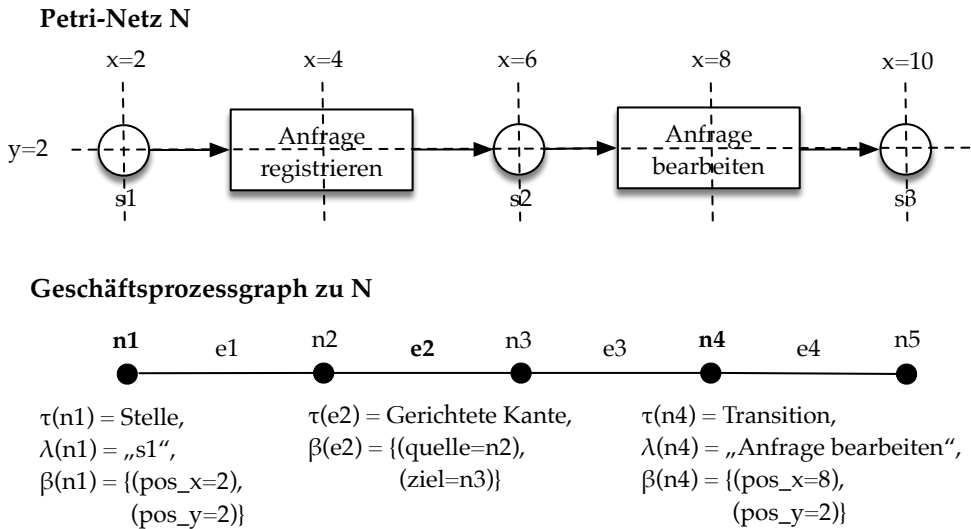


Abb. 3.11.: Geschäftsprozessgraph zu einem gegebenen Petri-Netz N

Mittels dieser Informationen lässt sich z. B. die Flussrichtung der Modellelemente bestimmen. Es ist anzumerken, dass Geschäftsprozessgraphen nicht auf die Formalisierung von Geschäftsprozessmodellen beschränkt sind, sondern grundsätzlich für graphbasierte Modellierungssprachen (z. B. ER- oder UML-Diagramme) Gültigkeit besitzen²⁵. Auf der Basis der in Definition 3.1 beschriebenen Elemente lassen sich daher Metriken für graphbasierte Diagramme unabhängig von der verwendeten Modellierungssprache präzise definieren. Dies verspricht im Kontext der Entwicklung von Bewertungsdiensten im Rahmen von Forschungsfrage FF2 einen bedeutsamen Vorteil, da einige Metriken somit sprachübergreifend eingesetzt werden können.

Das folgende Beispiel stellt eine Metrik aus [Di11] zur Bestimmung der syntaktischen Ähnlichkeit von Beschriftungen vor, die auf der Levenshtein-Distanz²⁶ beruht. Bei dieser handelt es sich zwar nicht per se um eine Metrik mit Bezug zur Modellqualität, sie spielt aber bei der späteren Entwicklung von Bewertungsdiensten eine wichtige Rolle, um ein studentisches Modell mit einer Musterlösung zu vergleichen.

²⁵ Auch der in Kapitel 2 beschriebene Aufbau der Syntax von graphbasierten Modellierungssprachen (siehe Abb. 2.10 auf S. 48) deckt sich mit der hier eingeführten formalen Definition des GPG. Dabei können die sogenannten Spezialtypen aus Ersterer im GPG anhand passender Attribute durch die Abbildungsfunktion α beschrieben werden

²⁶ Bei der nach ihrem Erfinder benannten Levenshtein-Distanz handelt es sich um die minimale Anzahl an benötigten Operationen (Einfügen, Löschen und Ersetzen eines Zeichens) für die Umwandlung einer gegebenen Zeichenkette in eine andere gegebene Zeichenkette [Le66].

3. Verwandte Arbeiten

Metrik 3.1: Syntaktische Ähnlichkeit

Seien $l, l_1, l_2 \in \Omega$ textuelle Beschriftungen, $|l|$ die Länge einer Beschriftung (Anzahl der Zeichen) und $lev(l_1, l_2)$ die Levenshtein-Distanz der Beschriftungen l_1 und l_2 . Die Funktion $max(a, b)$ gibt den größeren der beiden Werte a und b zurück.

Die **syntaktische Ähnlichkeit** zwischen den Beschriftungen l_1 und l_2 ist daraufhin definiert wie folgt:

$$syn(l_1, l_2) = 1 - \frac{lev(l_1, l_2)}{max(|l_1|, |l_2|)}$$

Das Messverfahren für Metrik 3.1 wird demnach anhand zweier Messfunktionen realisiert. Während die Messfunktion $lev(l_1, l_2)$ zur Bestimmung der Levenshtein-Distanz eine absolute Messfunktion darstellt (Anzahl der benötigten Änderungsoperationen), handelt es sich bei der Funktion $syn(l_1, l_2)$ um eine relative Messfunktion. Bei einer relativen Messfunktion wird das Ergebnis einer absoluten Messfunktion in Bezug zu einer anderen Größe gesetzt, so dass die Ergebnisse verschiedener relativer Messfunktionen miteinander verglichen werden können (siehe z. B. [OBS12]). Beispielsweise gilt für die Beschriftungen $\lambda(n2)$ und $\lambda(n4)$ des GPG aus Abb. 3.11 folgendes:

$$lev(\text{Anfrage bearbeiten}, \text{Anfrage registrieren}) = 8$$

$$syn(\text{Anfrage bearbeiten}, \text{Anfrage registrieren}) = 0,6$$

Für zwei andere Beschriftungen *Bearbeiten* und *Registrieren* gilt:

$$lev(\text{Bearbeiten}, \text{Registrieren}) = 8$$

$$syn(\text{Bearbeiten}, \text{Registrieren}) \approx 0,33$$

Demnach sind die Beschriftungen »Anfrage bearbeiten« und »Anfrage registrieren« ($syn = 0,6$) einander ähnlicher als die Beschriftungen »Bearbeiten« und »Registrieren« ($syn \approx 0,33$), obwohl für beide Paare eine Levenshtein-Distanz von 8 ermittelt wurde. Dies macht deutlich, dass nur die Betrachtung der Ergebnisse der relativen Messfunktion über die Funktion $syn(l_1, l_2)$ einen sinnvollen Vergleich ermöglicht.

Im Kontext der Beschreibung von Qualitätsmetriken für Geschäftsprozessmodelle definiert PFLANZL [Pfl17] eine absolute Messfunktion als Abbildungsfunktion $a : G \rightarrow \mathbb{Q}$, die einen Geschäftsprozessgraphen G auf eine rationale Zahl abbildet.

Auf das Ergebnis einer absoluten Messfunktion a kann eine Überföhrungsfunktion s angewendet werden, so dass gilt: $s \circ a : G \rightarrow [0, 1]^{27}$. Somit entspricht $s \circ a$ der Erzeugung einer relativen Messfunktion r , deren Ergebnis immer im Intervall $[0, 1]$ liegt. In der Regel stellt 0 bei Qualitatsmetriken den schlechtesten und 1 den besten moglichen Wert dar. Vor der Anwendung der Überföhrungsfunktion s mussen neben a zusatzlich zwei weitere Parameter bekannt sein: das Optimierungsziel sowie die jeweilige Bezugsgroe. Das Optimierungsziel gibt an, ob das Ergebnis der absoluten Messfunktion maximiert oder minimiert werden muss, um das Ergebnis der relativen Messfunktion (die fur eine Qualitatsmetrik stehen kann) zu verbessern. Zahlt die absolute Messfunktion z. B. Fehler, so gilt es die Anzahl dieser zu minimieren, d.h. das Optimierungsziel ist die Minimierung. Überpruft die absolute Messfunktion jedoch z. B. die Erföllung bestimmter Kriterien, so ist das Ziel der Optimierung die Maximierung. Sei die individuelle Bezugsgroe fur einen gegebenen Geschäftsprozessgraphen G durch die Funktion b bestimmt, dann wird die Überföhrungsfunktion s im Hinblick auf das Optimierungsziel wie folgt definiert:

$$s = \begin{cases} 1 - \frac{a}{b}, & \text{falls das Optimierungsziel Minimierung vorliegt,} \\ \frac{a}{b}, & \text{falls das Optimierungsziel Maximierung vorliegt.} \end{cases}$$

Betrachtet man dies am Beispiel der Metrik 3.1, so bestimmt die absolute Messfunktion $a = lev$ die Anzahl an nderungsoperationen. Um eine moglichst geringe Anzahl an nderungsoperationen positiv zu werten, wird die Überföhrungsfunktion s entsprechend dem Optimierungsziel Minimierung angewendet. Dabei wird die Bezugsgroe anhand der Funktion $b = max$ ermittelt. Somit liegt das Ergebnis im Intervall $[0, 1]$, wobei der Wert 1 der grotmoglichen hnlichkeit bzw. der vollstandigen bereinstimmung zweier Beschriftungen entspricht²⁸.

Metriken konnen in Bezug auf ihre Anwendbarkeit unterschiedlich generell bzw. speziell ausgepragt sein (siehe z. B. [BRU00; Pf17]). Generelle Metriken wie z. B. Groe oder Umfang lassen sich so definieren, dass sie auf verschiedene Modelltypen (z. B. Datenmodelle, UML-Diagramme oder Geschäftsprozessmodelle) anwendbar sind. Andere Metriken werden speziell fur die Anwendung auf Geschäftsprozessmodellen definiert, beispielsweise die Anzahl moglicher unterschiedlicher Ausföhrungspfade (*»distinct paths«* [Ni98]) in einem Geschäftsprozessmodell. Besonders spezifische

²⁷ Fur die Verknufpfung dieser beiden Funktionen gilt $(s \circ a)(G) = s(a(G))$ [Pf17, S. 63]

²⁸ Die hier beschriebenen Funktionen s , a und b wurden zwar in Bezug auf die Betrachtung eines Geschäftsprozessgraphen G definiert, lassen sich aber auch auf (Teil-)Eigenschaften von G anwenden. Im hier aufgeföhrten Beispiel der Metrik 3.1 sind dies Beschriftungen.

3. Verwandte Arbeiten

Metriken betreffen Merkmale von Modellen einer konkreten Modellierungssprache, hierzu könnte z. B. die Systemeigenschaft Lebendigkeit für Petri-Netze (siehe Abschnitt 2.2.4) gezählt werden. Das letzte Beispiel zeigt darüber hinaus, dass hinter Metriken für Geschäftsprozessmodelle nicht zwangsläufig die Absicht steckt, diese Modelle an sich zu bewerten. Vielmehr dienen Geschäftsprozessmodelle als Stellvertreter für einen zu betrachtenden Geschäftsprozess, dessen Eigenschaften über die Analyse eines entsprechenden Geschäftsprozessmodells erfasst werden sollen. Ebenso wenig werden alle Metriken für Geschäftsprozessmodelle mit einem direkten oder indirekten Bezug zur Qualitätsbestimmung festgelegt.

Es ist nicht Ziel dieses Abschnitts, die Vielzahl an vorgeschlagenen Metriken im Zusammenhang mit der Qualitätsbestimmung aus der Literatur der Geschäftsprozessmodellierung umfassend zu beschreiben. Stattdessen wird an dieser Stelle ein grober Überblick über die betrachteten Merkmale ausgewählter Metriken gegeben. Überraschenderweise decken sich diese Merkmale kaum mit den zuvor beschriebenen Qualitätsaspekten für Geschäftsprozessmodelle bzw. konzeptuelle Modelle (siehe Abschnitte 3.2.1 und 3.2.2). Stattdessen beruhen etliche Metriken vielmehr auf Anpassungen von Softwaremetriken zum Betrachtungsgegenstand Quelltext.

Basierend auf der Annahme, dass die Komplexität einen hohen negativen und somit unerwünschten Einfluss auf die Korrektheit, Wartbarkeit und Verständlichkeit von Geschäftsprozessmodellen besitzt [Ca06], setzen sich viele Arbeiten mit diesem Merkmal im Zusammenhang von Geschäftsprozessmodellen auseinander. Neben simplen Metriken wie der Größe eines Modells (z. B. »number of activities« [Ca06; GL06b]) oder dem »Coefficient of Network Complexity« (CNC) als Verhältnis von Kanten zu Knoten im Modell [Ca06; La01]) wurden auch komplexere Metriken vorgeschlagen. CARDOSO et al. [Ca06] übertragen unter anderem die Halstead-Metrik [Ha87] und die zyklomatische Metrik von MCCABE [Mc76] zur Bestimmung von Softwarekomplexität auf Geschäftsprozessmodelle. Letztere nimmt Bezug auf die beschränkten kognitiven Verarbeitungsfähigkeiten des Menschen. Die darauf beruhende Kontrollflusskomplexität von CARDOSO et al. [Ca06] basiert auf der Idee, die Anzahl erreichbarer Zustände zu evaluieren, die sich in einem Prozessmodell durch die Verzweigung anhand XOR-, OR und AND-Splits ergeben und durch den Entwickler oder Anwender mental zwischengespeichert werden müssen. Auch GRUHN und LAUE [GL06a] geben eine Metrik zur Messung der kognitiven Komplexität an, die betreffende Merkmale in Prozessmodellen betrachtet und mit unterschiedlicher Gewichtung in die Gesamtbewertung einfließen lässt. Ebenso entwickeln VANDERFEESTEN et al. [Va08] mit der Cross-Connectivity-Metrik einen Ansatz zum

3.3. Automatisierte Bewertung graphbasierter Diagramme

Bestimmen des kognitiven Aufwands für das Verstehen der Beziehungen zwischen allen Paaren von Elementen in einem Prozessmodell.

Eine umfangreiche Beschreibung von Qualitätsmetriken für Geschäftsprozessmodelle in Form von EPK-Modellen ist in [Me08a] zu finden. Diese Metriken sollen dabei unterstützen, die Wahrscheinlichkeit des Auftretens von Fehlern in Geschäftsprozessmodellen vorherzusagen. Viele dieser Metriken wurden aus der Literatur zusammengetragen, es wurden aber auch eigene Metriken entwickelt, wie beispielsweise das »Connector Mismatch« als Anzahl nicht zueinanderpassender Konnektoren²⁹ [Me08a, S. 125] oder die Konnektorenheterogenität, die sich auf die Verwendung verschiedener Konnektorentypen bezieht [Me08a, S. 126f].

Unter der Annahme, dass die Komplexität und Fehlerwahrscheinlichkeit Einfluss auf die Lesbarkeit und Verständlichkeit von Modellen besitzen, lassen sich die in diesem Abschnitt vorgestellten Metriken am ehesten dem pragmatischen Qualitätsaspekt zuordnen. Derartige Einflussfaktoren werden in Arbeiten zur Verständlichkeit von Geschäftsprozessmodellen untersucht (z. B. [DTD18; HFL12]). Im Rahmen des Forschungsvorhabens gilt es zu prüfen, welche dieser Metriken dafür geeignet sind, bei der Bewertung studentischer Modelle eingesetzt zu werden.

3.3. Automatisierte Bewertung graphbasierter Diagramme

Mit dem vorliegenden Abschnitt wird nun der Fokus auf Arbeiten gelegt, die im Hinblick auf die Realisierung einer E-Assessment-Plattform und Bewertungsdienste zur automatisierten Bewertung von graphbasierten Diagrammen relevant sind. Diese Arbeiten beschreiben einerseits generelle Ansätze, die sich z. B. der automatisierten Analyse von textuellen Beschriftungen oder dem Prozessmodell-Matching widmen, ohne explizit auf den Anwendungskontext dieser Arbeit abzielen (siehe Abschnitt 3.3.1). Andererseits existieren auch bereits Ansätze, die sich mit der automatisierten Bewertung von graphbasierten Diagrammen unter explizitem Bezug zum Kontext Hochschullehre befassen. Diese wurden im Rahmen einer systematischen Literaturrecherche erfasst (siehe Abschnitt 3.3.2).

²⁹ Diese Metrik bezieht sich auf die Richtlinie G4 aus den 7PMG [MRA10] (siehe Abschnitt 3.2.2).

3.3.1. Allgemeine Ansätze

Zahlreiche Ansätze befassen sich in unterschiedlichen Forschungsrichtungen anhand diverser Fragestellungen mit der automatisierten Analyse von Modellen in Form graphbasierter Diagramme. Auch wenn diese Ansätze sich nicht explizit auf die Bewertung graphbasierter Diagramme im Anwendungskontext Lehre beziehen, bergen sie das Potenzial, auch in diesem Kontext gewinnbringend eingesetzt zu werden. In diesem Abschnitt soll keine vollständige Erfassung erfolgen, vielmehr sollen prominente Strömungen anhand ausgewählter Publikationen charakterisiert werden, die auf dem Gebiet der Geschäftsprozessmodellierung öfter³⁰ auf einschlägigen Konferenzen oder Workshops (z. B. Enterprise Modelling and Information Systems Architectures - EMISA, Business Process Management - BPM oder Conference on Advanced Information Systems Engineering - CAiSE) sowie in Form von Zeitschriftenbeiträgen (z. B. Information Systems, Decision Support Systems) thematisch vertreten werden.

Analyse von Modellelementbeschriftungen

Im Zusammenhang mit der Analyse von Modellelementbeschriftungen werden häufig pragmatische Qualitätsaspekte behandelt. Beispielsweise werden mehrdeutige Bezeichnungen oder Abweichungen von sprachlichen Konventionen in Modellen automatisiert identifiziert und teilweise auch Verbesserungsvorschläge entwickelt, um somit die Verständlichkeit von Modellen zu erhöhen. Darüber hinaus ist die Analyse von Modellelementbeschriftungen gleichzeitig auch eine fundamentale Voraussetzung für viele Verfahren, bei denen eine Identifikation von Modellelementen durchgeführt werden soll. Somit ist eine solche Analyse auch in den Ansätzen, die in den folgenden Absätzen (z.B. zur ähnlichkeitsbasierten Suche und zum Prozessmodell-Matching) beschrieben werden, sowie bei zahlreichen automatisierten Bewertungsverfahren im Kontext Hochschullehre (Abschnitt 3.3.2) enthalten. Da Modellelemente in Geschäftsprozessmodellen üblicherweise in knapper Form durch Begriffe in natürlicher Sprache beschriftet werden (siehe auch Abschnitt 2.2.3), kommen hierbei zumeist Werkzeuge und Verfahren aus dem Gebiet des Natural Language Processing (NLP)³¹ zum Einsatz. In Tab. 3.6 wird ein Überblick über

³⁰ D.h. nicht nur vereinzelt, sondern es finden sich jeweils mehrere Publikationen von unterschiedlichen Autoren dazu.

³¹ An anderer Stelle wird auch der Begriff »Computational linguistics« verwendet, insbesondere im deutschen Sprachraum ist auch der Begriff Computerlinguistik als Bezeichnung für das Gebiet geläufig.

3.3. Automatisierte Bewertung graphbasierter Diagramme

Tab. 3.6.: Ausgewählte Quellen, die sich mit der Analyse von Modellelementbeschriftungen in Prozessmodellen befassen, I: Identifikation von Beschriftungsstilen bzw. Schwachstellen, V: Zusätzliche Verbesserungsvorschläge.

Quelle	Ansatz	NLP-Werkzeuge	I	V
FRIEDRICH [Fr09]	Erkennung von Ambiguitäten, die durch die Verwendung von Synonymen und Hyperonymen entstehen	WordNet	✓	-
GRUHN und LAUE [GL09]	Erkennung häufiger Modellierungsfehler in EPK	Synonym-Datenbank	✓	-
LEOPOLD et al. [LSM12]	Überarbeitung von Aktivitätsbeschriftungen, die vom festgelegten Beschriftungsstil abweichen	WordNet, Stanford Parser, Korpus (Englisch)	✓	✓
LEOPOLD et al. [Le13]	Prüfung auf die Einhaltung festgelegter Beschriftungskonventionen (sprachunabhängig)	Korpus der jeweiligen Sprache	✓	-
STORCH et al. [SLG15]	Prüfung auf die Einhaltung individueller Beschriftungskonventionen	WordNet, Stanford Tagger, NER	✓	-
PITKE et al. [PLM15]	Auflösen von Ambiguitäten, die durch die Verwendung von Synonymen und Homonymen entstehen	BabelNet, WSD	✓	✓
LEOPOLD et al. [Le19]	Entwicklung eines speziellen POS-Taggers für Prozessmodellbeschriftungen	HMM	✓	-

verschiedene Publikationen dargestellt, die sich mit der Analyse von Modellelementbeschriftungen befassen. Im Folgenden sollen diese Publikationen zusammen mit verwendeten NLP-Werkzeugen, linguistischen Fachbegriffen und typischen Herausforderungen im jeweiligen Kontext kurz vorgestellt werden.

FRIEDRICH [Fr09] verwendet die WordNet-Datenbank³² für die Erkennung von Synonymen (Wörter mit gleicher bzw. ähnlicher Bedeutung) und Hyperonymen (Wörter mit übergeordneter Beziehung zu anderen Wörtern, z. B. Fahrzeug als Oberbegriff für Automobil, Fahrrad u.a.) in Modellelementbeschriftungen im SAP R/3-Referenzmodell. Das Ziel dabei ist es, die Verwendung von Begriffen mit gleicher Bedeutung im Sinne mangelnder pragmatischer Qualität aufzudecken. WordNet ist eine an der Princeton University entwickelte Datenbank für die englische Sprache, welche lexikalische und semantische Beziehungen zwischen Wörtern in einem Netzwerk abbildet und somit auch Synonyme und Hyperonyme erfasst.

GRUHN und LAUE [GL09; GL11] entwickeln einen Ansatz zur Überprüfung häufiger Modellierungsfehler in EPK durch die Analyse von Modellelementbeschriftungen. Der Ansatz basiert auf einer früheren Arbeit [GL06c], bei der die Inhalte des Pro-

32 <https://wordnet.princeton.edu/>

3. Verwandte Arbeiten

zessmodells zunächst in PROLOG-Fakten überführt werden und auf die Einhaltung der syntaktischen Anforderungen an EPKs geprüft wird. In [GL09] wird dieser Ansatz erweitert, indem die Beschriftungen zusätzlich in eine Normalform umgewandelt werden. In einem ersten Schritt werden festgelegte Wörter entfernt, die vergleichsweise wenig Informationsgehalt besitzen. Dabei handelt es sich um sogenannte Stoppwörter wie z.B. Artikel oder Hilfsverben. Anschließend werden Wörter mit gleicher Bedeutung in allen Beschriftungen durch eine gemeinsame Normalform ersetzt. Dafür wird eine selbst erstellte Synonym-Datenbank verwendet. Außerdem werden auch Antonyme (verschiedene Wörter mit gegensätzlicher Bedeutung, z. B. Empfang vs. Versand) auf diese Weise in eine gemeinsame Normalform überführt. Auch das Schlüsselwort »nicht« wird für die Erkennung von Antonymen herangezogen. Bei einem der beiden Wörter wird dabei zusätzlich eine Markierung gesetzt, die kennzeichnet, dass das Gegenteil gemeint ist. Durch diese Vorverarbeitungsschritte wird beispielsweise aus den beiden Beschriftungen `Der Antrag ist genehmigt` und `Dem Antrag wurde zugestimmt` die Zeichenkette `Antrag nf_genehmigt`. Auch die Beschriftung `Dem Antrag wurde nicht zugestimmt` wird in diese Normalform überführt, allerdings mit Markierung für das Gegenteil. Anschließend kann durch Abfragen der Faktenbasis in PROLOG auf typische Fehlermuster geprüft werden, wie z. B. ob zwei Ereignisse den gleichen oder einen widersprüchlichen Sachverhalt beschreiben, obwohl durch die umgebenden Kontrollflusskonnektoren etwas anderes ausgedrückt wird. In [GL11] wird dieser Ansatz für die englische Sprache adaptiert.

Auch LEOPOLD et al. [LSM12] verwenden WordNet und zudem den Stanford Parser³³, mit welchem die grammatikalische Struktur von Sätzen abgeleitet werden kann. Zum sogenannten Parsing auf dem NLP-Gebiet gehört z. B. die Gruppierung von Wörtern in syntaktische Einheiten (Phrasen) anhand einer Grammatik oder die Identifikation von Subjekt und Objekt eines Verbs. Besonders in der englischen Sprache ist die sogenannte Konversion oder Nullableitung (engl. *conversion* bzw. *zero derivation*) eine Herausforderung. Hier geht es darum, dass ein Wort unterschiedlichen Wortarten (engl. *part of speech*, POS) zugehörig sein kann, z.B. wie beim englischen Wort »order«, welches als Verb im Infinitiv (to order) oder als Substantiv (the order) auftreten kann. Zur Lösung wird in [LSM12] ein Korpus mit Annotationen der Wortarten in Sätzen der englischen Sprache³⁴ herangezogen, um die Häufigkeit des Auftretens eines Wortes in einer bestimmten Wortart auszulesen und darauf basierend die Wahrscheinlichkeit dafür abzuleiten, dass ein Wort

³³ <https://nlp.stanford.edu/software/lex-parser.shtml>

³⁴ Corpus of Contemporary American English - <https://www.english-corpora.org/coca/>

3.3. Automatisierte Bewertung graphbasierter Diagramme

einer bestimmten Wortart entspricht. Für die Umformung von Aktivitätsbeschriftungen im sogenannten »action-noun«-Stil (z. B. *invoice verification*) zum vorgesehenen »verb-object«-Stil (z. B. *verify invoice*), wird erneut WordNet herangezogen, um anhand der dort enthaltenen Substantivierungs-Beziehungen das zu einem Substantiv (*verification*) passende Verb (*verify*) zu identifizieren.

Da für andere Sprachen keine vergleichbar umfangreichen und frei verfügbaren lexikalisch-semantische Datenbanken wie WordNet existieren, entwickeln LEOPOLD et al. [Le13] einen weiteren Ansatz zur Erkennung von Abweichungen vom festgelegten Beschriftungsstil, der nicht von WordNet abhängig ist. Somit kann dieser Ansatz vergleichsweise einfach auf Sprachen übertragen werden, für die linguistische Korpora existieren. Die in diesen Korpora verfügbaren Annotationen werden verwendet, um die Wortarten der Wörter in Modellbeschriftungen und anschließend anhand linguistischer Muster den verwendeten Beschriftungsstil zu identifizieren. Die Identifikation der korrekten Wortarten von Wörtern in Texten natürlicher Sprache wird auf dem Gebiet des NLP auch als Part-of-Speech-Tagging (kurz: POS-Tagging) bezeichnet. Für diese Aufgabe steht z. B. neben dem Stanford Parser³⁵ auch der Stanford Tagger³⁶ zur Verfügung.

Der Stanford Tagger wird im Ansatz von STORCH et al. [SLG15] für das POS-Tagging verwendet. Deren Ansatz unterscheidet sich von den bislang vorgestellten Ansätzen grundsätzlich dadurch, dass die zu prüfenden Beschriftungsstile nicht festgelegt sind, sondern vom Anwender individuelle Muster definiert werden können. Dies wird als sinnvoll angesehen, da trotz existierender Richtlinien (siehe Abschnitt 3.2.2) über Unternehmen und Einsatzgebiete hinweg nicht überall die gleichen Konventionen für die Beschriftung gelten müssen. Zusätzlich wird ein Glossar aus dem Unternehmen, dessen Prozessmodelle überprüft werden, eingesetzt, um eine sogenannte Named-Entity Recognition (NER, deutsch: Eigennamenerkennung) zu ermöglichen. Dies schließt auch die Erkennung des Typs (z. B. Personennamen, Firmenname, Ort) sogenannter »named entities« mit ein.

Ebenso wie FRIEDRICH [Fr09] beschäftigen sich PITKE et al. [PLM15] mit dem Thema Ambiguität in Prozessmodellbeschriftungen, neben Synonymen werden in ihrem Ansatz zudem auch Homonyme (einzelne Wörter mit mehr als einer Bedeutung) untersucht und aufgelöst. Hier kommt BabelNet³⁷ zum Einsatz, das mehrere Sprachen unterstützt und dabei viele externe Quellen integriert, unter anderem

³⁵ Das Parsing im NLP beinhaltet in den meisten Fällen ein POS-Tagging als Vorverarbeitungsschritt, auch wenn es beim Parsing vorrangig um die Analyse der grundlegenden Satzstruktur geht.

³⁶ <https://nlp.stanford.edu/software/tagger.shtml>

³⁷ <https://babelnet.org/>

3. Verwandte Arbeiten

z. B. WordNet und Wikipedia. Einen besonderen Augenmerk legen PITTKE et al. [PLM15] auf die Entwicklung einer passenden Strategie für die Disambiguierung (engl. *word sense disambiguation*, WSD). Beim Auftreten von Homonymen spielt die Erkennung der jeweiligen Bedeutung eine große Rolle, beispielsweise kann mit dem englischen Begriff »application« z. B. eine Softwareanwendung oder eine Bewerbung gemeint sein³⁸. Für die Erkennung der beabsichtigten Bedeutung muss der Kontext des Wortes herangezogen werden, was bei Prozessmodellen eine besonders große Herausforderung bedeutet, da Modellelementbeschriftungen typischerweise nur wenige Worte aufweisen³⁹. Daher schlagen PITTKE et al. [PLM15] die Verwendung eines speziellen WSD-Verfahrens für BabelNet von NAVIGLI und PONZETTO [NP12] vor, für welches sie positive Evaluationsergebnisse beschreiben.

In einer jüngeren Veröffentlichung beschreiben LEOPOLD et al. [Le19] die Entwicklung eines POS-Taggers für Aktivitätsbeschriftungen in Prozessmodellen durch die Verwendung eines Hidden Markov Modells (HMM), einem stochastischen Modell, das in vielen Gebieten beispielsweise zur Mustererkennung eingesetzt wird. Trotz vielversprechender Evaluationsergebnisse besitzt dieser Ansatz den Nachteil, dass umfangreiche Trainingsdaten (in diesem Fall Prozessmodelle, die manuell annotiert wurden) benötigt werden.

Ähnlichkeitsbasierte Suche und Prozessmodell-Matching

Unternehmen und Organisationen, die umfangreiche Bibliotheken mit Prozessmodellen besitzen, stehen vor der Herausforderung, diese große Menge an Modellen auch gewinnbringend handhaben zu können. Nach SCHOKNECHT et al. [Sc22] wird bei der ähnlichkeitsbasierten Suche ein Prozessmodell als Anfrage verwendet, um in einer Bibliothek zu diesem ähnliche Modelle zu identifizieren. Viele Ansätze zur ähnlichkeitsbasierten Suche beruhen auf dem sogenannten Prozessmodell-Matching, bei dem die Ähnlichkeit von Aktivitäten (oder anderen Modellelementen) zwischen verschiedenen Prozessmodellen berechnet wird, um daraufhin mögliche Zuordnungen (engl. *matches*, auch: *alignment* oder *mapping*) identifizieren zu können. Andere Ansätze verwenden bestimmte Metriken oder Analyseverfahren aus dem Information Retrieval. Somit kann in großen Prozessmodellbibliotheken beispielsweise die

³⁸ Auch beim Wort »order« aus dem zuvor angeführten Beispiel für die Nullableitung handelt es sich um ein Homonym (mögliche Bedeutungen sind z. B. Bestellung, Befehl, Reihenfolge, ...).

³⁹ Dies wird auch in Richtlinien angeführt, siehe beispielsweise die in Abschnitt 3.2.2 beschriebenen 7PMG [MRA10], die den »verb-object«-Stil (z. B. *create receipt*) für Aktivitätsbeschriftungen empfehlen.

3.3. Automatisierte Bewertung graphbasierter Diagramme

Konformität von Prozessmodellen überprüft werden, ebenso können Duplikate oder Varianten von Prozessmodellen erkannt und Referenzmodelle abgeleitet werden. Die Suche nach bestimmten Prozessmodellen ermöglicht gleichzeitig auch die Wiederverwendung bereits bestehender Prozessmodelle (oder Teilen daraus) [Sc17].

Einer der ersten Ansätze hierzu wurde von EHRIG et al. [EKO07] entwickelt. Sie definieren eine kombinierte Ähnlichkeitsmetrik auf Grundlage syntaktischer, linguistischer und struktureller Ähnlichkeit, die für OWL-DS basierte Petri-Netze angewendet werden kann. Zur Berechnung der syntaktischen Ähnlichkeit wird eine Metrik von MAEDCHE und STAAB [MS02] verwendet, die eine Modifikation der Levenshtein-Distanz darstellt (siehe auch Metrik 3.1 in Abschnitt 3.2). Über die linguistische Ähnlichkeit werden unter Einsatz von WordNet auch Synonyme und Homonyme berücksichtigt. Zudem wird über die strukturelle Ähnlichkeit auch der Kontext der jeweils betrachteten Modellelemente mit einbezogen.

Van DONGEN et al. [DDM08] vergleichen zwei Prozessmodelle miteinander, indem sie zunächst sogenannte »causal footprints« ableiten und diese dann mit Hilfe des sogenannten Vektorraum-Retrievals abgleichen. Für den Abgleich von Beschriftungen wird eine linguistische Vorverarbeitung durchgeführt. Dazu zählt neben der Entfernung von Sonderzeichen, der Umwandlung in Kleinbuchstaben auch die Entfernung von Stoppwörtern. Anschließend erfolgt mittels Stemming eine Reduktion von Wörtern auf eine Stammform durch das Entfernen bestimmter Buchstaben am Wortende. Dabei werden z. B. Pluralformen von Substantiven in die jeweilige Singularform (assessments→assessment) oder konjugierte Verben in den Infinitiv (gives→give) überführt. Auf dem Ergebnis der Vorverarbeitung erfolgt die Identifikation von Synonymen. Bei der Bestimmung der Ähnlichkeit wird ebenfalls der Kontext anhand der Ähnlichkeiten der umgebenden Modellelemente berücksichtigt.

Das von WEIDLICH et al. [WDM10] entwickelte ICOP-Framework bietet die Möglichkeit, aus einer Auswahl verschiedener Komponenten individuelle Matching-Werkzeuge zusammenzustellen um somit die Entwicklung passender Ansätze und den Vergleich dieser Ansätze zu vereinfachen. Zudem erlaubt das ICOP-Framework erstmals auch die Identifikation komplexer Matches, wobei Modellelemente einander nicht 1:1 zugeordnet werden müssen, sondern auch 1:N Zuordnungen identifiziert werden können, wie sie z.B. bei einer abstrakteren (wenige zusammengefasste Aktivitäten) vs. einer detaillierteren Modellierung (mehrere atomare Aktivitäten) auftreten können.

Der Triple-S-Ansatz von CAYOGLU et al. [Ca13c; Ca14] baut auf der Idee von EHRIG

3. Verwandte Arbeiten

et al. [EKO07] auf und kombiniert syntaktische, semantische⁴⁰ und strukturelle Ähnlichkeiten, um eine Gesamtähnlichkeit von Modellen zu bestimmen. Hier basiert die syntaktische Ähnlichkeit direkt auf der Levenshtein-Distanz. Diese wird allerdings nicht für die (aus mehreren Wörtern bestehenden) vollständigen Beschriftungen zweier Modellelemente berechnet, sondern für jede Kombination von einzelnen Wörtern dieser Beschriftungen. Dadurch führt eine ledigliche Veränderung der Wortreihenfolge in zwei Beschriftungen zu einer guten Übereinstimmung, wohingegen eine Berechnung der Levenshtein-Distanz direkt auf den vollständigen Wortbeschriftungen unerwünschterweise zu sehr niedrigen Ähnlichkeitswerten führen würde. Auch die Bestimmung der semantischen Qualität basiert auf der Betrachtung aller Kombinationen einzelner Wörter in den Beschriftungen, hierfür kommt die Ähnlichkeitsmetrik von WU und PALMER [WP94] zum Einsatz, welche die semantische Distanz zwischen zwei Wörtern in einer hierarchischen Baumstruktur auf Basis einer lexikalisch-semantischen Datenbank wie WordNet ermittelt. Durch das Verhältnis von ein- und ausgehenden Kanten sowie der relativen Position eines Modellelements im Gesamtmodell wird die strukturelle Ähnlichkeit bestimmt.

Im Rahmen der BPM-Konferenz 2013 und der EMISA 2015 wurden zwei Wettbewerbe zum Prozessmodell-Matching [An15; Ca14] veranstaltet, bei denen Forscher ihre Ansätze auf vorgegebenen Datensätzen demonstrieren und die Ergebnisse konkret gegenüberstellen konnten⁴¹. Auf Grundlage der Ergebnisse des Triple-S Ansatzes beim Wettbewerb im Jahr 2013 wurde dieser überarbeitet und in modifizierter Version erneut beim Wettbewerb im Jahr 2015 eingereicht, wobei auch bessere Resultate erzeugt wurden. Insgesamt ist die Qualität der Ergebnisse über alle teilnehmenden Ansätze hinweg jedoch noch nicht befriedigend [An15], so dass mit den Wettbewerben weiterer Forschungsbedarf konstatiert werden konnte.

Der Ansatz von SCHOKNECHT und OBERWEIS [SO17] verzichtet bei der Ähnlichkeitsbestimmung gänzlich auf den Einsatz von Prozessmodell-Matching-Techniken und verwendet stattdessen ein Verfahren aus dem Information Retrieval, die sogenannte Latent Semantic Analysis (LSA). Hier werden Prozessmodelle als Textdokumente aufgefasst und durch Vektoren repräsentiert. Bei einem Vergleich mit fünf anderen Ansätzen, die auf Prozessmodell-Matching-Techniken basieren, schneidet dieser Ansatz signifikant besser ab, besonders bei schwer zu bestimmenden Zuordnungen [Sc22].

⁴⁰ In [EKO07] wird die semantische Ähnlichkeit als linguistische Ähnlichkeit bezeichnet.

⁴¹ Bei der Entwicklung des Triple-S Ansatzes [Ca13c] und dem Wettbewerb zum Prozessmodell-Matching im Jahr 2013 [Ca14] hat die Verfasserin dieser Dissertation außerhalb des Kontexts des hier beschriebenen Forschungsvorhabens als Ko-Autorin mitgewirkt.

Vergleich von Modellen mit einer textuellen Beschreibung

In Unternehmen werden neben Prozessmodellen häufig auch dazugehörige textuelle Beschreibungen von Prozessen erstellt, um Stakeholder einbeziehen zu können, die keine Kenntnisse der verwendeten Modellierungssprache besitzen. Bei Änderungen am Prozessmodell oder an der textuellen Beschreibung besteht die Gefahr, dass diese Änderungen nicht an beiden Stellen durchgeführt werden und infolgedessen Prozessmodelle und deren textuelle Beschreibungen nicht mehr übereinstimmen. Ansätze zum Vergleich von Modellen und textuellen Beschreibungen haben daher zum Ziel, bestehende Inkonsistenzen aufzudecken.

Van der AA et al. [ALR15; ALR17] verwenden hierfür den Stanford Parser zur Analyse der textuellen Beschreibung. Dabei werden in einem ersten Schritt sogenannte Anaphern wie z. B. »it« oder »this« aufgelöst, die Verweise auf zuvor im Text eingeführte Entitäten darstellen. Anschließend werden aus den Sätzen relevante Informationen über den Prozess extrahiert. Dazu wird bei der Identifikation von Aktionen im Satz nur der dafür relevante Satzteil betrachtet und ein weiterer Satzteil, der lediglich z. B. eine Bedingung ausdrückt, ignoriert. Auch diese Informationen können aus dem Parsing-Baum des Stanford Parsers unter Verwendung des sogenannten Dependency Parsers abgelesen werden. Darauf folgt die Bereinigung des Texts durch die Entfernung von Stoppwörtern und die Durchführung einer Lemmatisierung. Bei der Lemmatisierung handelt es sich wie beim Stemming um eine Stammformreduktion, allerdings werden hier nicht nur bestimmte Buchstaben entfernt, sondern auch die Bedeutung der Wörter berücksichtigt, so dass z. B. Überführungen wie bought→buy oder better→good möglich sind. Anschließend wird die Ähnlichkeit zwischen dem Text und den Elementen im Prozessmodell berechnet und auf dieser Basis die bestmögliche Übereinstimmung gesucht. Dadurch lassen sich a) Modellelemente identifizieren, die nicht durch die textuelle Beschreibung abgedeckt werden und b) Widersprüche in der Reihenfolge von Aktivitäten in textueller Beschreibung und dem Prozessmodell aufdecken.

Die Grundidee aus [ALR15; ALR17] wird von SÀNCHEZ-FERRERES et al. [Sà18; SCP17] aufgegriffen, sie wählen aber mit dem Verfahren der ganzzahligen linearen Programmierung (engl. integer linear programming, ILP), einem mathematischen Optimierungsverfahren einen fundamental anderen Ansatz für die Lösung. Zudem kommt hier die FreeLing-Bibliothek⁴² zum Einsatz, die eine Reihe von NLP-Analyseverfahren für verschiedene Sprachen anbietet. Im ersten Schritt der »feature

42 <http://nlp.cs.upc.edu/freeling>

3. Verwandte Arbeiten

extraction« werden prozessbezogene Informationen aus der textuellen Beschreibung extrahiert (wie z. B. Akteur, Aktion, Geschäftsobjekt). Es folgt die Ähnlichkeitsberechnung zwischen Text und Modell. Die Bestimmung des optimalen Alignment von Elementen im Prozessmodell zu Sätzen oder Satzteilen der textuellen Beschreibung wird mit Hilfe von ILP durchgeführt, zur Lösung wird der Gurobi Solver⁴³ verwendet.

Es bleibt anzumerken, dass keiner der in diesem Abschnitt vorgestellten Ansätze vollständig korrekte Ergebnisse erzielt⁴⁴. Auch wenn die meisten Ansätze auf etablierte NLP-Verfahren zurückgreifen, stellt die automatisierte Analyse von natürlichsprachlichen Beschriftungen noch immer eine große Herausforderung dar. Für einen möglichen Einsatz im Kontext Hochschullehre bedeutet dies, dass die Ergebnisse stets entsprechend vorsichtig interpretiert und ggf. manuell überprüft und bei Bedarf korrigiert werden sollten. Dennoch soll anhand dieses Abschnitts belegt werden, dass bereits zahlreiche Ansätze und Forschungsbestrebungen existieren, die zwar nicht auf den Kontext Hochschullehre ausgerichtet sind, aber vielversprechendes Potenzial bieten, hier gewinnbringend aufgegriffen und entsprechend adaptiert zu werden.

Ein konkretes Beispiel dafür bildet der Ansatz von SÀNCHEZ-FERRERES et al. [Sà18; SCP17] für den Vergleich von Modellen mit einer textuellen Beschreibung. Dieser findet in [Sà20]⁴⁵ zur Überprüfung studentischer BPMN-Modelle Anwendung, indem diese Modelle mit dem textuellen Szenario der Aufgabenstellung verglichen werden. Ansätze zur Analyse von Modellelementbeschriftungen könnten im E-Assessment für die grafische Modellierung weiterhin eingesetzt werden, um Defizite im Sinne der pragmatischen Qualität in studentischen Modellen aufzudecken, z. B. Abweichungen von Beschriftungskonventionen für Modellelemente oder die inkonsistente Verwendung von Beschriftungsstilen. Auch die Verwendung von Synonymen, Hyperonymen oder Homonymen, die eine Modellinterpretation erschweren, könnte aufgezeigt werden. Bislang werden pragmatische Aspekte im E-Assessment graph-basierter Diagramme allerdings kaum berücksichtigt (siehe auch den folgenden Abschnitt 3.3.2). Ebenso können Ansätze für die ähnlichkeitsbasierte Suche und das Prozessmodell-Matching zum Einsatz kommen, um studentische Modelle mit Musterlösungen abzugleichen. Tatsächlich verfolgen viele Arbeiten auf dem Gebiet

⁴³ <https://www.gurobi.com/>

⁴⁴ Auf eine Darstellung der jeweiligen Evaluationsergebnisse wurde an dieser Stelle aufgrund geringer Vergleichbarkeit durch unterschiedliche Problemstellungen und Evaluationsansätze bewusst verzichtet, der interessierte Leser sei an dieser Stelle auf die jeweiligen Originalquellen verwiesen.

⁴⁵ Diese Publikation wurde auch über die im folgenden Abschnitt 3.3.2 beschriebene Literaturstudie identifiziert.

des E-Assessment für die grafische Modellierung einen derartigen Ansatz (siehe ebenso Abschnitt 3.3.2), allerdings zumeist ohne auf die hier vorgestellten Vorarbeiten zurückzugreifen. In dieser Arbeit sind im Kontext von Forschungsfrage FF2 etliche Ansätze und NLP-Methoden aus diesem Abschnitt bei der Entwicklung von Bewertungsdiensten berücksichtigt worden (siehe Abschnitt 5.3).

3.3.2. Ansätze aus dem Kontext Hochschullehre

Am engsten mit dem Thema dieser Arbeit verwandt sind bestehende Ansätze, die ebenfalls die automatisierte Bewertung von Modellen in Form graphbasierter Diagramme im Kontext Hochschullehre verfolgen. Um den aktuellen Forschungsstand zu diesen Ansätzen auf systematische Weise zu erheben, wurde unter Beteiligung verschiedener Personen aus dem Projekt KEA-Mod eine umfangreiche Literaturstudie⁴⁶ unter Berücksichtigung etablierter methodischer Vorgehensweisen durchgeführt. Die Vorgehensweise bei der Durchführung sowie die Ergebnisse der Literaturstudie wurden in [Ul23a] veröffentlicht⁴⁷ und werden in den beiden folgenden Abschnitten übersetzt und grob wiedergegeben. Dabei wird ein Fokus auf diejenigen Aspekte gelegt, die im Rahmen dieser Dissertation besonders relevant sind. An passender Stelle werden in diesem Zusammenhang auch individuelle Kommentare eingefügt.

Vorgehensweise bei der Durchführung

Die Durchführung der Literaturstudie orientiert sich primär an den Richtlinien, die vom BROCKE et al. [Br09; Br15] zur Gestaltung von Literaturstudien in der Forschung auf dem Gebiet Information Systems vorschlagen. Zusätzlich wurden weitere Empfehlungen mit Bezug auf eine systematische Vorgehensweise berücksichtigt, die an der entsprechenden Stelle genannt werden. Mit dem Ziel, einen Überblick zum aktuellen Stand der Technik und dem Einsatz der automatisierten Bewertung graphbasierter Diagramme zu erstellen, wurden die folgenden fünf Forschungsfragen SLR-FF1 - SLR-FF5⁴⁸ ausgewählt und betrachtet:

⁴⁶ Mit Literaturstudie ist der englische Begriff »literature review« gemeint. Eine Literaturstudie umfasst demnach sowohl die Suche nach als auch die Auswertung von bereits existierenden wissenschaftlichen Arbeiten in Bezug auf festgelegte Forschungsfragen.

⁴⁷ Die Verfasserin dieser Dissertation nimmt in Übereinstimmung mit den beteiligten Ko-Autoren bei der hier verarbeiteten Veröffentlichung den Rang der Erstautorin ein, da sie die führende Rolle bei der Organisation und Umsetzung sowie bei der Manuskripterstellung übernommen hat.

⁴⁸ Bei den Kürzeln der Forschungsfragen wurde jeweils ein SLR vorangestellt, um innerhalb dieses Dokuments eindeutig zwischen den primären Forschungsfragen der Dissertation FF1 und FF2 (siehe Abschnitt 1.2) und den hier betrachteten fünf Forschungsfragen unterscheiden zu können.

3. Verwandte Arbeiten

SLR-FF1: *Welche Aspekte werden bei der automatisierten Bewertung graphbasierter Diagramme untersucht?*

SLR-FF2: *Welche Verfahren kommen bei der technischen Umsetzung der automatisierten Bewertung graphbasierter Diagramme zum Einsatz?*

SLR-FF3: *Werden didaktische Aspekte wie Kompetenzen oder Lernziele für die Modellierung im Kontext der automatisierten Bewertung graphbasierter Diagramme berücksichtigt?*

SLR-FF4: *Welche Art von Feedback für Lernende wird bei der automatisierten Bewertung graphbasierter Diagramme produziert?*

SLR-FF5: *Zu welchem Zeitpunkt im Lernprozess (formativ oder summativ) und in welchem Kontext (Lehrmethode, Kursform) wird die automatisierte Bewertung graphbasierter Diagramme eingesetzt?*

Vor dem Hintergrund dieser Forschungsfragen wurden in der Planungsphase der Literaturstudie gemäß der Empfehlung von KITCHENHAM und CHARTERS [KC07] entsprechende Relevanzkriterien festgelegt. Um im Rahmen der Literaturstudie bei der Analyse berücksichtigt zu werden, war es erforderlich, dass eine Publikation alle vier folgenden Kriterien K1-K4 erfüllt:

K1: *Die Publikation bezieht sich auf graphbasierte Diagramme.*

K2: *Die Publikation beschreibt einen Ansatz zur (teil-)automatisierten Bewertung dieser Diagrammtypen.*

K3: *Der in der Publikation beschriebene Ansatz ist für die Anwendung im Kontext Hochschullehre vorgesehen.*

K4: *Der in der Publikation betrachtete Diagrammtyp wird in den Disziplinen Informatik, Wirtschaftsinformatik oder Information Systems gelehrt.*

Die Anwendung der Relevanzkriterien führte wie beabsichtigt dazu, im Laufe des Suchprozesses diejenigen Publikationen auszuschließen, die außerhalb des abgesteckten Rahmens der Literaturstudie liegen. Abb. 3.12 zeigt einen Überblick über die Vorgehensweise bei der Literaturstudie in Form eines Flussdiagramms in Anlehnung an die »Preferred Reporting Items for Systematic reviews and Meta-Analyses« (PRISMA) Empfehlungen [Pa21]. Zur initialen Suche nach relevanten Publikationen wurde auf die Scopus-Datenbank⁴⁹ zurückgegriffen. Diese deckt wichtige Verlage der betreffenden Disziplinen (Wirtschafts-)Informatik und Information Systems wie z. B.

⁴⁹ <http://scopus.com>

3.3. Automatisierte Bewertung graphbasierter Diagramme

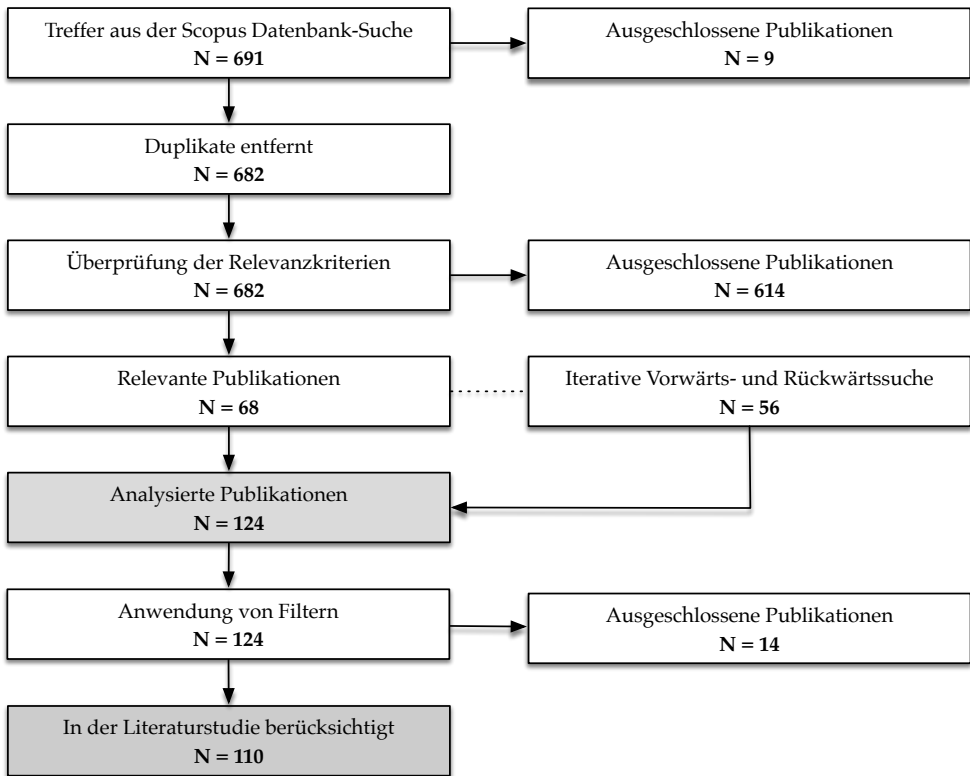


Abb. 3.12.: Überblick über die Vorgehensweise bei der Literatursuche im Rahmen der Literaturstudie, übersetzt nach [UI23a]

Elsevier, Springer oder IEEE ab. Die Gestaltung des Suchstrings für die Datenbank-suche greift die Idee der systematischen Entwicklung von Suchstrings von ZHANG et al. [ZBT11] auf. Dabei wurde zunächst ein sogenannter »Quasi-Gold-Standard« (QGS) [ZBT11] mit 26 zuvor manuell identifizierten Publikationen erstellt, die alle vier Relevanzkriterien erfüllen. In einem iterativen Prozess wurde ein Suchstring aus einer Kombination von bekannten Begriffen und Schlüsselwörtern erstellt, dessen Ergebnismenge möglichst viele Publikationen aus dem QGS enthielt und gleichzeitig von ihrem Umfang her für die anschließende Überprüfung der Relevanzkriterien noch ausreichend handhabbar war. Der finale Suchstring (siehe [UI23a]) deckt 24 von 26 Publikationen aus dem QGS ab und berücksichtigt dabei unterschiedliche Schreibweisen einiger Begriffe (z. B. sowohl »modeling« als auch »modelling«) und beinhaltet zusätzlich spezifische Filter der Scopus-Datenbank, die zur passenden Ausrichtung der Ergebnismenge beitragen sollen (z. B. durch die Auswahl relevan-

3. Verwandte Arbeiten

ter Themengebiete/Disziplinen, Festlegung der Suche im Titel, Abstract und den Schlüsselwörtern). Die Anwendung des Suchstrings im Februar 2021 führte zu einer Ergebnismenge von 691 Publikationen, von denen in einem ersten Schritt 9 als Duplikate ausgeschlossen werden konnten. Im darauffolgenden Schritt wurden alle Publikationen im Hinblick auf die festgelegten Relevanzkriterien überprüft, indem zunächst Titel und Abstract, sowie falls notwendig auch der Volltext untersucht wurden. Dies führte zu einer deutlichen Verringerung der Ergebnismenge auf 68 relevante Publikationen. Auf dieser Ergebnismenge wurden anschließend iterativ eine Rückwärts- sowie Vorwärtssuche durchgeführt, bis keine weiteren neuen Publikationen mehr ermittelt werden konnten. Durch die Rückwärts- und Vorwärtssuche konnten 56 weitere relevante Publikationen ermittelt werden, so dass die Ergebnismenge 124 relevante Publikationen enthielt, darunter alle 26 Publikationen aus dem QGS. Für die Analyse dieser Publikationen wurde ein gemeinsames Rechercheprotokoll definiert, anhand welchem verschiedene Aspekte mit Bezug auf die fünf Forschungsfragen systematisch erfasst werden konnten. Das Rechercheprotokoll deckt 25 solcher Aspekte ab, die in unterschiedlicher Form in einzelnen Spalten kodiert wurden (z. B. »ja/nein«-Fragen, Einteilung in Kategorien, oder Freitext). Von diesen 25 Aspekten beziehen sich 11 Aspekte direkt auf die hier ausgewählten Forschungsfragen. Zur Unterstützung der Kodierung wurde ein Formular zur Datenextraktion vorbereitet, welches in Tab. 3.7 dargestellt ist.

Die 124 zu analysierenden Publikationen wurden in einem ersten Schritt gleichmäßig auf die an der Durchführung der Literaturstudie beteiligten Personen aufgeteilt. Dabei wurde für jede Publikation anhand des Formulars zur Datenextraktion eine Zeile im Rechercheprotokoll durch Einträge in den Spalten zu allen 25 Aspekten ausgefüllt. In einem zweiten Schritt wurde jede Spalte im Rechercheprotokoll durch eine einzelne Person konsolidiert. Auf diese Weise war es möglich, Inkonsistenzen bei der Kodierung der zu betrachtenden Aspekte durch unterschiedliche Personen aufzudecken. Diese Inkonsistenzen wurden in gemeinsamen Diskussionen besprochen und anschließend die Einträge im Rechercheprotokoll angepasst, um eine einheitliche Kodierung zu gewährleisten. Da die meisten Fragen als geschlossene Fragen formuliert wurden, war es zumeist möglich, die im Datenextraktions-Formular festgelegte Kodierung umzusetzen (Top-down-Prinzip). Lediglich bei den Feldern betreffend der untersuchten Aspekte (Tab. 3.7, SLR-FF1) wurde vereinbart, »indirekt« als zusätzlichen Wert neben »ja« und »nein« aufzunehmen, um beispielsweise die Fälle abzudecken, in denen syntaktische Regeln nicht explizit untersucht werden, aber der Vergleich mit einer Musterlösung implizit auch die syntaktische Korrektheit einschließt (siehe auch den folgenden Abschnitt). Für die Felder mit offener

3.3. Automatisierte Bewertung graphbasierter Diagramme

Tab. 3.7.: Formular zur Datenextraktion als Ergänzung zum Rechercheprotokoll

Kurzname	Beschreibung/Frage	Kodierung/Werte
Metadaten & Generelle Aspekte		
Autor(en), Titel und Jahr Quelle	Scopus-Feld oder manuelle Extraktion bei Rückwärts-/Vorwärtssuche Name der Zeitschrift oder Konferenz/Workshop-Reihe	selbsterklärend Name (ohne Iteration, Jahr, usw.)
Dokumententyp	Handelt es sich um einen Konferenzbeitrag oder einen Zeitschriftenartikel?	Eins aus: Konferenzbeitrag, Zeitschriftenartikel
Quasi-Gold-Standard Schritt	Ist die Publikation im Quasi-Gold-Standard enthalten? In welchem Schritt der Literatursuche wurde die Publikation identifiziert?	Eins aus: ja, nein Schritt
Modelltyp(en)	Welche Modelltypen werden behandelt? (Konsolidierung nach Bottom-up-Prinzip)	Liste aus: siehe Tab. 3.9
Universell	Ist der Ansatz als universell beschrieben?	Eins aus: ja, nein
SLR-FF1: Untersuchte Aspekte		
Syntaktische, semantische, pragmatische Aspekte	Untersucht der beschriebene Ansatz syntaktische, semantische und pragmatische Aspekte? (Jeweils einzeln betrachtet)	Eins aus: nicht spezifiziert, nein, ja, indirekt
SLR-FF2: Technische Realisierung		
Verfahren	Welche technischen Verfahren werden in der Publikation beschrieben? (Ableitung von Klassen nach Bottom-up-Prinzip)	Liste aus: Modellvergleich, regelbasiert, maschinelles Lernverfahren, Simulation, Teststrategie, Clusterbildung oder individuell im Einzelfall
Verarbeitung von Beschriftungen	Beschreibt die Publikation den Einsatz spezieller Verarbeitungsschritte von Modellelementbeschriftungen?	Eins aus: ja, nein (kurze Beschreibung falls ja)
Grad der Automatisierung	Wie hoch ist der Automatisierungsgrad des Ansatzes?	Eins aus: vollständig automatisiert, semi-automatisiert (kurze Beschreibung, falls semi-automatisiert)
SLR-FF3: Kompetenzen und Lernziele		
Didaktische Aspekte	Werden didaktische Aspekte wie Kompetenzen oder Lernziele explizit erwähnt?	Eins aus: ja, nein, indirekt (kurze Beschreibung, falls ja oder indirekt)
SLR-FF4: Feedback		
Textuelles, grafisches, numerisches Feedback	Wird textuelles, grafisches und numerisches Feedback erzeugt? (Jeweils einzeln betrachtet)	Eins aus: nicht spezifiziert, nein, ja, indirekt
SLR-FF5: Zeitpunkt und Kontext des Einsatzes		
Zeitpunkt und Kontext	Welcher Zeitpunkt im Lernprozess wird in der Publikation als für den Ansatz passend beschrieben? Was ist der Einsatzkontext?	Liste aus: formativ, summativ, (kurze Beschreibung des Einsatzkontextes), nicht spezifiziert

3. Verwandte Arbeiten

Kodierung (Tab. 3.7, SLR-FF2), wurden die kodierten Werte nach dem Bottom-up-Prinzip durch die Identifikation konsistenter Bezeichner beziehungsweise passender Klassen generiert. Im Anschluss an die Analyse wurde auf Basis der Einträge im Rechercheprotokoll ein Filter angewendet, um Publikationen auszuschließen, die sich lediglich auf konzeptionelle Ansätze beziehen, die noch nicht implementiert wurden. Dadurch wurden weitere 14 Publikationen ausgeschlossen, so dass in der Literaturstudie schlussendlich 110 Publikationen berücksichtigt wurden, um die eingangs aufgestellten Forschungsfragen zu beantworten. Um die Forschungsergebnisse nachvollziehbar zu machen, wurde das Rechercheprotokoll online veröffentlicht⁵⁰.

Zusammenfassung und Diskussion der Ergebnisse

Bei der Betrachtung der Ergebnisse ist zu berücksichtigen, dass keine Aggregation der Publikationen im Hinblick auf die dort beschriebenen Ansätze vorgenommen wurde. Dies kann zu einer Überrepräsentation im Hinblick auf verschiedene Aspekte führen, da einige Ansätze in mehreren Publikationen der Ergebnismenge beschrieben werden.

Die Menge der 110 berücksichtigten Publikationen besteht aus 81 Konferenzbeiträgen und 29 Zeitschriftenartikeln. Tab. 3.8 zeigt eine Übersicht über die Outlets, bei denen zwei oder mehr Publikationen veröffentlicht wurden (getrennt nach Konferenzen und Zeitschriften). Insgesamt sind die 110 Publikationen dabei über 77 Outlets verteilt. Diese Ergebnisse weisen darauf hin, dass die automatisierte Bewertung graphbasierter Diagramme ein Nischenthema ist, welches häufig in Zeitschriften oder bei Konferenzen mit einem Fokus auf Lehre platziert wird. Allerdings wird ebenso beobachtet, dass entsprechende Beiträge gelegentlich bei fachspezifischen Konferenzen im Kontext der Modellierung veröffentlicht werden, z. B. MODELS, Diagrams oder SLATE.

Die Verteilung der berücksichtigten Publikationen über die Jahre hinweg ist in Abb. 3.13 dargestellt. Während besonders viele Publikationen im Zeitraum von 2007 bis 2011 veröffentlicht wurden, sinkt die Anzahl danach, und steigt ab dem Jahr 2020 wieder an. Möglicherweise folgt die Kurve dem allgemeinen Interesse am Thema »E-Learning«, beispielsweise zeigen die Publikationsdaten für dieses Schlüsselwort auf der Web of Science Plattform⁵¹ einen ähnlichen Verlauf. Da die Suche nach

⁵⁰ <http://tinyurl.com/literaturereview-graphbased>

⁵¹ <https://www.webofknowledge.com>

3.3. Automatisierte Bewertung graphbasierter Diagramme

Tab. 3.8.: Outlets, bei denen zwei oder mehr Publikationen aus der Ergebnismenge veröffentlicht wurden [UI23a]

Zeitschrift	N
International Journal of Artificial Intelligence in Education	5
ACM Journal on Educational Resources in Computing	2
IEEE Transactions on Learning Technologies	2
International Journal of Distance Education Technologies	2
Konferenz	N
ACM Innovation and Technology in Computer Science Education Conference (ITiCSE)	10
International Conference on Intelligent Tutoring Systems (ITS)	5
International Computer Assisted Assessment Conference (CAA)	4
International Conference on Model Driven Engineering Languages and Systems (MODELS)	4
ASEE Annual Conference & Exposition	2
World Conference on Educational Multimedia, Hypermedia and Telecommunications	2
International Conference on Artificial Intelligence in Education (AIED)	2
International Conference on Computer Supported Education (CSEDU)	2
International Conference on Computers in Education (ICCE)	2
International Conference on the Theory and Application of Diagrams (Diagrams)	2
Koli Calling Conference on Computing Education Research	2
Symposium on Languages, Applications and Technologies (SLATE)	2

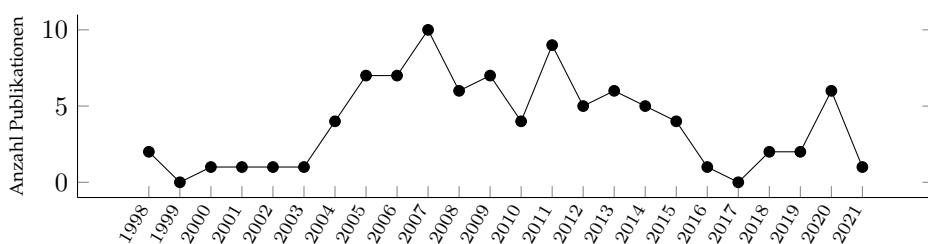


Abb. 3.13.: Verteilung der Publikationen aus der Ergebnismenge über Jahre [UI23a]

Publikationen im Februar 2021 durchgeführt wurde, sollte das Jahr 2021 bei der Betrachtung außer Acht gelassen werden.

Tab. 3.9 stellt einen Überblick über die Modelltypen dar, die in den in der Studie berücksichtigten Publikationen betrachtet werden. Zusätzlich wurde in 27 Publikationen herausgestellt, dass der vorgeschlagene Ansatz von universeller Natur in Bezug auf graphbasierte Diagramme ist (z. B. durch die Verwendung eines generischen Begriffs im Titel wie »graph-based diagram« oder indem dies im Text explizit unterstrichen wurde). Von diesen 27 Publikationen beschreiben 22 eine Implementierung und Evaluation ihres Ansatzes für einen bestimmten Modelltyp (und wurden somit auch in Tab. 3.9 berücksichtigt), wohingegen 5 Publikationen vollständig von einem

3. Verwandte Arbeiten

Tab. 3.9.: Modelltypen, die in den Publikationen in der Ergebnismenge betrachtet werden. Erweitert und übersetzt nach [UI23a]

Modelltyp	N	Modelltyp	N
UML-Klassendiagramm	40	Erweitertes ER-Diagramm	2
ER-Diagramm	38	CASE-Diagramm	1
UML-Sequenzdiagramm	7	BPMN	1
UML-Anwendungsfalldiagramm	6	Entscheidungsbaum	1
Deterministischer endlicher Automat	5	Nichtdeterministischer endl. Automat	1
Schaltplan (digital)	4	Kellerautomat	1
UML-Aktivitätsdiagramm	4	Blockdiagramm	1
UML-Zustandsdiagramm	3	EPK	1
Flussdiagramm	3	Netzplan (Metra-Potenzial-Methode)	1
Gefärbtes Petri-Netz	2	Endlicher Automat (generisch)	1

konkreten Modelltyp abstrahieren. Wie bereits erwähnt, sollten diese Zahlen mit Vorsicht interpretiert werden, da einige Ansätze in der Ergebnismenge überrepräsentiert sind. Zudem beschreiben einige Publikationen Ansätze, die für mehr als einen Modelltyp eingesetzt werden können. Im Anhang A ist in Tab. A.1 eine komplette Auflistung aller berücksichtigten Publikationen zusammen mit der Angabe der jeweils behandelten Modelltypen beigefügt.

SLR-FF1: Untersuchte Aspekte Ein großer Anteil der analysierten Publikationen beschreibt Ansätze, die syntaktische Aspekte untersuchen (71/110). Diese beziehen sich auf die Überprüfung der Einhaltung von Regeln der eingesetzten Modellierungssprache. Ein noch größerer Anteil der analysierten Publikationen beschreibt Ansätze, die semantische Aspekte untersuchen (101/110). Hierbei geht es um die inhaltliche Korrektheit von studentischen Lösungen im Zusammenhang mit der Modellierungsaufgabe. Dahingegen werden Ansätze, die sich pragmatischen Aspekten⁵² befassen, z. B. anhand von Richtlinien zur Modellverständlichkeit (siehe auch Abschnitt 3.2.2), nur in einem sehr kleinen Anteil der Publikationen beschrieben (6/110). Werden Modelle für den Einsatzzweck der Kommunikation mit einem größeren Anwenderkreis erstellt, spielen pragmatische Aspekte jedoch eine große Rolle, um den Erfolg von Modellierungsprojekten zu gewährleisten [HFL12]. Daher ist es naheliegend, dass pragmatischen Aspekten von Modellen im Kontext Lehren und Lernen der Modellierung die entsprechende Bedeutung zukommen sollte. Damit einhergehend sollten diese Aspekte dann auch verstärkt in zukünftigen Systemen

⁵² Wie in Abschnitt 3.2.1 beschrieben, wird die pragmatische Qualität im SEQUAL-Framework [KLS95] als Übereinstimmung zwischen einer Modelldenotation M und deren Interpretation I durch den Leser definiert. Ohne die Interpretation I beobachten zu können, lassen sich lediglich bekannte Einflussfaktoren auf die pragmatische Qualität wie z. B. Lesbarkeit und Verständlichkeit bestimmen.

3.3. Automatisierte Bewertung graphbasierter Diagramme

zur automatisierten Bewertung von Modellen untersucht werden können. Ebenso wurde bei der Literaturanalyse deutlich, dass die Wechselwirkungen zwischen syntaktischen und semantischen Aspekten von Modellen (nach [Mo02], siehe auch Abschnitt 3.2.1 auf S. 75) im Zusammenhang mit der automatisierten Bewertung eine besondere Herausforderung darstellen. Beide dieser offenen Punkte (Einbezug pragmatischer Aspekte sowie Wechselwirkungen zwischen syntaktischen und semantischen Aspekten) werden bei der Entwicklung der Bewertungsdienste im Rahmen dieser Dissertation berücksichtigt (siehe Abschnitt 5.3).

SLR-FF2: Technische Realisierung In Bezug auf die technische Realisierung wurden drei Publikationen der berücksichtigten 110 Publikationen von der Untersuchung ausgeschlossen, da sie keine ausreichenden Informationen über die technische Realisierung lieferten. Von den verbleibenden 107 Publikationen beschreiben 78 Publikationen den Einsatz von Verfahren zum Modellvergleich. Hier wird das zu bewertende Modell mit einer (oder mehreren) Musterlösung(en) verglichen. Dies geschieht beispielsweise über Ähnlichkeitsmessungen [Fa20; SL15; SM08; VP19] oder Graph-Matching [OMA21; RM19]. Der Einsatz von regelbasierten Ansätzen, die bestimmte Kriterien überprüfen, wird in 41 Publikationen beschrieben. Einige Beispiele dafür, in welcher unterschiedlicher Form die Regeln dabei gefasst werden können, sind Graph-Anfragen [SG11], Eigenschaften [WFS12], Metriken [Li19], Defekte [HR11] oder Suchmuster [VSV19]. Zu den regelbasierten Ansätzen wurden auch die sogenannten »constraint-based«-Ansätze aus dem Forschungsgebiet der »intelligent tutoring systems« (ITS) (z. B. [BM07a; MS15a]) gezählt (15 Publikationen). Vier jüngere Arbeiten präsentieren Ansätze, die das Potenzial maschineller Lernverfahren ausnutzen [BMM20; Li19; RM19; St19]. Weitere Beispiele für spezifische Techniken, die nicht zu den oben genannten drei Kategorien (Modellvergleich, regelbasierte Ansätze und Methoden des maschinellen Lernens) gehören, sind die Simulation des zu bewertenden Modells [Ja18; OK19; OKO19], Teststrategien [Be15; Sh14], die Cluster-Bildung von Modellen [A118; BH06; BSH09] sowie der Abgleich des zu bewertenden Modells mit einer annotierten textuellen Beschreibung [Sà20]. Die hier aufgeführten Kategorien sind nicht exklusiv. Die Mehrheit der Publikationen wurde zwar einer einzelnen Kategorie oder Technik zugeordnet (70/107), allerdings treten bei 37 der 107 Publikationen Kombinationen auf. Häufig handelt es sich um einen Ansatz basierend auf dem Modellvergleich, der um regelbasierte Ansätze ergänzt wird (z. B. um die syntaktische Korrektheit zu überprüfen [CLP18; TWS08]).

Weiterhin wurde im Kontext dieser Forschungsfrage ebenso untersucht, welche Werkzeuge und Techniken für die Analyse von Beschriftungen in graphbasierten Dia-

3. Verwandte Arbeiten

grammen zum Einsatz kommen. Beispielsweise wird in 38 Publikationen angegeben, dass Techniken zur Handhabung von Rechtschreibfehlern oder Synonymen in den Beschriftungen von Modellelementen verwendet werden. Dies ist in der Regel bei einem hohen Freiheitsgrad bei der Wahl der Beschriftungen von Modellelementen erforderlich, um Modellelemente in den zu bewertenden Modellen korrekt identifizieren zu können. Anhand einer solchen Vorverarbeitung wird dann die Grundlage für den anschließenden Einsatz von Methoden des Modellvergleichs oder regelbasierte Ansätze gebildet. Beispiele für derartige Werkzeuge und Techniken sind die Berechnung der Levenshtein-Distanz ([TSW09a; TWS09], siehe auch Metrik 3.1 in Abschnitt 3.2.3), reguläre Ausdrücke [DW09; RK19], lexikalische Datenbanken für die Synonymauflösung, Wörterbücher und Soundex-Datenbank für die Rechtschreibprüfung [VPG12], individuell erstellte domänenspezifische Synonym-Datenbanken [TSW09b] sowie Stemming-Algorithmen [VPG12]. Drei Publikationen in der Ergebnismenge beschäftigen sich eingehend mit dem Problem des sogenannten Label-Matching (Identifikation zusammengehöriger Modellelemente in zu bewertendem Modell und Musterlösung anhand der Beschriftungen) und präsentieren ausgereifte Ansätze, die von den meisten zuvor erwähnten Werkzeugen und Techniken Gebrauch machen [JS09; TSW09b; VPG12].

Ebenso wurde der Automatisierungsgrad der Bewertung (siehe Tab. 2.1 in Abschnitt 2.1.3) der Ansätze in allen 110 Publikationen unter die Lupe genommen: 88 Publikationen stellen einen vollständig automatisierten Ansatz vor. Die verbleibenden 22 Publikationen wurden als semi-automatisiert (d.h. anteilig manuell bzw. automatisiert) klassifiziert, da manuelle Aktivitäten während dem Bewertungsprozess notwendig sind. Beispielsweise werden unbekannte oder uneindeutige Modellelemente manuell bewertet [Al18; BH07; Ic19; Si16], oder die manuelle Überprüfung und Korrektur der automatisierten Bewertung wird explizit gefordert [CM07; OKO19]. Ebenso wurden die vier genannten Ansätze auf Basis von maschinellen Lernverfahren zu den semi-automatisierten Bewertungsverfahren gezählt, da hier Trainingsdaten aus früheren manuellen Bewertungen benötigt werden.

Die hier vorliegenden Ergebnisse sollen einen breiten Überblick über die technischen Verfahren darstellen, die im Kontext der automatisierten Bewertung von Modellen zum Einsatz kommen. Da in den untersuchten wissenschaftlichen Publikationen allerdings häufig nur eine vereinfachte und rudimentäre Beschreibung der technischen Implementierung dargestellt wird, war es nicht möglich ein präzises Klassifikationssystem abzuleiten und eine zuverlässige Zuordnung zu bestimmten Kategorien zu ermöglichen. Dies würde zusätzliche Nachforschungen erfordern (z. B. Inter-

3.3. Automatisierte Bewertung graphbasierter Diagramme

views mit den Entwicklern oder – sofern möglich – die Analyse des Quelltexts). Zudem wurde durch einen Blick auf die Evaluationen mit Vergleich von automatisierten Bewertungsansätzen mit den Ergebnissen manueller Bewertungen deutlich, dass noch immer Raum für Verbesserungen in Bezug auf die Zuverlässigkeit der automatisierten Bewertung vorhanden ist [OKO19; St19; Th13]. Um dies anzugehen, wäre ein Vergleich der identifizierten Ansätze zur automatisierten Bewertung im Hinblick auf deren Zuverlässigkeit wünschenswert. Idealerweise würde dieser Vergleich aufzeigen, welche Kategorien von Verfahren demnach am besten abschneiden (und für welche Modelltypen) und welche Stärken und Schwächen sie jeweils besitzen. Dies würde entweder eine vertiefte Auseinandersetzung mit den gegebenenfalls bestehenden und recht unterschiedlich angelegten Evaluationen in den berücksichtigten Publikationen (Meta-Studie) erfordern oder aber die Durchführung einer größer angelegten vergleichenden Evaluation, bei der verschiedene (öffentlich nutzbare/erhältliche) Ansätze auf Basis eines einheitlichen Datensatzes von studentischen Modellen miteinander verglichen werden. Auf einer derartigen Ergebnisgrundlage könnten passende und vielversprechende Ansätze zur automatisierten Bewertung von Modellen identifiziert werden, sei es für den Einsatz in der eigenen Lehrveranstaltung oder für eine Weiterentwicklung in Bezug auf individuelle Anforderungen. Die im Forschungsvorhaben entstandene E-Assessment-Plattform ist aufgrund ihrer offenen Architektur für eine derartige vergleichende Evaluation verschiedener Ansätze geeignet und könnte in Zukunft hierbei einen Beitrag leisten (siehe Kapitel 5).

SLR-FF3: Kompetenzen und Lernziele Ein überwältigender Großteil der Publikationen (98/110) nähert sich dem Thema von einer rein technischen Perspektive aus und liefert keine oder nur sehr vage und abstrakte Hinweise auf zugrundeliegende Kompetenzen, Lernziele oder didaktischen und lerntheoretische Ansätze. Dabei wird der Bedarf an einer automatisierten Bewertung in vielen Fällen hauptsächlich durch eine hohe Studierendenzahl und den Aufwand der manuellen Bewertung von Übungs- und Klausuraufgaben begründet, ohne dabei spezifische Kompetenzen oder zu trainierende Fähigkeiten zu benennen. Letzteres ist lediglich bei vier Publikationen der Fall. In [RK16] werden abstrakt »higher-level software engineering skills«, in [Ic19] etwas konkreter z. B. »ability to create a model«, »ability to understand requirements of software« unter Verweis auf SENDALL und KOZACZYNSKI [SK03] sowie in [BM07b; BMI07] die grundlegende Idee des »computer-supported collaborative learning« aufgeführt. Eine weitere Gruppe von vier Publikationen bezieht sich auf Kompetenzen und Lernziele durch die explizite Nennung der Lernzieltaxo-

3. Verwandte Arbeiten

nomien von BLOOM [Bl73] und ANDERSON und KRATHWOHL [AK01] (siehe auch Abschnitt 2.1.2). Dabei wird die Entwicklung eines Ansatzes zur automatisierten Bewertung von Modellen damit motiviert, dass höhere Stufen in diesen Taxonomien erreicht werden sollen (»Anwenden« auf der Prozessdimension) und um sich von generischen E-Assessment-Systemen abzugrenzen, die lediglich den Einsatz von geschlossenen, bzw. konvergenten Aufgaben (siehe auch Abschnitt 2.1.3) ermöglichen. Drei weitere Publikationen verweisen auf andere Ansätze: dem Lernen aus Fehlern [ZMO05], das »Cognitive Apprenticeship Model« [If12] sowie empirischer Forschung zu Schwierigkeiten bei der objektorientierten Modellierung [PAA13]. Die spärliche Auseinandersetzung mit didaktischen Aspekten im Zusammenhang mit der automatisierten Bewertung lässt allerdings nicht den Schluss zu, dass eine solche nicht stattfindet oder stattgefunden hat. Es ist durchaus möglich, dass didaktische Aspekte eher in Publikationen mit einem anderen Schwerpunkt zu dem gleichen Ansatz veröffentlicht wurden und in der Literaturstudie aufgrund der Einschränkungen auf tatsächlich implementierte Ansätze zur automatisierten Bewertung nicht erfasst wurden. Allerdings weist die Nennung des hohen Aufwands der manuellen Korrektur als häufig genannte Motivation zur Entwicklung einer automatisierten Bewertung aufgrund darauf hin, dass durch die Automatisierung lediglich die manuelle Arbeit ersetzt werden soll. Ist dies tatsächlich der Fall, wird an dieser Stelle die Gelegenheit versäumt, eine kritische Reflektion der zu überprüfenden Kompetenzen und Lernziele vorzunehmen oder auf bestehende didaktische und lerntheoretische Ansätze zurückzugreifen, um das Lehren, Lernen und die Bewertung im Kontext der Modellierung grundsätzlich voranzubringen. Im Rahmen dieser Dissertation wurde von Beginn an das Ziel verfolgt, Kompetenzen und Lernziele der Modellierung im Kontext der automatisierten Bewertung zu berücksichtigen. Der Bildung einer fundierten hochschuldidaktischen Grundlage kommt dabei im KEA-Mod-Projekt in interdisziplinärer Zusammenarbeit mit dem Verbundpartner aus dem Gebiet der Psychologie und Hochschuldidaktik eine bedeutende Rolle zu (siehe Kapitel 4).

