

Institut für Produktion und Industrielles Informationsmanagement

Universität Duisburg-Essen, Campus Essen
Fakultät für Wirtschaftswissenschaften
Universitätsstraße 9, 45141 Essen
Tel.: +49 (0) 201 18 34007

Arbeitsbericht Nr. 58

zugleich

KI-LiveS-Projektbericht Nr. 12

Ontologiegestütztes Case-based Reasoning im Projektmanagement

**– Einsatz künstlicher Intelligenz
zur Wiederverwendung von projektbezogenem Erfahrungswissen –**

Mustafa, E. • Heeb, T. • Zelewski, S.



Verbundprojekt KI-LiveS: KI-Labor für verteilte und eingebettete Systeme
Förderkennzeichen: 01IS19068

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

E-Mail: stephan.zelewski@pim.uni-due.de

Internet: <https://www.pim.wiwi.uni-due.de/team/stephan-zelewski/>

ISSN 1614-0842

Essen 2022

Alle Rechte vorbehalten.

Zusammenfassung

Das BMBF-Forschungsprojekt „KI-LiveS“ (KI-Labor für verteilte und eingebettete Systeme) verfolgt primär das Transferziel („Third Mission“), Erkenntnisse aus der universitären Erforschung Künstlicher Intelligenz (KI) besser in der gewerblichen Wirtschaft zu verankern, um dort Entwicklungen von innovativen Produkten, insbesondere Dienstleistungen anzuregen, die den Wirtschaftsstandort Deutschland nachhaltig stärken. In diesem Kontext befasst sich der vorliegende Projektbericht Nr. 12 des KI-LiveS-Projekts mit der Anwendung eines ontologiegestützten Case-based-Reasoning-Systems, des prototypischen CBR-Tools jCORa, das die Wiederverwendung von Erfahrungswissen im betrieblichen Projektmanagement unterstützen soll.

Abstract

The BMBF research project ‘KI-LiveS’ (AI laboratory for distributed and embedded systems) pursues primarily the third-mission-based aim of a more effective implementation of the university research of Artificial Intelligence (AI) into trade and industry in order to stimulate the development of innovative products, especially services, which strengthen the business location Germany sustainably. In this context, this project report no. 12 of the project ‘KI-LiveS’ deals with the application of an ontology-supported case-based reasoning system, the prototypical CBR tool jCORa, which is intended to support the reuse of experience-based knowledge in corporate project management.

Danksagung

Dieser Projektbericht entstand durch die Kooperation verschiedener Personen, die am KI-LiveS-Projekt mitwirkten. Dazu zählen neben den Verfassern des Projektberichts vor allem studentische Mitarbeiter des Instituts für Produktion und Industrielles Informationsmanagement, die mit großartigem Engagement die Verfasser bei der Erstellung dieses Projektberichts unterstützt haben.

Darüber hinaus fühlen sich die Mitglieder des KI-LiveS-Projektkonsortiums („Universitätspartner“) dem BMBF als Förderer des Drittmittel-Verbundprojekts sowie dem Deutschen Zentrum für Luft- und Raumfahrt e. V. (DLR) als zuständigem Projektträger für die großzügige finanzielle Projektförderung bzw. für die professionelle Projektbegleitung zu großem Dank verbunden.

Inhaltsverzeichnis

	<u>Seite</u>
Abkürzungs- und Akronymverzeichnis	IV
Symbolverzeichnis	VIII
Abbildungsverzeichnis	X
Tabellenverzeichnis	XII
1 Exposition	1
1.1 Wiederverwendung von Erfahrungswissen im Projektmanagement	1
1.2 State of the Art der verfügbaren Techniken	3
1.3 Zielsetzung des Projektberichts.....	4
1.4 Aufbau des Projektberichts	5
2 Grundlagen der Problembearbeitung	8
2.1 Projektmanagement als relevanter Realitätsbereich.....	8
2.2 Ontologien.....	10
2.2.1 Definition von Ontologien	10
2.2.2 Ontologiekomponenten.....	11
2.2.3 Ontologiearten.....	12
2.2.4 Repräsentationssprachen von Ontologien.....	13
2.2.5 Vor- und Nachteile der Erstellung einer Ontologie	14
2.3 Case-based Reasoning.....	14
2.3.1 Definition von Case-based Reasoning	14
2.3.2 Vor- und Nachteile des Case-based Reasonings.....	17
2.4 Ontologiegestütztes Case-based Reasoning für das Management von Projektwissen.....	18
2.4.1 Eignung des ontologiegestützten Case-based Reasonings.....	18
2.4.2 Ontologiegestützte Ähnlichkeitsbestimmung	19
2.4.2.1 Ähnlichkeitsalgorithmus nach BEIBEL	19
2.4.2.2 Rekursiver Ähnlichkeitsalgorithmus nach BERGENRODT/KOWALSKI ..	20
3 Erstellung einer Projektmanagement-Ontologie mittels des Ontologie-Editors Protégé	27
3.1 Vorgehensweise bei der Erstellung einer Ontologie	27
3.2 Erstellung der Projektmanagement-Ontologie	28
4 Prototypische Implementierung des ontologiegestützten CBR-Systems jCORa.....	35
4.1 Struktur der Implementierung	35
4.2 Implementierung des rekursiven Ähnlichkeitsalgorithmus	37

4.2.1	Ein Anwendungsbeispiel	37
4.2.1.1	Überblick über das Anwendungsbeispiel	37
4.2.1.2	Falldefinitionen	37
4.2.2	Manuelle Ähnlichkeitsberechnung	39
4.2.2.1	Ausgangspunkt der Berechnung	39
4.2.2.2	Ähnlichkeitsberechnung „Neuer Fall“ in Bezug auf „Alter Fall I“	39
4.2.2.3	Ähnlichkeitsberechnung „Neuer Fall“ in Bezug auf „Alter Fall II“	44
4.2.3	Ähnlichkeitsberechnung mithilfe des CBR-Systems jCORA.....	47
4.2	Kritische Würdigung der Implementierung des Prototyps.....	48
5	Fazit und Ausblick.....	51
	Literaturverzeichnis.....	52
	Anhang 1: Quellen der Marktrecherche	64
	Anhang 2: Erstellung der Domänenontologie in Protégé.....	69
	Anhang 2.1: Installationsanleitung von Protégé	69
	Anhang 2.2: Erstellen von Klassen.....	72
	Anhang 2.3: Erstellen von Relationen	73
	Anhang 2.4: Prototyp der Domänenontologie	74
	Anhang 3: Tabelle mit Sloteigenschaften	79
	Anhang 4: Grundsätzliches Bedienungskonzept von jCORA.....	88
	Anhang 5: Fallgraphen des Projektberichts	94
	Anhang 6: Neue CBR-Anfrage und Falladaption in jCORA.....	96

Abkürzungs- und Akronymverzeichnis

AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
API	Application Programming Interface
ASWC	Asian Semantic Web Conference
ATMS	Assumption-based-Truth-Maintenance-System
Aufl.	Auflage
BProjM	Business Project Management
bzw.	beziehungsweise
BWL	Betriebswirtschaftslehre
CBR	Case-based Reasoning
CBR-Anfrage	Case-based-Reasoning-Anfrage
CBR-Anwendung	Case-based-Reasoning-Anwendung
CBR-Cycle	Case-based-Reasoning-Cycle
CBR-System	Case-based-Reasoning-System
CBR-Tool	Case-based-Reasoning-Tool
CBR-Zyklus	Case-based-Reasoning-Zyklus
C2	Command and Control
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
d. h.	das heißt
DIN	Deutsches Institut für Normung
DM	Data Mining
Dr.	Doktor
E-Commerce	Electronic Commerce
ed.	editor
eds.	editors
et al.	et alii
etc.	et cetera
f.	folgende
FAQs	Frequently Asked Questions
ff.	fortfolgende
FQAS	Flexible Query Answering Systems
ggf.	gegebenenfalls
GmbH	Gesellschaft mit geschränkter Haftung
Hrsg.	Herausgeber

html	hypertext markup language
https	hypertext transfer protocol secure
IADS	International Association for the Development of the Information Society
ICCBR	International Conference on Case-Based Reasoning
IDCS	Internet and Distributed Computing Systems
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
i.H.v.	in Höhe von
Inc.	Incorporated
IOS	Impacting the World of Science
IR	Information Retrieval
ISO	International Organization for Standardization
i.S.v.	im Sinne von
ISPRI	Forschungszentrum für Informationssysteme in Projekt- und Innovationsnetzwerken
IT	Informationstechnik
IT-Projekt	Informationstechnik-Projekt
jCOLIBRI	java (based) Cases and Ontology Libraries Integration for Building Reasoning Infrastructure
jCORA	java based Case- and Ontology-based Reasoning Application
Jg.	Jahrgang
KI	Künstliche Intelligenz
KOWIEN	Kooperatives Wissensmanagement in Engineering-Netzwerken
LCS	Least Common Subsumer
LKW	Lastkraftwagen
MRD	Machine Readable Dictionary
M. Sc.	Master of Science
NLDB	Natural Language to Data Bases
NLP	Natural Language Processing
NLP-Technik	Natural-Language-Processing-Technik
No.	Number
Nr.	Nummer
o. a.	oben angeführt
OBSE	Ontologie-basierte Software-Entwicklung
o. D.	ohne Datumsangabe
OIL	Ontology Inference Layer
OrGoLo	Organisatorische Innovationen mit Good Governance in Logistik-Netzwerken

ORPMS	Ontology-based Real-time Project Monitoring System
OTM	Online Translation Manager
o. V.	ohne Verfasserangabe
OWL	Web Ontology Language
OWL-Framework	Web-Ontology-Language-Framework
OWL-Ontologie	Web-Ontology-Language-Ontologie
OWL-Reasoner	Web-Ontology-Language-Reasoner
PAKDD	Pacific-Asia Conference on Knowledge Discovery and Data Mining
PDF	Portable Document Format
PMBOK® Guide	A guide to the Project Management Body of Knowledge
PMCatalog	Project Management Catalog
PMI	Project Management Institute
PMIS	Projektmanagement-Informationssysteme
PNG	Portable Network Graphics
Prof.	Professor
PROMONT	Project Management Ontology
RDF	Resource Description Framework
rdf	Resource Description Framework (Namensraum)
RDFS	Resource Description Framework Schema
rdfs	Resource Description Framework Schema (Namensraum)
RDQL	RDF Data Query Language
RTA-CSIT	Recent Trends and Applications in Computer Science and Information Technology
S.	Seite
SHIQ	System Hardware Input Queue
s. o.	siehe oben
Tsd.	Tausend
u. a.	unter anderem
UDE-AO	UDE-Assortment Optimization
UK	United Kingdom
Univ.-Prof.	Universitäts-Professor
URL	Uniform Resource Locator
usw.	und so weiter
vgl.	vergleiche
Vol.	Volume
vs.	versus
WiSt	Wirtschaftswissenschaftliches Studium

WS	Wintersemester
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	Extensible Markup Language
xsd	Extensible Markup Language Schema Definition (Namensraum)
z. B.	zum Beispiel

Symbolverzeichnis

\emptyset	Leere Menge
$ \dots $	Kardinalität einer Menge
$[\dots]$	(geschlossenes) Intervall
$\{\dots\}$	Menge
Σ	Summenoperator
\cap	Schnittmenge
\cup	Vereinigungsmenge
k	Klasse
k_a	Klasse k_a
k_b	Klasse k_b
$K(i)$	Klasse, von der i eine Instanz ist
$kr(k)$	Menge der Relationen und Attribute, in deren Vorbereichen sich Instanzen einer Klasse k oder einer k übergeordneten Klasse befinden
$ksim(k_a, k_b)$	Klassenähnlichkeit der Klassen k_a und k_b
$lcs(k_a, k_b)$	Least Common Subsumer der Klassen k_a und k_b
$dist(k_a, k_b)$	taxonomische Distanz zweier Klassen k_a und k_b
\in	Element (aus einer Menge oder einem Intervall)
i	Instanz
$\max(\dots)$	Maximum
$\min(\dots)$	Minimum
n	Ähnlichkeitsfunktion
N	Menge definierter Ähnlichkeitsfunktionen
$pfad(k_a, k_b)$	kleinstmögliche Menge an Kanten zwischen einer Klasse k_a und einer Klasse k_b in einer Taxonomie
r	Relation oder Attribut
R	Menge von Relationen oder Attributen
$R(i)$	Menge aller auf einer Instanz i definierten Relationen und Attribute, für die eine Ähnlichkeitsfunktion definiert ist.
$R(i,n)$	Menge aller auf einer Instanz i definierten Relationen und Attribute, deren Ähnlichkeit sich mit der Ähnlichkeitsfunktion n berechnen lässt.
$relsim(i_a, i_b, R, n)$	kumulierte Ähnlichkeit zweier Instanzen i_a und i_b bezüglich einer Menge R von Relationen oder Attributen gemessen mit einer Ähnlichkeitsfunktion n
$sim_{instanz}$	Ähnlichkeitsfunktion für Instanzen
$sim_{instanz}(r, a, b)$	Ähnlichkeitsfunktion für zwei Instanzen a und b bezüglich der Relation oder des Attributs r
$sim_n(r, a, b)$	Ähnlichkeitsfunktion n zweier Werte a und b bezüglich der Relation oder des Attributs r

$\text{sim}_{\text{numerisch}}(r, a, b)$	Ähnlichkeitsfunktion für zwei numerische Werte $a, b \in \mathbb{R}$ im Nachbereich eines Attributs r
$\text{sim}_{\text{taric}}(r, T_a, T_b)$	Ähnlichkeitsfunktion für zwei Zolltarifnummern T_a und T_b im Nachbereich eines Attributs r
$\text{sim}_{\text{bool}}(r, a, b)$	Ähnlichkeitsfunktion für zwei Boolesche Werte $a, b \in \{0; 1\}$ im Nachbereich eines Attributs r
$\text{wert}(r)$	Menge aller Werte im Nachbereich einer Relation oder eines Attributs r in einer Ontologie
$\text{wert}(i, r)$	Menge aller Werte im Nachbereich einer Relation oder eines Attributs r bezüglich einer Instanz i
\mathbb{R}	Menge der reellen Zahlen
=	ist gleich
≠	ist ungleich
≈	ist ungefähr gleich
>	ist größer
%	Prozent
+	Additionsoperator („plus“)
-	Differenzoperator („minus“) oder Bindestrich
*	Multiplikationsoperator („mal“)
€	Euro
@	At-Zeichen
®	Registered Trademark
–	Gedankenstrich
#	Rautezeichen („Nummer“)
\	Backslash
.	Punkt
,	Komma
—	Unterstrich
:	Doppelpunkt
;	Semikolon

Abbildungsverzeichnis

	<u>Seite</u>
Abbildung 1: Aufbau des Projektberichts	6
Abbildung 2: Wissensgebiete im PMBOK® Guide	9
Abbildung 3: Beispiel eines Tripels aus Subjekt, Prädikat und Objekt	11
Abbildung 4: CBR-Zyklus	15
Abbildung 5: Ausschnitt aus einer Domänenontologie	21
Abbildung 6: Subklassen der Klasse „Thing“	30
Abbildung 7: Subklassen der Klasse „Eigenschaft“	31
Abbildung 8: Instanzen der Subklasse „OriginäreQuantitativeEigenschaft“	34
Abbildung 9: Planogramm zur Sortimentsoptimierung	37
Abbildung 10: IT-Sicherheit im Unternehmen	38
Abbildung 11: Elektronische Preisschilder	38
Abbildung 12: Klassenhierarchie der Klassen „Mitarbeiter“ und „Projektleiter“	41
Abbildung 13: Klassenhierarchie für die Klassen „Vertriebskompetenz“ und „SoftwareProgrammierung“	41
Abbildung 14: Ergebnisse der Ähnlichkeitsberechnung mithilfe des CBR-Systems jCORa.....	47
Abbildung 15: Quellen der Marktrecherche	64
Abbildung 16: „Wortwolke“ der Internetrecherche.....	68
Abbildung 17: „Introduction“	69
Abbildung 18: „Choose Install Folder“	69
Abbildung 19: „Choose Shortcut Folder“	70
Abbildung 20: „Choose Java Virtual Machine“	70
Abbildung 21: „Pre-Installation Summary“	71
Abbildung 22: „Install Complete“	71
Abbildung 23: Hinzufügen einer neuen Klasse	72
Abbildung 24: Festlegung des Namensraums einer neuen Klasse	72
Abbildung 25: Festlegung der Klasseneigenschaften	72
Abbildung 26: Hinzufügen einer neuen Relation	73
Abbildung 27: Festlegung des Namensraums einer neuen Relation	73
Abbildung 28: Festlegung der Relationseigenschaften.....	73
Abbildung 29: Protégé-Oberfläche	74
Abbildung 30: Protégé-Klassen Teil 1	75
Abbildung 31: Protégé-Klassen Teil 2.....	76
Abbildung 32: Protégé-Klassen Teil 3.....	77
Abbildung 33: Ausschnitt aus den Protégé-Relationen	78