SLR-FF4: Feedback In den untersuchten Publikationen wird das vom System automatisiert generierte Feedback überwiegend nicht explizit oder nur vage im Text beschrieben. Nichtsdestotrotz konnte die Art des Feedbacks in vielen Fällen aus Abbildungen und Screenshots abgeleitet werden. Nur in 7/110 Publikationen konnten nicht genügend Informationen über das Feedback gewonnen werden. Bei den anhand der verbleibenden 103 Publikationen identifizierten Feedbackarten handelt es sich um a) textuelles Feedback, b) grafisches Feedback und c) numerisches Feedback. Textuelles Feedback besteht zumeist aus vordefinierten Textblöcken, die auf Basis des

3.3. Automatisierte Bewertung graphbasierter Diagramme

jeweiligen Bewertungsergebnisses generiert werden. Es bezieht sich in generischer Form entweder auf das gesamte Modell oder spezifisch auf einzelne Modellelemente. In einigen Fällen wird das textuelle Feedback auch in abgestufter Form generiert. Ein zweistufiges Verfahren zur Beschreibung der identifizierten Designmuster findet sich z. B. in [RK19], ein mehrstufiges Verfahren wird von [CLP18] eingesetzt, um nach und nach detailliertere Verbesserungshinweise zu geben, wenn eine falsche Lösung hochgeladen wird. Grafisches Feedback wird direkt in das studentische Modell integriert und dort angezeigt. Dabei werden betreffende Modellelemente grafisch hervorgehoben (z. B. durch Verwendung anderer Farben oder Linienbreite), häufig in Kombination mit spezifischem textuellem Feedback. Eine weitere Möglichkeit ist die Überlagerung der studentischen Lösung mit der Musterlösung (Overlay), um Abweichungen erkennbar zu machen [TSW07b]. Als numerisches Feedback werden weiterhin rechnerische Ergebnisse der automatisierten Bewertung bezeichnet, sei es eine Gesamtpunktzahl (z. B. [BAK19]), eine Prozentangabe (z. B. [Ec17; SSL13]) oder eine skalenbasierte Angabe wie eine Note (z. B. [SM08; TSW07a]). In 44/103 Publikationen wird nur eine einzige der oben beschriebenen Feedbackarten eingesetzt, obwohl eine Kombination grundsätzlich möglich ist. Die Verwendung aller drei Feedbackarten konnte jedoch nur in 11/103 Publikationen festgestellt werden (z. B. [BAK19; TS07; TWS08]). Insbesondere bleibt anzumerken, dass die Art des Feedbacks in keiner der untersuchten Publikationen durch den Lehrenden oder Lernenden steuer- bzw. konfigurierbar ist. Die optimale Art sowie auch Inhalt und Detailtiefe des Feedbacks hängt jedoch stark mit den persönlichen Präferenzen der Lehrenden und Lernenden sowie auch mit dem Einsatzkontext der automatisierten Bewertung zusammen, so dass die Möglichkeit der Anpassung des automatisiert generierten Feedbacks an die jeweiligen Bedürfnisse durchaus sinnvoll erscheint. Die im Forschungsvorhaben entwickelte E-Assessment-Plattform ist darauf ausgerichtet, eine Kombination aller drei Feedbackarten sowie eine individuelle Anpassung und Steuerung des Feedbacks durch Lehrende zu ermöglichen (siehe Kapitel 5).

SLR-FF5: Zeitpunkt und Kontext des Einsatzes Ein deutlich überwiegender Anteil der untersuchten Publikationen beschreibt den Einsatz von automatisierten Bewertungsansätzen als formatives Assessment (57/110). Nur gering fällt der Anteil an Publikationen aus, die Werkzeuge für die automatisierte Bewertung als summatives Assessment vorstellen (7/110). Allerdings werden in den untersuchten Publikationen auch Ansätze beschrieben, die sich für beide Einsatzzeitpunkte eignen (12/110), zudem findet sich in zahlreichen Publikationen keine klare Aussage in Bezug auf den Einsatzzeitpunkt (34/110). Keine der Publikationen beschreibt den Einsatz als dia-

3. Verwandte Arbeiten

agnostisches Assessment (siehe Abschnitt 2.1.3). Der Einsatzkontext beim summativen Assessment ist naheliegend: hier geht es um die Bewertung von studentischen Einreichungen bei Prüfungen oder Klausuren. Bei den Publikationen, die ein formatives Assessment beschreiben, variiert der Einsatzkontext. Genannt werden Übungsblätter, die ebenfalls nach der Einreichung bewertet werden, jedoch ist hier im Gegensatz zum summativen Assessment die Wiedereinreichung nach einer Überarbeitung möglich (20/110). Ebenso werden Szenarien beschrieben, bei denen das Feedback durch die automatisierte Bewertung direkt während der Bearbeitung (Erstellung von Modellen) stattfindet (20/110). Weiterhin finden sich auch Ansätze, bei denen der Einsatz der automatisierten Bewertung während einer (Präsenz-)Veranstaltung »in class« (7/110), beim kollaborativen Lernen (3/110) und als freiwillige Lernhilfe (7/110) stattfindet. Dass viele Ansätze keine Einschränkung auf einen bestimmten Einsatzkontext besitzen ist einerseits vielversprechend: es existiert demnach bereits eine Reihe von Systemen die nur lose an einen spezifischen Einsatzkontext gekoppelt sind. Andererseits besitzt jeder Einsatzkontext unterschiedliche Anforderungen rund um die Ausführung der automatisierten Bewertung. Anstatt spezialisierte Ansätze zu entwerfen, die sich auf spezifische Einsatzkontexte konzentrieren, ist vielmehr der Bedarf an entsprechenden Konfigurationsoptionen zu erkennen. Mit Hilfe ebensolcher könnten Lehrende in die Lage versetzt werden, ein generell gehaltenes System zur automatisierten Bewertung auf die Anforderungen des jeweiligen Einsatzkontextes anzupassen. Beispielsweise könnten Konfigurationsoptionen die Art und Detailtiefe des Feedbacks steuern, wie bereits im vorangegangenen Absatz zu Forschungsfrage SLR-FF4 beschrieben. Andere mögliche Parameter sind z. B. Zeitbeschränkungen für die Bearbeitung im Prüfungskontext, die Aktivierung bzw. Deaktivierung einer Modellierungsunterstützung (integrierte syntaktische Prüfung im Modellierungswerkzeug) oder die Freigabe von Modellen unter Studierenden für die kollaborative Bearbeitung oder einem Peer-Review. Diese Grundidee eines flexiblen Systems, das für verschiedene Einsatzkontexte konfiguriert werden kann, spiegelt sich in der E-Assessment-Plattform wieder, die im Rahmen des Forschungsvorhabens entwickelt wurde (siehe Kapitel 5).

3.3. Automatisierte Bewertung graphbasierter Diagramme

Tab. 3.10.: Überblick über den aktuellen Stand und offene Punkte im Zusammenhang mit den betrachteten Forschungsfragen, übersetzt nach [U123a]

Aktueller Stand	Offene Punkte
SLR-FF1: <i>Welche Aspekte von Modellen werden bei der automatisierten Bewertung untersucht?</i>	
<ul style="list-style-type: none"> • Die meisten Publikationen beschreiben Ansätze, die syntaktische und semantische Aspekte untersuchen • Pragmatische Aspekte wie Modellverständlichkeit werden nur in wenigen Publikationen untersucht 	<ul style="list-style-type: none"> • Stärkere Berücksichtigung von pragmatischen Aspekten bei der automatisierten Bewertung sowie bei Lerninhalten und -zielen • Vertiefte Auseinandersetzung mit der Wechselwirkung von syntaktischen und semantischen Aspekten
SLR-FF2: <i>Welche Verfahren kommen bei der technischen Umsetzung der automatisierten Bewertung von Modellen zum Einsatz?</i>	
<ul style="list-style-type: none"> • Modellvergleich und regelbasierte Ansätze sind weit verbreitete Methoden • Jüngere Publikationen beschreiben zunehmend Ansätze auf Basis maschineller Lernverfahren • Für die Verarbeitung von Beschriftungen existieren bereits ausgeklügelte Ansätze 	<ul style="list-style-type: none"> • Bedarf an einer Meta-Studie zum Vergleich der Zuverlässigkeit verschiedener Methoden zur automatisierten Bewertung • Stärkerer Fokus auf die Entwicklung von sprachübergreifenden bzw. universellen Ansätzen für graphbasierte Diagramme • (Wieder-)Verwendung und Weiterentwicklung bereits existierender Ansätze
SLR-FF3: <i>Werden didaktische Aspekte wie Kompetenzen oder Lernziele für die Modellierung im Kontext der automatisierten Bewertung berücksichtigt?</i>	
<ul style="list-style-type: none"> • Bedarf an einer automatisierten Bewertung hauptsächlich durch den Aufwand einer manuellen Bewertung in Veranstaltungen mit hohen Teilnehmerzahlen motiviert • Didaktische Aspekte werden nur selten explizit aufgeführt (und wenn, nur vage) 	<ul style="list-style-type: none"> • Referenzen zu lerntheoretischen Ansätzen sollten im Kontext der automatisierten Bewertung explizit gemacht werden, z. B. durch die Verknüpfung von Bewertungsergebnissen und gewünschten Kompetenzen oder Lernzielen
SLR-FF4: <i>Welche Art von Feedback für Lernende wird bei der automatisierten Bewertung von Modellen produziert?</i>	
<ul style="list-style-type: none"> • Feedback wird in textueller, grafischer oder numerischer Form dargestellt (oft nur anhand der Abbildungen abgeleitet) 	<ul style="list-style-type: none"> • Möglichkeit zur Anpassung des Typs und der Detailtiefe beim Feedback in Abhängigkeit vom Anwendungskontext
SLR-FF5: <i>Zu welchem Zeitpunkt im Lernprozess (formativ oder summativ) und in welchem Kontext (Lehrmethode, Kursform) wird die automatisierte Bewertung von Modellen eingesetzt?</i>	
<ul style="list-style-type: none"> • Automatisierte Bewertung von Modellen kommt zu allen Zeitpunkten im Lernprozess zum Einsatz, besonders häufig ist der formativ Einsatz • Ansätze sind häufig nicht auf einen bestimmten Einsatzkontext eingeschränkt 	<ul style="list-style-type: none"> • Notwendigkeit für spezialisierte Konfigurationsoptionen, um ein generisches System zur automatisierten Bewertung im Hinblick auf individuelle Anforderungen oder ein bestimmtes Einsatzszenario anzupassen

Teil II

Bearbeitung der Forschungsfragen

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

In diesem Kapitel werden die didaktischen Ergebnisse aus dem KEA-Mod-Projekt (siehe Abschnitt 1.4) vorgestellt, die im Rahmen von Forschungsfrage FF1 (siehe Abschnitt 1.2) erarbeitet wurden. Hauptverantwortlich für die Planung und Durchführung war der Verbundpartner aus der Hochschuldidaktik (UPB). In der interdisziplinären Zusammenarbeit übernahmen die drei Verbundpartner KIT (in Person der Verfasserin dieser Dissertation), DFKI und UDE die Rolle der fachlichen Unterstützung von Seiten der Informatik. Es wurden zahlreiche gemeinsame Aktivitäten zur Konstruktion und Validierung der im Kontext von FF1 gebildeten Artefakte sowie regelmäßige Besprechungen zur Abstimmung in Bezug auf fachliche und methodische Aspekte durchgeführt. Ohne die Zusammenarbeit der beiden Disziplinen (Psychologie bzw. Hochschuldidaktik) und Informatik wäre die Bearbeitung der Forschungsfrage nur eingeschränkt möglich gewesen. Im Kontext von FF1 sind mehrere wissenschaftliche Publikationen entstanden, an denen die Verfasserin dieser Dissertation beteiligt war. Sie werden in den folgenden Abschnitten an entsprechender Stelle zitiert.

Unabhängig vom Modus (digital, analog) oder dem Zeitpunkt eines Assessments (formativ, summativ) umfasst das Prüfen sowohl das Stellen passender Aufgaben zu den gewünschten Lernzielen sowie die Bewertung von Einreichungen zu diesen Aufgaben. Nach METZGER und NÜESCH [MN04] sind beim kompetenzorientierten Prüfen stets die drei zentralen Fragen »Was wird geprüft« (Frage I), »Wie wird geprüft« (Frage II) und »Wie wird bewertet?« (Frage III) zu betrachten. Dementsprechend setzen sich die wesentlichen Ergebnisse der interdisziplinären Zusammenarbeit zum Thema kompetenzorientiertes Prüfen auf dem Gebiet der grafischen Modellierung aus einem Kompetenzmodell (Frage I, Abschnitt 4.1), einem Aufgabenkatalog (Frage II, Abschnitt 4.2) und Vorschlägen für kriteriengeleitete und kompetenzorientierte

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

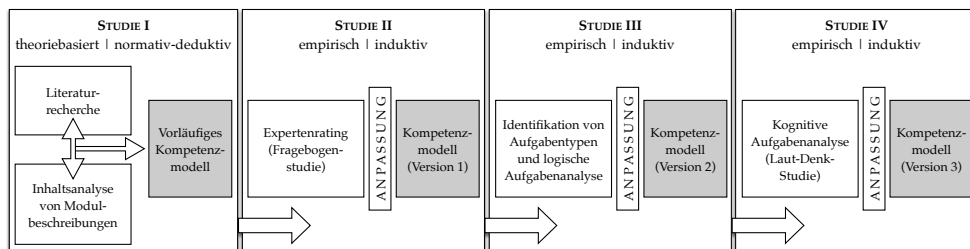


Abb. 4.1.: Entwicklungsprozess des Kompetenzmodells, in Anlehnung an [So22f].

Bewertungsschemata als sogenannte Rubrics sowie Feedback-Nachrichten (Frage III, Abschnitt 4.3) zusammen. Sie bilden das didaktische Fundament für die technische Umsetzung der E-Assessment-Plattform für die grafische Modellierung (Kapitel 5).

4.1. Kompetenzmodell

Wie bereits in den Grundlagen (siehe Abschnitt 2.1.2) ausgeführt, ist es im Sinne eines Constructive Alignment notwendig, Prüfungen auf die erwünschten Lernziele und die dahinterstehenden Kompetenzen auszurichten. Somit muss als erstes die grundlegende Frage geklärt werden, welche Kompetenzen auf dem Gebiet der grafischen Modellierung erforderlich sind. Im Hinblick darauf wurde im Rahmen des KEA-Mod-Projekts ein Kompetenzmodell schrittweise entwickelt, validiert und angepasst. Eine detaillierte Beschreibung des Entwicklungsprozesses wurde in [So22f] veröffentlicht. Abb. 4.1 liefert einen Überblick über die vier Studien, die zur Entwicklung des Kompetenzmodells beigetragen haben.

In den ersten beiden Studien I und II (siehe [So22f]) lag der Fokus darauf, relevante Kompetenzen für die grafische Modellierung zu identifizieren und adäquat zu strukturieren. Gemäß den Empfehlungen von SCHAPER [Sc09] wurden dabei ein theoriebasierter (Studie I) und ein empirischer Ansatz (Studie II) zur Kompetenzmodellierung kombiniert. In der ersten Studie wurden zunächst die Struktur des Kompetenzmodells sowie relevante Kompetenzbereiche auf der Grundlage von Literaturrecherchen zur Kompetenzmodellierung in der Lehre und Fachliteratur zur grafischen Modellierung abgeleitet (deduktives Verfahren). Bezüglich der Fachliteratur wurden die in Kapitel 3 vorgestellten Arbeiten berücksichtigt. Entsprechend eines normativen Ansatzes wurde zusätzlich eine Inhaltsanalyse bestehender Modul- und Kursbeschreibungen auf dem Gebiet der grafischen Modellierung durchgeführt,

die durch die Verbundpartner zur Verfügung gestellt wurden. Dadurch konnten relevante Kompetenzfacetten identifiziert und in die zuvor erstellte Struktur des Kompetenzmodells integriert werden.

In Studie II wurde das in Studie I entwickelte vorläufige Kompetenzmodell von Fachexperten überprüft und mittels eines Online-Fragebogens bewertet, um die inhaltliche Validität des Modells zu prüfen und Verbesserungsvorschläge umzusetzen. An diesem Expertenrating im zweiten Halbjahr des Jahres 2020 nahmen 97 Personen teil. Nach Ausschluss unvollständiger Fragebögen und Personen mit geringer Modellierungserfahrung konnten 78 Fragebögen berücksichtigt werden. Basierend auf den Ergebnissen des Expertenratings¹ wurde das Kompetenzmodell umfassend überarbeitet. Dabei wurden insbesondere ähnliche Kompetenzfacetten zu verschiedenen Anwendungsbereichen der grafischen Modellierung (z.B. Softwareentwicklung und Geschäftsprozessmodellierung) zusammengeführt und die Formulierung der Kompetenzfacetten anhand eines eingeführten Glossars angepasst, so dass diese eine einheitliche Terminologie aufweisen (induktives Verfahren). Somit handelt es sich um ein generisches Kompetenzmodell für die grafische Modellierung, welches von spezifischen Inhalten für bestimmte Anwendungsbereiche abstrahiert. Die Kompetenzfacetten sind möglichst allgemeingültig formuliert, um möglichst viele Anwendungsbereiche der Modellierung zu umfassen, ohne auf bestimmte Fachgebiete oder Modellierungssprachen Bezug zu nehmen.

In Studie III (siehe [So23b]) wurde zunächst eine Identifikation von Aufgabentypen durchgeführt und anschließend das Kompetenzmodell (Version 1) im Rahmen einer logischen Aufgabenanalyse untersucht (siehe auch Abschnitt 4.2). Dadurch konnte seine Eignung für die Analyse und Entwicklung von Prüfungsaufgaben überprüft und ein weiterer Beleg für die Validität (bezüglich Angemessenheit, Relevanz und Repräsentativität) geliefert werden. Dabei wurden auch geringfügige Anpassungen an den Kompetenzfacetten vorgenommen. In Studie IV wurde eine kognitive Aufgabenanalyse anhand der Laut-Denk-Methode durchgeführt (siehe auch Abschnitt 4.2). Bei dieser Methode wird die teilnehmende Person aufgefordert, ihre Gedanken, Gefühle und Entscheidungen beim Lösen einer exemplarischen Modellierungsaufgabe kontinuierlich laut auszusprechen. Mit Hilfe einer Videoaufzeichnung werden zusätzliche Beobachtungen (z. B. Blickrichtung, Bewegungen und Handlungen) protokolliert. Anhand der Laut-Denk-Protokolle wurde untersucht, ob a) mit den gestellten Modellierungsaufgaben tatsächlich die im Kompetenzmodell (Version 2) postulierten Kompetenzfacetten angesprochen werden und ob b)

1 Eine detaillierte Auswertung ist in [So22f] zu finden.

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

INHALTSDIMENSION		PROZESSDIMENSION			
		Verstehen	Anwenden & Übertragen	Analysieren & Bewerten	Erschaffen
Fachliche(s) Wissen & Prozeduren	Modellverstehen & -interpretieren	MV 1	MV 2	MV 3	MV 4
	Modellbilden & -modifizieren	MB 1	MB 2	MB 3	MB 4
Werte / Haltungen / Überzeugungen		WH 1	WH 2	WH 3	WH 4
Fachübergreifende(s) Wissen & Fähigkeiten	Metakognitive(s) Wissen & Fähigkeiten	MK 1	MK 2	MK 3	MK 4
	Sozial-kommunikative(s) Wissen & Fähigkeiten	SK 1	SK 2	SK 3	SK 4
Komplexität und Kontextualität der Modellierungsaufgabe					

Abb. 4.2.: Struktur des Kompetenzmodells für die grafische Modellierung, in Anlehnung an [So22f]. Für die grau markierten Kompetenzbereiche wurden Kompetenzfacetten definiert, die in [So22b] vollständig aufgelistet werden.

die Kompetenzfacetten die tatsächlichen kognitiven Prozesse verständlich und auf der passenden Stufe der Prozessdimension beschreiben. Durch Studie IV konnte das Kompetenzmodell weiterhin inhaltlich sowie kognitiv validiert werden. Einige Kompetenzfacetten wurden umformuliert, anderen Prozessstufen zugeordnet und fehlende Kompetenzfacetten ergänzt.

In Abb. 4.2 ist die Struktur des Kompetenzmodells (Version 3) als Ergebnis von Studie IV dargestellt. Es ist als zweidimensionale Matrix mit einer Prozessdimension (basierend auf der Lernzieltaxonomie von ANDERSON und KRATHWOHL [AK01] (siehe auch Abschnitt 2.1.2) und einer Inhaltsdimension (basierend auf der Kompetenzdefinition nach WEINERT [We01] (siehe auch Abschnitt 2.1.2) aufgebaut. Die Kompetenzbereiche des Kompetenzmodells entsprechen einzelnen Zellen der Matrix und kombinieren dabei jeweils einen Inhaltsbereich und eine Prozessstufe. Sie sind mit entsprechenden Kürzeln versehen, z. B. »MB 2« für den Inhaltsbereich »Modellbilden & -modifizieren« auf der Prozessstufe »Anwenden & Übertragen«.

Auf der Inhaltsdimension enthält das Kompetenzmodell zunächst fachliches Wissen und Prozeduren der Inhaltsbereiche »Modellverstehen & -interpretieren« und »Modellbilden & -modifizieren«, die als zentrale Handlungsbereiche der grafischen Modellierung in der beruflichen Praxis und der Hochschullehre identifiziert wurden. Die über das Verstehen und Erstellen von Modellen hinausgehende Verwendung von Modellen (z. B. im Rahmen von »Model Driven Engineering« oder zur Simulati-

on) ist durch das Kompetenzmodell lediglich implizit durch die Berücksichtigung des jeweiligen Modellierungszwecks während der Modellbildung abgedeckt. Um der umfassenden Kompetenzdefinition nach WEINERT [We01] gerecht zu werden, sind auf der Inhaltsdimension weiterhin motivationale Aspekte im Inhaltsbereich »Werte / Haltungen / Überzeugungen« sowie fachübergreifende Kompetenzen in zwei weiteren Inhaltsbereichen angesiedelt. Der Inhaltsbereich »Metakognitive(s) Wissen & Fähigkeiten« berücksichtigt die Selbstreflexion und Regulation von Lern- und Problemlösungsprozessen im Kontext der Bearbeitung von Modellierungsaufgaben. Kompetenzen zur sozialen Interaktion, Teamarbeit oder z. B. zur zielgruppengerechten Präsentation von Modellen werden durch den Inhaltsbereich »Sozial-kommunikative(s) Wissen & Fähigkeiten« abgedeckt.

Die Prozessdimension des Kompetenzmodells ist an die Taxonomie von ANDERSON und KRATHWOHL [AK01] (siehe Abschnitt 2.1.2) angelehnt, um Kompetenzen entsprechend ihres kognitiven Anforderungsniveaus beschreiben und einordnen zu können. Da der Fokus des KEA-Mod-Projekts im Sinne der Kompetenzorientierung und des Kontexts Hochschullehre auf der Prüfung höherer Anforderungsniveaus liegt, wurde die niedrigste Prozessstufe »Erinnern« im Kompetenzmodell nicht berücksichtigt². Weiterhin wurden die Prozessstufen »Analysieren« und »Evaluieren« zu »Analysieren & Bewerten« zusammengefasst, da eine Abgrenzung dieser Stufen im Kontext der ermittelten Kompetenzfacetten als nicht zielführend erachtet wurde. Im Sinne einer kompetenz- und problemorientierten Gestaltung von Modellierungsaufgaben sollte zudem deren Kontextualität und Komplexität berücksichtigt werden. Variationen auf dieser Ebene können z. B. im Bereich der Modellbildung durch eine schrittweise Erhöhung des Freiheitsgrades bei der Modellierung, die Größe des zu erstellenden Modells oder die Einbettung in authentische und praxisrelevante Szenarien gesteuert werden.

Die in der Matrix des Kompetenzmodells (siehe Abb. 4.2) dargestellten Kompetenzbereiche enthalten jeweils eigene Kompetenzfacetten, die als Lernziele formuliert wurden. Da aus Platzgründen nicht alle 70 Kompetenzfacetten des Kompetenzmodells in der Abbildung dargestellt werden können, sind in der folgenden Tabelle 4.1 auszugsweise die Kompetenzfacetten des Kompetenzbereichs MB 4 »Modellbilden & -modifizieren« auf der Stufe »Erschaffen« aufgeführt. Da die Kompetenzfacetten für das Gebiet der grafischen Modellierung generisch sein sollen und weitgehend von konkreten Inhalten abstrahiert wurde, finden sich entsprechende Platzhalter an manchen Stellen in der Formulierung, z. B. »relevante Modellinhalte« (MB 4.04) oder

² Bei Bedarf können entsprechende Kompetenzfacetten, die das Erinnern theoretischen Wissens adressieren, aus den Inhalten einer Lehrveranstaltung abgeleitet werden.

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

Tab. 4.1.: Kompetenzfacetten für den Kompetenzbereich MB 4 »Modellbilden & -modifizieren« auf der Stufe »Erschaffen«

MB 4	Modellbilden & -modifizieren / Erschaffen
MB 4.01	Die Lernenden können selbstständig grafische Modelle (wie z. B. UML-Diagramme, ER-Modelle und Petri-Netze) erstellen, um einen Sachverhalt abzubilden.
MB 4.02	Die Lernenden können zueinander passende Sichten auf ein System oder einen Sachverhalt mit passenden Modellen (ergänzend und konsistent zu einander) erstellen.
MB 4.03	Die Lernenden können ein bestehendes Modell aufgrund neuer oder geänderter Anforderungen oder Inkonsistenzen anpassen bzw. weiterentwickeln und entsprechende Modellelemente hinzufügen, modifizieren oder entfernen.
MB 4.04	Die Lernenden können ein Modell in Bezug auf einen Sachverhalt a) semantisch korrekt und b) vollständig erstellen und sich dabei c) auf relevante Modellinhalte beschränken (Prägnanz).
MB 4.05	Die Lernenden können ein Modell a) auf Basis bekannter Richtlinien oder Konventionen gut lesbar und strukturiert sowie b) für die jeweilige Zielgruppe verständlich erstellen.
MB 4.06	Die Lernenden können selbstständig Bezeichner a) angemessen / verständlich und b) konventionskonform entwickeln sowie c) einheitlich verwenden.
MB 4.07	Die Lernenden können a) einen in Bezug auf den Modellierungszweck angemessenen Abstraktionsgrad bei der Erstellung eines Modells wählen und b) diesen innerhalb des Modells konsistent beibehalten.

»bekannter Richtlinien oder Konventionen« (MB 4.05). An dieser Stelle sind Lehrende gefragt, die Kompetenzfacetten auf individuelle Inhalte und Voraussetzungen auszurichten. Eine vollständige Auflistung aller Kompetenzfacetten im Kompetenzmodell (Version 3) wurde auf der Projekt-Homepage veröffentlicht [So22b], ebenso findet sich dort ein Glossar mit Definitionen der verwendeten zentralen Begriffe [So22c].

4.2. Aufgabenkatalog

Als Ergänzung zum Kompetenzmodell wurde ein Aufgabenkatalog entwickelt, der Aufgabentypen für das Gebiet der grafischen Modellierung zusammen mit den jeweils adressierten Kompetenzfacetten des Kompetenzmodells (siehe Abschnitt 4.1) aufführt. In diesem Zusammenhang wurden eine logische und eine kognitive Aufgabenanalyse durchgeführt, die im vorangegangenen Abschnitt als Studie III [So23b] und IV bereits kurz vorgestellt wurden. Dieser Abschnitt vertieft nun deren Beschreibung im Hinblick auf die Entwicklung des Aufgabenkatalogs.

Die Identifikation bestehender Aufgabentypen auf dem Gebiet der grafischen Modellierung im Kontext der logischen Aufgabenanalyse [So23b] berücksichtigt ver-

schiedene Anwendungsbereiche. Zunächst wurde die im Rahmen dieser Dissertation durchgeführte Analyse der Aufgabentypen aus dem Anwendungsbereich Geschäftsprozessmodellierung (siehe Abschnitt 3.1.2) aufgegriffen (siehe die in [So23b] beschriebene Vorstudie). Aus der wissenschaftlichen Literatur konnten weitere Aufgabentypen aus Publikationen z. B. aus dem Anwendungsbereich der objektorientierten Modellierung ([Br04; BSK09]) entnommen und ein vorläufiges Kategoriensystem von Aufgabentypen deduktiv abgeleitet werden. Zusätzlich wurde von den Verbundpartnern bereitgestelltes Aufgabenmaterial aus Veranstaltungen zu diversen Anwendungsbereichen (z. B. auch Unternehmensmodellierung, Datenmodellierung und Softwareentwicklung) mit Bezug zur grafischen Modellierung anhand einer Inhaltsanalyse nach MAYRING [Ma19] untersucht. Durch ein induktives Vorgehen konnte das vorläufige Kategoriensystem von Aufgabentypen angepasst und erweitert werden.

Um zu ermitteln, welche Kompetenzfacetten des Kompetenzmodells durch die verschiedenen Aufgabentypen adressiert werden, wurden drei Hochschullehrende aus dem KEA-Mod-Projekt mit Erfahrung in der Modellierungsausbildung (daunter auch die Verfasserin dieser Dissertation) getrennt voneinander gebeten, eine erste Zuordnung der Kompetenzfacetten zu den bestehenden Aufgabentypen vorzunehmen. Anschließend wurden die Ergebnisse miteinander verglichen und in einem diskursiven Ansatz ein Konsens über die Zuordnung hergestellt. Anhand der Zuordnung konnte aufgezeigt werden, welche Kompetenzfacetten des Kompetenzmodells bislang nur marginal oder durch keinen der bestehenden Aufgabentypen abgedeckt werden. Für die so identifizierten Lücken wurden Vorschläge für zusätzliche Aufgabentypen in den Aufgabenkatalog aufgenommen. Abb 4.3 stellt die identifizierten Aufgabentypen des Aufgabenkatalogs anhand ihrer Zuordnung zu den Kompetenzbereichen des Kompetenzmodells dar. Eine detaillierte Beschreibung des Aufgabenkatalogs und den enthaltenen 24 Aufgabentypen auf dem Stand von Studie III ist in [So23b] zu finden.

Weiterhin wurde im Rahmen der kognitiven Aufgabenanalyse eine Laut-Denk-Studie durchgeführt (Studie IV). An drei Standorten der Verbundpartner (DFKI, KIT, UDE) wurden insgesamt 15 Studierende gebeten, jeweils drei Modellierungsaufgaben zu UML-Klassendiagrammen und Petri-Netzen zu bearbeiten, die auf Basis von drei Aufgabentypen des Aufgabenkatalogs entwickelt wurden. Die Untersuchungen fanden im August 2021 in Präsenz statt. Dabei wurde neben einer Videoaufnahme der teilnehmenden Person anhand einer Dokumentenkamera auch die Bearbeitung auf dem Aufgabenblatt aufgezeichnet. Anschließend wurden die Aufnahmen durch die

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

INHALTSDIMENSION	PROZESSDIMENSION			
	Verstehen	Anwenden & Übertragen	Analysieren & Bewerten	Erschaffen
Modellverstehen & -interpretieren	Modellinhalt interpretieren	Problemlösen auf Basis eines Modells	Modelle prüfen	
	Aussagen in natürlicher Sprache wiedergeben (Freitext)	Inhalts- bzw. problembezogene Fragen beantworten	Fehler im Modell finden, Formale Eigenschaften prüfen	
Modellbilden & -modifizieren	Spezifische Fragen beantworten	Modellierungssprache anwenden	Modelle beurteilen	
		Beispiel-Modell erstellen, Modell vervollständigen, Syntax-Fehler korrigieren	Modelleignung prüfen, Modelle vergleichen	Modell anpassen
Werte / Haltungen / Überzeugungen			Peer-Feedback	
Metakognitive(s) Wissen & Fähigkeiten		Fallstudie	Modell bewerten	
Sozial-kommunikative(s) Wissen & Fähigkeiten		Diskussion	Berücksichtigung ethischer oder gesellschaftlicher Aspekte	
		Reflexion	Fallstudie	
			Peer-Feedback	
			Rollenspiele	
Komplexität und Kontextualität der Modellierungsaufgabe				

Abb. 4.3.: Aufgabentypen (abgerundete Textboxen) des Aufgabenkatalogs für die grafische Modellierung, verortet im Kompetenzmodell. Aufgabentypen mit gelber Hintergrundfarbe wurden nach Identifikation einer Lücke ergänzt. Eckige Textboxen mit weißem Hintergrund beinhalten Beispiele für den darüber befindlichen Aufgabentyp. In Anlehnung an [So23a; So23b]

UPB transkribiert und inhaltsanalytisch ausgewertet. Bei der Untersuchung der Laut-Denk-Protokolle (siehe Beispiel in Abb. 4.4) wurde geprüft, ob sich die Zuordnung der Kompetenzfacetten zum Aufgabentyp der jeweiligen Modellierungsaufgabe mit den Äußerungen und beobachtbaren Handlungen der teilnehmenden Personen deckt. Die Angemessenheit der Zuordnung von Kompetenzfacetten wurde dabei in gemeinsamen Besprechungen der Verbundpartner KIT, UDE und UPB diskutiert, was auch zur Ergänzung und Umformulierung einiger Kompetenzfacetten aus dem Kompetenzmodell führte. Insgesamt konnte die kognitive Aufgabenanalyse empirische Beweise für die kognitive Validität des Kompetenzmodells und des Aufgabenkatalogs liefern und die Angemessenheit der Zuordnung der Kompetenzfacetten zu den Aufgaben weitestgehend bestätigen. Es zeigt sich, dass durch die betrachteten Aufgabentypen relevante Kompetenzfacetten für die grafische Modellierung angesprochen werden.

..A2.2_MB3.04 Entwurfsentscheidung aufgr. plausibler Annahme		16	Und Start- und Enddatum hat man eben Daten. Also könnte auch ein <i>string</i> sein, aber wahrscheinlich gibt es irgendein spezielles Datumsformat. Ich schreibe einfach mal Datum hin und beim Enddatum genauso.
..A2.2_MB3.03 Typen von Modellelemente identifizieren (pb)			
..A1_MB2.04 Syntaktische Regeln anwenden (eb)	🔗	17	((notiert „+startdatum: Datum“ und „+enddatum: Datum“))
..A2.1_MB2.05 Bezeichner übernehmen (eb)			
..A2.1_MB3.05 Entwurfsentscheidungen erläutern (pb)		18	So (...) genau, ob es öffentlich ist oder vereinsintern. Also da steht, jetzt <i>kann</i> eingetragen werden. Das heißt, das ist optional. Ähm, da ist jetzt ein einzelner <i>boolean</i> wahrscheinlich nicht so gut, wenn es eben auch leer sein kann. Mhhh. (...) Ja, dann machen wir einfach zwei Attribute, einmal öffentlich und einmal vereinsintern. Dann kann man halt sagen, wenn beides <i>void</i> ist, ist halt nichts eingetragen. Wir haben dann den Nachteil, dass man eben beides oft umsetzen kann. Oder man könnte es natürlich auch mit einem - <i>ahn</i> - mit einer <i>enumeration</i> machen. Also das ist dann wahrscheinlich besser. Ach, dann machen wir das so.
..A2.2_MB3.03 Typen von Modellelemente identifizieren (pb)			
..A2.1_MB3.05 Entwurfsentscheidungen erläutern (pb)			
..A1_MB4.04 Bezeichner entwickeln (eb)		19	Ähm. (...) Ach, ich nenne das jetzt einfach mal. ((zeichnet den Rahmen einer weiteren Klasse)). Ach, wie nenne ich das (...) Äh Öffentlichkeit, oder? Ja. ((notiert Öffentlichkeit in die zuvor gezeichneten Klasse))

Abb. 4.4.: Auszug aus dem Laut-Denk-Protokoll zur Bearbeitung einer Aufgabe zur Erstellung eines UML-Klassendiagramms. Rechts Zuordnung zu entsprechenden Kompetenzfacetten, links transkribierte Aussagen. Screenshot aus der Software MAXQDA, zur Verfügung gestellt vom Verbundpartner UPB.

Die Laut-Denk-Methode ermöglichte es, auch Kompetenzfacetten zu untersuchen, die während des Lösungsprozesses relevant sind, da diese anhand der Protokolle evident wurden. Somit konnten auch diejenigen Kompetenzfacetten identifiziert und verifiziert werden, die sich nicht direkt anhand der eingereichten studentischen Lösung zu einer Aufgabe messen lassen. In der finalen Fassung des Aufgabenkatalogs (siehe [So22a]) werden bei der Zuordnung von Kompetenzfacetten zu Aufgabentypen die folgenden drei Typen festgelegt:

- **Ergebnisbezogene** Kompetenzfacetten, die sich beim betrachteten Aufgabentyp anhand der eingereichten studentischen Lösung (z. B. ein erstelltes UML-Klassendiagramm) bewerten lassen
- **Prozessbezogene** Kompetenzfacetten, die bei dem betrachteten Aufgabentyp nicht auf der Basis der eingereichten studentischen Lösung bewerten lassen, da sie nur im Verhalten (Denken und Handeln) deutlich werden³
- **Optionale** Kompetenzfacetten, die durch den betrachteten Aufgabentyp nur dann berücksichtigt werden, wenn dies durch eine bestimmte Variation der Aufgabenstellung durch den Aufgabenersteller gefordert wird⁴

Im Fokus dieser Arbeit steht der Aufgabentyp »Modell erstellen«. Definition 4.1 gibt die Beschreibung dieses Aufgabentyps im Aufgabenkatalog wieder.

³ Es ist möglich, prozessbezogene Kompetenzfacetten durch die Anpassung der Aufgabenstellung in ergebnisbezogene Kompetenzfacetten umzuwandeln. Dabei kann das prozessbezogene Denken oder Handeln beobachtbar (und somit messbar) gemacht werden, indem z.B. gefordert wird, eine Erläuterung dazu abzugeben oder den Lösungsweg aufzuzeichnen (siehe z. B. Variante 3 in Tab. 4.3). Auch ein Wechsel des Prüfungsmodus zu einer mündlichen Prüfung ist denkbar.

⁴ Optionale Kompetenzfacetten können ergebnis- oder prozessbezogen sein.

Definition 4.1: Aufgabentyp »Modell erstellen« im Aufgabenkatalog

»Bei diesem Aufgabentyp sind die Lernenden gefordert, selbstständig und von Grund auf ein grafisches Modell auf Basis einer gegebenen Problemstellung (i.d.R. eines in natürlicher Sprache beschriebenen Sachverhaltes) zu erstellen. Die zu verwendende Modellierungssprache ist hierbei vorgegeben.

Der Schwierigkeitsgrad dieses Aufgabentyps variiert insbesondere durch den unterschiedlichen Komplexitätsgrad des Kontexts, d. h. den Umfang des zu erstellenden Modells (im Sinne der Anzahl der zu erstellenden Modellelemente) sowie durch die Hinweise, die in der Aufgabenstellung gegeben sind (z. B. Hinweise auf zu berücksichtigende Modellelemente).

Die Aufgaben unterscheiden sich zudem in der Darstellungsform der Problemstellung. Hauptsächlich wird diese in einer reinen Prosa- Textform beschrieben. Mögliche Varianten sind z. B. Text in Interviewform, in Dokumentenform (Lieferschein, Fertigungsauftrag), als Datenbankstruktur oder Anforderungsspezifikationen in natürlicher Sprache. Der Kontext des beschriebenen Sachverhalts kann fiktiv, aus dem Alltag der Lernenden oder aus einer praxisrelevanten Anwendungsdomäne sein.« (Quelle: [So22a, S. 10]).

Diese Definition ist umfassender als die vorangegangene Beschreibung im Abschnitt 3.1.2 und berücksichtigt den Schwierigkeitsgrad, der sich über Komplexität und Kontextualität steuern lässt, sowie verschiedene mögliche Darstellungsformen der Problemstellung.

Tab. 4.2 listet die adressierten Kompetenzfacetten für den Aufgabentyp »Modell erstellen« auf. Weiterhin werden im Aufgabenkatalog 11 Varianten für diesen Aufgabentyp beschrieben, über welche zusätzliche optionale ergebnis- und prozessbezogene Kompetenzfacetten mit berücksichtigt werden können. Diese lassen sich teilweise miteinander kombinieren, so dass es möglich ist, mit einer einzelnen Aufgabe zugleich viele verschiedene Kompetenzfacetten zu adressieren. Einige der identifizierten Varianten werden in Tab. 4.3 dargestellt. Für eine vollständige Aufführung sei auf [So22a] verwiesen.

Tab. 4.2.: Mit dem Aufgabentyp »Modell erstellen« adressierte Kompetenzfacetten aus dem Aufgabenkatalog [So22a, S. 10]

Ergebnisbezogene Kompetenzfacetten	
MB 4.01	Die Lernenden können selbstständig grafische Modelle (wie z. B. UML-Diagramme, ER-Modelle und Petri-Netze) erstellen, um einen Sachverhalt abzubilden.
MB 4.04	Die Lernenden können ein Modell in Bezug auf einen Sachverhalt a) semantisch korrekt und b) vollständig erstellen und sich dabei c) auf relevante Modellinhalte beschränken (Prägnanz).
MB 4.05	Die Lernenden können ein Modell a) auf Basis bekannter Richtlinien oder Konventionen gut lesbar und strukturiert sowie b) für die jeweilige Zielgruppe verständlich erstellen.
MB 2.04	Die Lernenden können a) die syntaktischen Regeln und b) die Notation der betrachteten Modellierungssprache/n korrekt anwenden.
MB 2.05	Die Lernenden können Bezeichner für Modellelemente a) aus einer Problemstellung übernehmen, b) bei Bedarf konventionskonform anpassen und c) einheitlich verwenden.
Prozessbezogene Kompetenzfacetten	
MB 3.01	Die Lernenden können relevante Informationen, Strukturen und Zusammenhänge zielgerichtet aus einer Problemstellung ermitteln und somit die Anforderungen analysieren.
MB 3.03	Die Lernenden können Typen von Modellelementen (oder Modellierungsmuster) in einer Problemstellung ermitteln bzw. passende Typen von Modellelementen zur Darstellung von spezifischen Aspekten eines Sachverhaltes auswählen.
MB 3.04	Die Lernenden können Entwurfsentscheidungen treffen, indem sie (z. B. auf Basis von Intuition, logischem Denken, Domänenwissen) plausible Annahmen machen.
MB 3.05	Die Lernenden können ihre Entwurfsentscheidungen für ein selbst erstelltes Modell beurteilen und begründen.
MK 3.01	Die Lernenden sind bei der Bearbeitung von Modellierungsaufgaben in der Lage, a) ihr Vorgehen zu planen, b) geeignete Strategien auszuwählen sowie c) ihren Fortschritt, ihr Verständnis und ihre Problemlösung zu überwachen.

4.3. Rubrics und Feedback

In diesem Abschnitt werden zunächst die als Ergänzung für den Aufgabenkatalog entwickelten Bewertungsschemata (Rubrics) zur Unterstützung der kriteriengeleiteten bzw. kompetenzorientierten Bewertung von Modellierungsaufgaben vorgestellt. Dabei kommen aufgabenübergreifende Rubrics (siehe Abschnitt 2.1.4) zum Einsatz, die sich für Aufgaben des jeweiligen Typs anwenden lassen. Wie bereits in Abschnitt 2.1.4 dargelegt, können Rubrics neben einer höheren Objektivität und Transparenz der Bewertung auch dazu beitragen, lernförderliches Feedback zu generieren. Studierende können anhand dieser erkennen, welche Kriterien (und zugrundeliegende Kompetenzen) grundsätzlich erforderlich sind, was sie bereits beherrschen und wo noch Steigerungsbedarf in Bezug auf die eigene Leistung besteht.

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

Tab. 4.3.: Optionale Kompetenzfacetten für den Aufgabentyp »Modell erstellen« aus dem Aufgabenkatalog [So22a, S. 10ff].

Variantenbeschreibung	Kompetenzfacette (ergebnisbezogen)
1: Bei dieser Aufgabenvariante wird den Lernenden nicht vorgegeben, welche Modellierungssprache verwendet werden muss.	MB 3.06 Die Lernenden können Modellierungssprachen bzw. Modelltypen auf ihre Eignung für eine konkrete Anwendungsdomäne und den jeweiligen Modellierungszweck prüfen, beurteilen und auswählen.
2: Die Problemstellung erfordert, dass die Lernenden einen angemessenen Abstraktionsgrad für ihr Modell wählen und diesen beibehalten. D.h. sie müssen z. B. entscheiden, wie detailliert sie bestimmte Aspekte darstellen oder ob sie ggf. Aspekte zusammenfassen und somit ggf. verkürzt darstellen.	MB 4.07 Die Lernenden können a) einen in Bezug auf den Modellierungszweck angemessenen Abstraktionsgrad bei der Erstellung eines Modells wählen und b) diesen innerhalb des Modells konsistent beibehalten.
3: Die Lernenden werden aufgefordert, ihre Entwurfsentscheidung zu begründen (d.h. z.B. zu verschriftlichen). Hierzu gehört auch zu erläutern, warum welches Modellierungskonzept oder welche Typen von Modellelementen ausgewählt wurden sowie welche Annahmen jeweils zugrunde liegen.	MB 3.05 Die Lernenden können ihre Entwurfsentscheidungen für ein selbst erstelltes Modell beurteilen und begründen.
	MB 3.03 Die Lernenden können Typen von Modellelementen (oder Modellierungsmuster) in einer Problemstellung ermitteln bzw. passende Typen von Modellelementen zur Darstellung von spezifischen Aspekten eines Sachverhaltes auswählen.
	MB 3.04 Die Lernenden können Entwurfsentscheidungen treffen, indem sie (z. B. auf Basis von Intuition, logischem Denken, Domänenwissen) plausible Annahmen machen.
	MB 3.02 Die Lernenden können Modellierungskonzepte in einer Problemstellung ermitteln bzw. passende Modellierungskonzepte zur Darstellung von spezifischen Aspekten eines Sachverhaltes auswählen.
4: In dieser Aufgabenvariante werden ein Szenario und spezifische Richtlinien oder Konventionen (z.B. fiktive, firmeninterne Regelungen) vorgegeben bzw. genannt, die bei der Modellierung berücksichtigt werden müssen.	MB 4.05 Die Lernenden können ein Modell a) auf Basis bekannter Richtlinien oder Konventionen gut lesbar und strukturiert sowie b) für die jeweilige Zielgruppe verständlich erstellen.
5: Die Lernenden werden aufgefordert, für das Modellerstellen ein bestimmtes Modellierungswerkzeug zu verwenden, anstatt die Lösung mit Stift und Papier zu erstellen.	MB 2.01 Die Lernenden können Modellierungswerkzeuge praktisch einsetzen oder anwenden.

Aus dem Aufgabenkatalog wurden Aufgabentypen jeder Prozesstufe aus den beiden fachbezogenen Inhaltsbereichen des Kompetenzmodells ausgewählt und entsprechende Rubrics erstellt. Die aufgeführten Kriterien wurden dabei zunächst anhand der für den jeweiligen Aufgabentyp dem Aufgabenkatalog nach relevanten ergebnisbezogenen Kompetenzfacetten abgeleitet. Der Entwurf wurde anschließend zusammen mit drei Hochschullehrenden aus dem KEA-Mod-Projekt mit Erfahrung in der Modellierungslehre (darunter die Verfasserin dieser Dissertation) diskutiert und angepasst. Danach wurden die Rubrics von zwei der Hochschullehrenden praktisch eingesetzt und auf den eingereichten Lösungen für die Modellierungsaufgaben aus der Laut-Denk-Studie (siehe Abschnitt 4.2) angewendet. Im Anschluss wurden weitere Anpassungen an den Rubrics vorgenommen. Als Ergebnis wurden Vorschläge für Rubrics für sieben Aufgabentypen auf dem Gebiet der grafischen Modellierung zusammen mit Empfehlungen für die kompetenzorientierte Bewertung und Feedback erschaffen und auf der Projekt-Homepage veröffentlicht [So22d]. Tab. 4.4 stellt das Rubric für den Aufgabentyp »Modell erstellen« vor.

Die veröffentlichten Rubrics sollen ausdrücklich als Vorschlag dienen und lassen sich auf die individuelle Aufgabenstellung und die Anforderungen des Aufgabenerstellers anpassen. Beispielsweise kann die Dimension des Leistungsniveaus individuell gestaltet werden, so dass zwischen mehr (oder weniger) Stufen als den vorgeschlagenen drei Stufen unterschieden wird. Zudem ist es ratsam, die eingesetzten Kriterien entsprechend der Zielsetzung und des Lernstands der Studierenden zu gewichten. So können Kriterien mit hoher Bedeutung mit mehr Punkten versehen werden, so dass der Anteil dieser an der Gesamtpunktzahl steigt. Ebenso ist es möglich, für die jeweilige Aufgabenstellung nicht relevante Kriterien aus den Rubrics zu entfernen oder aber sie lediglich mit 0 Punkten zu gewichten. Somit fließen sie zwar nicht in die Bewertung über die Gesamtpunktzahl mit ein, Studierende erhalten dennoch das entsprechende Feedback. Ebenso kann beim Kriterium Syntax bei Bedarf eine volle Punktzahl nur dann vergeben werden, wenn das studentische Modell einen ausreichenden Umfang besitzt. Dadurch soll berücksichtigt werden, dass mit sehr kleinen und unvollständigen Modellen die vorhandenen Kompetenzen zur Syntax einer Modellierungssprache nicht ausreichend demonstriert werden können.

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

Tab. 4.4.: Rubric für die Bewertung von Aufgaben des Typs »Modell erstellen«. Entnommen aus [So22d]

Kriterium und Beschreibung	Gute bis sehr gute Kriterienerfüllung	Befriedigende Kriterienerfüllung	Unzureichende Kriterienerfüllung
<p>Korrekte Verwendung der Modellierungssprache (Syntax): Korrekte Umsetzung der Syntax (Regeln zur Verwendung und Verknüpfung von Elementen), Korrekte Verwendung der Modellelemente (Knoten und Kanten), Korrekte Umsetzung der Notation (korrekte grafische Symbole der Elemente)</p>	<p>Alle Modellelemente sind Bestandteil der Modellierungssprache und werden sinn- und notationsgemäß verwendet. Das Modell verletzt keine syntaktischen Regeln der Modellierungssprache. Volle Punktzahl nur, wenn das Modell $\geq 25\%$ der Aussagen abbildet! (Abhängig vom Kriterium Vollständigkeit)</p>	<p>Die verwendeten Modellelemente sind Bestandteil der Modellierungssprache und werden weitgehend sinn- und notationsgemäß verwendet. Einzelne Modellelemente (Notation) werden konsequent falsch verwendet bzw. vertauscht. Das Modell verletzt weitgehend keine syntaktischen Regeln der Modellierungssprache.</p>	<p>Einige Modellelemente sind nicht Bestandteil der Modellierungssprache. Einige Modellelemente werden nicht sinn- und notationsgemäß verwendet. Das Modell verletzt in großen Teilen syntaktische Regeln der Modellierungssprache.</p>
<p>Validität der Aussagen (Semantik): Inhaltlich und sinngemäß korrekte Modellierung des Sachverhaltes. Plausibilität von Annahmen: Sofern es die Aufgabe erfordert (uneindeutige Sachverhaltsbeschreibung), werden plausible Annahmen auf Basis von Domänenwissen, logischem Denken oder »gesundem Menschenverstand« getroffen.</p>	<p>Das Modell bildet den beschriebenen Sachverhalt inhaltlich korrekt ab. Sofern notwendig, werden Annahmen zu im Sachverhalt nicht eindeutig definierten Aspekten getroffen und diese im Modell plausibel / sinnvoll ergänzt.</p>	<p>Das Modell bildet den beschriebenen Sachverhalt inhaltlich weitgehend korrekt ab. Es werden einzelne unplausible Annahmen zu weniger relevanten Aspekten des Sachverhalts getroffen und im Modell integriert.</p>	<p>Das Modell bildet den beschriebenen Sachverhalt inhaltlich nicht korrekt ab. Es werden unnötige bzw. unplausible Annahmen getroffen und im Modell integriert.</p>
<p>Vollständigkeit (Semantik): Anteil der für die Modellierung relevanten Aussagen der Sachverhaltsbeschreibung, die im Modell dargestellt werden (Anteil fehlender Modellelemente)</p>	<p>Das Modell ist vollständig, d.h. es enthält alle in der Sachverhaltsbeschreibung beschriebenen und für den Modellierungszweck relevanten Aspekte</p>	<p>In dem Modell fehlen einzelne in der Sachverhaltsbeschreibung beschriebenen und für den Modellierungszweck relevanten Aspekte</p>	<p>Größere »Modelleinheiten« fehlen, d.h. werden nicht entsprechend der Sachverhaltsbeschreibung im Modell abgebildet</p>

wird fortgesetzt auf der nachfolgenden Seite

Tab. 4.4 – fortgesetzt von der vorherigen Seite

Kriterium und Beschreibung	Gute bis sehr gute Kriterienerfüllung	Befriedigende Kriterienerfüllung	Unzureichende Kriterienerfüllung
Prägnanz (Semantik): Anteil der für die Modellierung irrelevanten Aussagen der Sachverhaltsbeschreibung, die im Modell dargestellt werden (Anteil überflüssiger Modellelemente)	Das Modell enthält keine für das Modellierungsziel irrelevanten oder überflüssigen Aspekte	Das Modell enthält einzelne, kleinere irrelevante Aspekte	Das Modell enthält einige für das Modellierungsziel irrelevanten oder uneindeutige Aspekte
Einheitlicher Abstraktionsgrad: Wahl eines passenden Abstraktionsniveaus (Detailgrad, verkürzte Darstellung von Aspekten); idR. nur bei komplexeren Modellen relevant	Es wird ein angemessenes Abstraktionsniveau gewählt, welches konsequent beibehalten wird	Es wird ein zu niedrigeres / hohes Abstraktionsniveau gewählt, welches jedoch konsequent beibehalten wird ODER es wird größtenteils ein passendes Abstraktionsniveau gewählt, welches vereinzelt nicht konsequent beibehalten wird	Es wird ein unangemessenes Abstraktionsniveau gewählt, welches zudem nicht konsequent beibehalten wird
Bezeichner (Pragmatik): Bezeichnungsstil (ggf. Umsetzung auf Basis bestimmter Richtlinien), Konsistenz der Bezeichner, Verständlichkeit der Bezeichner (sinnvolle Bezeichnungen)	Bezeichner werden auf Basis des Sachverhaltes plausibel entwickelt bzw. wenn möglich übernommen und konsistent verwendet (dazu zählt auch eine konsistente Wortwahl / Sprache)	Bezeichner werden passend entwickelt, jedoch nicht konsistent (z.B. auch in ihrer Wortwahl / Sprache) verwendet; Bezeichner werden teilweise unnötigerweise neu entwickelt, obwohl sie übernommen werden könnten; Bezeichner sind z.T. nicht konventionskonform	Es werden größtenteils unangemessene, unplausible Bezeichner gewählt, die teilweise uneinheitlich verwendet werden
Element-Anordnung (Pragmatik): Gesamt-Layout des Modells, Anordnung der Elemente im Modell	Die Modellelemente werden gut lesbar, übersichtlich und kreuzungsfrei angeordnet	Die Modellelemente werden in weiten Teilen gut lesbar, übersichtlich und kreuzungsfrei angeordnet	Die Modellelemente werden schwer lesbar, unstrukturiert und mit vielen Überschneidungen angeordnet

Weiterhin wurden Vorschläge für exemplarische Feedback-Nachrichten erstellt, die sich an den Empfehlungen von HATTIE und TIMPERLEY [HT07] (siehe Abschnitt 2.1.4) orientieren. Im Sinne eines Feed-Up (»Wohin gehe ich?«) können die durch eine Aufgabe adressierten ergebnis- und prozessbezogenen Kompetenzfacetten aufgelistet werden, z. B. wie in Abb. 4.5 dargestellt. Alternativ können auch die verwendeten

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

Mithilfe dieser Aufgabe sollen Sie lernen, ...

- ... selbstständig grafische Modelle zu erstellen, um einen Sachverhalt abzubilden.
- ... ein Modell in Bezug auf einen Sachverhalt a) semantisch korrekt und b) vollständig zu erstellen und sich dabei c) auf relevante Modellinhalte zu beschränken (Prägnanz).
- ... ein Modell (a) auf Basis bekannter Richtlinien oder Konventionen gut lesbar und strukturiert sowie (b) für die jeweilige Zielgruppe verständlich zu erstellen.
- ... wenn möglich Bezeichner für Modellelemente a) aus einer Problemstellung zu übernehmen, b) bei Bedarf konventionskonform anpassen und c) einheitlich zu verwenden.
- ... wenn erforderlich selbstständig Bezeichner a) angemessen / verständlich und b) konventionskonform zu entwickeln sowie c) einheitlich zu verwenden.
- ... a) einen in Bezug auf den Modellierungszweck angemessenen Abstraktionsgrad bei der Erstellung eines Modells zu wählen und b) diesen innerhalb des Modells konsistent beizubehalten.
- ... a) die syntaktischen Regeln und b) die Notation der betrachteten Modellierungssprache/n korrekt anzuwenden.

Außerdem sollen Sie lernen, ...

- ... relevante Informationen sowie Strukturen und Zusammenhänge zielgerichtet aus einer Problemstellung zu ermitteln und somit die Anforderungen zu analysieren.
- ... Modellierungskonzepte und Typen von Modellelementen (oder Modellierungsmuster) in einer Problemstellung zu ermitteln bzw. passende Modellierungskonzepte zur Darstellung spezifischer Aspekte eines Sachverhaltes auszuwählen.
- ... Entwurfsentscheidungen zu treffen, indem Sie (z. B. auf Basis von Intuition, logischem Denken, Domänenwissen) plausible Annahmen machen.
- ... bei der Bearbeitung von Modellierungsaufgaben a) Ihr Vorgehen zu planen, b) geeignete Strategien auszuwählen sowie c) Ihren Fortschritt, Ihr Verständnis und Ihre Problemlösung zu überwachen.

Abb. 4.5.: Beispiele für Feed-Up-Nachrichten, entnommen aus [So22e, S. 42]

Rubrics bereitgestellt werden, um die Bewertungskriterien transparent zu machen.

Im Sinne eines Feed-Back («Wo stehe ich gerade?») sollten die identifizierten Fehler oder positive Rückmeldungen in Bezug zur entsprechenden Kompetenzfacette bzw. zum jeweiligen Beurteilungskriterium gesetzt werden. Eine simple und naheliegende Option ist die Nutzung von Rubrics, um die erreichte Stufe der Kriterienerfüllung in der Matrix zu markieren. Zusätzlich können für jedes Kriterium jeweils negative (Fehler) und positive Rückmeldungen aufgelistet werden. Den Lernenden wird dabei eine Orientierungshilfe geboten, wenn Fehler direkt im Modell visuell markiert werden. Dies entspricht der Feedback-Variante »grafisches Feedback«, welche in der systematischen Literaturrecherche zu E-Assessment-Systemen für die grafische Modellierung (siehe Abschnitt 3.3.2) identifiziert wurde. Weiterhin können sich Feedback-Nachrichten auf die verschiedenen Ebenen nach HATTIE und TIMPERLEY

Wo stehen Sie?**Feedback auf Aufgabenebene:**

- Semantik/Vollständigkeit: »Das Modell enthält keine Klasse „Kunde“ mit passenden Attributen.« Alternativ: »Das Modell bildet den Sachverhalt nicht vollständig ab. Achten Sie darauf, dass Sie alle relevanten Aspekte des Sachverhalts im Modell darstellen.«
- Syntax/Korrekte Verwendung der Modellierungssprache: »Gut! Das Modell besitzt keine Modellierungselemente, die nicht zu Petri-Netzen gehören.«
- Pragmatik/Element-Anordnung: »Achtung: Das Modell weist Kantenüberschneidungen auf. Diese lassen sich oft vermeiden.«
- Pragmatik/Bezeichner: »Achten Sie auf die Verwendung von angemessenen Bezeichnern. Bezeichner sollten, wenn möglich, übernommen werden oder konventionskonform angepasst bzw. entwickelt werden.«

Feedback auf Prozessebene:

- »Denken Sie darüber nach, welche Qualitätskriterien Sie für grafische Modelle kennen und überlegen Sie sich, wie Sie ihr Modell diesbezüglich verbessern können.«
- »Versuchen Sie, die Problemstellung mithilfe von Organisations- und Verständnisstrategien (z. B. farbiges Markieren) erneut zu lesen und durcharbeiten, um Strukturen und Zusammenhänge zu erkennen und die für die Modellbildung tatsächlich relevanten Informationen herauszufiltern.«
- »Versuchen Sie, Ihr Modell unter Berücksichtigung des Feedbacks weiter zu verbessern, indem Sie weitere passende Modellierungskonzepte oder Typen von Modellelementen nutzen (z.B. Einfügen von Multiplizitäten zur Präzisierung der Beziehungen).«

Selbstregulationsebene:

- »Im zeitlichen Verlauf der Fallstudienarbeit haben Sie erst spät mit der Erstellung des Modells begonnen. Versuchen Sie zukünftig, anstehende Arbeitspakete zeitlich sinnvoll einzuplanen und frühzeitig mit der Erstellung eines ersten Modellentwurfs zu beginnen.«

Abb. 4.6.: Beispiele für Feedback-Nachrichten, entnommen aus [So22e, S. 42]

[HT07] beziehen. Dies wird durch die exemplarischen Feedback-Nachrichten in Abb. 4.6 verdeutlicht. Für ein Feed-Forward (»Wohin gehe ich als nächstes?«) werden in Abb. 4.7 einige exemplarische Feedback-Nachrichten vorgestellt.

4.4. Diskussion

Mit diesem Abschnitt soll nun ein Bogen zur eingangs formulierten Forschungsfrage FF1 »Wie kann das Prüfen auf dem Gebiet der grafischen Modellierung kompetenzorientiert gestaltet werden?« (siehe Abschnitt 1.2) geschlagen und die in diesem Kapitel präsentierten Ergebnisse kritisch reflektiert werden. Die im Kontext der Forschungsfrage angestrebten Teilziele wurden mit der Entwicklung dreier Artefakte für das Gebiet der grafischen Modellierung, (i) des Kompetenzmodells, (ii) des Auf-

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

Wie erreichen Sie das Ziel?

- »Diese Fehler deuten darauf hin, dass Sie die syntaktischen Regeln von UML Klassendiagrammen wiederholen sollten. Wir empfehlen hierfür das Kapitel 2 im Lehrbuch.«
- »Wir empfehlen, als Nächstes Aufgabenreihe 123456 in KEA-Mod zu lösen, um Ihre Fähigkeiten in anderen Anwendungskontexten zu festigen.«
- »Sie sind bereit, sich neuen Herausforderungen zu stellen. Wir empfehlen Ihnen, mit Aufgabenreihe 456123 in KEA-Mod fortzufahren.«
- »Um mehr Übung im Bereich der grafischen Modellierung zu bekommen, empfehlen wir Ihnen, weitere Aufgaben aus der Aufgabenreihe 424242 zu bearbeiten.«
- »Den Aufgabentyp ›Modell erstellen‹ beherrschen Sie sehr gut. Wir empfehlen, als Nächstes Aufgabenreihe 612345 in KEA-Mod zu lösen, um Ihre Fähigkeiten im Bereich ›Modellverstehen‹ zu trainieren.«

Abb. 4.7.: Beispiele für Feed-Forward-Nachrichten, entnommen aus [So22e, S. 42]

gabenkatalogs und (iii) der Rubrics erreicht. Diese sollen im Folgenden sukzessive beleuchtet und dabei Grenzen sowie Möglichkeiten für die Anschlussforschung aufgezeigt werden.

4.4.1. Kompetenzmodell

Das Kompetenzmodell soll Lehrende in der Hochschullehre dabei unterstützen, ihre Lehrveranstaltungen und Prüfungen auf dem Gebiet der grafischen Modellierung kompetenzorientiert (d.h. unter Berücksichtigung der beabsichtigten Lernziele) zu gestalten. Während bestehende Kompetenzmodelle oder Lernzieltaxonomien (siehe Abschnitt 3.1.1 entweder sehr knapp und abstrakt [Ge16a; Ge16b] ausfallen oder auf einen speziellen Anwendungsbereich der Modellierung zugeschnitten sind [Bo19; BS19], war es im KEA-Mod-Projekt durch die interdisziplinäre Zusammenarbeit mit einem Verbundpartner aus der Hochschuldidaktik möglich, eine umfassende Kompetenzmodellierung für die grafische Modellierung im Allgemeinen durchzuführen. Dabei wurden auch überfachliche Kompetenzen berücksichtigt. Auf Basis fundierter Empfehlungen [Sc09] wurden ein theoriegestützter und ein empirischer Ansatz zur Kompetenzmodellierung kombiniert. Anschließend wurde das Kompetenzmodell in drei weiteren Studien (siehe Abschnitte 4.1 und 4.2) weiterentwickelt und parallel Belege für die Validität gesammelt, indem die inhaltliche Angemessenheit, Relevanz und Repräsentativität der formulierten Kompetenzfacetten anhand des Experten-Ratings und der Laut-Denk-Studie empirisch überprüft wurden ([So22f; So23b]).

Während der Entwicklung des Kompetenzmodells wurden aus praktischen Gründen (Verfügbarkeit und Handhabbarkeit des Umfangs) einige Einschränkungen in Kauf genommen. So wurden die Modul- bzw. Kursbeschreibungen und das Aufgabenmaterial zur grafischen Modellierung lediglich aus Veranstaltungen von drei Verbundpartnern (an deutschen Universitäten) entnommen. Diese Unterlagen konnten im Rahmen des KEA-Mod-Projekts in angemessener Zeit ausgewertet werden. Zudem konnten beim Expertenrating in Studie II über die Verbreitungskanäle des Verbundpartners GI nur Fachexperten aus dem deutschsprachigen Raum rekrutiert werden. Diese Einschränkungen in Bezug auf den Umfang des untersuchten Materials und eine Fokussierung auf den deutschsprachigen Raum könnten in anschließenden Forschungsarbeiten gezielt adressiert werden. Denkbar wäre, das Kompetenzmodell in der finalen Version 3 erneut anhand eines Expertenratings von internationalen Fachexperten bewerten zu lassen. Zudem kann das Kompetenzmodell durch die praktische Anwendung in der Lehre an internationalen Hochschulen weiter validiert werden. Dabei könnten nationale Unterschiede sichtbar gemacht werden.

Bei den im Kompetenzmodell berücksichtigten Kompetenzen wurde ein Fokus auf handwerkliche Kenntnisse und Fähigkeiten zum grafischen Modellieren gelegt. Es wäre möglich, in weiteren Untersuchungen mentale und kognitive Prozesse während der Bearbeitung von Modellierungsaufgaben vertieft in den Blick zu rücken. Dabei könnte auch die im Rahmen des FRISCO-Frameworks [Fa98] entwickelte Unterscheidung zwischen Modell und Modelldenotation (siehe Abschnitt 2.2.2) berücksichtigt werden. Weiterhin gilt es, Gemeinsamkeiten und Unterschiede zwischen dem grafischen Modellieren mit einer Modellierungssprache (Modellinterpretation und Modellbildung) und dem Anwenden einer natürlichen Sprache (Textverständnis und Textproduktion) in Bezug auf die beteiligten Kompetenzen genauer zu analysieren. Die Untersuchung der Sprachkompetenz (z. B. [Os09; Sc16]) mit den Bereichen Lese- und Schreibkompetenz als Teilbereich der Sprachdidaktik kann auf eine längere Forschungshistorie zurückblicken als die Informatikdidaktik, sodass möglicherweise vorhandene Erkenntnisse auf das Gebiet der grafischen Modellierung übertragen werden können.

Das Kompetenzmodell wurde als generisches Kompetenzmodell für die grafische Modellierung in der Hochschullehre entworfen. Eine zusätzliche Verfeinerung des Kompetenzmodells im Hinblick auf die verschiedenen Anwendungsbereiche (z. B. Softwareentwicklung, Datenmodellierung, Geschäftsprozessmodellierung) könnte für Lehrende in der Anwendung eine zusätzliche Unterstützung bieten. Dabei könnten spezifische Kompetenzfacetten eines Anwendungsbereichs ergänzt, die Re-

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

relevanz der Kompetenzfacetten im Hinblick auf den Anwendungsbereich individuell bewertet und die allgemein gehaltene Formulierung der Kompetenzfacetten auf Spezialbegriffe des Anwendungsbereichs übertragen werden.

Durch die Gestaltung des Kompetenzmodells anhand einer Lernzieltaxonomie ist ein sogenanntes Kompetenzstrukturmodell (siehe [Sc09]) entstanden. Dieses könnte zukünftig durch ein Kompetenzniveaumodell ergänzt werden, welches Kompetenzfacetten auf verschiedenen Anforderungsniveaus beschreibt und im Hinblick auf den Schwierigkeitsgrad ausdifferenziert. Beispielsweise wäre eine Differenzierung zwischen Bachelor- und Masterebene möglich. Erreicht werden könnte dies durch weitere Laut-Denk-Studien, in denen die Laut-Denk-Protokolle von Studierenden mit besseren und schlechteren Leistungen verglichen werden. Dabei könnten schwierighkeitsbestimmende Faktoren identifiziert und erfolgskritische Kompetenzen weiter herausgearbeitet werden.

4.4.2. Aufgabenkatalog

Mit dem Aufgabenkatalog erhalten Lehrende nicht nur eine Beschreibung einer Bandbreite verschiedener Aufgabentypen auf dem Gebiet der grafischen Modellierung, sondern diese Aufgabentypen wurden mit Kompetenzfacetten aus dem Kompetenzmodell verknüpft, um Modellierungsaufgaben gezielt entsprechend der intendierten Lernziele in Übungen und Prüfungen einsetzen zu können. Aus diesem Grund bietet der hier vorgestellte Aufgabenkatalog einen deutlichen Mehrwert gegenüber bestehenden verwandten Aufgabenklassifikationen (z. B. [Br04; BSK09]). Letztere konzentrieren sich zudem auf einen spezifischen Anwendungsbereich (objektorientierte Modellierung), wohingegen der hier vorgestellte Aufgabenkatalog das übergeordnete Gebiet der grafischen Modellierung berücksichtigt.

Dabei wurden im Rahmen der Inhaltsanalyse nach MAYRING [Ma19] zunächst bestehende Aufgabentypen identifiziert. Die durch die Zuordnung von Kompetenzfacetten zu den bestehenden Aufgabentypen aufgezeigte Lücke wurde anschließend durch eine Ergänzung um weitere Aufgabentypen geschlossen. Somit können mit den im Aufgabenkatalog enthaltenen Aufgabentypen alle Kompetenzfacetten des Kompetenzmodells adressiert werden, d.h. der Aufgabenkatalog bietet in Bezug auf die Kompetenzfacetten des Kompetenzmodells eine vollständige Abdeckung. Allerdings schließt das eine mögliche Ergänzung des Aufgabenkatalogs um weitere oder spezifischere Aufgabentypen nicht aus. Insbesondere könnte detailliertes Beispielmateriale oder Best Practices für komplexere Aufgabentypen wie Fallstudie,

Rollenspiel oder Peer-Review gesammelt und der Allgemeinheit zugänglich gemacht werden⁵.

Auf Basis der Klassifikation von Aufgabentypen im Aufgabenkatalog sind darüber hinaus weitere Untersuchungen denkbar. Es wäre besonders aufschlussreich, zu erkennen, in welchem Umfang und in welchem didaktischen Kontext (Übung, Prüfung) oder in welchem Anwendungsbereich diese Aufgabentypen tatsächlich in der Lehre zum Einsatz kommen. Diese Ergebnisse könnten in Form entsprechender Empfehlungen direkt in den Aufgabenkatalog integriert werden. Auch kann die dadurch mögliche Identifikation bislang vernachlässigter Kompetenzfacetten einen Anstoß für die Reflexion der eigenen Lehre und für die Diskussion innerhalb der Fachcommunity bieten. So wurde beispielsweise durch die systematische Literaturrecherche (siehe Abschnitt 3.3.2) bereits angedeutet, dass E-Assessment-Systeme für die automatisierte Bewertung grafischer Modelle nur selten pragmatische Qualitätsaspekte berücksichtigen, obwohl die Lesbarkeit und zielgruppengerechte Darstellung von Modellen insbesondere beim Einsatzzweck als Kommunikationsinstrument von großer Bedeutung ist. Es gilt nun empirisch zu untersuchen, ob pragmatische Qualitätsaspekte auch im Kontext nicht-digitaler Lehre und Prüfungen eine untergeordnete Rolle spielen.

4.4.3. Rubrics und Feedback

Die auf Basis des Aufgabenkatalogs entwickelten Rubrics sollen Lehrenden einen Vorschlag für die kriteriengeleitete und kompetenzorientierte Bewertung auf dem Gebiet der grafischen Modellierung liefern. Die in den Rubrics angegebenen Kriterien besitzen einen direkten Bezug zu den ergebnisbezogenen Kompetenzfacetten, die für den jeweiligen Aufgabentyp als relevant definiert wurden. Durch den im Rahmen des Entwicklungsprozesses durchgeführten praktischen Einsatz der Rubrics durch Lehrende konnte deren Anwendbarkeit demonstriert werden. Allerdings wurde darauf hingewiesen, dass bei der Bewertung vom Aufgabentyp »Modell verstehen« Fehler nicht eindeutig zu nur einem Kriterium bezüglich der Qualitätsaspekte Syntax und Semantik zugeordnet werden können. Diese wechselseitige Abhängigkeit gilt es bei der Bewertung zu berücksichtigen. Sie wurde bereits von MOODY et al. [Mo02]

⁵ In diesem Zusammenhang sei ein Hinweis auf die Initiative zur Sammlung und Verbreitung herausragender Modellierungsaufgaben gegeben, welche von der Verfasserin dieser Dissertation 2020 mitgegründet wurde: <http://butler.aifb.kit.edu/initiative/>. In Anlehnung an die sogenannten »nifty assignments« der Special Interest Group on Computer Science Education (SIGCSE) ist es das Ziel der Initiative, Lehrenden besonders reizvolles Lehr- und Lernmaterial oder spezielle Aufgabenformate auf dem Gebiet der grafischen Modellierung zur Verfügung zu stellen.

4. Kompetenzorientiertes Prüfen im Gebiet der grafischen Modellierung

(siehe Abschnitt 3.2.1) identifiziert und ebenso bei der Betrachtung berücksichtigter Aspekte während der Literaturstudie evident (siehe SLR-FF1, Abschnitt 3.3.2).

Dennoch bieten die Rubrics einen Fortschritt gegenüber herkömmlichen Bewertungsschemata in Form eines punktebasierten Abzugsverfahrens (z.B. Punktabzug je Syntax-Fehler), da sie a) weitere relevante Kriterien wie Pragmatik oder Abstraktionsgrad betonen und b) bei entsprechender transparenter Gestaltung des Feedbacks nicht nur Defizite kommuniziert werden, sondern den Lernenden auch deutlich gemacht wird, welche Kompetenzen oder Lernziele bereits erreicht wurden.

Ein weiterer Ansatz wäre es, Rubrics für zusätzliche Aufgabentypen aus dem Aufgabenkatalog zu entwickeln. Eine besondere Herausforderung stellt hierbei die Prüfung einiger prozessbezogener Kompetenzfacetten dar, beispielsweise SK 2.03 »Die Lernenden können im Rahmen der Modellbildung Ideen anderer Teammitglieder annehmen und aufgreifen.«. Diese Kompetenzfacette wird durch den Aufgabentyp Fallstudie adressiert. Eine Überprüfung dieser Kompetenzfacette würde allerdings eine Untersuchung des Prozesses der Teamarbeit durch Prüfende erfordern (direkte Beobachtung oder Aufzeichnung der schriftlichen oder mündlichen Kommunikation), so dass hier praktische Hürden bestehen. Kreative Lösungsmöglichkeiten sind denkbar, allerdings sollten Kosten und Nutzen insbesondere bei hohen Teilnehmerzahlen einer Veranstaltung sorgfältig gegeneinander abgewogen werden.

In Bezug auf die Gestaltung von Feedback könnten anhand von Kontrollgruppen-Studien die Effekte (insbesondere die Lernförderlichkeit) verschiedener Arten von Feedback (Punktzahl vs. elaboriertes Feedback auf Basis der Rubrics) miteinander verglichen werden. Daraus ließen sich erweiterte Empfehlungen für die Feedback-Gestaltung ableiten. Allerdings ist die Planung und Durchführung derartiger Kontrollgruppen-Studien im praktischen Lehr-Lernbetrieb mit einigen Herausforderungen verbunden (z. B. mangelnde Repräsentativität bei möglicherweise geringen Teilnehmendenzahlen, Schaffung ungleicher Voraussetzungen bei Trennung in Kontroll- und Experimentalgruppe).

5. E-Assessment für das Gebiet der grafischen Modellierung

In diesem Kapitel werden die Ergebnisse des Forschungsvorhabens bezüglich Forschungsfrage FF2 (siehe Abschnitt 1.2) vorgestellt. Dabei erfolgt zunächst eine Vorstellung der E-Assessment-Plattform, die im Projekt KEA-Mod in Zusammenarbeit mit den Verbundpartnern auf Basis der Ergebnisse zu FF1 aus Kapitel 4 entwickelt wurde (siehe Abschnitt 5.1), um das kompetenzorientierte Prüfen auf dem Gebiet der grafischen Modellierung zu unterstützen. Anschließend werden Konzeption und Implementierung der Plattform-Komponenten vorgestellt, die speziell auf den in Kapitel 2 und 3 dieser Dissertation erarbeiteten Grundlagen basieren: Die Abwicklung der automatisierten Bewertung graphbasierter Diagramme (siehe Abschnitt 5.2) und die Bewertungsdienste für Petri-Netze (siehe Abschnitt 5.3). Darauffolgend wird die Nutzendensicht auf die KEA-Mod-Plattform durch ein Anwendungsbeispiel (Abschnitt 5.4) anschaulich illustriert. Zuletzt erfolgt die Vorstellung verschiedener Evaluationsmaßnahmen (siehe Abschnitt 5.5) sowie eine kritische Reflexion und Diskussion (siehe Abschnitt 5.6).

5.1. KEA-Mod-Plattform

Ein zentrales Ziel im Projekt KEA-Mod war die Erstellung einer E-Assessment-Plattform, die eine automatisierte Bewertung von Modellierungsaufgaben für verschiedene graphbasierte und grafische Modellierungssprachen ermöglicht. Die Anforderungserhebung für die KEA-Mod-Plattform wurde im Projektverbund durchgeführt und in Form eines Lastenhefts (siehe z. B. [Ba09]) anhand funktionaler und nichtfunktionaler Anforderungen sowie User Stories dokumentiert. Bei der Anforderungserhebung wurden neben der im Projektantrag skizzierten Funktionalität der Plattform auch die Ergebnisse zu den didaktischen Grundlagen des Prüfens auf dem

5. E-Assessment für das Gebiet der grafischen Modellierung

Gebiet der grafischen Modellierung (siehe FF1, Kapitel 4) und der systematischen Literaturstudie zu verwandten Ansätzen (siehe Abschnitt 3.3.2) berücksichtigt.

Funktionale Schwerpunkte bildeten die Einbindung kompetenzorientierter Modellierungsaufgaben sowie die Möglichkeit der Festlegung individueller Bewertungsschemata für die Bewertung von Modellierungsaufgaben. Zusätzlich sollte die Plattform so gestaltet werden, dass sie an unterschiedlichen (Hochschul-)Standorten und in einem variablen Lehr- und Lernkontext (z. B. formativ bei Übungen und summativ bei Prüfungen) eingesetzt werden kann. Erforderlich war zudem, dass die von den Verbundpartnern bereits entwickelten Bewertungsdienste über eine noch zu gestaltende Schnittstelle an die Plattform angeschlossen werden konnten. Nichtfunktionale Anforderungen betrafen die Benutzungsfreundlichkeit sowie Datenschutz und Sicherheit. Ebenso lag ein Augenmerk auf der Skalierbarkeit (je nach Ressourcenbedarf bzw. Größe der betreffenden Veranstaltung) und der Erweiterbarkeit (Weiterentwicklung sowie individuelle Anpassung) einzelner Komponenten.

Die KEA-Mod Plattform wurde unter Leitung der Verfasserin dieser Dissertation entwickelt und auf der 20. Fachtagung Bildungstechnologien (DELFI 2022) in Rahmen einer Demo-Session vorgeführt [UI22b] und während der Projektlaufzeit des Projekts KEA-Mod in Lehrveranstaltungen bei den Verbundpartnern [UI22c] pilotiert. Im Folgeprojekt PePP (siehe Abschnitt 1.4) kam die KEA-Mod-Plattform auch in Prüfungen zum Einsatz.

5.1.1. Architektur

Die Architektur der KEA-Mod-Plattform wurde im KEA-Mod-Projektverbund entworfen und eine Beschreibung in [St21; UI21] veröffentlicht. Die Plattform-Komponenten wurden dabei als Microservices konzipiert, um eine spätere Skalierung in Bezug auf die benötigten Ressourcen sowie eine verhältnismäßig einfache Anpassung einzelner Komponenten unabhängig vom Gesamtsystem zu ermöglichen und dadurch die beiden nichtfunktionalen Anforderungen Skalierbarkeit und Erweiterbarkeit (siehe Abschnitt 5.1) zu erfüllen. Ein zusätzlicher pragmatischer Beweggrund für die Trennung in mehrere kleinere UI-Komponenten war die Notwendigkeit der Aufteilung der Entwicklungsarbeiten auf verschiedene Verbundpartner und entwickelnde Personen im KEA-Mod-Projekt. Aus diesem Blickwinkel bot die Trennung den Vorteil, dass eine gemeinsame Abstimmung lediglich in Bezug auf die verwendeten Schnittstellen, jedoch nicht auf die interne Realisierung der jeweiligen Komponenten erfolgen musste. Abb. 5.1 zeigt einen Überblick über die enthaltenen

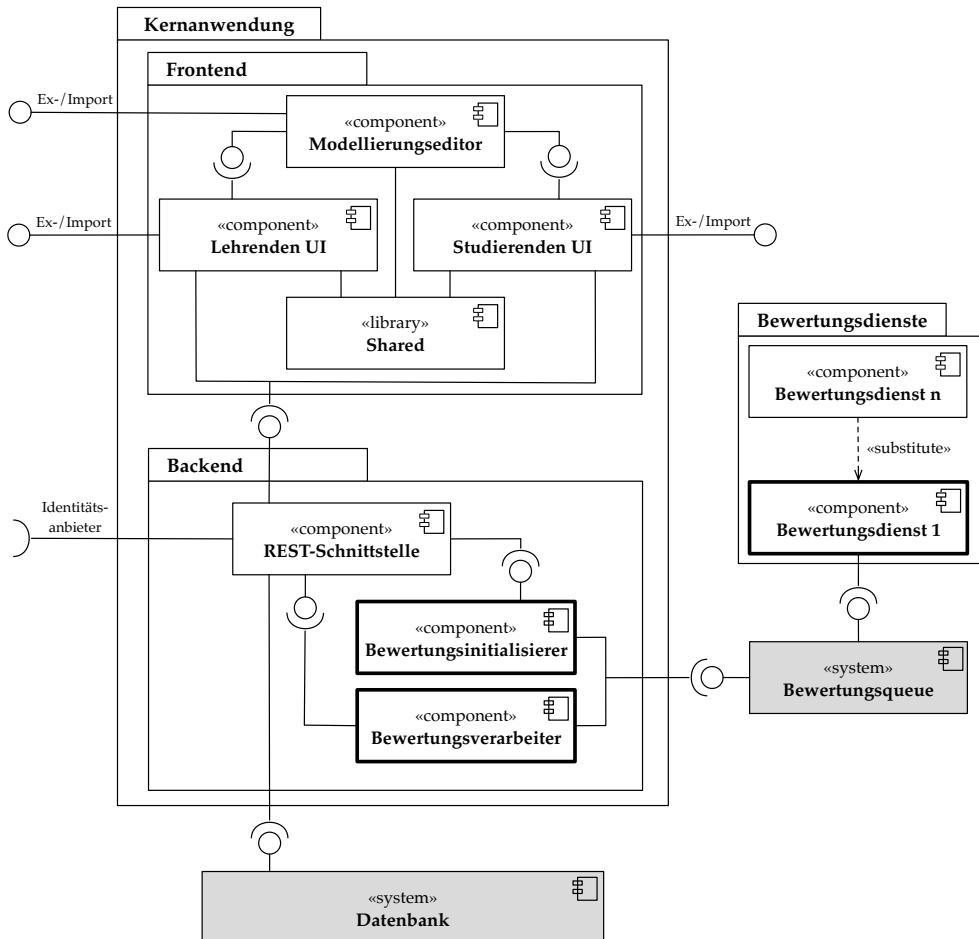


Abb. 5.1.: Überblick über die Architektur der KEA-Mod-Plattform [St21, S. 1358]. Im Rahmen dieser Dissertation entwickelte Komponenten sind fett umrandet.