Abbildung 34	jCORA – Ansicht Fallbasis	88
Abbildung 35:	jCORA – Spracheinstellungen.....	89
Abbildung 36:	jCORA – Fallbasis-Einstellungen.....	89
Abbildung 37:	jCORA – Fallstruktur-Einstellungen	90
Abbildung 38:	jCORA – Fallbasis mit drei fiktiven Fällen	90
Abbildung 39:	jCORA – Falleditor.....	91
Abbildung 40:	jCORA – Relationseditor.....	91
Abbildung 41:	Erstellen von Instanzen in jCORA – Teil 1	92
Abbildung 42:	Erstellen von Instanzen in jCORA – Teil 2	92
Abbildung 43:	Hinzufügen von Instanzen in jCORA.....	93
Abbildung 44:	Falleditor nach Hinzufügen einer Relation.....	93
Abbildung 45:	Fallgraph des Falls „Regaloptimierung“	94
Abbildung 46:	Fallgraph des Falls „Sicherheit“	94
Abbildung 47:	Fallgraph des Falls „Pricingtool“	95
Abbildung 49:	CBR-Anfrage in jCORA.....	96
Abbildung 50:	Gewichte-Ansicht in jCORA.....	96
Abbildung 51:	Anzeige nach Abschluss der Ähnlichkeitsberechnung.....	97
Abbildung 52:	Anfrageergebnisse in jCORA	97
Abbildung 53:	Falladaption – Teil 1	98
Abbildung 54:	Falladaption – Teil 2	98

Tabellenverzeichnis

	<u>Seite</u>
Tabelle 1: Exemplarische Klassenrelationen und -attribute	22
Tabelle 2: Klasse „Projekt“ mit ihren Slots und Soteigenschaften	33
Tabelle 3: Beispiele für Relations- oder Attributwerte einer Instanz	34
Tabelle 4: Recherche wissenschaftlicher Fachliteratur zu Ontologien.....	67
Tabelle 5: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 1	79
Tabelle 6: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 2	80
Tabelle 7: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 3	81
Tabelle 8: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 4	82
Tabelle 9: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 5	83
Tabelle 10: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 6	84
Tabelle 11: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 7	85
Tabelle 12: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 8	86
Tabelle 13: Klassen der Domänenontologie mit Slots und Soteigenschaften – Teil 9	87

1 Exposition

1.1 Wiederverwendung von Erfahrungswissen im Projektmanagement

Das langjährig gesammelte Wissen aus Projekten diverser Branchen stellt eine wichtige Komponente des Projektmanagements dar. Dieses Wissen bildet für Unternehmen eine wertvolle Ressource, die oftmals in zwei Formen vorliegt.¹ Zum einen handelt es sich um spezifisches Humankapital,² d. h., dass die Ressource Wissen personengebunden ist und das Risiko eines Wissensverlusts bei einem „Wegfall“ dieser Person verloren geht.³ Zum anderen liegt es in Form von Textdokumenten vor, die in der Praxis häufig unstrukturiert sind und sich von Dritten oftmals nicht direkt erschließen lassen.⁴ Folglich besteht nach einem Projektende das Risiko, das erlangte projektspezifische Erfahrungswissen zu verlieren. Bei Neueintritt eines Mitarbeiters in ein Projekt sind vergangene Projektberichte oft mangels knapper zeitlicher Ressourcen nicht auswertbar. Infolgedessen sammelt der neue Mitarbeiter seine eigenen Erfahrungen – trotz vorhandenen Wissens zu projektspezifischen Erfolgs- und Misserfolgsk Faktoren.

Gemäß einer Studie der Technischen Universität München und der Fachzeitschrift Computerwoche zum Thema IT-Management waren lediglich 43 % aller untersuchten Projekte erfolgreich.⁵ Die gemessene Abbruchquote von IT-Projekten lag bei 20 %.⁶ Bei bis zu 50 % der Projekte gab es deutliche Abweichungen gegenüber der Kosten- und Zeitplanung, die ausschlaggebend für Qualitätseinbußen waren. Neuere Messungen zeigen, dass lediglich 29 % der untersuchten Projekte erfolgreich⁷ abgeschlossen wurden.⁸ 52 % der in diesen Zeitraum verglichenen Projekte waren teilweise⁹ erfolgreich. Die Abbruchquote¹⁰ lag 2015 bei 19 %.

-
- 1) Vgl. BERGENRODT/KOWALSKI (2015), S. 1; BEIBEL (2011), S. 2.
 - 2) Diejenigen Personen, die viele Erfahrungen in einem Unternehmen gesammelt und sich folglich viel Erfahrungswissen angeeignet haben, können als Experten betrachtet werden, die eine knappe Ressource darstellen. Diese Personen verfügen über wertvolles (Erfahrungs-)Wissen organisatorischer, sozialer und technischer Art; vgl. BRANDT/EHRENBERG/ALTHOFF et al. (2001), S. 252.
 - 3) Des Öfteren sind Personen nicht lange genug in einem Unternehmen beschäftigt, um alle Vorgehensweisen zu kennen; vgl. Fong/Kwok (2009), S. 1348; BRANDT/EHRENBERG/ALTHOFF et al. (2001), S. 252.
 - 4) Die Dokumentation von Projekten wird in erster Linie papierbasiert oder elektronisch durchgeführt, wodurch über die Zeit hinweg eine unüberschaubare Anzahl an Projekt-Dokumenten angesammelt wird, die nur noch mittels eines großen Aufwands verwaltet und wiederverwertet werden können. Der Anteil an unstrukturierten Daten in einem Unternehmen (in erster Linie Textdokumente) liegt bei ca. 80 %; vgl. TAN (1999), S. 65. Dies stellt eine Barriere hinsichtlich der Wiederverwertung von Wissen dar. Die Verfügbarkeit des Wissens ist für unerfahrene Mitarbeiter oftmals undurchsichtig. Die Angabe „80 %“ wird als Grobschätzung betrachtet. Es bleibt fraglich, wie dieser Schätzwert entstanden ist und ob es sich um eine Ad-hoc-Schätzung handelt. Zudem ist unklar, ob sich die Angabe von 80 % speziell für den Bereich des Projektmanagements als in etwa zutreffend erweist. Außerdem existieren Abweichungen von der vielfach zitierten 80 %-Regel; vgl. GRIMES (2008), S. 1-2 u. 4.
 - 5) Vgl. FRIEDMANN (2006), S. 1 (gemäß eigener Paginierung).
 - 6) Vgl. RICHTER et al. (2008), S. 4. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf die Quelle.
 - 7) Die Kriterien „Zeit“, „Budget“ und „Zielsetzung“ werden im Chaos Report als Bewertungskriterien verwendet; vgl. THE STANDISH GROUP INTERNATIONAL (2015), S. 1.
 - 8) Vgl. THE STANDISH GROUP INTERNATIONAL (2015), S. 2. Der Inhalt der nächsten beiden Sätze bezieht sich ebenfalls auf diese Quelle.
 - 9) Die Projekte wurden mit Abweichungen bezüglich der Zeit-, Budget- und Zielsetzung abgeschlossen.
 - 10) Die Projekte wurden abgebrochen.

Die Wiederverwendung¹¹ von projektspezifischem Wissen¹² stellt im Projektmanagement eine essenzielle Aufgabe dar, um sowohl Kosten einzusparen als auch um das Wissen zuvor durchgeführter, sowohl erfolgreicher als auch gescheiterter Projekte zu nutzen. Mithilfe der Erfahrungen diverser Akteure kann ein Unternehmen vor der Durchführung eines neuen Projekts die jeweiligen Anforderungen spezifizieren, bewerten und das Erfahrungswissen nutzen, um den Projektverlauf möglichst ressourcensparend zu gestalten und eine Wiederholung von Fehlern zu vermeiden.¹³

Schwachstellen bezüglich der Wissensspeicherung entstehen dadurch, dass es keine Einheitlichkeit von Begriffen gibt und so verschiedene Akteure unterschiedliche Begriffe benutzen, um die gleichen Wissensinhalte auszudrücken und zu speichern.¹⁴ Dadurch wird es oft erschwert, Ähnlichkeiten zwischen Projekten festzustellen und sich bereits vorhandenes Wissen anzueignen. Deshalb besteht ein großer Bedarf hinsichtlich einer stärkeren Strukturierung und begrifflichen Vereinheitlichung des (Erfahrungs-)Wissens aus Projekten, um den wissensbezogenen Vergleich zwischen Projekten und die Wiederverwendung von (Erfahrungs-)Wissen aus möglichst ähnlichen Projekten erheblich zu erleichtern.

Das betriebswirtschaftliche Desiderat des vorliegenden Projektberichts¹⁵ besteht aus einer qualitativ¹⁶ hochwertigeren Identifikation von möglichst ähnlichen Projekten zur Wiederverwendung von projektbezogenem Erfahrungswissen mittels Künstlicher Intelligenz (KI). Diese Identifikation soll die Grundlage zukünftiger Projekte bilden und sowohl kostengünstiger¹⁷ als auch weniger zeitaufwändig¹⁸ im Vergleich zu konventionellen IT-Techniken¹⁹ sein.

11) Die Begriffe „Wiederverwertung“ und „Wiederverwendung“ werden im Folgenden als Synonyme verstanden.

12) Das Erfahrungswissen aus Projekten wird im weiteren Verlauf auch als Projektwissen oder projektbezogenes Erfahrungswissen bezeichnet; vgl. FONG/KWOK (2009), S. 1348 f.; KOSKINEN (2004), S. 13; SNIDER/NISSEN (2003), S. 4 ff., KAMARA/AUGENBROE/ANUMBA ET AL. (2002), S. 54; FRANK/SCHÖNERT (2001), S. 25.

13) Vgl. BEIBEL (2011), S. 2. Eine der größten Schwachstellen des Projektmanagements stellt das Fehlen einer Wiederverwendung von Wissen dar; vgl. ENGEL/TAMDJIDI/QUADEJACOB (2008), S. 5.

14) Vgl. BEIBEL (2011), S. 3.

15) Der Projektbericht beruht im Wesentlichen auf der Masterarbeit von Frau MUSTAFA. Frau MUSTAFA wird daher als (Haupt-)Autorin des Projektberichts an erster Stelle genannt. Sie wurde von der erstgenannten Koautorin des Projektberichts – Frau HEEB – während ihrer Masterarbeit aus universitärer Sicht maßgeblich „gefördert und gefordert“. Frau HEEB steuerte während der Überarbeitung des Projektberichtsentswurfs weitere Impulse bei. Der zweitgenannte Koautor – Herr ZELEWSKI – hat sich vor allem um die inhaltliche und redaktionelle Endredaktion des vorliegenden Projektberichts gekümmert.

16) Eine höhere Qualität der Daten sorgt dafür, dass kein Vorwissen zum Verständnis von vorherigen Projekten notwendig ist. Die Aufarbeitung der Terminologien zur Projektbeschreibung bietet eine Grundlage für projektübergreifende Vergleiche. Dadurch soll eine einheitliche Wissensbasis geschaffen werden; vgl. BEIBEL (2011), S. 4.

17) Das Ziel der Ressourceneinsparung ist die Reduzierung des Ressourceneinsatzes von Projektwissen, Sachgütern und Dienstleistungen. Die in dieser Arbeit verwendete KI-Technik des Case-based Reasonings soll dies – im Vergleich zu den konventionellen IT-Techniken – ermöglichen, vgl. BEIBEL (2011), S. 3 f.

18) Durch Beschleunigung der Prozesse innerhalb eines Projekts werden Zeiteinsparungen generiert. Ein möglichst einfacher und schneller Zugriff auf Projektwissen ermöglicht einen schnelleren Vergleich von Projektinhalten und folglich eine leichtere Wiederverwendung von Projektwissen; vgl. BEIBEL (2011), S. 4.

19) Unter konventionellen Techniken werden Techniken verstanden, die ohne den Einsatz von Künstlicher Intelligenz auskommen.

1.2 State of the Art der verfügbaren Techniken

Zur automatischen Informationsverarbeitung mittels eines computergestützten Wissensmanagements werden in der einschlägigen Fachliteratur Techniken wie z. B. Natural Language Processing²⁰ (NLP), Information Retrieval²¹ (IR) und Data Mining²² (DM) vorgeschlagen. Nachfolgend werden Wissensmanagementtechniken Ontologien und Case-based Reasoning (CBR) näher betrachtet. Diese beiden KI-Techniken lassen sich als spezielle Ausprägungen unter dem Oberbegriff des Natural Language Processings subsumieren. Deshalb liegt der Fokus hinsichtlich der vorgenannten Techniken auf der NLP-Technik.²³

Die Techniken des Natural Language Processings umfassen eine Kommunikation zwischen menschlichen und maschinellen Akteuren, die den Inhalt von natürlichsprachigen Texten erfassen und verstehen sollen.²⁴ NLP-Techniken sind darauf ausgelegt, die Semantik eines Satzes zu erkennen.²⁵ Laut JACKSON/MOULINIER²⁶, DING/LONSDALE/EMBLEY et al.²⁷ und WITTE/GITZINGER²⁸ gehören zu diesen Techniken unter anderem die Spracherkennung sowie die interaktive Maschinenübersetzung und -interpretation.

NLP-Techniken arbeiten meist mit Wörterbüchern, die domänenspezifische Begriffe enthalten.²⁹ Die Herausforderung dieser Techniken liegt nicht nur in der Erstellung eines umfangreichen Wörterbuchs, sondern auch in der Notwendigkeit eines linguistischen Wissens zum Verständnis dieser Wörter. Je spezifischer und umfangreicher die benötigten Wörter sind, desto höher ist der Aufwand zur Erstellung eines geeigneten Wörterbuchs. Ein Vergleich zwischen Texten – hier: zwischen textuellen Projektbeschreibungen – basiert demnach nicht nur auf Ähnlichkeiten zwischen Schlüsselwörtern, sondern auch auf ihren Kontexten.³⁰

20) Der Begriff „Natural Language Processing“ lässt sich als „Verarbeitung natürlicher Sprache“ übersetzen.

21) Der Begriff „Information Retrieval“ wird definiert als das Ausfindigmachen von unstrukturiertem Material (z. B. Textdokumenten) innerhalb großer Informationsmengen zur Befriedigung des Informationsbedarfs; vgl. MANNING/RAGHAVAN/SCHÜTZE (2009), S. 1. Für den aktuellen Untersuchungsgegenstand ist IR jedoch ungeeignet, da diese Technik auf Statistiken über Dokumentensammlungen basiert und die Ähnlichkeiten häufig lediglich auf rein syntaktisch definierte Zeichenfolgen zurückzuführen sind; vgl. BEIBEL (2011), S. 4 f.

22) Der Begriff „Data Mining“ lässt sich ins Deutsche als „Datenanalyse“ übersetzen. Diversen Ansätzen zufolge ist der Begriff „Data Mining“ gleichbedeutend mit dem Begriff „Text Mining“; vgl. RAJMAN/BESANÇON (1998); FAYYAD/PIATETSKY-SHAPIRO/SMYTH (1996). Der Mining-Prozess wird lediglich auf Sequenzen von Wörtern – nicht auf Zahlen und Tabellen – angewendet; vgl. HAARMANN (2014), S. 44 f. Mit Hilfe von Textmining-Techniken sollen unstrukturierte Informationen aus Texten extrahiert werden; vgl. HAARMANN (2014), S. 44; VASILI/XHINA/NINKA et al. (2018), S. 2. Dieser Ansatz analysiert Texte anhand von statistischen und vor allem rein syntaktischen Gesichtspunkten, wodurch semantische, linguistische und heuristische Regeln unberücksichtigt bleiben; vgl. HAARMANN (2014), S. 45.

23) Vgl. KULICHIKHINA/RUBAN (2013), S. 103; DING/LONSDALE/EMBLEY et al. (2007), S. 131 ff.; LEE/NG (2006), S. 834; LEPRATTI (2006), S. 656; CHOI/PARK (2005), S. 1; Jackson/Moulinier (2002), S. 2 f.

24) Vgl. Kabel (2020), S. 46 ff.; CAMBRIA/WHITE (2014), S. 48 f.;

25) Ein Satz wird mit Hilfe der NLP-Techniken in seine Hauptbestandteile zerlegt und Algorithmen führen anschließend auf unterschiedlichen Ebenen inhaltliche Analysen durch; vgl. ACHANANUPARP/HU/ZHOU et al. (2008), S. 204.

26) Vgl. JACKSON/MOULINIER (2002), S. 3 f.

27) Vgl. DING/LONSDALE/EMBLEY et al. (2007), S. 131.

28) Vgl. WITTE/GITZINGER (2008), S. 366.

29) Vgl. LENZ/HÜBNER/KUNZE (1998), S. 246.

30) Vgl. BEIBEL (2011), S. 6.

Im Folgenden wird eine Domänenontologie³¹ erstellt, die möglichst viele relevante Begriffe aus dem Bereich des Projektmanagements enthält. Die Erstellung der Domänenontologie stützt sich – u. a. aufgrund umfangreicher Recherchen – auf den Inhalt projektbezogener natürlichsprachlicher Texte, die anschließend mit Hilfe von ontologiegestütztem Case-based Reasoning zum Vergleich von Projekten genutzt werden können.³²

1.3 Zielsetzung des Projektberichts

Das gesammelte Projektwissen eines Unternehmens liegt meist in einer nicht oder nur ansatzweise strukturierteren Form vor. Folglich bedarf es eines größeren Aufwands zur Erfassung der einzelnen Projekteinhalte. Um dieses un- oder allenfalls teilstrukturierte Wissen schneller in seiner Gänze erfassen zu können, ist eine Technik erforderlich, mit der sich das Projektwissen wiederverwenden lässt.³³ Die Basis bildet eine Formalisierung des Wissens. Für das Formalisieren und Analysieren von Wissen sollte die Technik computergestützt³⁴ sein (im Zweifelsfall eine „moderne“ KI-Technik), weil maschinelle Akteure schneller und präziser arbeiten als menschliche Akteure.³⁵ Unter diesen Umständen können die Qualität der Datenaufbereitung und -abfrage gesteigert und zeitgleich die entstehenden Kosten reduziert werden.³⁶ Außerdem sind durch den Computereinsatz deutliche Zeiteinsparungen möglich.