Komponenten in Form eines UML-Komponentendiagramms. Auf der obersten Ebene wird zwischen der Kernanwendung (bestehend aus Frontend und Backend) sowie Datenbank und Bewertungsqueue als Standardsoftwaresystemen (grau gefärbt), sowie Bewertungsdiensten unterschieden. Im Folgenden sollen die zugrundeliegenden Entwurfsentscheidungen und Funktionen aller Komponenten der Plattform jeweils kurz vorgestellt werden.

Das Frontend besteht aus drei voneinander unabhängigen Komponenten, die jeweils eine Benutzungsoberfläche (engl. »User Interface«, UI) für die Ansicht im Webbrowser

5. E-Assessment für das Gebiet der grafischen Modellierung

ser zur Verfügung stellen: Lehrenden UI, Studierenden UI und Modellierungsektor - die sogenannten UI-Komponenten. Die Lehrenden UI und die Studierenden UI sind von ihrem Funktionsumfang her auf die jeweilige Zielgruppe zugeschnitten. Sie bieten die Möglichkeit zum Anlegen und Verwalten von Aufgabenreihen mit dazugehörigen Aufgaben sowie der Konfiguration von Bewertungsschemata zur automatisierten Bewertung (Lehrenden UI) und dem Bearbeiten und Einreichen von Aufgaben (Studierenden UI) sowie dem Einsehen von Feedback (Lehrenden UI und Studierenden UI). Der Modellierungsektor ist grundsätzlich eigenständig nutzbar und wird sowohl in die Lehrenden UI als auch in die Studierenden UI integriert, um z. B. Musterlösungen oder Modelle zu Aufgaben vom Typ »Modell erstellen« einzugeben, aber auch um bestehende Modelle anzusehen, z. B. im Kontext von Aufgaben vom Typ »Modell verstehen«. Alle drei UI-Komponenten erlauben einen Im- und Export definierter Objekte (z. B. Modelle oder Aufgaben). Gemeinsame Elemente der drei UI-Komponenten werden über die Shared-Bibliothek verwaltet, z. B. um eine einheitliche Darstellung zentral festzulegen. Die Aufteilung der Benutzungsoberfläche in drei eigenständige Anwendungen bietet überdies verschiedene Vorteile. Zum einen entfällt eine aufwändige Überprüfung der den Nutzenden zur Verfügung stehenden Funktionen je nach Rollenzugehörigkeit (Lehrende bzw. Studierende), da entsprechende Funktionen jeweils nur in der betreffenden Komponente implementiert werden. Zum anderen kann der Modellierungsektor als eigenständige Anwendung sowohl in der Lehrenden als auch in der Studierenden UI eingebunden und darüber hinaus völlig unabhängig vom Einsatzkontext E-Assessment als reines Modellierungswerkzeug verwendet werden. Letzteres ermöglicht das Erstellen, Importieren und Exportieren der verschiedenen Diagrammtypen ohne vorheriges Anmelden an der Plattform. Ebenso lassen sich auch die Lehrenden und die Studierenden UI unabhängig voneinander betreiben, was die Ausfallsicherheit erhöht und es erlaubt, im individuellen Bedarfsfall auch nur eine der beiden Benutzungsoberflächen anzubieten.

Im Backend bildet die REST-Schnittstelle die zentrale Komponente, über welche Anfragen aus den UI-Komponenten eingehen. Für diese Anfragen stellt die REST-Schnittstelle zahlreiche Endpunkte zur Verfügung, um über die HTTP-Anfragemethoden POST, GET, PUT und DELETE Einträge in der Datenbank zu erstellen, zu lesen, zu modifizieren beziehungsweise zu löschen und weitere Aktionen im Backend auszulösen. Der primäre Vorteil einer zentralen Datenbank ist es, dass die gespeicherten Daten jederzeit aktuell sind und den beteiligten Komponenten zur Verfügung stehen - es entfällt die Auseinandersetzung mit dem Thema Konsistenz, das bei einer entkoppelten Datenhaltung eines nicht unerheblichen Aufwands be-

darf. Zudem reduziert sich die Komplexität der beteiligten Komponenten, da sie durch die REST-Schnittstelle die Aufgabe einer eigenen Datenhaltung abgenommen bekommen, was der Idee einer Microservice-Architektur mit einer Vielzahl an leichtgewichtigen Komponenten entgegenkommt. Im Hinblick auf die Anforderung des gewählten Anwendungsszenarios, welches einen Einsatz der Plattform an unterschiedlichen (Hochschul-)standorten vorsieht, können über die REST-Schnittstelle die Identitätsanbieter der teilnehmenden Institutionen eingebunden werden. Dadurch wird die Authentifizierung der Nutzenden sowie die Autorisierung der eingehenden Anfragen ermöglicht, um die Endpunkte der REST-Schnittstelle entsprechend der Sicherheitsanforderungen vor unbefugtem Zugriff zu schützen. Dabei wird aufgrund der bestehenden Anforderungen an den Datenschutz im Regelbetrieb bewusst darauf verzichtet, personenbezogene Daten in der Datenbank zu speichern. Lediglich beim Einsatz eines Identitätsanbieters beispielsweise im Prüfungskontext werden der betreffende Identifikator sowie Name und ggf. Matrikelnummer der Nutzenden verschlüsselt in der Datenbank abgelegt.

Identifiziert die REST-Schnittstelle über die UI-Komponenten eine Aktion, die zur Auslösung der automatisierten Bewertung einer studentischen Einreichung führt, werden die notwendigen Daten an den Bewertungsinitialisierer weitergereicht. Dieser prüft das zur betreffenden Aufgabe individuell konfigurierte Bewertungsschema und fügt entsprechende Aufträge zur Bewertung für alle im Bewertungsschema aufgeführten Bewertungsdienste in die Bewertungsqueue ein (siehe auch Abschnitt 5.2.1). Bei der Bewertungsqueue handelt es sich um eine »Message Queue« (dt. etwa Nachrichtenschlange oder auch Nachrichtenwarteschlange), die neben Aufträgen zur Bewertung auch Ergebnisse von Bewertungsdiensten in einem jeweils vordefinierten Nachrichtenformat speichert (siehe auch Abschnitt 5.2.2). Die Bewertungsdienste holen sich dementsprechend bei der Bewertungsqueue die für sie bestimmten Aufträge ab, bearbeiten sie und schreiben anschließend ihre Ergebnisse zusammen mit einer Liste von individuellen Feedback-Nachrichten wieder dorthin zurück. Dort werden sie vom Bewertungsverarbeiter abgeholt, der das bisherige Gesamtergebnis auf Grundlage des Teilergebnisses des Bewertungsdienstes und der über das Bewertungsschema festgelegten Gewichtung aktualisiert. Ebenso werden die individuellen Feedback-Nachrichten des Bewertungsdienstes zurück an die REST-Schnittstelle gereicht. Anschließend können das Gesamtergebnis und die Feedback-Nachrichten über die UI-Komponenten von den Nutzenden eingesehen werden.

Diese Gestaltung der Architektur bietet die Möglichkeit der flexiblen Integration weiterer Bewertungsdienste über die definierten Schnittstellen, unabhängig von der

5. E-Assessment für das Gebiet der grafischen Modellierung

verwendeten Technologie und vom physischen Standort. Die Bewertungsdienste wurden in der Plattform-Architektur bewusst außerhalb der Kernanwendung angesiedelt, so dass diese bei Bedarf auch auf externen Servern liegen können, die sich möglicherweise nicht unter Kontrolle der Plattformbetreibenden (Kernanwendung und Standardsoftwaresysteme) befinden. Dies soll die Anbindung bereits bestehender Dienste erleichtern. Durch die gewählte Form der Bewertungsqueue ist die Ausfallsicherheit auf diesem Kommunikationsabschnitt erhöht: Sollte ein Bewertungsdienst zeitweise nicht aktiv sein, werden zwischenzeitlich eingehende Aufträge in der Bewertungsqueue gespeichert und können regulär verarbeitet werden, sobald der entsprechende Bewertungsdienst wieder zur Verfügung steht.

Die Verfasserin dieser Dissertation hat insbesondere die Gestaltung der Schnittstellen zwischen den Plattform-Komponenten sowie entsprechend ihres Forschungsschwerpunktes die Konzeption und Implementierung der Komponenten Bewertungsinitialisierer und Bewertungsverarbeiter im Backend der Kernanwendung und mehrerer Bewertungsdienste für die Bewertung von Petri-Netzen übernommen. Darüber hinaus war sie für die Einrichtung und den Betrieb mehrerer Instanzen der KEA-Mod-Plattform zu Test- und Produktionszwecken in der BW-Cloud¹ sowie am Rechenzentrum des KIT verantwortlich.

5.1.2. Datenmodell

Ergänzend zum Komponentendiagramm bezüglich der Plattform-Architektur soll das semantische Datenmodell in Abb. 5.2 einen groben Überblick über die zentralen Entitäten bieten, die in der KEA-Mod-Plattform verwaltet werden. Das Datenmodell wurde als einfaches UML-Klassendiagramm (d.h. ohne Attribut- und Methodenliste, siehe auch Abschnitt 2.2.3 zu UML-Klassendiagrammen) dargestellt². Eine Aufgabenreihe dient dazu eine Menge von Aufgaben zu bündeln, um Studierenden diese als Bearbeitungseinheit zur Verfügung zu stellen. Dadurch können in der Praxis gängige Konzepte wie z.B. Übungsblätter oder Klausuren abgebildet werden. Aufgaben haben jeweils einen speziellen Aufgabentyp, gemeinsame Eigenschaften werden über die Vererbungshierarchie abgebildet. In Abstimmung mit den Beteiligten im Projektverbund wurden während der Projektlaufzeit zunächst die beiden folgenden Aufgabentypen in der KEA-Mod-Plattform implementiert:

¹ »Infrastructure-as-a-Service«-Umgebung, speziell für den Betrieb virtueller Maschinen im Kontext von Forschung und Lehre in Baden-Württemberg - <https://www.bw-cloud.org/>

² Es handelt sich dabei um ein konzeptuelles Datenmodell als abstrakte Beschreibung des konkret umgesetzten logischen Datenmodells.

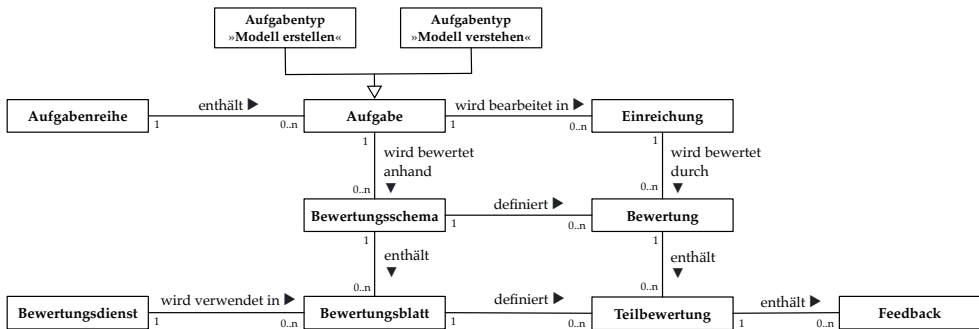


Abb. 5.2.: Überblick über die zentralen Entitäten im Datenmodell der KEA-Mod-Plattform (abstrahierte Form)

- »Modell erstellen«: Entspricht dem gleichnamigen Aufgabentyp, der in Abschnitt 3.1.2 vorgestellt wurde. Gegeben ist eine kurze Aufgabenbeschreibung zusammen mit einem textuellen Szenario, welches modelliert werden soll.
- »Modell verstehen«: Bei diesem Aufgabentyp ist ein Modell gegeben, zu welchem im Antwort-Wahl-Verfahren gegebene Antwortoptionen als richtig oder falsch deklariert werden müssen. Damit lassen sich die in Abschnitt 3.1.2 vorgestellten Aufgabentypen »Modell erläutern«, »Fehler im Modell finden« sowie »Modell analysieren« in konvergenter Form (siehe Abschnitt 2.1.3) darstellen.

Um die automatisierte Bewertung individuell auf die Wünsche der Lehrenden anpassen zu können, wird jeder Aufgabe ein flexibel zu gestaltendes Bewertungsschema zugewiesen. Dieses bestimmt, wie die Einreichungen, die von Studierenden zu dieser Aufgabe erstellt werden, bewertet werden sollen. Ein Bewertungsschema weist eine Baumstruktur auf. Dabei legen die Blätter dieses Baumes (Bewertungsblatt) fest, welche Bewertungsdienste mit welcher Gewichtung aufgerufen werden sollen, um die Bewertung der Aufgabe vorzunehmen. In einer Bewertung zu einer Einreichung werden die einzelnen Teilergebnisse der beteiligten Bewertungsdienste zu einer Gesamtbewertung aggregiert. Die interne Struktur von Bewertungsschemata und die Abwicklung der automatisierten Bewertung auf dieser Basis werden in Abschnitt 5.2 detailliert vorgestellt.

5.1.3. Technische Realisierung

Die technische Realisierung der in Abschnitt 5.1.1 vorgestellten Plattformkomponenten kann auf unterschiedlichen Wegen erfolgen. Im Rahmen der hier beschriebenen Microservice-Architektur sind die jeweils verwendeten Technologien³ grundsätzlich austauschbar. In enger Abstimmung zwischen den an der Entwicklung beteiligten Personen aus dem Projektkonsortium wurden bei der Technologieauswahl daher folgende Auswahlkriterien herangezogen, die sich aus den Rahmenbedingungen und den Anforderungen des KEA-Mod-Projekts ergeben:

- **Open-Source-Lizenz:** Die einzelnen Komponenten der KEA-Mod-Plattform werden unter einer Open-Source-Lizenz veröffentlicht, um die Anschlussfähigkeit und den Transfer der Forschungsergebnisse zu anderen Projekten, Hochschulen, sowie der interessierten Öffentlichkeit und Praxis zu unterstützen. Daher müssen auch die bei der Realisierung der Komponenten eingesetzten Technologien unter einer kompatiblen Open-Source-Lizenz veröffentlicht sein.
- **Nachhaltigkeit und Weiterentwicklungspotenzial:** Die eingesetzten Technologien sollen sich neben einer ausreichend weiten Verbreitung (Betrachtung der Nutzungs- und Download-Zahlen) durch längeres Bestehen sowie eine aktive Weiterentwicklung auszeichnen, um das Weiterbestehen der Plattform sowie die Weiterentwicklung durch andere (z. B. innerhalb der Wissenschaftsgemeinde auf dem Gebiet der Modellierung) zu begünstigen.
- **Bestehendes Know-How:** Die Entwicklung der einzelnen Komponenten wurde auf verschiedene Personen aus dem Projektkonsortium aufgeteilt. Aufgrund der knappen Ressourcen für die Entwicklung sollen die jeweiligen Hauptverantwortlichen für die Umsetzung einer Komponente auf bisherige Erfahrungen im Umgang mit konkreten Technologien und bereits bestehende Lösungen zurückgreifen, um die Einarbeitungszeit zu verkürzen und eine zügige Realisierung zu ermöglichen.

Alle Frontend-Komponenten basieren auf der weit verbreiteten JavaScript-Bibliothek React⁴ für die Entwicklung interaktiver und webbasierter Benutzungsoberflächen. Der Modellierungseeditor bindet zusätzlich die Bibliothek JointJS⁵ ein, anhand derer die dynamische Erstellung von Diagrammen umgesetzt wird. Sowohl React als auch

³ Der Begriff Technologie soll hier stellvertretend für die jeweils verwendete Kombination aus Programmiersprache und gegebenenfalls Frameworks und/oder externen Bibliotheken stehen. Auch ausgewählte Standardsoftwaresysteme sollen dabei mit eingeschlossen werden.

⁴ <https://reactjs.org/>

⁵ <https://www.jointjs.com/opensource>

JointJS sind lizenztechnisch mit den Projektanforderungen kompatibel und wurden bereits in früheren Prototypen eingesetzt, die im Zusammenhang mit vorangehenden Experimenten im Rahmen dieser Dissertation am KIT entstanden sind.

Die REST-Schnittstelle im Backend basiert auf Express⁶, einem etablierten Node.js⁷-Framework. Dies ermöglicht die Konfiguration der Endpunkte anhand von HTTP-Anfragemethoden. Zusätzlich wird Sequelize⁸ für die objektrelationale Abbildung (engl. *object-relational mapping*, *ORM*) zur MySQL-Datenbank⁹ verwendet, um die Programmierung der Datenbankanbindung zu erleichtern. Auch die hier genannten Technologien sind unter kompatiblen Open-Source-Lizenzen veröffentlicht. Der Bewertungsinitialisierer und -verarbeiter sind ebenfalls in Javascript (Node.js) umgesetzt, da somit einige Basisfunktionen, die bereits in der REST-Schnittstelle umgesetzt wurden, wiederverwendet werden konnten.

Die Kommunikation über Datentransferobjekte in Form von Protocol Buffers¹⁰ über die mittels Apache Kafka¹¹ realisierte Bewertungsqueue entspricht dem Ansatz, der sich im E-Assessment-System JACK des Projektpartners Universität Duisburg-Essen bewährt hat. Für die Realisierung der Bewertungsdienste können beliebige Technologien zum Einsatz kommen, solange die Abwicklung der Kommunikation über die Bewertungsqueue (Apache Kafka) anhand der festgelegten Datentransferobjekte (Protocol Buffers) unterstützt wird. Die im Rahmen des KEA-Mod-Projekts entwickelten Bewertungsdienste wurden mit Java, JavaScript und Python umgesetzt, da sie auf verschiedenen Vorarbeiten der beteiligten Projektpartner basieren, die im Rahmen des Projekts an die KEA-Mod-Plattform angeschlossen wurden. Für Apache Kafka und Protocol Buffers existieren weiterhin Bibliotheken für z. B. die Sprachen C++, Dart, Go, Kotlin, Objective-C und PHP, so dass eine Integration entsprechender bereits bestehender Ansätze zur automatisierten Bewertung graphbasierter Diagramme an die KEA-Mod-Plattform möglich ist, ohne diese Ansätze in eine bestimmte Sprache überführen bzw. reimplementieren zu müssen.

6 <https://expressjs.com/>

7 <https://nodejs.org/>

8 <https://sequelize.org/>

9 <https://www.mysql.com/>

10 <https://developers.google.com/protocol-buffers>

11 <https://kafka.apache.org/>

5.1.4. Modellierungseeditor und Austauschformat

Für den im Forschungsvorhaben behandelten Kontext Hochschullehre sind sowohl herkömmliche Modellierungseeditoren mit Modellierungsunterstützung als auch die bestehenden sprachspezifischen Austauschformate für Modelle nicht geeignet. Wenn Lernziele zur Kenntnis der zu einer Modellierungssprache gehörigen Modellierungselemente sowie Syntax-Regeln zu deren Verknüpfung überprüft werden sollen (siehe Abschnitte 2.2.3 und 4.3), muss es möglich sein, syntaktisch falsche Modelle (i) erstellen und entsprechend (ii) in einem Austauschformat abspeichern zu können.

Diese Anforderungen wurden im Rahmen des Forschungsvorhabens bereits vor Beginn des KEA-Mod-Projekts erkannt und im Kontext einer Abschlussarbeit¹² behandelt. In dieser wurde ein Prototyp für einen entsprechenden Modellierungseeditor (moEd) entwickelt, der in mehreren Lehrveranstaltungen für die Sammlung studentischer Petri-Netze zum Einsatz kam (siehe auch Abschnitt 5.5.1). Die im moEd-Prototyp enthaltene Palette von Modellierungselementen bietet neben Petri-Netz-Elementen auch weitere Elemente als sogenannte Distraktoren an, zusätzlich wird keine Modellierungsunterstützung angeboten. Somit können auch Modelle erstellt werden, die die Syntax-Regeln von Petri-Netzen verletzen. Um diese Modelle speichern und automatisiert bewerten lassen zu können, kam das für Petri-Netze verbreitete Austauschformat PNML (Petri Net Markup Language)¹³ nicht in Frage, da dieses keine Möglichkeit für die Speicherung der Distraktoren oder unvollständiger Kanten (fehlendes Ziel und/oder Quelle) bietet. Anstatt das XML-basierte PNML-Austauschformat entsprechend zu erweitern, wurde das interne Format zur Repräsentation von Diagrammen der im moEd-Prototypen verwendeten Javascript-Bibliothek JointJS (siehe Abschnitt 5.1.3) in JSON (Javascript Object Notation) aufgegriffen. Auf Basis dieses Formats wurde nicht nur die Speicherung von Petri-Netzen, sondern von graphbasierten Diagrammen im Allgemeinen ermöglicht. Dabei können Knoten-, Kanten- und Spezialtypen (siehe Abschnitt 2.2.3) als Bestandteile der betrachteten grafischen und graphbasierten Modellierungssprachen ebenso wie alle Informationen der formalen Darstellung von Geschäftsprozessgraphen (siehe Abschnitt 3.2.3) entsprechend abgebildet werden.

Im Projektverbund wurde die Entwicklung eines sprachübergreifenden Modellierungseeditors in der KEA-Mod-Plattform vom Verbundpartner UDE durchgeführt.

¹² Breno Aberle, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Entwurf und Implementierung eines Petri-Netz Editors als Microservice für den Einsatz in der Hochschullehre*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2019.

¹³ <https://www.pnml.org>

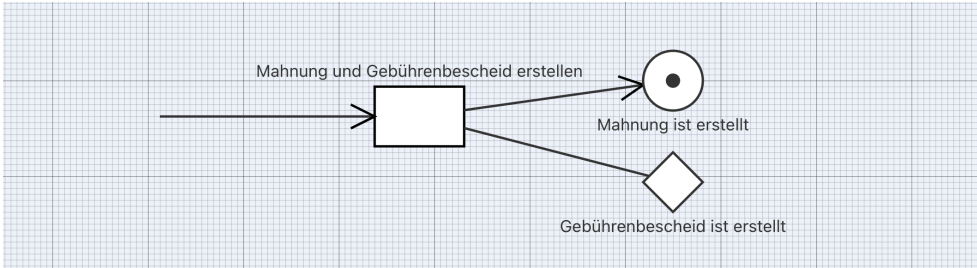


Abb. 5.3.: Modell, das die Petri-Netz-Syntax-Regeln verletzt. Screenshot aus dem Modellierungseditor der KEA-Mod-Plattform.

Dabei wurde der moEd-Prototyp als Ausgangsbasis übernommen und das JSON-Format der internen JointJS-Repräsentation als sprachübergreifendes Austauschformat mit dem Namen KEA (Dateiendung `.kea`) deklariert. Das JSON-Format dient im Kontext der KEA-Mod-Plattform nicht nur zur Serialisierung von Modellen aus dem Modellierungseditor in der Datenbank, sowie zum Im- und Export über Dateien. Es kann zudem über vorhandene Parsing-Bibliotheken mit den meisten Programmiersprachen bequem verarbeitet werden, so dass es auch als Austauschformat bei der Auswertung durch die Bewertungsdienste in der KEA-Mod-Plattform zum Einsatz kommt.

In Abb. 5.3 ist ein Modell dargestellt, das mehrere Petri-Netz-Syntax-Regeln verletzt. So besitzt die Transition `Mahnung und Gebührenbescheid erstellen` eine eingehende Kante ohne Quelle, das Element mit der Beschriftung `Gebührenbescheid ist erstellt` ist kein Petri-Netz-Modellierungselement, ebenso ist die ungerichtete Kante kein (Standard-)Petri-Netz-Modellierungselement. Auflistung 5.1 zeigt eine gekürzte Fassung¹⁴ dieses Beispiels im KEA-Format. Im KEA-Format ist ein Modell (`graph`) als Menge von Modellierungselementen (`cells`) gespeichert. Kanten sind dabei vom Typ `kea.Link`, Knoten und Gruppierungselemente (entsprechend Spezialtypen in Abschnitt 2.2.3) werden durch jeweils eigene Typen (`type`) repräsentiert. Alle Modellierungselemente besitzen neben ihrem Typ eine eindeutige ID (`id`), sowie Informationen zur grafischen Darstellung, z.B. Größe (`size`, `width`, `height`) und Position (`position`). Optional ist die Angabe einer Beschriftung und deren Position (`label` bzw. `labelPosition`). Kanten enthalten zusätzlich Angaben zu Quelle und Ziel (`source` und `target`). Zeigen diese auf ein Modellierungselement, so wird hier die ID dieses Elements eingetragen, anderenfalls die Position anhand der Koordinaten (`x` und `y`). Die Abbildung der Koordinaten im

¹⁴ Aus Platzgründen wurden einige Attribute wie z. B. Farben und weitere interne Parameter entfernt.

5. E-Assessment für das Gebiet der grafischen Modellierung

```
1 { "graph": { "cells": [  
2   { "type": "kea.Rectangle",  
3     "size": { "width": 75, "height": 50 },  
4     "label": "Mahnung_und_Gebührenbescheid_erstellen",  
5     "labelPosition": "ABOVE",  
6     "id": "e64b8bdf-cb59-49b8-bf09-68107814",  
7     "position": { "x": 1230, "y": 405 } },  
8   { "type": "kea.Link",  
9     "linkType": "solid",  
10    "targetMarker": "arrow",  
11    "sourceMarker": "none",  
12    "source": { "id": "e64b8bdf-cb59-49b8-bf09-68107814" },  
13    "target": { "id": "11260d98-2c67-4289-98cf-35f2685d" },  
14    "id": "c8db443f-7ca1-404f-8f1a-5642b293" },  
15   { "type": "kea.Circle",  
16     "size": { "width": 50, "height": 5 },  
17     "id": "11260d98-2c67-4289-98cf-35f2685d",  
18     "position": { "x": 1455, "y": 375 },  
19     "labelPosition": "BELOW",  
20     "label": "Mahnung_ist_erstellt" },  
21   { "type": "kea.Circle",  
22     "size": { "width": 10, "height": 10 },  
23     "id": "708e5ccb-4a35-4f91-b373-36016432",  
24     "position": { "x": 1475, "y": 395 } },  
25   { "type": "kea.Link",  
26     "linkType": "solid",  
27     "targetMarker": "none",  
28     "sourceMarker": "none",  
29     "source": { "id": "e64b8bdf-cb59-49b8-bf09-68107814" },  
30     "target": { "id": "f20d9537-a3c5-442f-bd83-fba0a11a" },  
31     "id": "18c2defc-73b3-42c2-b54e-c0a989f6" },  
32   { "type": "kea.Polygon",  
33     "size": { "width": 50, "height": 50 },  
34     "label": "Gebührenbescheid_ist_erstellt",  
35     "id": "f20d9537-a3c5-442f-bd83-fba0a11a",  
36     "position": { "x": 1455, "y": 460 },  
37     "labelPosition": "BELOW" },  
38   { "type": "kea.Link",  
39     "linkType": "solid",  
40     "targetMarker": "arrow",  
41     "sourceMarker": "none",  
42     "source": { "x": 1050, "y": 430 },  
43     "target": { "id": "e64b8bdf-cb59-49b8-bf09-68107814" },  
44     "id": "76bb4919-3c84-444a-87d9-10b693a6", }  
45 ] } }
```

Auflistung 5.1: Modell aus Abb. 5.3 im KEA-Format (gekürzte Form)

KEA-Format ermöglicht es auch, pragmatische Qualitätsaspekte eines Modells, wie z.B. das Auftreten von Kantenüberschneidungen zu überprüfen. Dies ist beispielsweise mit dem EPML-Austauschformat, welches für EPK verwendet wird, nicht möglich.

Das KEA-Format lässt sich bei Bedarf auch in gängige sprachspezifische Austauschformate (z. B. PNML oder EPML) konvertieren. Allerdings müssen dabei vorhandene Syntaxfehler und im Ziel-Austauschformat unbekannte Modellierungselemente berücksichtigt und gegebenenfalls bereinigt werden, so dass dabei mit einem Informationsverlust zu rechnen ist. Da das KEA-Format ein gemeinsames Austauschformat für die in der Plattform berücksichtigten Modellierungssprachen¹⁵ verwirklicht, ist die Erstellung sprachübergreifender Bewertungsdienste möglich (siehe z. B. den Pragmatik-Checker in Abschnitt 5.3.4, der im Rahmen dieser Dissertation entwickelt wurde).

Um im Rahmen der automatisierten Bewertung nicht nur das Ergebnisartefakt »studentisches Modell« sondern perspektivisch auch die Arbeitsschritte während der Erstellung des Modells berücksichtigen zu können (z. B. zur Überprüfung prozessbezogener Kompetenzen, siehe Abschnitt 4.2 und Abschnitt 6.3), ist es im Modellierungsektor möglich, bei Bedarf auch die entsprechenden Interaktionsdaten der Nutzenden aufzuzeichnen.

5.2. Prozess der automatisierten Bewertung

Der Schwerpunkt dieses Abschnitts liegt auf der Beschreibung der Kernelemente, die am Prozess der automatisierten Bewertung in der KEA-Mod-Plattform beteiligt sind. Verantwortlich für die Steuerung des Bewertungsprozesses sind das für eine Aufgabe festgelegte Bewertungsschema (Abschnitt 5.2.1), die Datentransferobjekte, in welchen die Informationen zwischen den an der Bewertung beteiligten Komponenten weitergereicht werden (Abschnitt 5.2.2) sowie die beiden im Backend für die Ausführung der Bewertung verantwortlichen Komponenten (Bewertungsinitiator, Abschnitt 5.2.3 und Bewertungsverarbeiter, Abschnitt 5.2.4). Abschließend werden die Einsatzmöglichkeiten der KEA-Mod-Plattform anhand der Konfigurationsoptionen für den Beginn der automatisierten Bewertung (Abschnitt 5.2.5) kurz skizziert.

¹⁵ Stand 01/2024 ist in der KEA-Mod-Plattform die Erstellung von UML-Klassen, Aktivitäts- und Zustandsdiagrammen, ER-Diagrammen, EPK, Petri-Netzen und BPMN-Modellen möglich.

5.2.1. Bewertungsschema

Wie in der Zielsetzung zu Forschungsfrage FF2 (Abschnitt 1.2) und in den Zielen des Projekts KEA-Mod (Abschnitt 5.1) aufgeführt, soll es Lehrenden in der KEA-Mod-Plattform ermöglicht werden, eine flexible und individuelle Gestaltung der automatisierten Bewertung vorzunehmen. Zu diesem Zweck wurde ein Konzept für den Aufbau von Bewertungsschemata¹⁶ entworfen und umgesetzt. An ein Bewertungsschema wurden im Projektkontext folgende Anforderungen gestellt:

- A1:** Ein Bewertungsschema legt für eine Aufgabe fest, welche Bewertungsdienste bei der automatisierten Bewertung einer Einreichung zu dieser Aufgabe zum Einsatz kommen sollen.
- A2:** Eine individuelle Konfiguration der beteiligten Bewertungsdienste soll möglich sein, z. B. um eine aufgabenspezifische Regel-Datei anzuhängen oder die Ausführung eines Bewertungsdienstes nach eigenen Wünschen zu parametrisieren.
- A3:** Einzelne Bestandteile eines Bewertungsschemas, die nicht aufgabenspezifisch sind, sollen aufgabenübergreifend wiederverwendet werden können.
- A4:** Das Bewertungsschema soll für die Nutzenden (Lehrende und Studierende) verständlich darstellbar und das Ergebnis der Bewertung nachvollziehbar sein.
- A5:** Die Teilergebnisse der an der Bewertung beteiligten Bewertungsdienste sollen zu einem skalaren Gesamtergebnis aggregiert werden können.

Unter Berücksichtigung dieser Anforderungen bietet sich eine Baumstruktur für den Aufbau eines Bewertungsschemas an. Dies deckt sich nicht nur mit dem von MOODY [Mo05a] vorgeschlagenen hierarchischen Aufbau von Qualitätsframeworks (siehe Abschnitt 3.2.3), auch andere technische Umsetzungen kombinierter Bewertungsansätze basieren auf einer Baumstruktur. Beispielsweise werden in der Lernplattform Moodle¹⁷ Bewertungskategorien eingesetzt, über welche Einzelergebnisse aggregiert werden und in eine Gesamtbewertung einfließen. Auch das ProFormA-Format zur Spezifikation von Bewertungsschemata für automatisiert bewertbare Programmieraufgaben [Ga19a; Ga19b], Grappa, eine Middleware zwischen Lehr-Lernsystemen

¹⁶ In dieser Arbeit wird anhand der Begriffe Bewertungsschema und Rubric die folgende Unterscheidung getroffen: ein Bewertungsschema ist ein digitales Artefakt in der Datenbank der KEA-Mod-Plattform und legt fest, (i) welche Bewertungsdienste mit (ii) welcher Gewichtung und (iii) welchen individuellen Konfigurationsparametern bei der automatisierten Bewertung eines studentischen Modells zum Einsatz kommen. Bei einem Rubric handelt es sich dahingegen um ein analoges Artefakt, welches die kriteriengeleitete Bewertung von studentischen Modellen durch eine bewertende Person ermöglicht (siehe Abschnitte 2.1.4 und 4.3).

¹⁷ <https://moodle.org>

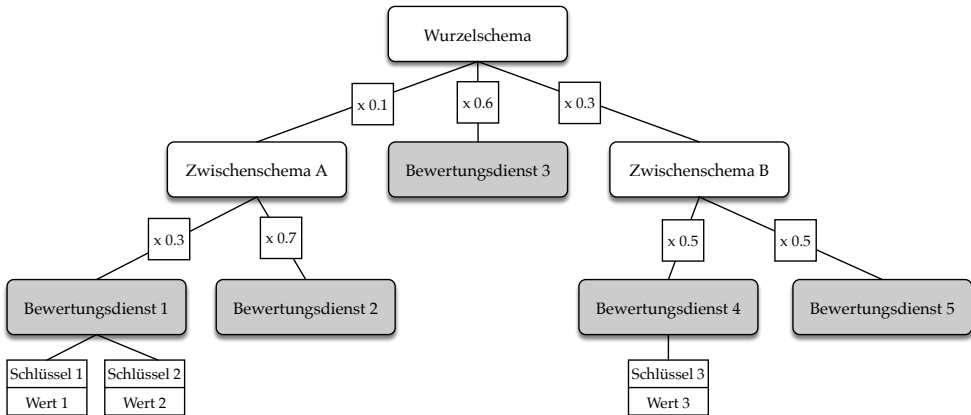


Abb. 5.4.: Bewertungsschema mit Baumstruktur in der KEA-Mod-Plattform. Knoten werden unterschieden in Schemata (weiß) und Bewertungsblätter (grau).

und automatisierten Bewertungsverfahren [GHW15] und Graja, zur Bewertung von Java-Programmieraufgaben [Ga17b], unterstreichen den Nutzen von Bewertungsschemata in Form einer Baumstruktur. Die durch den Baum dargestellten Zusammenhänge erlauben es Nutzenden, das Zustandekommen des Bewertungsergebnisses auf übersichtliche Weise nachzuvollziehen (Anforderung A4). Darüber hinaus lassen sich die in Abschnitt 4.3 vorgestellten Rubrics ebenfalls durch eine Baumstruktur abbilden und somit in der KEA-Mod-Plattform entsprechend umsetzen.

Abb. 5.4 zeigt den Aufbau eines Bewertungsschemas in der KEA-Mod-Plattform. Bei dem Schema auf der obersten Ebene handelt es sich um das sogenannte Wurzelschema. Dieses wird mit einer Aufgabe verknüpft, für die es Anwendung finden soll (Anforderung A1). Kind-Elemente eines Schemas können entweder wiederum auf ein Schema (sog. Zwischenschema) oder aber auf einen aufzurufenden Bewertungsdienst verweisen (Anforderung A1). Alle Kind-Elemente sind mit ihrem jeweiligen Eltern-Element über eine Gewichtung im Intervall $[0, 1]$ verknüpft, zusätzlich gilt die Einschränkung, dass die Summe der Gewichte aller Kind-Elemente zum gleichen Eltern-Element (sog. benachbarte Elemente) jeweils 1 ergibt.

Kind-Elemente, die auf einen Bewertungsdienst verweisen, repräsentieren die Blätter des Baumes¹⁸. An diese können keine weiteren Kind-Elemente angehängt werden. Stattdessen können an dieser Stelle beliebig viele frei definierbare Schlüssel-Wert-Paare (auch in Form von Text- oder Binärdateien) übergeben werden. Somit ist es

¹⁸ Diese Art von Kind-Elementen entspricht der im Datenmodell in Abschnitt 5.1.2 als »Bewertungsblatt« bezeichneten Entität.

5. E-Assessment für das Gebiet der grafischen Modellierung

möglich den Aufruf eines Bewertungsdienstes individuell zu parametrisieren (Anforderung A2), ohne dass die intern verwendeten Parameter jedes in der Plattform integrierten Bewertungsdienstes im Datenbankschema festgeschrieben werden müssen. Beispielsweise könnte ein Bewertungsdienst für Petri-Netze über einen Schlüssel `workflow` mit den Werten `true` oder `false` angewiesen werden, auf die Erfüllung der Workflow-Netzeigenschaft (siehe Abschnitt 2.2.4) zu prüfen. Darüber hinaus lassen sich auch weitere aufgabenspezifische Informationen, die in der Datenbank zur jeweiligen Aufgabe hinterlegt wurden, optional übermitteln, wenn diese vom jeweiligen Bewertungsdienst interpretiert werden sollen (z. B. Musterlösungen oder die textuelle Beschreibung des Szenarios bei einer »Modell erstellen«-Aufgabe).

Durch den Einsatz von Zwischenschemata kann ein Teilbaum eines Bewertungsschemas zu einem bestimmten Aspekt (z.B. Pragmatik) aufgabenübergreifend wiederverwendet werden, indem dieser an andere Bewertungsschemata angehängt wird (Anforderung A3). Dabei kann eine individuelle bzw. aufgabenspezifische Gewichtung für den kompletten Teilbaum erfolgen. Beispielsweise kann Zwischenschema A, welches in Abb. 5.4 mit einer Gewichtung von 0.1 zum Einsatz kommt, in einem anderen Bewertungsschema mit einer anderen Gewichtung eingesetzt werden.

Durch die Umsetzung der Gewichtung von Teilergebnissen anhand relativer Gewichte im Intervall $[0, 1]$ können Nutzende bequem und unmittelbar ablesen bzw. festlegen, mit welchem Anteil (z. B. $0.6 = 60\%$) ein Teilergebnis in die jeweilige obere Ebene einfließt (Anforderung A4). Bei einer Verwendung absoluter Gewichte ist dies für Nutzende umständlicher, da bei der Bestimmung des Gewichtungsanteils eines einzelnen Knotens auch die Gewichte aller benachbarten Baumknoten betrachtet werden müssen¹⁹.

Anhand der Baumstruktur des Bewertungsschemas lassen sich die Teilergebnisse der Bewertungsdienste zu einem Gesamtergebnis aggregieren (Anforderung 5), solange alle Bewertungsdienste ein relatives Ergebnis im Intervall $[0, 1]$ zurückliefern. Somit lässt sich stufenweise das Gesamtergebnis für die jeweilige Aufgabe am Wurzelknoten des Schemas berechnen. Da sich das Gesamtergebnis ebenfalls im Intervall $[0, 1]$ befindet, kann es in eine beliebige Skala überführt werden (siehe auch Abschnitt 2.1.4). In den Frontend-Komponenten der KEA-Mod-Plattform erfolgt eine Umrechnung in die erreichte Punktzahl, gemessen an der maximal erreichbaren Punktzahl für die betreffende Aufgabe. Somit entspricht ein beispielhaftes Gesamtergebnis von

¹⁹ Sei das G_k das absolute Gewicht eines Knotens k und $G_{n_1} \dots G_{n_n}$ die Gewichte seiner Nachbarknoten n_1 bis n_n im Baum. Für die Berechnung des Anteils der Gewichtung von G_k gilt $\frac{G_k}{G_k + G_{n_1} + \dots + G_{n_n}}$.

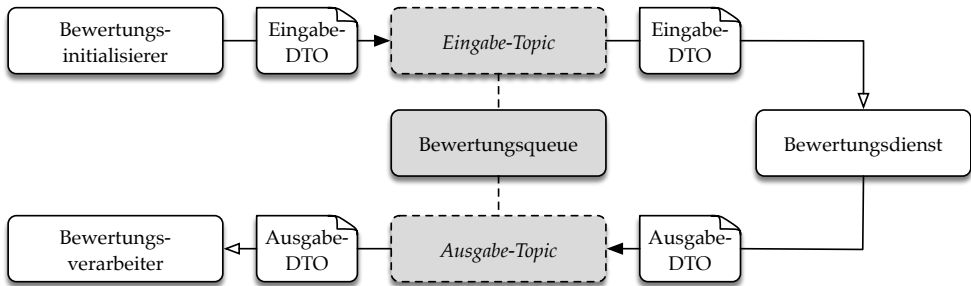


Abb. 5.5.: Datentransferobjekte (DTOs) in der KEA-Mod-Plattform. Kanten mit ausgefüllter Pfeilspitze (schwarz) deuten an, dass ein DTO produziert wird, Kanten mit nicht ausgefüllter Pfeilspitze (weiß) dahingegen, dass ein DTO konsumiert wird.

0.5 (50 %) je nach festgelegter maximal erreichbarer Punktzahl z. B. 1/2, 2/4, 2.5/5 oder 5/10 Punkten.

5.2.2. Datentransferobjekte

Für die anstehende Beschreibung weiterer Komponenten der KEA-Mod-Plattform in den folgenden Abschnitten ist eine vorherige Einführung der sogenannten Datentransferobjekte (DTOs) zur Abwicklung der automatisierten Bewertung hilfreich. Abb. 5.5 illustriert dabei zunächst den Weg der DTO als Nachrichten durch die beteiligten Plattformkomponenten. Für jeden laut Bewertungsschema an der Bewertung beteiligten Bewertungsdienst wird ein entsprechendes Eingabe-DTO mit den notwendigen Informationen zur Bewertung vom Bewertungsinitialisierer erzeugt (siehe auch Abschnitt 5.2.3) und eine Nachricht über das individuelle Eingabe-Topic dieses Bewertungsdienstes in der Bewertungsqueue produziert. Der Bewertungsdienst konsumiert die Nachrichten seines individuellen Eingabe-Topics und kann mit den enthaltenen Informationen im Eingabe-DTO die entsprechende Bewertung durchführen. Die berechneten Ergebnisse werden in einem Ausgabe-DTO verpackt und an das generelle Ausgabe-Topic der Bewertungsqueue übergeben. Dieses Ausgabe-Topic wird vom Bewertungsverarbeiter überwacht, der alle dort eingehenden Nachrichten konsumiert und entsprechend weiterverarbeitet (siehe auch Abschnitt 5.2.4).

Je nach Aufgabentyp werden unterschiedliche Eingabe-DTO erzeugt, da jeweils unterschiedliche Dinge bewertet werden müssen und somit unterschiedliche Informationen benötigt werden. Abb. 5.6 stellt die Spezifikation für das Ein- und

5. E-Assessment für das Gebiet der grafischen Modellierung

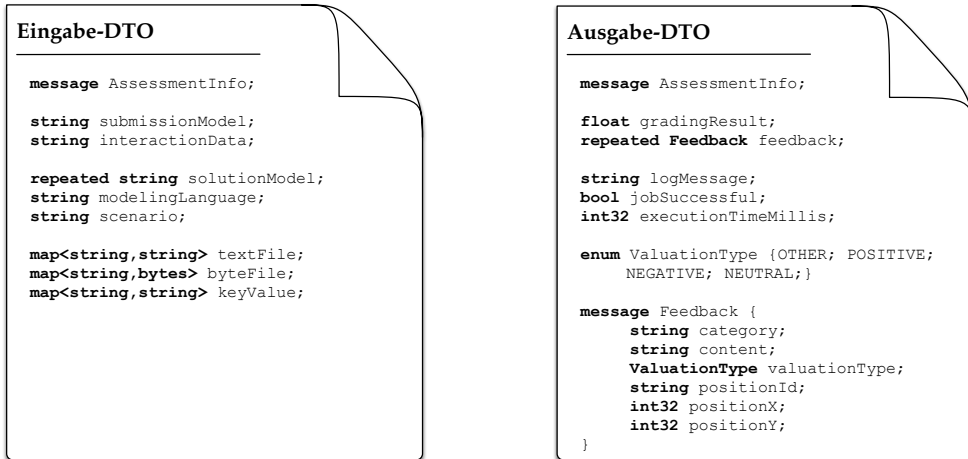


Abb. 5.6.: Spezifikation von Ein- und Ausgabe-DTO zum Aufgabentyp »Modell erstellen«. Darstellung in Anlehnung an das proto3-Format der verwendeten Protocol Buffers.

Ausgabe-DTO zum Aufgabentyp »Modell erstellen« im proto3-Format²⁰ dar. Dabei enthalten sowohl das Ein- als auch das Ausgabe-DTO eine gemeinsame Nachricht mit Meta-Informationen zur betreffenden Bewertung (AssessmentInfo). Diese Meta-Informationen dürfen durch die Bewertungsdienste nicht verändert werden und dienen dazu, deren Ergebnisse am Ende des Bewertungsprozesses in der Datenbank der korrekten Bewertung zuordnen zu können.

Neben dem studentischen Modell, das zu einer Aufgabe vom Typ »Modell erstellen« eingereicht wurde (submissionModel), können über das Eingabe-DTO optional²¹ auch die Interaktionsdaten, die durch das Modellierungswerkzeug erstellt wurden (interactionData), übergeben werden (siehe Abschnitt 5.1.4). Ebenso optional sind eine Liste von Musterlösungen zur Aufgabe (solutionModel), die Angabe der Modellierungssprache (modelingLanguage) und der textuellen Beschreibung des zu modellierenden Szenarios (scenario), die für die Aufgabe festgelegt wurden. Die individuellen Parameter für die Bewertungsdienste (siehe auch Abschnitt 5.2.1) können in Form von Schlüssel-Wert-Paaren (keyValue), als Text- oder Binärdatei (textFile bzw. byteFile) in das DTO integriert werden.

Das Ausgabe-DTO, welches vom Bewertungsdienst erzeugt wird, enthält primär das

²⁰ <https://developers.google.com/protocol-buffers/docs/proto3>

²¹ Das heißt, wenn dies im Bewertungsschema für die Aufgabe für den jeweiligen Bewertungsdienst so konfiguriert wurde (siehe Abschnitt 5.2.1).

Bewertungsergebnis (`gradingResult`). Im Fall von »Modell erstellen«-Aufgaben handelt es sich dabei um ein relatives Ergebnis im Intervall $[0, 1]$. Ebenso können optional individuelle Feedback-Nachrichten (`feedback`) zur studentischen Einreichung angehängt werden. Feedback-Nachrichten enthalten jeweils eine durch den Bewertungsdienst frei wählbare Kategorie (`category`) anhand derer Nachrichten bei der Anzeige in den Frontend-Komponenten gruppiert werden können, den eigentlichen Inhalt der Feedback-Nachricht (`content`) sowie einen Beurteilungstyp (`ValuationType`), der angibt, ob das Feedback positiv, negativ, neutral oder anders eingestuft und in den Frontend-Komponenten mit entsprechenden Signalfarben dargestellt werden kann. Damit Feedback-Nachrichten zur einfacheren Zuordnung direkt in die grafische Darstellung des studentischen Modells integriert werden können, ist die Angabe der ID des betreffenden Modellelements (`positionId`) oder der Koordinaten im Modell (`positionX` und `positionY`) erforderlich.

Zusätzlich liefert das Ausgabe-DTO Log-Informationen zur Ausführung des Bewertungsdienstes (`logMessage`), eine Auskunft darüber ob die Bewertung erfolgreich war oder nicht (`jobSuccessful`) sowie die Ausführungszeit der automatisierten Bewertung in Millisekunden (`executionTimeMilis`) zurück.

Wie in Abschnitt 5.1.3 bereits beschrieben, bietet die Verwendung von DTOs in Form von Protocol Buffers den Vorteil, dass diese von den gängigsten²² Programmiersprachen unterstützt werden. Dazu werden die hier vorgestellten Spezifikationsdateien (siehe Abb. 5.6) mit einem Compiler (`protoc`) in entsprechende sprachspezifische Programmpakete übersetzt. Diese können dann in die Plattform-Komponenten eingebunden werden, um DTOs zu lesen und zu erzeugen.

5.2.3. Bewertungsinitialisierer

Der Bewertungsinitialisierer ist eine Backend-Komponente der KEA-Mod-Plattform und grob für die folgenden Aufgaben zuständig:

- (i) Entgegennehmen von Aufträgen zur Bewertung einer studentischen Einreichung bestimmter Aufgabentypen
- (ii) Erzeugung individueller Eingabe-DTOs auf Grundlage der im Auftrag übermittelten Informationen

²² Siehe Abschnitt 5.1.3 für eine Auflistung der unterstützten Programmiersprachen.

5. E-Assessment für das Gebiet der grafischen Modellierung

- (iii) Übergeben der Eingabe-DTOs für die beteiligten Bewertungsdienste an die Bewertungsqueue

Bezüglich Punkt (i) stellt die Komponente eine Schnittstelle zur Verfügung, die REST-Anfragen für die Bewertung der in der Plattform verfügbaren Aufgabentypen über entsprechende Endpunkte annehmen kann. Über diese Endpunkte können Aufträge zur Bewertung einer studentischen Einreichung von der REST-Schnittstelle an den Bewertungsinitialisierer übergeben werden. Diese Anfragen werden in der KEA-Mod-Plattform zu unterschiedlichen Zeiten durch die Frontend-Komponenten ausgelöst (siehe Abschnitt 5.2.5).

Die eingehenden Aufträge von der REST-Schnittstelle enthalten zunächst alle mit der Bewertung verknüpften Informationen, die in der Datenbank hinterlegt sind – mit Ausnahme von Kennungen der beteiligten Nutzenden. Neben der studentischen Einreichung selbst sind dies z. B. Angaben zur betreffenden Aufgabe und das dazugehörige Bewertungsschema. Um für Punkt (ii) individuelle Eingabe-DTOs für die beteiligten Bewertungsdienste erstellen zu können, muss zunächst die Baumstruktur des Bewertungsschemas geparkt werden, um an die Bewertungsblätter an den Blattknoten zu gelangen. Für die Traversierung des Baumes kommt eine rekursive Funktion zur Tiefensuche zum Einsatz. Dabei wird vom Wurzelknoten ausgehend entlang des Suchpfades die Gewichtung des jeweils besuchten Knotens aktualisiert, um schlussendlich den Anteil des Ergebnisses eines Bewertungsdienstes (Blattknoten) am Gesamtergebnis zu erhalten²³. Für alle Blattknoten bzw. Bewertungsdienste werden nun die entsprechenden DTOs erzeugt und mit den benötigten Inhalten gefüllt (siehe Abschnitt 5.2.2). Anschließend wird jeweils gemäß Punkt (iii) eine Verbindung zur Bewertungsqueue als Nachrichtenproduzent²⁴ aufgebaut und das entsprechende DTO an das individuelle Eingabe-Topic des dazugehörigen Bewertungsdienstes übergeben.

5.2.4. Bewertungsverarbeiter

Die primäre Aufgabe des Bewertungsverarbeiters im Backend der KEA-Mod-Plattform ist es, die Ergebnisse aus den Ausgabe-DTO der Bewertungsdienste entsprechend ihrer Gewichtung auf das Gesamtergebnis zu übertragen und in der Datenbank zu spei-

²³ Beispielsweise beträgt in Abb. 5.4 (siehe Abschnitt 5.2.1) der Anteil des Ergebnisses von Bewertungsdienst 4 am Zwischenschema B 50%, am Gesamtergebnis am Wurzelschema jedoch 15%, da Zwischenschema B wiederum mit einer Gewichtung von 30% am Wurzelschema beteiligt ist.

²⁴ Client-Verbindungen mit der Bewertungsqueue (Apache Kafka Server) sind entweder in Form eines Nachrichtenproduzenten (producer) oder Nachrichtenkonsumenten (consumer) möglich.

chern. Dazu baut diese Komponente eine persistente Verbindung zur Bewertungsqueue als Nachrichtenkonsument auf und überwacht das generelle Ausgabe-Topic kontinuierlich auf das Eingehen neuer Nachrichten von den Bewertungsdiensten.

Für jede eingehende Nachricht werden auf Basis der aus dem Ausgabe-DTO ausgelesenen Informationen folgende Schritte für die betreffende Bewertung ausgeführt:

- (i) Erstellen einer Teilbewertung (siehe Datenmodell in Abb. 5.2 in Abschnitt 5.1.2) für die betreffende Bewertung in der Datenbank.
- (ii) Falls vorhanden, werden Feedback-Nachrichten zu dieser Teilbewertung in der Datenbank erstellt (siehe ebenfalls das Datenmodell in Abb. 5.2).
- (iii) Aktualisierung des Gesamtergebnisses der Bewertung in der Datenbank, indem das Ergebnis der Teilbewertung entsprechend seiner Gewichtung mit in das vorliegende Gesamtergebnis einfließt.
- (iv) Prüfung, ob alle Teilbewertungen der Bewertungsblätter, die im Bewertungsschema aufgeführt werden, in der Datenbank eingetragen wurden. Falls ja, wird das Gesamtergebnis der Bewertung als vollständig deklariert.

Alle aufgeführten Schritte (i) bis (iv) werden vom Bewertungsverarbeiter über entsprechende Anfragen an die REST-Schnittstelle der KEA-Mod-Plattform abgewickelt. Diese stellt passende Endpunkte für die benötigten Zugriffsoperationen auf die Datenbank bereit.

5.2.5. Beginn der automatisierten Bewertung

Gemäß der Anforderungserhebung (siehe Abschnitt 5.1) ist es vorgesehen, dass der Einsatzkontext der KEA-Mod-Plattform möglichst flexibel auf die jeweiligen Anforderungen der betreffenden Lehrveranstaltung ausgerichtet werden können soll. Dies deckt sich auch mit den Ergebnissen aus der systematischen Literaturrecherche zum Zeitpunkt und Kontext des Einsatzes von E-Assessment-Systemen (siehe SLR-FF5 im Abschnitt 3.3.2). Diese legen nahe, dass es sinnvoll ist, generell gehaltene Systeme mit vielfältigen Konfigurationsoptionen anstelle spezialisierter Systeme für einzelne spezifische Einsatzkontexte (z. B. entweder nur formatives oder nur summatives Assessment) zu entwickeln. Daher kann der Zeitpunkt für den Beginn der automatisierten Bewertung individuell gesteuert werden. Zum Stand 01/2024 können über die Lehrenden UI entsprechende Einstellungen anhand von Konfigurationsoptionen

5. E-Assessment für das Gebiet der grafischen Modellierung

vorgenommen werden, so dass sich daraus die folgenden möglichen Szenarien für den Einsatz ergeben (siehe auch [U122b]):

- (i) Die automatisierte Bewertung wird bei jedem Zwischenspeichern einer Aufgabe in der Studierenden UI gestartet. Somit ist es Studierenden während der Bearbeitung möglich, ihre Lösung zur Aufgabe auf der Grundlage des Bewertungsergebnisses und der Feedback-Nachrichten sukzessive weiterzuentwickeln (sog. »multiple try feedback«).
- (ii) Die automatisierte Bewertung wird nach der Abgabe einer kompletten Aufgabenreihe gestartet. Das bedeutet, dass Studierende zwar ohne Zeitverzögerung ein Bewertungsergebnis sowie Feedback-Nachrichten erhalten, allerdings ihre Aufgaben hinterher nicht mehr überarbeiten können (sog. »outcome feedback«).
- (iii) Die automatisierte Bewertung wird durch Lehrende zu einem beliebigen Zeitpunkt nach der Einreichung einer Aufgabenreihe manuell gestartet. Das Bewertungsergebnis und das generierte Feedback können so vor der Veröffentlichung an die Studierenden kontrolliert und angepasst werden, ebenso kann individuelles Feedback ergänzt werden. Es handelt sich ebenfalls um ein »outcome feedback«, das im Gegensatz zu Szenario (ii) allerdings nicht vollständig automatisiert sondern semi-automatisiert (siehe Abschnitt 2.1.3) erstellt wird.

Die Szenarien (i) und (ii) eignen sich besonders für Zwecke des (Selbst-)Lernens, z.B. in einem formativen Einsatzkontext. Für den Einsatz zur Leistungsfeststellung in einem summativen Kontext (z.B. bei elektronischen Prüfungen), bei welcher aus prüfungsrechtlicher Sicht die manuelle Kontrolle durch Lehrende erforderlich ist, kommt unabhängig von der Zuverlässigkeit der beteiligten Bewertungsdienste ausschließlich das semi-automatisierte Assessment in Szenario (iii) in Frage.

5.3. Bewertungsdienste

Während der Projektlaufzeit des KEA-Mod-Projekts sind verschiedene Bewertungsdienste für die Bewertung von Aufgaben vom Typ »Modell erstellen« entstanden (siehe Tab. 5.1). Vier dieser Bewertungsdienste, die sich auf Petri-Netzen anwenden lassen, wurden im Rahmen dieser Dissertation entwickelt und in die KEA-Mod-Plattform integriert. Für Petri-Netze existieren bislang keine Ansätze zur automatisierten Bewertung im Hochschulkontext, die dem hier behandelten Aufgabentyp

Tab. 5.1.: Überblick über die Bewertungsdienste für den Aufgabentyp »Modell erstellen« in der KEA-Mod-Plattform. Fett markierte Bewertungsdienste wurden im Rahmen dieser Dissertation entwickelt.

Name	Austauschformat	Modellierungssprache	Beschriftungssprache	Programmiersprache
BPMN Semantic Checker	KEA	BPMN	Englisch	Java
BPMN Syntax Checker	KEA	BPMN	Alle	Java
EPC Grading Services	EPML	EPK	Englisch	Python
ER Checker	KEA	ER	Alle	Java/Kotlin
Petri Net Syntax Checker	KEA	Petri-Netze	Alle	Java
Petri Net Model Matcher	KEA	Petri-Netze	Englisch	Java
Petri Net Reachability Checker	KEA	Petri-Netze	Alle	Java/Kotlin
UML Checker	XMI	UML-KD	Alle	Java
Universal Pragmatic Checker	KEA	Alle	Alle	Java

»Modell erstellen« und den dazugehörigen Kompetenzfacetten des Kompetenzmodells entsprechen. In der Literaturstudie (siehe Abschnitt 3.3.2) wurden lediglich zwei Publikationen von WESTERGAARD et al. [WFS12; WFS13] identifiziert. Der dort vorgestellte Ansatz arbeitet allerdings auf gefärbten Petri-Netzen und prüft von Studierenden durchgeführte Veränderungen an einem bestehenden Modell sowie dynamische Eigenschaften anhand von Regeln, die in einer Beschreibungssprache ähnlich der linearen temporalen Logik (LTL) definiert wurden.

Alle Bewertungsdienste zum Aufgabentyp »Modell erstellen« in der KEA-Mod-Plattform erhalten studentische Modelle in Form des KEA-Austauschformats (siehe Abschnitt 5.1.4), jedoch arbeiten nicht alle direkt auf diesem sondern konvertieren es zuvor in ein sprachspezifisches Austauschformat (siehe Tab. 5.1). Dies geschieht aus dem Grund, dass diese Dienste auf bereits vor Projektbeginn bestehenden Anwendungen basieren und ohne interne Modifikationen an die Plattform angebunden werden sollten. Der überwiegende Anteil an Bewertungsdiensten ist sprachspezifisch und konzentriert sich auf die Überprüfung von Modellen in einer bestimmten Modellierungssprache. Die derzeit einzige Ausnahme bildet der Universal Pragmatic Checker, dessen Metriken für Verständlichkeit und Lesbarkeit prinzipiell auf allen graphbasierten Diagrammen Anwendung finden können (siehe Abschnitt 5.3.4). Einige Bewertungsdienste interpretieren die Beschriftungen von Modellen und wenden dabei NLP-Verfahren für die englische Sprache an, so dass diese nur auf Modellen angewendet werden können, deren Beschriftungen in englischer Sprache verfasst wurden. Bis auf die EPC Grading Services (Python) sind alle Bewertungsdienste in

5. E-Assessment für das Gebiet der grafischen Modellierung

der Programmiersprache Java geschrieben, der Petri Net Reachability Checker und der ER Checker enthalten zusätzlich zu Java kompatiblen Kotlin-Quelltext.

Um einen Bewertungsdienst in die KEA-Mod-Plattform zu integrieren, muss sich dieser über eine persistente Verbindung als Nachrichtenkonsument an der Bewertungsqueue anmelden und dort kontinuierlich auf eingehende Nachrichten in seinem individuellen Eingabe-Topic lauschen. Über die Nachrichten erhält er in Form von Eingabe-DTO Bewertungsaufträge für studentische Einreichungen. Anhand der Informationen im Eingabe-DTO (siehe Abschnitt 5.2.2) kann die Bewertung durchgeführt und das Ergebnis sowie gegebenenfalls auch Feedback-Nachrichten in einem Ausgabe-DTO (siehe Abschnitt 5.2.2) gebündelt werden, welches über das generelle Ausgabe-Topic in der Bewertungsqueue an den Bewertungsverarbeiter weitergereicht wird (siehe Abschnitt 5.2.4). Das hier skizzierte Verfahren findet gleichermaßen für alle Bewertungsdienste Anwendung.

Der Fokus der folgenden Abschnitte liegt auf der Beschreibung der Kernfunktionen der entwickelten Bewertungsdienste, nämlich der Berechnung eines (Teil-)Bewertungsergebnisses sowie der Generierung von Feedback-Nachrichten für die untersuchten Aspekte von studentischen Modellen. Abschnitt 5.3.1 stellt das Grundgerüst von drei der vier entwickelten Bewertungsdienste vor, die jeweils in den Abschnitten 5.3.2, 5.3.3 und 5.3.4 beschrieben werden. Diese orientieren sich an den in Abschnitt 4.3 entwickelten Rubrics für Aufgaben zum Typ »Modell erstellen« und deren wesentlichen Kriterien Syntax, Semantik und Pragmatik. Als Ergänzung wird in Abschnitt 5.3.5 ein weiterer Bewertungsdienst zur Prüfung dynamischer Eigenschaften von Petri-Netzen vorgestellt.

5.3.1. Grundgerüst

Die in den folgenden Abschnitten 5.3.2, 5.3.3 und 5.3.4 vorgestellten Bewertungsdienste basieren auf einem gemeinsamen Grundgerüst, welches die folgenden Basisfunktionen zur Verfügung stellt:

- Überführung des erhaltenen studentischen Modells in eine interne Datenstruktur zur Repräsentation eines graphbasierten Diagramms
- Aufruf der gewünschten Methoden zur Überprüfung des studentischen Modells gemäß der übermittelten Konfiguration

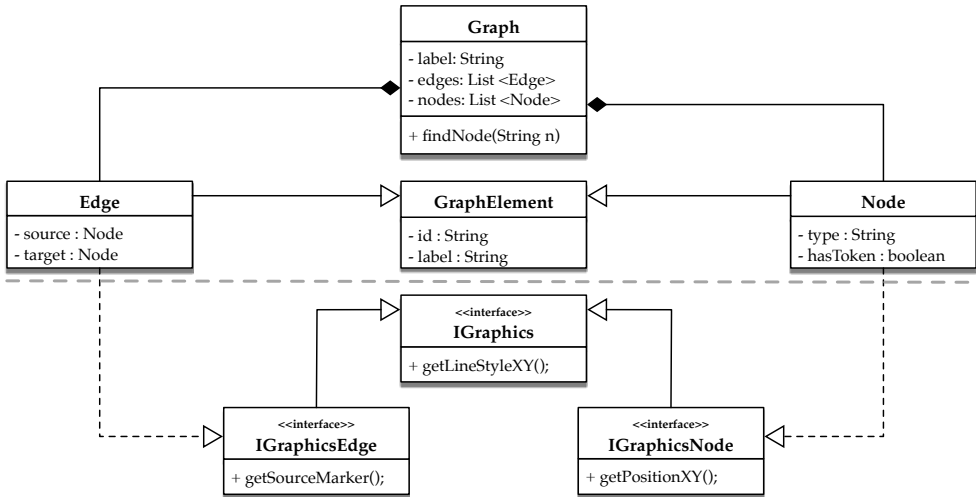


Abb. 5.7.: UML-Klassendiagramm zur Datenstruktur eines graphbasierten Diagramms. Obere Hälfte (geteilt anhand der grau gestrichelten Linie): Graphstruktur, untere Hälfte: Informationen zur grafischen Darstellung.

- Berechnung einer Gesamtbewertung anhand der Teilergebnisse der angewendeten Methoden
- Generierung von Feedback-Nachrichten auf Basis der Teilergebnisse und/oder der Gesamtbewertung

Beim Eintreffen einer Bewertungsanfrage über die Bewertungsqueue wird das übermittelte studentische Modell im KEA-Format (siehe Abschnitt 5.1.4) anhand eines JSON-Parsers in die interne Datenstruktur (siehe Abb. 5.7) überführt. Dieser verarbeitet schrittweise das studentische Modell und fügt dabei die vorhandenen Knoten und Kanten in eine Graphstruktur ein. Dabei werden auch deren Attribute eingebunden. Entsprechend der vorgestellten Definition eines Graphen $G = (V, E)$ (siehe Def. 2.15) verwaltet die Klasse `Graph` Listen von Knoten (V : `Node`) und Kanten (E : `Edge`). Zur Arbeit mit dem Graph werden Basisfunktionen zur Verfügung gestellt, z. B. die Methode `findNode(String n)` um einen Knoten anhand seiner ID zu suchen und zurückzugeben. Die Klassen `Edge` und `Node` erben gemeinsame Attribute von der Klasse `GraphElement` und implementieren das jeweilige Interface `IGraphicsEdge` bzw. `IGraphicsNode`. Das Interface `IGraphics` ist die Oberklasse dieser beiden Interfaces. Mit den drei Interfaces werden die Informationen zur grafischen Darstellung des studentischen Modells in der internen Datenstruktur ergänzt, wie z. B. die Position eines Knotens (`getPositionXY()`) oder die Form des Endpunktes

5. E-Assessment für das Gebiet der grafischen Modellierung

einer Kante (`getTargetMarker()`). Mit dieser Datenstruktur werden alle Informationen aus dem KEA-Format (siehe 5.1.4) abgebildet, die für die entwickelten Bewertungsdienste benötigt wurden. Alle Methoden der Bewertungsdienste erhalten das studentische Modell in Form eines `Graph`-Objekts als Eingabe und können auf dieser Basis die Analyse entsprechender Kriterien durchführen.

Bei den entwickelten Bewertungsdiensten wurden zwei verschiedene Typen von Methoden eingesetzt: Checks und Metriken. Mit einer Methode vom Typ **Check** wird das Vorliegen gewünschter (bzw. die Abwesenheit unerwünschter) Eigenschaften untersucht und als direktes Ergebnis $R_b(g)$ ²⁵ der Anwendung auf einem Graphen g ein Wahrheitswert zurückgeliefert. Darüber hinaus kann gegebenenfalls eine Fehlermenge $R_f(g)$ zurückgegeben werden, welche diejenigen Elemente des Graphen g enthält, die die vom Check geprüfte Eigenschaft nicht erfüllen. Bei einer Methode vom Typ **Metrik** ist das direkte Ergebnis $R_d(g)$ ein Wert im Intervall $[0, 1]$, wobei 0 dem schlechtesten und 1 dem besten möglichen Resultat entspricht (siehe Abschnitt 3.2.3).

Bei beiden Typen von Methoden kann es vorkommen, dass eine Methode sich nicht sinnvoll anwenden lässt, da spezifische Voraussetzungen nicht vorliegen. Beispielsweise kann nicht geprüft werden, ob die Knoten im Modell einem bestimmten Typ entsprechen, wenn das Modell überhaupt keine Knoten enthält. Um in derartigen Randfällen die Anwendbarkeit einer Methode bei der Bewertung berücksichtigen und entsprechend angepasste Feedback-Nachrichten erstellen zu können, wird beim Aufruf einer Methode zusätzlich mit einem Wahrheitswert die Anwendbarkeit $R_a(g)$ auf einem Graphen g festgehalten.

Zur Berechnung eines Gesamtergebnisses $G'(g)$ für einen Graphen g müssen alle Teilergebnisse der eingesetzten Methoden eines Bewertungsdienstes berücksichtigt werden. Bei der Bewertungsstrategie zur einzelnen Gewichtung von Methoden werden dafür die jeweiligen Produkte aus dem Teilergebnis $R_{e_m}(g)$ einer beteiligten Methode m und deren individuellem Gewicht $C_{w_m} \in [0, 1]$ ²⁶ zu einem Gesamtergebnis $G'(g) \in [0, 1]$ aufsummiert (gewichtete Summe).

$$G'(g) = \sum_m R_{e_m}(g) \cdot C_{w_m}$$

²⁵ Die Bedeutung des Symbols $R_b(g)$ und der anderen in den folgenden Abschnitten verwendeten Symbole ist im Symbolverzeichnis auf Seite xii dieser Dissertation zur Übersicht dargestellt.

²⁶ Das individuelle Gewicht C_w wird anhand der in der Bewertungsanfrage an den Bewertungsdienst übergebenen Konfigurationsparameter bestimmt.

Bei der Wahl individueller Gewichte wird stets vorausgesetzt, dass jedes Gewicht ≥ 0 ist und die Summe aller Gewichte 1 ergibt, um sicherzustellen, dass sich das Gesamtergebnis weiterhin im Intervall $[0, 1]$ befindet:

$$C_{w_m} \geq 0 \wedge \sum_m C_{w_m} = 1.$$

Da das direkte Ergebnis $R_b(g)$ bei Methoden des Typs Check ein Wahrheitswert ist, muss dieses zuvor in das Intervall $[0, 1]$ überführt werden, um als Teilergebnis $R_e(g)$ in das Gesamtergebnis $G'(g)$ einzufließen. Prüft ein Check eine erforderliche Eigenschaft, ohne dabei eine Fehlermenge $R_f(g)$ zu erstellen, gilt:

$$R_e(g) = \begin{cases} 1, & \text{falls } R_b(g) = \text{wahr,} \\ 0 & \text{sonst.} \end{cases}$$

Basiert der Check auf der Identifikation bestimmter Fehler, die in der Fehlermenge $R_f(g)$ festgehalten werden, so kann die Fehlerzahl zur Bildung einer entsprechenden Abstufung verwendet werden. Wie in Abschnitt 3.2.3 skizziert, gilt bei der Fehlerzahl das Optimierungsziel der Minimierung, so dass hier die Überföhrungsfunktion $s = 1 - \frac{a}{b}$ eingesetzt werden kann. Hierbei entspricht das absolute Ergebnis a der Fehlerzahl $|R_f(g)|$, die individuelle Bezugsgröße b sei als obere Grenze $R_o(g)$ festgelegt.

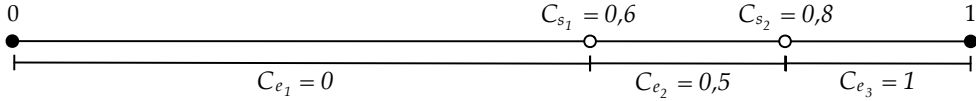
$$R_e(m) = \begin{cases} 1 - \frac{|R_f(g)|}{R_o(g)}, & \text{falls } |R_f(g)| \leq R_o(g), \\ 0 & \text{sonst.} \end{cases}$$

$R_o(g)$ kann anhand von C_o individuell auf einen gewünschten Wert ≥ 1 gesetzt werden. Möglich ist auch, $R_o(g)$ durch die Methode selbst bestimmen zu lassen, wenn bestimmte Grapheigenschaften berücksichtigt werden sollen (z.B. die Größe des studentischen Modells). Die obere Grenze $R_o(g)$ legt somit die maximale Anzahl möglicher bzw. zu berücksichtigender Fehler fest, die zur Verschlechterung des Bewertungsergebnisses (bzw. zum Punktabzug) führen können. Damit keine negativen Ergebnisse entstehen können, wenn die Fehlerzahl größer als die obere Grenze ist, d.h. $|R_f(g)| > R_o(g)$, wurde dieser Fall in der Formel entsprechend berücksichtigt.

Bei Metriken befindet sich das Ergebnis $R_d(g)$ bereits im Intervall $[0, 1]$ und kann direkt als Teilergebnis $R_e(g)$ für die Berechnung des Gesamtergebnisses weiterver-

5. E-Assessment für das Gebiet der grafischen Modellierung

wendet werden ($R_e(g) = R_d(g)$). Um eine Einstufung analog zu den Rubrics (siehe Abschnitt 4.3) zu erhalten, kann das Ergebnis $R_d(g)$ aber auch je nach Wertebereich auf eine diskrete Wertemenge aus $[0, 1]$ abgebildet werden. Werden für eine Metrik eine Menge von Schwellenwerten C_s und für die daraus resultierenden Wertebereiche eine Menge von dazugehörigen Ergebniswerten C_e angegeben, werden diese zur Bestimmung von $R_e(g)$ herangezogen. Seien $C_s = \{0.6, 0.8\}$ und $C_e = \{0, 0.5, 1\}$, wie folgt illustriert.



Somit wird beispielsweise das Ergebnis $R_d(g) = 0,8694010$ auf das Teilergebnis $R_e(g) = C_{e_1} = 1$ abgebildet.

Wenn bei der eingangs vorgestellten Bewertungsstrategie der einzelnen Gewichtung sehr viele Checks gleichzeitig zum Einsatz kommen, ist der jeweilige Anteil am Gesamtergebnis $G'(g)$ des Bewertungsdienstes zwangsläufig gering²⁷. Zudem kann es eine Reihe von Checks geben, die sich auf einen gemeinsamen Aspekt (z.B. Marken oder Kanten) beziehen. Daher wurde eine optionale gruppenweise Betrachtung ergänzt. So können Check-Methoden, die eine Fehlermenge erzeugen, gruppiert werden. Innerhalb einer solchen Gruppe kann eine methodenübergreifende Betrachtung der Fehlermengen erfolgen. Das Gesamtergebnis $R_{e_k}(g)$ einer solchen Gruppe k bestehend aus einer Menge von Checks m wird auf Basis der oberen Grenze C_{o_k} der Gruppe wie folgt berechnet:

$$R_{e_k}(g) = \begin{cases} 1 - \frac{\sum_{m \in k} |R_{f_m}(g)|}{C_{o_k}}, & \text{falls } \sum_{m \in k} |R_{f_m}(g)| \leq C_{o_k}, \\ 0 & \text{sonst.} \end{cases}$$

Die Methodenergebnisse $R_{e_m}(g)$ eines Bewertungsdienstes können nun wahlweise einzeln gewichtet werden (siehe G') oder im Fall von Checks mit Fehlermenge auch über ein Gruppenergebnis $R_{e_k}(g)$ mit der Gewichtung der Gruppe C_{w_k} in das

²⁷ Sei beispielsweise ein Bewertungsdienst laut Bewertungsschema für 4 Punkte des Ergebnisses zur studentischen Einreichung verantwortlich und es werden Check-Methoden zu 16 möglichen Fehlertypen verwendet. Dann führt ein einzelner Fehler bei einer der Methoden bei Gleichgewichtung zu maximal 0,25 Punkten Abzug von diesen 4 Punkten, wenn $R_o = 1$ für diese Methode gilt. Ist R_o höher, wird der Abzug sogar noch geringer, z. B. 0,125 Punkte bei $R_o = 2$.

Gesamtergebnis einfließen. Das so kombinierte Gesamtergebnis $G(g)$ setzt sich wie folgt zusammen:

$$G(g) = \sum_m R_{e_m}(g) \cdot C_{w_m} + \sum_k R_{e_k}(g) \cdot C_{w_k}$$

Auch hier wird bei der Wahl der Gewichtungen folgendes vorausgesetzt:

$$C_{w_m}, C_{w_k} \geq 0 \wedge \sum_m C_{w_m} + \sum_k C_{w_k} = 1.$$

Anhand der Bewertungsstrategie beim Gesamtergebnis $G(g)$ können im Extremfall auch alle Methoden vom Typ Check eines Bewertungsdienstes zu einer einzelnen Gruppe zusammengesetzt werden, z.B. wenn die Fehler, die in den verwendeten Methoden identifiziert werden, ohne weitere Differenzierung in die Bewertung einfließen sollen. Das hieße, jeder Fehler, unabhängig von seiner Art führt zu einem Abzug, bis die festgelegte obere Grenze erreicht wurde.

Feedback-Nachrichten werden unabhängig von der gewählten Bewertungsstrategie auf Basis der Ergebnisse $R_d(g)$, $R_b(g)$, $R_f(g)$ und $R_a(g)$ der eingesetzten Methoden erzeugt. Für alle Methoden ist es möglich, individuelle Textvorlagen zu erfassen, die auf unterschiedliche Fälle zugeschnitten sind. Tab. 5.2 stellt einen Überblick über die verschiedenen Arten und mögliche Inhalte lernförderlicher Feedback-Nachrichten dar, die von den Bewertungsdiensten unterstützt werden. Daneben wird in der Spalte Typ auch der jeweilige Beurteilungstyp der Feedback-Nachricht (siehe Abschnitt 5.2.2) angegeben, der für die entsprechende Darstellung im Frontend verwendet wird (z. B. grün für positive, rot für negative und gelb für neutrale Feedback-Nachrichten). Die beiden Spalten Bedingung und Methode zeigen an, unter welchen Voraussetzungen und für welche Typen von Methoden die jeweilige Feedback-Nachricht erzeugt wird. Für Metriken ist es möglich, einen einzelnen Schwellenwert C_t festzulegen, der bestimmt, ob eine positive oder negative Feedback-Nachricht erzeugt werden soll.

Alle in diesem Abschnitt aufgeführten Parameter C für die Berechnung des Gesamtergebnisses und F für die Erzeugung von Feedback-Nachrichten können beim Aufruf der Bewertungsdienste über entsprechende Schlüssel-Wert-Paare im Bewertungsschema übergeben werden. Dadurch wird die Standardkonfiguration überschrieben und es werden nur die in der übermittelten Konfiguration ausgewählten Methoden angewendet. Ebenso werden nur diejenigen Feedback-Nachrichten erzeugt, für die dabei eine Textvorlage angegeben wurde. Somit lassen sich sowohl die Berechnung des Bewertungsergebnisses als auch die Arten und Inhalte der Feedback-Nachrichten

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.2.: Lernförderliche Feedback-Nachrichten in den Bewertungsdiensten

Nachricht	Möglicher Inhalt	Typ	Bedingung	Methode
F_{na}	Warnung, dass das Kriterium nicht überprüft werden konnte (und ggf. auch warum)	Neutral	$R_a(g) = \text{falsch}$	Check, Metrik
F_{pos}	Angabe dazu, dass das geprüfte Kriterium erfüllt wurde	Positiv	$R_b(g) = \text{wahr}$	Check
F_{neg}	Angabe dazu, dass das geprüfte Kriterium nicht erfüllt wurde, kann die Fehlerzahl $ R_f(g) $ an einem Platzhalter einfügen	Negativ	$R_b(g) = \text{falsch}$	Check
F_{id}	Fehlerbeschreibung, bezogen auf ein bestimmtes Modellelement $i \in R_f(g)$ zur integrierten grafischen Darstellung des Fehlers im Modell	Negativ	$R_b(g) = \text{falsch},$ $R_f(g) \neq \emptyset$	Check
F_{pos}	Angabe, dass das Metrikergebnis auf dem bzw. oberhalb des Schwellenwerts C_t liegt	Positiv	$R_d(g) \geq C_t$	Metrik
F_{neg}	Angabe, dass das Metrikergebnis unterhalb des Schwellenwerts C_t liegt	Negativ	$R_d(g) < C_t$	Metrik
F_{met}	Angabe zum Metrikergebnis, kann $R_d(g)$ an einem Platzhalter einfügen	Neutral	keine	Metrik

von Lehrenden auf umfassende Weise individuell steuern.

5.3.2. Syntax-Checker

Der Syntax-Checker (Projektname: *Petri Net Syntax Checker*) bezieht sich auf die Prüfung der folgenden Kompetenzfacette:

MB 2.04: Die Lernenden können a) die syntaktischen Regeln und b) die Notation der betrachteten Modellierungssprache/n korrekt anwenden.²⁸

Diese ist im Kompetenzmodell auf der Inhaltsdimension »Modellbilden & -modifizieren« und auf der Prozessdimension »Anwenden & Übertragen« verortet (siehe Abschnitt 4.1). Die Prüfung dieser Kompetenzfacette ist anhand von Modellen, die mit herkömmlichen Modellierungswerkzeugen erstellt und im PNML-Austauschformat abgebildet wurden nicht durchführbar, da Verletzungen der Syntax von Petri-Netzen hierbei nicht auftreten können. Sie kann allerdings auf Modellen geprüft werden, die

²⁸ Mit Punkt b) ist hier gemeint, dass Lernende dazu in der Lage sind, die zur Modellierungssprache gehörigen Modellierungselemente einzusetzen.

auf Papier erstellt wurden. Dadurch, dass der Modellierungseditor keine Modellierungsunterstützung anbietet und das KEA-Format auch syntaktisch nicht korrekte Petri-Netze abbilden kann (siehe auch Abschnitt 5.1.4), wird die Prüfung dieser Kompetenzfacette in der KEA-Mod-Plattform ebenfalls ermöglicht.

Der Syntax-Checker stellt diesbezüglich verschiedene Methoden zur Verfügung, mit denen die Einhaltung syntaktischer Vorgaben für Petri-Netze überprüft werden kann. Die syntaktischen Vorgaben für Petri-Netze gehen aus den formalen Definitionen (siehe Abschnitt 2.2.4) hervor. Nach Def. 2.16 (zur besseren Veranschaulichung an dieser Stelle wiederholt) ist ein Netz ein Tripel (S, T, F) , für welches gilt:

- (i) S ist die Menge von Stellen;
- (ii) T ist die Menge von Transitionen;
- (iii) $F \subseteq (S \times T) \cup (T \times S)$ ist eine Flussrelation;
- (iv) $S \cap T = \emptyset$;
- (v) $S \cup T \neq \emptyset$.

Durch die Punkte (i) und (ii) werden die beiden Modellierungselemente Stelle und Transition definiert. Diese werden notationsgemäß (siehe Tab. 2.9) üblicherweise durch Knoten in Form von Kreisen (Stelle) bzw. Rechtecken (Transition) repräsentiert. Ein weiteres Modellierungselement in markierten Petri-Netzen, z.B. in Stellen-Transitionen-Netzen (siehe Def. 2.19) ist die Marke (siehe Tab. 2.9), dargestellt durch kleinere, schwarz ausgefüllte Kreise. Diese Darstellungsformen werden vom Modellierungseditor im KEA-Format (siehe Abschnitt 5.1.4) durch Knoten der drei Typen `kea.Circle`, `kea.Rectangle` und `kea.PNMarker` kodiert²⁹. Der Check **NodeTypesCorrect** prüft somit für jeden Knoten im übergebenen Graphen g , ob dessen Typ einem der drei genannten Typen entspricht. Ist dies der Fall, ist das direkte Ergebnis des Checks $R_b(g) = \text{wahr}$. Sind keine Knoten im Graphen vorhanden, ist der Check nicht anwendbar, d.h. $R_a(g) = \text{falsch}$. Jeder Knoten, der die geprüften Eigenschaften nicht erfüllt, wird der Fehlermenge $R_f(g)$ hinzugefügt, das Ergebnis des Checks ist dann $R_b(g) = \text{falsch}$.

Punkt (iii) bestimmt drei weitere Eigenschaften. Erstens müssen alle im Graph vorhandenen Kanten gerichtet sein, zweitens darf eine Kante nur Stellen und Transitionen miteinander verbinden und drittens muss jede Kante eine Stelle und eine Transition miteinander verbinden. Bezüglich Ersterem prüft der Check **EdgeTypesCorrect**

²⁹ Im PNML-Austauschformat werden Marken nicht als Knoten sondern als Eigenschaften von Stellen abgebildet. Da diese Handhabung die Verwendung von Marken einschränkt und im vorliegenden Kontext eine größtmögliche Modellierungsfreiheit angestrebt wird, wurden Marken stattdessen als Knoten umgesetzt, die sich frei positionieren lassen.

5. E-Assessment für das Gebiet der grafischen Modellierung

den verwendeten Kantentyp anhand der Formen der beiden Kantenendpunkte, die im KEA-Format abgebildet werden. Zulässige Formen sind Pfeilspitzen in verschiedenen Variationen (`ARROW`, `ARROW_EMPTY` und `ARROW_FILLED`) an einem der beiden Kantenendpunkte. Sind weiterhin an keinem oder an beiden Kantenendpunkten Formen hinterlegt, handelt es sich um ungerichtete bzw. um beidseitig gerichtete Kanten, die nicht zulässig sind und ebenso der Fehlermenge $R_f(g)$ hinzugefügt werden sowie zum Fehlschlagen des Checks führen. In Bezug auf die zweite Eigenschaft prüft der Check **DifferentTypesConnected** für alle Kanten, ob die Typen der Quell- und Zielknoten unterschiedlich sind. Hier wird bewusst darauf verzichtet, explizit auf die Verbindung von Stellen und Transitionen zu prüfen, da die Verwendung falscher Modellierungselemente bereits durch den Check `NodeTypesCorrect` berücksichtigt wird und es sonst gleichzeitig zu einem Abzug bzw. Folgefehler käme. Analog hierzu prüft der Check **EdgesConnected** bezogen auf die dritte Eigenschaft lediglich, ob jede Kante einen Quell- sowie Zielknoten besitzt.

Da im KEA-Format ein Knoten nur einen Typ besitzen kann, trifft Punkt (iv) implizit zu. Weiterhin wird vor Aufruf aller Methoden des Syntax-Checkers die Bedingung geprüft, dass der Graph mindestens einen Knoten besitzt, wodurch Punkt (v) abgedeckt wird.

Wie mit Bezug auf den Check `NodeTypesCorrect` zuvor beschrieben, sind auch Knoten vom Typ Marke zulässig. Für diesen Knotentyp müssen zusätzliche Anforderungen überprüft werden, die z.B. im Kontext von Stellen-Transitionen-Netzen (siehe Def. 2.19) definiert werden. Demnach wird durch eine Markierung M jeder Stelle eine Anzahl von Marken zugewiesen. In der visuellen Repräsentation liegen Marken dabei als einzelne kleinere schwarz ausgefüllte Kreise innerhalb von Knoten des Typs Stelle (siehe Tab. 2.9). Im Modellierungseditor der KEA-Mod-Plattform können Marken allerdings auch außerhalb von Stellen oder anderen Knoten positioniert werden. Um für die folgenden zwei Fälle differenzierte Feedback-Nachrichten erzeugen zu können, wird durch den Check **TokensInNodes** zunächst für jede Marke geprüft, ob sie sich innerhalb eines Knotens und durch den Check **TokensInPlaces**, ob sie sich innerhalb eines Knotens vom Typ Stelle befindet. Dabei wird die Eigenschaft des KEA-Formats ausgenutzt, über das Feld `parent` denjenigen Knoten zu referenzieren, auf dem ein weiteres Modellierungselement (d.h. Knoten oder Kante) abgelegt wurde. Da sich Marken im Modellierungseditor wie alle anderen Knoten über Kanten mit weiteren Knoten verbinden lassen, wird dieser Fall durch den Check **TokensUnconnected** abgedeckt. Dieser betrachtet alle Kanten und prüft, ob Quell- und Zielknoten nicht vom Typ Marke sind.

Die bis zu dieser Stelle beschriebenen Checks beziehen sich allesamt auf einzelne Modellelemente, d.h. Knoten und Kanten des Graphen werden betrachtet und dabei jeweils in der Fehlermenge $R_f(g)$ die abweichenden Modellelemente gesammelt. Eine Ausnahme hiervon bildet der Check **WorkflowNet**, der den Graphen als Ganzes auf die Einhaltung der Workflow-Netz-Eigenschaften überprüft. Für Workflow-Netze werden zusätzliche statische Eigenschaften im Hinblick auf die Struktur eines Netzes N festgelegt, die insbesondere im Kontext der Modellierung von Geschäftsprozessen häufig vorausgesetzt werden. Aus diesem Grund soll die Überprüfung dieser Eigenschaft im Rahmen des Syntax-Checkers ermöglicht werden. Die beiden zu überprüfenden Eigenschaften lauten gemäß Def. 2.24:

- (i) Das Netz besitzt genau eine Stelle i mit $\bullet i = \emptyset$ (Quelle) und genau eine Stelle o mit $o\bullet = \emptyset$ (Senke).
- (ii) Wenn eine Transition t^* zu N hinzugefügt wird, welche die Stellen o und i miteinander verbindet, ist das daraus resultierende Netz stark zusammenhängend.

Bezüglich Eigenschaft (i) werden alle Knoten (außer Knoten vom Typ Marke) betrachtet und untersucht, ob der Graph Kanten enthält, deren Quell- bzw. Zielknoten dem betrachteten Knoten n entsprechen. Kann die Existenz mindestens einer derartigen Kante nachgewiesen werden, so handelt es sich beim betreffenden Knoten nicht um eine Quelle bzw. Senke, da $\bullet n \neq \emptyset$ bzw. $n\bullet \neq \emptyset$ gilt, andernfalls ja. Hierbei möglicherweise auftretende Probleme, die mit entsprechenden Feedback-Nachrichten zurückgemeldet werden, sind:

- Es gibt nicht genau eine Quelle
- Es gibt nicht genau eine Senke
- Der Quellknoten ist nicht vom Typ Stelle
- Der Senkenknoten ist nicht vom Typ Stelle

Tritt eines der geschilderten Probleme auf, ist $R_b(g) = \text{falsch}$. Anderenfalls wird mit der Prüfung der Eigenschaft (ii) fortgefahren. Dazu wird zunächst im Graphen eine Transition t^* ergänzt, welche die identifizierte Senke mit der identifizierten Quelle verbindet. Anschließend wird geprüft, ob der Graph stark zusammenhängend ist. Auf Basis von Def. 2.18 (stark zusammenhängend) wird dabei für alle Paare aus Knoten x, y über alle Knoten im Graph (ohne Marken) mit $x \neq y$ untersucht, ob eine Kantenfolge $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)$ vom Knoten $v_1 = x$ zum Knoten $v_n = y$ existiert. Wird ein Knotenpaar identifiziert, für welches keine entsprechende

5. E-Assessment für das Gebiet der grafischen Modellierung

Kantenfolge existiert, schlägt der Check fehl und das betreffende Knotenpaar wird über eine entsprechende Feedback-Nachricht zurückgemeldet. Andernfalls sind beide Workflow-Netz-Eigenschaften erfüllt und $R_b(g) = \text{wahr}$.

Da die Workflow-Netz-Definition ein Netz voraussetzt, müssen strenggenommen auch Fehler der Checks bezüglich der Netz-Definition (NodeTypesCorrect, EdgeTypesCorrect, EdgesConnected, DifferentTypesConnected) zu einem Ergebnis $R_b(g) = \text{falsch}$ des Checks WorkflowNet führen. Dies kann wahlweise durch das Setzen des Konfigurationsparameters $C_x = \text{wahr}$ aktiviert werden und führt zwangsläufig zu einem Folgefehler. Wenn jedoch gewünscht wird, diesen Folgefehler nach Möglichkeit zu verhindern, kann der Parameter auch deaktiviert werden. Der Check WorkflowNet wird dann wie oben beschrieben ausgeführt, ohne die Verletzungen der Netz-Definition zu berücksichtigen. Dies führt je nach Fehlertyp (bzw. Check) zu folgenden Effekten:

- Falsche Knotentypen (NodeTypesCorrect) im Graph führen ausschließlich im Falle einer Quelle oder einer Senke zu einem Folgefehler.
- Kanten, die Knoten gleichen Typs verbinden (DifferentTypesConnected) spielen im Zusammenhang mit der Prüfung der Workflow-Netz-Eigenschaften (i) und (ii) keine Rolle und führen nicht zu einem Folgefehler.
- Falsche Kantentypen (EdgeTypesCorrect) führen dazu, dass die Flussrichtung der Kante nicht eindeutig ermittelt werden kann. In diesen Fällen wird die Flussrichtung gemäß Quelle und Ziel der Kante angenommen³⁰. Es besteht eine theoretische Wahrscheinlichkeit von 50%, dass die Flussrichtung dadurch im Sinne der erstellenden Person bestimmt wird³¹ und somit ein Folgefehler verhindert werden kann.
- Unverbundene Kanten (EdgesConnected) können in zwei Formen auftreten: Kanten, die mit keinem und Kanten, die mit nur einem Knoten verbunden sind. Erstere führen nicht zu einem Folgefehler, da sie für die Bestimmung der Eigenschaften (i) und (ii) nicht relevant sind. Zweitere führen jedoch dazu, dass die Anzahl von Quellen und Senken verfälscht wird und somit ein Folgefehler auftritt.

³⁰ Auch die Kantenenden von ungerichteten oder beidseitig gerichteten Kanten werden als Quelle und Ziel bezeichnet.

³¹ Unter der Annahme, dass bei der Modellierung Kanten häufig entlang der Flussrichtung von Quellknoten zum Zielknoten eingeführt werden, kann die Wahrscheinlichkeit tatsächlich auch höher sein. Denn es wird beim Einfügen im Modellierungseditor das zuerst gesetzte Ende einer Kante (Quelle) mit dem Quellknoten und das anschließend gesetzte Ende einer Kante (Ziel) mit dem Zielknoten verbunden.

Tab. 5.3.: Vom Syntax-Checker bereitgestellte Methoden zur Überprüfung syntaktischer Vorgaben für Petri-Netze

Methoden	Fehlermenge $R_f(g)$	Anwendbarkeit $R_a(g) = \text{wahr, wenn}$	Obere Grenze $R_o(g)$
NodeTypesCorrect	ja	Knoten vorhanden	Knotenanzahl im Graph g
EdgeTypesCorrect	ja	Kanten vorhanden	Kantenanzahl im Graph g
EdgesConnected	ja	Kanten vorhanden	Kantenanzahl im Graph g
DifferentTypesConnected	ja	Knoten und Kanten vorhanden	Anzahl Kanten im Graph g , die Quell- und Zielknoten besitzen
TokensInNodes	ja	Marken vorhanden	Markenanzahl im Graph g
TokensInPlaces	ja	Marken vorhanden	Markenanzahl im Graph g
TokensUnconnected	ja	Marken vorhanden	Markenanzahl im Graph g
WorkflowNet	nein	Knoten und Kanten vorhanden	-

Da Marken im Check WorkflowNet explizit von der Betrachtung ausgeschlossen werden, hat die fehlerhafte Verwendung von Marken (TokensInNodes, TokensInPlaces, TokensUnconnected) keine Auswirkungen auf das Ergebnis des Checks und verursacht keinen Folgefehler.

Tab. 5.3 zeigt einen Überblick über die im Syntax-Checker implementierten Methoden mit Angaben zu grundlegenden Eigenschaften, wie dem Vorhandensein einer Fehlermenge $R_f(g)$ und der Anwendbarkeit. Darüber hinaus wird die (standardmäßige) obere Grenze $R_o(g)$ angegeben, die von den Methoden zurückgegeben wird. Sofern gewünscht, kann diese obere Grenze verwendet werden, um Fehler an der maximal möglichen Anzahl ihres Auftretens zu messen und auf diese Weise in das Gesamtergebnis einfließen zu lassen. Sie kann aber auch über den Konfigurationsparameter C_o direkt auf den gewünschten Wert gesetzt werden.

Die Methoden des Syntax-Checkers sind im Hinblick auf die zu prüfende Kompetenzfacette zunächst »fragil« [Ga19b], da lediglich die Abwesenheit von syntaktischen Fehlern untersucht wird. Dies hat zur Folge, dass sehr einfache oder gar triviale studentische Modelle (z.B. ein Petri-Netz, das aus nur einer einzigen Stelle besteht) die volle Punktzahl vom Syntax-Checker erhalten können. Dies kann im Rahmen bestimmter Aufgaben durchaus gewünscht sein. Eine umfassende aussagekräftige Beurteilungsmöglichkeit in Bezug auf diese Kompetenzfacette besteht jedoch erst dann, wenn das studentische Modell einen gewissen Umfang besitzt. Zu diesem Zweck wurden Konfigurationsvariablen C_n und $C_{n,s}$ eingeführt. Ist $C_n = \text{wahr}$, so wird unter zutreffenden Bedingungen eine sogenannte »Nullifikation« [Ga19b] des

5. E-Assessment für das Gebiet der grafischen Modellierung

Gesamtergebnisses durchgeführt, d.h. dieses wird nach Aufruf aller Methoden auf $G = 0$ gesetzt und eine entsprechende Feedback-Nachricht ergänzt. Die konkreten Bedingungen für die Nullifikation werden durch die Anzahl an Knoten und Kanten des als Musterlösung bzw. Referenz übergebenen Graphen und auf Basis von C_{ns} bestimmt. Enthält der übergebene Graph beispielsweise 28 Knoten und 24 Kanten, so muss das zu prüfende studentische Modell bei $C_{ns} = 0,25$ mindestens 25%, also 7 Knoten und 6 Kanten enthalten. Eine sinnverwandte Forderung wird auch im zuvor entwickelten Rubric für die Bewertung von Aufgaben des Typs »Modell erstellen« (siehe Tab. 4.4) für das Kriterium Syntax angegeben. Dort wird die Punktzahl an das Kriterium Vollständigkeit geknüpft und eine volle Punktzahl nur gewährt, wenn das Modell $\geq 25\%$ der Aussagen abbildet. Die vorgeschlagene Einführung einer Abhängigkeit zwischen den Kriterien Syntax und Vollständigkeit (Semantik) wird jedoch als überflüssig erachtet. Um die hier untersuchte Kompetenzfacette bezüglich des Aspekts Syntax zu prüfen, sollte es ausreichend sein, wenn das zu betrachtende Modell einen entsprechenden Umfang besitzt. Schließlich ist es an dieser Stelle unerheblich, ob die Aussagen des Modells dabei einen vorgegebenen Sachverhalt abbilden³².

5.3.3. Semantik-Checker

Mit dem Semantik-Checker (Projektname: *Petri Net Model Matcher*) soll die folgende Kompetenzfacette des Kompetenzmodells (siehe Abschnitt 4.1) mit Bezug auf die semantische Qualität überprüft werden:

MB 4.04: Die Lernenden können ein Modell in Bezug auf einen Sachverhalt a) semantisch korrekt und b) vollständig erstellen und sich dabei c) auf relevante Modellinhalte beschränken (Prägnanz).

Nach der veröffentlichten Literaturstudie [Ul23a] (siehe Abschnitt 3.3.2) zur automatisierten Bewertung von Modellen wird in rund 72% der untersuchten Publikationen der Einsatz von Verfahren zum Vergleich des studentischen Modells mit einer (oder mehreren) Musterlösung(en) zum Zweck der inhaltlichen Überprüfung beschrieben. Der größte Vorteil dieser Art von Verfahren liegt für die Lehrenden darin, dass die Erstellung einer Musterlösung einen vergleichsweise geringen Aufwand bedeutet. Bei regelbasierten Verfahren (rund 38% der untersuchten Publikationen)

³² Vorstellbar wäre jedoch, eine gewisse Komplexität des Modells im Hinblick auf die Graphstruktur vorauszusetzen und in die Bedingung für die Nullifikation einzuflechten.

müssen für jede Aufgabe individuelle Regeln konfiguriert werden. Aus diesem Grund wurde im Forschungsvorhaben ein verwandter Ansatz aus dem Gebiet des Prozessmodell-Matching ausgewählt und implementiert. Dabei wurde die Grundidee des Triple-S-Ansatzes von CAYOGLU et al. [Ca13c; Ca14] und SCHOKNECHT [Sc18] aus der eigenen Forschungsgruppe (siehe Abschnitt 3.3.1) aufgegriffen und auf die vorliegende Problemstellung angepasst. Ein erster Prototyp wurde im Rahmen einer Abschlussarbeit³³ entworfen und in dieser Dissertation grundlegend überarbeitet.

Der Semantik-Checker verwendet als Eingabe beliebig viele (und mindestens eine) Musterlösungen in Form von Modellen im KEA-Format, die von Lehrenden für die jeweilige Aufgabe erstellt wurden. Dabei soll vorausgesetzt werden, dass die Musterlösungen die syntaktischen Anforderungen an Petri-Netze erfüllen (siehe Abschnitt 5.3.2). Die Musterlösungen werden nacheinander mit dem studentischen Modell verglichen und diejenige Musterlösung mit dem höchsten Gesamtergebnis ausgewählt. Diese wird als Basis für die Bestimmung der Abweichungen herangezogen. Dabei werden sowohl fehlende Modellelemente (d.h. Knoten und Kanten), die in der Musterlösung, nicht aber im studentischen Modell enthalten sind, als auch zusätzliche Modellelemente, die im studentischen Modell, aber nicht in der Musterlösung identifiziert werden konnten, bestimmt. Auf eine Abbildung der entsprechenden Fehler auf die drei in der Kompetenzfacette MB 4.04 genannten Kriterien a) Korrektheit, b) Vollständigkeit und c) Prägnanz wird verzichtet, da dies im vorliegenden Kontext nicht zielführend ist. Eine triviale Annahme wäre es, fehlende Modellelemente als Fehler in Bezug auf die Vollständigkeit und zusätzliche Modellelemente als Fehler in Bezug auf die Prägnanz zu interpretieren. Allerdings können sowohl fehlende als auch zusätzliche Modellelemente als Fehler in Bezug auf die Korrektheit gewertet werden. Gleichzeitig kann nicht davon ausgegangen werden, dass das Matching-Verfahren in allen Fällen zuverlässige Ergebnisse liefert. Somit würde ein Knoten im studentischen Modell, dessen tatsächliche Übereinstimmung mit einem Knoten in der Musterlösung nicht identifiziert werden konnte, gleichzeitig zu einem fehlenden und einem zusätzlichen Knoten führen.

In Abb. 5.8 ist ein Überblick über die wesentlichen Arbeitsschritte des Semantik-Checkers dargestellt. Bei der Berechnung der Knoten-Ähnlichkeit spielt der Vergleich der Beschriftungen eine tragende Rolle, da über diese ein direkter Bezug zu den Inhalten des abzubildenden Sachverhalts hergestellt wird. Da üblicherweise der Freiheitsgrad bei der Wahl der Beschriftungen im Rahmen einer Modellierungsaufgabe

³³ Christian Bitterwolf, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Semantische Analyse von Petri-Netzen mit Methoden des Natural Language Processing*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2021.

5. E-Assessment für das Gebiet der grafischen Modellierung



Abb. 5.8.: Überblick über die Arbeitsschritte des Petri Net Model Matcher (Semantik-Checker)

vom Typ »Modell erstellen« hoch ist³⁴, gestaltet sich der der Abgleich mit dem vorgegebenen Sachverhalt, den es zu modellieren gilt, als besonders herausfordernd. Im Rahmen des Ansatzes werden daher verschiedene Strategien und Verfahren aus dem Bereich des NLP (siehe auch Abschnitt 3.3.1) eingesetzt, um diese Herausforderung in Angriff zu nehmen. Zudem können Lehrende die Knoten der zu verwendenden Musterlösungen mit mehreren alternativen jedoch bedeutungsgleichen Beschriftungen versehen. Aus diesen wird beim späteren Vergleich die Beschriftung mit der höchsten Ähnlichkeit ausgewählt und somit die Wahrscheinlichkeit erhöht, eine Übereinstimmung zu erzielen. Im Folgenden sollen die einzelnen Arbeitsschritte (I)-(IV) des Semantik-Checkers skizziert werden.

In Arbeitsschritt (I) werden alle Beschriftungen der Knoten im studentischen Modell und in den übergebenen Musterlösungen umgewandelt. Eine derartige Vorverarbeitung findet in vielen Ansätzen auf dem Gebiet der automatisierten Bewertung von Modellen (z. B. [JS09; TSW09b; VPG12]) oder auch beim Prozessmodell-Matching (u.A. [DDM08; Sc18]) eine besondere Beachtung. Zunächst werden alle Zeichen einer Beschriftung in Kleinbuchstaben umgewandelt und anschließend die gesamte Beschriftung mittels Tokenisierung in einzelne Wörter zerlegt. Diese Zerlegung ist notwendig, um später die einzelnen Wörter der Beschriftungen paarweise miteinander vergleichen zu können. Anschließend werden alle Wörter mit einer Liste aus Stoppwörtern abgeglichen und diejenigen Wörter entfernt, die einen vergleichsweise geringen Informationsgehalt aufweisen (siehe z. B. [GL09; JS09; Sc18; VPG12]). Diese Liste wurde aus [Sc18] übernommen, da diese im Gegensatz zu weiteren verfügbaren Stoppwortlisten speziell auf den Einsatz im Kontext Prozessmodellierung abgestimmt worden ist. Anstelle einen Stemmer zur Stammformreduktion (beispielsweise wie in [DDM08; Sc18], siehe Abschnitt 3.3.1) für eine weitere Vereinheitlichung der Wörter in den Beschriftungen einzusetzen, wird im Semantic-Checker eine Lem-

³⁴ Denkbar wäre es auch, bei der Aufgabengestaltung die Beschriftung der Modellelemente oder sogar bereits beschriftete Modellelemente vorzugeben, so dass diese nur noch passend ausgewählt und zusammengesetzt werden müssen.

matisierung (wie in [ALR15; ALR17], siehe Abschnitt 3.3.1) durchgeführt. Dabei wird auch die Bedeutung der Wörter berücksichtigt (für ein Beispiel siehe Abschnitt 3.3.1). Die Lemmatisierung ist zwar deutlich rechenintensiver als die Stammformreduktion, dies hat sich allerdings für den vorgesehenen Einsatzzweck als unproblematisch erwiesen (siehe auch Abschnitt 5.5.5 zur Bewertung des Laufzeitverhaltens). Für die Tokenisierung und Lemmatisierung kommt das CoreNLP-Framework³⁵ der Stanford University zum Einsatz.

Im Arbeitsschritt (II) wird eine Ähnlichkeits-Matrix A_N erstellt, in der die jeweiligen Ähnlichkeitswerte basierend auf dem Vergleich aller Knoten des studentischen Modells mit allen Knoten der aktuell betrachteten Musterlösung eingetragen werden. Zur Berechnung der Gesamtähnlichkeit $sim_{total}(n_s, n_m)$ zwischen einem Knoten n_s aus dem studentischen Modell und einem Knoten n_m aus der Musterlösung werden die Ähnlichkeitswerte zu vier unterschiedlichen Aspekten der Knoten gewichtet summiert:

$$sim_{total}(n_s, n_m) = w_1 \cdot sim_{lab}(n_s, n_m) + w_2 \cdot sim_{pos}(n_s, n_m) \\ + w_3 \cdot sim_{type}(n_s, n_m) + w_4 \cdot sim_{neigh}(n_s, n_m)$$

Für die Wahl der Gewichte gilt dabei die bekannte Voraussetzung:

$$w_1, w_2, w_3, w_4 \geq 0 \wedge w_1 + w_2 + w_3 + w_4 = 1$$

Die in der Gesamtähnlichkeit enthaltenen vier Ähnlichkeiten in Bezug auf die Beschriftung ($sim_{lab}(n_s, n_m)$), die relative Position ($sim_{pos}(n_s, n_m)$), die Nachbarschaft ($sim_{neigh}(n_s, n_m)$) und den Knotentyp ($sim_{type}(n_s, n_m)$) werden in den folgenden Absätzen erläutert.

Wie bereits erwähnt, kommt der **Ähnlichkeit der Beschriftungen** $sim_{lab}(n_s, n_m)$ von zwei Knoten n_s und n_m eine große Bedeutung zu. In Experimenten hat sich gezeigt, dass mit einer hohen Gewichtung w_1 dieses Aspekts die besten Resultate erzielt werden. In den Triple-S-Ansätzen [Sc18] werden Beschriftungen auf ihre syntaktische und semantische Ähnlichkeit hin untersucht. Diese beiden Aspekte werden auch im Semantik-Checker aufgegriffen, allerdings in einer einzelnen Funktion $sim_{lab}(n_s, n_m)$ kombiniert berücksichtigt. Sei $\ell(n)$ die Beschriftung eines Knotens n und die Menge $T(n) = \{t_1, \dots, t_n\}$ die in Arbeitsschritt (I) ermittelten Wörter (engl. »tokens«) dieser Beschriftung $\ell(n)$. Nun wird für jedes Wortpaar (t_s, t_m) mit

35 <https://stanfordnlp.github.io/CoreNLP/>

5. E-Assessment für das Gebiet der grafischen Modellierung

$t_s \in T(n_s)$ und $t_m \in T(n_m)$ die syntaktische $sim_{syn}(t_s, t_m)$ und die semantische Ähnlichkeit $sim_{sem}(t_s, t_m)$ ermittelt und der höhere der beiden Werte als Ähnlichkeit für das Wortpaar (t_s, t_m) in einer Ähnlichkeitsmatrix A_W festgehalten.

Die Bestimmung der syntaktischen Ähnlichkeit $sim_{syn}(t_s, t_m)$ wird anhand der in Abschnitt 3.2.3 vorgestellten Metrik 3.1 auf Basis der Levenshtein-Distanzfunktion lev durchgeführt:

$$sim_{syn}(t_s, t_m) = 1 - \frac{lev(t_s, t_m)}{\max(|t_s|, |t_m|)}$$

Die Ähnlichkeitsmetrik von Wu und Palmer [WP94] (siehe auch Abschnitt 3.3.1) kommt für die Bestimmung der semantischen Ähnlichkeit $sim_{sem}(t_s, t_m)$ zum Einsatz. Auf Basis von hierarchisch geordneten Beziehungen von Konzepten wird die semantische Distanz zwischen den beiden Wörtern wie folgt ermittelt:

$$sim_{sem}(t_s, t_m) = 2 \cdot \frac{depth(lcs(t_s, t_m))}{depth(t_s) + depth(t_m)}$$

Dabei steht $lcs(t_s, t_m)$ für *least common subsumer*, dem in der Hierarchie niedrigsten gemeinsamen Oberbegriff der beiden Wörter t_s und t_m , $depth(t)$ bezeichnet die Tiefe des Wortes t ab dem Wurzelknoten der Hierarchie. Neben der WordNet-Datenbank (wie auch z. B. [Fr09; LSM12; SLG15], siehe Abschnitt 3.3.1) verwendet der Semantik-Checker das WS4J-Framework³⁶ der Carnegie Mellon University für die Berechnung dieser Metrik.

Anschließend werden aus den jeweils höchsten Ähnlichkeitswerten in allen Zeilen bzw. Spalten der Ähnlichkeitsmatrix A_W arithmetische Mittel gebildet. Dabei gibt $avg(T(n_s), T(n_m))$ an, zu welchem Anteil die Wörter der Beschriftung eines Knotens des studentischen Modells $T(n_s)$ mit den Wörtern in der Beschriftung des betrachteten Knotens in der Musterlösung $T(n_m)$ übereinstimmen. Da $avg(T(n_s), T(n_m)) = 1$ für $T(n_s) = \text{»check feasible«}$ und $T(n_m) = \text{»check not feasible«}$ gilt, muss mit $avg(T(n_m), T(n_s))$ auch der umgekehrte Fall berücksichtigt werden. Damit kann dem Umstand gerecht werden, dass möglicherweise Wörter in der einen, nicht aber in der anderen Beschriftung enthalten sind.

Die Ähnlichkeit der Beschriftungen zweier Knoten $sim_{lab}(n_s, n_m)$ wird letztendlich aus dem harmonischen Mittel der beiden arithmetischen Mittel gebildet:

$$sim_{lab}(n_s, n_m) = 2 \frac{avg(T(n_s), T(n_m)) \cdot avg(T(n_m), T(n_s))}{avg(T(n_s), T(n_m)) + avg(T(n_m), T(n_s))}$$

³⁶ <https://code.google.com/archive/p/ws4j/>

Mit der Betrachtung der **Ähnlichkeit der relativen Position** zweier Knoten n_s, n_m soll dem Umstand Rechnung getragen werden, dass diese möglicherweise einen hohen Ähnlichkeitswert im Hinblick auf ihre Beschriftung besitzen, sich aber an eher unterschiedlichen Positionen im Petri-Netz befinden. Dabei wird für die Berechnung der relativen Position $relpos(n, G)$ eines Knotens n in einem Graphen G die Distanz $dist(i, n)$ zu einer Quelle $i \in G$ verwendet. Ebenso wird der Umfang von G als längste Distanz von der Quelle i zu einem Knoten $z \in G$ ermittelt, um daraus das Verhältnis abzuleiten:

$$relpos(n, G) = \frac{dist(i, n)}{max(dist(i, z))}$$

Die Ähnlichkeit der relativen Positionen zweier Knoten n_s und n_m im jeweiligen Graphen G_s bzw. G_m wird dann über die Differenz zwischen diesen berechnet:

$$sim_{pos}(n_s, n_m) = 1 - |relpos(n_s, G_s) - relpos(n_m, G_m)|$$

In den in Abschnitt 3.3.1 beschriebenen Ansätzen zum Prozessmodell-Matching werden lediglich Aktivitäten aus Prozessmodellen – d.h. Transitionen in Petri-Netzen – miteinander verglichen. Im Semantik-Checker jedoch wird für alle Knoten des studentischen Modells ein entsprechender Match in den Musterlösungen gesucht. Dabei wird die Ähnlichkeit des Knotentyps zweier Knoten n_s und n_m wie folgt berücksichtigt: Ist $type(n_s) = \text{Transition}$ und $type(n_m) = \text{Stelle}$ oder vice versa, wird in der Ähnlichkeitsmatrix A_N direkt ein Ähnlichkeitswert $sim_{total}(n_s, n_m) = 0$ eingetragen. Damit wird sichergestellt, dass bei ansonsten ähnlichen Eigenschaften ($sim_{lab}, sim_{pos}, sim_{neigh}$) dennoch keine Übereinstimmung zwischen den zwei zentralen Modellierungselementen von Petri-Netzen mit ihren unterschiedlichen Bedeutungen hergestellt werden kann. Handelt es sich bei einem der beiden Knoten n_s und n_m um einen Knotentyp, der nicht für Petri-Netze vorgesehen ist, wird die **Ähnlichkeit der Knotentypen** n_s und n_m mit $sim_{type}(n_s, n_m)$ wie folgt berücksichtigt:

$$sim_{type}(n_s, n_m) = \begin{cases} 1 & \text{falls } type(n_s) = type(n_m) \\ 0 & \text{sonst.} \end{cases}$$

Dies führt dazu, dass für gleiche Knotentypen von n_s und n_m (z.B. Stelle und Stelle oder Transition und Transition, aber auch sonstige Typen von Modellierungselementen aus dem KEA-Format) ein Ähnlichkeitswert von 1 berechnet wird. Handelt es sich bei n_s um einen Knoten mit einem Typ, der weder Stelle noch Transition ist, wird ein Ähnlichkeitswert von 0 angenommen. Dadurch kann ein syntaktischer Fehler im

5. E-Assessment für das Gebiet der grafischen Modellierung

studentischen Modell durch die Auswahl eines falschen Knotentyps (siehe Check CorrectNodeTypes in Abschnitt 5.3.2) gegebenenfalls ausgeglichen werden, sofern die anderen Ähnlichkeitswerte (sim_{lab} , sim_{pos} , sim_{neigh}) hoch genug ausfallen.

Ein weiteres Indiz dafür, dass zwei Knoten n_s und n_m einander ähnlich sind, stellt die **Ähnlichkeit der Nachbarschaft** dar (siehe z. B. [DDM08], Abschnitt 3.3.1). Dabei werden die Knoten im Vor- und im Nachbereich der beiden Knoten miteinander verglichen. Für den Vergleich wird die Gesamtähnlichkeit sim_{total} abzüglich der Nachbarschaftsähnlichkeit herangezogen:

$$sim_{total \setminus neigh}(n_s, n_m) = w_1 \cdot sim_{lab}(n_s, n_m) + w_2 \cdot sim_{pos}(n_s, n_m) + w_3 \cdot sim_{type}(n_s, n_m)$$

Auch hier gilt:

$$w_1, w_2, w_3 \geq 0 \wedge w_1 + w_2 + w_3 = 1$$

Mit $avg(\bullet n_s, \bullet n_m)$ sei das arithmetische Mittel der jeweils höchsten Ähnlichkeiten $sim_{total \setminus neigh}(n_i, n_j)$ für jeden Knoten $n_i \in \bullet n_s$ und jeden Knoten $n_j \in \bullet n_m$ bezeichnet. Dieses gibt an, zu welchem Grad die Knoten in $\bullet n_s$ den Knoten in $\bullet n_m$ ähneln. Um den umgekehrten Fall auch zu berücksichtigen, wird $avg(\bullet n_m, \bullet n_s)$ ebenfalls berechnet. Analog wird für die Knoten in $n_s \bullet$ und $n_m \bullet$ verfahren. Über alle vier Teilergebnisse wird nun die Ähnlichkeit der Nachbarschaften $sim_{neigh}(n_s, n_m)$ zweier Knoten n_s, n_m über die Bildung des arithmetischen Mittels bestimmt:

$$sim_{neigh}(n_s, n_m) = \frac{avg(\bullet n_s, \bullet n_m) + avg(\bullet n_m, \bullet n_s) + avg(n_s \bullet, n_m \bullet) + avg(n_m \bullet, n_s \bullet)}{4}$$

Anhand der vier beschriebenen Ähnlichkeitsmaße kann nun in Arbeitsschritt (III) das eigentliche Matching durchgeführt werden. Dabei werden die Gesamtähnlichkeitswerte $sim_{total}(n_s, n_m)$ in der Ähnlichkeitsmatrix A_n aus Arbeitsschritt (II) nacheinander betrachtet und für jeden Knoten n_s im studentischen Modell und jeden Knoten n_m in der Musterlösung entschieden, ob eine Übereinstimmung vorliegt. Eine solche Übereinstimmung wird Match genannt. Ein Match $\lambda : N_s \times N_m$ ordnet einem Knoten $n_s \in N_s$ einem Knoten in $n_m \in N_m$ zu. Die Menge aller Matches sei mit Λ bezeichnet. Dabei können komplexe $n : m$ -Matches (siehe auch [WDM10], Abschnitt 3.3.1) gebildet werden, d.h. ein Knoten $n_s \in N_s$ kann auch mehreren Knoten $n_m \in N_m$ der Musterlösung zugeordnet werden. Anhand des Konfigurationsparameters C_m mit möglichen Werten $w \in \mathbb{N}$ kann jedoch die maximale Anzahl möglicher Matches gesteuert werden, so dass sich mit $C_m = 1$ auch $n : 1$ -Matches herstellen lassen, falls gewünscht. Die Vorgehensweise und Entscheidungsparameter für die Bildung einer

Menge von Matches Λ wird in Auflistung 5.2 dargestellt.

```

1  Eingabe: Zwei Graphen  $G_s$  und  $G_m$  mit Knotenmengen  $N_s$  und  $N_m$ 
2  Ausgabe: Eine Menge von Matches  $\Lambda$ 
3   $\Lambda \leftarrow \emptyset$ 
4   $\theta \leftarrow$  Schwellenwert;
5   $\Delta \leftarrow$  Maximalabstand zum besten Match;
6   $M \leftarrow$  Maximalanzahl möglicher Matches  $C_m$ 
7
8  for each ( $n_s \in N_s$ ) do
9     $\Lambda_{temp} \leftarrow \emptyset$ 
10   for each ( $n_m \in N_m$ ) do
11     if ( $sim_{total}(n_s, n_m) \geq \theta$ ) do
12       Füge  $\lambda \leftarrow (n_s, n_m)$  zu  $\Lambda_{temp}$  hinzu;
13     end
14   end
15    $\Lambda_{temp} \leftarrow sort(\Lambda_{temp})$ 
16   for each ( $\lambda \in \Lambda_{temp}$ ) do
17     if ( $\lambda$  ist das erste Element  $\lambda_{max} \in \Lambda_{temp}$ ) do
18       Füge  $\lambda$  zu  $\Lambda$  hinzu;
19     end
20     else if ( $sim(\lambda_{max}) - sim(\lambda) < \Delta \wedge \lambda$  ist innerhalb der ersten  $M$  Elemente  $\in \Lambda_{temp}$ ) do
21       Füge  $\lambda$  zu  $\Lambda$  hinzu;
22     end
23   end
24
25  return  $\Lambda$ 

```

Auflistung 5.2: Bildung einer Menge von Matches Λ für die Knoten des studentischen Modells G_s

Demnach wird ein Match $\lambda = (n_s, n_m)$ zu zwei Knoten n_s, n_m zunächst zu einer Menge Λ_{temp} hinzugefügt, wenn die Gesamtähnlichkeit $sim_{total}(n_s, n_m)$ größer gleich einem Schwellenwert θ ist. Nachdem ein Knoten n_s mit allen Knoten N_m der Musterlösung verglichen wurde, wird eine Sortierung von Λ_{temp} in absteigender Richtung nach dem Ähnlichkeitswert durchgeführt. Sind in Λ_{temp} für einen Knoten n_s nach dem Vergleich mit allen Knoten N_m mehrere Matches enthalten, wird so in jedem Fall das Match mit dem höchsten Ähnlichkeitswert λ_{max} in die finale Menge an Matches Λ aufgenommen. Sofern C_m entsprechend gesetzt ist, können auch weitere Matches aus Λ_{temp} in Λ aufgenommen werden. Dazu darf allerdings die Differenz zwischen dem Ähnlichkeitswert von λ_{max} und dem Ähnlichkeitswert der weiteren Matches in Λ_{temp} einen festgelegten Abstand Δ nicht überschreiten. Damit wird verhindert,

5. E-Assessment für das Gebiet der grafischen Modellierung

dass weitere Matches in Λ aufgenommen werden, wenn der Ähnlichkeitswert von λ_{max} deutlich (bzw. Δ entsprechend) über deren Ähnlichkeitswert liegt.

Anhand der Menge von Matches Λ kann nun in Arbeitsschritt (IV) untersucht werden, für welche Knoten im studentischen Modell eine Übereinstimmung mit der Musterlösung ermittelt werden konnte und für welche nicht. Aus letzterem lassen sich zwei Mengen N_{fehlt} und N_{extra} ableiten. Die Menge N_{fehlt} bezeichnet diejenigen Knoten $n_m \in N_m$ der Musterlösung, die keine Entsprechung im studentischen Modell besitzen. Umgekehrt bezeichnet die Menge N_{extra} diejenigen Knoten $n_s \in N_s$ im studentischen Modell, die in der Musterlösung nicht gefunden werden konnten. Ebenso soll für die Kanten verfahren werden. Dabei gilt, dass eine Kante $k_s = (n_{s1}, n_{s2})$ mit dem Quellknoten n_{s1} und dem Zielknoten n_{s2} genau dann einer Kante $k_m = (n_{m1}, n_{m2})$ in der Musterlösung entspricht, wenn entsprechende Matches der jeweiligen Quell- bzw. Zielknoten in Λ enthalten sind: $(n_{s1}, n_{m1}) \in \Lambda \wedge (n_{s2}, n_{m2}) \in \Lambda$. Das Kantenpaar (k_s, k_m) bildet dann ein sogenanntes Kantenmatch $\psi = (k_s, k_m)$ und wird der Menge aller Kantenmatches Ψ hinzugefügt. Ist ein Knoten n , der kein Match besitzt ($(n, z) \notin \Lambda \wedge (z, n) \notin \Lambda$ für beliebige Knoten z) vorhanden, kommt es allerdings dazu, dass alle ein- und ausgehenden Kanten von n keine Entsprechung finden. Daher werden derartige Knoten n und ihre ein- und ausgehenden Kanten von der Betrachtung ausgeschlossen. Anhand der Menge aller Kantenmatches Ψ können nun analog zu Λ die Mengen K_{fehlt} (Kanten, die in der Musterlösung vorhanden sind, aber nicht im studentischen Modell identifiziert werden konnten) und K_{extra} (Kanten, die im studentischen Modell vorhanden sind, aber keine Entsprechung in der Musterlösung besitzen) abgeleitet werden.

Der Semantik-Checker liefert die finalen Ergebnisse anhand der Methode **ModelMatch** vom Typ Metrik zurück. Das Gesamtergebnis $R_d(g)$ im Intervall $[0, 1]$ basiert auf den Mengen N_{fehlt} , N_{extra} und K_{fehlt} sowie K_{extra} , die in Arbeitsschritt (IV) ermittelt wurden. Dabei können entsprechende Strafwerte C_{nf} , C_{ne} , C_{kf} , C_{ke} nach individuellem Ermessen festgelegt werden. Diese bestimmen, welchen Abzug es vom oberen Ende 1 des Intervalls für die entsprechenden Fehler geben soll. Das Gesamtergebnis darf nicht schlechter als das untere Ende 0 des Intervalls werden, d.h.:

$$R_d(g) = \max(0, 1 - (C_{nf} \cdot |N_{fehlt}| + C_{ne} \cdot |N_{extra}| + C_{kf} \cdot |K_{fehlt}| + C_{ke} \cdot |K_{extra}|))$$

In der Methode ModelMatch werden zudem für alle Einträge in den Mengen N_{fehlt} , N_{extra} , K_{fehlt} sowie K_{extra} entsprechende Feedback-Nachrichten vom Beurteilungs-

Tab. 5.4.: Feedback-Nachrichten des Semantik-Checkers. IDM: Integrierte Darstellung im studentischen Modell

Name	Inhalt	Platzhalter %s	Typ	IDM
F_{nf}	Das Modell enthält keinen Knoten »%s«	Beschriftung des Knotens in N_m	Negativ	nein
F_{ne}	Dieser Knoten ist nicht in der verwendeten Musterlösung enthalten.	-	Negativ	ja
F_{kf}	Das Modell enthält keine Kante von »%s« zu »%s«	Beschriftungen von Quell- und Zielknoten in N_m dieser Kante	Negativ	nein
F_{ke}	Diese Kante ist nicht in der verwendeten Musterlösung enthalten.	-	Negativ	ja
F_{match}	Dieser Knoten entspricht »%s«	Beschriftung des Knotens in N_m mit dem ein Match existiert	Neutral	ja
F_{form}	Versuchen Sie, das Modell anhand der gegebenen Hinweise weiter anzupassen!	-	Neutral	nein

typ »Negativ« erzeugt. Knoten $n \in N_{extra}$ bzw. Kanten in K_{extra} werden dabei direkt im studentischen Modell entsprechend grafisch hervorgehoben. Ebenso werden für alle Matches $\lambda \in \Lambda$ zu den Knoten der studentischen Lösung entsprechende Feedback-Nachrichten mit Beurteilungstyp »Neutral« erstellt. Dadurch können die Matches ebenfalls direkt im studentischen Modell angezeigt werden, wodurch sich die Ergebnisse des Semantik-Checkers einfach nachvollziehen lassen. Die Arten und Inhalte der Feedback-Nachrichten sind in Tab. 5.4 dargestellt.

Werden alle Feedback-Nachrichten zu F_{nf} , F_{ne} , F_{kf} und F_{ke} zurückgegeben, kann auf deren Grundlage leicht in einem Schritt die Musterlösung rekonstruiert werden. Da dies im Selbstlern-Einsatz ungünstig sein kann, wurde ein Konfigurationsparameter C_f mit einem Wert $w \in \mathbb{N}$ erstellt. Der Wert w gibt an, für maximal wie viele Einträge in den Mengen N_{fehlt} , N_{extra} , K_{fehlt} sowie K_{extra} jeweils Feedback-Nachrichten zurückgegeben werden sollen. Ist $C_f = 1$, so wird für jede Menge maximal eine Feedback-Nachricht erstellt. Zudem wird die Feedback-Nachricht F_{form} ergänzt. Die Berechnung des Methodenergebnisses $R_d(g)$ bleibt dabei unverändert.

5.3.4. Pragmatik-Checker

Der Pragmatik-Checker (Projektname *Universal Pragmatic Checker*) bezieht sich auf die folgende Kompetenzfacette des Kompetenzmodells (siehe Abschnitt 4.1) zum

5. E-Assessment für das Gebiet der grafischen Modellierung

Aufgabentyp »Modell erstellen«:

MB 4.05: Die Lernenden können ein Modell a) auf Basis bekannter Richtlinien oder Konventionen gut lesbar und strukturiert sowie b) für die jeweilige Zielgruppe verständlich erstellen.

Bei der untersuchten Aufgabenstellung (siehe Evaluation im Abschnitt 5.5) ist die Berücksichtigung einer bestimmten Zielgruppe nicht vorgesehen. Somit soll der Fokus auf Punkt a) gelegt werden. In Abschnitt 3.2 wurden existierende Richtlinien und Metriken für das Gebiet der Geschäftsprozessmodellierung vorgestellt. Insbesondere die in der Literatur beschriebenen Metriken lassen sich häufig dem pragmatischen Qualitätsaspekt zuordnen. Eine umfangreiche Zusammenstellung und Implementierung dieser Metriken ist im Rahmen eines Gamification-Ansatzes für die Geschäftsprozessmodellierung von PFLANZL [Pf17] und PFLANZL und VOSEN [PV18] durchgeführt worden. Diese Vorarbeiten wurden im vorliegenden Forschungsvorhaben aufgegriffen und ein Großteil der Metriken³⁷ in den Pragmatik-Checker integriert, um zu untersuchen, inwieweit sich diese für den hier betrachteten Einsatzzweck verwenden lassen. Dabei wurden die Metriken auf die interne Graphstruktur (siehe Abschnitt 5.3.1) angepasst und dabei die Besonderheiten des KEA-Formats³⁸ berücksichtigt. Darüber hinaus wurden einige weitere Metriken ergänzt. Teilweise wurden Metriken auch in eine Methode vom Typ Check überführt, sofern dies für das intendierte Bewertungsschema notwendig war.

Tab. 5.5 stellt einen Überblick über die Methoden des Pragmatik-Checkers dar. Diese sind grundsätzlich universell, d.h. sie können nicht nur für die Untersuchung von Petri-Netzen, sondern auch für andere graphbasierte Diagramme eingesetzt werden (Spalte »Universell«). In einigen Fällen kann es jedoch nötig sein, spezielle Eigenschaften bestimmter Modellierungssprachen zu berücksichtigen. Die im Pragmatik-Checker implementierten Methoden können anhand ihres Betrachtungsgegenstands klassifiziert werden. Einige Methoden untersuchen die **grafische Darstellung** des studentischen Modells. Dabei werden die Positionen der Modellelemente berücksichtigt, um deren Anordnung auf der Arbeitsfläche des Modellierungseeditors zu beurteilen. Eine weitere Klasse von Methoden bezieht sich auf spezielle **Graphigenschaften** des dem studentischen Modell zugrundeliegenden Graphen G , wie z. B. Anzahl der Knoten. Hierbei spielt die Position der Modellelemente keine Rolle.

³⁷ Die unberücksichtigten Metriken beziehen sich auf Modellierungselemente, die nur im speziellen Modellierungswerkzeug des Gamification-Ansatzes verfügbar sind.

³⁸ Dazu zählen die im Abschnitt 5.3.2 beschriebenen möglichen Abweichungen, wie z.B. Kanten ohne Start- und/oder Zielknoten.

Tab. 5.5.: Übersicht über die Methoden des Pragmatik-Checkers

Name	Universell	Typ	Ziel
Betrachtungsgegenstand: Grafische Darstellung			
RightwardFlowDirection	ja	Metrik	Alle Kanten zeigen in Flussrichtung links nach rechts
DownwardFlowDirection	ja	Metrik	Alle Kanten zeigen in Flussrichtung oben nach unten
ConsistentFlowDirection	ja	Metrik	Alle Kanten zeigen einheitlich in eine der beiden Flussrichtungen
EdgeCrossings	ja	Metrik	Keine Kantenüberschneidungen vorhanden
EdgeOrthogonality	ja	Metrik	Alle Kanten verlaufen orthogonal
Betrachtungsgegenstand: Grapheigenschaften			
Size	ja	Metrik	Knotenanzahl unterhalb eines festgelegten Schwellenwerts
CoefficientOfConnectivity	ja	Metrik	Möglichst geringer durchschnittlicher Knotengrad
Density	ja	Metrik	Möglichst geringe Anzahl an Kanten
Cyclicity	ja	Metrik	Keine Zyklen im Graph vorhanden
ConnectorMismatch	Petri-Netze	Metrik	Übereinstimmung der Anzahl von Split- und Join-Mustern des gleichen Grads
OneStartOneEnd	ja	Metrik	Genau ein Start und ein Endknoten im Graph
Betrachtungsgegenstand: Beschriftung			
NodesLabeled	Petri-Netze	Check	Alle Knoten beschriftet
LabelStyle	Petri-Netze	Check	Alle Knoten entsprechen einem vorgegebenen Beschriftungsstil

Andere Methoden betrachten ausschließlich die **Beschriftungen** der Modellelemente im studentischen Modell.

Die beiden Metriken **DownwardFlowDirection** und **RightwardFlowDirection** können verwendet werden, um zu prüfen, ob die Kanten im Modell in einer konsistenten Flussrichtung von oben nach unten bzw. von links nach rechts verlaufen³⁹. Dies wird in der Literatur mit einer besseren Lesbarkeit des Modells in Verbindung gebracht (siehe z. B. [FS15; Pu02]). Metrik 5.1 definiert die Variante **RightwardFlowDirection** für eine Flussrichtung von links nach rechts als Menge aller Kanten mit entsprechen-

³⁹ Um Knickpunkte der Kanten zu berücksichtigen, arbeiten alle Metriken des Pragmatik-Checkers, die sich auf Kantenpositionen beziehen, mit einem Duplikat des Graphen G , bei welchem Hilfsknoten an alle Knickpunkte einer Kante eingefügt werden. Anschließend werden alle Kanten mit Knickpunkten entfernt und stattdessen geradlinige Kanten von bzw. zu den Hilfsknoten eingefügt (siehe auch [Pu02]). Die Erstellung von gebogenen Kanten ist mit dem Modellierungseditor nicht möglich.

5. E-Assessment für das Gebiet der grafischen Modellierung

der Flussrichtung gemessen an der Anzahl aller Kanten.

Metrik 5.1: RightwardFlowDirection

Sei $|E|$ die Anzahl der Kanten in einem Graph $G = (V, E)$, $x(v)$ die X-Koordinate eines Knotens $v \in V$ und $n_s(e)$ bzw. $n_t(e)$ Start- bzw. Zielknoten einer Kante $e \in E$.

Eine Kante $e \in E$ besitzt genau dann eine Flussrichtung von links nach rechts, wenn die X-Koordinate ihres Startknotens kleiner oder gleich der X-Koordinate ihres Zielknotens ist: $x(n_s(e)) \leq x(n_t(e))$. Sie wird der Menge E_{rwf} hinzugefügt.

Die Metrik **RightwardFlowDirection** für einen Graphen G sei daraufhin definiert wie folgt:

$$\text{RightwardFlowDirection}(G) = \frac{|E_{rwf}|}{|E|}$$

Die Variante **DownwardFlowDirection** kann analog definiert werden. Dabei wird lediglich anstelle der X-Koordinate $x(v)$ die Y-Koordinate $y(v)$ des Start- bzw. Zielknotens einer Kante betrachtet. Die obere Grenze entspricht der Anzahl Kanten $|E|$ im Graph G . Im Pragmatik-Checker werden bei der Ausführung der beiden Metriken Feedback-Nachrichten für diejenigen Kanten erstellt, die von der vorgegebenen Flussrichtung abweichen. Diese werden zusätzlich im Modell hervorgehoben. Es kann gewünscht sein, nicht speziell eine der beiden Metriken auszuwählen, sondern die Wahl der modellierenden Person zu überlassen. Dafür wurde die Methode **ConsistentFlowDirection** ergänzt, die zunächst beide Metriken aufruft und dann das größere der beiden Ergebnisse $R_d(g)$ der Methoden **RightwardFlowDirection** und **DownwardFlowDirection** zurückgibt.

Der Literatur zufolge hat insbesondere die Anzahl von Kantenüberschneidungen einen signifikanten Einfluss auf die Lesbarkeit eines Modells (z. B. [EJS10; Pu02]). Kantenüberschneidungen erschweren es, einen Pfad im Modell visuell zu verfolgen. Die Metrik 5.2 **EdgeCrossings** misst die Anzahl an Kantenüberschneidungen an der Anzahl maximal möglicher Kantenüberschneidungen. Diese lassen sich durch die Anzahl aller möglichen Kantenpaare abzüglich der Anzahl der Kantenpaare, die sich niemals schneiden können, bestimmen (siehe [Pu02]).

Metrik 5.2: EdgeCrossings

Sei $|E_{ec}|$ die Anzahl an Kantenüberschneidungen der Kanten $e \in E$ eines Graphen $G(V, E)$ und $|E_{poss}|$ die maximal mögliche Anzahl an Kantenüberschneidungen.

Die Metrik **EdgeCrossings** für einen Graphen G sei daraufhin definiert wie folgt:

$$EdgeCrossings(G) = 1 - \frac{|E_{ec}|}{|E_{poss}|}$$

Falls für eine Aufgabe vom Typ »Modell erstellen« feststeht, dass es möglich ist, ein Lösungsmodell zu erstellen, welches frei von Kantenüberschneidungen ist, kann mit dem Check **NoEdgeCrossings** auch das Auftreten von Kantenüberschneidungen direkt mit einem Abzug bestraft werden, ohne die maximal mögliche Anzahl an Kantenüberschneidungen als obere Grenze zu berücksichtigen. Sich überschneidende Kanten werden anhand einer Feedback-Nachricht zurückgemeldet, ebenso werden diese direkt im studentischen Modell hervorgehoben.

PURCHASE [Pu02] schlägt die orthogonale Anordnung von Kanten vor, so dass diese ausschließlich horizontal oder vertikal verlaufen. Um dies zu unterstützen, können Knickpunkte verwendet werden. Durch diese Art der Linienführung soll das Modell übersichtlicher und einfacher zu lesen sein. Dies kann mit der Metrik **EdgeOrthogonality** gemessen werden (siehe Metrik 5.3).

Metrik 5.3: EdgeOrthogonality

Sei $|E_{orthogonal}|$ die Anzahl orthogonaler (d.h. ausschließlich vertikal bzw. horizontal verlaufender) Kanten eines Graphen $G(V, E)$ und $|E|$ die Anzahl aller Kanten $e \in E$.

Die Metrik **EdgeOrthogonality** für einen Graphen G sei daraufhin definiert als:

$$EdgeOrthogonality(G) = \frac{|E_{orthogonal}|}{|E|}$$

Dabei sollen möglichst wenige Kanten diese Vorgabe verletzen. In empirischen Studien konnte bislang allerdings kein signifikanter Einfluss einer orthogonalen Linienführung auf die Lesbarkeit von Modellen nachgewiesen werden ([PPP12; Pu01]).

Wie in Abschnitt 3.2.3 dargelegt, wird eine hohe Modellkomplexität mit einer hohen Fehlerwahrscheinlichkeit [Me08a] und einem negativen Einfluss auf die Wartbarkeit

5. E-Assessment für das Gebiet der grafischen Modellierung

und Verständlichkeit assoziiert [Ca06; HFL14]. Zahlreiche Metriken mit Bezug auf die Modellkomplexität beziehen sich auf Grapheneigenschaften, die über die Analyse der Knoten und Kanten im Graph abgeleitet werden. In den Kontext des im Forschungsvorhaben betrachteten Aufgabentyps »Modell erstellen« lassen sich diese Metriken nur schwer einbetten. Da durch den fest vorgegebenen Sachverhalt bereits spezifische Einschränkungen in Bezug auf die Graphstruktur bestehen, ist es nicht sinnvoll, über diese Metriken eine Bewertung allgemein vorteilhafter (bzw. wenig vorteilhafter) Grapheneigenschaften vorzunehmen. Allerdings können diese Metriken im Rahmen einer frei gestalteten Modellierungsaufgabe vom Typ »Modell erstellen« durchaus gewinnbringend eingesetzt werden⁴⁰. Sie sollen im Folgenden kurz skizziert werden:

- **Size:** Hierbei handelt es sich um eine der simpelsten Metriken in Bezug auf die Modellkomplexität. Es wird die Anzahl der Modellelemente (üblicherweise Knoten [Me08a], teilweise auch nur die Anzahl der Aktivitäten [Ca06]) im Graphen bestimmt. Die 7PMG-Richtlinien [MRA10] empfehlen beispielsweise, möglichst wenige Modellelemente zu verwenden (G1) und konkret, ein Modell zu zerlegen, wenn es mehr als 50 (nach [Me12] 31) Elemente enthält.
- **Density:** Diese Metrik bestimmt die Dichte eines Graphen G als das Verhältnis zwischen der Anzahl existierender und maximal möglicher Kanten in G anhand $\frac{|V|}{|E|(|E|-1)}$. Da eine geringere Dichte mit einer geringeren Anzahl an Kanten einhergeht, lässt sich der Kontrollfluss leichter nachvollziehen. Dieser positive Effekt auf die Verständlichkeit konnte empirisch nachgewiesen werden (z. B. [MRC07]).
- **CoefficientOfConnectivity:** Das Verhältnis zwischen Kanten- und Knotenanzahl in einem Graphen G wird durch den sogenannten »Coefficient of Connectivity« bestimmt. Diese Metrik betrachtet die Anzahl an Kanten gemessen an der Anzahl an Knoten mit $\frac{|E|}{|V|}$ und beschreibt somit den durchschnittlichen Knotengrad [Ca06].
- **Cyclicity:** Die Zyklizität gibt an, wieviele Knoten des Graphen G Teil eines Zyklus sind. Dazu wird für jeden Knoten $v \in V$ untersucht, ob ein gerichteter Pfad von v nach v existiert. Nach MENDLING [Me08a] führt eine höhere Anzahl an Zyklen in einem Graphen zu einer schlechteren Verständlichkeit und höheren Fehlerwahrscheinlichkeit. Im Pragmatik-Checker wurde ein entsprechender

⁴⁰ Hier soll ebenfalls ein Modell erstellt werden, jedoch ohne feste Vorgabe eines Sachverhalts, z. B. beim Aufgabentyp »Beispielmodell erstellen« oder im Rahmen einer Fallstudie (siehe Abschnitt 4.2).

Check **NoCycles** ergänzt, mit welchem das Vorhandensein von Zyklen im studentischen Modell geprüft wird. Dieser kann bei Bedarf eingesetzt werden, um anhand von Feedback-Nachrichten entsprechende Hinweise zurückzumelden.

- **ConnectorMismatch**: Der 7PMG-Richtlinie G7 (siehe Abschnitt 3.2.2) von MENDLING et al. [MRA10] nach sollten Prozessmodelle für jede Verzweigung des Kontrollflusses auch eine entsprechende Zusammenführung des gleichen Typs besitzen. Abweichungen von dieser Regel erschweren die Verständlichkeit und erhöhen die Fehlerwahrscheinlichkeit. Beispielsweise resultiert in einem Petri-Netz die Verwendung eines XOR-Splits gefolgt von einem AND-Join in einem Deadlock. Die Metrik **ConnectorMismatch** summiert dabei jeweils die Differenz zwischen der Anzahl entsprechender Split-Konstrukte und der Anzahl dazugehöriger Join-Konstrukte für alle möglichen Typen von Konnektoren [Me08a].
- **OneStartOneEnd**: Die Verwendung von genau einem Start- und einem Endknoten in einem Prozessmodell trägt zur besseren Lesbarkeit und geringeren Fehlerwahrscheinlichkeit bei [MNA07], dies wird auch in der 7PMG-Richtlinie G3 [MRA10] empfohlen. Die im Pragmatik-Checker umgesetzte Methode liefert 1 zurück, wenn sowohl die Anzahl von Quellen als auch von Senken 1 beträgt. Weicht einer der beiden Werte von 1 ab, wird 0,5 zurückgegeben, weichen beide Werte ab, 0⁴¹.

Die Beschriftungen von Modellelementen lassen sich ebenfalls aus pragmatischer Sicht bewerten. Hierbei soll an erster Stelle untersucht werden, ob alle Modellelemente beschriftet worden sind. Ohne eine Beschriftung lässt sich eine inhaltliche Interpretation nur anhand des Typs des betreffenden Elements sowie anhand des Kontextes durchführen, so dass ein negativer Einfluss auf die Verständlichkeit des Modells abgeleitet werden kann. Mit der Methode **NodesLabeled** wird für alle Knoten (mit Ausnahme von Marken) untersucht, ob diese beschriftet worden sind. Sollen anstelle von Petri-Netzen andere Modellierungssprachen betrachtet werden, so kann es nötig sein, z.B. auch die Kanten auf Beschriftungen zu prüfen. Beispielsweise werden bei ER-Diagrammen die Kardinalitäten der Beziehungen anhand von Kantenbeschriftungen festgelegt (siehe Abschnitt 2.2.3). Ebenso muss berücksichtigt

⁴¹ Die Methode **OneStartOneEnd** sollte selbstverständlich nicht in Kombination mit der Methode **WorkflowNet** des Syntax-Checkers eingesetzt werden, da sie äquivalent zur Prüfung von Punkt (i) der **Workflow-Netz-Definition** gemäß Def. 2.24 ist. Sie kann jedoch für die Bewertung von Petri-Netzen, für die keine **Workflow-Eigenschaften** gefordert werden, oder aber für die Bewertung von Modellen anderer Modellierungssprachen nützlich sein.

5. E-Assessment für das Gebiet der grafischen Modellierung

werden, dass Knoten, die Konnektoren repräsentieren (z. B. bei EPK oder BPMN), üblicherweise keine Beschriftung besitzen.

Für eine tiefergehende Prüfung der Beschriftung von Modellelementen kann die Methode **LabelStyle** eingesetzt werden. Diese untersucht Stellen und Transitionen im studentischen Modell daraufhin, ob deren Beschriftungen einem vorgegebenen Muster aus sogenannten Part-Of-Speech-Tags (POS-Tags, siehe auch Abschnitt 3.3.1) entsprechen. Für das Part-Of-Speech-Tagging wird wie beim Semantik-Checker (siehe Abschnitt 5.3.3) das Core-NLP-Framework der Stanford-University verwendet. Die Beschriftungen werden zunächst in eine Sequenz aus den korrespondierenden POS-Tags ihrer Wörter transformiert, beispielsweise »NN VBZ VBN«⁴² für die Stellenbeschriftungen »offer is created« und »request is rejected«. Die Transitionsbeschriftungen »reject request« und »inform customer« entsprechen der POS-Tag-Sequenz »VB NN«⁴³. Letzteres entspricht der 7PMG-Richtlinie G7 [MRA10], welche für die Beschriftung von Aktivitäten in Geschäftsprozessen den Verb-Objekt-Stil empfiehlt. Mit regulären Ausdrücken, die sich je nach den umzusetzenden Konventionen individuell konfigurieren lassen, können die Sequenzen aus POS-Tags daraufhin geprüft werden, ob das gewünschte Muster für Stellen bzw. Transitionen vorliegt.

5.3.5. Erreichbarkeits-Checker

Um im Kontext des Aufgabentyps »Modell erstellen« auch Informationen zu dynamischen Eigenschaften von Petri-Netzen (siehe Abschnitt 2.2.4) liefern zu können, kann der im Rahmen einer Abschlussarbeit⁴⁴ entwickelte Erreichbarkeits-Checker (Projektname *Petri Net Reachability Checker*) eingesetzt werden. Dieser erstellt einen Erreichbarkeitsgraphen des studentischen Modells, mit dessen Hilfe die Prüfung dynamischer Eigenschaften (siehe Abschnitt 2.2.4) ermöglicht wird.

Bei der Modellierung mit Petri-Netzen im Kontext der Geschäftsprozessmodellierung spielt insbesondere die Soundness-Eigenschaft (siehe Def. 2.25) von Workflow-Netzen eine wichtige Rolle. Zur Erfüllung müssen die folgenden drei Kriterien zutreffen:

⁴² NN: noun - singular or mass, VBZ: verb - present tense, 3rd person singular, VBN: Verb - past participle. Eine Übersicht der möglichen POS-Tags ist beispielsweise der Dokumentation des Penn-Treebank-Korpus unter <https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf> zu entnehmen.

⁴³ VB: verb - base form.

⁴⁴ Fabian Aisenbrey, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Generic framework for assessment services in e-assessment of graphical modeling*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2023.

Tab. 5.6.: Feedback-Nachrichten des Erreichbarkeits-Checkers bezüglich der Soundness-Eigenschaft. IDM: Integrierte Darstellung im studentischen Modell

Name	Methode	Inhalt	Typ	IDM
F_{pos}	Soundness	Das Netz ist sound.	Positiv	nein
F_{neg}	Soundness	Das Netz ist nicht sound.	Negativ	nein
F_{na}	Soundness	Es konnte nicht überprüft werden, ob das Netz sound ist. Entweder handelt es sich nicht um ein Workflow-Netz, oder das Netz ist möglicherweise nicht beschränkt.	Neutral	nein
F_{pos}	ProperCompletion	Es gibt keine erreichbare Markierung, bei der sich eine Marke in der Senke befindet und noch weitere Marken im Netz auftreten.	Positiv	nein
F_{neg}	ProperCompletion	Es gibt mindestens eine erreichbare Markierung, bei der sich eine Marke in der Senke befindet und noch weitere Marken im Netz auftreten.	Negativ	nein
F_{pos}	Completeness	Es gibt keine erreichbare Markierung, von der aus kein Endzustand erreicht werden kann.	Positiv	nein
F_{neg}	Completeness	Es gibt mindestens eine erreichbare Markierung, von der aus kein Endzustand erreicht werden kann.	Negativ	nein
F_{pos}	NoDeadTransitions	Im Netz sind keine toten Transitionen vorhanden.	Positiv	nein
F_{neg}	NoDeadTransitions	Das Netz besitzt mindestens eine tote Transition.	Negativ	nein
F_{id}	NoDeadTransitions	Diese Transition ist tot, das bedeutet, sie ist unter keiner erreichbaren Markierung aktiviert.	Negativ	ja

- (i) *Completeness*: Befindet sich eine Marke in der Startstelle (Quelle), muss von dieser Startmarkierung aus eine Folgemarkierung erreichbar sein, in welcher sich eine Marke in der Endstelle (Senke) befindet.
- (ii) *ProperCompletion*: Falls sich eine Marke in der Endstelle (Senke) befindet, dürfen sich keine weiteren Marken in anderen Stellen des Netzes befinden.
- (iii) *NoDeadTransitions*: Ausgehend von der Startmarkierung – siehe Punkt (i) – muss für jede Transition eine Folgemarkierung erreichbar sein, unter welcher sie aktivierbar ist.

Für die Anwendung dieser Prüfungen gelten im Erreichbarkeits-Checker zwei Einschränkungen: (i) das betrachtete Modell muss ein Workflow-Netz und (ii) die Anzahl von Knoten und Kanten des Erreichbarkeitsgraphen kleiner als 10.000^{45} sein. Tab. 5.6 liefert einen Überblick über die entsprechenden Methoden des Erreichbarkeits-Checkers und die dazugehörigen Feedback-Nachrichten. Darüber hinaus können mit dem Erreichbarkeits-Checker auch die Eigenschaften Lebendigkeit, partielle bzw. totale Verklemmung und Reversibilität (siehe z. B. [Ba90; Re10]) geprüft werden.

⁴⁵ Eingeführt, um eine Terminierung auch bei unbeschränkten Workflow-Netzen sicherzustellen.

5. E-Assessment für das Gebiet der grafischen Modellierung

Für die Analyse von Petri-Netzen existieren bereits zahlreiche Werkzeuge⁴⁶. Es hat sich gezeigt, dass viele bestehende Bibliotheken zur Analyse von Petri-Netzen nicht immer sinnvoll im Kontext von Bewertungsdiensten der KEA-Mod-Plattform eingesetzt werden können. Beispielsweise bietet das PM4PY-Framework⁴⁷ die Möglichkeit, zu einem gegebenen Petri-Netz z. B. die Soundness-Eigenschaft oder das Vorliegen eines Workflow-Netzes zu prüfen. Allerdings liefern die dort verwendeten Methoden nur boolesche Ergebnisse zurück und keine weiteren Informationen zu den konkreten vorliegenden Verletzungen. Daher könnten unter Verwendung des PM4PY-Frameworks keine Feedback-Nachrichten generiert werden, die auf die zugrundeliegende Fehlerquelle hinweisen und z.B. tote Transitionen direkt im Modell hervorgehoben werden.

5.4. Anwendungsbeispiel

Dieser Abschnitt soll die Kernfunktionalitäten der KEA-Mod-Plattform aus Sicht der Nutzenden demonstrieren. Der Zugriff auf die Plattform ist mit einem Webbrowser über die Einstiegsseite (siehe Abb. 5.9) möglich. Dort kann in einem ersten Schritt die Anmeldung an der Plattform erfolgen. Hierbei ist es unter dem Link »Registrierung« möglich, zunächst eine frei gewählte Kennung mit einem dazugehörigen Passwort an der Plattform zu erstellen und die gewünschte Rolle (Lehrende oder Studierende) zu wählen. Wurde die Instanz der KEA-Mod-Plattform an einen Identitätsanbieter angebunden, ist die Anmeldung auch direkt über den Button »OpenID Login« möglich.

Nach erfolgreicher Anmeldung gelangt eine nutzende Person der Rolle »Lehrende« zur Übersichtsseite der Lehrenden UI auf welcher die bisher von ihr erstellten Aufgabenreihen (z. B. Übungsblätter oder Klausuren) aufgelistet werden. Neben der Bearbeitung der Aufgabenreihen können an dieser Stelle die studentischen Einreichungen zu diesen Aufgabenreihen betrachtet und gegebenenfalls manuell korrigiert werden. Ebenso ist das Anlegen neuer Aufgabenreihen möglich, wie in Abb. 5.10 dargestellt. Neben einem Namen und einer Beschreibung müssen hierbei auch ein Start- und Enddatum festgelegt werden, um den Bearbeitungszeitraum der Aufgabenreihe zu definieren. Der sechsstellige Zugriffscode für das Hinzufügen der Aufgabenreihe durch Studierende wird während der Erstellung der Aufgabenreihe automatisch

⁴⁶ Siehe z.B. die Übersicht auf der Webseite »Petri Nets World« der Universität Hamburg: https://www2.informatik.uni-hamburg.de/tgi/PetriNets/tools/complete_db.html

⁴⁷ <https://pm4py.fit.fraunhofer.de/>

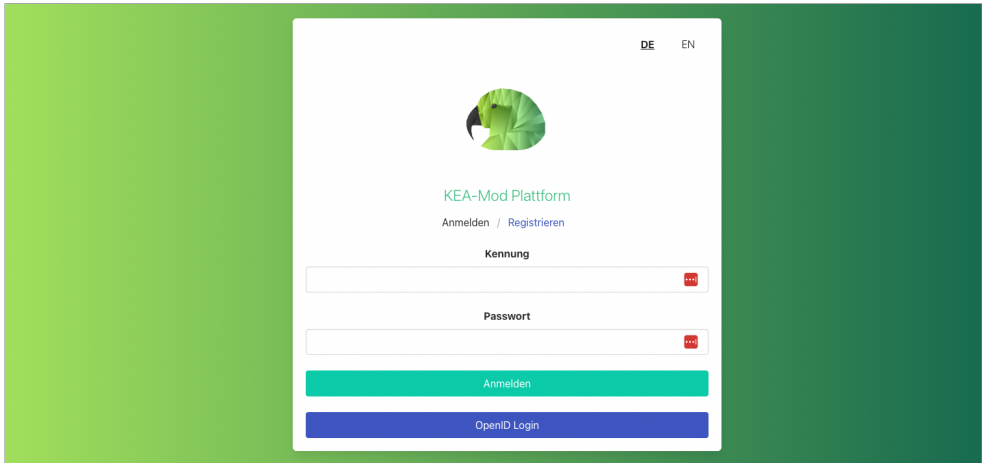


Abb. 5.9.: Die Anmeldung an der KEA-Mod-Plattform ist entweder mit einer bei der Registrierung frei gewählten Kennung oder über die Anbindung an den jeweiligen Identitätsanbieter (OpenID Login) möglich.

generiert, um zu gewährleisten, dass derselbe Zugriffscode nicht mehrmals vergeben wird. Anschließend können über den Button »Aufgaben erstellen« neue Aufgaben zu dieser Aufgabenreihe hinzugefügt werden. Dabei sind die beiden Aufgabentypen »Modell erstellen« und »Modell verstehen« verfügbar. Für den Aufgabentyp »Modell erstellen« ist die Angabe einer Aufgabenbeschreibung und eines textuellen Szenarios erforderlich. Zudem kann eine Modellierungssprache ausgewählt werden, um die verfügbaren Modellierungselemente im Modellierungseditor auf die entsprechende Palette einzuschränken. Die Auswahl des gewünschten Bewertungsschemas erfolgt über ein Drop-Down-Menü, ebenso ist es optional möglich, individuelle Texte zum Lernziel (Feed Up) sowie »Feed Forward«-Nachrichten anhand eines entsprechenden Schwellenwerts festzulegen. Darüber hinaus muss die maximal erreichbare Gesamtpunktzahl für die Aufgabe definiert werden. Zusätzlich können Musterlösungen hinzugefügt werden, die bei der Bewertung an die gewählten Bewertungsdienste gemäß des Bewertungsschemas weitergereicht werden. Die Konfiguration der gewünschten Art automatisierter Bewertung ist sowohl auf der Ebene der Aufgabenreihe (erfolgt dann bei Einreichung der gesamten Aufgabenreihe durch Studierende als sog. »outcome-Feedback«) als auch auf der Ebene einzelner Aufgaben (erfolgt dann beim Speichern einer Lösung zur jeweiligen Aufgabe durch Studierende als sog. »multiple-try-Feedback«) möglich (siehe auch Abschnitt 5.2.5). Analog dazu ist die Erstellung von Aufgaben des Typs »Modell verstehen« möglich. Dies erfordert

5. E-Assessment für das Gebiet der grafischen Modellierung

The image shows two screenshots of the KEA-Mod-Plattform interface. The top screenshot displays the configuration for a task series named 'Modellierung von Geschäftsprozessen'. It includes fields for Name, Beschreibung, Startdatum, and Enddatum. A tooltip explains that the 'Automatisierte Bewertung bei Einreichung' option, when activated, automatically evaluates all solutions for tasks in the series (Petri-Netze, EPK, BPMN, UML) as soon as a student submits a solution. The bottom screenshot shows the configuration for a specific task 'Kreditkartenantrag'. It details the evaluation schema (25% Syntax-Checker + 50% Semantik-Checker + 25% Pragmatik-Checker), learning objectives, and feedback settings. A tooltip explains that the 'Automatisierte Bewertung bei der Bearbeitung' option, when activated, automatically evaluates solutions for this task (Petri-Netze) every time a student saves a solution. The interface also features buttons for 'Aufgabe erstellen', 'Modell bearbeiten', and 'Modell löschen'.

Top Screenshot: Aufgabenreihen Konfiguration

Name: Modellierung von Geschäftsprozessen

Beschreibung: Mit dieser Aufgabenreihe soll... Petri-Netze, EPK, BPMN und UML-Aktivität

Startdatum: 01.10.2023, 09:00

Enddatum: 26.11.2023, 23:59

Automatisierte Bewertung bei Einreichung:

Zugriffscode: 413941

Aufgaben:

Name	Typ	Sprache	Optionen
Kreditkartenantrag	Modell erstellen	Petri Netz	Bearbeiten Löschen
Flugbuchung	Modell erstellen	EPK	Bearbeiten Löschen

Bottom Screenshot: Aufgabe Konfiguration

Bewertungsschema: 25% Syntax-Checker + 50 % Semantik-Checker + 25% Pragmatik-Checker

Lernziele (Feed-Up): Mithilfe dieser Aufgabe sollen Sie lernen, einen Geschäftsprozess in Form eines Petri-Netzes abzuzeichnen

Feed-Forward (positiv): Prima! Wir empfehlen Ihnen als nächstes, die weiteren Aufgaben in dieser Aufgabenreihe zu bearbeiten

Feed-Forward (negativ): Wir empfehlen Ihnen, sich zunächst mit den Grundlagen der Petri-Netze zu beschäftigen

Feed-Forward Schwellwert: 0.85

Automatisierte Bewertung bei der Bearbeitung:

Maximale Gesamtpunktzahl: 20

Musterlösung(en):

[Modell bearbeiten](#) [Modell löschen](#) [Modell bearbeiten](#) [Modell löschen](#)

Abb. 5.10.: Anlegen einer Aufgabenreihe (oben) und Konfiguration einer Aufgabe vom Typ »Modell erstellen« (unten) in der Lehrenden UI der KEA-Mod-Plattform. Jeweils hervorgehoben sind die Tooltips zur Beschreibung der verschiedenen Arten automatisierter Bewertung.

zunächst die Erstellung eines vorgegebenen Modells, gefolgt von der Eingabe der auszuwählenden Antwortoptionen.

Studierende gelangen nach der Anmeldung auf die Übersichtsseite der Studierenden UI (siehe Abb. 5.11, oben). Hier können unter Eingabe des sechsstelligen Zugriffs-codes neue Aufgabenreihen für die Bearbeitung hinzugefügt werden und werden zunächst unter »Offene Aufgabenreihen« angezeigt. Dort können sie – sofern der Bearbeitungszeitraum aktiv ist – bearbeitet werden. Nach erfolgter Einreichung einer Aufgabenreihe ist diese unter »Eingereichte Aufgabenreihen« zu sehen. Dort ist eine weitere Bearbeitung nicht möglich, jedoch können – sofern vorhanden – die Ergebnisse zu den eigenen Lösungen betrachtet werden. Innerhalb einer offenen Aufgabenreihe erhalten Studierende eine Übersicht über die enthaltenen Aufgaben und die jeweiligen Gesamtpunkte (siehe Abb 5.11, unten). Auch zusätzliche Informationen wie die Modellierungssprache, der Aufgabentyp und die adressierten Lernziele werden dabei dargestellt. Aufgaben können bearbeitet und – sofern vorhanden – auch die jeweiligen Ergebnisse betrachtet werden. Beim Bearbeiten einer Aufgabe vom Typ »Modell erstellen« wird der Modellierungseitor geöffnet und Aufgabenstellung und Szenario werden angezeigt (siehe Abb. 5.12). Dabei ist es möglich, die Fenster mit der Palette der verfügbaren Modellierungselemente sowie der Aufgabenstellung inklusive Szenario nach Bedarf zu vergrößern bzw. zu verkleinern und zu verschieben. Der Modellierungseitor bietet zusätzliche Funktionen wie z. B. Zoom-In/Out, Undo/Redo, Import und Export von Modellen im KEA-Format (siehe Abschnitt 5.1.4) und Anpassung von Einstellungsoptionen wie z.B. Hintergrundfarbe oder Rasterdarstellung. In der Palette der verfügbaren Modellierungselemente befinden sich neben den bekannten Petri-Netz-Modellierungselementen (siehe Abschnitt 2.2.4) auch Ovale und Rauten, sowie ungerichtete bzw. beidseitig gerichtete Kanten als sogenannte Distraktoren. Somit sind Studierende an dieser Stelle gefragt, bei der Modellierung die korrekten Modellierungselemente für Petri-Netze auszuwählen.