Liegt eine Vielzahl von Projekten vor, kann die Art und Weise der Darstellung von Projektberichten sehr unterschiedlich ausfallen. Diese Divergenz der Datenspeicherung führt oftmals zu Kommunikations- und Verständnisproblemen.³⁷ Die Barrieren hinsichtlich der Kommunikation und eines einheitlichen inhaltlichen Verständnisses von Begriffen gilt es mittels des Einsatzes von Ontologien zu überwinden.

Ontologien ermöglichen ein gemeinsames Verständnis von domänenspezifischen Informationen – z. B. mit Hilfe der Projektmanagementdomäne.³⁸ Mit eindeutigen Semantiken können unterschiedliche Begriffe für dieselben Inhalte (z. B. als Synonyme) vereinheitlicht werden. Die Nutzung von Ontologien ermöglicht allen beteiligten Akteuren das Auffinden von Ähnlichkeiten in Projekten.³⁹ Aktuell werden sie noch wenig beachtet, sind aber verstärkt als „neuartige“ Erkenntnis- und Gestaltungsobjekte zu würdigen.⁴⁰ Ontologien und Case-based Reasoning werden in der wissenschaftlichen Literatur bislang meist getrennt betrachtet.

Ontologien wurden im Projektmanagement z. B. von HUGHES⁴¹ zur Modellierung und Anpassung von Projektmanagementprozessen verwendet. SHEEBA/KRISHNAN/BERNARD⁴² haben Ontologien zur automatischen Klassifizierung von Lernmaterialien des Projektmanagements genutzt. ZELEWSKI/

31) Der Begriff „Domäne“ wird synonym zum Begriff „Gegenstandsbereich“ verwendet.

32) In der einschlägigen Fachliteratur existieren Ansätze, welche die Techniken NLP und CBR miteinander kombinieren; vgl. PFUHL (2003), S. 22 f.

33) Konzeptbasierte Verfahren greifen auf semantische Netzwerke und Ontologien zurück, um eine semantische Textanalyse durchzuführen. Diese Textanalyse geht über die reine Auswertung des Auftretens von Schlagwörtern hinaus und erfasst zudem subtilere inhaltliche Zusammenhänge; vgl. MERTENS/BODENDORF/KÖNIG et al. (2017), S. 55.

34) Unter dem Begriff „computergestützt“ wird der Einsatz von automatischer Informationsverarbeitung verstanden.

35) Als langfristiges Ziel wird eine „sinnvolle“ Vollautomation angestrebt; vgl. MERTENS (1995), S. 48.

36) Vgl. MERTENS/BODENDORF/KÖNIG et al. (2017), S. 4.

37) Vgl. BEIBEL (2011), S. 7.

38) Vgl. FENSEL (2004), S. 123.

39) Vgl. BEIBEL (2011), S. 8.

40) Vgl. ZELEWSKI (2005), S. 115.

41) Vgl. HUGHES (2010), S. 7 ff.

42) Vgl. SHEEBA/KRISHNAN/BERNARD (2012), S. 1 ff.

ALAN/ALPARSLAN et al.⁴³ haben im Rahmen des Projekts KOWIEN eine Ontologie für Kompetenzmanagementsysteme entwickelt. Eine Kombination von Ontologien und Case-based Reasoning ist z. B. in der Veröffentlichung von BERGENRODT/KOWALSKI⁴⁴ wiederzufinden. Innerhalb des Verbundprojekts OrGoLo wurde ein ontologiegestütztes CBR-System im Bereich der Logistik erstellt.

Dadurch, dass im Bereich des Projektmanagements meist natürlichsprachlich verfasste Dokumente vorliegen, bietet sich eine Kombination von Case-based Reasoning und Ontologien an.⁴⁵ Diese Technik des ontologiegestützten Case-based Reasonings ist im Vergleich zu konventionellen Techniken nicht nur kostengünstiger⁴⁶ und qualitativ⁴⁷ besser, sondern ermöglicht deutliche Zeiteinsparungen.⁴⁸

Das Ziel des vorliegenden Projektberichts ist es, das Einsatzpotenzial eines ontologiegestützten CBR-Systems in der Domäne des Projektmanagements zu untersuchen. Diese KI-Technik soll die manuelle Identifikation von Vergleichsprojekten ersetzen und Projektwissen maschinell systematisieren. Im Rahmen dieser KI-Technik werden Ähnlichkeiten zwischen Projekten mittels spezieller, graphenbasierter KI-Algorithmen ermittelt.

Im Idealfall wird bei einem neuen Projekt eine Suche in einem ontologiegestützten CBR-System durchgeführt, um als Output das (Erfahrungs-)Wissen über möglichst ähnliche, bereits durchgeführte und im CBR-System gespeicherte Projekte zu identifizieren, um beispielsweise Erfolgsfaktoren zu übernehmen und Misserfolgsfaktoren zu vermeiden. Die Suche wird unter Verwendung eines KI-Tools, des ontologiegestützten CBR-Systems Tools jCORa, mittels Ähnlichkeitsalgorithmen unterstützt.⁴⁹ Dadurch soll das gesammelte (Erfahrungs-)Wissen des Projektmanagements eines Unternehmens nutzenstiftend wiederverwendet werden können.

1.4 Aufbau des Projektberichts

Die Abbildung 1 auf der nächsten Seite verdeutlicht den Aufbau des Projektberichts.

Zunächst erfolgt in Kapitel 2 eine Einführung in die Themenkomplexe „Ontologien“ und „Case-based Reasoning“. Der Schwerpunkt dieses Kapitels liegt in der Konzeptualisierung aller relevanten Begrifflichkeiten, wie z. B. „Projektmanagement“, „Ontologie“ und „Case-based Reasoning“. Diese Grundlagen zur Problembearbeitung umfassen auch Vor- und Nachteile von Ontologien und Case-based Reasoning. Die Kombination von Ontologien und Case-based Reasoning wird in Kapitel 2 ebenfalls thematisiert. Anschließend erfolgt eine Erläuterung des Ähnlichkeitsalgorithmus zur Berechnung von Ähnlichkeiten zwischen Projekten anhand ihrer Projektbeschreibungen.

Das Kapitel 3 umfasst die Erstellung einer Projektmanagement-Ontologie. Hierfür wird in Anlehnung an NOY/MCGUINNESS ein Prototyp für die vorgenannte Domänenontologie schrittweise entwickelt.

43) Vgl. ZELEWSKI/ALAN/ALPARSLAN et al. (2005), S. 483 ff.

44) Vgl. BERGENRODT/KOWALSKI (2015).

45) Vgl. BEIBEL (2011), S. 10.

46) Durch die Ressourceneinsparung von menschlichen Akteuren entstehen weniger Kosten. Diese Einsparungen sind jedoch der Anschaffung von Hard- und Software, der Software-Nutzung und der Software-Pflege gegenüberzustellen; vgl. BEIBEL (2011), S. 10 f.

47) Die Technik des ontologiegestützten Case-based Reasonings kann die Qualität der Suchergebnisse erhöhen, indem sie nicht nur das eigene Wissen, sondern auch das gebündelte Wissen aller – auch an vorherigen Projekten – beteiligter Personen zur Verfügung stellt; vgl. BEIBEL (2011), S. 11 f.

48) Zeitliche Einsparungen entstehen durch die Wiederverwendung von gespeichertem Wissen. Dabei entfällt der Prozess des manuellen Suchens und Erfassens von älteren Dokumenten, da dieser Vorgang computergestützt nur wenige Sekunden benötigt. Die Technik des ontologiegestützten Case-based Reasonings filtert die relevanten, möglichst ähnlichen Projekte heraus und ermöglicht so eine Reduzierung des Ressourceneinsatzes gegenüber einer manuellen Suche.

49) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 492 ff.

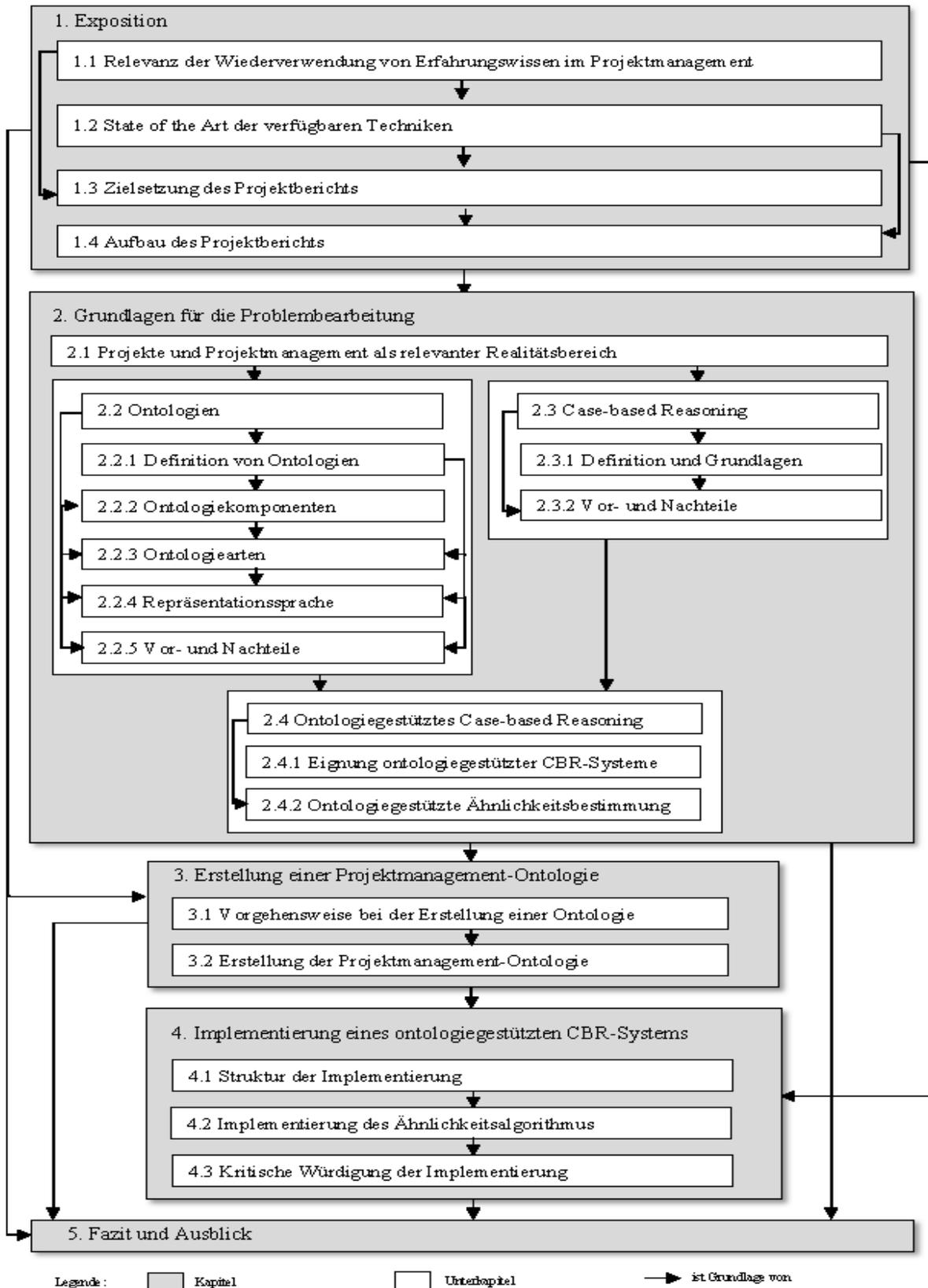


Abbildung 1: Aufbau des Projektberichts

Die Domänenontologie wird in Kapitel 4 innerhalb des ontologiegestützten CBR-Systems jCORA implementiert und mittels drei fiktiver Beispiele testweise angewendet. Der in jCORA implementierte Algorithmus zur Berechnung der Ähnlichkeiten zwischen alten und neuen Projekten wird überprüft („validiert“), indem maschinell und manuell ermittelte Ähnlichkeitswerte miteinander verglichen werden. In Kapitel 4.4 erfolgt eine kritische Würdigung des verwendeten Ähnlichkeitsalgorithmus.

Abschließend wird in Kapitel 5 ein Fazit gezogen. Außerdem erfolgt ein Ausblick auf eine mögliche Weiterentwicklung des ontologiegestützten CBR-Systems jCORA.

2 Grundlagen der Problembearbeitung

2.1 Projektmanagement als relevanter Realitätsbereich

Ein Projekt ist definiert als ein Vorhaben, das durch die Einmaligkeit der Bedingungen und der Vorgaben, wie z. B. zeitlicher, finanzieller und personeller Begrenzungen, in seiner Gesamtheit gekennzeichnet ist.⁵⁰ Mittels dieser Vorgaben soll innerhalb einer definierten Zeitspanne ein definiertes Ziel erreicht werden.

Projektmanagement ist eine Methode, die die Gesamtheit von Führungsaufgaben, -techniken, -organisationen und -mitteln für die Abwicklung von Projekten umfasst, und zwar mit dem Ziel, eine termingerechte, sachgerechte und kostengerechte Abwicklung von Projekten zu ermöglichen.⁵¹

Der am Project Management Institute entwickelte PMBOK[®] Guide stellt eine Methode für das Projektmanagement dar, die als internationaler Standard weithin anerkannt wird.⁵² Der Fokus des PMBOK[®] Guides liegt auf der Prozessperspektive des Projektmanagements. Die 47 identifizierten Projektmanagement-Prozesse werden in typische Wissensgebiete klassifiziert.⁵³ Vgl. dazu die Abbildung 2 auf der nächsten Seite.

Eine Erweiterung des PMBOK[®] Guides um das Erfahrungswissen von bereits durchgeführten Projekten könnte zur Wiederverwertung von Wissen für neue Projekte dienen.⁵⁴ Die Technik des Case-based Reasonings bietet in diesem Kontext eine Möglichkeit, Ähnlichkeiten zwischen alten Projekten und einem neuen Projekt zu ermitteln und den Planungsprozess für das neue Projekt durch die Wiederverwendung von Wissen über die alten Projekte zu unterstützen.⁵⁵

Zur natürlichsprachliche Repräsentation von Projektwissen gilt es zunächst, „Wissen“ zu definieren. Im weiteren Verlauf dieses Projektberichts wird Wissen als eine Vernetzung von zweckorientierten und kontextbezogenen Informationen verstanden, die sich aus den Erkenntnissen von Akteuren ergeben.⁵⁶

50) Gemäß der DIN-Begriffsnorm 69901 umfasst die Definition von Projekten diverse Vorgaben (Zielvorgaben; zeitliche, personelle, finanzielle und andere Begrenzungen; Abgrenzungen gegenüber anderen Vorhaben; projektspezifische Organisationen); vgl. DIN 69901-5 (2009). Bei einer kritischen Betrachtung der Definition wird erkenntlich, dass das Merkmal „Beteiligung von mehreren Personen, Arbeitsgruppen und Institutionen“ fehlt. Demnach sind Projekte arbeitsteilige Prozesse. Ein weiterer Aspekt der Definition beinhaltet die „Einmaligkeit“, die nicht bedeutet, dass sich verschiedene Teile des Projekts nicht wiederholen können oder nicht bereits Gegenstand anderer Projekte waren. Die „Einmaligkeit“ kann als Gegenteil von Routine- oder Daueraufgaben verstanden werden. Daher lautet die Definition „Einmaligkeit der Bedingungen in ihrer Gesamtheit“; vgl. SCHELLE (2014), S. 19 f. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

51) Vgl. BURGHARDT (2013), S. 10 f.; DIN 69901-5 (2009), S. 11.