Abb. 5.13 zeigt die Darstellung von Feedback-Nachrichten und dem Bewertungsergebnis zu einer Aufgabe vom Typ »Modell erstellen«. Im oberen Bereich werden Aufgabenstellung und Szenario sowie das studentische Modell angezeigt. Im studentischen Modell werden diejenigen Feedback-Nachrichten, die sich auf spezifische Modellelemente beziehen, direkt in die Darstellung integriert. Dies ist anhand der Umrandung (grau bzw. rot) oder der Kantenfarbe (rot) zu erkennen. Die Farbe entspricht dabei dem jeweiligen Beurteilungstyp der Feedback-Nachricht (negativ: rot, neutral: grau und positiv: grün). Wird der Mauscursor auf ein derart markiertes Modellelement bewegt, werden alle Feedback-Nachrichten zu diesem links unten in

5. E-Assessment für das Gebiet der grafischen Modellierung

Aufgabenreihen

Füge eine Aufgabenreihe hinzu

Hier kannst du eine Aufgabenreihe deinem Account hinzufügen. Nutze dafür den für dich bereitgestellten 6-stelligen Zugriffscode.

Hier Zugriffscode eingeben Hinzufügen

Offene Aufgabenreihen

Hier findest du deine aktuell offenen Aufgabenreihen. Jede Aufgabenreihe kann mehrere Aufgaben enthalten. Du kannst die Bearbeitung starten, indem du eine Aufgabenreihe anklickst.

Name	Zugriffscode	Aufgaben	Zeit verbleibend	
Öffentlicher Aufgabenpool	424242	13	364 Tag(e)	Bearbeiten
Übungsblatt 1: Petri-Netz-Grundlagen	296024	3	abgelaufen	Bearbeiten
Modellierung von Geschäftsprozessen	413941	4	8 Tag(e)	Bearbeiten

Eingereichte Aufgabenreihen

Hier findest du deine bereits bearbeiteten Aufgabenreihen und kannst deine Lösungen zusammen mit Feedback einsehen.

Name	Zugriffscode	Einreichungsdatum	
Übungsblatt 2: Petri-Netz-Eigenschaften	805690	Mo., 27. Nov. 2023	Einsehen
EBSS Klausur	286625	Mo., 27. Nov. 2023	Einsehen

Modellierung von Geschäftsprozessen

Mit dieser Aufgabenreihe sollen die Modellierungssprachen Petri-Netze, EPK, BPMN und UML-Aktivitätsdiagramme für die Modellierung von Geschäftsprozessen einzusetzen. Dabei sollen Sie erkennen, welches die Vor- und Nachteile der jeweiligen Modellierungssprachen für diesen Einsatzzweck sind.

Name	Modellierungssprache	Aufgabentyp	Gesamtpunkte	Lernziele	
Kreditkartenantrag	Petri Netz	Modell erstellen	6.95 / 10		Bearbeiten Ergebnisse
Flugbuchung	EPK	Modell erstellen			Bearbeiten Ergebnisse
Reparaturabwicklung	BPMN	Modell erstellen			Bearbeiten Ergebnisse
Auslandsanerkennung	UML Aktivitätsdiagramm	Modell erstellen			Bearbeiten Ergebnisse
Aufgabenreihe gesamt:			16.4/40		

Einreichen

Mithilfe dieser Aufgabe sollen Sie lernen, einen Geschäftsprozess in Form eines Petri-Netzes abzubilden und dabei syntaktische, semantische und pragmatische Qualitätsaspekte zu erfüllen.

Abb. 5.11.: Verwaltung von Aufgabenreihen in der Studierenden UI (oben) und Betrachtung einer offenen Aufgabenreihe (unten).

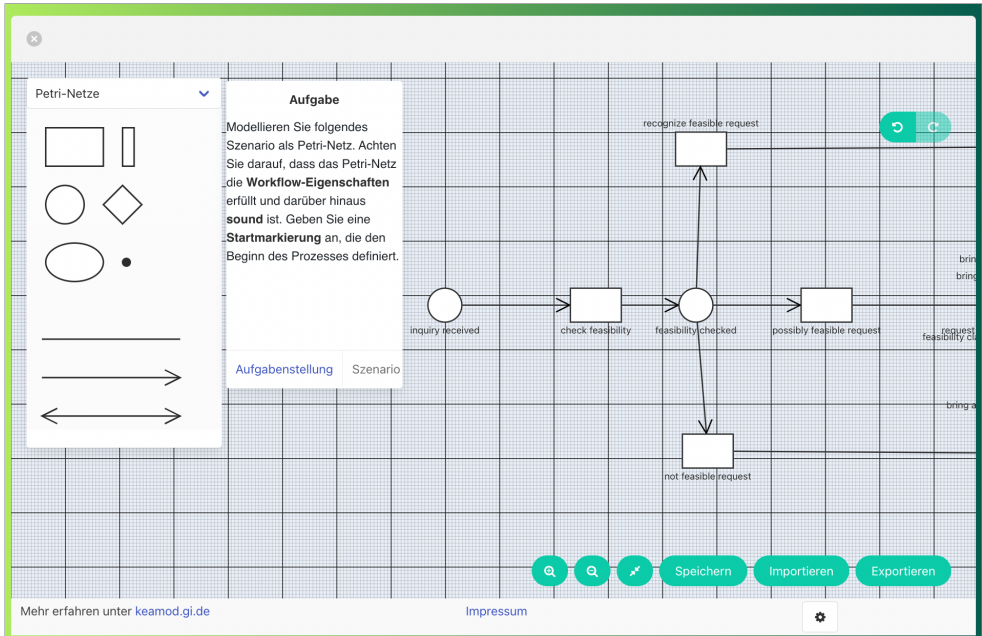


Abb. 5.12.: Bearbeitung einer Aufgabe vom Typ »Modell erstellen« mit dem in der KEA-Mod-Plattform integrierten Modellierungseditor.

einem Kästchen angezeigt. Der untere Bereich ist in drei Unterbereiche mit sonstigen Feedback-Nachrichten, der Bewertung sowie einem Vorschlag für nächste Schritte eingeteilt. Sonstige Feedback-Nachrichten zum studentischen Modell sind alle weiteren Feedback-Nachrichten, die sich auf das gesamte Modell und nicht auf einzelne Modellelemente beziehen. Diese werden anhand ihrer Kategorie gruppiert. Je nach Beurteilungstyp (positiv, negativ bzw. neutral) werden entsprechende Icons daneben angezeigt. Somit ist es auch möglich, Feedback-Nachrichten zur gleichen Kategorie durch unterschiedliche Bewertungsdienste erstellen zu lassen. Ebenso kann ein Bewertungsdienst Feedback-Nachrichten verschiedener Kategorien erzeugen. Bei der Darstellung der Bewertungsergebnisse sind neben der Bewertungsdauer und Gesamtpunktzahl auch die Teilergebnisse aller beteiligten Bewertungsdienste aufgeschlüsselt, um diese transparent zu gestalten. Die Anzeige des Unterbereichs »Nächste Schritte« erfolgt, wenn die optionalen Felder für Feed-Forward-Nachrichten bei der Aufgabenkonfiguration mit Werten belegt wurden (siehe auch Abb. 5.10).

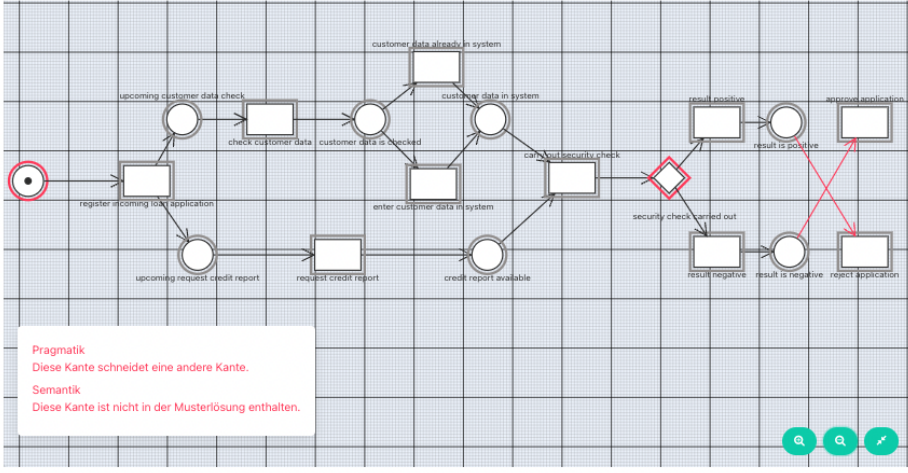
Ein Beispiel für die Anzeige der Ergebnisse für Aufgaben vom Typ »Modell verstehen« ist in Abb. 5.14 zu sehen. Hier besteht das Feedback darin, die gewählten

5. E-Assessment für das Gebiet der grafischen Modellierung

Feedback zur Aufgabe »Kreditkartenantrag«

Modellieren Sie folgendes Szenario als Petri-Netz. Achten Sie darauf, dass das Petri-Netz die **Workflow-Eigenschaften** erfüllt. Geben Sie eine **Startmarkierung** an, die den Beginn des Prozesses definiert und berücksichtigen Sie die Richtlinien zur Gestaltung von Geschäftsprozessen, die Sie aus der Übung kennen.

On an incoming loan application, the customer data is checked and a credit report is requested. If the customer data check shows that the customer data is not yet in the system, the customer data is entered in the system. As soon as the customer data is entered in the system and the credit report is available, a security check will be carried out. In case of a positive result of the security check, the application is approved; if the result of the security check is negative, the application is rejected. Afterwards, the loan application is closed.



Pragmatik
Diese Kante schneidet eine andere Kante.

Semantik
Diese Kante ist nicht in der Musterlösung enthalten.

Feedback

Syntax

- Das Modell besitzt einen Knoten eines Typs, der nicht für Petri-Netze vorgesehen ist.
- Alle Kanten im Modell entsprechen den für Petri-Netze vorgesehenen Kantenarten.
- Keine Kanten verbinden zwei Knoten desselben Typs.

Workflow-Netz

- Das Modell besitzt nicht genau eine Senke (2).
- Das Modell verletzt die Eigenschaften eines Workflow-Netzes.

Pragmatik

- Nicht alle Knoten sind beschriftet.
- Es sind Kantenüberschneidungen vorhanden.
- Anteil rechtsgerichteter Kanten (Flussrichtung): 100%

Semantik

- Versuchen Sie, das Modell anhand der gegebenen Hinweise weiter anzupassen!
- Es wurden 18 mit einer Musterlösung übereinstimmenden Knoten identifiziert.
- Das Modell enthält keinen Knoten »close loan application«.
- Das Modell enthält keine Kante von »result is positive« zu »approve application«.

Bewertung

Angelegt	Fertiggestellt	Bewertungsdienste	Anteil	Gesamtpunkte
Fr., 8. Dez. 2023, 11:27:09	Fr., 8. Dez. 2023, 11:27:11	3	3750 %	3.75 / 10

Bewertungsdienst	Ergebnis	Ergebnistyp	Gewichtung	Teilpunkte
PetriNet Syntax Checker	40 %	relativ	25 %	1 / 2.5
Universal Pragmatic Checker	20 %	relativ	25 %	0.5 / 2.5
Semantic Checker	45 %	relativ	50 %	2.25 / 5

Nächste Schritte

Berücksichtigen Sie zunächst die Feedback-Nachrichten. Bei Unsicherheiten können Sie in den Kursmaterialien zu Übung 2 die Richtlinien für die Gestaltung von Geschäftsprozessmodellen noch einmal ansehen.

Abb. 5.13.: Darstellung von Feedback-Nachrichten und Bewertungsergebnis zu einer Aufgabe vom Typ »Modell erstellen« in der KEA-Mod-Plattform. Im Textfeld links unter dem studentischen Modell ist die integrierte Feedback-Nachricht zur rotgefärbten Kante von der Stelle »result is positive« zum Knoten »reject application« zu sehen.

Feedback zur Aufgabe »Personenzug«

Betrachten Sie das gegebene Petri-Netz, welches das Verhalten eines Personenzugs beschreibt.

Bitte kreuzen Sie für jede der folgenden Aussagen an, ob diese richtig oder falsch sind. Wenn Sie sich nicht sicher sind, können Sie auch "keine Angabe" wählen, wenn Sie nicht raten möchten. Für jede korrekt klassifizierte Aussage erhalten Sie einen Punkt, für jede fehlerhaft klassifizierte Aussage wird Ihnen ein Punkt abgezogen. Sie können insgesamt nicht weniger als 0 Punkte für diese Aufgabe erzielen.

Es handelt sich um ein B/E-Netz	<input checked="" type="radio"/> Richtig	<input type="radio"/> Falsch	<input type="radio"/> Keine Antwort	✓
Der Zug kann nur fahren, wenn die Türen geschlossen sind	<input checked="" type="radio"/> Richtig	<input type="radio"/> Falsch	<input type="radio"/> Keine Antwort	✓
Die Türen können auch geöffnet werden, während der Zug fährt	<input type="radio"/> Richtig	<input type="radio"/> Falsch	<input checked="" type="radio"/> Keine Antwort	⚠
In einer Stelle des Netzes kann sich im weiteren Verlauf des Schaltens mehr als eine Marke befinden	<input checked="" type="radio"/> Richtig	<input type="radio"/> Falsch	<input type="radio"/> Keine Antwort	✗

Bewertung

Angelegt	Fertiggestellt	Bewertungsdienste	Anteil	Gesamtpunkte
Fr., 8. Dez. 2023, 16:15:42	Fr., 8. Dez. 2023, 16:15:42	1	25 %	1 / 4
<i>Bewertungsdienst</i>		<i>Ergebnis</i>	<i>Ergebnistyp</i>	
Multiple Choice		1	absolut	

Abb. 5.14.: Darstellung der Ergebnisse zu einer Aufgabe vom Typ »Modell verstehen« in der KEA-Mod-Plattform.

5. E-Assessment für das Gebiet der grafischen Modellierung

Antwortoptionen mit einem Haken (korrekte Option) oder einem Kreuz (falsche Option) zu markieren. Wurde die Antwortoption »Keine Antwort« gewählt, wird ein Fragezeichen angezeigt. Die Bewertung wird hier von einem Bewertungsdienst vorgenommen, der speziell für die Bewertung von Aufgaben des Typs »Modell verstehen« entwickelt wurde. Er liefert ein absolutes Bewertungsergebnis zurück, welches der erreichten Punktzahl entspricht. Die Bewertung von »Modell verstehen«-Aufgaben durch einen Bewertungsdienst statt durch die Kernanwendungen der KEA-Mod-Plattform sorgt für Flexibilität. Selbst für die Bewertung von Aufgaben im Antwort-Wahl-Verfahren bestehen zahlreiche Bewertungsstrategien⁴⁸ und unterschiedliche prüfungsrechtliche Vorgaben an den Hochschulen. Daher können individuelle Bewertungsstrategien in Form eines Bewertungsdienstes realisiert und an die Plattform angeschlossen werden, ohne Modifikationen an den Kernanwendungen vornehmen zu müssen.

In der Lehrenden UI der KEA-Mod-Plattform ist es möglich, die automatisierte Bewertung von studentischen Einreichungen zu einem beliebigen Zeitpunkt anzustoßen. Dabei können auch unterschiedliche Bewertungsschemata ausprobiert werden. Ebenso können die Ergebnisse der automatisierten Bewertung vor Veröffentlichung zunächst kontrolliert werden, was beim Einsatz in einer Prüfung erforderlich ist. Abb. 5.15 zeigt die Möglichkeiten zur manuellen Revision. Hierbei können die erzielte Punktzahl sowie die erreichbare Gesamtpunktzahl angepasst und Feedback-Nachrichten geändert oder gelöscht werden. Auch besteht die Möglichkeit, weitere Feedback-Nachrichten oder einen individuellen Kommentar zu ergänzen. Zusätzlich kann bestimmt werden, ob die Ergebnisse veröffentlicht werden sollen, um sicherzustellen, dass Studierende nur Ergebnisse sehen können, die bereits manuell kontrolliert wurden.

Sowohl in der Lehrenden als auch in der Studierenden UI ist der Modellierungsektor zum freien Modellieren ohne Aufgabenbezug auch direkt über die Navigationsleiste oben erreichbar (siehe Abb. 5.10 und Abb. 5.11). Ebenso kann die Sprache der Oberfläche auf Englisch bzw. Deutsch umgeschaltet werden. In der Rolle »Lehrende« kann zwischen der Studierenden und der Lehrenden UI hin- und hergewechselt werden. Dies ermöglicht es Lehrenden, ihre erstellten Aufgabenreihen vor dem Einsatz aus Studierendensicht zu erproben.

⁴⁸ Siehe z.B. die Dokumentation des Bewerbers für das Online-Übungssystem der Fernuniversität Hagen: <https://online-uebungssystem.fernuni-hagen.de/download/XausNBewertung/XausNBewertung.html>

The screenshot shows the 'Manuelle Revision' (Manual Revision) interface. At the top, there is a navigation bar with 'Aufgabenreihen', 'Übungsblatt 1: Petri-Netz-Grundlagen', 'Teststudent', and 'Aufgabe 3: Petri-Netz-Modellierung'. The main content area is titled 'Manuelle Revision' and contains a form with the following fields:

- Kommentar**: A text input field containing the text 'Schöne Lösung, leider wohl ein kleiner Flüchtigkeitsfehler :-)'
- Gesamtpunktzahl (relativ)**: A text input field containing the value '0.95'
- Maximale Gesamtpunktzahl**: A text input field containing the value '8'
- Ergebnisse veröffentlichen**: A checkbox that is currently unchecked.
- Feedback**: A blue button labeled 'Feedback erstellen'.

Below the form is a table with the following data:

Inhalt	Kategorie	Beurteilung	Bewertungsdienst	Optionen
Alle Kanten im Modell entsprechen den für Petri-Netze vorgesehenen Kantentypen.	Syntax	POSITIVE	PetriNet Syntax Checker	Bearbeiten Löschen
Keine Kanten verbinden zwei Knoten desselben Typs.	Syntax	POSITIVE	PetriNet Syntax Checker	Bearbeiten Löschen

Abb. 5.15.: Manuelle Revision von Ergebnissen zu einem studentischen Modell in der Lehrenden UI

5.5. Evaluation

In diesem Abschnitt werden die Ergebnisse vorgestellt, die während der Evaluation des Forschungsvorhabens erzielt wurden. Zur Evaluation der in dieser Arbeit beschriebenen Bewertungsdienste wurde eine Datenbasis mit studentischen Modellen in Form von Petri-Netzen erstellt (siehe Abschnitt 5.5.1). Anschließend wurden die Bewertungsdienste anhand eines Bewertungsschemas auf dieser Datenbasis angewendet. Die Ergebnisse werden in Abschnitt 5.5.2 vorgestellt. In den beiden folgenden Abschnitten wird der Bewertungsdienst Semantik-Checker genauer betrachtet. Für diesen wird zunächst die Zuverlässigkeit des Matching-Verfahrens ermittelt (Abschnitt 5.5.3) und anschließend eine Gegenüberstellung mit einer manuellen Bewertung vorgenommen (Abschnitt 5.5.4). Darüber hinaus wird eine kurze Betrachtung der Dauer der Bewertung in der KEA-Mod-Plattform vorgestellt (Abschnitt 5.5.5). In den Abschnitten 5.5.6 und 5.5.7 wird die Evaluation der KEA-Mod-Plattform durch Studierendenbefragungen im Rahmen eines Übungs- und Prüfungseinsatzes präsentiert.

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.7.: Überblick über die Datenbasis studentischer Modelle (Petri-Netze). HBM: Horus Business Modeler, moEd: früher Prototyp, KEA: KEA-Mod-Plattform

Aufgabe/Semester (WS)	16/17	17/18	18/19	19/20	20/21	21/22	22/23	Summe
A1: Kundenanfrage	180	196	180	178	222	219	0	1175
A2: Kreditantrag	0	77	178	177	222	0	0	654
A3: Mietwagen	0	99	0	0	0	182	84	365
Eingabewerkzeug	HBM	HBM	HBM	moEd	moEd	KEA	KEA	2.194

5.5.1. Datenbasis

Mit der Sammlung studentischer Modelle zu Forschungs- und Evaluationszwecken wurde im Wintersemester 2016/2017 begonnen. Einen Überblick über die seitdem gesammelten Modelle bietet Tab. 5.7. Alle Modelle sind im Rahmen der obligatorischen Rechnerübung⁴⁹ der Wahlpflichtveranstaltung »Programmierung kommerzieller Systeme - Einsatz betrieblicher Standardsoftware« am KIT entstanden. Diese findet jeweils im Wintersemester im Studiengang Wirtschaftsingenieurwesen statt und wird überwiegend von Studierenden im fünften Semester besucht. Die Gruppengröße liegt bei ca. 190-230 Studierenden. In dieser Lehrveranstaltung macht die Modellierung von Petri-Netzen, EPK und ER-Diagrammen einen Teil des Lehrinhalts aus. Da der Fokus dieser Dissertation auf Petri-Netzen liegt, wurden im Rahmen der Rechnerübung studentische Modelle zu Petri-Netz-Aufgaben in der Datenbasis gesammelt.

Über den Erhebungszeitraum hinweg wurden drei verschiedene Aufgaben A1-A3 eingesetzt. Es handelt sich hierbei um Aufgaben des Typs »Modell erstellen« zu verschiedenen Geschäftsprozessszenarien wie der Abwicklung einer Kundenanfrage (A1), der Beantragung eines Kredits (A2) und dem Mietwagenverleih (A3). Aufgabe A1 ist in Abb. 5.16 dargestellt, die beiden anderen Aufgaben sind im Anhang B in Abb. B.1 und B.2 zu finden. Da die entwickelten Bewertungsdienste für Modelle mit englischsprachigen Beschriftungen vorgesehen sind, wurde das Szenario bei allen drei Aufgaben in englischer Sprache verfasst. Dadurch wird eine größere Reichweite für die Anwendung der Bewertungsdienste im internationalen Raum ermöglicht und erlaubt den Einsatz bestehender NLP-Werkzeuge für die englische Sprache (siehe Abschnitte 3.3.1, 5.3.3 und 5.3.4).

Die Bearbeitung der Modellierungsaufgaben durch Studierende fand vom WS 16/17

⁴⁹ Das erfolgreiche Bestehen der Rechnerübung ist Voraussetzung für die Klausurzulassung.

Aufgabe A1: Modellieren Sie die folgende Prozessbeschreibung als Ablaufmodell in Form eines Petri-Netzes. Verwenden Sie für die Beschriftung der Modellelemente die englische Sprache.

›As soon as an inquiry from a customer is received, the feasibility is checked first. This check leads to three different results: either the request is not feasible, possibly feasible or feasible. If the request is not feasible, a rejection of the request is created and the customer is informed subsequently. If the request is possibly feasible, a clarification is brought about. The result of the clarification may be either positive or negative. In case of a negative clarification, a rejection is created and the customer is informed subsequently. In case of a positive clarification, an offer is created and the customer is informed subsequently. If the outcome of the feasibility check showed that the request is feasible, an offer is created and the customer is informed subsequently.‹

Musterlösung:

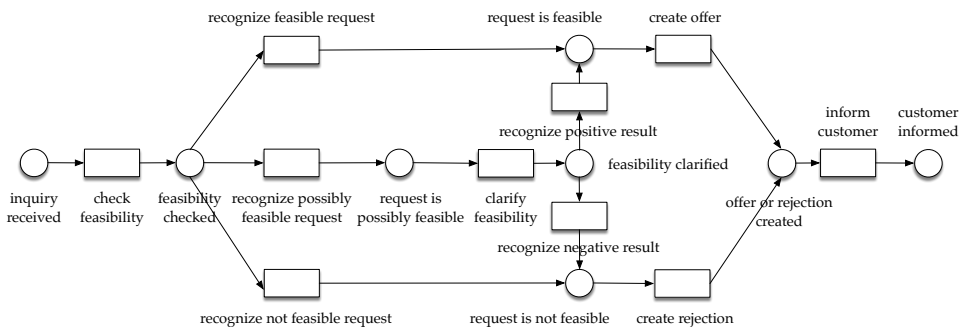


Abb. 5.16.: Aufgabe A1 (Abwicklung einer Kundenanfrage) mit Musterlösung

bis zum WS 19/20 an festgelegten Terminen in mehreren Poolräumen am KIT unter Aufsicht statt. Seit dem WS 20/21 wurden die Modellierungsaufgaben jedoch ortsunabhängig und ohne Aufsicht über die bereitgestellten Eingabewerkzeuge (beides Webanwendungen) bearbeitet und anschließend von Tutorinnen und Tutoren kontrolliert. Die Modelle wurden in diesem Kontext nicht bewertet, für das Bestehen der Rechnerübung wurde lediglich eine bemühte und sinnvolle Bearbeitung der Modellierungsaufgaben vorausgesetzt.

In den ersten drei Jahren der Datenerhebung wurde das Modellierungswerkzeug Horus Business Modeler⁵⁰ als Eingabewerkzeug verwendet, da dies in der Lehrveranstaltung auch für die Bearbeitung einer Fallstudie zum Einsatz kam und den Studierenden somit bereits bekannt war. Da es mit diesem Werkzeug jedoch nicht möglich

⁵⁰ <https://www.horus.biz/de/produkte/business-modeler/>

5. E-Assessment für das Gebiet der grafischen Modellierung

war, syntaktisch fehlerhafte Modelle zu erstellen und entsprechende Lernziele zu kontrollieren, wurde im WS 19/20 und 20/21 der frühe Prototyp für einen Modellierungseditor moEd (siehe Abschnitt 5.1.4) als Eingabewerkzeug eingesetzt. Mit der Fertigstellung des integrierten Modellierungseditors in der KEA-Mod-Plattform fand ab dem WS 21/22 ein erneuter Wechsel statt. Um alle studentischen Modelle in der Datenbasis mit den Bewertungsdiensten in der KEA-Mod-Plattform evaluieren zu können, wurden die mit dem Horus Business Modeler und moEd erstellten Modelle vom PNML bzw. moEd-Format in das KEA-Austauschformat konvertiert.

Alle Modelle der Datenbasis wurden auf der KEA-Mod-Webseite veröffentlicht⁵¹. Für die Benennung der Dateien wurde das Schema `Aufgabe_Semester_ID.ke` verwendet, so dass anhand des Dateinamens festgestellt werden kann, zu welcher Aufgabe und zu welchem Semester ein Modell gehört. Beispielsweise wurde das Modell in Datei `ex1_17_196.ke` im Wintersemester 17/18 zur Aufgabe A1 erstellt. Es wurden keinerlei personenbezogene Daten zu den Modellen gespeichert.

5.5.2. Ergebnisse unter Anwendung eines Bewertungsschemas

Für die drei Modellierungsaufgaben wurde das in Abschnitt 4.3 vorgeschlagene Rubric für die Bewertung von Aufgaben des Typs »Modell erstellen« herangezogen. Auf dessen Basis wurde ein Bewertungsschema entwickelt, welches auf die vorliegende Aufgabenstellung der Aufgaben A1-A3 zugeschnitten ist (siehe Tab. 5.8).

Dabei wurden diejenigen Kriterien aus dem Rubric ausgewählt, die sich anhand der studentischen Modelle in der Datenbasis mit den bestehenden Bewertungsdiensten im Kontext der spezifischen Aufgabenstellung überprüfen lassen. Ebenso wurden individuelle Gewichtungen (hier in Form von Punkten dargestellt) für die einzelnen Kategorien und die jeweiligen Kriterien festgelegt, die als Grundlage für die Berechnung der Teil- und Gesamtergebnisse verwendet werden. Die im folgenden beschriebenen Gewichtungsvorschläge können auf die individuellen Anforderungen angepasst werden. Sie sollen an dieser Stelle primär die unterschiedlichen Strategien für die Berechnung eines Teilergebnisses (z.B. Punktevergabe anhand Fehlerzahl, Abbildung eines Ergebnisses im Intervall $[0,1]$ auf festgelegte Werte, Gruppenbildung) demonstrieren, die anhand des Grundgerüsts der Bewertungsdienste (siehe Abschnitt 5.3.1) eingesetzt werden können.

51 https://docs.keamod.de/_media/Datenbasis_Evaluation.zip

Tab. 5.8.: Bewertungsschema für die Bewertung der Aufgaben des Typs »Modell erstellen« (Petri-Netze) in der Evaluation. BWD: Bewertungsdienst, SY-C: Syntax-Checker (Abschnitt 5.3.2), SE-C: Semantik-Checker (Abschnitt 5.3.3), P-C: Pragmatik-Checker (Abschnitt 5.3.4)

Kriterium	BWD	Methode(n)	Punkte	Punktevergabe bzw. Punktabzug
Kategorie Syntax - Korrekte Verwendung der Modellierungssprache (5 Punkte)				
Punktevergabe nur, wenn Modellgröße $\geq 25\%$ der durchschnittl. Größe der Musterlösungen.				
SYN01	SY-C	NodeTypesCorrect	1	0,5 Punkte Abzug pro Fehler
SYN02	SY-C	EdgeTypesCorrect + EdgesConnected	1	0,5 Punkte Abzug pro Fehler
SYN03	SY-C	DifferentTypesConnected	1	0,5 Punkte Abzug pro Fehler
SYN04	SY-C	WorkflowNet	2	0 Punkte bei Nicht-Erfüllung der WF-Netzeigenschaften
Kategorie Semantik- Inhaltliche Übereinstimmung mit Szenario der Aufgabe (10 Punkte)				
SEM	SE-C	ModelMatch	10	1 Punkt Abzug pro fehlendem Knoten, 0,5 Punkte Abzug pro zusätzlichem Knoten, 0,25 Punkte Abzug pro fehlender oder zusätzlicher Kante
Kategorie Pragmatik - Layout und Elementbeschriftungen (5 Punkte)				
Punktevergabe nur, wenn Modellgröße $\geq 25\%$ der durchschnittl. Größe der Musterlösungen.				
PRA01	P-C	NoEdgeCrossings	2	2 Punkte Abzug pro Fehler
PRA02	P-C	RightwardFlowDirection	1	Je nach Ergebnis x der Metrik wie folgt: 1 Punkt wenn $x \geq 0.9$, 0,5 Punkte wenn $0.9 < x \leq 0.7$, 0 Punkte sonst
PRA03	P-C	NodesLabeled	2	1 Punkt Abzug pro Fehler
Summe:			20	

Zunächst werden die Kategorien Syntax, Semantik und Pragmatik im Verhältnis 1:2:1 (5 Punkte, 10 Punkte, 5 Punkte) gewichtet, um einen Schwerpunkt auf die Kategorie Semantik zu legen. Für die Kriterien SYN01, SYN02, SYN03, PRA03 wurde eine obere Grenze von 2 für die Fehlerzahl festgelegt, um eine entsprechende Abstufung beim Punktabzug vorzunehmen (siehe Abschnitt 5.3.1). Für das Kriterium PRA01 wurde dahingegen eine obere Grenze von 1 festgelegt, um bereits beim Auftreten einer einzelnen Kantenüberschneidung keine Punkte für dieses Kriterium zu vergeben. Die bei Kriterium SYN04 eingesetzte Methode WorkflowNet des Petri Net Syntax Checkers liefert keine Fehlerzahl sondern nur einen Wahrheitswert zurück. Dementsprechend werden die für das Kriterium SYN04 festgelegten 2 Punkte entweder ganz vergeben oder nicht vergeben. Die Ergebnisse der Metrik PRA03 zur Bestimmung des Anteils der Kanten, die nicht nach rechts gerichtet sind, liegen im Intervall $[0, 1]$ und werden anhand zweier Schwellenwerte 0,9 und 0,7 auf drei Punktzahlen (1

5. E-Assessment für das Gebiet der grafischen Modellierung

Punkt sowie 0,5 und 0 Punkte) abgebildet (siehe auch Abschnitt 5.3.1). Für das Kriterium SEM wird die erreichte Punktzahl anhand der festgelegten Strafwerte für fehlende Knoten oder Kanten (C_{nf} und C_{ef}) bzw. zusätzliche Knoten oder Kanten (C_{ne} und C_{ke}) bestimmt (siehe Abschnitt 5.3.3). Sie wurden hier wie folgt festgelegt: $C_{nf} = 0, 1$, $C_{ef} = 0, 05$, $C_{ne} = C_{ke} = 0, 025$. Dies entspricht 1 Punkt sowie 0,5 bzw. 0,25 Punkten (siehe Tab. 5.8). Kriterium SYN02 demonstriert die Bündelung zweier Check-Methoden zum Aspekt Kanten zu einer gemeinsamen Gruppe. Bei der in der Tabelle beschriebenen Punktevergabe bzw. dem Punktabzug gilt in allen Fällen, dass Punkte nur abgezogen werden können, bis die untere Grenze von 0 Punkten für das jeweilige Kriterium erreicht wurde. Darüber hinausgehende Fehler bzw. Abzüge werden nicht berücksichtigt.

Das in Tab. 5.8 vorgestellte Bewertungsschema wurde mit einer entsprechenden Konfiguration der beteiligten Bewertungsdienste Syntax-Checker (SYN01-SYN04), Semantik-Checker (SEM) und Pragmatik-Checker (PRA01-PRA03) in die KEA-Mod-Plattform übertragen und anschließend alle 2.194 Modelle der Datenbasis damit bewertet. Das Ziel ist es hierbei, einen Überblick darüber zu erhalten, wie die studentischen Modelle in der Datenbasis in Bezug auf die ausgewählten Kriterien abschneiden. Bei der Interpretation der Ergebnisse muss berücksichtigt werden, dass einige der überprüften Kriterien in der ursprünglichen Aufgabenstellung (siehe z. B. Abb. 5.16) nicht gefordert wurden. Dabei handelt es sich um die drei Kriterien PRA01-03 der Kategorie Pragmatik, sowie um das Kriterium SYN04 bezüglich der Workflow-Netz-Eigenschaften. In einem künftigen Einsatz des Rubric im Zusammenhang mit den vorgestellten Aufgaben sollte die Aufgabenstellung fairerweise mit einem entsprechenden Zusatz versehen werden⁵², da diese Kriterien in die Gesamtbewertung einfließen⁵³.

Die folgende Tabelle 5.9 liefert einen Überblick über die anhand dieses Bewertungsschemas verteilten Punkte für alle Modelle der Datenbasis. Es zeigt sich, dass die Mehrheit der Modelle in den Kategorien Syntax (Ø: 4,31 P von 5 P) und Pragmatik (Ø: 4,53 P von 5 P) recht gut abschneidet, wobei berücksichtigt werden muss, dass syntaktische Fehler erst in den Modellen ab WS 2019/2020 ermöglicht wurden (siehe auch Tab. 5.10). Dafür erreichen die Modelle im Schnitt nur 2,70 P von 10 P in der Kategorie Semantik, die durchschnittlich erreichte Gesamtpunktzahl beträgt 11,53 P von 20 P. Das Bewertungsschema in Tab. 5.8 schreibt vor, dass eine Punktevergabe

⁵² Beispielsweise »Achten Sie darauf, dass das Petri-Netz die Workflow-Netz-Eigenschaften erfüllt und berücksichtigen Sie die aus der Vorlesung bekannten Konventionen in Bezug auf das Layout und die Elementbeschriftungen.« (Ggf. auch konkrete Aufführung der betreffenden Konventionen)

⁵³ Es ist alternativ auch möglich, Kriterien mit einer Gewichtung von 0 zu versehen. Studierende erhalten dann das entsprechende Feedback, Fehler führen jedoch nicht zum Punktabzug.

Tab. 5.9.: Übersicht über die erzielten Punkte aller 2.194 Modelle der Datenbasis. SD: Standardabweichung, med: Median.

Kriterium	Methode(n)	Ø	SD	min	med	max
	Kategorie Syntax (Gesamt)	4,31 P	0,98	0 P	5 P	5 P
SYN01	NodeTypesCorrect	0,99 P	0,09	0 P	1 P	1 P
SYN02	EdgeTypesCorrect + EdgesConnected	0,97 P	0,15	0 P	1 P	1 P
SYN03	DifferentTypesConn.	0,98 P	0,13	0 P	1 P	1 P
SYN04	WorkflowNet	1,36 P	0,93	0 P	2 P	2 P
SEM	ModelMatcher	2,70 P	3,11	0 P	1,75 P	10 P
	Kategorie Pragmatik (Gesamt)	4,53 P	0,85	0 P	5 P	5 P
PRA01	NoEdgeCrossings	1,80 P	0,60	0 P	2 P	2 P
PRA02	RightwardFlowDirection	0,85 P	0,29	0 P	1 P	1 P
PRA03	NodesLabeled	1,88 P	0,48	0 P	2 P	2 P
	Gesamtpunktzahl	11,53 P	3,57	0 P	10,75 P	20 P

nur dann erfolgt, wenn die Modellgröße $\geq 25\%$ der durchschnittlichen Größe der Musterlösungen ist (Nullifikation, siehe Abschnitt 5.3.2). Bei vier Modellen in der Datenbasis ist die Modellgröße tatsächlich $< 25\%$ der durchschnittlichen Größe der Musterlösungen. Bei diesen werden entsprechend der Forderung für die beiden Kategorien Syntax und Pragmatik nur 0 Punkte vergeben. In diesen vier Fällen konnte verhindert werden, dass diese Modelle zwischen 2,5 und 5 Punkten (Kategorie Syntax) und 3 und 5 Punkten (Kategorie Pragmatik erhalten), obwohl anhand des geringen Umfangs keine aussagekräftige Beurteilung dieser beiden Kategorien möglich ist.

In Tab. 5.10 werden diejenigen Modelle der Datenbasis genauer betrachtet, bei denen anhand der verwendeten Methoden vom Typ Check im Bewertungsschema eine Verletzung der Kriterien detektiert wurde. Eine Verletzung der Kriterien SYN01-SYN03 ist erst in den Modellen der Datenbasis möglich, die ab dem Wintersemester 2019/2020 erstellt wurden (siehe Abschnitt 5.5.1). Der Anteil fehlerhafter Modelle sollte daher nicht an der Gesamtanzahl aller Modelle gemessen werden, sondern anhand der Zahl der Modelle ab dem Wintersemester 2019/2020 (in Klammern dargestellt). Es zeigt sich deutlich, dass Fehler in Bezug auf die Syntax von Petri-Netzen selten auftreten und der überwiegende Anteil von Studierenden dazu in der Lage ist, die syntaktischen Basisregeln anhand der Kriterien SYN01-SYN03 für Petri-Netze einzuhalten. So enthalten nur 1,17% der Modelle fehlerhafte Knotentypen (Kriterium SYN01 - Methode NodeTypesCorrect) und 6,39% der Modelle fehlerhafte Kanten-

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.10.: Übersicht über die Modelle der Datenbasis, bei denen anhand der verwendeten Methoden vom Typ Check eine Verletzung der Kriterien detektiert wurde. Die Spalten \emptyset , SD: Standardabweichung, min, max, med: Median beziehen sich dabei auf die Fehlerzahl der jeweiligen Methode. #: Anzahl, FM: Fehlerhafte Modelle. Die Zahlen in der Spalte »Anteil FM« beziehen sich auf die Gesamtmenge von 2.194 Modellen im vollständigen Datensatz, die Zahl in Klammern auf die Teilmenge von 1.284 Modellen ab dem WS 19/20, die syntaktische Fehler enthalten können.

Kriterium	Methode	\emptyset	SD	min	med	max	# FM	Anteil FM
SYN01	NodeTypesCorrect	5,87	3,46	1	7	11	15	0,68% (1,17%)
SYN02	EdgeTypesCorrect	2,32	4,10	1	1	22	82	3,74% (6,39%)
SYN02	EdgesConnected	14,79	12,96	1	19	35	19	0,87% (1,48%)
SYN03	DifferentTypesConn.	1,35	0,61	1	1	3	31	1,41% (2,41%)
SYN04	WorkflowNet	-	-	-	-	-	698	31,81%
PRA01	NoEdgeCrossings	1,94	1,37	1	1	9	218	9,94%
PRA03	NodesLabeled	3,45	3,31	1	2	17	129	5,88%

typen (Kriterium SYN02 - Methode EdgeTypesCorrect). Allerdings kann dies auch darauf zurückzuführen sein, dass die Anzahl an Distraktoren in der Palette des Modellierungseditors für die Petri-Netz-Modellierung sehr gering ist (siehe Abschnitt 5.4). Neben Kreisen und Rechtecken werden lediglich Ovale (wurden teilweise anstelle von Kreisen verwendet) und Rauten (wurden für die Modellierung von XOR-Split bzw. -Join verwendet⁵⁴) als nicht korrekte Knotentypen angeboten. Neben gerichteten Kanten sind ungerichtete und beidseitig gerichtete Kanten als nicht korrekte Kantentypen vorhanden (siehe auch Abb. 5.12). So gibt es Modelle, die ausschließlich einseitig gerichtete Kanten verwenden (z. B. bei dem Modell mit der maximalen Fehlerzahl von 22 laut Tab. 5.10). Beidseitig gerichtete Kanten, die häufiger auftreten, wurden vor allem im Rahmen von Modellen zu Aufgabe A3 (siehe Abb. B.2 im Anhang) eingesetzt, um eine Schleife (korrekterweise durch zwei Kanten (s, t) und (t, s) abgebildet) darzustellen. Gleichzeitig muss den Studierenden ebenso zugute gehalten werden, dass anhand der Aufgabenstellung und der vorhandenen Anweisungen nicht erkennbar war, dass die Palette des Modellierungseditors Distraktoren enthielt. So sind einige Studierende möglicherweise davon ausgegangen, dass alle in der Palette vorhandenen Modellierungselemente in Petri-Netzen gültig sind.

Fehler im Kontext der Methode EdgesConnected, bei denen Kanten nicht mit zwei Knoten verbunden sind (Anteil fehlerhafter Modelle von 1,48%) können möglicher-

⁵⁴ Die Raute wird z. B. in UML-Aktivitätsdiagrammen und BPMN-Modellen derart verwendet, siehe Abschnitt 2.2.3.

weise auch auf Unkenntnisse in der Bedienung des Modellierungseeditors zurückgeführt werden. Es wurden Modelle registriert, bei welchen tatsächlich alle vorhandenen Kanten einen deutlichen Abstand zu den Knoten (die sie augenscheinlich verbinden sollen) besitzen (z. B. bei dem Modell mit der maximalen Fehlerzahl von 35 Fehlern laut Tab. 5.10). Zusätzlich sind die Werte für Durchschnitt und Median hier mit \emptyset : 14,79 und med: 19 vergleichsweise hoch, so dass es sich bei dem Großteil der Modelle eher nicht um vereinzelte Kanten handelt, die nicht vollständig verbunden wurden, da das Modell möglicherweise während der Bearbeitungszeit nicht fertiggestellt wurde. Eine geringe Anzahl von 31 Modellen (2,41%) verbindet Knoten gleichen Typs miteinander (Kriterium SYN03 - Methode `DifferentTypesConnected`). Dieser Fehler tritt meist jeweils nur einmal auf (\emptyset : 1,35, med: 1), maximal jedoch dreimal (max: 3).

Von den 2.194 Modellen erfüllen 698 (Anteil von 31,81%) nicht die Workflow-Netz-Eigenschaften (Kriterium SYN04)⁵⁵. Eine Auswertung der Feedback-Nachrichten zeigt, dass die Ursache dafür überwiegend eine falsche Anzahl Senken (603 Modelle) und Quellen (166 Modelle) ist. Bei 16 Modellen treten Senken auf, die nicht vom Typ Stelle sind, bei 8 Modellen werden Quellen eingesetzt, die nicht vom Typ Stelle sind. Nicht stark zusammenhängend sind lediglich 4 Modelle. Da die Erfüllung der Workflow-Netz-Eigenschaften allerdings wie oben beschrieben nicht in der Aufgabenstellung gefordert wurde, können daraus zum aktuellen Zeitpunkt keine Rückschlüsse über vorhandene bzw. nicht vorhandene Kompetenzen im Hinblick auf dieses Kriterium gezogen werden. Treten bei weiteren Durchführungen mit einer angepassten Aufgabenstellung weiterhin zahlreiche Abweichungen auf, kann dies ein Hinweis darauf sein, das Thema Workflow-Netze in der Lehre vertieft zu behandeln und durch Übungen zu festigen.

Auf den 1.496 Modellen, welche die Workflow-Netz-Eigenschaften erfüllen, wurde eine zusätzliche Überprüfung der Soundness-Eigenschaft mit dem Erreichbarkeits-Checker (siehe Abschnitt 5.3.5) durchgeführt. Die Ergebnisse sollten weitere Rückschlüsse über die studentischen Modelle ermöglichen, ohne dass sie in das Bewertungsschema einfließen, da die Einhaltung der Soundness-Eigenschaft in der Aufgabenstellung nicht gefordert wurde. Bei 22 Modellen (1,47%) wurde die Größengrenze von 10.000 Knoten plus Kanten bei der Erstellung des Erreichbarkeitsgraphen überschritten, 934 Modelle (62,34%) erfüllen alle drei Soundness-Kriterien. Von den übrigen 540 Modellen, welche mindestens eine der drei Soundness-Kriterien

⁵⁵ Da hier keine Fehler gezählt werden sondern nur das Vorliegen bzw. Nichtvorliegen der Workflow-Netz-Eigenschaften geprüft wird, sind die Spalten \emptyset , SD, min, max, med der Tabelle nicht ausgefüllt.

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.11.: Übersicht über die Ergebnisse der Methoden vom Typ Metrik auf der vollständigen Datenbasis. SD: Standardabweichung, med: Median.

Kriterium	Methode	Ø	SD	min	med	max
SEM	ModelMatcher	0,27	0,31	0	0,175	1
PRA02	RightwardFlowDirection	0,93	0,12	0	1	1

verletzen, besitzen 400 Modelle (74,07%) tote Transitionen, 398 Modelle (73,70%) verletzen die Eigenschaft *Completeness* und 139 Modelle (25,74%) die Eigenschaft *Proper Completion*. Somit könnte eine Behandlung der Soundness-Kriterien in der Lehre und im Kontext des Aufgabentyps »Modell erstellen« bei der Modellierung von Geschäftsprozessen mit Petri-Netzen angebracht sein.

218 Modelle (Anteil von 9,94%) in der Datenbasis weisen Kantenüberschneidungen auf (Kriterium PRA01), obwohl dies für alle drei eingesetzten Modellierungsaufgaben A1-A3 vermeidbar ist. In mindestens 50% der Fälle tritt nur eine Kantenüberschneidung auf (med: 1), es gibt aber auch mindestens ein Modell, welches 9 Kantenüberschneidungen aufweist (max: 9). Bezüglich Kriterium PRA03 zur Prüfung der Beschriftung von Knoten ist lediglich ein Anteil von 5,98% Modellen fehlerhaft. Da es sich bei der Methode PRA02 nicht um eine Methode vom Typ Check handelt, sondern um eine Metrik, wird die Darstellung der Ergebnisse durch Tab. 5.11 ergänzt. Anhand der Tabelle wird deutlich, dass die meisten Modelle einen überwiegenden Anteil an rechtsgerichteten Kanten aufweisen (Ø: 0,93) und mindestens 50% der Modelle ausschließlich rechtsgerichtete Kanten besitzen (med: 1). Die Ergebnisse zeigen, dass die große Mehrheit der Studierenden die Kriterien PRA01-PRA03 bereits erfolgreich umsetzt. Es wird angenommen, dass eine explizite Aufnahme dieser und möglicherweise weiterer Kriterien in Bezug auf den pragmatischen Qualitätsaspekt in ein Bewertungsschema in der Lehre in künftigen Durchführungen eine weitere Verbesserung der Ergebnisse bewirken kann und das Bewusstsein der Studierenden für die Relevanz dieses Aspekts stärkt. Dabei soll selbstverständlich vorausgesetzt werden, dass das Thema zuvor anhand entsprechender Richtlinien in der Lehre aufgegriffen wird und die Einhaltung der Kriterien explizit in der Aufgabenstellung gefordert wird.

Die Ergebnisse der Methode ModelMatch in Tab. 5.11 wurden direkt in entsprechende Punktzahlen überführt (siehe Tab. 5.9). Sie müssen mit Vorsicht interpretiert werden, da das Bewertungsverfahren des Semantik-Checkers nicht mit einer manuellen Punktevergabe vergleichbar ist. Ebenso besteht im Gegensatz zu den

anderen Kriterien des Bewertungsschemas Unsicherheit über die Korrektheit der Bewertungsergebnisse, da beim zugrundeliegenden Matching Fehler auftreten können. Die Ergebnisse können daher nur als grober Hinweis dienen und bedürfen im Einzelfall einer manuellen Überprüfung. Daher wird auf einer Stichprobe eine detaillierte Analyse zur Zuverlässigkeit des Matching-Verfahrens in Abschnitt 5.5.3 vorgestellt. Zusätzlich wird eine Gegenüberstellung des Bewertungsverfahrens des Semantik-Checkers und einer manuellen Korrektur durchgeführt (siehe Abschnitt 5.5.4).

5.5.3. Analyse des Matching-Verfahrens im Semantik-Checker

Wie auf dem Gebiet des Prozessmodell-Matching (siehe Abschnitt 3.3.1) üblich, sollen die Matching-Ergebnisse des Semantik-Checkers mit einem manuell erstellten Gold-Standard verglichen werden. Da es nicht möglich war, einen Gold-Standard für alle Modelle der umfangreichen Datenbasis zu erstellen, wurde eine zufällige Auswahl von 30 Modellen zu Aufgabe A1 und je 10 Modellen zu Aufgabe A2 und A3 aus jeweils allen vorhandenen Semestern getroffen⁵⁶. Bei der Auswahl wurde darauf geachtet, dass die Modelle der Stichprobe eine Bandbreite unterschiedlicher Gesamtpunktzahlen abbilden. Beispielsweise wurden 3 Modelle mit 20 Punkten und zwei Modelle mit 0 Punkten entnommen. Die restlichen Modelle verteilen sich auf den dazwischenliegenden Punktebereich. Die durchschnittlich erzielte Gesamtpunktzahl der Modelle der Stichprobe beträgt 11,82 Punkte (Standardabweichung: 4,58).

Für die ausgewählten 50 Modelle der Stichprobe wurde anschließend ein Gold-Standard erstellt. Dabei wurde für jedes studentische Modell ein manueller Vergleich mit der jeweils ähnlichsten Musterlösung vorgenommen. Dabei wurden übereinstimmende Knotenpaare als Matches festgelegt. Die Matches des Goldstandards befinden sich in der Menge Λ_{gold} , die Matches des Semantik-Checkers in der Menge Λ_{sc} . Auf dieser Basis wurde jeder Knoten n_s im studentischen Modell wie folgt klassifiziert:

- Richtig positiv: In Λ_{sc} und in Λ_{gold} befindet sich je ein Match, welches n_s auf den übereinstimmenden Knoten n_m der Musterlösung abbildet.
- Falsch positiv: In Λ_{sc} befindet sich ein Match, welches n_s auf einen Knoten n_m der Musterlösung abbildet, in Λ_{gold} befindet sich jedoch ein Match welches n_s auf einen anderen Knoten $n_{m'}$ der Musterlösung abbildet oder aber in Λ_{gold} befindet sich kein Match für den Knoten n_s .

⁵⁶ Zu Aufgabe A1 wurden mehr Modelle entnommen, da in der Datenbasis zu dieser Aufgabe die meisten studentischen Einreichungen vorhanden sind (siehe Tab. 5.5.1)

5. E-Assessment für das Gebiet der grafischen Modellierung

- Richtig negativ: Weder in Λ_{sc} noch in Λ_{gold} befindet sich ein Match für den Knoten n_s .
- Falsch negativ: In Λ_{sc} befindet sich kein Match für den Knoten n_s aber in Λ_{gold} befindet sich ein Match für den Knoten n_s .

Sei TP die Anzahl richtig positiv (engl. *true positive*) klassifizierter, FP die Anzahl falsch positiv klassifizierter, TN die Anzahl richtig negativ (engl. *true negative*) klassifizierter und FN die Anzahl falsch negativ klassifizierter Knoten des studentischen Modells. Auf dieser Basis können die Kennzahlen Precision, Recall und F-Score (siehe z. B. [An15]), die sich im Intervall $[0, 1]$ befinden, berechnet werden.

Die Precision gibt den Anteil richtig positiver Matches des Semantik-Checkers an allen Matches in Λ_{sc} an:

$$Precision = \frac{TP}{TP + FP}$$

Dahingegen drückt das Recall den Anteil richtig positiver Matches des Semantik-Checkers an allen im Gold-Standard befindlichen Matches Λ_{gold} aus:

$$Recall = \frac{TP}{TP + FN}$$

Um die beiden Werte zusammenzuführen, wird der F-Score als harmonisches Mittel aus Precision und Recall gebildet:

$$F\text{-Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Stimmen die Mengen Λ_{gold} und Λ_{sc} überein, ergibt sich ein F-Score von 1. Werden mehrere Precision, Recall- und F-Score-Werte zusammengerechnet, wird üblicherweise die Standardabweichung angegeben und bei der Bildung des Mittelwerts zwischen dem sogenannten Micro-Average (\emptyset -mic) und Macro-Average (\emptyset -mac) unterschieden. Der Macro-Average ist der Mittelwert über alle Precision, Recall bzw. F-Score-Werte der betrachteten Modelle. Beim Micro-Average hingegen werden die Werte TP und FP bzw. FN aller betrachteten Modelle summiert und in die oben angegebenen Formeln eingesetzt. Der Micro-Average berücksichtigt die Größe der untersuchten Modelle, so dass ein schlechtes Recall-Ergebnis bei einem kleinen Modell nur geringe Auswirkungen auf das Ergebnis des Micro-Average-Recall hat (siehe auch [An15]).

Für die Bestimmung der Menge an Matches Λ_{sc} wurde der Semantik-Checker mit

Tab. 5.12.: Precision, Recall und F-Score auf einer Stichprobe von 50 Modellen zu den drei Aufgaben A1-A3 der Datenbasis. SD: Standardabweichung.

Aufgabe	Precision			Recall			F-Score		
	Ø-mic	Ø-mac	SD	Ø-mic	Ø-mac	SD	Ø-mic	Ø-mac	SD
A1	0,98	0,98	0,04	0,97	0,98	0,06	0,98	0,98	0,04
A2	0,93	0,90	0,11	0,97	0,93	0,14	0,95	0,90	0,10
A3	0,88	0,85	0,15	0,87	0,86	0,28	0,88	0,84	0,24
Ø A1-A3	0,93	0,91	0,10	0,94	0,92	0,16	0,93	0,91	0,12

den folgenden Konfigurationsparametern ausgeführt, die experimentell⁵⁷ bestimmt wurden: $w_1 = 0,6$; $w_2 = w_3 = 0,1$; $w_4 = 0,2$; $\theta = 0,7$; $\Delta = 0,1$; $C_m = 1$. Zudem wurden für die Aufgaben A1 und A2 je zwei, für die Aufgabe A3 vier Musterlösungen eingesetzt, die geringfügig von den anderen Musterlösungen abweichende Alternativen für die Modellierung des jeweiligen Szenarios beinhalten. Alle Musterlösungen enthalten zudem Knoten, die mit Beschriftungsalternativen versehen wurden (siehe Abschnitt 5.3.3). Die Übersicht über die zusammengefassten Ergebnisse für Precision, Recall und F-Score ist in Tab. 5.12 dargestellt, die individuelle Auswertung aller 50 Modelle in Tab. B.1 befindet sich im Anhang.

Die Ergebnisse zeigen, dass das Matching-Verfahren des Semantik-Checkers mit einem über die drei Aufgaben gemittelten F-Score von 0,93 (Ø-mic) bzw. 0,91 (Ø-mac) größtenteils mit dem manuellen Matching im Gold-Standard übereinstimmt. Dabei enthält ein Modell im Schnitt 1,36 falsch positiv und 1,2 falsch negativ klassifizierte Knoten. Bei der Untersuchung der Stichprobenmodelle hat sich gezeigt, dass die studentischen Modelle zu Aufgabe A2 und insbesondere Aufgabe A3 stärker variierende Beschriftungen aufweisen und somit das Matching-Verfahren schlechter abschneidet, als bei Aufgabe A1. Dies ist darauf zurückzuführen, dass die textuelle Beschreibung des zu modellierenden Szenarios bei Aufgabe A1 deutlich präziser gestaltet ist als bei Aufgabe A2 und A3, so dass hier viele Formulierungen direkt aus dem Text übernommen werden können und sich kaum alternative Modellierungsmöglichkeiten ergeben.

Eine genauere Untersuchung der falsch positiv oder falsch negativ klassifizierten Knoten bei den Modellen der Stichprobe legt die folgenden Hauptursachen offen.

⁵⁷ Hierfür wurde das Matching auf einer Testmenge wiederholt ausgeführt und dabei die Gewichte w_1 - w_4 für die vier Metriken zur Bestimmung der Knotenähnlichkeit ausgehend von den aus [Sc18] bekannten Werten variiert. Ebenso wurden der Schwellenwert θ und der Maximalabstand zum besten Match Δ und die Maximalanzahl möglicher Matches C_m variiert. Mit den hier genannten Konfigurationsparametern konnte dabei der höchste F-Score erzielt werden.

5. E-Assessment für das Gebiet der grafischen Modellierung

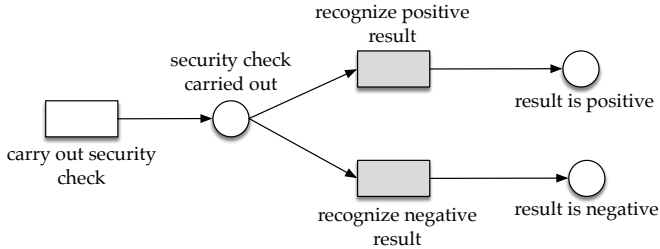


Abb. 5.17.: Ausschnitt aus der Musterlösung zu Aufgabe A2 (siehe Abb. B.1). Hilfstransitionen sind grau markiert.

Unpräzise Beschriftung von Hilfstransitionen: Zur Darstellung einiger Sachverhalte ist in Petri-Netzen die Verwendung von Hilfstransitionen notwendig, beispielsweise um für die Aktivität (Transition) »carry out security check« die alternativen Ergebnisse »result is positive« und »result is negative« in Form einer Stelle abbilden zu können⁵⁸, wie in Abb. 5.17 dargestellt. Die Beschriftung dieser Hilfstransitionen lässt sich nicht direkt aus der textuellen Beschreibung des Szenarios ableiten, so dass Studierende gefordert sind, selbst Beschriftungen zu entwickeln. Einige Modelle enthalten Hilfstransitionen mit Beschriftungen wie »H1« und »H2« (ex1_20_149.kea) oder »...(Hilfstransition)« (ex2_20_137.kea). Da für die Gewichtung der Beschriftungsähnlichkeit der Wert $w_1 = 0,6$ festgelegt wurde, kann mit den verbleibenden Ähnlichkeitskriterien der Schwellenwert von 0,7 nicht erreicht werden, um ein Matching zu erzielen. Eine mögliche Lösung könnte es sein, derartige Hilfstransitionen zuvor mit Hilfe von Heuristiken zu identifizieren und beim Matching dieser die Gewichtung der anderen Ähnlichkeitskriterien zu erhöhen. In anderen Fällen verwenden Studierende Hilfstransitionen mit etwas aussagekräftigeren Beschriftungen, wie z. B. »leads to result«. In Modell ex1_17_62.kea sind fünf Hilfstransitionen mit dieser Beschriftung vorhanden, davon konnten immerhin zwei korrekt zugeordnet werden. Diesbezüglich könnte im Aufgabentext explizit darauf hingewiesen werden, alle Knoten mit inhaltsgemäßen Beschriftungen zu versehen.

Keine Quelle vorhanden: In der Stichprobe befinden sich zwei Modelle, die keine Quelle besitzen (ex1_17_196.kea und ex3_21_110.kea)⁵⁹. In Abb. 5.18 ist ein entsprechender Ausschnitt aus Modell ex3_21_110.kea dargestellt. In einem solchen Fall greift die Bestimmung der relativen Position zum Ähnlichkeitskriterium sim_{pos} als Distanz zu einer Quelle im Modell nicht. Da die Gewichtung dieses Kri-

⁵⁸ Dies entspricht dem Kontrollflussmuster Exklusive Auswahl (WCP-4) nach der Workflow Pattern Initiative [Ru06], siehe Tab. 2.10 in Abschnitt 2.2.4.

⁵⁹ In der gesamten Datenbasis wurden 50 Modelle identifiziert, die keine Quelle besitzen.

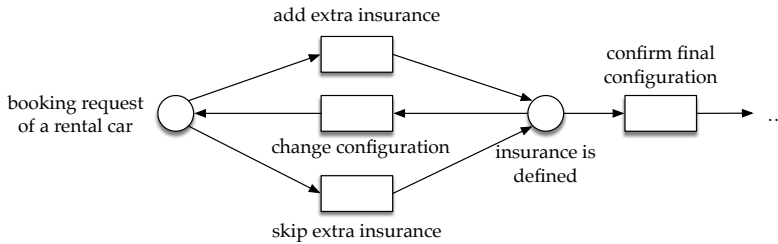


Abb. 5.18.: Ausschnitt aus Modell `ex3_21_110.kea`, welches keine Quelle besitzt.

teriums mit $w_2 = 0,1$ recht niedrig angesetzt ist, funktioniert die Zuordnung von Knoten über die anderen Ähnlichkeitskriterien zwar in den meisten Fällen weiterhin. Allerdings ist diese Information beim Auftreten von Aktivitäten mit gleicher Beschriftung an unterschiedlichen Stellen im Modell (z. B. »skip insurance« bei Aufgabe A3) von hoher Bedeutung, so dass hier falsch positive Matches entstehen. Die Matching-Ergebnisse könnten durch eine Anpassung des Ähnlichkeitskriteriums verbessert werden, indem in diesem Fall z. B. durch Heuristiken ein alternativer Knoten als Startknoten für die Berechnung der Distanz ausgewählt wird.

Negation innerhalb der Beschriftung: Der Semantik-Checker ist zwar dazu in der Lage, für einzelne Wörter in der Beschriftung auch Synonyme zu erkennen. In Modell `ex3_22_33.kea` wird jedoch die Beschriftung »do not add insurance« fälschlicherweise dem Knoten »add insurance« der Musterlösung zugeordnet anstatt dem Knoten »skip insurance« (falsch positiv). Da der Vergleich der Beschriftungen auf der Ebene einzelner Wörter durchgeführt wird, kann nicht erkannt werden, dass es sich durch die Negation bei »do not add« und »skip« um Synonyme handelt. Um dies zu ermöglichen, müsste der Semantik-Checker so erweitert werden, dass die Wörter der Beschriftungen auch in ihrem jeweiligen Kontext untersucht werden. In der Literatur finden sich Ansätze auf dem Gebiet des NLP für den Umgang mit der Negation, z. B. [MB21].

Niedrige Beschriftungsqualität: Bei vielen weiteren Knoten in studentischen Modellen, die nicht erfolgreich auf einen passenden Knoten in der jeweils betrachteten Musterlösung abgebildet werden konnten (falsch negativ), sind die Beschriftungen von niedriger Qualität. Beispiele dafür sind:

- Tippfehler: Der Knoten mit der Beschriftung »costumer gets checkt feasible« (`ex1_16_96.kea`) konnte nicht auf den Knoten »recognize feasible request« abgebildet werden.

5. E-Assessment für das Gebiet der grafischen Modellierung

- Kombination aus deutschen und englischen Wörtern: Das Modell `ex2_19_76.kea` enthält einen Knoten »daten is entered«, der nicht auf den Knoten »customer data in system« abgebildet werden konnte⁶⁰.
- Sonderzeichen: Bei allen Beschriftungen der Knoten im Modell `ex3_17_77.kea` wurde anstelle eines Leerzeichens das Zeichen »_« verwendet, z. B. »customer_check_in«, so dass keine Zuordnung zu den entsprechenden Knoten der Musterlösung möglich war. Dies liegt daran, dass der verwendete Tokenizer des Stanford Parsers dieses Zeichen nicht als Trennzeichen interpretiert, so dass keine Zerlegung der Beschriftung in einzelne Wortbestandteile möglich war.
- Unbeschriftete Knoten: Vier Modelle in der Stichprobe enthalten unbeschriftete Knoten (`ex1_21_125.kea`, `ex2_19_147.kea`, `ex2_19_76.kea` und `ex2_20_137.kea`), die zu falsch negativen Matching-Ergebnissen führen. Hierbei könnte ebenfalls der Lösungsvorschlag zum Einsatz kommen, der für den Umgang mit Hilfstransitionen skizziert wurde (höhere Gewichtung der anderen Ähnlichkeitskriterien).

Das vom Semantik-Checker erstellte Matching Λ_{sc} bildet die Grundlage für die Bestimmung von fehlenden und zusätzlichen Knoten sowie Kanten (siehe Abschnitt 5.3.3). Das bedeutet, dass fehlerhafte Matches $\lambda \in \Lambda_{sc}$ entsprechende Auswirkungen mit sich bringen. Im folgenden sollen je ein Beispiel für die Konsequenzen eines falsch positiven und eines falsch negativen Matches vorgestellt werden. Abb. 5.19 zeigt einen Ausschnitt aus Modell `ex1_19_22.kea`. Hier wurde der Knoten mit der Beschriftung »request possibly not being feasible« fälschlicherweise dem Knoten mit der Beschriftung »recognize not feasible request« anstatt »recognize possibly feasible request« zugeordnet (falsch positiv). Da es laut Musterlösung eine Kante vom Knoten »feasibility checked« zum Knoten »recognize not feasible request« gibt, wird die eingehende Kante nicht als zusätzliche Kante markiert. Dies ist allerdings bei der ausgehenden Kante der Fall, da es laut Musterlösung keine Kante vom Knoten »recognize not feasible request« zum Knoten »request is possibly feasible« gibt. Hieraus ergibt sich demnach eine zusätzliche Kante. Da im Modell `ex1_19_22.kea` kein Knoten gefunden wurde, welcher der Aktivität »recognize possible feasible request« entspricht, wird dieser Knoten als fehlender Knoten markiert und eine entsprechende Feedback-Nachricht erzeugt (siehe Abb. 5.19). Somit ergeben sich in diesem Beispiel aus dem falsch positiven Matching-Ergebnis ein fehlender Knoten und eine zusätzliche Kante.

⁶⁰ Knoten mit ähnlicher Beschriftung in englischer Sprache konnten erfolgreich abgebildet werden, z. B. der Knoten »data entered« im Modell `ex2_19_5.kea`.

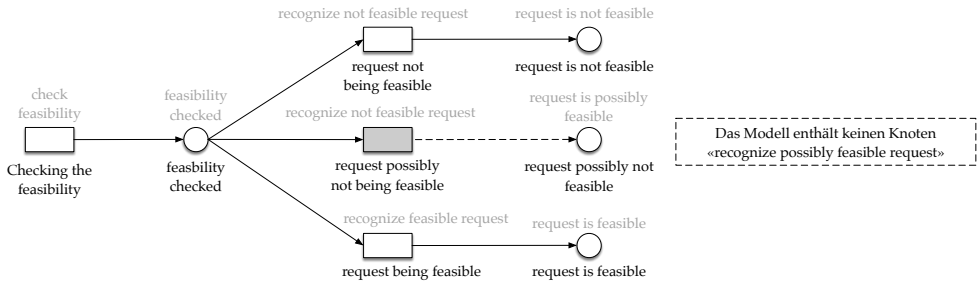


Abb. 5.19.: Ausschnitt aus Modell $ex1_19_22$. kea mit einem falsch zugeordneten Knoten (grau gefüllt). Beschriftungen in schwarzer Farbe: studentisches Modell, Beschriftungen in grauer Farbe: Zugeordneter Knoten der Musterlösung. Die entstandenen Folgefehler sind eine zusätzliche Kante (gestrichelt), sowie ein fehlender Knoten (Kasten mit gestrichelter Umrandung).

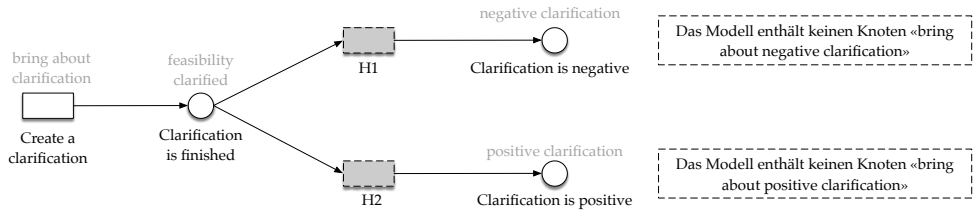


Abb. 5.20.: Ausschnitt aus Modell $ex1_20_149$. kea mit zwei falsch negativ klassifizierten Knoten (grau gefüllt). Beschriftungen in schwarzer Farbe: studentisches Modell, Beschriftungen in grauer Farbe: Zugeordneter Knoten der Musterlösung. Die entstandenen Folgefehler sind jeweils ein zusätzlicher Knoten (gestrichelte Umrandung) sowie ein fehlender Knoten (Kasten mit gestrichelter Umrandung).

Abb. 5.20 zeigt ein Beispiel für zwei falsch negativ klassifizierte Knoten anhand eines Ausschnitts aus Modell $ex1_20_149$. kea. Hier wurden die Knoten mit den Beschriftungen »H1« und »H2« (siehe die vorherigen Ausführungen zu Hilfstransitionen) keinem Knoten in der Musterlösung zugeordnet, obwohl es nach dem Gold-Standard entsprechende Knoten »bring about negative clarification« bzw. »bring about positive clarification« gibt. Da der Semantik-Checker vor der Bestimmung fehlender und zusätzlicher Kanten alle Knoten ohne Entsprechung im studentischen Modell bzw. ohne Entsprechung in der Musterlösung sowie deren ein- und ausgehende Kanten entfernt (siehe Abschnitt 5.3.3), können entsprechende Folgefehler in Bezug auf fehlende bzw. zusätzliche Kanten vermieden werden. Dennoch entsteht jeweils ein zusätzlicher Knoten (»H1« bzw. »H2«) und ein fehlender Knoten (»bring

5. E-Assessment für das Gebiet der grafischen Modellierung

about negative clarification« bzw. »bring about positive clarification«).

Auch wenn die automatisierte Bewertung mit Hilfe des Semantik-Checkers in wenigen Fällen falsch negativ bzw. falsch positive Matching-Ergebnisse produziert, können die daraus resultierenden Hinweise dennoch konstruktiv genutzt werden. Beispielsweise können Studierende in einem Selbstlern-Einsatz (» multiple-try«-Feedback, siehe Abschnitt 5.2.5) erkennen, wie sie ihr Modell schrittweise verbessern können, indem sie die Beschriftungen der betreffenden Knoten entsprechend der Feedback-Nachrichten anpassen. Vor Veröffentlichung der Ergebnisse im Prüfungskontext ist die manuelle Kontrolle der Ergebnisse der automatisierten Bewertung verpflichtend. Hierbei wäre es denkbar, dass die automatisiert erstellten Matching-Ergebnisse in einem Zwischenschritt ebenfalls manuell kontrolliert und gegebenenfalls korrigiert werden können, damit die daraus resultierenden Folgefehler vermieden werden können.

5.5.4. Vergleich manuelle Korrektur und Semantik-Checker

Der Semantik-Checker basiert auf der Übertragung eines Verfahrens zum Prozessmodell-Matching auf den Anwendungsbereich der automatisierten Bewertung. Infolgedessen werden durch den Semantik-Checker für ein studentisches Modell zu einer dazugehörigen Musterlösung bestehende Abweichungen in Form zusätzlicher bzw. fehlender Knoten und Kanten identifiziert. Bei der Bewertung im Rahmen einer manuellen Korrektur werden Abweichungen von einer Musterlösung zum studentischen Modell nicht in dieser Form gezählt, sondern in der Regel so aggregiert, dass sie sich auf einen einzelnen Fehler beziehen. Dies führt neben den gelegentlichen Schwierigkeiten beim Matching (siehe Abschnitt 5.5.3) dazu, dass die erreichte Punktzahl der Kategorie Semantik bei der automatisierten Bewertung durch den Semantik-Checker niedriger ist (2,7 P von 10 P, siehe Tab. 5.9), als es bei einer manuellen Korrektur erwartet würde.

Abb. 5.21 zeigt einen beispielhaften Ausschnitt aus einem studentischen Modell, bei welchem bei der manuellen Korrektur zwei Fehler (I und II) identifiziert wurden. Beide Fehler beziehen sich darauf, dass anstelle eines AND- ein XOR-Split verwendet wird. Zu Fehler I meldet der Semantik-Checker vier fehlende Knoten sowie drei zusätzliche Kanten, für Fehler II ergeben sich drei fehlende Knoten und zwei zusätzliche Kanten. Bei einer manuellen Korrektur werden diese Abweichungen zusammengefasst und als ein bzw. zwei Fehler gezählt. Dabei wird auch festgehalten, was ursächlich dahintersteht, nämlich die Verwendung eines falschen Kontrollflussmusters. Obwohl der Semantik-Checker diesen finalen Abstraktionsschritt nicht

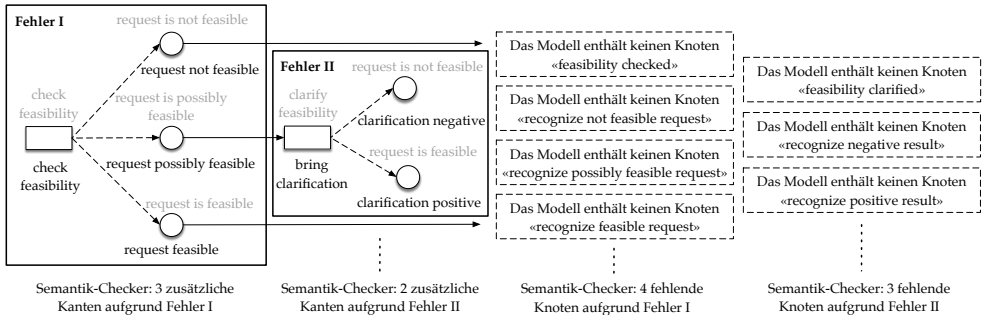


Abb. 5.21.: Ausschnitt aus Modell `ex1_18_1.ke` mit zwei Fehlern I und II. Beschriftungen in schwarzer Farbe: studentisches Modell, Beschriftungen in grauer Farbe: Zugeordneter Knoten der Musterlösung. Vom Semantik-Checker werden zusätzliche Kanten (gestrichelt) sowie fehlende Knoten (Feedback-Nachricht mit gestrichelter Umrandung) zurückgemeldet.

leistet, zeigt das Beispiel auf, dass die Rückmeldungen des Semantik-Checkers dennoch eine Unterstützung bieten. Sowohl bei der Korrektur durch Lehrende oder im Selbststudium durch Studierende werden bei der Betrachtung der Ergebnisse die entsprechenden fehlenden bzw. zusätzlichen Elemente im Modell angezeigt und somit auf den Fehler hingewiesen.

In diesem Abschnitt soll gezeigt werden, welche Fehler in Bezug auf die Kategorie Semantik bei der manuellen Korrektur auffallen und ob diese auch anhand der Rückmeldungen des Semantik-Checkers erkennbar werden, wie im obigen Beispiel verdeutlicht. Als Grundlage für die manuelle Korrektur dient das entwickelte Rubric für den Aufgabentyp »Modell erstellen« (siehe Abschnitt 4.3). Daraus ergeben sich für die Kategorie Semantik primär die Kriterien Vollständigkeit (SEM01), Validität (SEM02) und Prägnanz (SEM03). Ebenso sollte die Wahl eines passenden Abstraktionsgrads (SEM04) als Kriterium berücksichtigt werden. In einer früheren Publikation [Fe16] wurde bereits ein Bewertungsschema für die manuelle Bewertung von EPK-Modellen vorgestellt. Die dort beschriebenen Kriterien für die Kategorie Semantik lassen sich mit diesen vier Kriterien vereinbaren⁶¹, was die Anwendbarkeit dieser Kriterien für die Korrektur von Aufgaben des Typs »Modell erstellen« bestätigt. Für die manuelle Korrektur wurden anschließend entsprechende Bewertungsvorgaben abgeleitet, die in Tab. 5.13 aufgelistet werden.

⁶¹ Lediglich das Kriterium SEM01 (Sprachliche Korrektheit bzw. konventionsgemäße Beschriftung) aus [Fe16] wird im Kontext dieser Arbeit der Kategorie Pragmatik zugeordnet. Das Kriterium SEM02 (Semantische Korrektheit) aus [Fe16] fasst die in diesem Abschnitt als SEM02 und SEM03 bezeichneten Kriterien zu einem einzelnen Kriterium zusammen.

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.13.: Vorgaben für die manuelle Korrektur und Bewertung in Bezug auf die Kategorie Semantik

Kriterium	Abzug	Verletzung und detaillierte Bewertungsanweisung
SEM01 (Vollständigkeit)	1 P	Im Modell fehlen Knoten, die im Szenario explizit beschrieben wurden. Fehlt ein Knoten nicht im gesamten Modell, sondern nur in einem bestimmten Pfad, wird dies als Fehler in Bezug auf SEM02 und nicht SEM01 gewertet. Kein erneuter Punktabzug wenn durch einen fehlenden Knoten auch die damit verbundenen Knoten im Vor- bzw. Nachbereich fehlen.
SEM02 (Validität)	1 P	Im Modell werden Inhalte des Szenarios falsch umgesetzt, beispielsweise durch die Verwendung falscher XOR- oder AND-Splits bzw. Joins oder durch eine falsche Reihenfolge der Knoten. Fehlende Knoten werden als Verletzung des Kriteriums SEM01 betrachtet, wenn der Knoten im gesamten Modell fehlt. Zusätzliche Knoten werden als Verletzung des Kriteriums SEM03 betrachtet.
SEM03 (Prägnanz)	0,5 P	Im Modell sind Knoten vorhanden, deren Inhalt der Beschriftung und Position nach nicht mit dem Szenario in Verbindung gebracht werden kann. Davon ausgenommen sind Hilfsknoten. Es folgt kein erneuter Punktabzug in Bezug auf Kriterium SEM02.
SEM04 (Abstraktion)	0,5 P	Im Modell treten Knoten mit gleicher Beschriftung mehrfach auf, obwohl dies durch eine Zusammenführung der Pfade hätte vermieden werden können. Kein erneuter Abzug, wenn durch einen nicht zusammengeführten Knoten auch die Knoten im Vor- und Nachbereich mehrfach auftreten.

Die in Abschnitt 5.5.3 vorgestellte Stichprobe bestehend aus 50 studentischen Modellen der Datenbasis wurde sodann anhand dieser Vorgaben manuell korrigiert. Für jedes studentische Modell wurde dabei notiert, wie viele Verletzungen der Kriterien jeweils identifiziert werden konnten. Anschließend wurde die erreichte Punktzahl bestimmt, indem für alle Verletzungen der in der Tabelle vorgegebene Abzug von der maximal erreichbaren Teilpunktzahl von 10 Punkten (siehe Bewertungsschema in Abschnitt 5.5.2) durchgeführt wurde. Wurde dabei eine negative Punktzahl erzielt, wurde die erreichte Punktzahl auf den Wert 0 gesetzt. Anschließend wurde untersucht, auf wie viele dieser manuell bestimmten Kriterienverletzungen anhand fehlender und zusätzlicher Knoten sowie Kanten auch durch den Semantik-Checker hingewiesen wurde (wie in Abb. 5.21 dargestellt).

Die vollständigen Ergebnisse für die einzelnen studentischen Modelle der Stichprobe sind in Tab. B.2 im Anhang zu finden, Tab. 5.14 zeigt einen aggregierten Überblick. Der angegebene Anteil bezieht sich auf den Anteil der Fehler, auf welche der Semantik-Checker konstruktive Hinweise liefert, gemessen an der Anzahl der bei der manuellen Korrektur identifizierten Fehler. Für die Kriterien SEM01 und SEM03 ist dieser Anteil mit 93% bzw. 90% recht hoch. Die Abweichung ist dabei auf einzelne Matching-Fehler (siehe Abschnitt 5.5.3) zurückzuführen.

Tab. 5.14.: Gegenüberstellung manuelle Korrektur (M) und Semantik-Checker (S) auf einer Stichprobe von 50 Modellen im Hinblick auf die jeweils identifizierten Verletzungen der Semantik-Kriterien SEM01-SEM04.

	SEM01		SEM02		SEM03		SEM04		Teilpunktzahl (max. 10P)	
	M	S	M	S	M	S	M	S	M	S
Summe	89	83	76	56	31	28	72	0	-	-
Ø	1,78	1,66	1,52	1,12	0,62	0,56	1,44	0	5,78 P	3,54 P
Anteil	93%		74%		90%		0%		-	-

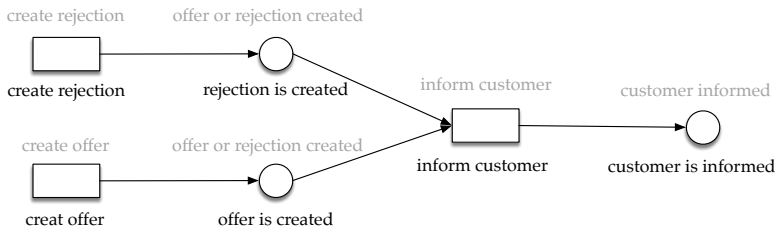


Abb. 5.22.: Ausschnitt aus Modell `ex1_16_73.ke.a`. Beschriftungen in schwarzer Farbe: studentisches Modell, Beschriftungen in grauer Farbe: Zugeordneter Knoten der Musterlösung. Hier wird ein AND-Join anstelle eines XOR-Joins verwendet, was vom Semantik-Checker nicht bemängelt wird.