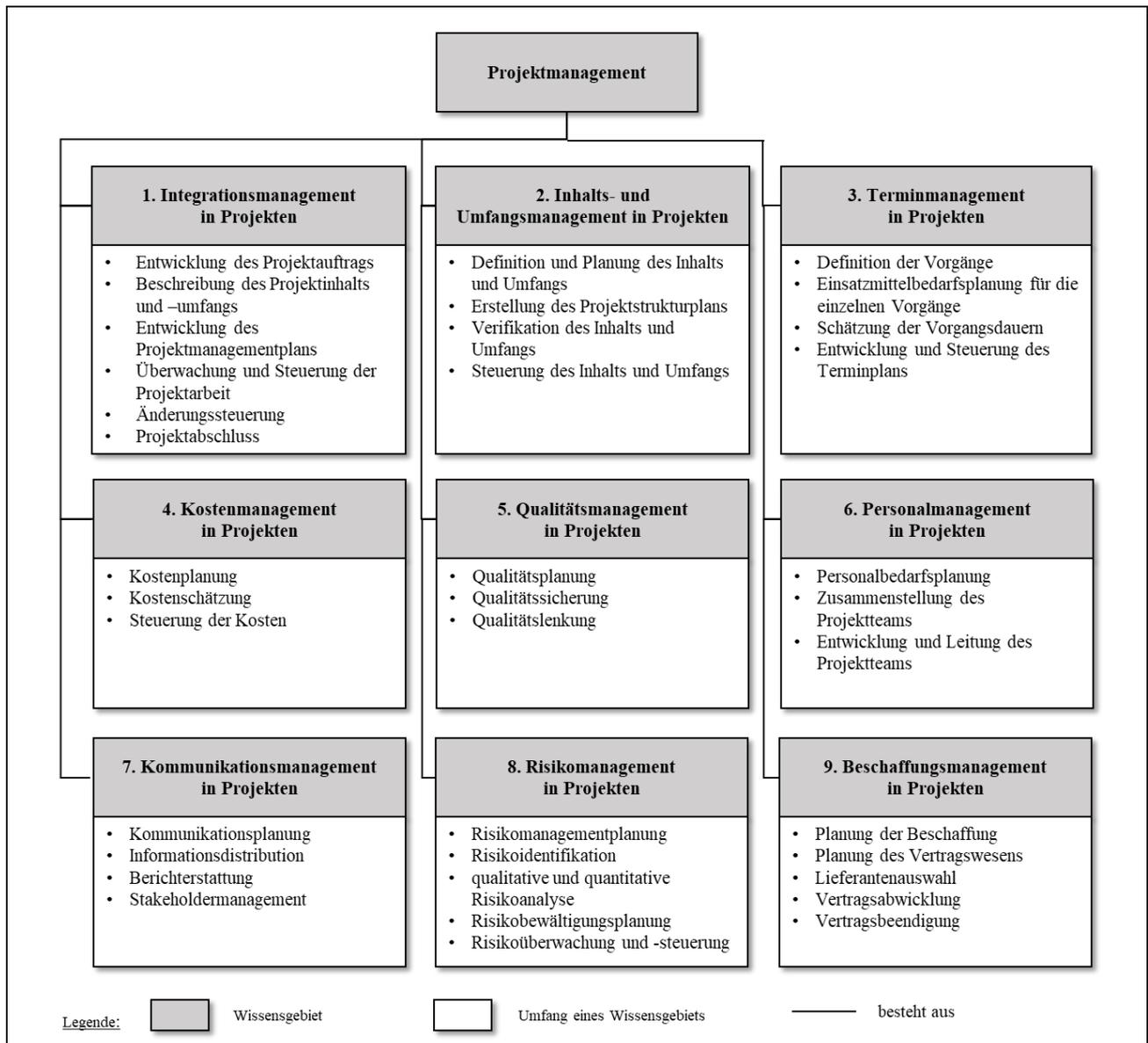
52) Vgl. PROJECT MANAGEMENT INSTITUTE (2013), S. 1.

53) Vgl. PROJECT MANAGEMENT INSTITUTE (2013), S. 60 f.

54) Vgl. BEIBEL (2011), S. 17.

55) Vgl. BEIBEL (2011), S. 17 f.

56) Die Definition von Wissen ist in der Literatur nicht eindeutig festgelegt. BERGMANN definiert den Begriff als eine Menge verknüpfter Informationen in einem aufgabenbezogenen oder zielorientierten Kontext; vgl. BERGMANN (2002), S. 26 f. Eine betriebswirtschaftliche Definition gemäß FELBERT beschreibt Wissen als das Ergebnis der Verarbeitung von Informationen und Daten durch Lernen und Intelligenz; vgl. FELBERT (1998), S. 122. PROBST/RAUB/ROMHARDT (2010), S. 23, definieren Wissen als die Gesamtheit der Fähigkeiten und Erkenntnisse, die Akteure zur Lösung von Problemen verwenden. Für den weiteren Verlauf dieses Projektberichts wird in Anlehnung an BEIBEL eine Definition gewählt, die verschiedene Wissensdefinitionen und den Schwerpunkt des Bereichs Wissensmanagement berücksichtigt; vgl. BEIBEL (2011), S. 19 f.

Abbildung 2: Wissensgebiete im PMBOK® Guide⁵⁷

Informationen sind für die Bildung von Wissen notwendig. Daher können sie als Grundlage des Wissens bezeichnet werden.⁵⁸ Sie liefern neue Gesichtspunkte zur Interpretation von Ereignissen und enthüllen Bedeutungen sowie Zusammenhänge.

Das spezifische Wissen im Kontext des Projektmanagements wird als Projektwissen bezeichnet.⁵⁹ Projektwissen besteht demnach aus einer Vernetzung projektbezogener Informationen, die aus den Erfahrungen von Akteuren bestehen.⁶⁰ Das Projektwissen wird häufig aus einer prozessorientierten und einer zustandsorientierten Perspektive betrachtet.⁶¹ Eine prozessorientierte Betrachtung intendiert in diesem Kontext einen Prozess des Projektmanagements, der bestehendes Wissen z. B. mittels Erfahrungen von Akteuren verändert und gegebenenfalls erweitert. Aus der zustandsorientierten

57) Eigene Darstellung in Anlehnung an PROJECT MANAGEMENT INSTITUTE (2013), S. 60 f. Die Abbildung umfasst lediglich einen Ausschnitt der vorhandenen Wissensgebiete.

58) Vgl. NONAKA/TAKEUCHI (2012), S. 74. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

59) Vgl. Beißel (2011), S. 20.

60) Vgl. BEIßEL (2011), S. 20.

61) Vgl. HUMPL (2004), S. 91 ff. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

Perspektive wird das Projektwissen in eine ergebnisorientierte und eine ressourcenorientierte Betrachtungsweise differenziert.⁶² Bei der ergebnisorientierten Betrachtungsweise wird angenommen, dass das projektspezifische Wissen bereits im Projektergebnis enthalten ist, z. B. in Form von Projektdokumentationen.

Risiken können in einem Projekt an verschiedenen Stellen von Prozessen auftreten. Das Nutzen von Projekterfahrungswissen bietet bei neuen Projekten die Möglichkeit, die Risiken zu reduzieren, indem Projekterfolge adaptiert und Projektmisserfolge vermieden werden.⁶³

2.2 Ontologien

2.2.1 Definition von Ontologien

Der Ursprung des Begriffs „Ontologie“ liegt in der Philosophie und wird als Teil der Metaphysik angesehen. Diese philosophische Disziplin bezeichnet die Lehre vom Seienden. ARISTOTELES definiert den Begriff als Wissenschaft vom Seienden als Seiendes.⁶⁴ Er unterscheidet verschiedene Arten des Seins, mit Hilfe derer ein System von Kategorien etabliert werden kann, wie z. B. die Kategorisierungen bezüglich Zeit, Ort, Relation, Qualität, Quantität und Substanz.⁶⁵ Dieses System sollte nicht nur „alles“ in der Welt klassifizieren, sondern auch Vorhersagen ermöglichen. Die einzelnen Kategorien können als allgemein gefasste Klassen⁶⁶ betrachtet werden, wodurch reale Phänomene strukturiert werden können.

Ontologien dienen im aktuellen Wissenschaftsbetrieb zur Erfassung und Strukturierung von Wissen. Innerhalb dieser Prozesse werden die relevanten Begriffe einer Domäne (hier: Projektmanagement), unter Berücksichtigung der Semantik von Begriffen und ihrer Beziehungen zueinander gespeichert.⁶⁷ Gemäß ZELEWSKI⁶⁸ ist eine Ontologie „eine explizite und formalsprachliche Spezifikation derjenigen

62) Vgl. HUMPL (2004), S. 92. Der Inhalt der nächsten beiden Sätze bezieht sich ebenfalls auf diese Quelle.

63) DEY (2002), S. 16, unterteilt die einzelnen Risiken u. a. in die Risikoarten „technisches Risiko“, „finanzielles Risiko“ und „organisatorisches Risiko“, vgl. . Technische Risiken umfassen beispielsweise mangelnde Sicherheit, Material-Risiken, Equipment-Risiken, Scope-Veränderungen und fehlende Projektdokumentation. Finanzielle Risiken beinhalten Kostenüberschreitungen, Inflationsrisiken und Fehleinschätzungen. Organisatorische Risiken umfassen fehlende Befugnisse, mangelnde Qualifikationen, das Fehlen eines Projektstrukturplans, kein systematisches Änderungsmanagement, fehlende Zielformulierungen, überlastete Kapazitäten und mangelnde Kommunikation.

64) Vgl. AMBÜHL (1994), S. 223.

65) Vgl. GÓMEZ-PÉREZ/CORCHO/FERNÁNDEZ-LÓPEZ (2004), S. 3. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

66) Siehe Kapitel 2.2.2.

67) Vgl. BEIBEL (2011), S. 22.

68) Der Begriff „Ontologie“ ist in der Literatur nicht eindeutig definiert. Die Definition von ZELEWSKI weist auf drei relevante Eigenschaften von Ontologien hin. Erstens sind Ontologien eindeutig und formal beschrieben. Zweitens bestehen sie aus Konzeptualisierungen, die eine Beschreibung der Realität ermöglichen. Die dritte Eigenschaft ist die Nützlichkeit für Akteure. Mit Hilfe einer Ontologie kann umfangreiches Wissen, das im eigenen Unternehmen und in öffentlich zugänglichen Publikationen zu finden ist, vereinigt und wiederverwendet werden. GRUBER weist ebenfalls auf zwei Eigenschaften von Ontologien hin, die ähnlich zu denen von ZELEWSKI sind; vgl. GRUBER (1993), S. 199. Demnach besitzen Ontologien die Eigenschaften, formal eindeutige Klassen vorzuweisen und eine gemeinsame Konzeptualisierung zu besitzen. STUDER/BENJAMINS/FENSEL erweitern GRUBERS Definition um den Aspekt einer gemeinsamen Nutzung von Ontologien; vgl. STUDER/BENJAMINS/FENSEL (1998), S. 184. LEITNER betont, dass zur gemeinsamen Kommunikation zwischen mindestens zwei Akteuren ein „gewisses Einverständnis über die Grundstruktur der Welt erforderlich [ist]“, LEITNER (2006), S. 113. Der Begriff „Grundstruktur“ ist jedoch auch in dieser Definition nicht eindeutig definiert. Daher ist in den obigen Definitionen die eindeutige Festlegung der Begriffe „Grundstruktur“, „Konzeptualisierung“ und „Spezifikation“ kritisch zu betrachten, da keine eindeutige Definition vorliegt. Die im Folgenden zugrunde liegende Definition des Begriffs „Ontologie“ umfasst ebenfalls die oben genannten Eigenschaften der formalsprachlichen Eindeutigkeit, der Konzeptualisierung und der Nutzbarkeit von Ontologien. Der Begriff „Konzeptualisierung“ wird im Sinne einer Beschreibung der Realität verstanden.

sprachlichen Ausdrucksmittel (für die Konstruktion repräsentationaler Modelle), die nach Maßgabe einer von mehreren Akteuren gemeinsam verwendeten Konzeptualisierung von realen Phänomenen, die in einem subjekt- und zweckabhängig eingesetzten Realitätsausschnitt als wahrnehmbar oder vorstellbar gelten und für die Kommunikation zwischen den o. a. Akteuren benutzt oder benötigt werden, für „sinnvoll“ erachtet werden.“⁶⁹

2.2.2 Ontologiekomponenten

Zu den wesentlichen Komponenten einer Ontologie zählen Klassen, Instanzen, Attribute und Relationen.⁷⁰ Klassen⁷¹ sind Begriffe, die abstrakte Objekte repräsentieren, die allgemein formuliert sind und für einzelne konkrete Objekte einen Überbegriff darstellen. Die Objekte mit dem höchsten Konkretisierungsgrad werden als Instanzen bezeichnet.⁷² Ein Attribut⁷³ ist eine konkrete Ausprägung eines Merkmals eines Objekts.⁷⁴ Attribute besitzen einen konkreten Wert.⁷⁵ Mittels Relationen werden Beziehungen zwischen Klassen und ihren Instanzen dargestellt.⁷⁶ Die zweistelligen Relationen besitzen die Form Subjekt-Prädikat-Objekt, siehe die nachfolgende Abbildung 3.⁷⁷

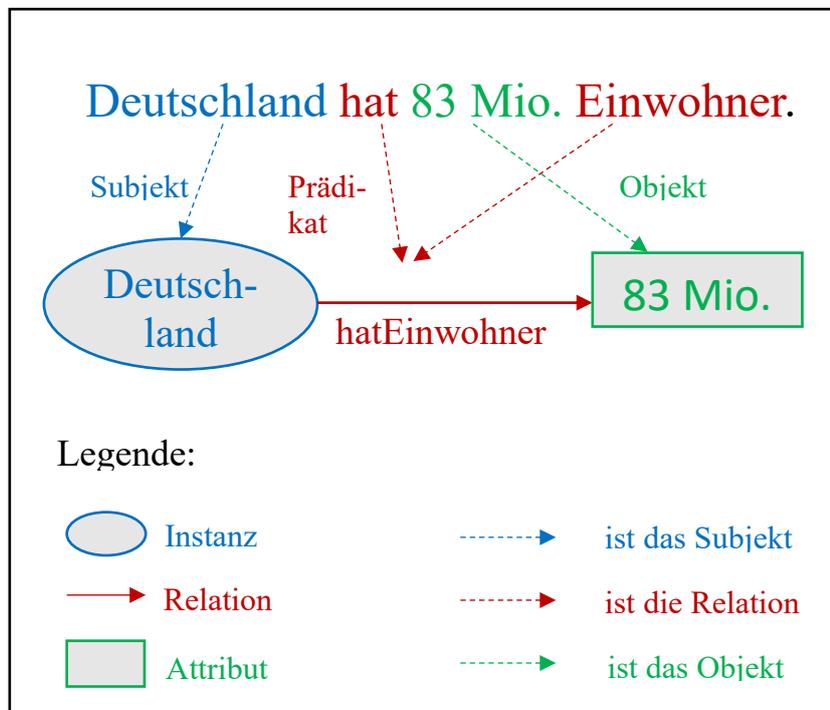


Abbildung 3: Beispiel eines Tripels aus Subjekt, Prädikat und Objekt

69) ZELEWSKI (2005), S. 153. Die Hervorhebungen wurden nicht übernommen.

70) Vgl. JAKUS/MILUTINOVIĆ/OMEROVIĆ (2013), S. 31; SCHUHBAUER/FUHR/WITTMANN (2008), S. 99.

71) In der einschlägigen Fachliteratur wird für diese Terminologie häufig entweder der Begriff „Klasse“ oder der Begriff „Konzept“ verwendet. BERGENRODT/KOWALSKI verwenden beispielsweise den Begriff „Konzept“; vgl. BERGENRODT/KOWALSKI (2015). Da in diesem Projektbericht eine Ontologie mittels des Ontologie-Editors Protégé erstellt wird und innerhalb der Software der Begriff „class“ (zu Deutsch: Klasse) verwendet wird, wird im weiteren Verlauf des Projektberichts der Begriff „Klasse“ verwendet.

72) Vgl. KOWALSKI/ZELEWSKI (2015), S. 604.

73) Attribute werden als zweistellige Relationen zwischen einer Klasse und einer weiteren Klasse spezifiziert. In diesem Projektbericht werden Attribute als eigenständige Komponenten einer Ontologie dargestellt.

74) Vgl. BERGENRODT/KOWALSKI (2015), S. 10.

75) Vgl. SCHUHBAUER/FUHR/WITTMANN (2008), S. 99.

76) Vgl. SCHWARZ/SMITH (2008), S. 155 ff.

77) Vgl. LIM/LIU/LEE (2011), S. 56.

Wird einer Klasse eine Subklasse hinzugefügt, so „vererbt“ die Subklasse alle Eigenschaften und verfügbaren Relationen der „oberen“ Klasse.⁷⁸ In diesem Zusammenhang wird von einer „Vererbung“ gesprochen.

Um das vorhandene Wissen systematisch zu strukturieren, bedarf es expliziter Spezifikationen der terminologischen, der taxonomischen, der syntaktischen und der semantischen Eigenarten des jeweiligen Wissens.⁷⁹

Terminologische Aspekte dienen der Bezeichnung von Klassen. Sie erstrecken sich auf die Normierung von Bezeichnern in bestimmten Sprachfeldern.⁸⁰ Synonyme Bezeichner für denselben Inhalt werden erfasst und im weiteren Verlauf bedeutungsgleich behandelt.⁸¹

Taxonomische Aspekte erstrecken sich über die Gesamtheit aller begrifflichen Klassen, um Über- und Unterordnungsbeziehungen zwischen diesen Klassen darzustellen. Der Ausgangspunkt für eine Taxonomie ist eine „maximal“ abstrakte Klasse („Ding“⁸²), die alle weiteren Klassen einer Ontologie subsumiert.⁸³ Jede Klasse der Ontologie bildet dadurch eine Subklasse⁸⁴ von „Ding“. Eine Hierarchisierung der Klassen erfolgt mittels einer taxonomischen „ist_ein“-Relation. Legt man z. B. fest, dass die Klasse „Unternehmen“ eine Subklasse von „Betrieb“ ist, die taxonomisch übergeordnet ist, so erfolgt die Vererbung aller Merkmale der Klasse „Betrieb“ auf die Subklasse „Unternehmen“. Die Bezeichner sind folglich nicht mehr als Synonyme zu verwenden.⁸⁵

Syntaktische Aspekte umfassen die Formen, mit denen das vorhandene Wissen dargestellt wird.⁸⁶ Dies umfasst die Frage, ob dieses Wissen mit prädikatenlogischen Formeln oder in objektartiger Form repräsentiert wird.