Bei Kriterium SEM02 weist der Semantik-Checker lediglich auf einen Anteil von 74% der manuell identifizierten Fehler hin. Dabei konnten fehlerhafte Split-Konstrukte im Kontrollfluss erfolgreich identifiziert werden (siehe Abb. 5.21). Jedoch wurden aufgrund der Arbeitsweise des Semantik-Checkers fehlerhafte Join-Konstrukte nicht aufgezeigt, wie z. B. in Modell `ex1_16_73.ke.a` (siehe Abb. 5.22). In diesem Ausschnitt wird ein AND-Join anstelle eines XOR-Joins (siehe Musterlösung in Abb. 5.16) verwendet. Aufgrund des $n : 1$ -Matching bei der Ausführung des Semantik-Checkers wurden beide Knoten »rejection is created« und »offer is created« dem Knoten »offer or rejection created« der Musterlösung zugeordnet. Da sich in der Musterlösung eine Kante vom Knoten »offer or rejection created« zum Knoten »inform customer« befindet, handelt es sich bei beiden Kanten von den Knoten »rejection is created« und »offer is created« zum Knoten »inform customer« um korrekte Kanten. Folglich werden in dieser Situation durch den Semantik-Checker keine Abweichungen von der Musterlösung detektiert.

Tritt in Bezug auf SEM02 eine fehlerhafte Reihenfolge von Aktivitäten auf, so wird dies vom Semantik-Checker anhand zusätzlicher Kanten markiert, wie in Abb. 5.23

5. E-Assessment für das Gebiet der grafischen Modellierung

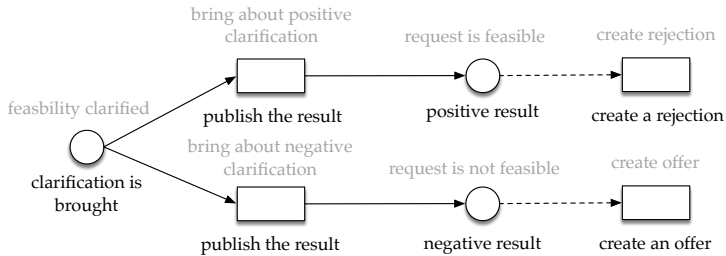


Abb. 5.23.: Ausschnitt aus Modell `ex1_17_62.ke.a`. Beschriftungen in schwarzer Farbe: studentisches Modell, Beschriftungen in grauer Farbe: Zugeordneter Knoten der Musterlösung. Zwei Kanten werden aufgrund der von der Musterlösung abweichenden Reihenfolge vom Semantik-Checker als zusätzliche Kante (grau gestrichelt) markiert.

dargestellt. Dieses Beispiel dient als anekdotischer Beleg für einen Fehler, der bei der manuellen Korrektur zunächst übersehen und erst durch die Hinweise des Semantik-Checkers erkannt wurde. Hier folgt fälschlicherweise die Aktivität »create a rejection« auf die Feststellung eines positiven Ergebnisses (Knoten »positive result«) und die Aktivität »create an offer« auf die Feststellung eines negativen Ergebnisses (Knoten »negative result«). Da es in der Musterlösung zwischen den jeweiligen Knoten keine Kante gibt, werden die entsprechenden Kanten vom Semantik-Checker als zusätzliche Kanten klassifiziert, womit auf den Fehler hingewiesen wird (siehe auch Abb. 5.13).

Das Kriterium SEM04 wurde durch den Semantik-Checker bislang nicht berücksichtigt, so dass hier keine entsprechenden Fehler identifiziert wurden. Die Ergebnisse zu den Kriterien SEM02 und SEM04 legen nahe, den Semantik-Checker in der künftigen Weiterentwicklung so anzupassen, dass ein 1 : 1-Matching durchgeführt werden kann. Dadurch würden auch Fehler in Bezug auf diese Kriterien im studentischen Modell hervorgehoben werden können.

Die durchschnittlich erzielte Teilpunktzahl von 5,78 P bei der manuellen Korrektur ist deutlich höher als die durchschnittlich erzielte Teilpunktzahl von 3,54 P, die durch den Semantik-Checker ermittelt wurde (siehe Tab. 5.14). Obwohl sich die erzielten Punkte durch die Wahl der Gewichtungen (Höhe des Punktabzugs bei der manuellen Korrektur und Höhe des Abzugs für fehlende und zusätzliche Knoten sowie Kanten beim Semantik-Checker) weiter annähern ließen, ist dies aufgrund der fundamental unterschiedlichen Bewertungsverfahren nicht zielführend. Im Beispiel aus Abb. 5.21 werden die zwei manuell identifizierten Fehler vom Typ SEM02 mit einem Abzug von jeweils 1 P bestraft, in Summe entstehen also 2 P Abzug. Diese beiden Fehler führen beim Semantik-Checker zu 7 fehlenden Knoten (7 P Abzug)

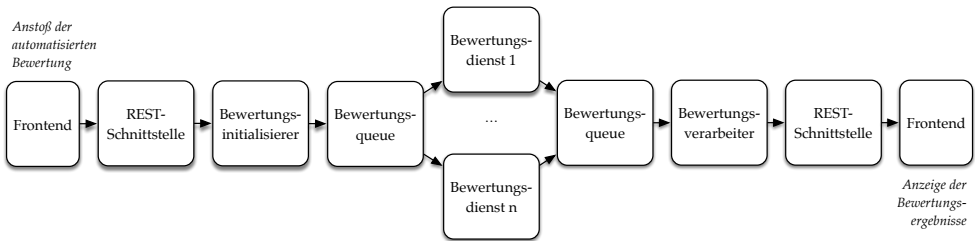


Abb. 5.24.: Abwicklung einer automatisierten Bewertung in der KEA-Mod-Plattform über die beteiligten Plattform-Komponenten hinweg

und fünf zusätzlichen Kanten (1,25 P Abzug), so dass sich in Summe 8,25 Punkte Abzug ergeben. Darüber hinaus entstehen beim Semantik-Checker weitere Abzüge durch ggf. auftretende Matching-Fehler (siehe Abschnitt 5.5.3). Dies verdeutlicht, warum die durch den Semantik-Checker berechneten Teilpunktzahlen niedriger ausfallen, als bei der manuellen Korrektur. Beim Einsatz der KEA-Mod-Plattform in einem summativen Kontext, beispielsweise bei einer elektronischen Klausur, muss nach geltendem Recht die erreichte Punktzahl ohnehin durch die prüfende Person kontrolliert werden. Während die Rückmeldungen des Semantik-Checkers bei der Identifikation der Fehler eine Hilfestellung leisten, sollte die erreichte Punktzahl dabei entsprechend eines Bewertungsschemas für die manuelle Korrektur (z. B. wie in Tab. 5.13) angepasst werden.

5.5.5. Dauer der automatisierten Bewertung

In diesem Abschnitt soll die Dauer der automatisierten Bewertung in der KEA-Mod-Plattform kurz beleuchtet werden. Im Projektkontext wurde auf die Festlegung strenger Anforderungen in Bezug auf die Laufzeit der Bewertungsdienste verzichtet. Jedoch sollte die Wartezeit auf das Ergebnis der automatisierten Bewertung im Hinblick auf die Benutzungsfreundlichkeit möglichst kurz (d.h. nicht länger als ca. 5 Sekunden) gehalten werden, um keine beeinträchtigenden Verzögerungen bei der Bedienung zu verursachen.

Anhand der Microservice-Architektur (siehe Abschnitt 5.1.1) ergibt sich die Wartezeit auf das Ergebnis der automatisierten Bewertung nicht nur aus der Laufzeit der beteiligten Bewertungsdienste, sondern zusätzlich aus den Kommunikationsschritten zwischen den beteiligten Plattform-Komponenten. So werden vom Start einer Bewertung bis zur Anzeige der Bewertungsergebnisse im Frontend etliche Zwischenstationen durchlaufen, wie in Abb. 5.24 dargestellt.

5. E-Assessment für das Gebiet der grafischen Modellierung

Da über die REST-Schnittstelle in der Datenbank sekundengenaue Zeitstempel für die Erstellung und Aktualisierung einer automatisierten Bewertung festgehalten werden, können diese Zeitstempel für eine grobe Auswertung herangezogen werden. Für die Ermittlung der Dauer der automatisierten Bewertung wurde die KEA-Mod-Plattform auf einer virtuellen Maschine mit 4 VCPUs und 8 GB RAM instanziiert. Dabei wurde bei einer Betrachtung der Bewertungsvorgänge für die Evaluation der 50 Modelle aus der Stichprobe unter Verwendung des Syntax-Checkers eine Bewertungsdauer zwischen 0 und 2 Sekunden festgestellt. Wurden der Syntax-Checker, der Semantik-Checker und der Pragmatik-Checker zur Bewertung eingesetzt, konnte eine Bewertungsdauer zwischen 1 und 3 Sekunden ermittelt werden. Dies ist für den vorliegenden Anwendungsfall als akzeptabel zu erachten. Weiterhin sind im Übungseinsatz der KEA-Mod-Plattform über die offenen Angaben des Online-Fragebogens keine negativen Rückmeldungen in Bezug auf die Dauer der automatisierten Bewertung eingegangen.

Sollte die Dauer der automatisierten Bewertung in einem künftigen Anwendungsfall stärker reduziert werden, ist anhand der Microservice-Architektur auch eine horizontale Skalierung denkbar. Beispielsweise können für besonders zeitkritische Bewertungsdienste bei Bedarf zusätzliche Instanzen gestartet werden (Service-Replikation), um die Last zu bewältigen. Die Lastverteilung (Load-Balancing) erfolgt dann durch die Bewertungsqueue.

5.5.6. Plattform-Evaluation im Übungseinsatz

Während der Laufzeit des Projekts KEA-Mod wurde die KEA-Mod-Plattform zur Pilotierung in Übungen mehrerer Lehrveranstaltungen der Verbundpartner eingesetzt und dabei begleitend evaluiert. Ziel der Evaluation war es, die Benutzungsfreundlichkeit, die wahrgenommene Nützlichkeit sowie die Wahrnehmung des Feedbacks in der KEA-Mod-Plattform anhand einer Fragebogenstudie zu ermitteln. Die Evaluationsergebnisse des Plattformeinsatzes im Wintersemester 2021/2022 am KIT wurden in [UI22c] veröffentlicht und in diesem Abschnitt zusammengefasst wiedergegeben.

Der Plattformeinsatz fand im Rahmen der Lehrveranstaltung »Programmierung kommerzieller Systeme - Einsatz betrieblicher Standardsoftware« (siehe auch Abschnitt 5.5.1) statt. Da die Studierenden die KEA-Mod-Plattform zuvor noch nicht genutzt hatten, erhielten sie ein Aufgabenblatt mit einer Beschreibung aller notwendigen Schritte. Beginnend mit der Registrierung bei der Plattform und der Eingabe des

Zugriffscodes für die zu bearbeitende Aufgabenreihe (siehe Abschnitt 5.4) mussten anschließend zwei Petri-Netz-Modellierungsaufgaben vom Typ »Modell erstellen« bearbeitet und die Lösungen gespeichert werden. Darauf folgte die finale Einreichung der Aufgabenreihe, so dass keine nachträglichen Änderungen der Lösungen möglich waren. Zugleich wurde die automatisierte Bewertung gestartet, woraufhin die Studierenden Feedback zu den von ihnen eingereichten Lösungen erhielten (»outcome«-Feedback, siehe Abschnitt 5.2.5). Dabei kamen der Syntax-Checker (siehe Abschnitt 5.3.2) sowie der Pragmatik-Checker (siehe Abschnitt 5.3.4) zum Einsatz⁶². Abschließend folgte das Ausfüllen eines Online-Fragebogens. Für die Bearbeitung der Modellierungsaufgaben und das Ausfüllen des Online-Fragebogens wurde ein Zeitaufwand von insgesamt ca. 60-90 Minuten geschätzt.

Der Online-Fragebogen bestand aus insgesamt 23 Items, die sich auf die Benutzungsfreundlichkeit (U1-10), die wahrgenommene Nützlichkeit der Plattform (N1-N3) sowie auf die Feedback-Wahrnehmung in der Plattform (F1-F10) beziehen. Die Items U1-U10 wurden auf Basis der Usability-Skala des »Web-based learning and usability questionnaire« [ML05] adaptiert. Das »Technology Acceptance Model 3« [VB08] wurde bei der Gestaltung der Items zur wahrgenommenen Nützlichkeit berücksichtigt, die Items zur Feedback-Wahrnehmung wurden auf der Grundlage verwandter Studien [MM21; SPN21] entwickelt. Alle Items sind in Tab. 5.15 aufgeführt und mussten auf einer siebenstufigen Likert-Skala (1=stimme überhaupt nicht zu; 7=stimme voll und ganz zu; zzgl. Antwortoption »keine Angabe«) bewertet werden. Zusätzlich wurden demografische Daten erhoben und sechs offene Fragen mit Freitexteingabe zu den Funktionen, dem Feedback und zur generellen Nützlichkeit der Plattform ergänzt.

Von ca. 250 Studierenden in der Lehrveranstaltung haben 235 Personen am Übungseinsatz und der Umfrage teilgenommen (71,1 % männlich, 28,5 % weiblich und 0,4 % divers). Der Altersdurchschnitt betrug 22 Jahre bei einer Altersspanne von 18 bis 33 Jahren. Die Hauptergebnisse zu den Items des Online-Fragebogens in Tab. 5.15 werden im folgenden kurz zusammengefasst und mit Auszügen aus den Freitextantworten zu den offenen Fragen ergänzt.

Insgesamt wurde die Benutzungsfreundlichkeit der Plattform (Items U1-U10) als durchschnittlich bewertet. Über alle 10 Items lag der Mittelwert bei 4,73 (SD=1,669). Dabei wurden die Einfachheit der Bedienung (Item U1; \bar{O} =5,38) und die Erlernbarkeit (U4; \bar{O} =5,94) der Plattform am besten beurteilt. Dies spiegelt sich auch in den offenen Angaben wieder (z.B. »Sehr überschaubar und wenig Erfahrung notwendig«, »Sehr

⁶² Die Entwicklung der weiteren Bewertungsdienste war zum Zeitpunkt des Übungseinsatzes noch nicht ausreichend vorangeschritten.

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.15.: Deskriptive Ergebnisse der Auswertung des Online-Fragebogens. med: Median, SD: Standardabweichung, N: Stichprobengröße nach Abzug der Items, die mit »Keine Angabe« bewertet wurden.

ID	Itemtext	N	med	Ø	SD
U1	Es war einfach, KEA-Mod zu nutzen.	235	6	5,38	1,579
U2	Ich kann meine Aufgaben mit KEA-Mod effizient erledigen (das heißt z.B. ohne unnötige Eingabeschritte).	234	5	4,40	1,647
U3	Ich fühle mich bei der Nutzung von KEA-Mod wohl.	235	5	4,46	1,698
U4	Es war einfach zu lernen, KEA-Mod zu benutzen.	231	6	5,94	1,418
U5	KEA-Mod gibt Systemmeldungen aus, die mir klar sagen, wie ich Probleme beheben kann.	168	4	4,07	1,872
U6	Wenn ich bei der Verwendung von KEA-Mod einen Fehler mache, kann ich diesen schnell und einfach beheben,	224	5	4,67	1,802
U7	Die in KEA-Mod bereitgestellten Support-Informationen (z.B. Bildschirm-Meldungen, Online-Hilfe und andere Dokumentationen) sind gut verständlich.	174	5	4,90	1,536
U8	Ich finde, dass die Benutzeroberfläche von KEA-Mod ansprechend gestaltet ist.	234	4	4,41	1,686
U9	KEA-Mod hat alle Funktionen und Fähigkeiten, die ich erwarte.	228	4,5	4,38	1,822
U10	Ich bin insgesamt mit KEA-Mod zufrieden.	235	5	4,69	1,630
N1	Ich empfinde KEA-Mod als nützlich für die von mir besuchte Lehrveranstaltung.	234	5	5,03	1,645
N2	Die Aufgabenreihe(n) in KEA-Mod zu bearbeiten, hat mir geholfen, meinen Lernfortschritt in der Lehrveranstaltung zu beurteilen.	233	5	5,01	1,645
N3	Ich bevorzuge Modellierungsaufgaben in KEA-Mod zu bearbeiten, anstatt mit Papier und Stift.	234	6	5,31	1,846
F1	Das Feedback in KEA-Mod hilft mir, die Lehrinhalte besser zu verstehen.	214	5	4,74	1,561
F2	Das Feedback in KEA-Mod hilft mir zu verstehen, wo ich persönlich Fehler gemacht habe.	213	5	4,94	1,653
F3	Das Feedback in KEA-Mod ist hilfreich für die Prüfungsvorbereitung.	215	5	4,93	1,634
F4	Ich empfinde das Feedback in KEA-Mod als nützlich.	219	5	5,15	1,591
F5	Ich bin mit dem Feedback in KEA-Mod zufrieden.	217	5	4,88	1,681
F6	Ich empfinde das Feedback in KEA-Mod als fair.	213	5	5,15	1,628
F7	Ich empfinde das Feedback in KEA-Mod als gerechtfertigt.	214	5	4,97	1,707
F8	Ich halte den Inhalt des Feedbacks für passend und zutreffend.	215	5	4,86	1,789
F9	Der Inhalt des Feedbacks in KEA-Mod ist gut verständlich.	217	6	5,47	1,627
F10	Die Darstellung des Feedbacks ist übersichtlich.	217	6	5,63	1,529

intuitiv«). Ebenso wurde das minimalistische und übersichtliche Design der Benutzungsoberfläche positiv hervorgehoben (»Simpler Aufbau, gute Übersichtlichkeit«, »Es ist minimal gestaltet«). Anhand der Freitextantworten zu den offenen Fragen wird deutlich, dass insbesondere bei der Bedienung des Modellierungseditors noch Verbesserungspotenzial besteht. Besonders häufig wurde das umständliche Einfügen

von Kanten zwischen Modellelementen, das Überlappen von Beschriftungen mit Modellbestandteilen sowie das Fehlen einer Copy&Paste-Funktion genannt⁶³.

In Bezug auf die wahrgenommene Nützlichkeit der Plattform (Items N1-N3) weisen die quantitativen Ergebnisse insbesondere darauf hin, dass die Mehrheit der Studierenden es bevorzugt, die Modellierungsaufgaben digital in der Plattform anstatt mit Papier und Stift zu bearbeiten (Item N3; $\bar{O}=5,31$). Die jeweiligen Vor- und Nachteile kommen in den offenen Angaben explizit zum Ausdruck: »Im Vergleich mit Stift und Papier hat KEA-Mod Vor- und Nachteile. Der Vorteil ist, dass sich Fehler leichter und besser verbessern lassen und Veränderungen durchgeführt werden können. Auch ist es deutlich übersichtlicher als wenn handgeschrieben/gezeichnet. Der Nachteil ist jedoch, dass die Verbindung durch die Pfeile vergleichsweise lang dauert.«, »Zu zeitaufwändig für Klausuren«.

Bezogen auf das Feedback (Items F1-F10) wurde insbesondere die Verständlichkeit des Feedback-Inhalts (Item F9) und die übersichtliche Darstellung (Item F10) positiv beurteilt. Den offenen Angaben nach erkannten die Studierenden durchaus den Mehrwert der Feedback-Funktion (»Feedback war gut und übersichtlich. Hat mich nochmal darin bestätigt wie ich das petri Netz aufzubauen habe und zu wissen dass ich keine (offensichtlichen) Fehler begangen habe.«⁶⁴). Es wurde aber auch darauf hingewiesen, dass einige Feedback-Nachrichten irritierend waren (z. B. »Fälschlicherweise markiert, dass es nicht nur einen Start & Endpunkt gibt«)⁶⁵. Weiterhin nannten einige Studierende die Ausbaufähigkeit des Feedbacks um weitere Bereiche, z.B. die inhaltliche Korrektheit der modellierten Lösung in Bezug auf das gegebene Szenario in der Aufgabenstellung (»Weitere Feedback-Bereiche wären interessant, wie z.B. logische Korrektheit.«, »Kontrolliert nur syntaktische Fehler und keine inhaltlichen.«). Dies bestätigt die Relevanz des Forschungsvorhabens und die Bestrebungen zur Weiterentwicklung der KEA-Mod-Plattform.

Insgesamt zeigen die Ergebnisse, dass der Einsatz der KEA-Mod-Plattform in einer großen Lehrveranstaltung von Studierenden positiv aufgenommen wurde. Darüber hinaus konnten Verbesserungsvorschläge gesammelt werden, die in die Entwicklungsarbeiten an der Plattform eingeflossen sind. Der entwickelte Online-Fragebogen kann auch künftig genutzt werden, um die Weiterentwicklung der KEA-Mod-Plattform

⁶³ Die Verbesserungsvorschläge wurden vom Entwicklungsteam gesammelt und sind zum aktuellen Zeitpunkt teilweise bereits berücksichtigt worden.

⁶⁴ Für alle Zitate aus den offenen Antworten gilt *sic scriptum errat* (lat.): »so wurde es geschrieben«.

⁶⁵ Dies ist auf die bereits genannte Schwierigkeit beim Einfügen von Kanten zurückzuführen, die mittlerweile behoben wurde. Dadurch konnte es passieren, dass eine Kante im Modellierungseditor optisch mit zwei Knoten verbunden zu sein schien obwohl dies technisch nicht der Fall war.

5. E-Assessment für das Gebiet der grafischen Modellierung

und den Einsatz in weiteren Szenarien zu evaluieren. Dabei könnte beispielsweise das »multiple try«-Feedback anstatt »outcome«-Feedback, zusätzliche Aufgabentypen (z. B. »Modell verstehen«) und weitere Modellierungssprachen bzw. Bewertungsdienste eingesetzt werden.

5.5.7. Plattform-Evaluation im Prüfungseinsatz

Im Rahmen des von der Stiftung Innovation in der Hochschullehre geförderten Projekts PePP (siehe Abschnitt 1.4) sollte die Weiterentwicklung und Erprobung der KEA-Mod-Plattform im Hinblick auf den summativen Einsatz in elektronischen Prüfungen erfolgen. Zu diesem Zweck wurde die KEA-Mod-Plattform an den Identitätsanbieter des KIT angebunden, um die an den Prüfungen teilnehmenden Studierenden darüber zu authentifizieren. Das vorrangige Ziel bei der Evaluation des Prüfungseinsatzes war es, die Akzeptanz von elektronischen Prüfungen anhand eines Fragebogens auszuloten und praktische Erfahrungen mit der Durchführung zu sammeln.

Der Prüfungseinsatz der KEA-Mod-Plattform fand im Sommersemester 2023 in den Nachklausuren zu den Veranstaltungen »Programmierung kommerzieller Systeme - Einsatz betrieblicher Standardsoftware« (14 Studierende) und »Angewandte Informatik - Modellierung« (31 Studierende) statt. Für die Vorbereitung auf die Prüfungen wurde den Studierenden nach der Klausuranmeldung ein Link zur KEA-Mod-Plattform mit einer entsprechenden Aufgabenreihe zur Übung mitgeteilt. Die beiden Prüfungen wurden im Hörsaal unter Aufsicht durchgeführt. Sie wurden in hybrider Form konzipiert, so dass ein Teil der Aufgaben wie üblich als Papierklausur gestellt wurde. Ein weiterer Teil der Aufgaben musste in der KEA-Mod-Plattform bearbeitet und eingereicht werden. Dafür stand für jede teilnehmende Person im Hörsaal ein Chromebook mit Maus am Sitzplatz bereit (siehe Abb. 5.25). Das Chromebook wurde zuvor so konfiguriert, dass beim Aufklappen direkt ein Browser mit zwei Tabs geöffnet wurde: ein Tab mit der Startseite der KEA-Mod-Plattform und ein weiterer Tab mit dem Online-Fragebogen. Der Netzzugriff war (neben den DNS-Servern) ausschließlich auf diese beiden Ressourcen begrenzt. In die Papierklausur wurde eine kurze Anleitung zur Verwendung der KEA-Mod-Plattform und der Zugangscode zur entsprechenden Aufgabenreihe eingebunden. Für den Fall technischer Schwierigkeiten wurden in der Papierklausur zusätzlich die Aufgaben aus der KEA-Mod-Plattform angegeben. Für die Anmeldung an der KEA-Mod-Plattform mussten die Studierenden ihr KIT-Konto verwenden. In der Prüfung zu



Abb. 5.25.: Prüfungseinsatz der KEA-Mod-Plattform mit Chromebooks im NTI-Hörsaal am KIT, Sommersemester 2023. Quelle: Eigene Aufnahme.

»Programmierung kommerzieller Systeme - Einsatz betrieblicher Standardsoftware« wurden eine Aufgabe vom Typ »Modell erstellen« (Petri-Netze) sowie zwei Aufgaben vom Typ »Modell verstehen« (EPK und ER-Diagramme) in der KEA-Mod-Plattform abgebildet. Bei der Prüfung »Angewandte Informatik - Modellierung« waren es drei Aufgaben vom Typ »Modell erstellen« (Petri-Netze, ER-Diagramme und UML-Klassendiagramme). Nach dem Ende der Prüfungen wurden die Teilnehmenden gebeten, den Online-Fragebogen direkt im Hörsaal auszufüllen.

Der für die Studierendenbefragung eingesetzte Online-Fragebogen wurde im PePP-Gesamtverbund unter Koordination der Universität Freiburg entwickelt und in der ILIAS-Plattform umgesetzt. Die übergeordneten Fragestellungen betreffen die Akzeptanz sowie Merkmale, die einen Einfluss auf die Akzeptanz haben können. Abgefragt wurden unter anderem Vorerfahrungen mit elektronischen Prüfungen, die Bedienung des Prüfungssystems (hier die KEA-Mod-Plattform), das Auftreten technischer Probleme, aber auch Faktoren wie Konzentration, Stress während der Prüfung und persönliche Präferenzen. Ein Großteil der Fragen war als Single-Choice-Frage mit Antwortoptionen wie z. B. »Ja« oder »Nein« gestellt. Oftmals gab es hierbei die Möglichkeit, einen Freitext zu ergänzen, beispielsweise um eine mit »Nein, weil:« beantwortete Frage mit zusätzlichen Angaben zu begründen. Die Bedienung des Prüfungssystems (B1-B7) sowie die persönlichen Präferenzen (P1 und P2) wurden

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.16.: Deskriptive Ergebnisse der Auswertung des Online-Fragebogens im Prüfungseinsatz. med: Median, SD: Standardabweichung, N: Anzahl der abgegebenen Antworten (beide Prüfungen).

ID	Itemtext	N	med	Ø	SD
B1	Ich bin mit dem Prüfungssystem (KEA-Mod) vertraut.	44	4	3,30	1,374
B2	Ich habe Aufgaben dieser Art schon vor der Prüfung (im Zuge der Prüfungsvorbereitung o.ä.) bearbeitet.	44	4	3,70	1,374
B3	Ich fand das Prüfungssystem (KEA-Mod) leicht zu bedienen.	44	4	3,43	1,087
B4	Es fiel mir leicht, fälschlich gegebene Antworten zu Prüfungsaufgaben zu verbessern.	44	3	3,36	1,080
B5	Ich habe mich gut auf dem Bildschirm zurecht gefunden.	44	4	3,95	1,012
B6	Ich hatte während der Prüfung Sorge, dass wegen Internet- oder Technikproblemen meine Antworten verloren gehen könnten.	44	3	3,11	1,333
B7	Die Bedienung des Computers und/oder des Prüfungssystems hat mich von den Prüfungsaufgaben abgelenkt.	44	2	2,55	1,247
P1	Ich schreibe meine Prüfungen lieber auf Papier.	44	4	3,52	1,000
P2	Ich wünsche mir mehr digitale Prüfungen in meinem Studium.	44	3	3,14	1,091

anhand von 9 Items erhoben, die auf einer fünfstufigen Likert-Skala (1=Trifft gar nicht zu; 5=Trifft voll zu) bewertet werden mussten. Sie sind in Tab. 5.16 aufgeführt. Zusätzlich wurden drei offene Fragen mit Freitextangaben zu positiven, negativen und sonstigen Anmerkungen ergänzt.

Aufgrund der geringen Anzahl an Teilnehmenden wurden die Fragebogenantworten aus den beiden Prüfungen kombiniert ausgewertet⁶⁶. 27 von 44 Studierende (60 %) gaben an, dass die Prüfung ihre erste digitale Prüfung gewesen sei⁶⁷. Die Frage »Konnten Sie sich gut auf die Aufgaben konzentrieren?« wurde von 31 Studierenden (ca. 70 %) mit »Ja« beantwortet. Bei denjenigen die diese Frage mit »Nein, weil:« beantwortet haben, wurden als Gründe überwiegend Klick- und Tastaturgeräusche angegeben, vereinzelt Aspekte wie beschränkter Platz auf dem Tisch durch das Chromebook oder Schwierigkeiten mit dem Umgang mit der Plattform. Internet- oder Technikprobleme wurden lediglich von 3 Studierenden (ca. 7 %) gemeldet, die laut einer anschließenden Frage auch alle zeitnah und ohne Folgen für den Prüfungsablauf gelöst werden konnten.

Die Haltung der Studierenden zu Papier- bzw. elektronischen Klausuren (»Wie bewerten Sie digitale Prüfungen allgemein?«) wurde mit den Items P1 und P2 abgefragt

⁶⁶ Von insgesamt 45 Teilnehmenden an den beiden Prüfungen haben 44 den Fragebogen ausgefüllt.

⁶⁷ In der Umfrage wurde zur Erläuterung folgende Definition gegeben: »Mit digitalen Prüfungen sind hier Prüfungen gemeint, die am Computer/Laptop/Tablet abgelegt wurden - remote oder auf dem Campus (ohne mündliche Prüfungen).«

(siehe auch Tab. 5.16). Hier zeigen die Ergebnisse kein eindeutiges Stimmungsbild. 23 Studierende (ca. 52 %) geben an, ihre Prüfungen lieber auf Papier schreiben zu wollen (»Trifft eher zu« und »Trifft voll zu«), 13 Studierende (ca. 30 %) sind unentschieden. Allerdings wünschen sich auch 16 Studierende (ca. 36 %) mehr digitale Prüfungen im Studium, ebensoviele Studierende sind diesbezüglich unentschieden. Es zeigt sich hier, dass die Formulierung der Items einen Einfluss auf die Bewertung besitzt. Die allgemein gestellte Frage nach dem Stressempfinden (»Fühlen Sie sich in einer digitalen Klausur mehr gestresst als in einer klassischen Papierklausur?«) wurde von 20 Studierenden (ca. 45 %) mit »Mehr«, 4 Studierenden (ca. 10 %) mit »Weniger« und 20 Studierenden mit »Gleich« (ca. 45 %) beantwortet. Die Begründungen bezüglich der Antwort »Mehr« in den Freitextantworten beziehen sich überwiegend auf die Möglichkeit technischer Probleme (z. B. »Pauschal nicht zu beantworten, aber es gibt einfach mehr Dinge die schief gehen können« oder »mehr Ungewissheiten / höheres Risikopotenzial (aufgrund potenzieller Probleme der Technik)«), zeitintensivere Bearbeitung am Computer, mangelnde Vertrautheit mit dem Prüfungssystem und dem unbekanntem Modus der Prüfung. Zur Antwort »Weniger« wird als Begründung angeführt, dass die elektronische Eingabe ein saubereres und schnelleres Arbeiten ermöglicht. Hier offenbarten sich auch diametrale Ansichten: »auf Papier zeichnen geht schneller« vs. »ich kann am PC schneller schreiben«. Bei der direkten Frage nach der persönlichen Präferenz (»Kommen Ihnen persönlich digitale Prüfungen eher entgegen als klassische Papierklausuren?«) fällt das Ergebnis noch weniger eindeutig aus als bei den Items P1 und P2. 10 Studierende (ca. 23 %) beantworten diese Frage mit »Ja«, 11 Studierende (ca. 25 %) mit »Nein« und die überwiegende Mehrheit von 23 Studierenden (ca. 52 %) mit »Es macht für mich keinen Unterschied«. Die Begründungen in den Freitextantworten für die Antworten »Ja« bzw. »Nein« decken sich mit den vorherigen Angaben (»Ja, weil:« Weniger Schreib/Zeichenarbeit als Papierklausuren bzw. »Nein, weil:« weniger Fehlerpotentiale in Papierklausur, mehr Unsicherheit, mangelnde Übung). Bei den Freitextantworten zur offenen Frage »Was hat Ihnen an der Prüfung gut gefallen« wurde der Nutzen einer elektronischen Prüfung insbesondere für Aufgaben vom Typ »Modell erstellen« hervorgehoben: »Für Modellierungsaufgaben hat sich das super geeignet, da man die ganzen Kästchen und Pfeile nicht selbst zeichnen musste«, »sauberes Design von Modellen und einfach zu korrigieren«, »Modellieren (und auch Programmieren) auf dem Papier ist nicht realitätsnah«. Weiterhin wurde eine Reihe von Verbesserungsvorschlägen aufgenommen wie z. B. eine frühere Einführung in das Prüfungssystem während Vorlesung und Übungen, eine Hochsetzung der Zeitvorgaben, größere Tische im Hörsaal, Minimierung von Störgeräuschen und vereinfachte Handhabung von Kanten

5. E-Assessment für das Gebiet der grafischen Modellierung

im Modellierungseditor der KEA-Mod-Plattform.

Zusammenfassend zeigen die Ergebnisse des Prüfungseinsatzes, dass die Studierenden mit der Bedienung der KEA-Mod-Plattform ausreichend gut zurechtgekommen sind (\bar{X} : 3,34; SD: 1,18 über alle 7 Items B1-B7, siehe Tab. 5.16) und auch den speziellen Nutzen digitaler Werkzeuge im Bereich der Modellierung erkennen (siehe Beispiele aus den Freitextantworten). Ebenso wird deutlich, dass sich die persönlichen Präferenzen bezüglich elektronischer oder Papierklausuren stark unterscheiden. Die genannten Begründungen für die Ablehnung von digitalen Prüfungen können jedoch durch geeignete Maßnahmen berücksichtigt werden. Insbesondere eine intensivere Vorbereitungsphase und stärkere Einbettung des Prüfungssystems in Vorlesungen und Übungen könnte nicht nur Unsicherheiten reduzieren, sondern letztendlich auch die Versiertheit und Geschwindigkeit bei der Eingabe am Computer erhöhen. Bezüglich des häufig befürchteten Falls von technischen Schwierigkeiten könnte das Angebot einer Ausweichlösung oder die Zusicherung eines eventuellen Nachteilsausgleichs eine weitere Sorge der Studierenden abmildern.

5.6. Diskussion

An dieser Stelle soll eine kritische Betrachtung der im vorangegangenen Abschnitt 5.5 beschriebenen Evaluationsergebnisse und der KEA-Mod-Plattform (siehe Abschnitte 5.1, 5.2 und 5.3) vorgenommen werden. Dabei werden in den folgenden Abschnitten verschiedene Aspekte herausgegriffen und vertieft diskutiert, um Grenzen sowie Erweiterungsmöglichkeiten aufzuzeigen.

5.6.1. Modellierungsaufgaben und Modelle der Evaluation

Die in Abschnitt 5.5.1 vorgestellte Datenbasis dient als Grundlage für die in den Abschnitten 5.5.2, 5.5.3 und 5.5.4 beschriebene Evaluation. Sie kann grundsätzlich als Instrument für die (Weiter-)Entwicklung und Evaluation von Bewertungsdiensten für Petri-Netze verwendet werden. Die in der Datenbasis enthaltenen Modelle wurden zu drei Modellierungsaufgaben A1 (siehe Abb. 5.16), A2 (siehe Abb. B.1) und A3 (siehe Abb. B.2) erstellt. Das zu modellierende Szenario wurde bei allen drei Modellierungsaufgaben in englischer Sprache verfasst, so dass die Modelle in der Datenbasis ausschließlich englischsprachige Beschriftungen enthalten. Während die Sprache der Beschriftungen für die meisten Bewertungsdienste der KEA-Mod-Plattform (siehe

Übersicht in Tab. 5.3) keine Rolle spielt, so ist sie bei der inhaltlichen Prüfung von Modellen von großer Bedeutung. Bei der Entwicklung des Semantik-Checkers wurde ebenso wie bei den anderen Bewertungsdiensten in der KEA-Mod-Plattform zunächst ein Fokus auf Modelle mit englischsprachigen Beschriftungen gelegt, um die Anwendung der Bewertungsdienste im internationalen Raum und den Einsatz bestehender NLP-Werkzeuge mit hohem Reifegrad zu ermöglichen. In einem nächsten Schritt könnte die Erweiterung dieser Bewertungsdienste auf zusätzliche Sprachen vorgenommen werden. Um beispielsweise den Semantik-Checker für die Anwendung auf Modellen mit deutschsprachiger Beschriftung vorzubereiten, müssten die Vorverarbeitungsschritte entsprechend angepasst werden (Stoppwort-Liste und Parametrisierung des Stanford-CoreNLP-Frameworks). Die für die Berechnung der semantischen Ähnlichkeit verwendete WordNet-Instanz könnte durch das deutschsprachige Pendant GermaNet [HF97] ersetzt werden. Um diese Erweiterung zu evaluieren, sollte eine Sammlung von Modellen zu Modellierungsaufgaben mit deutschsprachigen Texten erfolgen. Dies ist anhand der KEA-Mod-Plattform durchführbar.

Weiterhin ist eine Ausweitung der Datenbasis hinsichtlich einer Variation des Freiheitsgrades bei den Modellierungsaufgaben und Größe oder Komplexität der zu erstellenden Modelle denkbar. Zu erwarten ist, dass eine Einschränkung des Freiheitsgrades bei der Modellierung wie beispielsweise durch die Vorgabe bestimmter Beschriftungen innerhalb der Modellierungsaufgabe (siehe z. B. [So10a]) die Qualität des Matchings durch den Semantik-Checker bei der inhaltlichen Prüfung erhöht. Dies hat sich in der Evaluation im Vergleich zwischen den Aufgaben A1 (geringerer Freiheitsgrad durch präzise beschriebenes Szenario) und A2 bzw. A3 gezeigt (siehe Abschnitt 5.5.3). Gleichzeitig sinken dabei allerdings auch die Anforderungen an die Lernenden, da die Kompetenzfacette MB 2.05 »Die Lernenden können Bezeichner für Modellelemente a) aus einer Problemstellung übernehmen, b) bei Bedarf konventionkonform anpassen und c) einheitlich verwenden.« gegebenenfalls kaum oder gar nicht adressiert wird. Dies gilt es bei der Konzipierung von Modellierungsaufgaben in Lehrveranstaltungen entsprechend abzuwägen. Die gleiche Abwägung gilt es in Bezug auf die Wahl der Größe und Kontrollflusskomplexität der zu erstellenden Modelle zu treffen. Tab. 5.17 zeigt einen Überblick über die leicht variierende Anzahl an Stellen, Transitionen, Kanten, XOR- sowie AND-Splits und Joins der Musterlösungen zu den drei Aufgaben A1-A3. In diesem Vergleich ist die Musterlösung von Aufgabe A1 etwas weniger groß (Anzahl Stellen, Transitionen und Kanten), ebenso ist die Kontrollflusskomplexität (Anzahl XOR- bzw. AND- Split und Joins) niedriger. Auch hier wäre eine Ausweitung der Datenbasis auf Modellierungsaufgaben bzw.

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.17.: Größe und Kontrollflusskomplexität der Musterlösungen zu den Aufgaben A1-A3 der Datenbasis. SP: Splits, JOI: Joins.

Aufg.	# Stellen	# Trans.	# Kanten	# XOR-SP	# XOR-JOI	# AND-SP	# AND-JOI
A1	8	10	18	2	3	0	0
A2	11	11	24	2	2	1	1
A3	13	13	30	2	2	1	1

Modelle mit stärker variierenden Parametern interessant, um in einer anschließenden Evaluation untersuchen zu können, wie groß deren Einfluss auf die Ergebnisse der Bewertungsdienste ist.

5.6.2. Vergleich Bewertungsschema und Rubric

Dem in der Evaluation in Abschnitt 5.5.2 vorgestellten Bewertungsschema (siehe Tab. 5.8) liegt das Rubric für die Bewertung von Aufgaben des Typs »Modell erstellen« zugrunde (siehe Abschnitt 4.3, Tab. 4.4). Dabei wurden im Bewertungsschema jedoch nicht alle Eigenschaften des Rubrics übernommen. Während im Rubric die Bewertung der Kriterienerfüllung anhand dreier Stufen vorgeschlagen wird (gute bis sehr gute, befriedigende oder unzureichende Kriterienerfüllung), ermitteln die Bewertungsdienste zum aktuellen Zeitpunkt lediglich die erreichte Punktzahl für ein Kriterium. Beispielsweise wird die korrekte Verwendung der Modellierungssprache (Syntax) anhand des Bewertungsschemas über die Vergabe von mindestens 0 bis maximal 5 Punkten bewertet. Es gilt im Folgenden abzuwägen, ob eine weitere Einteilung der erreichten Punkte in drei verschiedene Stufen entsprechend des Rubrics einen Mehrwert für die benutzenden Personen bietet. Ebenso ist offen, wie das Ergebnis in der grafischen Benutzungsoberfläche dargestellt werden könnte, z. B. in tabellarischer Form wie das Rubric oder anhand eines Ampelsystems (≤ 1 Punkt - rot - unzureichende, > 1 und < 4 Punkte - gelb - befriedigende bzw. ≥ 4 - grün - gute bis sehr gute Kriterienerfüllung).

Weiterhin wurden die drei Kriterien Validität, Vollständigkeit und Prägnanz (Semantik) aus dem Rubric im vorliegenden Bewertungsschema zu einem gemeinsamen Kriterium zusammengesetzt, da der Semantik-Checker die festgestellten Abweichungen eines studentischen Modells von der Musterlösung nicht zu diesen drei Kriterien zuordnet (siehe Abschnitt 5.3.3). So wurde bereits darauf hingewiesen, dass beispielsweise eine fehlende Aktivität in einem Modell zunächst dafür sorgt,

dass das Modell nicht vollständig ist und unter Umständen gleichzeitig auch dafür, dass das Modell nicht korrekt ist. An dieser Stelle muss eine Entscheidung darüber gefällt werden, ob die Überschneidung der beiden Kriterien im Bewertungskontext wünschenswert ist. Einerseits ist es durchaus möglich, den Lernenden zurückzumelden, dass die fehlende Aktivität zur Verletzung beider Kriterien führt. Andererseits kommt es bei der Bewertung dadurch zum Punktabzug bei beiden Kriterien, was möglicherweise als unfair erachtet werden könnte und die Nachvollziehbarkeit der Bewertung erschwert. Bei der manuellen Bewertung, die in Abschnitt 5.5.4 beschrieben wurde, erfolgt der Punktabzug daher durch eine Erweiterung der Vorgaben stets nur zu einem der beiden Kriterien (siehe Tab. 5.13). In zukünftigen Arbeiten könnte z. B. zunächst anhand von Befragungen von Lehrpersonal und Lernenden untersucht werden, wie eine Überschneidung (bzw. Nicht-Überschneidung) dieser Kriterien bei der Bewertung aufgenommen wird, um bei der (Weiter-)Entwicklung von Bewertungsdiensten eine entsprechende Ausrichtung vornehmen zu können.

Nicht im Bewertungsschema enthalten ist das Kriterium »Einheitlicher Abstraktionsgrad« aus dem Rubric. Bei diesem geht es darum, bei der Modellierung ein angemessenes Abstraktionsniveau zu wählen, welches konsequent beibehalten wird. Eine Prüfung dieses Kriteriums ist anhand der Modellierungsaufgaben A1-A3 nicht möglich, da die Aufgabenstellung es erfordert, alle im textuellen Szenario beschriebenen Aussagen abzubilden. Dabei sind keine Abstrahierungsschritte vorgesehen. Um das Kriterium zu adressieren, ist eine Variante des Aufgabentyps »Modell erstellen« notwendig, die im Aufgabenkatalog [So22f, S. 10ff] beschrieben wird (siehe Tab. 4.3). Hierbei erfordert die Problemstellung explizit, dass ein angemessener Abstraktionsgrad ausgewählt und beibehalten wird. Das textuelle Szenario muss entsprechend detailliert gestaltet werden, so dass eine Zusammenfassung bzw. verkürzte Darstellung verschiedener Aspekte möglich ist. Die Entwicklung eines entsprechenden Bewertungsdienstes auf Basis des Abgleichs mit einer Musterlösung dürfte allerdings noch herausfordernder sein als in dieser Arbeit für den Semantik-Checker herausgestellt, da durch den Zusammenfassungsschritt bei der Wahl der Beschriftungen ein entsprechend hoher Freiheitsgrad entsteht.

Zum Aspekt Pragmatik enthält das Rubric zwei Kriterien: Eines mit Bezug auf den Stil der Beschriftungen und eines mit Bezug auf das Layout bzw. die Anordnung der Modellelemente auf der Arbeitsfläche. Für beide Kriterien enthält der Pragmatik-Checker entsprechende Methoden für die Überprüfung (siehe Abschnitt 5.3.4, Tab. 5.5). In das Bewertungsschema eingeflossen sind drei dieser Methoden. Zum Einsatz kommen (i) die Überprüfung des Vorhandenseins von Kantenüberschneidungen und

5. E-Assessment für das Gebiet der grafischen Modellierung

dem (ii) Verlauf der Kanten in Flussrichtung von links nach rechts sowie die Überprüfung der Knoten auf das Vorhandensein einer Beschriftung. Auf die Anwendung der vorhandenen Methode zur Überprüfung des Stils der Knotenbeschriftungen wurde verzichtet. Eine Betrachtung der Musterlösungen zu den drei verwendeten Modellierungsaufgaben macht deutlich, dass hier die Beschriftungen teilweise etwas komplexer sind (z. B. »enter customer data in system« für Aufgabe A2) als der Verb-Objekt-Stil der unter anderem in der 7PMG-Richtlinie [MRA10] vorgeschlagen wird. Ausblickend bieten sich einige der in Abschnitt 3.3.1 beschriebenen Ansätze zur Analyse von Modellelementbeschriftungen an, in einen Bewertungsdienst in der KEA-Mod-Plattform integriert zu werden. Erste Schritte dazu wurden bereits im Rahmen einer Abschlussarbeit⁶⁸ eingeleitet. In der Arbeit wurde ein Prototyp entwickelt, der das Verfahren von LEOPOLD et al. [Le19] zur Identifikation des Beschriftungsstils von Aktivitäten unter Verwendung eines HMM-Modells (siehe Abschnitt 3.3.1) umsetzt. Dieser könnte in weiteren Arbeiten auf der vorhandenen Datenbasis evaluiert und angepasst werden. Eine Erweiterung des Prototyps um die Bestimmung des Beschriftungsstils von Stellen (Zuständen) in Petri-Netzen und die Überprüfung auf die konsistente Verwendung des jeweils gewählten Stils könnte im Rahmen anknüpfender Arbeiten umgesetzt werden. Ebenso ausbaufähig ist die Betrachtung des Modell-Layouts unter pragmatischen Gesichtspunkten. Sofern entsprechende Vorgaben definiert wurden, lassen sich diese auf Basis des KEA-Formats (siehe Abschnitt 5.1.4) auf unkomplizierte Weise automatisiert überprüfen. Beispielsweise kann die Anordnung von Modellelementen auf der Arbeitsfläche auf eine präzise und systematische Ausrichtung untersucht werden (z. B. immer gleiche oder zumindest ähnliche Abstände zwischen Modellelementen). Auch könnten dabei weitere Merkmale der sogenannten sekundären Notation (siehe Abschnitt 2.2.3) wie z. B. die Größe der Modellelemente oder die Platzierung der Beschriftungen in der Überprüfung berücksichtigt werden. Allerdings müssten entsprechende Vorgaben für das Layout zunächst herausgearbeitet werden. Einen Ausgangspunkt dafür könnten Studien zur Layout-Ästhetik bilden. Beispielsweise befassen sich EFFINGER et al. [EJS10], LÜBKE et al. [LAS21] und SCHOLZ und LÜBKE [SL19] mit der Layout-Ästhetik für BPMN-Modelle. In ihren Arbeiten untersuchen sie unter anderem Nutzerpräferenzen für die Layout-Gestaltung und den Einfluss des Layouts auf die Verständlichkeit von BPMN-Modellen.

⁶⁸ David Brüning, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Analyse von graphbasierten Diagrammen mit Methoden des Natural Language Processing*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2022.

5.6.3. Ergebnisse der Anwendung des Bewertungsschemas

Im Rahmen der Evaluation wurden 2.194 studentische Modelle in Form von Petri-Netzen untersucht, die im Laufe von 7 Semestern zu drei verschiedenen Modellierungsaufgaben gesammelt wurden. Diese Untersuchung fand durch die Anwendung des Bewertungsschemas anhand der im Forschungsvorhaben entstandenen Bewertungsdienste statt (siehe Abschnitt 5.5.2). Aus der Literatur ist bislang keine Untersuchung einer vergleichbar großen Menge studentischer Modelle bekannt. Im Folgenden soll der Einfluss des gewählten Bewertungsschemas sowie der durch die Bewertungsdienste durchgeführten Aggregation von Teilergebnissen (5.3.1) auf die Ergebnisse der Untersuchung diskutiert werden.

Die durchschnittlich erreichte Gesamtpunktzahl der studentischen Modelle ist mit 11,53 P von 20 P (siehe Tab. 5.9) recht niedrig. Dies ist nicht nur auf die besonders niedrige Durchschnittspunktzahl bei der Kategorie Semantik (2,7 P von 10 P) zurückzuführen, sondern auch darauf, dass diese Kategorie mit 50% doppelt so stark gewichtet wurde wie die anderen beiden Kategorien Syntax und Pragmatik (je 25%). Zur Ermittlung einer möglichen Gewichtung der Kategorien Syntax, Semantik und Pragmatik wurden bereits empirische Studien durchgeführt. Dabei wurden von MOODY et al. [Mo02] 194 und von OVERHAGE et al. [OBS12] 11 Teilnehmende befragt. Während in [Mo02] anhand einer Regressionsanalyse ein Gesamtergebnis von $0,33 \cdot \textit{Syntax} + 0,31 \cdot \textit{Semantik} + 0,34 \cdot \textit{Pragmatik} + 0,07$ ermittelt wird, bestätigen [OBS12] eine Expertenzustimmung zu einer Gleichverteilung der drei Kategorien (je $\approx 0,33$ bzw. 33%). Das hier festgelegte Bewertungsschema mit einem Verhältnis von 25%, 50%, 25% weicht jedoch von diesen Vorschlägen ab, da der Schwerpunkt bei der Aufgabenstellung auf der inhaltlichen Umsetzung des zu modellierenden Szenarios liegt. Die gewählte Gewichtung kann im Bewertungsschema auf den individuellen Wunsch des Lehrpersonals angepasst werden um auf dieser Basis das Gesamtergebnis bzw. die Gesamtpunktzahl neu zu berechnen.

Für die Methoden der Bewertungsdienste Syntax-Checker und Pragmatik-Checker vom Typ Check, welche eine Fehlermenge ermitteln (siehe SYN01-SYN03 sowie PRA01 und PRA03 in Tab. 5.5.2) wurden recht niedrige obere Grenzen gewählt. Somit fällt ein einzelner Fehler zwar zunächst recht stark ins Gewicht, allerdings führen weitere Fehler über die obere Grenze hinaus nicht mehr zu weiterem Punktabzug. Beispielsweise wird durch die gewählte Gewichtung für das Kriterium SYN01 zur Kategorie Syntax insgesamt maximal 1 Punkt vergeben. Die obere Grenze bei der dazugehörigen Check-Methode CorrectNodeTypes wurde auf 2 gesetzt, so dass zwei

5. E-Assessment für das Gebiet der grafischen Modellierung

Tab. 5.18.: Bestimmung der Punktevergabe (hier 1 Punkt) durch die gewählte obere Grenze (OG) und die Fehlerzahl (3 F = 3 Fehler etc.).

OG	0 F	1 F	2 F	3 F	4 F	5 F	6 F	7 F	8 F	9 F	10 F
1	1 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P
2	1 P	0,5 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P
5	1 P	0,8 P	0,6 P	0,4 P	0,2 P	0 P	0 P	0 P	0 P	0 P	0 P
10	1 P	0,9 P	0,8 P	0,7 P	0,6 P	0,5 P	0,4 P	0,3 P	0,2 P	0,1 P	0 P

oder mehr Fehler zum vollständigen Punktabzug führen (entsprechend der Formel $R_e(g)$ in Abschnitt 5.3.1). Während es angemessen sein mag, weitere Fehler derselben Methode nicht zusätzlich zu bestrafen, führt dieses Vorgehen allerdings auch dazu, dass das Vorliegen von zwei oder mehr Fehlern durch die vergebene Punktzahl bei der Bewertung nicht differenziert reflektiert wird. Das heißt, ein Modell mit 2 Fehlern wird ebenso wie ein Modell mit 8 Fehlern mit 0 Punkten bewertet. Wird die obere Grenze auf einen höheren Wert gesetzt, ist es möglich anhand der vergebenen Punktzahl bis zum gewählten Wert eine Differenzierung vorzunehmen, falls dies gewünscht ist. In Tab. 5.18 wird dargestellt, wie sich die Änderung der oberen Grenze über eine Fehlerzahl von 0 bis 10 Fehlern hinweg bei der Punktevergabe von einem Punkt auswirkt. Allerdings muss für die erweiterte Differenzierung auch der Umstand in Kauf genommen werden, dass einzelne Fehler nunmehr weniger stark ins Gewicht fallen. Spielraum für weitere Anpassungen bietet der Punktabzug über die Fehlerzahl hinweg. Wie in Tab. 5.18 zu sehen, verläuft diese nach der aktuellen Formel linear. Bei Bedarf wäre die Einführung eines Reduktionsfaktors möglich, so dass ein Fehler mit zunehmender Häufigkeit des Auftretens mit einem schrittweise abnehmendem Punktabzug bestraft wird⁶⁹.

Weiterhin können ebenso die gewählten Gewichtungen für alle Teilkriterien des Bewertungsschemas auf die individuellen Anforderungen angepasst werden. Beispielsweise wurden zum Kriterium Syntax falsch eingesetzte Modellelemente (Knoten, Kanten) gleich (jeweils 1P), die Verletzung der Worfklow-Eigenschaft dahingegen stärker gewichtet (2P). Für das Kriterium Semantik wurde jeder fehlende Knoten mit einem Punktabzug von einem Punkt bestraft, das Auftreten zusätzlicher Knoten mit dem Abzug eines halben Punktes. Dadurch wird ein stärkerer Fokus auf die Vollständigkeit des Modells gelegt. Die Auswahl der Gewichtungen bezieht sich letztendlich auf die individuellen Anforderungen im Kontext der jeweiligen Aufgabenstellung.

⁶⁹ Selbstverständlich kann der Faktor auch so gewählt werden, dass ein Fehler mit zunehmender Häufigkeit mit zunehmendem Punktabzug bestraft wird, sofern dies als sinnvoll erachtet wird.

Die Festlegung der Relevanz der verschiedenen Kompetenzfacetten, die hinter den verschiedenen Bewertungskriterien stecken, liegt dabei im Ermessensspielraum der bewertenden Person. Richtlinien oder grobe Vorgaben als Anhaltspunkt für die Bewertung von Modellierungsaufgaben existieren bislang nicht. Sinnvoll ist daher eine Begutachtung des Bewertungsschemas und den gewählten Gewichtungen durch geeignete Personen. In Frage kommen z. B. Mitarbeitende der jeweiligen Lehrveranstaltung (Tutorinnen und Tutoren) und Personen mit vertieften Kenntnissen auf dem betreffenden Gebiet der Modellierung. Im Sinne des Schlagworts Partizipation und verbesserter Transparenz könnte vor der Anwendung des Bewertungsschemas auch eine Abstimmung mit den Studierenden der jeweiligen Lehrveranstaltung in Frage kommen.

Die Bewertung der Kriterien Syntax und Pragmatik aus dem Bewertungsschema durch den Syntax- bzw. Pragmatik-Checker erfolgt anhand der beschriebenen Checks bzw. Metriken. Das Bewertungsergebnis der automatisierten Bewertung ist auf dieser Grundlage als zuverlässig einzustufen. Eine ergänzende Untersuchung der Interrater-Reliabilität (siehe Abschnitt 2.1.4) könnte aufzeigen, wie häufig bei der manuellen Bewertung derselben Eigenschaften (Flüchtigkeits-)Fehler auftreten und somit möglicherweise ein weiteres Argument für die automatisierte Bewertung darstellen. Bei der Bewertung des Kriteriums Semantik wurde bewusst auf die Überprüfung der Intrarater-Reliabilität verzichtet. Diese stellt zwar eine geeignete Methode dar, um die Ergebnisse der automatisierten Bewertung mit den Ergebnissen einer manuellen Bewertung zu vergleichen [Ga19b], jedoch ist dies nur sinnvoll, wenn die automatisierte Bewertung so ausgerichtet wurde, dass sie sich einer manuellen Bewertung annähert. Dies ist beim Semantik-Checker in seiner aktuellen Form nicht der Fall (siehe Abschnitt 5.5.4).

5.6.4. Zulassen syntaktischer Mängel im Modellierungseditor

In Anbetracht dessen, dass Modelle mit syntaktischen Mängeln in Bezug auf die Prüfung von formalen Netzeigenschaften und auch im Hinblick auf semantische Aspekte in der automatisierten Bewertung nicht eindeutig interpretiert werden können (siehe z. B. Abschnitt 5.3.2), gilt es abzuwägen, ob sich das grundsätzliche Zulassen syntaktischer Mängel im Modellierungseditor der KEA-Mod-Plattform lohnt. Ein möglicher Ansatz könnte es sein, die Prüfung von Modellierungskompetenzen in Bezug auf die Syntax anhand einer eigenen Aufgabe vorzunehmen, bei der die Erstellung nicht korrekter Modelle im Modellierungseditor so wie derzeit

5. E-Assessment für das Gebiet der grafischen Modellierung

möglich ist. Die Prüfung weiterer Modellierungskompetenzen in Bezug auf formale Netzeigenschaften oder semantische Aspekte könnte in einer separaten Aufgabe erfolgen, bei welcher der Modellierungseditor so konfiguriert ist, dass syntaktische Mängel (wie z. B. unverbundene Kanten oder falsche Modellierungselemente) nicht entstehen können. Dies könnte durch eine Erweiterung des Modellierungseditors in der KEA-Mod-Plattform erreicht werden. Alternativ könnte ein Bewertungsdienst wie der Syntax-Checker (siehe Abschnitt 5.3.2) für die jeweilige Modellierungssprache während der Aufgabenbearbeitung aktiviert werden, damit Studierende vor der finalen Abgabe ihrer Einreichung die aufgezeigten Syntaxverletzungen selbst korrigieren können. Somit würde die Auseinandersetzung mit möglichen Uneindeutigkeiten (z. B. die Interpretation eines syntaktisch falschen Modellierungselements) im Kontext der automatisierten Bewertung entfallen. Eine mögliche Trennung bei der Prüfung der beiden Kompetenzfacetten Syntax und Semantik wird auch durch die Ergebnisse der Evaluation unterstützt. Schließlich weist nur ein geringer Anteil der untersuchten studentischen Modelle syntaktische Mängel auf (siehe Abschnitt 5.5.2). Allerdings gilt es zu berücksichtigen, dass die Integration einer sogenannten Modellierungsunterstützung in den Modellierungseditor (d.h. dass syntaktische Mängel nicht entstehen können) für komplexe und umfangreiche Modellierungssprachen wie z. B. BPMN einen hohen Entwicklungsaufwand bedeutet.

5.6.5. Erkenntnisse aus dem Einsatz des Semantik-Checkers

Mit Hilfe des Semantik-Checkers (siehe Abschnitt 5.3.3) können studentische Modelle in Form von Petri-Netzen mit Musterlösungen verglichen und dabei Abweichungen identifiziert werden. Obwohl das zugrundeliegende Prozessmodell-Matching recht gute Ergebnisse liefert (siehe Abschnitt 5.5.3), können die Abweichungen bislang nicht aggregiert und zu den einzelnen Fehlern bezüglich der Semantik-Kriterien des Rubrics (Vollständigkeit, Korrektheit, Prägnanz) zugeordnet werden, wie dies bei einer manuellen Korrektur der Fall wäre (siehe Abschnitt 5.5.4). Aufbauend auf den Matching-Ergebnissen des Semantik-Checkers könnten alternative Klassifikationsansätze entwickelt und auf der Datenbasis erprobt werden.

Weiterhin könnte der Semantik-Checker so angepasst werden, dass ein 1:1-Matching von Knoten des studentischen Modells mit der Musterlösung durchgeführt werden kann, sofern dies bei der jeweiligen Modellierungsaufgabe angemessen ist. Aktuell führt das verwendete n:1-Matching dazu, dass verschiedene Knoten mit gleicher Beschriftung im studentischen Modell zu einem einzelnen Knoten in der Musterlösung

zugeordnet werden können. Dadurch können Verletzungen des Kriteriums SEM04 (Abstraktion) (siehe Abschnitt 5.5.4) nicht hervorgehoben werden. Ebenso führt das n:1-Matching dazu, dass die Verwendung eines AND-Joins anstelle eines XOR-Joins nicht identifiziert werden kann (siehe Abb. 5.22).

Gibt es an mehreren Stellen in einer Musterlösung Möglichkeiten für strukturelle Variationen, so müssen für den Semantik-Checker recht schnell viele Musterlösungen erstellt werden, um alle möglichen Kombinationen dieser Variationen abzubilden. Dies bedeutet einen hohen Aufwand bei der Aufgabenerstellung. Hier wäre es denkbar, eine Möglichkeit zu schaffen, so dass eine einzelne Musterlösung mit Annotationen versehen werden kann, die diese Variationen beschreiben. Anschließend könnten die sich daraus ergebenden möglichen Musterlösungen automatisiert generiert und an den Semantik-Checker übergeben werden. Alternativ könnte das Matching des Semantik-Checkers durch eine größere Anpassung des Verfahrens auch direkt auf der annotierten Musterlösung durchgeführt werden.

Eine weitere Auseinandersetzung mit alternativen Möglichkeiten für die automatisierte Bewertung der Semantik-Kriterien ist wünschenswert. Beispielsweise könnte der Vergleich mit einer Musterlösung auch durch regelbasierte Verfahren ergänzt werden. Durch eine Identifikation von Abweichungen zwischen studentischem Modell und Musterlösung auf der Basis von Kontrollfluss-Mustern von Petri-Netzen (siehe Abschnitt 2.2.4) anstelle auf Grundlage von fehlenden bzw. zusätzlichen Knoten und Kanten könnte es gelingen, die automatisierte Bewertung bzw. Feedback-Generierung in die Nähe einer manuellen Bewertung zu rücken.

5.6.6. Einsatz von KI-Ansätzen im Bewertungskontext

In künftigen Bewertungsdiensten der KEA-Mod-Plattform könnten auch KI-Ansätze, wie z. B. maschinelle Lernverfahren zum Einsatz kommen. In Abschnitt 5.6.2 wurde bereits die Identifikation des Beschriftungsstils von Aktivitäten unter Verwendung eines HMM-Modells vorgestellt. In der durchgeführten Literaturstudie (Stand: Februar 2021) wurden ebenfalls einige Publikationen ermittelt, die Verfahren des maschinellen Lernens aufgreifen [BMM20; Li19; RM19; St19]. Es ist davon auszugehen, dass mit den jüngsten Fortschritten und der zunehmenden Verbreitung von z. B. Large Language Models (LLM)⁷⁰ weitere Möglichkeiten für den Einsatz von KI im

⁷⁰ LLM (deutsch: große Sprachmodelle) wie z.B. GPT-3 und GPT-4 von OpenAI (<https://openai.com>) oder LaMDA und PaLM von Google (<https://ai.google>) bilden die Basis für sogenannte Generative-AI-Anwendungen wie z. B. ChatGPT oder Bard.

5. E-Assessment für das Gebiet der grafischen Modellierung

Hochschulkontext eröffnet werden.

Im Rahmen einer Abschlussarbeit⁷¹ wurden bereits 2018 Experimente mit einer Support Vector Machine (SVM) aus dem Methodenportfolio des maschinellen Lernens zur Bewertung von Petri-Netzen durchgeführt. Die Ergebnisse wurden auf dem E-Prüfungs-Symposium 2019 vorgestellt [BOU19]. In den Experimenten wurde eine Teilmenge aus der Datenbasis (siehe Abschnitt 5.5.1) von 1014 studentischen Petri-Netzen zu den drei Aufgaben A1-A3 manuell mit Punktzahlen von 0 bis 12 Punkten bewertet. Für Aufgabe A1 konnten über 90% der Modelle eines Testdatensatzes korrekt den zwei Klassen »bestanden« (7-12 Punkte) und »nicht bestanden« (0-6) Punkte zugeordnet werden. Dabei zeigte sich wie erwartet, dass die Größe des Datensatzes mit Trainingsdaten sowie die Anzahl der Klassen bei der Klassifikation einen großen Einfluss auf die Korrektheit der Klassifikationsergebnisse besitzt. Umso größer der Datensatz mit Trainingsdaten und je geringer die Anzahl der Klassen bei der Klassifikation, umso besser waren die Klassifikationsergebnisse gemessen anhand des F-Score (siehe Abschnitt 5.5.3). Neben der manuellen Bewertung mit einer Gesamtpunktzahl wurden auf dem Datensatz auch Teilpunktzahlen von jeweils 0-4 für die Kriterien Syntax, Semantik und Pragmatik vergeben und als Trainings-Input verwendet. Die Vorhersage der exakten Teilpunktzahlen für diese Kriterien auf einem Testdatensatz erreichte dabei nur befriedigende Ergebnisse (jeweiliger F-Score: Syntax: 0,57; Semantik: 0,65; Pragmatik: 0,77).

Auch wenn KI-Ansätze zunächst vielversprechend erscheinen, so bergen sich hierbei dennoch einige spezielle Herausforderungen. Beispielsweise leistet der in [BOU19] entwickelte Ansatz lediglich die Vorhersage einer (Teil-)Punktzahl und nicht die Erstellung von individuellen Feedback-Nachrichten zum jeweiligen studentischen Modell. Weiterhin werden für das Training von aufgabenspezifischen KI-Modellen viele Trainingsdaten benötigt. Der Aufwand für die manuelle Klassifikation der Trainingsdaten kann dabei sehr hoch ausfallen, insbesondere wenn für jede einzelne Modellierungsaufgabe zur inhaltlichen Prüfung ein eigener Trainingsdatensatz benötigt wird. Bei der regelmäßigen Erstellung neuer Modellierungsaufgaben (jedes Semester, für jede Klausur) stehen zunächst keine entsprechenden Trainingsdaten zur Verfügung. Eine Übertragbarkeit von KI-Modellen über Modellierungsaufgaben hinweg ist allenfalls für die Kriterien Syntax und Pragmatik denkbar. Diese lassen sich allerdings auch zuverlässig regelbasiert prüfen (siehe Abschnitt 5.3.2 und 5.3.4). Regelbasierte Ansätze haben darüber hinaus den Vorteil, dass das Verhalten der

⁷¹ Matthias Binder, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Automatisierte Bewertung von Prozessmodellen mit Methoden des Maschinellen Lernens*, Institut AIFB, Karlsruher Institut für Technologie, Masterarbeit, 2018.

Bewertungsdienste durch Konfigurationsparameter gesteuert werden kann. Somit kann die gewünschte Funktionsweise garantiert werden und es liegt keine Ergebnisunsicherheit vor. Beispielsweise können Feedback-Nachrichten individuell nach den Wünschen der Lehrperson in Form von Textschablonen konfiguriert werden, so dass die Formulierung der Rückmeldungen an die Studierenden fest definiert werden kann. Dies ist im entworfenen Grundgerüst für die Bewertungsdienste im Rahmen dieser Dissertation möglich (siehe Abschnitt 5.3.1). Die automatisch generierte Antwort eines Chatbots auf Basis eines LLM bietet diese Option nicht, so dass in diesem Fall die Kontrolle über die Gestaltung der Feedback-Nachrichten verloren geht. Insbesondere bei Feedback vom Typ »Feed Forward« (siehe Abschnitte 2.1.4 und 4.3) sind spezifische Verweise auf das entsprechende Kursmaterial notwendig.

5.6.7. Technische Aspekte zur Plattform

In diesem Abschnitt werden technische Aspekte diskutiert, welche den aktuellen Stand der KEA-Mod-Plattform⁷² betreffen. Die derzeitige technische Umsetzung von Bewertungsschemata (siehe Abschnitt 5.2.1) erlaubt die Kombination der Teilergebnisse von verschiedenen Bewertungsdiensten über die gewichtete Summe. Dabei ist ein parametrisierter Aufruf der Bewertungsdienste möglich, um deren Verhalten auf individuelle Wünsche anzupassen. Sofern ein Bewertungsdienst selbst verschiedene Aspekte von studentischen Modellen prüft, müssen diese Teilergebnisse wiederum zu einem Gesamtergebnis aggregiert werden. Dies ist in Abb. 5.26 zu erkennen, welche das in Abschnitt 5.5.2 entwickelte Bewertungsschema aus der Evaluation in Baumstruktur darstellt. Für den Anwender bedeutet dies, dass Gewichtungen auf zwei Ebenen zu definieren sind: Erstens auf der Ebene des Bewertungsschemas der KEA-Mod-Plattform, um die Teilergebnisse der Bewertungsdienste zu einem Gesamtergebnis zu aggregieren (z. B. 25% Syntax-Checker, 50% Semantik-Checker und 25% Pragmatik-Checker). Zweitens auf der Ebene der eingesetzten Bewertungsdienste, um die Teilergebnisse verschiedener Checks oder Metriken zusammenzuführen (wie beim Syntax- und Pragmatik-Checker). Die Umsetzung auf diese Weise bietet eine große Flexibilität bei der Konfiguration individueller Wünsche. Allerdings ist es aus Nutzendensicht nicht unbedingt komfortabel, die Konfiguration auf diesen zwei unterschiedlichen Ebenen vornehmen zu müssen. Darüber hinaus gibt es für Bewertungsdienste keine Vorgaben zur Gestaltung der Konfigurationsparameter oder dazu, wie die Aggregation von Teilergebnissen umgesetzt werden

72 Stand: November 2023

5. E-Assessment für das Gebiet der grafischen Modellierung

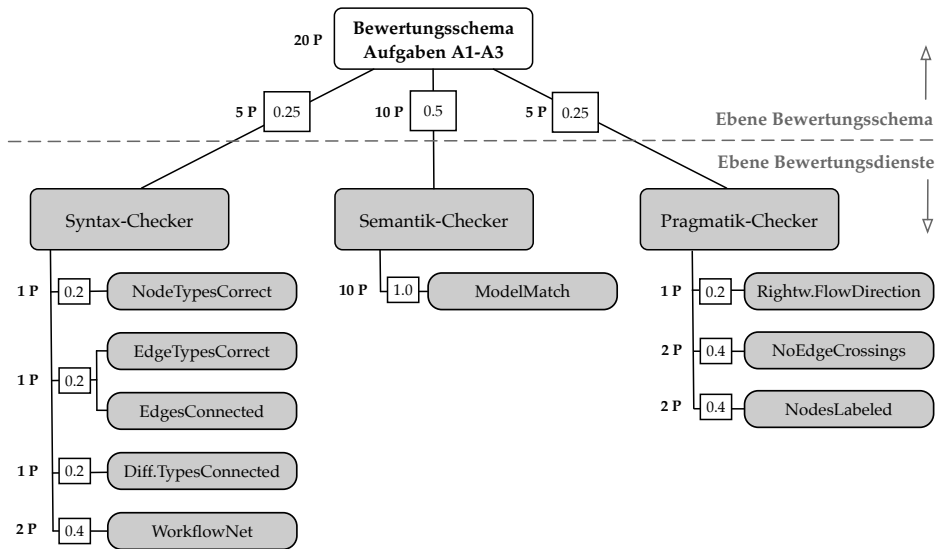


Abb. 5.26.: Bewertungsschema für Aufgaben A1-A3 (siehe auch Abb. 5.8), dargestellt in Baumstruktur. Erkennbar ist die Aufteilung der festzulegenden Gewichtungen auf den beiden Ebenen Bewertungsschema und Bewertungsdienste.

soll. Somit werden Nutzende damit konfrontiert, sich zunächst mit den jeweiligen Konfigurationsmöglichkeiten und Aggregierungsstrategien der verwendeten Bewertungsdienste vertraut zu machen. Auf Basis der aktuellen Architektur lässt sich die Gewichtung der beteiligten Bewertungsdienste und deren Checks auch auf die Ebene des Bewertungsschemas reduzieren, wenn Bewertungsdienste so aufgerufen werden, dass sie immer nur einen einzelnen Check durchführen. Beispielsweise würde der Pragmatik-Checker nach Abb. 5.26 dann dreimal aufgerufen und jeweils ein einzelnes Ergebnis zurückliefern (RightwardFlowDirection, NoEdgeCrossings und NodesLabeled). Allerdings bedeutet dies erstens, dass als Aggregierungsstrategie ausschließlich die gewichtete Summe über das Bewertungsschema zur Verfügung steht und zweitens, dass zusätzliche Aufrufe der verwendeten Bewertungsdienste vorgenommen werden müssen, was sich negativ auf den Ressourcenverbrauch auswirkt. Vor diesem Hintergrund erscheinen vor allem zwei Möglichkeiten für die zukünftige Anpassung sinnvoll: a) Erweiterung des Bewertungsschemas um zusätzliche Aggregierungsstrategien und b) Übertragung weiterer Informationen von der Ebene der Bewertungsdienste auf die Ebene des Bewertungsschemas. Bezüglich Punkt a) kommen zunächst die Aggregierungsstrategien in Frage, die in den

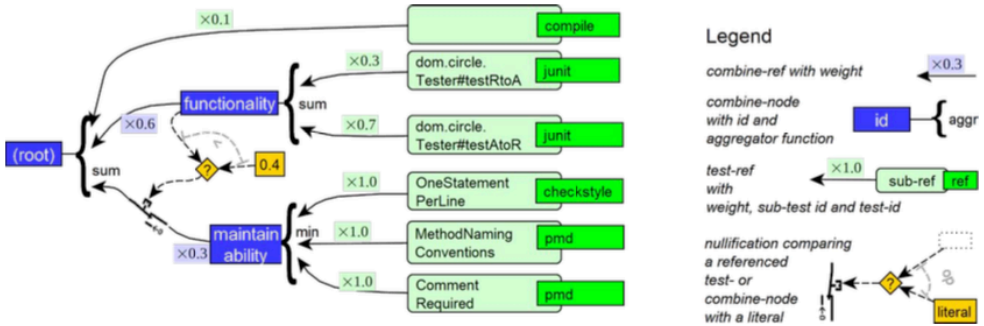


Abb. 5.27.: Illustration eines Bewertungsschemas im ProFormA-Format. Grafik entnommen aus [Ga19b].

im Kontext dieses Forschungsvorhabens entstandenen Bewertungsdiensten entwickelt wurden: die gruppenweise Betrachtung von Checks und die Festlegung von Schwellenwerten für Ergebnisse aus der Berechnung von Metriken (siehe Abschnitt 5.3.1). Zusätzlich können weitere Aggregierungsstrategien berücksichtigt werden. Das ProFormA-Format zur Spezifikation von Bewertungsschemata für automatisiert bewertbare Programmieraufgaben [Ga19a; Ga19b] verwendet neben der gewichteten Summe auch noch das gewichtete Minimum und gewichtete Maximum. Darüber hinaus können komplexe Abhängigkeiten über Bedingungen formuliert werden, was z. B. die Umsetzung der Nullifikation (siehe Abschnitt 5.3.2) auf der Ebene des Bewertungsschemas ermöglichen würde. Ein Beispiel für den Aufbau eines Bewertungsschemas im ProFormA-Format ist in Abb. 5.27 dargestellt. Das Beispiel zeigt die Anwendung des gewichteten Minimums über drei einzelne Testergebnisse für den Aspekt »maintainability« (Wartbarkeit). Für diesen Aspekt werden keine Punkte vergeben, wenn das Ergebnis des Aspekts »functionality« den Schwellenwert von 0,4 unterschreitet (Nullifikation über die Verwendung einer Bedingung).

Bezüglich des eingangs genannten Punktes b) können in der KEA-Mod-Plattform leicht weitere Informationen über die Teilergebnisse von Checks innerhalb der Bewertungsdienste auf die Ebene des Bewertungsschemas übertragen werden, indem entsprechende Felder in den Datentransferobjekten (siehe Abschnitt 5.2.2) und in der Datenbank ergänzt werden. Zusammen mit einer Erweiterung des Bewertungsschemas in der KEA-Mod-Plattform um die vorgestellten Funktionalitäten des ProFormA-Formats würde den Nutzenden eine umfassende individuelle Konfiguration der Bewertung ermöglicht, ohne dass sie sich vertieft mit den jeweiligen Konfigurationsparametern der verwendeten Bewertungsdienste auseinandersetzen müssten.

5. E-Assessment für das Gebiet der grafischen Modellierung

Zusätzlich würde dadurch der Aufwand bei der Entwicklung von Bewertungsdiensten reduziert, indem die Programmlogik für die Aggregation der Ergebnisse mehrerer Checks auf die Ebene des Bewertungsschemas ausgelagert wird.

Bei der Entwicklung von Bewertungsdiensten für die KEA-Mod-Plattform entsteht derzeit nicht nur Aufwand durch die Umsetzung einer entsprechenden Aggregierungsstrategie, sondern auch durch die Implementierung aller erforderlichen Basisfunktionen und der Kommunikationsschnittstelle. In dieser Dissertation ist ein Grundgerüst entstanden (siehe Abschnitt 5.3.1), welches diese Aufgaben bereits umsetzt. Allerdings erfordern Anpassungen am Grundgerüst derzeit, dass diese Anpassungen in allen Bewertungsdiensten, welche auf dem Grundgerüst aufsetzen, vorgenommen werden. Aus diesem Grund wurde im Rahmen einer Abschlussarbeit⁷³ ein generisches Framework für die Entwicklung von Bewertungsdiensten in der KEA-Mod-Plattform realisiert. Weiterentwicklungen am Framework können in den verwendenden Bewertungsdiensten nun bequem durch eine simple Versionsaktualisierung des Frameworks übernommen werden. Durch die Verwendung des Frameworks können sich Entwickelnde primär auf die Umsetzung der Programmlogik für die jeweiligen zu untersuchenden Modellaspekte konzentrieren, ohne sich z. B. mit der Kommunikation mit der Bewertungsqueue (Kafka), der Ein- bzw. Ausgabe im protbuf-Format der DTOs oder dem Parsing der Modelle im KEA-Format auseinandersetzen zu müssen. Zudem werden unterstützende Funktionen für die Arbeit mit Graphen durch die Anbindung der bewährten JGraphT-Bibliothek⁷⁴ zur Verfügung gestellt, beispielsweise für die Graphtraversierung anhand Tiefen- oder Breitensuche sowie Algorithmen für die Pfadsuche oder zur Identifikation von Zyklen.

5.6.8. Vergleich der Plattform mit verwandten Werkzeugen

Im Rahmen der Literaturstudie (siehe Abschnitt 3.3.2) wurden überwiegend Publikationen identifiziert, die Ansätze beschreiben, welche nur eine einzelne Modellierungssprache im Fokus haben (siehe Tab. A.1 im Anhang A). In diesem Abschnitt soll ein Vergleich mit denjenigen Werkzeugen vorgenommen werden, die – wie die

⁷³ Fabian Aisenbrey, Referent: Andreas Oberweis, Betreuerin: Meike Ullrich: *Generic framework for assessment services in e-assessment of graphical modeling*, Institut AIFB, Karlsruher Institut für Technologie, Bachelorarbeit, 2023.

⁷⁴ <https://jgrapht.org/>

KEA-Mod-Plattform – explizit auf eine allgemeine Unterstützung graphbasierter Diagramme und den Kontext Hochschullehre ausgerichtet wurden⁷⁵.

Das E-Assessment-Werkzeug **CourseMaster** der Universität von Nottingham, Großbritannien ist auf die automatisierte Bewertung verschiedener Eingaben, z. B. Quelltexte zu Programmieraufgaben ausgerichtet. Dieses Werkzeug wurde um die von HIGGINS und BLIGH [HB06] und HIGGINS et al. [HST02] beschriebenen Ansätze für die automatisierte Bewertung von graphbasierten Diagrammen ergänzt. Während die in den Publikationen angegebenen Verweise auf die Homepage von CourseMaster nicht mehr funktionieren, enthalten zumindest die Publikationen selbst eine Beschreibung der Architektur. Ähnlich wie bei der KEA-Mod-Plattform, wird hier eine Trennung verschiedener Komponenten vorgenommen. Eine Komponente dient der Erstellung von Modellierungsaufgaben, daneben existieren ein individuell konfigurierbarer Modellierungseditor für Studierende, eine Komponente für die automatisierte Bewertung sowie eine Komponente, um weitere Modellierungssprachen zu ergänzen. Für letzteres bietet die KEA-Mod-Plattform derzeit über die grafische Benutzeroberfläche für Endanwendende keine Möglichkeit. Die Palette der verfügbaren Symbole im Modellierungseditor muss durch Anpassung des Quelltextes erweitert werden. Bei Bedarf könnte eine entsprechende Funktionalität allerdings in die KEA-Mod-Plattform integriert werden. Für die in den Publikationen [HB06; HST02] erwähnte automatisierte Bewertung von Flussdiagrammen, Schaltplänen und ER-Diagrammen existiert keine weiterführende Beschreibung, so dass eine Übernahme der Ansätze in die KEA-Mod-Plattform nicht möglich ist.

Die in den zahlreichen⁷⁶ Publikationen von SMITH, THOMAS und WAUGH (siehe z. B. [STW04; TWS12]) beschriebenen Ansätze für die automatisierte Bewertung graphbasierter Diagramme wurden prototypisch in das E-Assessment-Werkzeug **OpenMark**⁷⁷ der Open University in Milton Keynes, Großbritannien eingebracht. Dieses bietet eine Reihe verschiedener Aufgabentypen (z. B. Multiple-Choice, Drag and Drop, Freitextaufgaben), die nicht spezifisch für einen bestimmten Anwendungsbereich sind. Die Weiterentwicklungsarbeiten an OpenMark im öffentlichen GitHub-Repository⁷⁸ wurden im Jahr 2016 eingestellt, die letzten Publikationen der Autorengruppe stammen aus dem Jahr 2013 [STW13; Th13]. Wie die KEA-Mod-

⁷⁵ Dabei ist zu berücksichtigen, dass der Vergleich auf den teilweise spärlichen Informationen basiert, die über die entsprechenden Werkzeuge anhand der Publikationen und ggf. (noch) vorhandenen Webseiten ermittelt werden konnten.

⁷⁶ Der Autorengruppe können 16 der 110 betrachteten Publikationen aus der Literaturstudie zugeschrieben werden.

⁷⁷ <https://www5.open.ac.uk/openmark-examples/>

⁷⁸ <https://github.com/OpenMark/openmark>

5. E-Assessment für das Gebiet der grafischen Modellierung

Plattform erlaubt auch OpenMark einen Einsatz wahlweise im formativen oder summativen Kontext. Die Entwicklung konkreter Ansätze für die automatisierte Bewertung graphbasierter Diagramme ist hauptsächlich auf den Anwendungsbe- reich der Softwareentwicklung und Datenmodellierung ausgerichtet (ER-Diagramme [TWS09], UML-Sequenz- und Klassendiagramme [STW10]), wohingegen bei der KEA-Mod-Plattform zusätzlich Ansätze für die Geschäftsprozessmodellierung be- rücksichtigt wurden (BPMN, EPK und Petri-Netze). Eine Integration der Ansätze für die automatisierte Bewertung von ER-, UML- und Sequenzdiagrammen aus OpenMark in Bewertungsdienste der KEA-Mod-Plattform würde es nicht nur erlau- ben, diese weiterhin praktisch einzusetzen, sondern auch, sie mit den bestehenden Bewertungsdiensten für ER- und UML-Klassendiagramme zu vergleichen.

Die jüngsten Ansätze für die automatisierte Bewertung graphbasierter Diagramme im Allgemeinen stammen von CORREIA et al. [CLP17; CLP18] und SOUSA und LEAL [SL15] an der Universität von Porto, Portugal. Sie wurden in **Mooshak**⁷⁹ integriert, einem Werkzeug, das hauptsächlich für die Durchführung von Programmierwett- bewerben entwickelt wurde. Wie auch bei CourseMaster können weitere Modellie- rungssprachen durch Endanwendende ergänzt werden. Dies setzt allerdings die Verwendung einer dafür speziell entwickelten Sprache (Diagrammatic Language Definition Language - DL^2 [CLP18]) voraus. Anhand dieser Sprache werden Syntax und Notation (siehe Abschnitt 2.2.3) der zu integrierenden Modellierungssprache beschrieben. Gleichzeitig dient die resultierende Beschreibung als Grundlage für die Überprüfung der Syntax-Regeln bei der automatisierten Bewertung. Die semantische Überprüfung erfolgt nach erfolgreicher Syntax-Prüfung anhand des Vergleichs mit ei- ner einzelnen Musterlösung [SL15]. Dieser zweistufige Ansatz für die automatisierte Bewertung ließe sich auch über einen Bewertungsdienst in die KEA-Mod-Plattform integrieren, um ihn weiter zu evaluieren. Lediglich die automatische Erweiterung der Symbolpalette des Modellierungseditors anhand der Beschreibung in DL^2 ist auf direktem Weg derzeit nicht möglich. Wie bereits zuvor angegeben, könnte eine derartige Möglichkeit bei Bedarf aber ergänzt werden.

Deutliche Alleinstellungsmerkmale der KEA-Mod-Plattform sind die starke Ausrich- tung auf die Kompetenzorientierung sowie die flexible Anbindung von Bewertungs- diensten über eine definierte Schnittstelle. In OpenMark und Mooshak kann das Feedback ebenfalls sowohl in textueller Form als auch im Modell integriert angezeigt werden. Allerdings bietet nur die KEA-Mod-Plattform die Möglichkeit, das Feedback der Bewertungsdienste in verschiedene Klassen einzuteilen (positiv, negativ, neutral)

79 <https://mooshak2.dcc.fc.up.pt/>

(siehe Abschnitt 5.2.2) und folgt den Feedback-Empfehlungen durch die Berücksichtigung der Konzepte des Feed Up und Feed Forward (siehe Abschnitte 4.3 und 5.4). Insbesondere die Rückmeldung von positivem Feedback ist aus didaktischen Gesichtspunkten ein wichtiger Bestandteil der kompetenzorientierten Lehre. Soweit aus den verfügbaren Publikationen und Dokumentationen ersichtlich, bieten alle hier betrachteten Werkzeuge den Aufgabentyp »Modell erstellen« an, jedoch nur in OpenMark ist zusätzlich auch der Aufgabentyp »Modell verstehen« aus der KEA-Mod-Plattform abbildbar. Um eine breitere Abdeckung der Kompetenzfacetten aus dem Kompetenzmodell zu erzielen, ist die KEA-Mod-Plattform zudem darauf ausgerichtet, künftig weitere Aufgabentypen für das Gebiet der grafischen Modellierung zu berücksichtigen.

5.6.9. Plattform-Evaluation im Übungs- und Prüfungseinsatz

Die Ergebnisse der Plattform-Evaluation im Übungs- und Prüfungseinsatz (siehe Abschnitte 5.5.6 und 5.5.7) liefern eine Bandbreite an Erkenntnissen. Während es beim Fragebogen des Übungseinsatzes um die Ermittlung der Benutzungsfreundlichkeit, der wahrgenommenen Nützlichkeit und die Wahrnehmung des Feedbacks geht, steht beim Fragebogen des Prüfungseinsatzes die Akzeptanz von elektronischen Prüfungen im Vordergrund. In diesem Abschnitt sollen einige Ergebnisse sowie Limitationen der beiden Evaluationen gemeinsam betrachtet und diskutiert werden.

Die quantitativen Ergebnisse der Plattform-Evaluation im Übungseinsatz weisen darauf hin, dass die Mehrheit von Studierenden es bevorzugt, Modellierungsaufgaben in der KEA-Mod-Plattform zu bearbeiten, anstatt mit Papier und Stift ($\bar{X}=5,31$ auf einer Likert-Skala von 1-7, siehe Tab. 5.15, Item N3). Dahingegen sprechen sich laut der Evaluationsergebnisse im Prüfungseinsatz deutlich weniger Studierende für den Einsatz digitaler Prüfungen im Studium aus ($\bar{X}=3,14$ auf einer Likert-Skala von 1-5, siehe Tab. 5.16, Item P2). Die höhere Akzeptanz der digitalen Bearbeitung im Kontext von Modellierungsaufgaben wurde auch in der Evaluation im Prüfungseinsatz in Freitextantworten erfasst (z. B. »Für Modellierungsaufgaben [...] super geeignet«, siehe Abschnitt 5.5.7). Zudem bestand bei der Bearbeitung der Modellierungsaufgaben im Übungseinsatz im Vergleich zum Prüfungseinsatz kein Zeitdruck, da die Aufgaben von zu Hause mit großzügigen zeitlichen Vorgaben (innerhalb von 5 Tagen) erledigt werden konnten. Dies könnte die größere Bereitschaft der Studierenden zur Verwendung eines digitalen Werkzeugs in diesem Kontext erklären. Eine

5. E-Assessment für das Gebiet der grafischen Modellierung

andere Studie zur Analyse der studentischen Wahl von Modellierungswerkzeugen in Prüfungen [SS21] kommt zu dem Schluss, dass die strikte Festlegung auf eine einzelne Variante (digitales Werkzeug vs. Papier und Stift) nicht den Präferenzen aller Studierenden gerecht wird. Vor diesem Hintergrund besteht eine besondere Herausforderung darin, Prüfungen so zu gestalten, dass den Studierenden kein Nachteil durch die Verwendung einer nicht präferierten Variante entsteht (Stichwort Chancengleichheit).

Obwohl Studierende im Übungseinsatz insbesondere die Einfachheit der Bedienung und die Erlernbarkeit der KEA-Mod-Plattform positiv bewertet haben ($\bar{X}=5,38$ und $\bar{X}=5,94$ auf einer Likert-Skala von 1-7, siehe Tab. 5.15, Items U1 und U4), bleibt die Gesamtbewertung der Benutzungsfreundlichkeit in beiden Evaluationen durchschnittlich (Übungseinsatz: $\bar{X}=4,37$ auf einer Likert-Skala von 1-7, Prüfungseinsatz: $\bar{X}=3,34$ auf einer Likert-Skala von 1-5). Auch die Freitextfelder liefern etliche Rückmeldungen zu technischen Schwierigkeiten, insbesondere mit der Bedienung des Modellierungseeditors. Zwischen dem Übungseinsatz (Dezember 2021) und dem Prüfungseinsatz (Juli/August 2023) wurde der Modellierungseeditor kontinuierlich weiterentwickelt. Dies hat sich auch über den Rückgang der entsprechenden Rückmeldungen in den beiden Evaluationen ausgedrückt. Insgesamt ist jedoch eine weitere Ausrichtung der KEA-Mod-Plattform im Sinne der Benutzungsfreundlichkeit erstrebenswert, um die Bereitschaft der Studierenden für deren Nutzung und den Einsatz in Prüfungen weiter zu erhöhen. Oftmals stehen derartige Implementierungsaspekte jedoch nicht im Vordergrund der Ausschreibungen für die Projektförderung. Dies hat sich insbesondere im Projekt KEA-Mod gezeigt. Hier standen primär wissenschaftliche Erkenntnisse im Fokus, so dass die Hinführung der Plattform zur Produktreife für die Endanwendung mangels entsprechender Ressourcen in den Hintergrund treten musste.

Aufgrund zeitlicher Verzögerungen bei den Entwicklungsarbeiten an der KEA-Mod-Plattform ist die Evaluation des Übungseinsatzes durch den Verbundpartner UPB zu einem Zeitpunkt erfolgt, zu dem noch nicht alle Bewertungsdienste vollständig fertiggestellt waren. Dies sollte bei der Interpretation der Ergebnisse zur Wahrnehmung des Feedbacks ($\bar{X}=5,07$ auf einer Likert-Skala von 1-7 über alle Items F1-F10, siehe Tab. 5.15) in der KEA-Mod-Plattform berücksichtigt werden. Hierzu wäre es wünschenswert, die Evaluation mit dem nun aktuellen Entwicklungsstand, insbesondere unter Einsatz des Semantik-Checkers, in einem der kommenden Semester zu wiederholen und die Ergebnisse zu vergleichen.

Mit beiden durchgeführten Evaluationen wurde die Zielgruppe der Studierenden be-

rücksichtigt. Eine breit angelegte Befragung von Lehrenden wurde aufgrund geringer Probandenzahlen bislang nicht vorgenommen. Jedoch wurden vom Verbundpartner UPB zwei Lehrpersonen am DFKI interviewt und im Dezember 2022 zwei sogenannte »Cognitive Walkthroughs« (CWT) durchgeführt, um die Handhabbarkeit und Benutzungsfreundlichkeit der KEA-Mod-Plattform aus Lehrendensicht zumindest qualitativ zu evaluieren [SS23]. Dabei wurde eine Lehrperson am KIT gebeten, während der Verwendung der Plattform in einer Videokonferenz »laut« zu denken (wie auch in der Laut-Denk-Studie in Abschnitt 4.1). Die Lehrperson hatte die KEA-Mod-Plattform zuvor nicht genutzt und war auch nicht in das Projekt KEA-Mod involviert. In der ersten CWT-Sitzung sollte die Lehrperson zunächst in der Lehrenden UI der KEA-Mod-Plattform eine Aufgabenreihe mit je einer vorgegebenen Modellierungsaufgabe vom Typ »Modell erstellen« und »Modell verstehen« einrichten. Zu diesen Aufgaben wurden anschließend von vier Tutorinnen und Tutoren individuelle Lösungen eingereicht. In der zweiten CWT-Sitzung mussten die Einreichungen durch die Lehrperson gesichtet und bewertet werden, inklusive der Erstellung manueller Feedbacks. Anhand anschließender Interviews und der Auswertung der Transkripte zu den CWT wurde deutlich, dass die Lehrperson die ihr gestellten Aufgaben ohne größere Schwierigkeiten lösen konnte. Hilfreich waren die zahlreichen Tooltips in der Lehrenden UI, welche Informationen zu den Konfigurationsparametern und Formularfeldern lieferten. Darüber hinaus könnte die Darstellung der Bedienelemente künftig besser auf die tatsächliche Handlungsabfolge bei der Korrektur angepasst werden und eine Erstellung von integrierten Feedback-Nachrichten während der gleichzeitigen Betrachtung des betreffenden Modells ermöglicht werden. Bezüglich des wahrgenommenen Nutzens gab die Lehrperson an, dass sie einen großen Mehrwert der KEA-Mod-Plattform für Lehrende sieht. Dieser Mehrwert wird vor allen Dingen durch die automatisierte Bewertung von Modellierungsaufgaben generiert, wodurch mögliche Flüchtigkeitsfehler und der Zeitaufwand bei der Korrektur reduziert werden können.

Eine weitere Ausrichtung der Benutzungsfreundlichkeit der KEA-Mod-Plattform für die Zielgruppe der Lehrenden wird als sinnvoll erachtet. Darüber hinaus sollte eine Evaluation mit höheren Teilnahmezahlen innerhalb dieser Zielgruppe durchgeführt werden, um ein differenziertes und umfassendes Meinungsbild zur KEA-Mod-Plattform und ihren Funktionen zu erhalten.

Teil III

Schluss

6. Zusammenfassung und Ausblick

Im Rahmen des in dieser Arbeit beschriebenen Forschungsvorhabens sind zwei zentrale Forschungsfragen FF1 und FF2 betrachtet worden: »Wie kann das Prüfen auf dem Gebiet der grafischen Modellierung kompetenzorientiert gestaltet (FF1) und durch eine E-Assessment-Plattform unterstützt werden (FF2)?« (siehe Abschnitt 1.2). Im Kontext von FF1 wurden zunächst hochschuldidaktische Grundlagen gelegt (siehe Kapitel 4). Diese Grundlagen wurden im Hinblick auf FF2 bei der Konzeption und Entwicklung einer entsprechenden E-Assessment-Plattform berücksichtigt (siehe Kapitel 5). In diesem Kapitel sollen die primären Ergebnisse bzw. Artefakte aus den beiden Forschungsfragen kurz zusammengefasst werden (siehe Abschnitte 6.1 und 6.2). Anschließend soll ein Ausblick auf weiterführende Forschungsfragen und Weiterentwicklungsmöglichkeiten gegeben werden (siehe Abschnitt 6.3).

6.1. FF1: Hochschuldidaktische Grundlagen

Durch das Verbundprojekt KEA-Mod wurde eine interdisziplinäre Zusammenarbeit zwischen den Disziplinen Psychologie (Fachgebiet: Hochschuldidaktik) und Informatik (Fachgebiet: Modellierung) initiiert. Dabei wurden die drei zentralen Fragen im kompetenzorientierten Prüfen nach METZGER und NÜESCH [MN04] im Kontext der grafischen Modellierung betrachtet: I - »Was wird geprüft?«, II - »Wie wird geprüft?« und III - »Wie wird bewertet?«.

Um Frage I beantworten zu können, wurde ein generisches **Kompetenzmodell für die grafische Modellierung** in einem vierstufigen Entwicklungsprozess entworfen (siehe Abschnitt 4.1). Bislang existierende Aufstellungen von Kompetenzen in Form von Lernzielen waren teilweise sehr abstrakt oder auf lediglich ein Spezialgebiet der Modellierung beschränkt (siehe Abschnitt 3.1.1). Das entwickelte Kompetenzmodell enthält insgesamt 70 Kompetenzfacetten und umfasst neben den zentralen Inhaltsbereichen »Modellverstehen & -interpretieren« und »Modellbilden & -modifizieren«

6. Zusammenfassung und Ausblick

auch motivationale Aspekte und fachübergreifende Kompetenzen. Unter Berücksichtigung des Kompetenzmodells können Lehrende ihre Lehrveranstaltungen und Prüfungen auf dem Gebiet der grafischen Modellierung kompetenzorientiert ausrichten, indem sie passende Kompetenzfacetten aus dem Kompetenzmodell auswählen.

Im Rahmen der Beantwortung der Frage II wurde ein **Aufgabenkatalog für die grafische Modellierung** entwickelt (siehe Abschnitt 4.2). Dieser geht über eine einfache Beschreibung von Aufgabentypen hinaus. Er verknüpft die enthaltenen Aufgaben mit den Kompetenzfacetten aus dem vorliegenden Kompetenzmodell. Dadurch ermöglicht er es Lehrenden, Modellierungsaufgaben gezielt gemäß der beabsichtigten Lernziele in Übungen und Prüfungen einzusetzen. Dies wurde durch existierende Aufgabenklassifikationen bislang nicht abgedeckt (siehe Abschnitt 4.4.2).

Zur Beantwortung der Frage III wurden auf Basis des Aufgabenkatalogs **Rubrics** entwickelt, die Lehrenden einen Vorschlag für die kriteriengeleitete und kompetenzorientierte Bewertung auf dem Gebiet der grafischen Modellierung liefern (siehe Abschnitt 4.3). Die Rubrics rücken weitere Kriterien in den Vordergrund, die bisher vernachlässigt wurden, beispielsweise pragmatische Qualitätsaspekte. Zudem wird die Abkehr von einer rein defizitorientierten Rückmeldung an Lernende eingeleitet, da hier auch die erreichten Lernziele deutlich gemacht werden. Ebenso wurden Empfehlungen für lernförderliche **Feedback-Nachrichten** auf Basis der Vorarbeiten von HATTIE und TIMPERLEY [HT07] ausgearbeitet.

Die Ergebnisse zu diesen drei zentralen Fragen des kompetenzorientierten Prüfens decken zusammengenommen die verschiedenen Phasen des Prüfungsprozesses ab. Angefangen bei der Definition der zu prüfenden Kompetenzen über die Formulierung passender Aufgaben bis hin zur Bewertung und dem Feedback für Lernende dienen sie als Werkzeug zur Unterstützung von Lehrenden. Neben den erfolgten wissenschaftlichen Publikationen in Fachzeitschriften [So22f; So23b] wurde auch eine anwendungsorientierte Handreichung für Lehrende erstellt [So22e]. Diese Handreichung soll die Einsatzmöglichkeiten des Kompetenzmodells im Rahmen der Hochschullehre verdeutlichen und anhand eines fiktiven Anwendungsszenarios eine didaktische Orientierungshilfe für die Anwendung durch Lehrende geben.

6.2. FF2: E-Assessment-Plattform

Um das kompetenzorientierte Prüfen auf dem Gebiet der grafischen Modellierung technisch zu unterstützen, wurde eine **E-Assessment-Plattform** entwickelt (siehe

Abschnitt 5.1). Die KEA-Mod-Plattform ermöglicht eine automatisierte Bewertung von Modellierungsaufgaben für verschiedene graphbasierte und grafische Modellierungssprachen. Mit den beiden Aufgabentypen »Modell verstehen« und »Modell erstellen« werden zentrale Aufgabentypen aus dem Aufgabenkatalog (siehe Abschnitt 4.2) abgebildet. Es können individuelle Bewertungsschemata und Textvorlagen für Feedback-Nachrichten konfiguriert werden. Die KEA-Mod-Plattform kann an den Identitätsanbieter der jeweiligen Hochschule angebunden werden und erlaubt den Einsatz in einem variablen Lehr- und Lernkontext (d.h. formativ bei Übungen und summativ bei Prüfungen). Aufgrund der Konzeption als Microservice-Architektur (siehe Abschnitt 5.1.1) ist es möglich, individuelle Bewertungsdienste oder bereits existierende Ansätze für die Überprüfung studentischer Modelle einzubinden. Im Übungs- und Prüfungseinsatz konnten Einsatzfähigkeit und Praxistauglichkeit der KEA-Mod-Plattform demonstriert werden (siehe Abschnitte 5.5.6 und 5.5.7). Die Vorstellung der KEA-Mod-Plattform ist über Konferenz- und Workshopbeiträge [UI21; UI22b; UI22c] und in einer Fachzeitschrift [St21] erfolgt. Zudem wurden Workshops auf der Fachtagung Modellierung (2020 und 2022) sowie auf der GI-Jahrestagung INFORMATIK (2021 und 2022) organisiert, um mit der Fachcommunity in Austausch zu treten und die Forschungsergebnisse zu diskutieren.

Im Rahmen des Forschungsvorhabens sind zudem mehrere **Bewertungsdienste für die Prüfung von Petri-Netzen** entstanden, die sich auf den Aufgabentyp »Modell erstellen« beziehen (siehe Abschnitt 5.3). Mit diesen Bewertungsdiensten werden verschiedene Kompetenzfacetten in Bezug auf die Qualitätsaspekte Syntax, Semantik und Pragmatik überprüft. Für die Berücksichtigung des Aspekts Semantik wurde ein Ansatz aus dem Prozessmodell-Matching [Sc18] auf die vorliegende Problemstellung übertragen, um den Vergleich eines studentischen Modells mit Musterlösungen zu ermöglichen. Eine Anwendung und detaillierte Auswertung der Ergebnisse der Bewertungsdienste ist auf einer Datenbasis von 2.194 studentischen Modellen erfolgt (siehe Abschnitt 5.5.2).

Weiterhin wurde eine umfangreiche **Literaturstudie** durchgeführt und in einer Fachzeitschrift veröffentlicht [UI23a]. Dabei wurden 110 Publikationen auf verschiedene Aspekte wie unter anderem eingesetzte technische Verfahren, Feedback-Arten und Einsatzkontext hin untersucht (siehe Abschnitt 3.3.2). Somit wurde ein **Überblick über den Stand der Forschung** zu bestehenden Ansätzen zur automatisierten Bewertung von graphbasierten Diagrammen im Hochschulkontext erstellt. Dieser stellt einen Ausgangspunkt für anknüpfende Forschungsfragen und die Ergänzung weiterer Bewertungsdienste in der KEA-Mod-Plattform dar.

6.3. Ausblick

Mit den im Forschungsvorhaben entwickelten hochschuldidaktischen Grundlagen (siehe Abschnitt 6.1) und der KEA-Mod-Plattform (siehe Abschnitt 6.2) wurden Ergebnisse geschaffen, die sowohl für die Anschlussforschung als auch für die praktische Nutzung in der Hochschullehre zur Verfügung stehen. Die Beteiligung der KEA-Mod-Plattform an den beiden Folgeprojekten PePP und DaTa-Pin (siehe Abschnitt 1.4) belegt insbesondere die Relevanz der Forschungsergebnisse für eine fortschreitende Digitalisierung im Lehr- und Prüfungskontext an Hochschulen. Ebenso wird die Bedeutung domänenspezifisch ausgerichteter E-Assessments unterstrichen. Mit diesen können auch Kompetenzen und Lernziele auf höheren Niveaustufen abgedeckt werden.

Bezüglich der hochschuldidaktischen Grundlagen steht zunächst die praktische Anwendung im Vordergrund. Eine Erprobung des Kompetenzmodells, des Aufgabenkatalogs sowie der Rubrics in zusätzlichen Lehrveranstaltungen könnte Hinweise auf Weiterentwicklungsbedarf liefern. So gilt es, die neu entwickelten Aufgabentypen im Aufgabenkatalog (siehe Abschnitt 4.2) in Lehrveranstaltungen einzusetzen und zu evaluieren. Denkbar ist auch eine Wiederholung des Expertenratings zur Validierung des Kompetenzmodells (siehe Abschnitt 4.1), um die weiterentwickelte Fassung des Kompetenzmodells zu berücksichtigen. Wünschenswert ist nicht zuletzt eine weiterführende Evaluation des Aspekts Lernförderlichkeit (siehe Abschnitt 4.4.3). Hierbei sollten die Auswirkungen des Einsatzes der KEA-Mod-Plattform in der Lehre und die Effekte verschiedener Arten von Feedback über Nutzendenbefragungen hinaus in Form von Kontrollgruppen-Studien empirisch untersucht werden. Auf diese Weise könnte überprüft werden, ob die KEA-Mod-Plattform neben den technischen Vorteilen (digitale Umsetzung und automatisierte Bewertung) auch einen Mehrwert im Hinblick auf die Lernförderlichkeit und die erzielten Lernergebnisse mitbringt.

Um die Akzeptanz einer digitalen Anwendung wie der KEA-Mod-Plattform durch die Nutzenden in der Lehre und in Prüfungen zu erhöhen, spielt die Benutzungsfreundlichkeit eine zentrale Rolle. Der bisherige Fokus auf eine leichte Erlernbarkeit und eine intuitive Gestaltung der Oberfläche sollte daraufhin ausgeweitet werden, die Erreichbarkeit der Funktionen in der KEA-Mod-Plattform stärker auf typische Arbeitsabläufe von Studierenden und Lehrenden auszurichten. Dafür bieten sich weitere Cognitive Walkthroughs sowohl mit Studierenden als auch Dozierenden an (siehe Abschnitt 5.6.9), um diese Abläufe zu erfassen. Ebenso sollten zusätzli-

che Funktionen wie beispielsweise ein Dashboard mit einer aggregierten Übersicht über die Ergebnisse von Studierenden ergänzt werden. Nur wenn die Gestaltung der Benutzungsoberfläche und die Funktionen der KEA-Mod-Plattform den Anforderungen ihrer Zielgruppe entsprechen, kann sie sich in Lehrveranstaltungen zur grafischen Modellierung als wichtiges Werkzeug etablieren.

Nachdem mit der KEA-Mod-Plattform eine grundlegende Infrastruktur geschaffen wurde, kann der Fokus nun auf die Weiterentwicklung bestehender Bewertungsdienste und der Erprobung weiterer Ansätze für die automatisierte Bewertung gelegt werden. Dies kann beispielsweise unter Einsatz von KI-Verfahren (siehe Abschnitt 5.6.6) oder durch die Integration bestehender Ansätze (siehe Abschnitte 3.3.1 und 3.3.2) erfolgen. Zudem sind weitere technische Ergänzungen anzustreben, um den Funktionsumfang der KEA-Mod-Plattform und deren Einsatzmöglichkeiten zu erweitern. Neben den bislang bestehenden könnten weitere graphbasierte, grafische Modellierungssprachen in den Modellierungseditor integriert und dazu passende Bewertungsdienste entwickelt werden. Mögliche Beispiele sind weitere UML-Diagrammtypen, Case Management Model and Notation (CMMN) [Ob16] oder Decision Model and Notation (DMN) bzw. Decision Requirements Diagram (DRD) [Ob23], die ebenfalls im Kontext des Geschäftsprozessmanagements anzutreffen sind. Ebenso ließen sich Modelltypen aus anderen informatikbezogenen Fachgebieten einbinden, z.B. Netzpläne oder Automaten. Auch die Einbindung graphbasierter Diagramme aus anderen Disziplinen wie z.B. Strukturformeln aus dem Gebiet der Chemie wäre umsetzbar.

Um das Spektrum an abgedeckten Kompetenzfacetten durch die KEA-Mod-Plattform zu erweitern, könnten zusätzliche Aufgabentypen ergänzt werden. Beispielsweise könnte ein Modell gezeigt werden, mit der Aufforderung, bestimmte Elemente oder Bereiche in diesem Modell zu markieren und optional zu beschreiben. Damit könnte der Aufgabentyp »Modell prüfen« bzw. »Fehler im Modell finden« aus dem Aufgabenkatalog (siehe Abschnitt 4.2) umgesetzt werden. Mit diesem lassen sich Kompetenzfacetten aus der Inhaltsdimension »Modellverstehen & -interpretieren« und der Prozessdimension »Analysieren & Bewerten« adressieren. Möglich ist auch die Erweiterung des bestehenden Aufgabentyps »Modell erstellen« dahingehend, dass bei Beginn der Bearbeitung ein vorgegebenes Modell im Modellierungseditor geladen wird. Dadurch können Aufgaben umgesetzt werden, welche die Vervollständigung oder Anpassung eines bestehenden Modells fordern (siehe Abschnitt 4.2). Um über ergebnisbezogene Kompetenzfacetten (anhand der Lösung messbar) hinaus auch prozessbezogene Kompetenzfacetten (anhand Denken und Handeln

6. Zusammenfassung und Ausblick

messbar) bewerten zu können, kommt die Berücksichtigung von Interaktionsdaten in Frage, die während des Modellierungsprozesses aufgezeichnet werden. Anhand dieser Interaktionsdaten lassen sich alle Arbeitsschritte im Modellierungsektor nachvollziehen, die an der Erstellung eines Modells beteiligt sind. Dadurch kann zumindest das Handeln nachvollzogen werden, was z. B. Rückschlüsse auf eine planvolle und strategische Herangehensweise erlaubt (siehe Kompetenzfacetten MK 3.01 und MK 3.02 aus dem Kompetenzmodell [So22b]). Anzustreben ist zudem eine vertiefte Auseinandersetzung mit möglichen Darstellungsformen des Bewertungsergebnisses im Hinblick auf das der Bewertung zugrundeliegende Bewertungsschema (siehe Abschnitt 2.1.4). Idealerweise wird dabei ein direkter Bezug der verwendeten Bewertungskriterien zu den jeweiligen Lernzielen hergestellt, so dass Lernende einen Überblick darüber erhalten, inwieweit die Lernziele erfüllt wurden.

Obwohl die digitale Handhabung studentischer Einreichungen und die automatisierte Bewertung durch die KEA-Mod-Plattform Lehrenden eine Unterstützung bieten, beansprucht die didaktisch sinnvolle Aufbereitung und initiale Erstellung kompetenzorientierter Übungs- und Prüfungsaufgaben nach wie vor eine beträchtliche Menge an Zeit. In der KEA-Mod-Plattform wurde daher ein öffentlicher Aufgabenpool angelegt, über welchen Lehrende von ihnen erstellte Aufgaben teilen können. Studierende können diesen Aufgabenpool ebenso nutzen, um ihre Fähigkeiten mit entsprechenden Aufgaben zu erproben. Eine Veröffentlichung unter einer offenen Lizenz als Open Educational Resource (OER) könnte sowohl die Weiterverbreitung als auch die gemeinschaftliche Weiterentwicklung einer Aufgabensammlung fördern. Ein weiterer Ansatz besteht in der Entwicklung von Werkzeugen zur automatisierten Aufgabenerstellung. Derartige Aufgabengeneratoren könnten an die KEA-Mod-Plattform angeschlossen werden und somit die Erstellung von Aufgaben technisch unterstützen. Beispiele für Ansätze zur automatisierten Generierung von Aufgaben aus dem Gebiet graphbasierter Diagramme wurden auf einem MoHoL-Workshop (siehe Abschnitt 1.3) bereits vorgestellt [Br22; CLM22] und bieten sich für eine künftige Zusammenarbeit an.

Neben den zuvor beschriebenen technischen und didaktischen Herausforderungen dürfen auch organisatorische und rechtliche Aspekte nicht vernachlässigt werden. Dahingehend wurden beim Prüfungseinsatz der KEA-Mod-Plattform im Rahmen des Folgeprojekts PePP bereits einige Herausforderungen identifiziert [UI23b]. Aus organisatorischer Sicht besteht Bedarf, die KEA-Mod-Plattform in bestehende Lehr-Lernsysteme wie z. B. ILIAS einzubinden, um einen Medienbruch zu vermeiden und auf weitere, nicht-fachspezifische Aufgabentypen (Sortieraufgaben, Lückentexte

etc.) zugreifen zu können, die durch das Lehr-Lernsystem gegebenenfalls bereits zur Verfügung gestellt werden. Ebenso muss bei der Durchführung einer elektronischen Prüfung geklärt werden, an welchem Ort (remote – mit oder ohne Videoüberwachung bzw. Proctoring – oder in Präsenz) und mit welchen Geräten (z. B. »Bring your own device«, BYOD oder auf Pool-Rechnern) diese durchgeführt werden soll. Weiterhin muss durch technische Maßnahmen sichergestellt werden, dass während der Prüfung kein Zugriff auf unerlaubte Webseiten oder Anwendungen möglich ist, beispielsweise durch die Verwendung sogenannter Safe-Exam-Browser und die Einrichtung von speziellen Firewall-Regeln.

Die Auseinandersetzung mit den rechtlichen Rahmenbedingungen hat gezeigt, dass an Hochschulen teilweise unterschiedliche Regelungen bestehen. So wird der Aufgabentyp »Modell verstehen« (siehe Abschnitt 5.4) in seiner aktuellen Form zwar von der Universität Paderborn empfohlen, jedoch durch die Prüfungsordnung des KIT nicht abgedeckt. Auch müssen die Anforderungen an eine datenschutzkonforme Umsetzung sowie die rechtssichere Archivierung im Prüfungskontext berücksichtigt werden. Wird eine Instanz der KEA-Mod-Plattform an einem bestimmten Standort betrieben, erfordert die Nutzung dieser Instanz durch andere Hochschulen zunächst eine individuelle datenschutzrechtlich passende Vereinbarung (z. B. einen Auftragsdatenverarbeitungsvertrag). All dies sind nur einige Beispiele für weitere Herausforderungen, die es zu überwinden gilt, um den Weg zu einem flächendeckenden Einsatz elektronischer Prüfungen im Hochschulkontext zu ebnen. Die Erfahrungen aus dem Prüfungseinsatz der KEA-Mod-Plattform und aus den anderen Teilprojekten im Verbundprojekt PePP haben einen Beitrag dazu geleistet, die erforderlichen nächsten Schritte auf diesem Weg aufzuzeigen.

Anhang

A. Ergänzung zur Literaturstudie

Tab. A.1.: Berücksichtigte Publikationen der Literaturstudie [UI23a]. Die Spalte QGS gibt an, ob die Publikation im verwendeten Quasi-Gold-Standard enthalten ist. Abkürzungen der Modelltypen: BD - Blockdiagramm, ER - Entity Relationship Diagramm, CASE - Computer Aided Software Engineering, UML-C - UML-Klassendiagramm, UML-S - UML-Sequenzdiagramm, UML-SD - UML-Zustandsdiagramm, UML-A - UML-Aktivitätsdiagramm, DFA - Deterministischer endlicher Automat, NFA - Nichtdeterministischer endlicher Automat, PDA - Kellerautomat.

Publikation	QGS	Jahr	Modelltyp(en)
HALL und GORDON [HG98]		1998	ER
HOGGARTH und LOCKYER [HL98]	x	1998	CASE
CONSTANTINO-GONZÁLEZ et al. [CSS00]		2000	ER
MITROVIC et al. [Mi01]		2001	ER
HIGGINS et al. [HST02]		2002	Flowchart, Schaltplan
SURAWEEERA und MITROVIC [SM02]	x	2002	ER
CONSTANTINO-GONZÁLEZ et al. [CSS03]		2003	ER
SMITH et al. [STW04]		2004	universell
SURAWEEERA et al. [SMM04]		2004	ER
SURAWEEERA und MITROVIC [SM04]		2004	ER
WAUGH et al. [WTS04]		2004	ER
BAGHAEI und MITROVIC [BM05]		2005	UML-C
BAGHAEI et al. [BMI05]		2005	UML-C
BLANK et al. [BI05]		2005	UML-C
MORITZ et al. [Mo05b]		2005	UML-C (weitere möglich)
THOMAS et al. [TWS05]	x	2005	universell (ER)
TSELONIS et al. [TSM05]		2005	universell (ER, UML-C)
ZAKHAROV et al. [ZMO05]		2005	EER
BAGHAEI et al. [BMI06]		2006	UML-C
BAGHAEI und MITROVIC [BM06]		2006	UML-C
BATMAZ und HINDE [BH06]	x	2006	ER

wird fortgesetzt auf der nachfolgenden Seite

A. Ergänzung zur Literaturstudie

Tab. A.1 – fortgesetzt von der vorherigen Seite

Publikation	QGS	Jahr	Modelltyp(en)
HIGGINS und BLIGH [HB06]	x	2006	universell (ER)
LE [Le06]		2006	UML-C
PRADOS et al. [Pr06]		2006	ER
THOMAS et al. [TWS06]	x	2006	ER
BAGHAEI et al. [BMI07]		2007	UML-C
BAGHAEI und MITROVIC [BM07b]		2007	UML-C
BAGHAEI und MITROVIC [BM07a]		2007	UML-C
BATMAZ und HINDE [BH07]		2007	ER
COELHO und MURPHY [CM07]		2007	UML-C
THOMAS et al. [TSW07b]		2007	universell (ER)
THOMAS et al. [TWS07]		2007	universell (ER)
THOMAS et al. [TSW07a]	x	2007	universell (ER)
TSELONIS und SARGEANT [TS07]		2007	universell (ER)
WAUGH et al. [WTS07]	x	2007	ER
AUXEPAULES et al. [APL08]		2008	UML-C
PARVEZ und BLANK [PB08]		2008	UML-C
SHUKUR und MOHAMED [SM08]	x	2008	Automaten
THOMAS et al. [TSW08b]		2008	universell (ER)
THOMAS et al. [TWS08]		2008	universell (UML-S)
THOMAS et al. [TSW08a]	x	2008	universell (UML-S)
BATMAZ et al. [BSH09]		2009	ER
DEMUTH und WEIGEL [DW09]		2009	UML-C
JAYAL und SHEPPERD [JS09]	x	2009	universell (UML-A)
STONE et al. [SBH09]		2009	universell (ER)
THOMAS et al. [TSW09b]		2009	universell (ER)
THOMAS et al. [TSW09a]	x	2009	universell (ER)
THOMAS et al. [TWS09]		2009	universell (ER, UML-S)
SMITH et al. [STW10]	x	2010	universell (UML-C, UML-S)
SOLER et al. [So10b]		2010	UML-C
SOLER et al. [So10a]	x	2010	UML-C
SURAWEERA et al. [SMM10]		2010	universell (ER)
BOLLOJU et al. [BLB11]		2011	UML-C
GUO et al. [Gu11]		2011	universell (ER, Flowchart)
HASKER und ROWE [HR11]		2011	UML-C, UML-UC
HASKER [Ha11]		2011	UML-C
HOLLAND et al. [Ho11]		2011	UML (unspezifiziert)
NÉMEC und FASUGA [NF11]		2011	universell (ER, UML-C, Schaltplan)
SOLER et al. [So11]		2011	ER
STRIEWE und GOEDICKE [SG11]	x	2011	UML-C (weitere möglich)
THOMAS et al. [TWS11]	x	2011	universell (ER, UML-S)
IFLAND et al. [If12]		2012	UML-C
SCHRAMM et al. [Sc12b]		2012	UML-C, UML-A

wird fortgesetzt auf der nachfolgenden Seite

Tab. A.1 – fortgesetzt von der vorherigen Seite

Publikation	QGS	Jahr	Modelltyp(en)
THOMAS et al. [TWS12]		2012	universell
VACHHARAJANI et al. [VPG12]		2012	UML-UC
WESTERGAARD et al. [WFS12]	x	2012	Gefärbtes Petri-Netz
ALUR et al. [Al13]	x	2013	DFA
PY et al. [PAA13]		2013	UML-C
SIEPERMANN et al. [SSL13]		2013	Netzplan (Metra Potential Methode)
SMITH et al. [STW13]	x	2013	universell (ER, UML-S)
THOMAS [Th13]		2013	universell (ER)
WESTERGAARD et al. [WFS13]	x	2013	Gefärbtes Petri-Netz
BAÑERES et al. [Ba14]		2014	Schaltplan
HASKER und SHI [HS14]		2014	UML-C
OUTAIR et al. [OLT14]		2014	UML-C, UML-SM, UML-S, UML-UC
SHEKHAR et al. [Sh14]		2014	DFA
STRIEWE und GOEDICKE [SG14]	x	2014	UML-A
BECK et al. [Be15]		2015	UML-C, UML-A
D'ANTONI et al. [DA05]		2015	DFA
ELMADANI et al. [El15]		2015	EER
SOUSA und LEAL [SL15]	x	2015	universell
DARSHAN und KUMAR [DK16]		2016	DFA
MITROVIC und SURAWEEERA [MS15a]		2016	ER
REISCHMANN und KUCHEN [RK16]	x	2016	UML-C
SIEPERMANN [Si16]		2016	ER
THALER et al. [Th16]	x	2016	EPC
CORREIA et al. [CLP17]	x	2017	universell
ECHEVERRIA et al. [Ec17]		2017	ER
AL-HOQANI [Al18]		2018	Entscheidungsbaum
CORREIA et al. [CLP18]	x	2018	universell
DAEHLI et al. [Da18]		2018	ER
JAYAWARDENA et al. [Ja18]		2018	Flowchart, Schaltplan, BD
REISCHMANN und KUCHEN [RK18]		2018	UML-C
BIAN et al. [BAK19]		2019	UML-C
ICHINOHE et al. [Ic19]		2019	UML-C
LINO et al. [Li19]		2019	ER
OGATA et al. [OKO19]		2019	UML-SM
OGATA und KAYAMA [OK19]		2019	UML-SM
REISCHMANN und MENEZES [RM19]		2019	UML-C
REISCHMANN und KUCHEN [RK19]		2019	UML-C
STIKKOLORUM et al. [St19]		2019	UML-C
VACHHARAJANI und PAREEK [VP19]		2019	UML-UC
VAN DOORN et al. [VSV19]		2019	UML-C
BIAN et al. [BAK20]		2020	UML-C
BOUBEKEUR et al. [BMM20]		2020	UML-C

wird fortgesetzt auf der nachfolgenden Seite

A. Ergänzung zur Literaturstudie

Tab. A.1 – fortgesetzt von der vorherigen Seite

Publikation	QGS	Jahr	Modelltyp(en)
D'ANTONI et al. [DA20]		2020	DFA, NFA, PDA
FAUZAN et al. [Fa20]		2020	UML-UC
SÁNCHEZ-FERRERES et al. [Sà20]		2020	BPMN (EPC und Petri-Netz möglich)
VACHHARAJANI und PAREEK [VP20]		2020	UML-UC
OUTAIR et al. [OMA21]		2021	UML-C

B. Ergänzung zur Evaluation

Aufgabe A2: Modellieren Sie die folgende Prozessbeschreibung als Ablaufmodell in Form eines Petri-Netzes. Verwenden Sie für die Beschriftung der Modellelemente die englische Sprache.

›On an incoming loan application, the customer data is checked and a credit report is requested. If the customer data check shows that the customer data is not yet in the system, the customer data is entered in the system. As soon as the customer data is entered in the system and the credit report is available, a security check will be carried out. In case of a positive result of the security check, the application is approved; if the result of the security check is negative, the application is rejected. Afterwards, the loan application is closed.‹

Musterlösung:

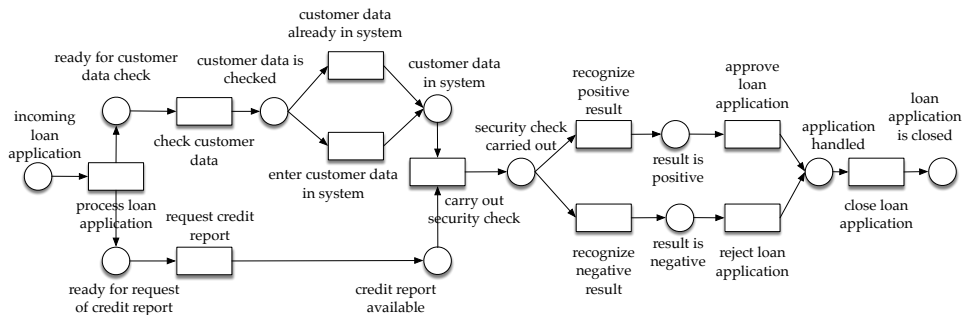


Abb. B.1.: Aufgabe A2 (Beantragung eines Kredits) mit Musterlösung

B. Ergänzung zur Evaluation

Aufgabe A3: Modellieren Sie die folgende Prozessbeschreibung als Ablaufmodell in Form eines Petri-Netzes. Verwenden Sie für die Beschriftung der Modellelemente die englische Sprache.

›The process starts with the booking request of a rental car. In the next step, either an extra insurance is added or the extra insurance is skipped. Afterwards it is possible to change the configuration of the rental car booking as many times as required until the configuration is finally confirmed. When the customer is checking in the rental car facility, he can again choose to add or skip an extra insurance. At the same time, he has to select a car and have his driver’s license checked. Additionally, his credit card has to be charged. After all those steps, the rental car is finally supplied to him and the process is closed.‹

Musterlösung:

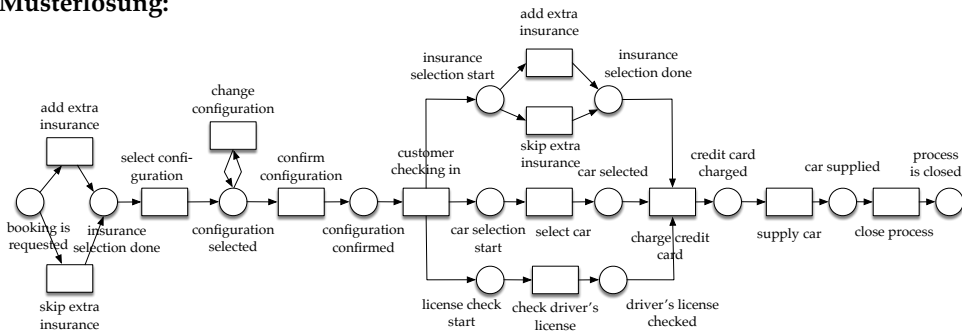


Abb. B.2.: Aufgabe A3 (Mietwagenverleih) mit Musterlösung

Tab. B.1.: Einzelergebnisse: Precision, Recall und F-Score auf einer Stichprobe von 50 Modellen der Datenbasis. TP: richtig positiv, TN: richtig negativ, FP: falsch positiv, FN: falsch negativ, GE: Gesamtergebnis (alle Bewertungsdienste).

Modell	Sem.	Aufg.	TP	TN	FP	FN	Precision	Recall	F-Score	GE
ex1_16_7.kea	WS1617	A1	10	0	0	0	1,0000	1,0000	1,0000	0,350
ex1_16_73.kea	WS1617	A1	14	0	0	0	1,0000	1,0000	1,0000	0,588
ex1_16_76.kea	WS1617	A1	28	0	0	0	1,0000	1,0000	1,0000	0,900
ex1_16_96.kea	WS1617	A1	20	0	1	1	0,9524	0,9524	0,9524	0,575
ex1_16_99.kea	WS1617	A1	10	1	0	0	1,0000	1,0000	1,0000	0,500
ex1_17_146.kea	WS1718	A1	16	0	0	0	1,0000	1,0000	1,0000	0,400
ex1_17_196.kea	WS1718	A1	21	1	0	0	1,0000	1,0000	1,0000	0,425
ex1_17_2.kea	WS1718	A1	9	0	1	0	0,9000	1,0000	0,9474	0,500
ex1_17_203.kea	WS1718	A1	24	0	0	0	1,0000	1,0000	1,0000	1,000

wird fortgesetzt auf der nachfolgenden Seite

Tab. B.1 – fortgesetzt von der vorherigen Seite

Modell	Sem.	Aufg.	TP	TN	FP	FN	Precision	Recall	F-Score	GE
ex1_17_62.kea	WS1718	A1	21	0	0	3	1,0000	0,8750	0,9333	0,750
ex1_18_1.kea	WS1819	A1	14	0	0	0	1,0000	1,0000	1,0000	0,588
ex1_18_133.kea	WS1819	A1	23	0	0	0	1,0000	1,0000	1,0000	0,588
ex1_18_184.kea	WS1819	A1	14	2	0	0	1,0000	1,0000	1,0000	0,375
ex1_18_42.kea	WS1819	A1	16	0	2	0	0,8889	1,0000	0,9412	0,650
ex1_18_47.kea	WS1819	A1	18	0	0	3	1,0000	0,8571	0,9231	0,675
ex1_19_22.kea	WS1920	A1	20	0	1	0	0,9524	1,0000	0,9756	0,663
ex1_19_52.kea	WS1920	A1	18	0	0	0	1,0000	1,0000	1,0000	0,538
ex1_19_60.kea	WS1920	A1	29	0	0	0	1,0000	1,0000	1,0000	0,813
ex1_19_9.kea	WS1920	A1	21	0	0	0	1,0000	1,0000	1,0000	0,588
ex1_19_98.kea	WS1920	A1	31	0	0	0	1,0000	1,0000	1,0000	0,900
ex1_20_145.kea	WS2021	A1	19	0	0	0	1,0000	1,0000	1,0000	1,000
ex1_20_149.kea	WS2021	A1	17	0	0	5	1,0000	0,7727	0,8718	0,500
ex1_20_230.kea	WS2021	A1	8	0	1	0	0,8889	1,0000	0,9412	0,475
ex1_20_29.kea	WS2021	A1	24	1	0	0	1,0000	1,0000	1,0000	0,625
ex1_20_66.kea	WS2021	A1	31	4	1	0	0,9688	1,0000	0,9841	0,900
ex1_21_125.kea	WS2122	A1	18	5	0	3	1,0000	0,8571	0,9231	0,388
ex1_21_153.kea	WS2122	A1	13	0	0	0	1,0000	1,0000	1,0000	0,638
ex1_21_68.kea	WS2122	A1	16	0	1	0	0,9412	1,0000	0,9697	0,513
ex1_21_70.kea	WS2122	A1	19	0	0	0	1,0000	1,0000	1,0000	1,000
ex1_21_99.kea	WS2122	A1	17	0	3	0	0,8500	1,0000	0,9189	0,550
ex2_17_39.kea	WS1718	A2	16	0	3	0	0,8421	1,0000	0,9143	0,500
ex2_17_84.kea	WS1718	A2	20	1	0	0	1,0000	1,0000	1,0000	0,450
ex2_18_170.kea	WS1819	A2	25	0	2	0	0,9259	1,0000	0,9615	0,813
ex2_18_180.kea	WS1819	A2	27	0	1	0	0,9643	1,0000	0,9818	0,988
ex2_19_147.kea	WS1920	A2	4	1	1	1	0,8000	0,8000	0,8000	0,000
ex2_19_5.kea	WS1920	A2	19	0	2	0	0,9048	1,0000	0,9500	0,575
ex2_19_74.kea	WS1920	A2	2	0	1	0	0,6667	1,0000	0,8000	0,000
ex2_19_76.kea	WS1920	A2	4	7	0	3	1,0000	0,5714	0,7273	0,300
ex2_20_137.kea	WS2021	A2	13	1	2	1	0,8667	0,9286	0,8966	0,300
ex2_20_95.kea	WS2021	A2	23	0	0	0	1,0000	1,0000	1,0000	0,775
ex3_17_77.kea	WS1718	A3	2	17	2	17	0,5000	0,1053	0,1739	0,375
ex3_17_78.kea	WS1718	A3	28	2	3	3	0,9032	0,9032	0,9032	0,725
ex3_17_96.kea	WS1718	A3	20	0	2	0	0,9091	1,0000	0,9524	0,738
ex3_21_110.kea	WS2122	A3	19	0	4	0	0,8261	1,0000	0,9048	0,438
ex3_21_147.kea	WS2122	A3	23	0	3	1	0,8846	0,9583	0,9200	0,725
ex3_21_61.kea	WS2122	A3	6	0	1	2	0,8571	0,7500	0,8000	0,450
ex3_21_79.kea	WS2122	A3	22	0	0	0	1,0000	1,0000	1,0000	0,863
ex3_21_86.kea	WS2122	A3	19	5	1	3	0,9500	0,8636	0,9048	0,350
ex3_22_145.kea	WS2223	A3	21	0	1	0	0,9545	1,0000	0,9767	0,700
ex3_22_33.kea	WS2223	A3	21	1	8	0	0,7241	1,0000	0,8400	0,550

B. Ergänzung zur Evaluation

Tab. B.2.: Gegenüberstellung manuelle Korrektur (M) und Semantik-Checker (S) auf einer Stichprobe von 50 Modellen im Hinblick auf die Verletzungen der Semantik-Kriterien SEM01-SEM04. S1M: Manuell identifizierte Verletzung von Kriterium SEM01, S1S: Vom Semantik-Checker dargestellte Verletzung von Kriterium SEM01. Weitere Kriterien analog. TP: Teilpunktzahl (maximal 10 Punkte) bestimmt durch den Semantik-Checker.

Modell	Sem.	Aufg.	S1M	S2M	S3M	S4M	S1S	S2S	S3S	S4S	TP
ex1_16_7.kea	WS1617	A1	2	3	0	1	2	3	0	0	4,5
ex1_16_73.kea	WS1617	A1	0	5	0	0	0	2	0	0	5,0
ex1_16_76.kea	WS1617	A1	0	0	0	2	0	0	0	0	9,0
ex1_16_96.kea	WS1617	A1	1	3	0	3	1	1	0	0	4,5
ex1_16_99.kea	WS1617	A1	4	0	1	1	4	0	1	0	5,0
ex1_17_146.kea	WS1718	A1	1	6	0	4	1	6	0	0	1,0
ex1_17_196.kea	WS1718	A1	0	3	1	4	0	3	1	0	4,5
ex1_17_2.kea	WS1718	A1	8	0	0	2	8	0	0	0	1,0
ex1_17_203.kea	WS1718	A1	0	0	0	2	0	0	0	0	9,0
ex1_17_62.kea	WS1718	A1	0	2	0	2	0	2	0	0	7,0
ex1_18_1.kea	WS1819	A1	0	5	0	0	0	2	0	0	5,0
ex1_18_133.kea	WS1819	A1	0	3	0	4	0	3	0	0	5,0
ex1_18_184.kea	WS1819	A1	5	0	2	2	5	0	2	0	3,0
ex1_18_42.kea	WS1819	A1	0	4	0	1	0	2	0	0	5,5
ex1_18_47.kea	WS1819	A1	0	0	0	1	0	0	0	0	9,5
ex1_19_22.kea	WS1920	A1	0	2	0	1	0	2	0	0	7,5
ex1_19_52.kea	WS1920	A1	8	0	0	3	8	0	0	0	0,5
ex1_19_60.kea	WS1920	A1	0	2	0	4	0	2	0	0	6,0
ex1_19_9.kea	WS1920	A1	0	2	0	3	0	2	0	0	6,5
ex1_19_98.kea	WS1920	A1	0	0	0	4	0	0	0	0	8,0
ex1_20_145.kea	WS2021	A1	0	0	0	0	0	0	0	0	10
ex1_20_149.kea	WS2021	A1	1	0	0	2	1	0	0	0	8,0
ex1_20_230.kea	WS2021	A1	1	2	0	0	1	2	0	0	7,0
ex1_20_29.kea	WS2021	A1	0	3	1	4	0	2	1	0	4,5
ex1_20_66.kea	WS2021	A1	0	0	4	3	0	0	4	0	6,5
ex1_21_125.kea	WS2122	A1	0	4	0	3	0	4	0	0	4,5
ex1_21_153.kea	WS2122	A1	6	1	0	0	6	0	0	0	3,0
ex1_21_68.kea	WS2122	A1	0	2	0	2	0	2	0	0	7,0
ex1_21_70.kea	WS2122	A1	0	1	0	0	0	0	0	0	9,0
ex1_21_99.kea	WS2122	A1	0	3	0	2	0	3	0	0	6,0
ex2_17_39.kea	WS1718	A2	4	2	0	0	4	1	0	0	4,0
ex2_17_84.kea	WS1718	A2	3	1	0	1	3	1	0	0	5,5
ex2_18_170.kea	WS1819	A2	0	0	0	2	0	0	0	0	9,0

wird fortgesetzt auf der nachfolgenden Seite

Tab. B.2 – fortgesetzt von der vorherigen Seite

Modell	Sem.	Aufg.	S1M	S2M	S3M	S4M	S1S	S2S	S3S	S4S	TP
ex2_18_180.kea	WS1819	A2	0	0	1	1	0	0	1	0	9,0
ex2_19_147.kea	WS1920	A2	8	2	1	0	8	2	0	0	0,0
ex2_19_5.kea	WS1920	A2	1	4	0	1	1	4	0	0	4,5
ex2_19_74.kea	WS1920	A2	10	2	0	0	10	2	0	0	0,0
ex2_19_76.kea	WS1920	A2	6	4	0	6	1	0	0	0	0,0
ex2_20_137.kea	WS2021	A2	3	0	0	0	3	0	0	0	7,0
ex2_20_95.kea	WS2021	A2	0	1	0	1	0	1	0	0	8,5
ex3_17_77.kea	WS1718	A3	1	0	7	0	0	0	7	0	5,5
ex3_17_78.kea	WS1718	A3	0	0	2	0	0	0	0	0	9,0
ex3_17_96.kea	WS1718	A3	1	0	0	0	1	0	0	0	9,0
ex3_21_110.kea	WS2122	A3	1	1	0	0	1	1	0	0	8,0
ex3_21_147.kea	WS2122	A3	0	1	1	0	0	1	1	0	8,5
ex3_21_61.kea	WS2122	A3	9	1	0	0	9	0	0	0	0,0
ex3_21_79.kea	WS2122	A3	1	0	0	0	1	0	0	0	9,0
ex3_21_86.kea	WS2122	A3	0	0	5	0	0	0	5	0	7,5
ex3_22_145.kea	WS2223	A3	3	0	0	0	3	0	0	0	7,0
ex3_22_33.kea	WS2223	A3	1	1	5	0	1	0	5	0	5,5

Literaturverzeichnis

- [Aa16] van der Aalst, W. M. P.: Process mining. Data Science in Action. Springer, 2016.
- [Aa96] van der Aalst, W. M. P., Structural Characterizations of Sound Workflow Nets, Computing Science Reports No. 96. Eindhoven University of Technology, 1996.
- [Aa97] van der Aalst, W. M. P.: Verification of Workflow nets. In (Azéma, P. und Balbo, G., Hrsg.): Proc. Int. Conf. on Application and Theory of Petri Nets (ICATPN), Toulouse, Frankreich. Bd. 1248. LNCS, Springer, S. 407–426, 1997.
- [Aa98] van der Aalst, W. M. P.: The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems and Computers 8/1, S. 21–66, 1998.
- [AH04] van der Aalst, W. M. P. und van Hee, K. M.: Workflow Management: Models, Methods and Systems. The MIT Press, 2004.
- [AK01] Anderson, L. W. und Krathwohl, P.: A Taxonomy For Learning, Teaching and Assessing - A Revision of Bloom's Taxonomy of Educational Objectives. Addison Wesley, 2001.
- [AKS99] Arnold, R.; Krämer-Stürzel, A. und Siebert, H.: Dozentenleitfaden. Planung und Unterrichtsvorbereitung in Fortbildung und Erwachsenenbildung. Cornelsen, 1999.
- [Al13] Alur, R.; D'Antoni, L.; Gulwani, S.; Kini, D. und Viswanathan, M.: Automated Grading of DFA Constructions. In (Rossi, F., Hrsg.): Proc. 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI), Peking, China. S. 1976–1982, 2013.
- [Al18] Al-Hoqani, W. M.: Recommendation of New Semi-Automated Approach of Decision Tree Diagram. In (Chen, L.; Serbedzija, N.; Makki, K. und Mallat, N. K., Hrsg.): Proc. 5th SAI Computing Conf., London, UK. IEEE, S. 1123–1130, 2018.
- [ALR15] van der Aa, H.; Leopold, H. und Reijers, H. A.: Detecting Inconsistencies Between Process Models and Textual Descriptions. In (Motahari-Nezhad, H. R.; Recker, J. und Weidlich, M., Hrsg.): Proc. 13th Int. Conf on Business Process Management (BPM), Innsbruck, Österreich. Bd. 9253. LNCS, Springer, S. 90–105, 2015.
- [ALR17] van der Aa, H.; Leopold, H. und Reijers, H. A.: Comparing textual descriptions to process models – The automatic detection of inconsistencies. Information Systems 64, S. 447–460, 2017.
- [An10] Antonucci, Y. L.: Business Process Management Curriculum. In (vom Brocke, J. und Rosemann, M., Hrsg.): Handbook on Business Process Management 2. International Handbooks on Information Systems, Springer, S. 423–442, 2010.

Literaturverzeichnis

- [An15] Antunes, G.; Bakhshandeh, M.; Borbinha, J.; Cardoso, J.; Dadashnia, S.; Francescomarino, Chiara Di Dragoni, M.; Fettke, P.; Gal, A.; Ghidini, C.; Hake, P.; Khat, A.; Klinkmüller, C.; Kuss, E.; Leopold, H.; Loos, P.; Meilicke, C.; Niesen, T.; Pesquita, C.; Péus, T.; Schoknecht, A.; Shee-trit, E.; Sonntag, A.; Stuckenschmidt, H.; Thaler, T.; Weber, I. und Weidlich, M.: The Process Model Matching Contest 2015. In (Kolb, J.; Leopold, H. und Mendling, J., Hrsg.): Proc. 7th Enterprise Modelling and Information Systems Architectures (EMISA), Innsbruck, Österreich. Bd. P-248. LNI, Gesellschaft für Informatik e.V., S. 127–155, 2015.
- [APL08] Auxepaules, L.; Py, D. und Lemeunier, T.: A Diagnosis Method that Matches Class Diagrams in a Learning Environment for Object-Oriented Modeling. In (Kinshuk, P. D.; Aedo, I. und Mora, E., Hrsg.): Proc. 8th IEEE Int. Conf. on Advanced Learning Technologies (ICALT), Santander, Spanien. IEEE, S. 26–30, 2008.
- [AS11] van der Aalst, W. M. P. und Stahl, C.: Modeling Business Processes. A Petri Net-Oriented Approach. The MIT Press, 2011.
- [As20] Association for Computing Machinery (ACM): Curricula Recommendations, 2020, Stand: 21. 06. 2020.
- [Au12] Aubertin, I.; Houy, C.; Fettke, P. und Loos, P., Stand der Lehrbuchliteratur zum Geschäftsprozessmanagement – Eine quantitative Analyse, Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi). Nr. 194. Deutsches Forschungszentrum für Künstliche Intelligenz, 2012.
- [Ba07] Bandara, W.; Rosemann, M.; Davies, I. und Tan, C.: A Structured Approach to Determining Appropriate Content for Emerging Information Systems Subjects : An Example for BPM Curricula Design. In (Toleman, M.; Cater-Steel, A. und Roberts, D., Hrsg.): Proc. 18th Australasian Conf. on Information Systems (ACIS), Toowoomba, Australien. Bd. 2007, S. 1132–1141, 2007.
- [Ba08] Balzert, H.: Lehrbuch der Softwaretechnik: Softwaremanagement. Springer Spektrum, 2008.
- [Ba09] Balzert, H.: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Springer Spektrum, 2009.
- [Ba10] Bandara, W.; Chand, D. R.; Chircu, A. M.; Hintringer, S.; Karagianis, D.; Recker, J.; van Rensburg, A.; Usoff, C. und Welke, R. J.: Business Process Management Education in Academia: Status, Challenges and Recommendations. Communications of the Association for Information Systems 27, S. 743–776, 2010.
- [Ba14] Bañeres, D.; Clarisó, R.; Jorba, J. und Serra, M.: Experiences in Digital Circuit Design Courses: A Self-Study Platform for Learning Support. IEEE Transactions on Learning Technologies 7/4, S. 360–374, 2014.
- [Ba90] Baumgarten, B.: Petri-Netze. Grundlagen und Anwendungen. BI Wissenschaftsverlag, 1990.
- [BAK19] Bian, W.; Alam, O. und Kienzle, J.: Automated Grading of Class Diagrams. In (Pretschner, A.; Voss, S. und Burgueño, L., Hrsg.): Proc. 22nd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), München, Deutschland. IEEE, S. 700–709, 2019.
- [BAK20] Bian, W.; Alam, O. und Kienzle, J.: Is Automated Grading of Models Effective? Assessing Automated Grading of Class Diagrams. In (Syriani, E.; Sahrhoui, H. A.; de Lara, J. und Abrahão, S., Hrsg.): Proc. 23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), Montreal, Kanada/Virtuell. ACM, S. 365–376, 2020.

- [Be15] Beck, P.-D.; Mahlmeister, T.; Ifland, M. und Puppe, F.: COCLAC - Feedback Generation for Combined UML Class and Activity Diagram Modeling Tasks. In (Priss, U. und Striewe, M., Hrsg.): Proc. 2nd Workshop Automatische Bewertung von Programmieraufgaben (ABP), Wolfenbüttel, Deutschland. Bd. 1496, CEUR-WS, 2015.
- [Be98] Berendt, B.: How to Support and to Bring About the Shift from Teaching to Learning through Academic Staff Development Programmes: Examples and Perspectives. *Higher Education in Europe* 23/3, S. 317–329, 1998.
- [BFP11] Betsch, T.; Funke, J. und Plessner, H.: Denken - Urteilen, Entscheiden, Problemlösen. Springer, 2011.
- [BH06] Batmaz, F. und Hinde, C.: A Diagram Drawing Tool for Semi-Automatic Assessment of Conceptual Database Diagrams. In (Danson, M., Hrsg.): Proc. 10th Int. Computer Assisted Assessment Conf. (CAA), Leicestershire, UK. Loughborough University, S. 71–84, 2006.
- [BH07] Batmaz, F. und Hinde, C. J.: A Web-based Semi-Automatic Assessment Tool for Conceptual Database Diagram. In (Uskov, V., Hrsg.): Proc. 6th Int. Conf. on Web-Based Education (WBE), Chamonix, Frankreich. ACTA Press, S. 427–432, 2007.
- [BHS17] Breitner, J.; Hecker, M. und Snelting, G.: Der Grader Praktomat. In (Bott, O. J.; Fricke, P.; Priss, U. und Striewe, M., Hrsg.): Automatisierte Bewertung in der Programmierausbildung. 6, Waxmann, S. 159–172, 2017.
- [Bi93] Biggs, N.: Algebraic Graph Theory. 2. Aufl., Cambridge Univ. Press, 1993.
- [Bi96] Biggs, J. B.: Enhancing Teaching through Constructive Alignment. *Higher Education* 32/1, S. 347–364, 1996.
- [BKR00] Becker, J.; Kugeler, M. und Rosemann, M.: Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 1. Aufl., Springer, 2000.
- [BI05] Blank, G.; Parvez, S.; Wei, F. und Moritz, S.: A Web-based ITS for OO Design. In (Looi, C.-K.; McCalla, G. und Bredeweg, B., Hrsg.): Proc. Workshop on Adaptive Systems for Web-based Education at 12th Int. Conf. on Artificial Intelligence in Education (AIED), Amsterdam, Niederlande. IOS Press, S. 59–64, 2005.
- [BI73] Bloom, B. S.: Taxonomie von Lernzielen im kognitiven Bereich. 4. Aufl., Beltz, 1973.
- [BLB11] Bolloju, N.; Lee, K. W. K. und Banerjee, P. K.: Formative and Summative Assessment of Class Diagrams: Development and Evaluation of a Prototype. In (Cordeiro, J.; Verbraeck, A. und Helfert, M., Hrsg.): Proc. 3rd Int. Conf. on Computer Supported Education (CSEDU), Noordwijkerhout, Niederlande. SciTePress, S. 402–410, 2011.
- [BM04] Bull, J. und McKenna, C.: Blueprint for Computer-Assisted Assessment. RoutledgeFalmer, 2004.
- [BM05] Baghaei, N. und Mitrovic, A.: COLLECT-UML: Supporting Individual and Collaborative Learning of UML Class Diagrams in a Constraint-Based Intelligent Tutoring System. In (Khosla, R.; Howlett, R. J. und Jain, L. C., Hrsg.): Proc. 9th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES), Melbourne, Australien. Bd. 3684. LNCS, Springer, S. 458–464, 2005.
- [BM06] Baghaei, N. und Mitrovic, A.: A Constraint-Based Collaborative Environment for Learning UML Class Diagrams. In (Ikeda, M.; Ashley, K. D. und Chan, T.-W., Hrsg.): Proc. 8th Int. Conf. on Intelligent Tutoring Systems (ITS), Jhongli, Taiwan. Bd. 4053. LNCS, Springer, S. 176–186, 2006.

Literaturverzeichnis

- [BM07a] Baghaei, N. und Mitrovic, A.: Evaluating a Collaborative Constraint-Based Tutor for UML Class Diagrams. In (Luckin, R.; Koedinger, K. R. und Greer, J., Hrsg.): Proc. 13th Int. Conf. on Artificial Intelligence in Education (AIED), Amsterdam, Niederlande. IOS Press, S. 533–535, 2007.
- [BM07b] Baghaei, N. und Mitrovic, A.: From Modelling Domain Knowledge to Metacognitive Skills: Extending a Constraint-Based Tutoring System to Support Collaboration. In (Conati, C.; McCoy, K. und Paliouras, G., Hrsg.): Proc. 11th Int. User Modeling Conf. (UM), Korfu, Griechenland. Bd. 4511. LNCS, Springer, S. 217–227, 2007.
- [BMI05] Baghaei, N.; Mitrovic, A. und Irwin, W.: A Constraint-Based Tutor for Learning Object-Oriented Analysis and Design Using UML. In (Looi, C.-K.; Jonassen, D. und Ikeda, M., Hrsg.): Proc. Int. Conf. on Computers in Education (ICCE), Singapur. IOS Press, S. 9–16, 2005.
- [BMI06] Baghaei, N.; Mitrovic, A. und Irwin, W.: Problem-Solving Support in a Constraint-Based Tutor for UML Class Diagrams. *Technology, Instruction, Cognition and Learning* 4, 2006.
- [BMI07] Baghaei, N.; Mitrovic, A. und Irwin, W.: Supporting Collaborative Learning and Problem-Solving in a Constraint-Based CSCL Environment for UML Class Diagrams. *Int. Journal of Computer-Supported Collaborative Learning* 2/2-3, S. 159–190, 2007.
- [BMM20] Boubekeur, Y.; Mussbacher, G. und McIntosh, S.: Automatic Assessment of Students' Software Models Using a Simple Heuristic and Machine Learning. In (Syriani, E.; Sahraoui, H. A.; de Lara, J. und Abrahão, S., Hrsg.): Proc. 23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), Montreal, Kanada/Virtuell. ACM, S. 84–93, 2020.
- [Bö07] Börger, E.: Modeling Workflow Patterns from First Principles. In (Parent, C.; Schewe, K.; Storey, V. C. und Thalheim, B., Hrsg.): Proc. 26th Int. Conf. on Conceptual Modeling (ER), Auckland, Neuseeland. Bd. 4801. LNCS, Springer, 2007.
- [Bo19] Bork, D.: A Framework for Teaching Conceptual Modeling and Metamodeling Based on Bloom's Revised Taxonomy of Educational Objectives. In (Bui, T., Hrsg.): Proc. 52nd Hawaii Int. Conf. on System Sciences (HICSS), Maui, Hawaii, USA. ScholarSpace, 2019.
- [Bo21] de Boer, I.; de Vegt, F.; Pluk, H. und Latijnhouwers, M.: Rubrics – A Tool for Feedback and Assessment Viewed From Different Perspectives. *Enhancing Learning and Assessment Quality*. Springer, 2021.
- [BOU19] Binder, M.; Oberweis, A. und Ullrich, M.: Bewertung grafischer Prozessmodelle mit Methoden des Maschinellen Lernens: Ein Experiment. In: Proc. E-Prüfungs-Symposium (ePS), Siegen, Deutschland. S. 12–14, 2019.
- [BP06] Becker, J. und Pfeiffer, D.: Konzeptionelle Modelle in der Wirtschaftsinformatik – Konstruktion und Evaluation. *Das Wirtschaftsstudium* 35/2006, S. 1551–1557, 2006.
- [BPR72] Boari, M.; Pellizzardi, P. und Rossi, R.: Performance Evaluation of a Process Control System by Simulation Model. *Computer Journal* 15/3, S. 365–370, 1972.
- [BPV12] Becker, J.; Probandt, W. und Vering, O.: Grundsätze ordnungsmäßiger Modellierung. Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement. Springer, 2012.
- [Br04] Brinda, T.: Integration of New Exercise Classes into the Informatics Education in the Field of Object-Oriented Modelling. *Education and Information Technologies* 9/2, S. 117–130, 2004.

- [Br09] vom Brocke, J.; Simons, A.; Niehaves, B.; Riemer, K.; Plattfaut, R. und Cleven, A.: Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In (Rowe, F.; El Amrani, R.; Limayem, M.; Matook, S.; Rosenkranz, C.; Whitley, E. A. und El Quammah, A., Hrsg.): Proc. 17th European Conf. on Information Systems (ECIS), Verona, Italien. AIS eLibrary, S. 2206–2217, 2009.
- [Br15] vom Brocke, J.; Simons, A.; Riemer, K.; Niehaves, B.; Plattfaut, R. und Cleven, A.: Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. Communications of the Association for Information Systems 37, S. 205–224, 2015.
- [Br22] Brandt, A.; Siegburg, M.; Voigtländer, J. und Wang, K.: A Report on Automatic Generation of Petri Net Exercise and Exam Task Instances. In (Michael, J.; Pfeiffer, J. und Wortmann, A., Hrsg.): Proc. Workshops Modellierung 2022, 3. Workshop zur Modellierung in der Hochschullehre (MoHoL 2022), Hamburg, Deutschland. Gesellschaft für Informatik e.V., S. 197–204, 2022.
- [BRS95] Becker, J.; Rosemann, M. und Schütte, R.: Grundsätze ordnungsmäßiger Modellierung. Wirtschaftsinformatik 37/5, S. 435–445, 1995.
- [BRU00] Becker, J.; Rosemann, M. und von Uthmann, C.: Guidelines of Business Process Modeling. In (van der Aalst, W. M. P.; Desel, J. und Oberweis, A., Hrsg.): Business Process Management. Models, Techniques and Empirical Studies. Bd. 1806, LNCS, Springer, S. 30–49, 2000.
- [BS07] Brahm, T. und Seufert, S., E-Assessment und E-Portfolio zur Kompetenzentwicklung: neue Potenziale für Ne(x)t Generation Learning?, Swiss Centre for Innovations in Learning, 2007.
- [BS17] Bogdanova, D. und Snoeck, M.: Domain Modelling in Bloom: Deciphering How We Teach It. In (Poels, G.; Gailly, F.; Asensio, E. S. und Snoeck, M., Hrsg.): Proc. 10th IFIP Working Conf. on The Practice of Enterprise Modeling (POEM), Leuven, Belgien. Bd. 305, LNBIP, Springer, S. 3–17, 2017.
- [BS19] Bogdanova, D. und Snoeck, M.: CaMeLOT: An educational framework for conceptual data modelling. Information and Software Technology 110, S. 92–107, 2019.
- [BSH09] Batmaz, F.; Stone, R. und Hinde, C.: Personalised Feedback With Semi-Automatic Assessment Tool For Conceptual Database Model. Innovation in Teaching and Learning in Information and Computer Sciences 9/1, S. 105–109, 2009.
- [BSK09] Brandsteidl, M.; Seidl, M. und Kappel, G.: Teaching Models @BIG: On Efficiently Assessing Modeling Concepts. In (France, R. B. und Gogolla, M., Hrsg.): Proc. Educators’ Symposium at 12th ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), Denver, Colorado, USA. IEEE, 2009.
- [BST15] vom Brocke, J.; Seidel, S. und Tumbas, S.: Class Notes: The BPM Curriculum Revisited. BPTrends Column, 2015.
- [BT08] Börger, E. und Thalheim, B.: Modeling Workflows, Interaction Patterns, Web Services and Business Processes: The ASM-Based Approach. In (Börger, E.; Butler, M. J.; Bowen, J. P. und Boca, P., Hrsg.): Proc. 1st Int. Conf. Abstract State Machines, B and Z, London, UK. Bd. 5238, LNCS, Springer, S. 24–38, 2008.
- [BT11] Biggs, J. und Tang, B.: Teaching for Quality Learning at University: What the Student Does. 4. Aufl., Open University Press, 2011.

Literaturverzeichnis

- [BT95] Barr, R. B. und Tagg, J.: From Teaching to Learning — A New Paradigm for Undergraduate Education. *Change: The Magazine of Higher Learning* 27/6, S. 12–25, 1995.
- [Bu18] Bundesministerium für Bildung und Forschung: Bekanntmachung - Richtlinie zur Förderung von Zuwendungen für die Forschung zur digitalen Hochschulbildung – Disziplin- und fachbezogene digitale Hochschulbildung – Bundesanzeiger vom 09.11.2018, 2018, URL: https://www.bmbf.de/bmbf/shreddocs/bekanntmachungen/de/2018/11/2082_bekanntmachung, Stand: 13.06.2022.
- [Bu23] Bundesministerium für Bildung und Forschung: Wissenschaft und Hochschulforschung: Digitale Hochschulbildung, Webseite, 2023, URL: <https://www.bmbf.de/bmbf/de/bildung/bildungsforschung/wissenschafts-und-hochschulforschung/digitale-hochschulbildung.html>, Stand: 23.12.2023.
- [BV20] Boulanger, D. und Vivekanadan, K.: SHAPed Automated Essay Scoring: Explaining Writing Features' Contributions to English Writing Organization. In (Vivekanandan, K. und Troussas, C., Hrsg.): *Proc. 16th Int. Conf. on Intelligent Tutoring Systems (ITS)*, Athen, Griechenland. Bd. 12149. LNCS, Springer, S. 68–78, 2020.
- [Ca06] Cardoso, J.; Mendling, J.; Neumann, G. und Reijers, H. A.: A Discourse On Complexity of Process Models. In (Eder, J. und Dustdar, S., Hrsg.): *Proc. Business Process Management (BPM) Workshops, Workshop on Business Process Design (BPI)*, Wien, Österreich. Bd. 4103. LNCS, Springer, S. 117–128, 2006.
- [Ca13a] Caporale, T.; Çitak, M.; Lehner, J.; Schoknecht, A.; Oberweis, A. und Ullrich, M.: Motivating Course Concept: Using Active Labs for BPM Education. In (Helms, R. und Brinkkemper, S., Hrsg.): *Proc. 21st European Conf. on Information Systems (ECIS)*, Utrecht, Niederlande. AIS eLibrary, S. 12, 2013.
- [Ca13b] Caporale, T.; Çitak, M.; Lehner, J.; Oberweis, A.; Schoknecht, A. und Ullrich, M.: Social BPM Lab – Characterization of a Collaborative Approach for Business Process Management Education. In (Huemer, C., Hrsg.): *Proc 15. IEEE Conf. on Business Informatics (CBI)*, Wien, Österreich. IEEE, S. 367–373, 2013.
- [Ca13c] Cayoglu, U.; Oberweis, A.; Schoknecht, A. und Ullrich, M.: Triple-S: A Matching Approach for Petri Nets on Syntactic, Semantic and Structural Level, 2013, URL: <https://semreuse.aifb.kit.edu/downloads/TripleS.pdf>, Stand: 15.06.2022.
- [Ca14] Cayoglu, U.; Dijkman, R.; Dumas, M.; Fettke, P.; García-Bañuelos, L.; Hake, P.; Klinkmüller, C.; Leopold, H.; Ludwig, A.; Loos, P.; Mendling, J.; Oberweis, A.; Schoknecht, A.; Sheetrit, E.; Thaler, T.; Ullrich, M.; Weber, I. und Weidlich, M.: Report: The Process Model Matching Contest 2013. In (Lohmann, N.; Song, M. und Wohed, P., Hrsg.): *Proc. 11th Int. Conf. on Business Process Management (BPM), Workshops*, Peking, China. Bd. 171. LNBP, Springer, S. 442–463, 2014.
- [CCG87] Collins, M.; Carnine, D. und Gersten, R.: Elaborative Corrective Feedback and the Acquisition of Reasoning Skills: A Study of Computer-Assisted Instruction. *Exceptional Children* 54/1, S. 254–262, 1987.
- [Ch76] Chen, P.P.: The Entity-Relationship Model – Toward a Unified View of Data. *ACM Transactions on Database Systems* 1/1, S. 9–36, 1976.
- [CK91] Chen, P.P. und Knöll, H.-D.: *Der Entity-Relationship-Ansatz zum logischen Systementwurf*. BI Wissenschaftsverlag, 1991.