Semantische Aspekte betreffen Inhalte des Wissens.⁸⁷ Eine wichtige Rolle spielen in diesem Zusammenhang Homonyme. Sie stehen für einen Bezeichner, der verschiedene begriffliche Klassen bezeichnet und unterscheidet. Beispielsweise kann der Bezeichner „Unternehmen“ zwei unterschiedliche Klassen darstellen: zum einen die Klasse einer ökonomischen Institution und zum anderen die Klasse einer Aktivität (im Sinne von „etwas unternehmen“).

2.2.3 Ontologiearten

Aufgrund der Vielzahl von Gegenstandsbereichen werden in der Literatur verschiedene Arten von Ontologien differenziert. ZELEWSKI unterscheidet zwischen fünf Arten von Ontologien.⁸⁸ Die Domänenontologien werden zur Spezifikation von branchenspezifischem Wissen einer Domäne aufgestellt.⁸⁹ Eine Domäne beschreibt einen Realitätsausschnitt, der abhängig oder unabhängig von einem

78) Sobald mehrere Instanzen einer Klasse zugeordnet werden, ist es nicht erforderlich, für jede einzelne Instanz die Eigenschaften und Relationen hinzuzufügen. Beispielsweise wird für die Klasse „Auto“ festgelegt, dass ihre Instanzen einen Namen und eine Farbe haben. Diese Annahmen gelten somit für alle Instanzen der Klasse „Auto“, d. h. die Eigenschaften und Relationen wurden „vererbt“; vgl. HAARMANN (2014), S. 34.

79) Vgl. ZELEWSKI (2015), S. 93 f.

80) Bezeichner in bestimmten Sprachfeldern können z. B. branchenspezifisches Vokabular, Thesauri und Nomenklaturen sein.

81) In einer Terminologie können die Bezeichner „Informationen“ und „Wissen“ als Synonyme festgelegt werden.

82) Wird in Protégé auch als „Thing“ bezeichnet.

83) Vgl. BERGENRODT/KOWALSKI (2015), S. 10. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

84) Die Begriffe „Subklasse“ und „Unterklasse“ werden synonym verwendet.

85) Vgl. ZELEWSKI (2015), S. 93.

86) Vgl. ZELEWSKI (2015), S. 94. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

87) Vgl. ZELEWSKI (2015), S. 94. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

88) Vgl. ZELEWSKI (1999), S. 12.

89) Vgl. MCDANIEL/STOREY/SUGUMARAN (2018), S. 32.

modellierenden Subjekt existieren kann.⁹⁰ Die Commonsense-Ontologien sind nicht an spezielle Anwendungsbereiche angepasst, sondern beinhalten das allgemeine Weltwissen.⁹¹ Die Repräsentations- und Metaontologien dienen der Spezifikation der Ausdrucksmöglichkeiten von Repräsentations- und Modellierungssprachen. Die Aufgabenontologien werden für allgemeine Aufgabentypen (generische Aufgaben) verwendet, da in unterschiedlichen Anwendungsdomänen Aufgaben ähnlicher Art auftreten können (z. B. Diagnoseaufgaben). Die Methodenontologien umfassen Terme und Syntax zur Erfassung von Problemen, für deren Lösung eine Problemlösungsmethode vorgesehen ist. Darüber hinaus spricht FENSEL von Ontologien für Metadaten, die das Vokabular zur Beschreibung von Informationen aus Online-Inhalten beinhalten.⁹²

Den Gegenstand dieses Projektberichts stellt eine Domänenontologie dar, die sich auf die Projektmanagementdomäne bezieht.

2.2.4 Repräsentationssprachen von Ontologien

Als Repräsentationssprache wird eine durch ein Ontologie-Tool maschinenlesbare Form von Ontologien bezeichnet.⁹³ Ein Ontologie-Tool kann entweder eine standardisierte oder eine proprietäre Repräsentationssprache nutzen. Die Ausdrucksstärke der möglichen Aussagen innerhalb des Ontologie-Tools wird durch die Menge der festgelegten Propositionen bestimmt.⁹⁴

Einen weit verbreiteten Standard für Repräsentationssprachen stellt die vom World Wide Web Consortium⁹⁵ (W3C) entwickelte Sprache OWL (Web Ontology Language⁹⁶) dar, die eine Überarbeitung von DAML+OIL⁹⁷ ist. Mittels OWL können Informationen inhaltlich verarbeitet werden, anstatt sie nur syntaktisch zu repräsentieren.⁹⁸ In OWL werden für den Ausdruck von Eigenschaften Relationen und Regeln benötigt, die bestimmte Verkettungen ausdrücken. Beispielsweise werden für den Ausdruck „Der Onkel ist der Bruder eines Elternteils“ mehrere Relationen benötigt. Hierzu werden die Regeln „hatElternteil“ und „hatBruder“ kombiniert, um den Ausdruck „hatOnkel“ zu implizieren.⁹⁹

Für die Erstellung von Ontologien hat sich die Repräsentationssprache OWL etabliert.¹⁰⁰ Die taxonomischen und die nicht taxonomischen Beziehungen einer Ontologie lassen sich mithilfe des Konstrukts „rdf: type owl: Class“ definieren¹⁰¹ und mithilfe des Prädikats „rdfs: subclassOf“ hierarchisieren¹⁰².

90) Vgl. ZELEWSKI/SCHÜTTE/SIEDENTOPF (2001), S. 212.

91) Vgl. ZELEWSKI (1999), S. 12. Der Inhalt der nächsten drei Sätze bezieht sich ebenfalls auf diese Quelle.

92) Vgl. FENSEL (2004), S. 5.

93) Vgl. BEIBEL (2011), S. 24.

94) Als Propositionen zur Beurteilung der Ausdrucksstärke können – innerhalb der Repräsentationssprachen von Ontologien – z. B. Instanzen und Meta-Klassen genutzt werden; vgl. BEIBEL (2011), S. 24 f.

95) W3C wurde als Kommunikation-Tool zum Informationsaustausch entwickelt; vgl. o. V. (2021).

96) Vgl. o. V. (2004).

97) DAML+OIL ist eine Zusammensetzung der beiden Sprachen DAML (DARPA Agent Markup Language) und OIL (Ontology Inference Layer); vgl. DARPA (2003); VAN HARMELEN/HORROCKS (2000).

98) Vgl. o. V. (2004). Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

99) Vgl. HORROCKS/PATEL-SCHNEIDER/VAN HARMELEN (2003), S. 15.

100) Vgl. BERGENRODT/KOWALSKI (2015), S. 11.

101) Vgl. HEBELER/FISHER/BLACE et al. (2009), S. 110 f.; GÓMEZ-PÉREZ/CORCHO/FERNÁNDEZ-LÓPEZ (2004), S. 276.

102) Vgl. HEBELER/FISHER/BLACE et al. (2009), S. 111 f.; GÓMEZ-PÉREZ/CORCHO/FERNÁNDEZ-LÓPEZ (2004), S. 276.

2.2.5 Vor- und Nachteile der Erstellung einer Ontologie

Die Entwicklung von Ontologien dient nicht nur zur Kommunikation von aufgabenspezifischem Fachwissen, sondern auch zur Kommunikation von allgemeinem Hintergrundwissens.¹⁰³ Für das Zusammenwirken unterschiedlicher Akteure werden nicht nur syntaktisch definierte Informationen (z. B. Daten), sondern auch ein inhaltliches Verständnis von sprachlichen Artefakten (z. B. Wissen) zur Erstellung von Ontologien benötigt.

Die Heterogenität der Konzeptualisierung eines Realitätsausschnittes kann zu Divergenzen in der Kommunikation führen.¹⁰⁴ Damit diese Kommunikationsbarrieren überwunden werden können, bedarf es der Berücksichtigung der Semantik, da in einer Ontologie verschiedene Begriffe für dieselbe Information vereinheitlicht werden.¹⁰⁵ Dadurch können Ähnlichkeiten zwischen heterogenen natürlichsprachlichen Dokumenten erkannt werden.

In der Projektmanagementdomäne ist die Nutzung einer Ontologie wichtig, da Projekte nur dann miteinander verglichen werden können, wenn Interpretationsvorschriften für das natürlichsprachlich repräsentierte Wissen existieren.¹⁰⁶ Dieses Wissen wird in Form einer Ontologie gebündelt und strukturiert. Daher ist eine gemeinsame Interpretation des Vokabulars für die Kommunikation der Akteure unverzichtbar.

Das Design einer Ontologie hängt nicht nur von den Ontologie-Anforderungen, sondern auch von dem Programmierer ab. Eine eindeutige Abgrenzung der einzelnen Komponenten ist nicht immer eindeutig möglich. Wo z. B. Klassen enden und Instanzen beginnen, das hängt von der Granularität der Informationen und letztendlich von den individuellen Entscheidungen des Designers ab.¹⁰⁷

Ein weiterer Aspekt, der als nachteilig zu betrachten wäre, ist die grafische Darstellung von Ontologien. Je komplexer und umfangreicher die Klassen, Instanzen, Relationen und Attribute vorliegen, desto unübersichtlicher wird die grafische Darstellung.¹⁰⁸

2.3 Case-based Reasoning

2.3.1 Definition von Case-based Reasoning

Das Case-based Reasoning (CBR) (zu Deutsch: Fallbasiertes Schließen¹⁰⁹) ist eine Technik aus dem Bereich der Künstlichen Intelligenz (KI). Es bildet in diesem Projektbericht die Basis der Problemlösung. Die Ontologie der Projektmanagementdomäne dient als Unterstützung des Hauptprozesses der Identifikation und Selektion möglichst ähnlicher Projekte (Fälle) im Rahmen des Case-based Reasonings.¹¹⁰ Das in vorherigen Projekten gesammelte und in einer Wissensbank gespeicherte (Erfahrungs-)Wissen kann für neue Projekte insbesondere zur Projektplanung (Falllösung) wiederverwendet werden. Die Projekte in der Wissensbank können auch negative Projekterfahrungen und unvollständiges Projektwissen enthalten.

103) Vgl. ZELEWSKI (2015), S. 92. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

104) Vgl. BEIßEL (2011), S. 30. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

105) Eine vollständige Rekonstruktion aller Begriffe wird nicht erwartet, da die Akteure meist ein natürlichsprachliches Vorverständnis teilen; vgl. Beißel (2011), S. 30 f.; ZELEWSKI (2005), S. 150.

106) Vgl. BEIßEL (2011), S. 31. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

107) Vgl. NOY/MCGUINNESS (2001), S. 18.

108) Deswegen beschränken sich die Ontologie-Darstellungen in diesem Projektbericht auf ausgewählte Ausschnitte.

109) Vgl. FREUDENTHALER (2008), S. 1.

110) Vgl. BEIßEL (2011), S. 32. Der Inhalt der nächsten beiden Sätze bezieht sich ebenfalls auf diese Quelle.

Die Grundannahme des Case-based Reasonings besagt, dass die Bearbeitungen (Lösungen) für zwei ähnliche Projekte (Fälle) ebenfalls ähnlich sein sollten.¹¹¹

Das Case-based Reasoning basiert auf einer Nachbildung menschlichen Problemlösens, indem bestehende Erfahrungen nutzbar gemacht und ständig erweitert werden.¹¹² Diese Erfahrungen können von verschiedenen Personen gesammelt worden sein.

Die Eigenschaften eines CBR-Systems ähneln dem menschlichen Denken insofern, als dass das CBR-System Wissen über neue Projekte sammeln und sich so mit der Zeit weiterentwickeln kann.¹¹³

Dieser Lernprozess wird in der Literatur als CBR-Cycle (zu Deutsch: CBR-Zyklus) bezeichnet. Er beschreibt einen zyklischen Prozess, der aus vier Phasen besteht:¹¹⁴ Retrieve, Reuse, Revise und Retain.¹¹⁵ Voraussetzung für diese vier Phasen bildet eine vorgelagerte Anfrage-Phase, in der ein neues Projekt an das CBR-System übermittelt wird.¹¹⁶

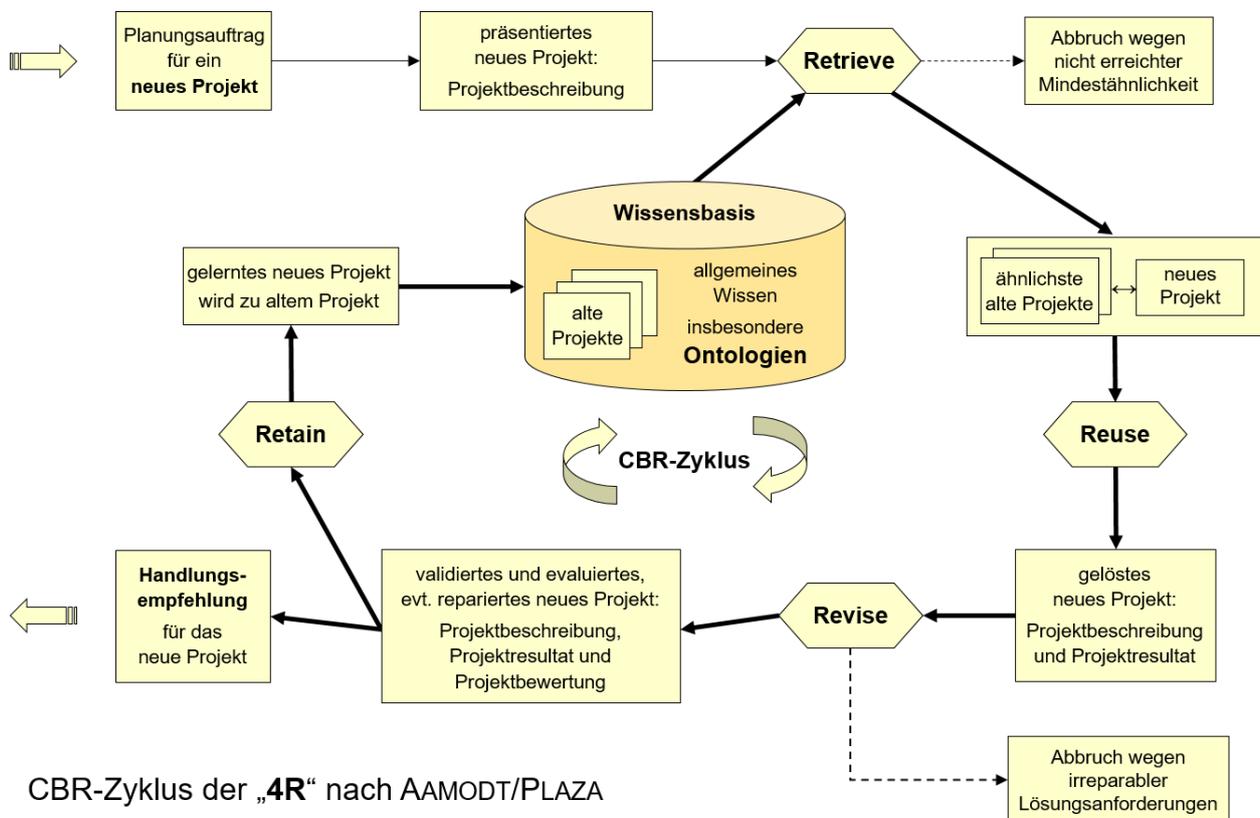


Abbildung 4: CBR-Zyklus¹¹⁷

111) Vgl. LEAKE (1996), S. 1.

112) Vgl. PAL/SHIU (2004), S. 5. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

113) Vgl. BEIBEL (2011), S. 33.

114) Vgl. AAMODT/PLAZA (1994), S. 44 ff.

115) Vgl. AAMODT/PLAZA (1994), S. 44 ff.; RECIO-GARCÍA/GONZÁLES-CALERO/DÍAZ-AGUDO (2014), S. 126 ff.

116) Vgl. BEIBEL (2011), S. 33.

117) In Anlehnung an AAMODT/PLAZA (1994), S. 44 f. Für die Erweiterungen des ursprünglichen CBR-Zyklus, insbesondere hinsichtlich der Abbruchkriterien; vgl. ZELEWSKI/SCHAGEN (2022), S. 18; ZELEWSKI (2015), S. 15. Vgl. auch leicht abweichend KOWALSKI/ZELEWSKI/GÜNES et al. (2011), S. 50.

Retrieve: Die Phase Retrieve (zu Deutsch: Abrufen) ermöglicht es dem Nutzer, einen möglichst ähnlichen alten Fall (Projekt)¹¹⁸ aus der Fallbasis (Wissensbank) des CBR-Systems abzurufen, um dessen Falllösung auf den neuen Fall anzuwenden.¹¹⁹ Dazu muss der Nutzer eine neue Fallbeschreibung in das CBR-System eingeben. Nach der Eingabe kann er die verschiedenen Aspekte (Relationen und Attribute) seiner Problemstellung subjektiv mit Gewichten versehen, die im Rahmen der Wissensverarbeitung vom System berücksichtigt werden.¹²⁰ Dadurch besteht die Möglichkeit festzulegen, welche Aspekte wichtiger und welche weniger relevant sind. Die Kombination aus der Fallbeschreibung und der Gewichtung von Aspekten wird im weiteren Verlauf als „CBR-Anfrage“ bezeichnet.

Um zu einem neuen Fall ähnlichste alte Fälle aus der Fallbasis zu erhalten, wird zunächst die Fallbasis in Form einer Grobsuche gemäß der Gewichtung nach Fällen durchsucht, die für eine Falllösung in Betracht kommen.¹²¹ Die vielversprechendsten Fälle werden anschließend in Form einer Feinsuche herausgefiltert. Somit erfolgt unter Berücksichtigung der Gewichte ein Vergleich der neuen Fallbeschreibung mit bereits hinterlegten, alten Fallbeschreibungen.¹²² Dieser Vergleich wird mittels einer Ähnlichkeitsfunktion durchgeführt, die die Ähnlichkeit zweier miteinander verglichenen Fallbeschreibungen als reelle Zahl im Intervall [0,1] ausgibt.¹²³

Ein erweiterter Ansatz des CBR-Zyklus umfasst in der Retrieve-Phase ein Abbruchkriterium bei Nichterreichen einer minimalen Ähnlichkeit zwischen alten und neuen Fällen.¹²⁴

Reuse: In der Phase Reuse (zu Deutsch: Wiederverwendung) wählt der Nutzer die für ihn am besten erscheinende Falllösung aus – in der Regel handelt es sich um einen alten Fall mit der größten Ähnlichkeit zum betrachteten neuen Fall – und verwendet das vorhandene Wissen der ausgewählten Falllösung eines ähnlichsten alten Falls für die Lösung des neuen Falls.¹²⁵

Wenn der neue Fall identisch zum ähnlichsten alten Fall ist, kann der Nutzer die Lösung für den alten Fall ohne Veränderungen auf seinen neuen Fall übertragen („Null-Adaption“). Unterscheiden sich die beiden Fälle jedoch in einigen Aspekten, so muss eine Adaption erfolgen. Die Lösung des ähnlichsten alten Falls wird dann an den neuen Fall angepasst.¹²⁶ Mittels der Adaption wird versucht, die Unterschiede zwischen dem neuen und dem ähnlichsten alten Fall auszugleichen.¹²⁷

118) Die Bezeichnungen „Fall“ und „Projekt“ werden hier sowie im gesamten Projektbericht synonym verwendet. Hier im Kontext des CBR-Zyklus wird vornehmlich von Fällen gesprochen.

119) Vgl. RICHTER/WEBER (2013), S. 26. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

120) Vgl. AAMODT/PLAZA (1994), S. 50.

121) Vgl. BEIERLE/KERN-ISBERNER (2019), S. 170. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

122) Vgl. BERGENRODT/KOWALSKI (2015), S. 21.

123) Bei einem Vergleich zweier Fallbeschreibungen werden die Relationen und Attribute miteinander verglichen. Sind diese identisch, wird ihrer Ähnlichkeit der Wert 1 zugewiesen. Sind diese jedoch völlig unterschiedlich, erhält ihre Ähnlichkeit den Wert 0. Vgl. RICHTER/WEBER (2013), S. 17.

124) Vgl. KOWALSKI/KLÜPFEL/ZELEWSKI et al. (2013), S. 292.

125) Vgl. RICHTER/WEBER (2013), S. 33. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

126) Vgl. KOLODNER (1993), S. 7.

127) Vgl. CHANGCHIEN/LIN (2005), S. 49 f.

Revise: In der Phase Revise (zu Deutsch: Überprüfung oder Evaluierung) überprüft der Nutzer die vom CBR-System vorgeschlagene Lösung.¹²⁸ Diese Evaluierung kann sowohl in der Praxis als auch mittels einer Simulation durchgeführt werden.¹²⁹ Die in dieser Überprüfungsphase gesammelten Erkenntnisse dienen anschließend zur Erweiterung, Anpassung¹³⁰ und Bewertung der vorgeschlagenen Lösung im Rahmen der Fallbewertung.¹³¹ Die Fallbewertung dient vor allem zur Identifikation von spezifischen Informationen zu Erfolgs- oder Misserfolgskriterien der durchgeführten Lösung.¹³²

Gemäß dem erweiterten Ansatz des CBR-Zyklus erfolgt ein Abbruch der Revise-Phase im Fall nicht realisierbarer Lösungsanforderungen.¹³³

Retain: In der Phase Retain (zu Deutsch: Aufbewahrung) bildet die Gesamtheit aus der aktuellen Fallbeschreibung, der gegebenenfalls überarbeiteten Falllösung und der Fallbewertung schließlich einen neuen Fall, der in die Fallbasis neu aufgenommen wird.¹³⁴ Wird ein neuer Fall erfolgreich gelöst, so werden die damit verbundenen Erfahrungen in der Fallbasis gesichert, um in Zukunft bei ähnlichen Fällen als mögliche Falllösung zu dienen.¹³⁵ Dadurch können bereits durchgeführte Fehler identifiziert und vermieden werden.

2.3.2 Vor- und Nachteile des Case-based Reasonings

Mittels des Case-based Reasonings können bereits wenige Fälle in der Fallbasis ausreichen, um nützliche Lösungen zu erhalten.¹³⁶ Die Problemlösung erfolgt im Allgemeinen durch das Zurückgreifen auf Erfahrungen vorheriger Projekte. Dieses Vorgehen lässt sich – bei sehr großzügiger Auslegung dieses Fachbegriffs – auch als „Lessons Learned“ bezeichnen.¹³⁷ Durch eine kontinuierliche Assimilation und Akkommodation von Wissen wird die Fallbasis laufend erweitert. Anhand dieser Integration und Anpassung fallbasierter Wissens entsteht eine Lernfähigkeit des CBR-Systems, die der Denkweise und dem Vorgehen menschlichen Lernens ähnelt.¹³⁸

Um mit Hilfe des Case-based Reasonings bereits in der Vergangenheit gemachte Fehler zu vermeiden und Erfolgsfaktoren bei neuen Projekten zu berücksichtigen, bedarf es einer ständigen Pflege der Fallbasis.¹³⁹

Die Erfahrungen aus vergangenen Projekten unterschiedlicher Akteure liegen in der Fallbasis als gesammeltes Projektwissen vor. Die Technik des Case-based Reasonings ermöglicht eine eigenständige Interpretation der Eignung von Fällen zur Problemlösung. Projektinformationen können ungenau und

128) Vgl. AAMODT/PLAZA (1994), S. 52.

129) Vgl. RICHTER/WEBER (2013), S. 33.

130) In der Überprüfungsphase wird u. a. untersucht, ob eine Lösung in einer ähnlichen Situation fehlgeschlagen ist. Der Fund solcher Fehlleistungen kann zu einer Wiederholung des gesamten Prozesses des fallbasierten Schließens führen. Es kann in diesem Zusammenhang jedoch auch eine entsprechende Lösungskorrektur vorgenommen werden, die als „Reparatur“ bezeichnet wird; vgl. BEIERLE/KERN-ISBERNER (2019), S. 172; AAMODT/PLAZA (1994), S. 52.

131) Vgl. AAMODT/PLAZA (1994), S. 52.

132) Vgl. BEIERLE/KERN-ISBERNER (2019), S. 171.

133) Vgl. KOWALSKI/KLÜPFEL/ZELEWSKI et al. (2013), S. 292.

134) Vgl. BEIERLE/KERN-ISBERNER (2019), S. 172.

135) Vgl. AAMODT/PLAZA (1994), S. 40 f. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

136) Vgl. FREUDENTHALER (2008), S. 70.

137) Vgl. REICH/WEE (2006), S. 19 f.

138) Vgl. BEIBEL (2011), S. 36.

139) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 515; NÜNNING/JÄGER (2015), S. 661; BEIBEL (2011), S. 39.

unvollständig sein, weshalb ein Informationsmangel Unsicherheiten erzeugen kann.¹⁴⁰ Diese Unsicherheiten können bereits bei der Planung der Projektanforderungen auftreten, da diese häufig nicht bis ins kleinste Detail vorgegeben werden.¹⁴¹

Da die Projektmanagementdomäne durch die Vielfalt der einzelnen Einflussgrößen als umfangreich bezeichnet werden kann, entstehen Unsicherheiten hinsichtlich der vollständigen Erfassung dieser Einflussgrößen. Die Nutzung des Case-based Reasonings erfordert jedoch keine vollständige Erfassung der Domäne, da das Vorgehen auf Ähnlichkeitsalgorithmen basiert. Ein Nachteil dieser Methode entsteht, wenn aufgrund mangelnder Repräsentativität ähnlicher Fälle die Aussagekraft der Ergebnisse unsicher bleibt.¹⁴²

Eine weitere Unsicherheit, die durch das Case-based Reasoning entsteht, ist die subjektive Bewertung der Projektergebnisse. Unterschiedliche Individuen werden nicht zwangsläufig dieselbe Betrachtungsweise und folglich dieselbe Interpretation der Ergebnisse haben. Sie können Projektverläufe durch einen unterschiedlichen Fokus oder durch persönliche Erfahrungen anders bewerten als andere Individuen.

Die Nutzung von Case-based Reasoning bietet den Nutzern dennoch den Vorteil eines zeitsparenden Auffindens von ähnlichen Fällen im Vergleich zu manuellen Techniken.¹⁴³

In diesem Projektbericht werden die möglichen Unsicherheiten des Case-based Reasonings akzeptiert, da die Vorteile in der Projektmanagementdomäne bezüglich einer Erfassung von neuem Wissen, der flexiblen Erweiterung vorhandenen Wissens und einer erfahrungsbasierten Problemlösung überwiegen.

2.4 Ontologiegestütztes Case-based Reasoning für das Management von Projektwissen

2.4.1 Eignung des ontologiegestützten Case-based Reasonings

Mit Hilfe des ontologiegestützten Case-based Reasonings wird das in der Ontologie definierte Vokabular (Klassen, Relationen, Attribute und Instanzen) dem CBR-System vorgegeben, sodass Fallstrukturen definiert und Berechnungen von Ähnlichkeiten zwischen Fällen durchgeführt werden können.

Die Struktur von Fällen in einem CBR-System kann sowohl homogen als auch heterogen sein.¹⁴⁴ Besitzt jeder Fall im CBR-System dieselben Relationen und Klassen, so handelt es sich um eine homogene Fallstruktur.¹⁴⁵ Hierbei ist jedoch erforderlich zu wissen, welche Relationen und Klassen für die Fallbeschreibung relevant sind. Eine heterogene Fallstruktur liegt vor, wenn die vorgegebene

140) Vgl. BEIBEL (2011), S. 37. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

141) Vorgaben bezüglich des Budgets und der zeitlichen Terminierung eines Projektes können Abweichungen enthalten, wodurch eine exakte und vollständige Formulierung dieser Projekteigenschaften ungenau oder unsicher werden kann.

142) Das Vorgehen des Case-based Reasonings umfasst die Berechnung von Ähnlichkeiten zwischen Fällen. Je geringer die Übereinstimmung von Fällen, desto weniger aussagekräftig kann die angepasste Problemlösung sein.

143) Pro Zeiteinheit können mehr ähnliche Fälle gefunden werden als dies bei einer manuellen Suche möglich ist; vgl. BEIBEL (2011), S. 39.

144) Vgl. WATSON (2003), S. 27 f. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

145) WATSON verwendet das Beispiel eines Maklerbüros. Jedes Haus in der Fallbasis hat dieselbe Struktur, d. h., es werden z. B. die gleichen 50 Attribute für jedes Haus festgelegt. Es ist relativ einfach, einen vollständigen Satz an Attributen aufzustellen, da alle Immobilienmakler ähnliche Aufzeichnungen über die Objekte besitzen. Verfügt ein Makler über keine Immobiliendatenbank, so wäre es einfach, sie anhand der Attribute zu erstellen. Die relevante Menge an Attributen, die ein Haus „vollständig“ beschreiben, lässt sich auf eine „überblickbare“ Anzahl begrenzen.

Fallstruktur das verwendete Vokabular vorgibt.¹⁴⁶ Die Konkretisierungen von Klassen, Relationen, Attributen und Instanzen hängen vom jeweiligen Sachverhalt ab.

Eine Anforderungsanalyse für das ontologiegestützte Case-based Reasoning enthält neben den Anforderungen an die Hauptbestandteile (Ontologien und Case-based Reasoning) einige weitere Anforderungen¹⁴⁷, die an das gesamte CBR-System gerichtet sind, um den Erfolg von Projekten mittels des Einsatzes von CBR-Systemen zu steigern. Die *Verfügbarkeit* eines gemeinsamen Verständnisses der Realität wird mit Hilfe eines einheitlich strukturierten Vokabulars ermöglicht. Je umfangreicher die zugrunde liegende Domänenontologie ist, desto umfassender beläuft sich die Konzeptualisierung der relevanten Begrifflichkeiten. Eine weitere Anforderung stellt die *Agilität* des Systems dar, die dem Nutzer erlaubt, Weiterentwicklungen und Modifizierungen in das CBR-System einfließen zu lassen. Damit einhergehend ist die Anforderung der *Intuitivität* und *Verständlichkeit*. Der Verzicht auf eine komplexe Benutzeroberfläche verringert den Schulungsaufwand der Anwender und ermöglicht gegebenenfalls eine schnelle Einarbeitung und eine einfache Wartung und Pflege des CBR-Systems. Neben der Übersichtlichkeit der Anwendung stellt das *Antwortzeitverhalten* eine weitere Anforderung dar, die es zu berücksichtigen gilt. Mit steigender Komplexität des Systems steigt auch der Rechenaufwand im Fall einer Anfrage an das CBR-System (CBR-Anfrage). Um eine möglichst hohe Benutzerakzeptanz für den betrieblichen Einsatz von ontologiegestütztem Case-based Reasoning zu erreichen, sollten die Antwortzeiten des CBR-Systems möglichst gering gehalten werden.¹⁴⁸ Ein ebenfalls relevanter Aspekt ist die *Plattformunabhängigkeit*. Der Einsatz einer Software sollte mit unterschiedlichen Betriebssystemen¹⁴⁹ möglich sein. Zudem ist die Grundlage eines ontologiegestützten CBR-Systems eine Domänenontologie, die innerhalb eines externen Ontologie-Editors erstellt wird. Weiterhin besteht die Notwendigkeit der Auswahl eines zentralen Ablageorts der verwendeten Datenbank¹⁵⁰, sodass das gebündelte Know-how des CBR-Systems jeder berechtigten Person zur Verfügung steht.

2.4.2 Ontologiegestützte Ähnlichkeitsbestimmung

2.4.2.1 Ähnlichkeitsalgorithmus nach BEIBEL

Die Berechnung von Ähnlichkeiten im Case-based Reasoning erfolgt mittels Ähnlichkeitsalgorithmen. Die Ähnlichkeitsberechnung basiert auf verschiedene Ähnlichkeitsindikatoren.¹⁵¹ Das Ziel dieser Vorgehensweise ist die Berechnung eines konkreten numerischen Werts (Ähnlichkeitswert) für die Ähnlichkeit zwischen zwei miteinander verglichenen Fällen.¹⁵²

146) WATSON verwendet das Beispiel einer Patientendiagnose. Die Patientenakten enthalten diverse Informationen zum Alter, zur Blutgruppe etc., aber auch patientenspezifische Informationen, die für jede Patientenakte einzigartig sind (Behandlungen, Prognosen etc.). Fälle können nicht problemlos aus einer Datenbank übernommen werden. Entwickler einer Fallbasis können sich nie sicher sein, alle relevanten Attribute erfasst zu haben.

147) Die Spezifikation von Anforderungen erfolgt mittels einer Auswahl an Qualitätskriterien, die von Anforderungen erfüllt werden sollten; vgl. RUPP (2014), S. 26 ff.; ISO/IEC/IEEE 29148 (2011), S. 14; Nagel (2009), S. 4 f.

148) Dies erfordert zunächst eine Praxiserprobung.

149) Zum Teil handelt es sich um veraltete Betriebssysteme, bei denen die Software dennoch funktionieren sollte.

150) Die Datenbank enthält das gesamte gebündelte Wissen und kann auch als „Wissensbank“, „Wissensdatenbank“, „Falldatenbank“ oder „Fallbasis“ bezeichnet werden.

151) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 492; BEIBEL (2011), S. 159.

152) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 492.

Gemäß BEIBELS Ähnlichkeitsalgorithmus existieren Ähnlichkeitsindikatoren für Symbole¹⁵³, numerische Werte¹⁵⁴, Klassen¹⁵⁵ und Elemente aus Taxonomien¹⁵⁶.

Ähnlichkeiten zwischen Symbolen¹⁵⁷ können anhand von Tabellen bestimmt werden.¹⁵⁸ Bei numerischen Werten wird der „Abstand“ zwischen Attributwerten als Ähnlichkeitsmaßstab verwendet.¹⁵⁹ Ähnlichkeiten zwischen Klassen werden anhand der Anzahl gemeinsamer Attribute und anhand des Abstandes zwischen Instanzen errechnet.¹⁶⁰ Je höher die Anzahl an gemeinsamen Attributen zweier Klassen und je kleiner der Abstand zwischen den Instanzen dieser Klassen ist, desto größer werden die Ähnlichkeitswerte der verglichenen Klassen ausfallen. Besteht der Ähnlichkeitsmaßstab aus einer Taxonomie, so wird die Ähnlichkeit zwischen Attributwerten anhand des Ähnlichkeitswerts innerhalb der Taxonomie bestimmt.¹⁶¹ Die Ähnlichkeitswerte sind hierbei an den Knoten der Taxonomie festzustellen, die untergeordnete Knoten besitzen.