- [CLM22] Christ, P.; Laue, R. und Munkelt, T.: ALADIN - Generator für Aufgaben und Lösung(shilf)en aus der Informatik und angrenzenden Disziplinen. In (Michael, J.; Pfeiffer, J. und Wortmann, A., Hrsg.): Proc. Modellierung 2022 - Workshopband, 3. Workshop zur Modellierung in der Hochschullehre (MoHoL 2022), Hamburg, Deutschland. Gesellschaft für Informatik e.V., S. 205–212, 2022.
- [CLP17] Correia, H.; Leal, J. P. und Paiva, J. C.: Enhancing Feedback to Students in Automated Diagram Assessment. In (Queirós, R.; Pinto, M.; Simões, A.; Leal, J. P. und Pereira, M. J. V., Hrsg.): Proc. 6th Symposium on Languages, Applications and Technologies (SLATE), Vila do Conde, Portugal. Dagstuhl Publishing, 2017.
- [CLP18] Correia, H.; Leal, J. P. und Paiva, J. C.: Improving Diagram Assessment in Mooshak. In (Ras, E. und Guerrero Roldán, A. E., Hrsg.): Proc. 20th Int. Conf. on Technology Enhanced Assessment (TEA), Barcelona, Spanien. Bd. 829. CCIS, Springer, S. 69–82, 2018.
- [CM07] Coelho, W. und Murphy, G.: ClassCompass: A software design mentoring system. *Journal on Educational Resources in Computing* 7/1, 2007.
- [CSS00] Constantino-González, M.; Suthers, D. D. und los Santos, J. G. E.: A Coached Collaborative Learning Environment for Entity-Relationship Modeling. In (Gauthier, G.; Frasson, C. und VanLehn, K., Hrsg.): Proc. 5th Int. Conf. on Intelligent Tutoring Systems (ITS), Montreal, Canada. Bd. 1839. LNCS, Springer, S. 324–333, 2000.
- [CSS03] Constantino-González, M.; Suthers, D. D. und los Santos, J. G. E.: Coaching Web-based Collaborative Learning based on Problem Solution Differences and Participation. *Int. Journal of Artificial Intelligence in Education* 13/2-4, S. 263–299, 2003.
- [CW05] Conole, G. und Warburton, W.: A Review of Computer-Assisted Assessment. *Research in Learning Technology (ALT-J)* 13/1, S. 17–31, 2005.
- [DA05] D’Antoni, L.; Kini, D.; Alur, R.; Gulwani, S.; Viswanathan, M. und Hartmann, B.: How Can Automatic Feedback Help Students Construct Automata? *ACM Transactions on Computer-Human Interaction* 22/2, 2005.
- [Da06] Davies, I.; Green, P.; Rosemann, M.; Indulska, M. und Gallo, S.: How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering* 58/3, S. 358–380, 2006.
- [Da18] Daehli, O.; Kristoffersen, B.; Lauvås P., J. und Myrbacken, H.: A Supportive Web-Based Tool for Learning Basic Data Modeling Skills. In (Ntalianis, K.; Sgouropoulou, C. und Andreas, A., Hrsg.): Proc. 17th European Conf. on e-Learning (ECEL), Athen, Griechenland. IEEE, S. 116–124, 2018.
- [DA20] D’Antoni, L.; Helfrich, M.; Kretinsky, J.; Ramneantu, E. und Weininger, M.: Automata Tutor v3. In (Lahiri, S. K. und Wang, C., Hrsg.): Proc. 32nd Int. Conf. on Computer Aided Verification (CAV), Los Angeles, USA. Bd. 12224. LNCS, Springer, S. 3–14, 2020.
- [DAH05] Dumas, M.; van der Aalst, W. M. P. und ter Hofstede, A. H. M.: *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley, 2005.
- [DDM08] van Dongen, B.; Dijkman, R. und Mendling, J.: Measuring Similarity between Business Process Models. In (Bellahsene, Z. und Léonard, M., Hrsg.): Proc. 20th Int. Conf. on Advanced Information Systems Engineering (CAiSE), Montpellier, Frankreich. Bd. 5074. LNCS, Springer, S. 450–464, 2008.
- [Di11] Dijkman, R.; Dumas, M.; Van Dongen, B.; Krik, R. und Mendling, J.: Similarity of business process models: Metrics and evaluation. *Information Systems* 36/2, S. 498–516, 2011.

Literaturverzeichnis

- [DK16] Darshan, K. M. und Kumar, V.: Hybrid ITS for DFA Construction Problems. In (Yang, J.; Chen, W.; Murthy, S.; Wong, S. L. und Iyer, S., Hrsg.): Proc. 24th Int. Conf. on Computers in Education (ICCE), Bombay, Indien. IEEE, S. 96–98, 2016.
- [DKO17] Drescher, A.; Koschmider, A. und Oberweis, A.: Modellierung und Analyse von Geschäftsprozessen: Grundlagen und Übungsaufgaben mit Lösungen. De Gruyter Studium, 2017.
- [DTD18] Dikici, A.; Turetken, O. und Demirors, O.: Factors influencing the understandability of process models: A systematic literature review. *Information and Software Technology* 93/October, S. 112–129, 2018.
- [Du13] Dumas, M.; La Rosa, M.; Mendling, J. und Reijers, H. A.: *Fundamentals of Business Process Management*. Springer, 2013.
- [DW09] Demuth, B. und Weigel, D.: Web Based Software Modeling Exercises in Large-Scale Software Engineering Courses. In (Varma, V.; Armarego, J. und Jalote, P., Hrsg.): Proc. 22nd Conf. on Software Engineering Education and Training (CSEET), Hyderabad, Indien. IEEE, S. 138–141, 2009.
- [Ec17] Echeverria, V.; Falcones, G.; Castells, J.; Martinez-Maldonado, R. und Chiluiza, K.: Exploring On-time Automated Assessment in a Co-located Collaborative System. In (Meier, A. und Terán, L., Hrsg.): Proc. 4th Int. Conf. on eDemocracy and eGovernment (ICEDEG), Quito, Ecuador. IEEE, S. 273–276, 2017.
- [EJS10] Effinger, P.; Jogsch, N. und Seiz, S.: On a Study of Layout Aesthetics for Business Process Models Using BPMN. In (Mendling, J.; Weidlich, M. und Weske, M., Hrsg.): Proc. 2nd. Int. Workshop on Business Process Modeling Notation (BPMN), Potsdam, Deutschland. Bd. 67. LNBI, Springer, S. 31–45, 2010.
- [EKO07] Ehrig, M.; Koschmider, A. und Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In (Roddick, J. F. und Hinze, A., Hrsg.): Proc. 4th Asia-Pacific Conf. on Conceptual Modelling (APCCM), Ballarat, Australien. Australian Computer Society, 2007.
- [El15] Elmadani, M.; Mitrovic, A.; Weerasinghe, A. und Neshatian, K.: Investigating student interactions with tutorial dialogues in EER-Tutor. *Research and Practice in Technology Enhanced Learning* 10/1, 2015.
- [Eu03] Europäische Bildungsminister: Den Europäischen Hochschulraum verwirklichen. Communiqué der Konferenz der europäischen Hochschulministerinnen und -minister, Berlin, Deutschland, 2003.
- [Fa20] Fauzan, R.; Siahaan, D.; Rochimah, S. und Triandini, E.: A Different Approach on Automated Use Case Diagram Semantic Assessment. *International Journal of Intelligent Engineering and Systems* 14/1, S. 496–505, 2020.
- [Fa98] Falkenberg, E. D.; Hesse, W.; Lindgreen, P.; Nilsson, B. E.; Oei, J. L. H.; Rolland, C.; Stamper, R. K.; van Assche, F. J. M.; Verrijn-Stuart, A. A. und Voss, K., *A Framework of Information Systems Concepts*, International Federation for Information Processing, 1998.
- [Fe09] Fettke, P.: How Conceptual Modeling Is Used. *Communications of the Association for Information Systems* 25/1, S. 571–592, 2009.

- [Fe13] Fellmann, M.; Bittmann, S.; Karhof, A.; Stolze, C. und Thomas, O.: Do We Need a Standard for EPC Modelling? The state of Syntactic, Semantic and Pragmatic Quality. In (Jung, R. und Reichert, M., Hrsg.): Proc. 5th Enterprise Modelling and Information Systems Architectures (EMISA), St. Gallen, Schweiz. Bd. P-222. LNI, Gesellschaft für Informatik e.V., S. 103–116, 2013.
- [Fe16] Fellmann, M.; Fettke, P.; Houy, C.; Loos, P.; Oberweis, A.; Schoknecht, A.; Striewe, M.; Thaler, T. und Ullrich, M.: Evaluation automatisierter Ansätze für die Bewertung von Modellierungsaufgaben. In (Lucke, U.; Schwill, A. und Zender, R., Hrsg.): Proc. 14. e-Learning Fachtagung Informatik (DeLFI), Potsdam, Deutschland. Bd. P-262. LNI, Gesellschaft für Informatik e.V., S. 203–214, 2016.
- [FL03] Frank, U. und van Laak, B.L., Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Universität Koblenz-Landau, 2003.
- [FP97] Frank, U. und Prasse, M., Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen - veranschaulicht am Beispiel von OML und UML, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Universität Koblenz-Landau, 1997.
- [Fr09] Friedrich, F.: Measuring Semantic Label Quality Using WordNet. In (Nüttgens, M.; Rump, F.J.; Mendling, J. und Gehrke, N., Hrsg.): Proc. 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK), Berlin, Deutschland. Bd. 554, CEUR-WS, S. 118–137, 2009.
- [Fr14] Frank, U.; Strecker, S.; Fettke, P.; vom Brocke, J.; Becker, J. und Sinz, E.: Das Forschungsfeld »Modellierung betrieblicher Informationssysteme«. *Wirtschaftsinformatik* 56/1, S. 49–54, 2014.
- [Fr94] Frank, U.: Multiperspektivistische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Oldenbourg, 1994.
- [FS13] Ferstl, O. und Sinz, E.: Grundlagen der Wirtschaftsinformatik. 7. Aufl., Gabler, 2013.
- [FS15] Figl, K. und Strembeck, M.: Findings from an Experiment on Flow Direction of Business Process Models. In (Kolb, J.; Leopold, H. und Mendling, J., Hrsg.): Proc. 7th Enterprise Modelling and Information Systems Architectures (EMISA), Innsbruck, Österreich. Bd. P-248. LNI, Gesellschaft für Informatik e.V., S. 59–73, 2015.
- [FSU16] Fellmann, M.; Schoknecht, A. und Ullrich, M.: Vorwort zum Workshop zur Modellierung in der Hochschullehre (MoHoL 2016). In (Betz, S. und Reimer, U., Hrsg.): Proc. Modellierung 2016, Karlsruhe, Deutschland - Workshopband. Bd. P-255. LNI, Gesellschaft für Informatik e.V., S. 45–46, 2016.
- [FW06] Frederiks, P.J. und van der Weide, T.P.: Information modeling: The process and the required competencies of its participants. *Data and Knowledge Engineering* 58/1, S. 4–20, 2006.
- [Ga17a] Gallagher, G.: Aligning for Learning: Including Feedback in the Constructive Alignment Model. *All Ireland Journal of Teaching and Learning in Higher Education (AISHE-J)* 9/1, S. 3011–30112, 2017.
- [Ga17b] Garmann, R.; Fricke, P.; Reiser, P.; Bersuch, C. und Bott, O.J.: Moodle, Grappa, aSQLg, Graja - Neue Entwicklungen bei der Grading-Software der Hochschule Hannover. In (Strickroth, S.; Müller, O. und Striewe, M., Hrsg.): Proc. 3rd Workshop Automatische Bewertung von Programmieraufgaben (ABP), Potsdam, Deutschland. Bd. 2015, CEUR-WS, 2017.

Literaturverzeichnis

- [Ga19a] Garmann, R.: Ein Format für Bewertungsvorschriften in automatisiert bewertbaren Programmieraufgaben. In (Pinkwart, N. und Konert, J., Hrsg.): Proc. 17. e-Learning Fachtagung Informatik (DeLFI), Berlin, Deutschland. Bd. P-297. LNI, Gesellschaft für Informatik e.V., S. 103–114, 2019.
- [Ga19b] Garmann, R., Ein Format für Bewertungsvorschriften in automatisiert bewertbaren Programmieraufgaben, Hochschule Hannover, Fakultät IV - Wirtschaft und Informatik, 2019.
- [Ga20] Gadatsch, A.: Grundkurs Geschäftsprozess-Management. 9. Aufl., Springer, 2020.
- [Ga83] Gaitanides, M.: Prozessorganisation. Entwicklung, Ansätze und Programme prozessorientierter Organisationsgestaltung. Vahlen, 1983.
- [Ge16a] Gesellschaft für Informatik e.V. (GI), Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen, 2016.
- [Ge16b] Gesellschaft für Informatik e.V. (GI), Rahmenempfehlung für die Ausbildung in Wirtschaftsinformatik an Hochschulen, 2016.
- [GHW15] Garmann, R.; Heine, F. und Werner, P.: Grappa - die Spinne im Netz der Autobewerter und Lernmanagementsysteme. In (Pongratz, H. und Keil, R., Hrsg.): Proc. 13. e-Learning Fachtagung Informatik (DeLFI), München, Deutschland. Bd. P-247. LNI, Gesellschaft für Informatik e.V., S. 169–181, 2015.
- [GL06a] Gruhn, V. und Laue, R.: Adopting the Cognitive Complexity Measure for Business Process Models. In (Zhang, D.; Wang, Y. und Kinsner, W., Hrsg.): Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI), Peking, China. IEEE, S. 116–119, 2006.
- [GL06b] Gruhn, V. und Laue, R.: Complexity Metrics for Business Process Models. In (Abramowicz, W. und Mayr, H. C., Hrsg.): 9th Int. Conf. on Business Information Systems (BIS), Klagenfurt, Österreich. Bd. P-085. LNI, Gesellschaft für Informatik e.V., S. 1–12, 2006.
- [GL06c] Gruhn, V. und Laue, R.: Validierung syntaktischer und anderer EPK-Eigenschaften mit PROLOG. In (Nüttgens, M.; Rump, F. J. und Mendling, J., Hrsg.): Proc. 5. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK), Wien, Österreich. Bd. 224, CEUR-WS, S. 69–84, 2006.
- [GI08] Glinz, M.: Modellierung in der Lehre an Hochschulen: Thesen und Erfahrungen. Informatik Spektrum 31/5, S. 425–434, 2008.
- [GL09] Gruhn, V. und Laue, R.: Ein einfaches Verfahren zur Erkennung häufiger Fehler in EPKs. In (Nüttgens, M.; Rump, F. J.; Mendling, J. und Gehrke, N., Hrsg.): Proc. 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK), Berlin, Deutschland. Bd. 554, CEUR-WS, S. 58–74, 2009.
- [GL11] Gruhn, V. und Laue, R.: Detecting Common Errors in Event-Driven Process Chains by Label Analysis. Enterprise Modeling and Information Systems Architecture (EMISA), Int. Journal of Conceptual Modeling 6/1, S. 3–15, 2011.
- [GL79] Genrich, H. J. und Lautenbach, K.: The Analysis of Distributed Systems by Means of Predicate/Transition-Nets. In (Kahn, G., Hrsg.): Proc. Int. Symposium on Semantics of Concurrent Computation, Evian, Frankreich. Bd. 70. LNCS, Springer, S. 123–146, 1979.

- [Gr09] Gruttmann, S. J.: Formatives E-Assessment in der Hochschullehre – Computerunterstützte Lernfortschrittskontrollen im Informatikstudium, Diss., Universität Münster, Deutschland, 2009.
- [Gu11] Guo, X.; Wu, C.; Chen, Y. und Xie, M.: Interactive Items-Based Online Examination System. In (He, Y., Hrsg.): Proc. Int. Conf. on Internet Technology and Applications (iTAP), Wuhan, China. IEEE, S. 1–4, 2011.
- [Ha11] Hasker, R. W.: UMLGrader: An Automated Class Diagram Grader. *Journal of Computing Sciences in Colleges* 27/1, S. 47–54, 2011.
- [Ha16] Hafer, J.; Matthé, F.; Schumann, W.; Wollersheim, H.-W.; Jeremias, C.; Grigat, F. und Schulz, A.: Schwerpunktthema: E-Klausuren. *Forschung & Lehre* 23/3, S. 195–209, 2016.
- [Ha87] Halstead, M. H.: *Elements of Software Science*. Elsevier, 1987.
- [Ha91] Harrington, H. J.: *Business Process Improvement. The Breakthrough Strategy for Total Quality, Productivity and Competitiveness*. McGraw-Hill, 1991.
- [HB06] Higgins, C. A. und Bligh, B.: Formative Computer Based Assessment in Diagram Based Domains. In (Davoli, R.; Goldweber, M. und Salomoni, P., Hrsg.): Proc. 11th Annual SIGCSE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Bologna, Italien. ACM, S. 98–102, 2006.
- [HC93] Hammer, M. und Champy, J.: *Reengineering the Corporation: A Manifesto for Business Revolution*. *Business Horizons* 36/5, S. 90–91, 1993.
- [HF97] Hamp, B. und Feldweg, H.: GermaNet - a Lexical-Semantic Net for German. In (Cohen, P. R. und Wahlster, W., Hrsg.): Proc. 35th Annual Meeting of the Association of Computational Linguistics (ACL). Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. Madrid, Spanien. ACM, 1997.
- [HFL12] Houy, C.; Fettke, P. und Loos, P.: Understanding Understandability of Conceptual Models – What Are We Actually Talking about? In (Atzeni, P.; Cheung, D. W. und Ram, S., Hrsg.): Proc. 31st Int. Conf. on Conceptual Modeling (ER), Florenz, Italien. Bd. 7532. LNCS, Springer, S. 64–77, 2012.
- [HFL14] Houy, C.; Fettke, P. und Loos, P.: On the Theoretical Foundations of Research into the Understandability of Business Process Models. In (Avital, M.; Leimeister, J. M. und Schultze, U., Hrsg.): Proc. of 22nd European Conference on Information Systems (ECIS), Tel Aviv, Israel. AIS eLibrary, 2014.
- [HG98] Hall, L. und Gordon, A.: A Virtual Learning Environment for Entity Relationship Modelling. In (D. Joyce, J. I., Hrsg.): Proc. 29th Technical Symposium on Computer Science Education (SIGCSE), Atlanta, USA. ACM, S. 345–349, 1998.
- [HL98] Hoggarth, G. und Lockyer, M.: An Automated Student Diagram Assessment System. In (Davies, G. und ÓHigearthaigh, M., Hrsg.): Proc. 3rd Annual SIGCSE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Dublin, Irland. ACM, S. 122–124, 1998.
- [HM08] Hesse, W. und Mayr, H. C.: Modellierung in der Softwaretechnik: eine Bestandsaufnahme. *Informatik Spektrum* 31/5, S. 377–393, 2008.
- [Ho11] Holland, J.; Baghaei, N.; Mathews, M. und Mitrovic, A.: The Effects of Domain and Collaboration Feedback on Learning in a Collaborative Intelligent Tutoring System. In (Biswas, G.; Bull, S.; Kay, J. und Mitrovic, A., Hrsg.): Proc. 15th Int. Conf. on Artificial Intelligence in Education (AIED), Auckland, Neuseeland. Bd. 6738. LNCS, Springer, S. 469–471, 2011.

Literaturverzeichnis

- [Ho95] Hollingsworth, D., Workflow Management Coalition: The Workflow Reference Model, Workflow Management Coalition, 1995.
- [HR00] Harel, D. und Rumpe, B., Modeling Languages : Syntax , Semantics and All That Stuff. Part I: The Basic Stuff, Weizmann Institute, Rehovot, Israel, 2000.
- [HR04] Harel, D. und Rumpe, B.: Meaningful Modeling: What's the Semantics of »Semantics«? Computer 37/10, S. 64–72, 2004.
- [HR11] Hasker, R. W. und Rowe, M.: UMLint: Identifying Defects in UML Diagrams. In: Proc. ASEE Annual Conf. & Exposition, Vancouver, Kanada. ASEE PEER, S. 22.1558.1–22.1558.14, 2011.
- [HS14] Hasker, R. W. und Shi, Y.: Teaching Basic Class Diagram Notation with UMLGrader. In: Proc. ASEE Annual Conf. & Exposition, Indianapolis, USA. ASEE PEER, S. 24.1157.1–24.1157.11, 2014.
- [HST02] Higgins, C.; Symeonidis, P. und Tsintsifas, A.: The Marking System for CourseMaster. In (Caspersen, M. E.; Joyce, D.; Goelman, D. und Utting, I., Hrsg.): Proc. 7th Annual SIGSCE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Aarhus, Dänemark. ACM, S. 46–50, 2002.
- [HT07] Hattie, J. und Timperley, H.: The Power of Feedback. Review of Educational Research 77/1, S. 81–112, 2007.
- [Ic19] Ichinohe, Y.; Hashiura, H.; Tanaka, T.; Hazeyama, A. und Takase, H.: Effectiveness of Automated Grading Tool Utilizing Similarity for Conceptual Modeling. In (Virvou, M.; Kumeno, F. und Oikonomou, K., Hrsg.): Proc. 12th Joint Conf. on Knowledge-Based Software Engineering (JCKBSE), Korfu, Griechenland. IOS Press, S. 117–126, 2019.
- [If12] Ifland, M.; Hörnlein, A.; Ott, J. und Puppe, F.: Geführtes Tutorsystem mit Unterstützung zunehmender Selbstständigkeit zur Modellierung von UML-Klassendiagrammen. In (Desel, J.; Haake, J. M. und Spannagel, C., Hrsg.): Proc. 10. e-Learning Fachtagung Informatik (DeLFI), Hagen, Deutschland. Bd. P-207. LNI, Gesellschaft für Informatik e.V., S. 75–86, 2012.
- [IW09] Iller, C. und Wick, A.: Das Hochschulwesen. Forum für Hochschulforschung, -praxis und -politik 57/6, S. 195–201, 2009.
- [Ja18] Jayawardena, R. R. A. M. P.; Thiwanthi, G. A. D.; Suriyaarachchi, P. S.; Withana, K. I. und Jayawardena, C.: Automated Exam Paper Marking System for Structured Questions and Block Diagrams. In (Thiruvaran, T., Hrsg.): Proc. 9th IEEE Int. Conf. on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka. IEEE, 2018.
- [Je81] Jensen, K.: Coloured Petri Nets and the Invariant-Method. Theoretical Computer Science 14, S. 317–336, 1981.
- [JM02] Jank, W. und Meyer, H.: Didaktische Modelle. 13. Aufl., Cornelsen, 2002.
- [JS07] Jonsson, A. und Svingby, G.: The Use of Scoring Rubrics: Reliability, Validity and Educational Consequences. Educational Research Review, 2007.
- [JS09] Jayal, A. und Shepperd, M.: An Improved Method for Label Matching in E-Assessment of Diagrams. Innovation in Teaching and Learning in Information and Computer Sciences (ITALICS) 8/1, S. 3–16, 2009.
- [KC07] Kitchenham, B. und Charters, S.: Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report No. EBSE-2007-01, Keele University, 2007.

- [Ki06] Kindler, E.: On the semantics of EPCs: Resolving the vicious circle. *Data and Knowledge Engineering* 56/1, S. 23–40, 2006.
- [KJ02] Krogstie, J. und Jørgensen, H. D.: Quality of Interactive Models. In (Olivé, A.; Yoshikawa, M. und Yu, E. S. K., Hrsg.): *Proc. Advanced Conceptual Modeling Techniques, Workshops at 23st Int. Conf. on Conceptual Modeling (ER)*, Tampere, Finnland. IFIPAICT, Springer, S. 351–363, 2002.
- [KK02] Karagiannis, D. und Kühn, H.: Metamodelling Platforms. In (Bauknecht, K.; Min Tjoa, A. und Quirchmayer, G., Hrsg.): *Proc. 3rd Int. Conf. EC-Web*, Aix en Provence, Frankreich. Bd. 2455. LNCS, Springer, S. 182–196, 2002.
- [Kl64] Klafki, W.: Didaktische Analyse als Kern der Unterrichtsvorbereitung. In: *Grundlegende Aufsätze aus der Zeitschrift "Die Deutsche Schule"*. Herrmann Schroedel Verlag KG, S. 5–34, 1964.
- [KI70] Klafki, W.: *Funk-Kolleg Erziehungswissenschaften*, Band 1. Fischer Verlag, Frankfurt, 1970.
- [KLC99] Keller, G.; Lietschulte, A. und Curran, T. A.: Business Engineering mit den R/3-Referenzmodellen. In (Scheer, A. W. und Nüttgens, M., Hrsg.): *Proc. 4th Int. Conf on Electronic Business Engineering (Internationale Tagung Wirtschaftsinformatik)*, Saarbrücken, Deutschland. Physica-Verlag, S. 22, 1999.
- [KLS95] Krogstie, J.; Lindland, O. I. und Sindre, G.: Defining quality aspects for conceptual models. In (Falkenberg, E. D.; Hesse, W. und Olivé, A., Hrsg.): *Proc. IFIP Int. Working Conf. on Information Systems Concepts (ISCO)*, Marburg, Deutschland. IFIPAICT, Springer, S. 216–231, 1995.
- [KNS92] Keller, G.; Nüttgens, M. und Scheer, A. W., *Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK)*, Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi). Nr. 89. Universität des Saarlandes, 1992.
- [Kr12] Krogstie, J.: *Quality of Models. Model-based Development and Evolution of Information Systems*. Springer, 2012.
- [Kr16] Krogstie, J.: *Quality in Business Process Modeling*. Springer, 2016.
- [KSJ06] Krogstie, J.; Sindre, G. und Jørgensen, H.: Process models representing knowledge for action: A revised quality framework. *European Journal of Information Systems* 15/1, S. 91–102, 2006.
- [La01] Latva-Koivisto, A. M., *Finding a Complexity Measure for Business Process Models*, Helsinki University of Technology, Systems Analysis Laboratory, Helsinki, Finnland, 2001.
- [La05] Lausen, G.: *Datenbanken. Grundlagen und XML-Technologien*. Elsevier, Spektrum Akadem. Verl., 2005.
- [LAS21] Lübke, D.; Ahrens, M. und Schneider, K.: Influence of diagram layout and scrolling on understandability of BPMN processes: an eye tracking experiment with BPMN diagrams. *Information Technology and Management* 22, S. 99–131, 2021.
- [Le06] Le, N.-T.: A Constraint-based Assessment Approach for Free- Form Design of Class Diagrams using UML. In (Lynch, C.; Ashley, K.; Mitrovic, T.; Dimitrova, V.; Pinkwart, N. und Aleven, V., Hrsg.): *Proc. 4th Int. Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at 10th Int. Conf. on Intelligent Tutoring Systems (ITS)*, Pittsburgh, USA. S. 11–19, 2006.
- [Le13] Leopold, H.; Eid-Sabbagh, R.-H.; Mendling, J.; Azevedo, L. G. und Baião, F.: Detection of naming convention violations in process models for different languages. *Decision Support Systems* 56, S. 310–325, 2013.

Literaturverzeichnis

- [Le19] Leopold, H.; van der Aa, H.; Offenber, J. und Reijers, H. A.: Using Hidden Markov Models for the accurate linguistic analysis of process model activity labels. *Information Systems* 83, S. 30–39, 2019.
- [Le66] Levenshtein, I.: Binary code capable of correcting deletions, insertions and reversals. *Cybernetics and Control Theory* 10/8, S. 707–710, 1966.
- [Li19] Lino, A.; Rocha, Á.; Macedo, L. und Sizo, A.: LabDER - Relational Database Virtual Learning Environment. In (Panach, J. I.; Guizzardi, R. und Barreiro Claro, D., Hrsg.): *Proc. Joint ER Forum and Poster and Demos Session at 38th Int. Conf. on Conceptual Modeling (ER)*, Salvador, Brasilien. Bd. 2469, CEUR-WS, S. 42–55, 2019.
- [LS87] Larkin, J. H. und Simon, H. A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science* 11/1, S. 65–100, 1987.
- [LSM12] Leopold, H.; Smirnov, S. und Mendling, J.: On the refactoring of activity labels in business process models. *Information Systems* 37/5, S. 443–459, 2012.
- [LSS94] Lindland, O. I.; Sindre, G. und Sølberg, A.: Understanding Quality in Conceptual Modeling. *IEEE Software* 11/2, S. 42–49, 1994.
- [Ma19] Mayring, P.: *Handbuch Qualitative Forschung in der Psychologie*. In (Mey, G. und Mruck, K., Hrsg.). Springer, Kap. Qualitative Inhaltsanalyse, S. 1–17, 2019.
- [Ma72] Mager, R. J.: *Lernziele und programmierter Unterricht*. Beltz, 1972.
- [MB21] Morante, R. und Blanco, E.: Recent advances in processing negation. *Natural Language Engineering* 27/2, S. 121–130, 2021.
- [Mc02] McAlpine, M., *Principles of Assessment*, Robert Clark Centre for Technological Education, University of Glasgow, Schottland, 2002.
- [Mc76] McCabe, T. J.: A Complexity Measure. *IEEE Transactions on Software Engineering* 2/4, S. 308–320, 1976.
- [Me08a] Mendling, J.: *Metrics for Process Models*. Springer, 2008.
- [Me08b] Mendling, J.; Verbeek, H. M. W.; van Dongen, B. F.; van der Aalst, W. M. P. und Neumann, G.: Detection and prediction of errors in EPCs of the SAP reference model. *Data and Knowledge Engineering* 64/1, S. 312–329, 2008.
- [Me12] Mendling, J.; Sánchez-González, L.; García, F. und La Rosa, M.: Thresholds for error probability measures of business process models. *Journal of Systems and Software* 85/5, S. 1188–1197, 2012.
- [Mi01] Mitrovic, A.; Mayo, M.; Suraweera, P. und Martin, B.: Constraint-Based Tutors: A Success Story. In: *Engineering of Intelligent Systems*. Springer, S. 931–940, 2001.
- [ML05] Meiselwitz, G. und Lu, C.: Questionnaire for Evaluation of Usability and Learning Outcomes in Online Instruction. In (Remenyi, D., Hrsg.): *Proc. 4th European Conf. on e-Learning (ECEL)*, IEEE, S. 253–260, 2005.
- [MM21] McCallum, S. und Milner, M.: The effectiveness of formative assessment: student views and staff reflections. *Assessment & Evaluation in Higher Education* 46/1, S. 1–16, 2021.
- [MN04] Metzger, C. und Nüesch, C.: Fair prüfen: Ein Qualitätsleitaden für Prüfende an Hochschulen. In: *Hochschuldidaktische Schriften*. Bd. 6, Institut für Wirtschaftspädagogik der Universität St. Gallen, 2004.

- [MNA07] Mendling, J.; Neumann, G. und van der Aalst, W. M. P.: Understanding the Occurrence of Errors in Process Models Based on Metrics. In (Meersman, R.; Tari, Z. und Herrero, P., Hrsg.): Proc. On the Move to Meaningful Internet Systems (OTM), Vilamoura, Portugal. Bd. 4805. LNCS, Springer, S. 113–130, 2007.
- [Mo02] Moody, D. L.; Sindre, G.; Brasethvik, T. und Sølvberg, A.: Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework. In (Spaccapietra, S.; March, S. T. und Kambayashi, Y., Hrsg.): Proc. 21st Int. Conf. on Conceptual Modeling (ER), Tampere, Finnland. Bd. 2503. LNCS, Springer, S. 380–396, 2002.
- [Mo03] Moody, D. L.; Sindre, G.; Brasethvik, T. und Sølvberg, A.: Evaluating the Quality of Information Models: Empirical Testing of a Conceptual Model Quality Framework. In (Clarke, L.; Dillon, L. und Tichy, W., Hrsg.): Proc. 25th Int. Conf on Software Engineering, Portland, USA. IEEE, S. 295–307, 2003.
- [Mo05a] Moody, D. L.: Theoretical and practical issues in evaluating the quality of conceptual models: current State and future directions. *Data and Knowledge Engineering* 55/3, S. 243–276, 2005.
- [Mo05b] Moritz, S. H.; Wei, F.; Parvez, S. M. und Blank, G. D.: From Objects-First to Design-First with Multimedia and Intelligent Tutoring. In (Cunha, J. C.; Fleischman, W. M.; Proulx, V. K. und Lourenço, J. M., Hrsg.): Proc. 10th Annual SIGSCE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Caparica, Portugal. ACM, S. 99–103, 2005.
- [Mo09] Moody, D. L.: The »Physics« of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering* 35/6, S. 756–779, 2009.
- [Mo15] Moreno-Montes De Oca, I.; Snoeck, M.; Reijers, H. A. und Rodríguez-Morffi, A.: A systematic literature review of studies on business process modeling quality. *Information and Software Technology* 58/1, S. 187–205, 2015.
- [Mo67] Morris, W. T.: On the Art of Modeling. *Management Science* 13/12, S. 707–717, 1967.
- [Mo82] Molloy, M. K.: Performance Analysis Using Stochastic Petri Nets. *IEEE Transactions on Computers* 9/31, S. 913–917, 1982.
- [Mo88] Morris, C. W.: *Grundlagen der Zeichentheorie*. Fischer Verlag, 1988.
- [MR08] zur Muehlen, M. und Recker, J.: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In (Bellahsene, Z. und Léonard, M., Hrsg.): Proc. 20th Int. Conf. on Advanced Information Systems Engineering (CAiSE), Montpellier, Frankreich. Bd. 5074. LNCS, Springer, S. 465–479, 2008.
- [MRA10] Mendling, J.; Reijers, H. A. und van der Aalst, W. M. P.: Seven Process Modeling Guidelines (7PMG). *Information and Software Technology* 52/2, S. 127–136, 2010.
- [MRC07] Mendling, J.; Reijers, H. A. und Cardoso, J.: What Makes Process Models Understandable? In (Alonso, G.; Dadam, P. und Rosemann, M., Hrsg.): Proc. 5th Int. Conf. on Business Process Management (BPM), Brisbane, Australien. Bd. 4714. LNCS, Springer, S. 48–63, 2007.
- [MS02] Maedche, A. und Staab, S.: Measuring Similarity Between Ontologies. In (Gómez-Pérez, A. und Benjamins, V. R., Hrsg.): Proc. 8th Int. Conf. on Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web (EKAW), Sigüenza, Spanien. Bd. 2473. LNCS, Springer, S. 251–263, 2002.
- [MS15a] Mitrovic, A. und Suraweera, P.: Teaching Database Design with Constraint-Based Tutors. *Int. Journal of Artificial Intelligence in Education* 26/1, S. 448–456, 2015.

Literaturverzeichnis

- [MS15b] Moreno-Montes de Oca, I. und Snoeck, M., Pragmatic Guidelines for Business Process Modeling, KU Leuven, Faculty of Economics und Business, Niederlande, 2015.
- [Mu89] Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77/4, S. 541–580, 1989.
- [Na14] Narciss, S.: Modelle zu den Bedingungen und Wirkungen von Feedback in Lehr-Lernsituationen. In: Feedback und Rückmeldungen. Theoretische Grundlagen, empirische Befunde, praktische Anwendungsfelder. Waxmann, S. 43–82, 2014.
- [NF11] Némec, M. und Fasuga, R.: The Automatic Evaluation Testing: Tasks with Graphics Character. In (Karlsson, G.; Burdescu, D. D. und Krämer, B., Hrsg.): Proc. Int. Conf. on Mobile, Hybrid, and On-line Learning (eL&mL), Guadeloupe, Frankreich. IEEE, S. 19–24, 2011.
- [Ni98] Nissen, M. E.: Redesigning Reengineering through Measurement-Driven Inference. MIS Quarterly: Management Information Systems 22/4, S. 509–527, 1998.
- [NP12] Navigli, R. und Ponzetto, S. P.: Joining Forces Pays Off: Multilingual Joint Word Sense Disambiguation. In (Tsujii, J.; Henderson, J. und Pasca, M., Hrsg.): Proc. 2012 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea. ACL, S. 1399–1410, 2012.
- [NR02] Nüttgens, M. und Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In (Desel, J. und Weske, M., Hrsg.): Proc. Workshop Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen (Promise), Potsdam, Deutschland. Bd. P-21. LNI, Gesellschaft für Informatik e.V., S. 64–77, 2002.
- [Ob14] Object Management Group (OMG): Business Process Model And Notation Specification, 2014, URL: <https://www.omg.org/spec/BPMN/2.0.2/PDF>, Stand: 27. 12. 2019.
- [Ob16] Object Management Group (OMG): Case Management Model and Notation Specification, 2016, URL: <https://www.omg.org/spec/CMMN/1.1/PDF>, Stand: 22. 12. 2023.
- [Ob17] Object Management Group (OMG): Unified Modeling Language Specification, 2017, URL: <https://www.omg.org/spec/UML/2.5.1/PDF>, Stand: 26. 11. 2019.
- [Ob23] Object Management Group (OMG): Decision Model and Notation Specification, 2023, URL: <https://www.omg.org/spec/DMN/1.5/Beta1/PDF>, Stand: 22. 12. 2023.
- [Ob96] Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen. Teubner, 1996.
- [OBS12] Overhage, S.; Birkmeier, D. Q. und Schlauderer, S.: Qualitätsmerkmale,-metriken und-messverfahren für Geschäftsprozessmodelle. Wirtschaftsinformatik 5, S. 217–235, 2012.
- [OK19] Ogata, S. und Kayama, M.: SML4C: Fully Automatic Classification of State Machine Models for Model Inspection in Education. In (Pretschner, A.; Voss, S. und Burgueño, L., Hrsg.): Proc. 22nd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), München, Deutschland. IEEE, S. 720–729, 2019.
- [OKO19] Ogata, S.; Kayama, M. und Okano, K.: Approach to Testing Many State Machine Models in Education. In (Uhomoihi, J.; H. C, L. und Zvacek, S., Hrsg.): Proc. 11th Int. Conf. on Computer Supported Education (CSEDU), Kreta, Griechenland. SciTePress, S. 481–488, 2019.
- [OLT14] Outair, A.; Lyhyaoui, A. und Tanana, M.: Towards a semi automatic assessment of UML diagrams by graph transformation. In (Essaaidi, M. und Zaz, Y., Hrsg.): Proc. Int. Conf. on Multimedia Computing and Systems (ICMCS), Ouarzazate, Marokko. IEEE, S. 668–673, 2014.

- [OMA21] Outair, A.; Mariam, T. und Abdelouahid, L.: New method for summative evaluation of UML class diagrams based on graph similarities. *Int. Journal of Electrical and Computer Engineering* 11/2, S. 1578–1590, 2021.
- [Os09] Ossner, J.: Kompetenzen und Kompetenzmodelle im Deutschunterricht. *Didaktik Deutsch* 21, S. 5–19, 2009.
- [Ös10] Österle, H.; Becker, J.; Frank, U.; Hess, T.; Karagiannis, D.; Krcmar, H.; Loos, P.; Oberweis, A.; Mertens, P. und Sinz, E. J.: Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. *Zeitschrift für betriebswirtschaftliche Forschung* 62/6, S. 664–669, 2010.
- [Pa21] Page, M.; McKenzie, J.; Bossuyt, P.; Boutron, I.; Hoffmann, T.; Mulrow, C.; Shamseer, L.; Tetzlaff, J.; Akl, E.; Brennan, S.; Chou, R.; Glanville, J.; Grimshaw, J.; Hróbjartsson, A.; Lalu, M.; Li, T.; Loder, E.; Mayo-Wilson, E.; McDonald, S.; McGuinness, L.; Stewart, L.; Thomas, J.; Tricco, A.; Welch, V.; Whiting, P. und Moher, D.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *Systematic Reviews* 10/89, 2021.
- [Pa68] Page, E. B.: The Use of the Computer in Analyzing Student Essays. *International Review of Education* 13/3, S. 253–263, 1968.
- [PAA13] Py, D.; Auxepaules, L. und Alonso, M.: Diagram, a Learning Environment for Initiation to Object-Oriented Modeling with UML Class Diagrams. *Journal of Interactive Learning Research* 24/4, S. 425–446, 2013.
- [PB08] Parvez, S. M. und Blank, G. D.: Individualizing Tutoring with Learning Style Based Feedback. In (Woolf, B. P.; Aimeur, E.; Nkambou, R. und Lajoje, S., Hrsg.): *Proc. 9th Int. Conf. on Intelligent Tutoring Systems (ITS)*, Montreal, Kanada. Bd. 5091. LNCS, Springer, S. 291–301, 2008.
- [Pe01] Petri, C. A.: Mathematical aspects of net theory. *Soft Computing* 5/2001, S. 3–8, 2001.
- [Pe06] Petre, M.: Cognitive dimensions »beyond the notation«. *Journal of Visual Languages and Computing* 17/4, S. 292–301, 2006.
- [Pe62] Petri, C. A.: *Kommunikation mit Automaten*, Diss., Universität Darmstadt, Deutschland, 1962.
- [Pe80] Petri, C. A.: Introduction to General Net Theory. In (Brauer, W., Hrsg.): *Proc. Advanced Course on General Net Theory of Processes and Systems*, Hamburg, Deutschland, 1979. Bd. 84. LNCS, Springer, S. 1–19, 1980.
- [Pe81] Peterson, J. L.: *Petri Net Theory and the Modelling of Systems*. Prentice Hall, 1981.
- [Pfl17] Pflanzl, N.: *Gamification for Business Process Modeling*, Diss., Universität Münster, Deutschland, 2017.
- [PLM15] Pittke, F.; Leopold, H. und Mendling, J.: Automatic Detection and Resolution of Lexical Ambiguity in Process Models. *IEEE Transactions on Software Engineering* 41/6, S. 526–544, 2015.
- [Po95a] Powell, S. G.: The Teachers' Forum: Teaching the Art of Modeling to MBA Students. *Interfaces* 25/3, S. 88–94, 1995.
- [Po95b] Powell, S. G.: The Teachers' Forum: Six Key Modeling Heuristics. *Interfaces* 25/4, S. 114–125, 1995.
- [PPP12] Purchase, H.; Pilcher, C. und Plimmer, B.: Graph Drawing Aesthetics – Created by Users, Not Algorithms. *IEEE Transactions on Visualization and Computer Graphics* 18/1, S. 81–92, 2012.

Literaturverzeichnis

- [Pr06] Prados, F.; Boada, I.; Soler, J. und Poch, J.: A Web-Based Tool for Entity-Relationship Modeling. In (Gervassi, O.; Murgante, B.; Misra, S.; Garau, C.; Blečić, I.; Taniar, D.; Apduhan, B. O.; Rocha, A. M. A. C.; Tarantino, E.; Torre, C. M. und Karaca, Y., Hrsg.): Proc. 20th Int. Conf. on Computational Science and Its Applications (ICCSA), Cagliari, Italien. Bd. 12251. LNCS, Springer, S. 364–372, 2006.
- [Pr26] Pressey, S. L.: A Simple Apparatus Which Gives Tests and Scores - and Teaches. *School and Society* 23/586, S. 373–376, 1926.
- [PS19] Pobel, S. und Striewe, M.: Domain-Specific Extensions for an E-Assessment System. In (Herzog, M. A.; Kubincová, Z.; Han, P. und Temperini, M., Hrsg.): Proc. 18th Int. Conf. Advances in Web-based Learning (ICWL), Magdeburg, Deutschland. Bd. 11841. LNCS, Springer, S. 327–331, 2019.
- [PS93] Peters, L. und Schultz, R.: The Application of Petri-Nets in Object-Oriented Enterprise Simulations. In (Milutinovic, V. und Hunter, L., Hrsg.): Proc. 26th Hawaii Int. Conf. on System Sciences (HICSS), Hawaii, USA. IEEE, S. 390–398, 1993.
- [Pu01] Purchase, H.; McGill, M.; Colpoys, L. und Carrington, D.: Graph Drawing Aesthetics and the Comprehension of UML Class Diagrams: An Empirical Study. In (Eades, P. und Pattison, T., Hrsg.): Proc. 1st Australasian Symposium on Information Visualisation (InVis), Sydney, Australien. Bd. 9. CRPIT, Australian Computer Society, S. 129–137, 2001.
- [Pu02] Purchase, H. C.: Metrics for Graph Drawing Aesthetics. *Journal of Visual Languages and Computing* 13/5, S. 501–516, 2002.
- [PV18] Pflanzl, N. und Vossen, G.: What do Business Process Modelling and Super Mario Bros. have in Common? A Games-perspective on Business Process Modelling. *Enterprise Modelling and Information Systems Architecture (EMISA)*, *Int. Journal of Conceptual Modeling* 13, S. 69–76, 2018.
- [Ra74] Ramchandani, C.: Analysis of Asynchronous Concurrent Systems by Timed Petri Nets, Diss., Massachusetts Institute of Technology, 1974.
- [Re10] Reisig, W.: Petrinetze. Modellierungstechnik, Analysemethoden, Fallstudien. Vieweg+Teubner, 2010.
- [Re12] Recker, J.: Class Notes: BPM Research and Education. How Was School Today ? BPTrends Column, 2012.
- [Re13] Recker, J.: Scientific Research in Information Systems. A Beginner's Guide. Springer, 2013.
- [Re15] Reinmann, G.: Lehrkompetenzen von Hochschullehrern: Kritik des Kompetenzbegriffs in fünf Thesen. In (Hartung, O. und Rumpf, M., Hrsg.): Lehrkompetenzen in der wissenschaftlichen Weiterbildung. Springer, Kap. 2, S. 17–36, 2015.
- [Ri99] Rittgen, P., Modified EPCs and Their Formal Semantics, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Universität Koblenz-Landau, 1999.
- [RK16] Reischmann, T. und Kuchen, H.: Towards an E-Assessment Tool for Advanced Software Engineering Skills. In (Sheard, J. und Suero Montero, C., Hrsg.): Proc. 16th Koli Calling Int. Conf. on Computing Education Research, Koli, Finnland. ACM, S. 81–90, 2016.
- [RK18] Reischmann, T. und Kuchen, H.: An Interactive Learning Environment for Software Engineering Design Patterns. In (Joy, M. und Ihantola, P., Hrsg.): Proc. 18th Koli Calling Int. Conf. on Computing Education Research, Koli, Finnland. ACM, 2018.

- [RK19] Reischmann, T. und Kuchen, H.: A Web-Based E-Assessment Tool for Design Patterns in UML Class Diagrams. In (Hung, C. und Papadopoulos, G. A., Hrsg.): Proc. 34th ACM/SIGAPP Symposium on Applied Computing (SAC), Limassol, Zypern. ACM, S. 2435–2444, 2019.
- [RM10] Ruedel, C. und Mandel, S., Hrsg.: E-Assessment. Einsatzszenarien und Erfahrungen an Hochschulen. Waxmann, 2010.
- [RM19] Reischmann, T. und Menezes, B.: Application of Swarm-intelligent Methods to Optimize Error-tolerant Graph Matching for Automatic E-Assessment. In (Alejandro, O., Hrsg.): Proc. 6th IEEE Latin American Conf. on Computational Intelligence (LA-CCI), Guayaquil, Ecuador. IEEE, 2019.
- [RR09] Recker, J. und Rosemann, M.: Teaching Business Process Modelling. Experiences and Recommendations. Communications of the Association for Information Systems 25/32, S. 379–394, 2009.
- [RS04] Richter-von Hagen, C. und Stucky, W.: Business-Process- und Workflow-Management. Prozessverbesserung durch Prozess-Management. Vieweg+ Teubner, 2004.
- [RSU20] Rehse, J. R.; Striewe, M. und Ullrich, M.: 2. Workshop zur Modellierung in der Hochschul-lehre (MoHoL). In (Michael, J.; Bork, D.; Fill, H.-G.; Fettke, P.; Karagiannis, D.; Köpke, J.; Koschmider, A.; Mayr, H. C.; Rehse, J.-R.; Reimer, U.; Striewe, M.; Tropmann-Frick, M. und Ullrich, M., Hrsg.): Proc. Modellierung 2020 Short, Workshop and Tools & Demo Papers, Wien, Österreich. Bd. 2542, CEUR-WS, 2020.
- [Ru06] Russell, N.; ter Hofstede, A. H. M.; van der Aalst, W. M. P. und Mulyar, N., Workflow Control-Flow Patterns: A Revised View, BPMcenter.org, 2006.
- [Ru20] Runyon, M.; Williford, B.; Linsey, J. und Hammond, T.: An Intelligent Interface for Automatic Grading of Sketched Free Body Diagrams. In: Proc. 25th Int. Conf. on Intelligent User Interfaces (IUI) Companion, Cagliari, Italien. ACM, S. 136–137, 2020.
- [Sa13] Sangwin, C.: Computer Aided Assessment of Mathematics. Oxford University Press, 2013.
- [Sà18] Sánchez-Ferreres, J.; van der Aa, H.; Carmona, J. und Padró, L.: Aligning textual and model-based process descriptions. Data and Knowledge Engineering 118, S. 25–40, 2018.
- [Sà20] Sánchez-Ferreres, J.; Delicado, L.; Andaloussi, A. A.; Burattin, A.; Calderon-Ruiz, G.; Weber, B.; Carmona, J. und Padro, L.: Supporting the Process of Learning and Teaching Process Models. IEEE Transactions on Learning Technologies 13/3, S. 552–566, 2020.
- [SBH09] Stone, R.; Batmaz, F. und Hinde, C.: Drawing and Marking Graph Diagrams. Innovation in Teaching and Learning in Information and Computer Sciences 8/2, S. 45–52, 2009.
- [Sc09] Schaper, N.: Aufgabenfelder und Perspektiven bei der Kompetenzmodellierung und -messung in der Lehrerbildung. Lehrerbildung auf dem Prüfstand 1/2, S. 166–199, 2009.
- [Sc12a] Schaper, N.; Reis, O.; Wildt, J.; Horvath, E. und Bender, E.: Fachgutachten zur Kompetenzorientierung in Studium und Lehre. Projekt nexus – Konzepte und gute Praxis für Studium und Lehre, 2012.
- [Sc12b] Schramm, J.; Strickroth, S.; Le, N.-T. und Pinkwart, N.: Teaching UML Skills to Novice Programmers Using a Sample Solution Based Intelligent Tutoring System. In (Lane, H. C.; Youngblood, M. und McCarthy, P. M., Hrsg.): Proc. 25th Int. Florida Artificial Intelligence Research Society Conference (FLAIRS), Marco Island, USA. AAAI, S. 472–477, 2012.

Literaturverzeichnis

- [Sc16] Schmitz, A.: Theoretischer Rahmen zum Konstrukt Textverständnis. In: Verständlichkeit von Sachtexten: Wirkung der globalen Textkohäsion auf das Textverständnis von Schülern. Springer Fachmedien Wiesbaden, S. 27–65, 2016.
- [Sc17] Schoknecht, A.; Thaler, T.; Fettke, P.; Oberweis, A. und Laue, R.: Similarity of Business Process Models: A State-of-the-Art Analysis. *ACM Computing Surveys* 50/4, 2017.
- [Sc18] Schoknecht, A.: Ähnlichkeitsbasierte Suche in Geschäftsprozessmodelldatenbanken, Diss., Karlsruher Institut für Technologie, Deutschland, 2018.
- [Sc21] Schaper, N.: Prüfen in der Hochschullehre. In (Kordts-Freudinger, R.; Schaper, N.; Scholkmann, A. und Szczyrba, B., Hrsg.): *Handbuch Hochschuldidaktik*. wbv, Bielefeld, S. 87–101, 2021.
- [Sc22] Schoknecht, A.; Thaler, T.; Laue, R.; Fettke, P. und Oberweis, A.: Process Model Similarity Techniques for Process Querying. In (Polyvyanyy, A., Hrsg.): *Process Querying Methods*. Springer, S. 459–478, 2022.
- [Sc98] Scheer, A. W.: *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. 3. Aufl., Springer, 1998.
- [SCP17] Sánchez-Ferreres, J.; Carmona, J. und Padró, L.: Aligning Textual and Graphical Descriptions of Processes Through ILP Techniques. In (Dubois, E. und Pohl, K., Hrsg.): *Proc. 29th Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, Essen, Deutschland. Bd. 10253. LNCS, Springer, S. 413–427, 2017.
- [Se10] Seethamraju, R.: Business Process Management – A Missing Link in Business Education. In (Leidner, D. und Elam, J., Hrsg.): *Proc. 16th Americas Conf. on Information Systems (AMCIS)*, Lima, Peru. AIS eLibrary, S. 4501–4511, 2010.
- [SG11] Striewe, M. und Goedicke, M.: Automated Checks on UML Diagrams. In (Rößling, G.; Naps, T. L. und Spannagel, C., Hrsg.): *Proc. 16th Annual SIGCSE Conf. on Innovation and Technology in Computer Science Education (ITiCSE)*, Darmstadt, Deutschland. ACM, S. 38–42, 2011.
- [SG14] Striewe, M. und Goedicke, M.: Automated Assessment of UML Activity Diagrams. In (Cajander, Å.; Daniels, M.; Clear, T. und Pears, A., Hrsg.): *Proc. 19th Annual SIGCSE Conf. on Innovation and Technology in Computer Science Education (ITiCSE)*, Uppsala, Schweden. ACM, S. 336, 2014.
- [SGB08] Striewe, M.; Goedicke, M. und Balz, M., *Computer Aided Assessments and Programming Exercises with JACK*, Forschungsberichte des ICB, Universität Duisburg-Essen, 2008.
- [Sh14] Shekhar, V. S.; Agarwalla, A.; Agarwal, A.; Nitish, B. und Kumar, V.: Enhancing JFLAP with automata construction problems and automated feedback. In (Parashar, M. und Bellur, U., Hrsg.): *Proc. 7th Int. Conf. on Contemporary Computing (IC3)*, Noida, Indien. IEEE, S. 19–23, 2014.
- [SHB13] Schaper, N.; Hilkenmeier, F. und Bender, E.: *Umsetzungshilfen für kompetenzorientiertes Prüfen. Projekt nexus – Konzepte und gute Praxis für Studium und Lehre*, 2013.
- [Si16] Siepermann, M.: Database Engineering Game. In (Bottino, R.; Jeurig, J. und Veltkamp, R. C., Hrsg.): *Proc. 5th Int. Conf. on Games and Learning Alliance (GALA)*, Utrecht, Niederlande. Bd. 10056. LNCS, Springer, S. 70–79, 2016.
- [SK03] Sendall, S. und Kozaczynski, W.: Model Transformation: The Heart and Soul of Model-Driven Software Development. *IEEE Software* 20/5, S. 42–45, 2003.

- [SL13] Stevens, D. und Levi, A. J.: Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback and Promote Student Learning. Stylus Publishing, 2013.
- [SL15] Sousa, R. und Leal, J. P.: A Structural Approach to Assess Graph-based Exercises. In (Sierra-Rodríguez, J. L.; Leal, J. P. und Simões, A., Hrsg.): Proc. 4th Int. Symposium on Languages, Applications and Technologies (SLATE), Madrid, Spanien. Bd. 563. CCIS, Springer, S. 182–193, 2015.
- [SL19] Scholz, T. und Lübke, D.: Improving Automatic BPMN Layouting by Experimentally Evaluating User Preferences. In (Rocha, A.; Adeli, H.; Reis, L. P. und Costanzo, S., Hrsg.): Proc. World Conf. on Information Systems and Technologies (WorldCIST), Galizien, Spanien. Bd. 930. AISC, Springer, S. 748–757, 2019.
- [SLG15] Storch, A.; Laue, R. und Gruhn, V.: Flexible Evaluation of Textual Labels in Conceptual Models. In (Kolb, J.; Leopold, H. und Mendling, J., Hrsg.): Proc. 7th Enterprise Modelling and Information Systems Architectures (EMISA), Innsbruck, Österreich. Bd. P-248. LNI, Gesellschaft für Informatik e.V., S. 17–30, 2015.
- [SM02] Suraweera, P. und Mitrovic, A.: KERMIT: A Constraint-based Tutor for Database Modeling. In (Cerri, S. A.; Guardères, G. und Paraguaçu, F., Hrsg.): Proc. 6th Int. Conf. on Intelligent Tutoring Systems (ITS), Biarritz, Frankreich und San Sebastian, Spanien. Bd. 2363. LNCS, Springer, S. 377–387, 2002.
- [SM04] Suraweera, P. und Mitrovic, A.: An Intelligent Tutoring System for Entity Relationship Modelling. *Int. Journal of Artificial Intelligence in Education* 14/3-4, S. 375–417, 2004.
- [SM08] Shukur, Z. und Mohamed, N. F.: The Design of ADAT: A Tool for Assessing Automata-Based Assignments. *Journal of Computer Science* 4/5, S. 415–420, 2008.
- [SMM04] Suraweera, P.; Mitrovic, A. und Martin, B.: The Use of Ontologies in ITS Domain Knowledge Authoring. In (Lester, J. C.; Vicari, R. M. und Paraguaçu, F., Hrsg.): Proc. 2nd Int. Workshop on Applications of Semantic Web for E-learning (SWEL), Bd. 3220. LNCS, Springer, S. 41–49, 2004.
- [SMM10] Suraweera, P.; Mitrovic, A. und Martin, B.: Widening the Knowledge Acquisition Bottleneck for Constraint-Based Tutors. *Int. Journal of Artificial Intelligence in Education* 20/2, S. 137–173, 2010.
- [SN00] Scheer, A.-W. und Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management. In (van der Aalst, W. M. P.; Desel, J. und Oberweis, A., Hrsg.): Business Process Management: Models, Techniques, and Empirical Studies. Bd. 1860. LNCS, Springer, S. 376–389, 2000.
- [So10a] Soler, J.; Boada, I.; Prados, F.; Poch, J. und Fabregat, R.: A web-based e-learning tool for UML class diagrams. In (Caeiro, M. und Carpio, J., Hrsg.): Proc. 1st IEEE Education Engineering Conf. (EDUCON), Madrid, Spanien. IEEE, S. 973–979, 2010.
- [So10b] Soler, J.; Boada, I.; Prados, F.; Poch, J. und Fabregat, R.: A Formative Assessment Tool for Conceptual Database Design Using UML Class Diagram. *Int. Journal on Emerging Technology Learning* 5, S. 27–33, 2010.
- [So11] Soler, J.; Boada, I.; Prados, F.; Poch, J. und Fabregat, R.: A Web-Based E-Learning Tool for Database Design Courses. *Int. Journal of Engineering Education* 27/1 PART 1, S. 61–69, 2011.

Literaturverzeichnis

- [SO17] Schoknecht, A. und Oberweis, A.: LS3: Latent Semantic Analysis-based Similarity Search for Process Models. Enterprise Modeling and Information Systems Architecture (EMISAJ), Int. Journal of Conceptual Modeling 12, 2017.
- [So22a] Soyka, C.; Schaper, N.; Striewe, M. und Ullrich, M.: Aufgabenkatalog für die grafische Modellierung, CC BY 4.0, Okt. 2022, URL: https://keamod.gi.de/fileadmin/PR/KEAMOD/Kompetenzmodell/Kompetenzorientierter_Aufgabenkatalog_fuer_die_grafische_Modellierung.pdf.
- [So22b] Soyka, C.; Schaper, N.; Striewe, M. und Ullrich, M.: Das Kompetenzmodell für die grafische Modellierung, CC BY 4.0, Okt. 2022, URL: https://keamod.gi.de/fileadmin/PR/KEAMOD/Kompetenzmodell/Kompetenzmodell_fuer_die_grafische_Modellierung_kompakt_.pdf.
- [So22c] Soyka, C.; Schaper, N.; Striewe, M. und Ullrich, M.: Glossar zum Kompetenzmodell für die grafische Modellierung, CC BY 4.0, Okt. 2022, URL: https://keamod.gi.de/fileadmin/PR/KEAMOD/Kompetenzmodell/Glossar_zum_Kompetenzmodell_fuer_die_grafische_Modellierung.pdf.
- [So22d] Soyka, C.; Schaper, N.; Striewe, M. und Ullrich, M.: Rubrics für Aufgabentypen im Bereich der grafischen Modellierung, CC BY 4.0, Okt. 2022, URL: https://keamod.gi.de/fileadmin/PR/KEAMOD/Kompetenzmodell/Rubrics_fuer_Modellierungsaufgaben.pdf.
- [So22e] Soyka, C.; Schaper, N.; Ullrich, M.; Striewe, M.; Schüler, S. und Schiefer, G.: Anwendungsorientierte Handreichung. Kompetenzorientiertes Prüfen in der Hochschullehre im Fachgebiet der grafischen Modellierung, CC BY-NC-SA 4.0, Okt. 2022, URL: https://keamod.gi.de/fileadmin/PR/KEAMOD/Kompetenzmodell/Anwendungsorientierte_Handreichtung_KEA-Mod.pdf.
- [So22f] Soyka, C.; Schaper, N.; Bender, E.; Striewe, M. und Ullrich, M.: Towards a Competence Model for Graphical Modeling. ACM Transactions on Computing Education, 2022.
- [So23a] Soyka, C.; Schaper, N.; Striewe, M. und Ullrich, M.: Leitfaden zum Einsatz des digitalen Fachkonzepts im Projekt KEA-Mod, CC BY 4.0, 2023, URL: https://keamod.gi.de/fileadmin/PR/KEAMOD/Kompetenzmodell/KEA_Mod_Leitfaden_zum_Einsatz_des_digitalen_Fachkonzepts.pdf.
- [So23b] Soyka, C.; Striewe, M.; Ullrich, M. und Schaper, N.: Comparison of Required Competences and Assessment Task Material in Modeling Education. Enterprise Modelling and Information Systems Architecture (EMISAJ), Int. Journal of Conceptual Modeling 18, 2023.
- [SPN21] Srijbos, J. W.; Pat-El, R. und Narciss, S.: Structural validity and invariance of the Feedback Perceptions Questionnaire. Studies in Educational Evaluation 68/100980, 2021.
- [SS21] Stottrop, T. und Striewe, M.: Analysen zur studentischen Wahl von Modellierungswerkzeugen in einer elektronischen Distanz-Prüfung. In (Wollersheim, H.; Karapanos, M. und Pengel, N., Hrsg.): Bildung in der digitalen Transformation, Tagungsband zur GMW-Jahrestagung. Bd. 78. Medien in der Wissenschaft, Waxmann, S. 131–142, 2021.
- [SS23] Soyka, C. und Schaper, N.: Verbundprojekt: Kompetenzorientiertes E-Assessment für die grafische Modellierung (KEA-Mod). Teilvorhaben der Universität Paderborn: Prüfungsdiagnostische Aspekte des kompetenzorientierten E-Assessments für die grafische Modellierung. Sachbericht zum Verwendungsnachweis, Laufzeit: 01.11.2019-31.12.2022, TIB, 2023.

- [SSL13] Siepermann, M.; Siepermann, C. und Lackes, R.: Electronic Exercises for the Metra Potential Method. In (Hammoudi, S.; Maciaszek, L. A.; Cordeiro, J. und Dietz, J., Hrsg.): Proc. 15th Int. Conf. on Enterprise Information Systems (ICEIS), Angers, Frankreich. SciTePress, S. 435–442, 2013.
- [St11] Stieler, J. F.: Validität summativer Prüfungen. Janus Presse, 2011.
- [St19] Stikkolorum, D. R.; Van Der Putten, P.; Sperandio, C. und Chaudron, M. R. V.: Towards automated grading of UML class diagrams with machine learning. In (Beuls, K.; Bogaerts, B.; Bontempi, G.; Geurts, P.; Harley, N.; Lebichot, B.; Lenaerts, T.; Louppe, G. und Eecke, P. V., Hrsg.): Proc. 31st Benelux Conf. on Artificial Intelligence and 28th Belgian Dutch Conf. on Machine Learning, Brüssel, Belgien. Bd. 2491, CEUR-WS, 2019.
- [St20] Striewe, M.; Houy, C.; Rehse, J. R.; Ullrich, M.; Fettke, P.; Schaper, N. und Oberweis, A.: Towards an Automated Assessment of Graphical (Business Process) Modelling Competences: A Research Agenda. In (Reussner, R.; Koziolok, A. und Heinrich, R., Hrsg.): 6. Workshop zum Stand und den Herausforderungen des Geschäftsprozessmanagements (ZuGPM). Jahrestagung der Gesellschaft für Informatik, Rostock, Deutschland. Bd. P-307. LNI, Gesellschaft für Informatik e.V., 2020.
- [St21] Striewe, M.; Forell, M.; Houy, C.; Pfeiffer, P.; Schiefer, G.; Schüler, S.; Soyka, C.; Stottrop, T.; Ullrich, M.; Fettke, P.; Loos, P.; Oberweis, A. und Schaper, N.: Kompetenzorientiertes E-Assessment für die grafische, konzeptuelle Modellierung. HMD Praxis der Wirtschaftsinformatik 58/1, S. 1350–1363, 2021.
- [St23a] Statistisches Bundesamt (DESTATIS): Statistik der Prüfungen: Prüfungsjahr 2022, EVAS-Nummer: 21321, Ergänzungen zur Datenbank GENESIS-Online, 2023.
- [St23b] Stiftung Innovation in der Hochschullehre: Aufgaben und Ziele - Innovationen, Vernetzung und Wissenstransfer, Webseite, 2023, URL: <https://stiftung-hochschullehre.de/ueber-uns/aufgaben-und-ziele/>, Stand: 23. 12. 2023.
- [St73] Stachowiak, H.: Allgemeine Modelltheorie. Springer, Wien, 1973.
- [STW04] Smith, N.; Thomas, P. und Waugh, K.: Interpreting Imprecise Diagrams. In (Blackwell, A. F.; Marriott, K. und Shimojima, A., Hrsg.): Proc. 3rd Int. Conf. on Theory and Application of Diagrams, Cambridge, UK. Bd. 2980. LNCS, Springer, S. 239–241, 2004.
- [STW10] Smith, N.; Thomas, P. und Waugh, K.: Diagram Interpretation and e-Learning Systems. In (Goel, A. K.; Jamnik, M. und Narayanan, N. H., Hrsg.): Proc. 6th Int. Conf. on the Theory and Application of Diagrams, Portland, USA. Bd. 6170. LNCS, Springer, S. 331–333, 2010.
- [STW13] Smith, N.; Thomas, P. und Waugh, K.: Automatic Grading of Free-Form Diagrams with Label Hypernymy. In (Pears, A., Hrsg.): Proc. 1st Int. Conf. on Learning and Teaching in Computing and Engineering (LaTiCE), Macau. IEEE, S. 136–142, 2013.
- [SWL17] Suchenia, A.; Wisniewski, P. und Ligeza, A.: Overview of Verification Tools for Business Process Models. In (Awad, A. I. und Andrzej, B., Hrsg.): Proc. Federated Conf. on Computer Science and Information Systems (FedCSIS), Prag, Tschechien. IEEE, S. 295–302, 2017.
- [Th05] Thomas, O., Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation, Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi). Nr. 184. Universität des Saarlandes, 2005.

Literaturverzeichnis

- [Th06] Thomas, O.: Management von Referenzmodellen: Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen. Logos-Verlag, 2006.
- [Th09] Thalheim, B.: Towards a Theory of Conceptual Modelling. In (Heuser, C. A. und Pernul, G., Hrsg.): Proc. Advances in Conceptual Modeling - Challenging Perspectives, Workshops at 28th Int. Conf. on Conceptual Modeling (ER), Gramado, Brasilien. Bd. 5833. LNCS, Springer, S. 45–54, 2009.
- [Th12] Thalheim, B.: The Art of Conceptual Modeling. *Frontiers in Artificial Intelligence and Applications* 237/1, S. 149–168, 2012.
- [Th13] Thomas, P.: Online Automatic Marking of Diagrams. *Systemic Practice and Action Research* 26, S. 349–359, 2013.
- [Th16] Thaler, T.; Houy, C.; Fettke, P. und Loos, P.: Automated Assessment of Process Modeling Exams: Basic Ideas and Prototypical Implementation. In (Betz, S. und Reimer, U., Hrsg.): Proc. Modellierung 2016 - Workshopband, 1. Workshop zur Modellierung in der Hochschullehre (MoHoL), Karlsruhe Deutschland. Bd. P-255. LNI, Gesellschaft für Informatik e.V., S. 63–70, 2016.
- [Th18] Thalheim, B.: Conceptual Model Notions – A Matter of Controversy: Conceptual Modelling and its Lacunas. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, *Int. Journal of Conceptual Modeling* 13, S. 9–27, 2018.
- [Ti16] Tiemeyer, E.: *Handbuch IT-System-Management*. Hanser Verlag, 2016.
- [TS07] Tselonis, C. und Sargeant, J.: Domain-Specific Formative Feedback through Domain-Independent Diagram Matching. In (Khandia, F., Hrsg.): Proc. 11th Int. Computer Assisted Assessment Conf. (CAA), Leicestershire, UK. Loughborough University, 2007.
- [TSM05] Tselonis, C.; Sargeant, J. und McGee Wood, M.: Diagram Matching for Human-Computer Collaborative Assessment. In (Danson, M., Hrsg.): Proc. 9th Int. Computer Assisted Assessment Conf. (CAA), Leicestershire, UK. Loughborough University, 2005.
- [TSW07a] Thomas, P.; Smith, N. und Waugh, K.: Computer Assisted Assessment of Diagrams. In (Hughes, J.; Ramanee Peiris, D. und Tymann, P., Hrsg.): Proc. 12th Annual SIGSCE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Dundee, Schottland. ACM, S. 68–72, 2007.
- [TSW07b] Thomas, P.; Smith, N. und Waugh, K.: Learning and Automatically Assessing Graph-based Diagrams. In (Riachi, R.; Samler, M.; Willis, H. und Pomfret, B., Hrsg.): Proc. 14th Int. Conf. of the Association for Learning Technology (ALT-C), Nottingham, UK. Association for Learning Technology, S. 249–267, 2007.
- [TSW08a] Thomas, P.; Smith, N. und Waugh, K.: Automatic Assessment of Sequence Diagrams. In (Khandia, F., Hrsg.): Proc. 12th Int. Computer Assisted Assessment Conf. (CAA), Leicestershire, UK. Loughborough University, 2008.
- [TSW08b] Thomas, P.; Smith, N. und Waugh, K.: Automatically Assessing Graph-Based Diagrams. *Learning, Media and Technology* 33/3, S. 249–267, 2008.
- [TSW09a] Thomas, P.; Smith, N. und Waugh, K.: Automatically Assessing Diagrams. In (Nunes, M. B. und McPherson, M., Hrsg.): Proc. Int. Conf. on e-Learning, Algarve, Portugal. IADIS Library, S. 263–270, 2009.

- [TWS09b] Thomas, P.; Smith, N. und Waugh, K.: The Role of Labels in the Automatic Assessment of Graph-Based Diagrams. In (Larsson, K.; Hansson, H. und Glimbert, L., Hrsg.): Proc. 23rd World Conf. on Open Learning and Distance Education, Maastricht, Niederlande. Open University, 2009.
- [TWS05] Thomas, P.; Waugh, K. und Smith, N.: Experiments in the Automatic Marking of ER-diagrams. In (Cunha, J. C.; Fleischman, W. M.; Proulx, V. K. und Lourenço, J. M., Hrsg.): Proc. 10th Annual SIGSCE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Caparica, Portugal. ACM, S. 158–162, 2005.
- [TWS06] Thomas, P.; Waugh, K. und Smith, N.: Using Patterns in the Automatic Marking of ER-diagrams. In (Davoli, R.; Goldweber, M. und Salomoni, P., Hrsg.): Proc. 11th Annual SIGSCE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Bologna, Italien. ACM, S. 83–87, 2006.
- [TWS07] Thomas, P.; Waugh, K. und Smith, N.: Tools for Supporting the Teaching and Learning of Data Modelling. In (Montgomerie, C. und Seale, J., Hrsg.): Proc. of 11th World Conf. on Educational Multimedia, Hypermedia and Telecommunications. AACE, S. 4014–4023, 2007.
- [TWS08] Thomas, P.; Waugh, K. und Smith, N.: A Revision Tool for Teaching and Learning Sequence Diagrams. In (Luca, J. und Weippl, E. R., Hrsg.): Proc. 13th World Conf. on Educational Multimedia, Hypermedia and Telecommunications. AACE, 2008.
- [TWS09] Thomas, P.; Waugh, K. und Smith, N.: Generalised Diagram Revision Tools with Automatic Marking. In (Brézillon, P.; Russell, I. und Labat, J., Hrsg.): Proc. 14th Annual SIGSCE Conf. on Innovation and Technology in Computer Science Education (ITiCSE), Paris, Frankreich. ACM, S. 318–322, 2009.
- [TWS11] Thomas, P.; Waugh, K. und Smith, N.: Generalised Diagramming Tools with Automatic Marking. *Innovation in Teaching and Learning in Information and Computer Sciences 10/1*, S. 22–34, 2011.
- [TWS12] Thomas, P.; Waugh, K. und Smith, N.: Automatically Assessing Free-Form Diagrams in E-Assessment Systems. In (Redfern, C., Hrsg.): Proc. STEM Annual Conf., London, UK. Imperial College London, 2012.
- [UFO20] Ullrich, M.; Franzen, S. E. und Oberweis, A.: Was sollen Studierende modellieren? Erweiterung klassischer Prozessmodellierungsaufgaben um den Aspekt der Prozesserhebung. In (Michael, J.; Bork, D.; Fill, H.-G.; Fettke, P.; Karagiannis, D.; Köpke, J.; Koschmider, A.; Mayr, H. C.; Rehse, J.-R.; Reimer, U.; Striewe, M.; Tropmann-Frick, M. und Ullrich, M., Hrsg.): Proc. Modellierung 2020 Short, Workshop and Tools & Demo Papers, 2. Workshop zur Modellierung in der Hochschullehre (MoHoL), Wien, Österreich. Bd. 2542, CEUR-WS, S. 113–118, 2020.
- [UI21] Ullrich, M.; Forell, M.; Houy, C.; Pfeiffer, P.; Schüler, S.; Stottrop, T.; Willems, B.; Fettke, P. und Oberweis, A.: Platform Architecture for the Diagram Assessment Domain. In (Götz, S.; Linsbauer, L.; Schäfer, I. und Wortmann, A., Hrsg.): Proc. Software Engineering 2021 Satellite Events, Workshop on Software Engineering for E-Learning Systems (SEELS), Braunschweig, Deutschland/Virtuell. Bd. 2814, CEUR-WS, 2021.
- [UI22a] Ullrich, M.; Fettke, P.; Pfeiffer, P.; Schüler, S. und Striewe, M.: 3. Workshop zur Modellierung in der Hochschullehre. In (Michael, J.; Pfeiffer, Jérôme und Wortmann, A., Hrsg.): Proc. Workshops Modellierung 2022, Hamburg, Deutschland. Gesellschaft für Informatik e.V., S. 174–175, 2022.

Literaturverzeichnis

- [UI22b] Ullrich, M.; Peiffer, P.; Schiefer, G.; Soyka, C.; Stottrop, T.; Striewe, M.; Fettke, P.; Loos, P.; Oberweis, A. und Schaper, N.: E-Assessment-Plattform für die grafische Modellierung. In (Henning, P. A.; Striewe, M. und Wölfel, M., Hrsg.): Proc. 20. Fachtagung Bildungstechnologien (DeLFI), Karlsruhe. Bd. P-322. LNI, Gesellschaft für Informatik e.V., S. 247–248, 2022.
- [UI22c] Ullrich, M.; Peiffer, P.; Schiefer, G.; Soyka, C.; Stottrop, T.; Striewe, M.; Fettke, P.; Loos, P.; Oberweis, A. und Schaper, N.: Piloteinsatz einer E-Assessment-Plattform für die grafische Modellierung. In (Henning, P. A.; Striewe, M. und Wölfel, M., Hrsg.): Proc. 20. Fachtagung Bildungstechnologien (DeLFI), Karlsruhe. Bd. P-322. LNI, Gesellschaft für Informatik e.V., S. 233–234, 2022.
- [UI23a] Ullrich, M.; Houy, C.; Stottrop, T.; Striewe, M.; Willems, B.; Fettke, P.; Loos, P. und Oberweis, A.: E-Assessment for Conceptual Models: A Systematic Literature Review. Enterprise Modelling and Information Systems Architecture (EMISAJ), Int. Journal of Conceptual Modeling 18, 2023.
- [UI23b] Ullrich, M.; Schüler, S.; Forell, M. und Schiefer, G.: Kompetenzorientiertes Prüfen in der Hochschullehre im Fachgebiet der grafischen Modellierung (KEA-Mod). In (Kleinn, K.; Slotosch, S.; Bandtel, M. und Bumann, E., Hrsg.): Digitale Prüfungen - flexibel, kompetenzorientiert und gerecht: Erfahrungen und Erkenntnisse aus dem Projekt PePP: Partnerschaft für innovative E-Prüfungen. Projektverbund der baden-württembergischen Universitäten. Freiburg im Breisgau, Heidelberg, Hohenheim, Karlsruhe, Konstanz, Mannheim, Stuttgart, Tübingen, Ulm., S. 31–34, 2023.
- [US16] Ullrich, M. und Schoknecht, A.: (Business Process) Models from an Educational Perspective. In (Hochreiner, C. und Schulte, S., Hrsg.): Proc. 8th Central European Workshop on Services and their Composition (ZEUS), Wien, Österreich. Bd. 1562, CEUR-WS, S. 53–55, 2016.
- [Va08] Vanderfeesten, I.; Reijers, H. A.; Mendling, J.; van der Aalst, W. M. P. und Cardoso, J.: On a Quest for Good Process Models: The Cross-connectivity Metric. In (Bellahsene, Z. und Léonard, M., Hrsg.): Proc. 20th Int. Conf. on Advanced Information Systems Engineering (CAISE), Montpellier, Frankreich. Bd. 5074. LNCS, Springer, S. 480–494, 2008.
- [VB08] Venkatesh, V. und Bala, H.: Technology Acceptance Model 3 and a Research Agenda on Interventions. Decision Sciences 39/2, S. 273–315, 2008.
- [VP19] Vachharajani, V. und Pareek, J.: Framework to Approximate Label Matching for Automatic Assessment of Use-case Diagram. Int. Journal of Distance Education Technologies 17/3, S. 75–95, 2019.
- [VP20] Vachharajani, V. und Pareek, J.: Effective Structure Matching Algorithm for Automatic Assessment of Use-case Diagram. Int. Journal of Distance Education Technologies 18/4, S. 31–50, 2020.
- [VPG12] Vachharajani, V.; Pareek, J. und Gulabani, S.: Effective Label Matching for Automatic Evaluation of Use-Case Diagrams. In (Banerjee, G.; Kothiyal, A. und Patwardhan, M., Hrsg.): Proc. 4th IEEE Int. Conf. on Technology for Education, Hyderabad, Indien. IEEE, S. 172–175, 2012.
- [VSV19] Van Doorn, E.; Stuurman, S. und Van Eekelen, M.: Static Detection of Design Patterns in Class Diagrams. In (Rahimi, E.; Stikkolorum, D. und Papadopoulos, G. A., Hrsg.): Proc. 8th Computer Science Education Research Conf. (CSERC), Larnaca, Zypern. ACM, S. 79–88, 2019.
- [WB05] Woodford, K. und Bancroft, P.: Multiple Choice Questions Not Considered Harmful. Conferences in Research and Practice in Information Technology Series 42, S. 109–116, 2005.

- [WDM10] Weidlich, M.; Dijkman, R. und Mendling, J.: The ICoP Framework: Identification of Correspondences between Process Models. In (Pernici, B., Hrsg.): Proc. 22nd Int. Conf. on Advanced Information Systems Engineering (CAiSE), Hammamet, Tunesien. Bd. 6051. LNCS, Springer, S. 483–498, 2010.
- [We01] Weinert, F.: Leistungsmessungen in Schulen. Beltz Pädagogik, 2001.
- [We19] Weske, M.: Business Process Management. Springer, 2019.
- [WFS12] Westergaard, M.; Fahland, D. und Stahl, C.: Grade/CPN: A Tool and Temporal Logic for Testing Colored Petri Net Models in Teaching. In (Cabac, L.; Duvigneau, M. und Moldt, D., Hrsg.): Proc. Int. Workshop on Petri Nets and Software Engineering, Hamburg, Deutschland. Bd. 851, CEUR-WS, S. 32–46, 2012.
- [WFS13] Westergaard, M.; Fahland, D. und Stahl, C.: Grade/CPN: Semi-Automatic Support for Teaching Petri Nets by Checking Many Petri Nets Against One Specification. In (Haddad, S. und Pomello, L., Hrsg.): Proc. 33rd Int. Conf. on Application and Theory of Petri Nets and Other Models of Concurrency, Hamburg, Deutschland. Bd. 7347. LNCS, Springer, S. 180–202, 2013.
- [Wi07] Wildt, J.: Vom Lehren zum Lernen. Handbuch Akkreditierung von Studiengängen, 2007.
- [Wi13] Wildt, J.: Entwicklung und Potentiale der Hochschuldidaktik. In (Heiner, M. und Wildt, J., Hrsg.): Professionalisierung der Lehre. Perspektiven formeller und informeller Entwicklung von Lehrkompetenz im Kontext der Hochschulbildung. Deutsche Gesellschaft für Hochschuldidaktik. Bertelsmann, S. 27–57, 2013.
- [Wi97] Wile, D. S.: Abstract Syntax from Concrete Syntax. In (Adrion, W. R., Hrsg.): Proc. 19th Int. Conf. on Software Engineering, Boston, USA. ACM, S. 472–480, 1997.
- [WP94] Wu, Z. und Palmer, M.: Verb Semantics and Lexical Selection. In (Pustejovsky, J., Hrsg.): Proc. 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, USA. ACM, S. 133–138, 1994.
- [WTS04] Waugh, K.; Thomas, P. und Smith, N.: Toward the Automated Assessment of Entity-relationship Diagrams. In: Proc. 2nd Workshop of the Learning and Teaching Support Network - Information and Computer Science (TLAD), Edinburgh, Schottland. Open University, 2004.
- [WTS07] Waugh, K.; Thomas, P. und Smith, N.: Teaching and Learning Applications Related to the Automated Interpretation of ERDs. In (Wilson, J. N.; Gourlay, R. S. und Tripney, B. G., Hrsg.): Proc. 24th British National Conf. on Databases (BNCOD), Glasgow, Schottland. IEEE, S. 39–47, 2007.
- [WW11] Wildt, J. und Wildt, B.: Lernprozessorientiertes Prüfen im Constructive Alignment. In (Barendt, B.; Fleischmann, A.; Schaper, N.; Szczyrba, B. und Wildt, J., Hrsg.): Neues Handbuch Hochschullehre. Deutsche Universitätszeitung, S. 1–46, 2011.
- [ZBT11] Zhang, H.; Babar, M. A. und Tell, P.: Identifying relevant studies in software engineering. Information and Software Technology 53/6, S. 625–637, 2011.
- [ZMO05] Zakharov, K.; Mitrovic, A. und Ohlsson, S.: Feedback Micro-Engineering in EER-Tutor. In (Looi, C.; McCalla, G.; Bredeweg, B. und Breuker, J., Hrsg.): Proc. 12th Int. Conf. on Artificial Intelligence in Education (AIED), Amsterdam, Niederlande. IOS Press, S. 718–725, 2005.

In vielen Anwendungsbereichen der Informatik spielt die grafische Modellierung eine wichtige Rolle. Grafische Modelle kommen beispielsweise bei der Geschäftsprozessmodellierung oder im Rahmen der Softwareentwicklung zum Einsatz, um komplexe Sachverhalte übersichtlich darzustellen. In der Hochschullehre kommt derzeit eine kompetenzorientierte Ausrichtung entsprechender Lehrveranstaltungen zu kurz, ebenso sind die Möglichkeiten zur technischen Unterstützung eingeschränkt. Die in dieser Arbeit behandelten Forschungsfragen sind daher einer kompetenzorientierten Ausrichtung des Prüfens auf dem Gebiet der grafischen Modellierung sowie der Entwicklung einer entsprechenden E-Assessment-Plattform gewidmet.