In BEIBELS Ähnlichkeitsalgorithmus umfassen die Kardinalitäten für Relationen und Attribute lediglich die Werte 0 und 1.¹⁶² Ein Sachverhalt, der mehrere Relationswerte abbildet, setzt jedoch eine Kardinalität von größer als 1 voraus.¹⁶³

2.4.2.2 Rekursiver Ähnlichkeitsalgorithmus nach BERGENRODT/KOWALSKI

Innerhalb des Verbundprojekts OrGoLo wurde ein erweiterter Algorithmus zur Ermittlung von Ähnlichkeiten von BERGENRODT/KOWALSKI konzipiert, der als Grundlage dieses Projektberichts verwendet wird. Der erweiterte Ähnlichkeitsalgorithmus besteht aus den Komponenten der Klassenähnlichkeit¹⁶⁴ und der Instanzenähnlichkeit¹⁶⁵.

Zur Verdeutlichung der Zusammenhänge erfolgt in der Abbildung 5 auf der nächsten Seite ein Ausschnitt aus der Projektmanagement-Ontologie, der nur die taxonomische Relation „ist_ein“ berücksichtigt:

153) Der Begriff „Symbol“ umfasst gemäß BEIBEL (2011), S. 159 f., Zeichenketten.

154) Der Ausdruck „numerische Werte“ bezeichnet Zahlenwerte; vgl. BEIBEL (2011), S. 159.

155) Vgl. BEIBEL (2011), S. 162 ff.

156) Vgl. BEIBEL (2011), S. 160. Unter „Elemente aus Taxonomien“ wird eine monohierarchische Struktur einer Klasse zu ihrer Oberklasse verstanden; vgl. BEIBEL (2011), S. 178.

157) Hierunter sind auch Zeichenketten zu verstehen.

158) Vgl. BEIBEL (2011), S. 159 f.

159) Vgl. BEIBEL (2011), S. 159.

160) Vgl. BEIBEL (2011), S. 162. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

161) Vgl. BEIBEL (2011), S. 160 f. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

162) Vgl. BEIBEL (2011), S. 152 f.

163) Vgl. BERGENRODT/KOWALSKI (2015), S. 26.

164) BERGENRODT/KOWALSKI (2015), S. 27, verwenden in diesen Zusammenhang den Begriff „Konzeptähnlichkeit“.

165) Vgl. BERGENRODT/KOWALSKI (2015), S. 31. Die Berechnung der Instanzenähnlichkeit umfasst die Ermittlung der Ähnlichkeiten von Klasseigenschaften (Relationen und Attribute). BERGENRODT/KOWALSKI (2015), S. 28, verwenden in diesem Zusammenhang die Begriffe „Konzeptrelationen“ und „Konzeptattribute“.

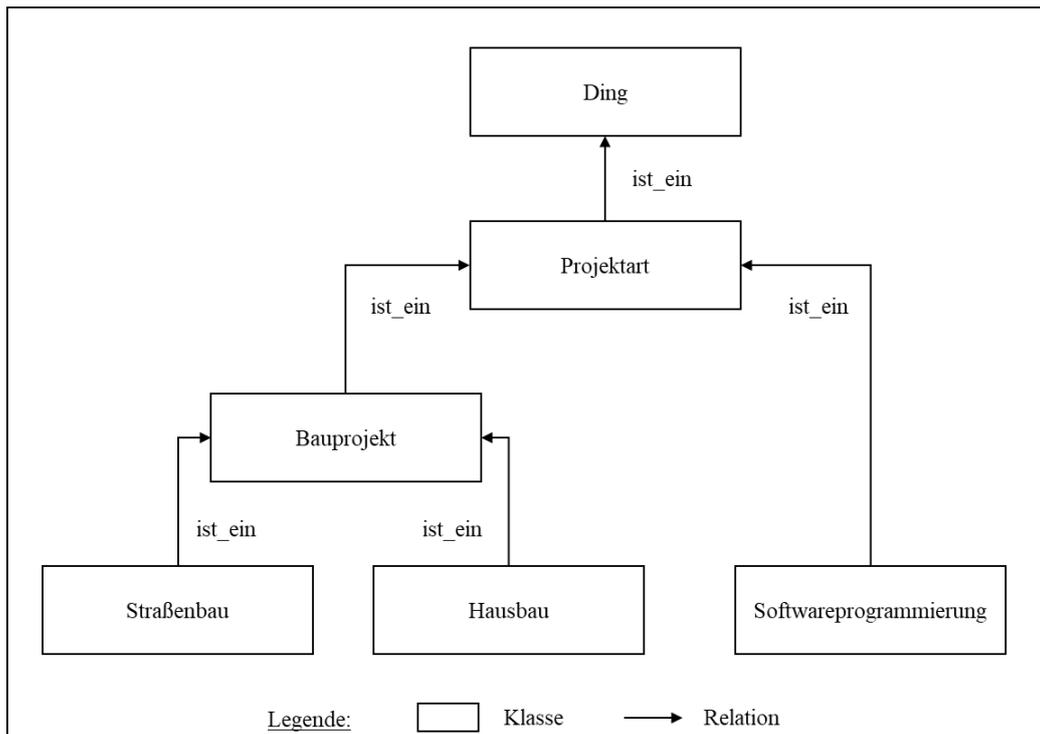


Abbildung 5: Ausschnitt aus einer Domänenontologie

Im weiteren Verlauf der hier vorgelegten Überlegungen besteht zum einen die Annahme, dass eine vollständig explizierte Ontologie vorliegt.¹⁶⁶ Zum anderen wird vorausgesetzt, dass ein Taxonomie-Graph¹⁶⁷ als Basis für die ontologiebasierte Ermittlung von Ähnlichkeiten vorliegt.¹⁶⁸

Ähnlichkeit von Klassen: Unter dem Begriff Klassenähnlichkeit wird die Ähnlichkeit zweier Klassen k_a und k_b verstanden, die Teil derselben Ontologie sind.¹⁶⁹ Zur Berechnung der Klassenähnlichkeit bedarf es einiger Hilfsfunktionen.

Zunächst wird die Hilfsgröße „lcs“ (Least Common Subsumer) eingeführt.¹⁷⁰ Sie bildet die tiefste Klasse (Minimaleigenschaft¹⁷¹) innerhalb der Ontologie, die sowohl die Klasse k_a als auch die Klasse k_b subsumiert (Subsumtionseigenschaft¹⁷²), zu der also die beiden vorgenannten Klassen als Unterklassen gehören.¹⁷³ Die Hilfsgröße lcs ermittelt diejenige Klasse k_{lcs} , die sowohl die Minimal- als auch die Subsumtionseigenschaft erfüllt, sodass gilt: $lcs(k_a, k_b) = k_{lcs}$.

166) D. h., dass alle Relationen (insbesondere Subsumtions-Relationen) und geerbte Attribute bekannt sind (Vererbungsprinzip); vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 494.

167) Ein Taxonomie-Graph vereinigt das gesamte taxonomische Wissen über alle Klassen und die taxonomische Relation „ist_ein“ der zugrunde liegenden Projektmanagement-Ontologie. Zur Ermittlung von Ähnlichkeiten zwischen Fällen werden die mathematisch definierten Konstrukte des Taxonomie-Graphen als zentraler Ansatzpunkt betrachtet. Ziel ist die „Berechnung“ der Ähnlichkeit zwischen natürlichsprachlich spezifizierten Klassen auf formal-sprachliche Weise; vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 494.

168) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 494.

169) Vgl. BERGENRODT/KOWALSKI (2015), S. 27.

170) Vgl. BERGENRODT/KOWALSKI (2015), S. 27 f. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

171) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 495.

172) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 495.

173) In Anlehnung an RECIO-GARCÍA et al. (2007), S. 153 f.

Der Pfad $\text{pfad}(k_x, k_y)$ zwischen zwei Klassen k_a und k_b gibt die kleinstmögliche Menge der gemeinsamen Klassen an, die im Taxonomie-Graph einer Ontologie auf einem Weg zwischen den Klassen k_x nach k_y erreicht werden können.¹⁷⁴

Die Distanz $\text{dist}(k_a, k_b)$ zweier Klassen k_a und k_b ist der längste derjenigen Pfade, welche die Klasse k_a oder k_b mit der Klasse k_{lcs} ihres gemeinsamen Least Common Subsumers verbinden:¹⁷⁵

$$\text{dist}(k_a, k_b) = \max(|\text{pfad}(k_a, \text{lcs}(k_a, k_b))|, |\text{pfad}(k_b, \text{lcs}(k_a, k_b))|)$$

Beispiel: Der Least Common Subsumer der beiden Klassen „Softwareprogrammierung“ (k_a) und „Hausbau“ (k_b) ist mithilfe der Funktion $\text{lcs}(\text{Softwareprogrammierung}, \text{Hausbau})$ als „Projektart“ definiert. Der Pfad $\text{pfad}(\text{Hausbau}, \text{Projektart})$ zwischen den Klassen „Hausbau“ und „Projektart“ lautet $\{\text{Hausbau}, \text{Bauprojekt}, \text{Projektart}\}$. Die Distanz $\text{dist}(\text{Softwareprogrammierung}, \text{Hausbau})$ lässt sich wie folgt ermitteln:

$$\begin{aligned} \text{dist}(k_a, k_b) &= \max(|\text{pfad}(k_a, \text{lcs}(k_a, k_b))|, |\text{pfad}(k_b, \text{lcs}(k_a, k_b))|) \\ &= \max\left(|\text{pfad}(\text{Softwareprogrammierung}; \text{lcs}(\text{Softwareprogrammierung}, \text{Hausbau}))|, |\text{pfad}(\text{Hausbau}, \text{lcs}(\text{Softwareprogrammierung}, \text{Hausbau}))|\right) \\ &= \max(|\text{pfad}(\text{Softwareprogrammierung}, \text{Projektart})|, |\text{pfad}(\text{Hausbau}, \text{Projektart})|) \\ &= \max(|\{\text{Softwareprogrammierung}, \text{Projektart}\}|, |\{\text{Hausbau}, \text{Bauprojekt}, \text{Projektart}\}|) \\ &= \max(2; 3) \\ &= 3 \end{aligned}$$

In einem weiteren Schritt wird die Menge der Klassenrelationen und -attribute $\text{kr}(k_a)$ ermittelt.¹⁷⁶ Diesbezüglich werden diejenigen Relationen und Attribute herangezogen, deren Vorbereiche die Klasse k_a oder eine k_a übergeordnete Klasse sind.¹⁷⁷

Beispiel: In der zuvor dargestellten Ontologie (vgl. die o. a. Abbildung 5) sind folgende Relation und folgendes Attribut definiert:

Relation	Attribut	Vorbereich der Relation/ des Attributs
hatEntscheidungsträger		Projektart
	AnzahlFenster	Hausbau

Tabelle 1: Exemplarische Klassenrelationen und -attribute

Die Menge der Klassenrelationen und -attribute für die Klasse „Hausbau“ wird in diesem Beispiel definiert als

$$\text{kr}(\text{Hausbau}) = \{\text{hatEntscheidungsträger}, \text{AnzahlFenster}\}$$

weil das Attribut „AnzahlFenster“ die Klasse „Hausbau“ und die Relation „hatEntscheidungsträger“ die Klasse „Projektart“ als Vorbereich hat.

174) Vgl. BERGENRODT/KOWALSKI (2015), S. 27 f.

175) Vgl. BERGENRODT/KOWALSKI (2015), S. 28; RICKLEFS/BLOMQVIST (2008), S. 1240 f.

176) Vgl. BERGENRODT/KOWALSKI (2015), S. 28. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

177) Die Relation „ist_ein“ wird zur Ermittlung der Menge betrachtet.

Zur finalen Berechnung der Klassenähnlichkeit wird die Funktion $\text{ksim}(k_a, k_b)$ verwendet, die wie folgt definiert ist:¹⁷⁸

$$\text{ksim}(k_a, k_b) = \begin{cases} 1,0 & \text{wenn } k_a = k_b \\ \frac{1}{\text{dist}(k_a, k_b)} * \begin{cases} 1,0, & \text{wenn } |\text{kr}(k_a) \cup \text{kr}(k_b)| = 0 \\ 0,0, & \text{wenn } |\text{kr}(k_a)| = 0 \text{ oder } |\text{kr}(k_b)| = 0 \text{ und } |\text{kr}(k_a) \cup \text{kr}(k_b)| > 0 \\ \frac{|\text{kr}(k_a) \cap \text{kr}(k_b)|}{|\text{kr}(k_a) \cup \text{kr}(k_b)|}, & \text{sonst} \end{cases} \end{cases}$$

Grundsätzlich erfolgt die Bestimmung der Ähnlichkeit zweier Klassen in zwei Schritten.¹⁷⁹ Zunächst wird zur Berechnung der Klassenähnlichkeit $\text{ksim}(k_a, k_b)$ die Distanz $\text{dist}(k_a, k_b)$ der zu vergleichenden Klassen berechnet. Anschließend wird der reziproke Wert dieser Klassenähnlichkeit ($\frac{1}{\text{dist}(k_a, k_b)}$) gebildet. Im zweiten Schritt erfolgt ein Vergleich der jeweiligen Relationen und Attribute. Hierzu teilt man die Anzahl der Relationen und Attribute, die für beide Klassen gemeinsam definiert sind, durch die Anzahl der Relationen und Attribute, die nur für die eine oder andere Klasse definiert sind. Sind beide Klassen gleich, entspricht die Ähnlichkeit dem Wert 1,0.

Beispiel: Die Bestimmung der Ähnlichkeit der Klassen „Softwareprogrammierung“ und „Hausbau“ sieht wie folgt aus:

$$\frac{1}{\text{dist}(\text{Softwareprogrammierung}, \text{Hausbau})} = \frac{1}{3}$$

$$\frac{|\text{kr}(k_a) \cap \text{kr}(k_b)|}{|\text{kr}(k_a) \cup \text{kr}(k_b)|} = \frac{|\{\text{hatEntscheidungsträger}\}|}{|\{\text{hatEntscheidungsträger}, \text{AnzahlFenster}\}|} = \frac{1}{2}$$

$$\text{ksim}(\text{Softwareprogrammierung}, \text{Hausbau}) = \frac{1}{3} * \frac{1}{2} = \frac{1}{6}$$

Ähnlichkeit von Instanzen: Die Ähnlichkeit von Instanzen (Instanzenähnlichkeit)¹⁸⁰ setzt sich aus der Klassenähnlichkeit derjenigen Klassen, die von den beiden Instanzen instanziiert werden, und der Ähnlichkeit der für die betrachteten Instanzen definierten Eigenschaften¹⁸¹ (Relationen und Attribute) zusammen.¹⁸²

Für die Berechnung der Ähnlichkeit von Instanzen werden spezifische Ähnlichkeitsfunktionen sim_n des Typs $n \in \mathbb{N}$ verwendet, welche die Ähnlichkeit zwischen zwei Attributs- oder Relationswerten a und b einer Instanzeigenschaft r ermitteln.¹⁸³

Die Ähnlichkeitsfunktionen $\text{sim}_n(r, a, b)$ sollte folgenden Eigenschaften erfüllen:

$$\begin{aligned} \text{sim}_n(r, a, b) &\in [0; 1] \\ \text{sim}_n(r, a, a) &= 1,0 \\ \text{sim}_n(r, a, b) &= \text{sim}_n(r, b, a) \end{aligned}$$

178) Vgl. BERGENRODT/KOWALSKI (2015), S. 28 f.

179) Vgl. BERGENRODT/KOWALSKI (2015), S. 29. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

180) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 500.

181) Bei der Ermittlung der Instanzenähnlichkeit wird nicht zwischen Relationen und Attributen unterschieden, vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 501.

182) Vgl. BERGENRODT/KOWALSKI (2015), S. 29.

183) Vgl. BERGENRODT/KOWALSKI (2015), S. 30.

Die erste Eigenschaft einer Ähnlichkeitsfunktion besagt, dass die Ähnlichkeit zweier Werte a und b im Intervall $[0;1]$ ausgedrückt wird. Die zweite und die dritte Eigenschaft bezeichnen die Reflexivität und die Symmetrie einer Ähnlichkeitsfunktion.¹⁸⁴

Zur Berechnung der Instanzähnlichkeit werden weitere Hilfsfunktionen benötigt. Die Funktion $R(i, n)$ gibt die Menge aller auf einer Instanz i definierten Attribute und Relationen wieder, deren Ähnlichkeitsbestimmung mit Hilfe einer Ähnlichkeitsfunktion $n \in N$ erfolgt.¹⁸⁵

Eine weitere Funktion $R(i)$ gibt die Menge aller auf einer Instanz i definierten Attribute und Relationen wieder, für die eine Ähnlichkeitsfunktion existiert.¹⁸⁶ Alle weiteren Klasseneigenschaften, für die keine Ähnlichkeitsfunktion definiert ist, nehmen keinen Einfluss auf die Ähnlichkeitsberechnung.

Zudem ist die Funktion $wert(i,r)$ erforderlich, welche die Menge der Werte im Nachbereich eines Attributs oder einer Relation r für die Instanz i wiedergibt.¹⁸⁷

Beispiel: Für die Klasse „Hausbau“ ist die Instanz „BobsHausbau“ mit den Relationselementen „BobsHausbau hatEntscheidungsträger Bob“ und „BobsHausbau hatEntscheidungsträger Petra“ definiert. $|N|$ ist hier gleich 1, da es nur eine Relation „hatEntscheidungsträger“ gibt und somit nur eine Ähnlichkeitsfunktion zur Messung der Ähnlichkeit hinsichtlich dieser Relation existiert: In diesem Fall nimmt die Funktion $R(i, n)$ für $R(\text{BobsHausbau}, \text{sim}_1)$ als Funktionswert die Menge $\{\text{hatEntscheidungsträger}\}$ an. Für die Funktion $wert(i,r)$ gilt, dass der Funktionswert $wert(\text{BobsHausbau}, \text{hatEntscheidungsträger})$ die Menge $\{\text{Bob}, \text{Petra}\}$ ist.

Die kumulierte Ähnlichkeit von Relationen und Attributen einer Menge R von zwei zu vergleichenden Instanzen i_a und i_b lässt sich wie folgt berechnen:¹⁸⁸

$$\text{relsim}(i_a, i_b, R, n) = \sum_{r \in R} \left(w_r * \frac{\sum_{a \in \text{wert}(i_a, r)} \max_{b \in \text{wert}(i_b, r)} (\text{sim}_n(r, a, b)) + \sum_{b \in \text{wert}(i_b, r)} \max_{a \in \text{wert}(i_a, r)} (\text{sim}_n(r, b, a))}{|\text{wert}(i_a, r)| + |\text{wert}(i_b, r)|} \right)$$

Der Term w_r bezeichnet die Gewichtung für das Attribut oder die Relation r und sim_n definiert die entsprechende Ähnlichkeitsfunktion vom Typ n . Die obige Formel sagt somit aus, dass die Elemente der beiden Wertemengen jeweils für beide Instanzen verglichen werden. Dafür wird jedes r aus der ersten Wertemenge ($\text{wert}(i_a, r)$) der Instanz i_a mit der zweiten Wertemenge ($\text{wert}(i_b, r)$) der Instanz i_b hinsichtlich des ähnlichsten Elements verglichen und umgekehrt.¹⁸⁹ Der daraus resultierende ungewichtete Ähnlichkeitswert für r ist schließlich der Durchschnitt der ermittelten Werte.¹⁹⁰

Zur finalen Berechnung der Instanzähnlichkeit zweier Instanzen i_a und i_b wird eine weitere Hilfsfunktion $K(i)$ benötigt, die auf die Klasse verweist, zu der eine Instanz gehört.¹⁹¹

Beispiel: In der zuvor definierten Ontologie gehört zur Klasse „Hausbau“ die Instanz „BobsHausbau“. Folglich weist $K(\text{BobsHausbau})$ den Funktionswert „Hausbau“ auf.

184) Vgl. RICHTER (2008), 29-30.

185) Vgl. BERGENRODT/KOWALSKI (2015), S. 30.

186) Vgl. BERGENRODT/KOWALSKI (2015), S. 29 f. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

187) Vgl. BERGENRODT/KOWALSKI (2015), S. 30.

188) Vgl. BERGENRODT/KOWALSKI (2015), S. 30.

189) Vgl. BERGENRODT/KOWALSKI (2015), S. 31. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

190) Durch die Berechnung der Ähnlichkeit in beide Richtungen ist die Symmetrie der Funktion gegeben, d. h., es gilt $\text{relsim}(i_a, i_b, R, n) = \text{relsim}(i_b, i_a, R, n)$; vgl. FINNIE/SUN (2002), S. 274 f.

191) BERGENRODT/KOWALSKI (2015), S. 31.

Mit Hilfe der Ähnlichkeitsfunktion $\text{sim}_{\text{instanz}}$ ¹⁹² lässt sich die Instanzenähnlichkeit, die aus der Klassenähnlichkeit und der Ähnlichkeit der für Instanzen definierten Attribute und Relationen r (Relations- und Attributsähnlichkeit)¹⁹³ zusammengesetzt ist, wie folgt definieren:¹⁹⁴

$$\begin{aligned} & \text{sim}_{\text{instanz}}(i_a, i_b) \\ &= \text{ksim}(K(i_a), K(i_b)) * \begin{cases} 1, & \text{wenn } |R(i_a) \cup R(i_b)| = 0 \\ 0, & \text{wenn } |R(i_a)| = 0 \text{ oder } |R(i_b)| = 0 \text{ und } |R(i_a) \cup R(i_b)| > 0 \\ \frac{\sum_{n \in N} \text{re} \text{sim}(i_a, i_b, R(i_a, n) \cap R(i_b, n), n)}{\sum_{r \in R(i_a) \cup R(i_b)} w_r}, & \text{sonst} \end{cases} \end{aligned}$$

Zur Berechnung der Relations- und Attributsähnlichkeit werden zwei Sonderfälle unterschieden:¹⁹⁵ Im ersten Fall existieren bei beiden zu vergleichenden Instanzen weder Relationen noch Attribute.¹⁹⁶ Im zweiten Fall sind die Relationen und Attribute nur für eine der zu vergleichenden Instanzen definiert.¹⁹⁷

Beispiel: Wie bereits zuvor definiert, verfügt die Klasse „Hausbau“ über die Instanz „BobsHausbau“. Darüber hinaus existiert eine Instanz „Regaloptimierung“, die zu Klasse „Softwareprogrammierung“ gehört. Zudem sei eine weitere Klasse „Person“ mit den Instanzen „Bob“ und „Petra“ gegeben. Die Instanz „BobsHausbau“ gehört zu den Vorbereichen der Relationselemente „BobsHausbau hatEntscheidungsträger Bob“ und „BobsHausbau hatEntscheidungsträger Petra“. Die Instanz „Regaloptimierung“ verfügt über das Relationselement „Regaloptimierung hatEntscheidungsträger Petra“.

Die Ähnlichkeit der beiden Instanzen „BobsHausbau“ und „Regaloptimierung“ wird im Folgenden berechnet. Die Gewichte für alle Attribute und Relationen betragen jeweils 1,0.¹⁹⁸

Da Relationen und Attribute nur im Nachbereich vorhanden sind, ergibt sich $N = \{\text{sim}_{\text{instanz}}\}$.

Die Klassenähnlichkeit ksim lässt sich wie folgt ermitteln:

$$\begin{aligned} & \text{ksim}(K(\text{BobsHausbau}), K(\text{Regaloptimierung})) \\ &= \text{ksim}(\text{Hausbau}, \text{Softwareprogrammierung}) \\ &= \frac{1}{6} \end{aligned}$$

Die Ähnlichkeit der Attribute und Relationen der Instanzen „BobsHausbau“ (i_a) und „Regaloptimierung“ (i_b) ergibt sich wie folgt:

$$\frac{\sum_{n \in N} \text{re} \text{sim}(i_a, i_b, R(i_a, n) \cap R(i_b, n), n)}{\sum_{r \in R(i_a) \cup R(i_b)} w_r} = \frac{\text{re} \text{sim}(i_a, i_b, \{\text{hatEntscheidungsträger}\}, \text{sim}_{\text{instanz}})}{w_{\text{hatEntscheidungsträger}} + w_{\text{AnzahlFenster}}}$$

192) Weitere zur Ähnlichkeitsberechnung verfügbare Ähnlichkeitsfunktionen, wie z. B. $\text{sim}_{\text{numerisch}}$, $\text{sim}_{\text{taric}}$ und sim_{bool} , werden im Folgenden nicht berücksichtigt. Die Ähnlichkeitsfunktion $\text{sim}_{\text{numerisch}}$ betrachtet im Vergleich zu $\text{sim}_{\text{instanz}}$ numerische Werte (Euro, Kilogramm etc.). Die Ähnlichkeitsfunktion $\text{sim}_{\text{taric}}$ wird bei Zolltarifnummern verwendet und sim_{bool} wird im Zusammenhang mit Booleschen Werten (Ja/Nein) angewandt; vgl. BERGENRODT/KOWALSKI (2015), S. 32 ff. u. 58.

193) Vgl. BERGENRODT/KOWALSKI (2015), S. 31.

194) Vgl. BERGENRODT/KOWALSKI (2015), S. 31.

195) Vgl. BERGENRODT/KOWALSKI (2015), S. 31. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

196) Die Vereinigungsmenge der Relationen und Attribute beider zu vergleichender Instanzen entspricht der leeren Menge. Die Ähnlichkeit der Relationen und Attribute wird in diesem Fall auf den Wert 1,0 festgelegt.

197) In diesem Fall entspricht die Ähnlichkeit der Relationen und Attribute dem Wert 0,0.

198) In Anlehnung an BERGENRODT/KOWALSKI (2015), S. 31.

Die kumulierte Ähnlichkeit relsim für die Instanzen i_a und i_b hinsichtlich der Relation „hatEntscheidungsträger“ lässt sich in Bezug zur Ähnlichkeitsfunktion $\text{sim}_{\text{instanz}}$ wie folgt ermitteln:

$$\text{relsim}(i_a, i_b, \{\text{hatEntscheidungsträger}\}, \text{sim}_{\text{instanz}}) = \frac{\max(\text{sim}_{\text{instanz}}(\text{Petra}, \text{Petra})) + \max(\text{sim}_{\text{instanz}}(\text{Bob}, \text{Petra})) + \max(\text{sim}_{\text{instanz}}(\text{Petra}, \text{Petra}); \text{sim}_{\text{instanz}}(\text{Petra}, \text{Bob}))}{|\{\text{Bob}, \text{Petra}\}| + |\{\text{Petra}\}|} \cdot \text{WhatEntscheidungsträger}$$

Zur Berechnung dieses Funktionswerts muss zunächst die Instanzähnlichkeit $\text{sim}_{\text{instanz}}$ rekursiv für die Instanzenpaare „Petra, Petra“ und „Bob, Petra“ berechnet werden. Hierfür werden zwei Annahmen getroffen: Erstens sei der Ähnlichkeitswert von $\text{sim}_{\text{instanz}}(\text{hatEntscheidungsträger}, \text{Petra}, \text{Petra})$ gleich 1,0. Zweitens sei der Ähnlichkeitswert von $\text{sim}_{\text{instanz}}(\text{hatEntscheidungsträger}, \text{Bob}, \text{Petra})$ gleich 0,5. Die kumulierte Relations- und Attributsähnlichkeit ergibt sich dann wie folgt:

$\text{relsim}(\text{BobsHausbau}, \text{Regaloptimierung}, \{\text{hatEntscheidungsträger}\}, \text{sim}_{\text{instanz}})$

$$\begin{aligned} & 1,0 * \frac{\max(\{1,0\}) + \max(\{0,5\}) + \max(\{1,0; 0,5\})}{2 + 1} \\ &= \frac{2,5}{3} \\ &= \frac{5}{6} \end{aligned}$$

Durch Einsetzen dieser Werte in die Ähnlichkeitsfunktion $\text{sim}_{\text{instanz}}$ ergibt sich folgender Ähnlichkeitswert für die Instanzen $i_a = \text{„BobsHausbau“}$ und $i_b = \text{„Regaloptimierung“}$:

$$\text{sim}_{\text{instanz}}(i_a, i_b) = \frac{1}{6} * \frac{5}{1+1} = \frac{1}{6} * \frac{5}{2} \approx 0,0694$$

Die Instanzen $i_a = \text{„BobsHausbau“}$ und $i_b = \text{„Regaloptimierung“}$ weisen folglich hinsichtlich der Relation „hatEntscheidungsträger“ eine Ähnlichkeit von knapp 7 % auf.¹⁹⁹

199) BERGENRODT/KOWALSKI (2015), S. 32 ff., gehen in ihren Berechnungen auf weitere Ähnlichkeitsfunktionen ein. Da diese Ähnlichkeitsfunktionen für die im Kapitel 4.2.2 durchgeführte manuelle Berechnung von Ähnlichkeiten nicht relevant sind, wird auf eine Erläuterung verzichtet.

3 Erstellung einer Projektmanagement-Ontologie mittels des Ontologie-Editors Protégé

3.1 Vorgehensweise bei der Erstellung einer Ontologie

Die Erstellung einer Ontologie soll dem Nutzer ermöglichen, ein gemeinsames Verständnis der sprachlichen Ausdrucksmittel für die Strukturierung und Repräsentation von Wissen mit anderen Interessenten zu teilen.²⁰⁰ Liegt dieses Wissen in Form einer Ontologie vor, so erhalten die Nutzer die Möglichkeit, durch ein ontologiegestütztes CBR-System Wissen über alte, bereits durchgeführte und über neue Projekte darzustellen, insbesondere zu archivieren und auch wiederzuverwenden. Neben der Wiederverwertung von domänenspezifischem Wissen sollten die Nutzer imstande sein, neues Wissen in die Ontologie zu implementieren und bereits vorhandenes Wissen zu modifizieren.²⁰¹

Die Entwicklung einer Ontologie für das Projektmanagement orientiert sich in diesem Projektbericht an den sieben Schritten zur Erstellung einer Ontologie nach NOY/MCGUINESS.²⁰² Der Einsatz dieser Methode wurde bereits in anderen Projekten ausgewählt, da sie für Erstentwickler von Ontologien gut verständlich und leicht anzuwenden ist.²⁰³

Dass es keine Standardmethode zur Erstellung von Ontologien gibt, zeigen GÓMEZ-PÉREZ, CORCHO und FERNÁNDEZ-LÓPEZ.²⁰⁴ Die Wahl der Methode hängt von individuellen und von projektspezifischen Einflussgrößen ab, wie z. B. von den Kompetenzen der Entwickler bzw. dem gewünschten Detaillierungsgrad einer Projekt-Ontologie.²⁰⁵

Unter Verwendung des Ontologie-Editors Protégé²⁰⁶ können Ontologien konstruiert und modifiziert werden.²⁰⁷ Hierbei handelt es sich um ein Software-Tool, das bereits NOY und MCGUINESS in ihrem Vorgehensmodell verwendet haben. Protégé stellt eine ab dem Jahr 2000 an der Universität Stanford entwickelte Open-Source-Software dar.²⁰⁸ Der Ontologie-Editor²⁰⁹ umfasst derzeit eine Anzahl von über 366.000 Benutzern. Mit diesem Software-Tool können Ontologien in der webbasierten Repräsentationssprache OWL erstellt und gegebenenfalls mit dem Plug-in OWLViz visualisiert werden.²¹⁰ Die Ontologien werden in der Form eines Graphen visualisiert, in dem Klassen durch Knoten und Relationen durch Kanten dargestellt werden.²¹¹

200) Vgl. MUSEN (1992), S. 435.

201) Vgl. NOY/MCGUINESS (2001), S. 1 f.

202) Vgl. NOY/MCGUINESS (2001), S. 5 ff.

203) Vgl. KOWALSKI/ZELEWSKI (2015), S. 600; CURTS/CAMPBELL (2005), S. 8; VAN LAARSCHOT (2005), S. 18 ff.

204) Die Autoren bieten eine Übersicht über die vielfältigen Methoden zur Erstellung von Ontologien; vgl. GÓMEZ-PÉREZ/CORCHO/FERNÁNDEZ-LÓPEZ (2004), S. 113 ff. Diese Alternativen werden im Folgenden jedoch nicht näher betrachtet.

205) Vgl. KOWALSKI/ZELEWSKI (2015), S. 601.

206) Die Anwendung dieses Ontologie-Editors wurde vom Institut für Produktion und Industrielles Informationsmanagement der Universität Duisburg-Essen im Hinblick auf die Erstellung einer Domänenontologie als naheliegend empfohlen. Daher wurde davon abgesehen, einen Vergleich zwischen verschiedenen Editoren, z. B. mittels der AHP-Methode, durchzuführen; vgl. stattdessen beispielsweise BEIBEL (2011), S. 49 ff.

207) Der Umgang mit dem Ontologie-Editor wird in diversen Tutorials aufgezeigt. Daneben bietet die Protégé-Website <https://protege.stanford.edu/> (letzter Zugriff am: 12.03.2022) eine Mailing-Liste und ein Wiki an, um den Benutzern bei ihren Problemen helfen zu können.

208) Vgl. NOY/CRUBÉZY/FEGERSON et al. (2003), S. 953.

209) Eine ausführliche Dokumentation zum Ontologie-Editor Protégé befindet sich auf der bereits erwähnten Protégé-Website <https://protege.stanford.edu/>.

210) Vgl. KOWALSKI/ZELEWSKI (2015), S. 601.

211) Vgl. STUCKENSCHMIDT (2011), S. 29 ff.

Aufgrund der Komplexität der modellierten Zusammenhänge im Bereich des Projektmanagements werden in diesem Projektbericht jeweils nur – tabellarische oder grafische – Ausschnitte aus der erstellten Projektmanagement-Ontologie wiedergegeben.

Um formale und konzeptuelle Inkonsistenzen bei der Entwicklung von Ontologien zu vermeiden, ist es zwar streng genommen notwendig, die Ontologie – mit allen ihren Klassen, Instanzen, Attributen und Relationen – in ihrer Gänze zu erfassen.²¹² Mithilfe automatischer Inferenzmechanismen, auch als semantische Reasoner bezeichnet, ist dies jedoch nicht notwendig. Protégé nutzt den Reasoner Fact++, der implizites Wissen durch logische Schlussfolgerungen explizieren und Konsistenzprüfungen durchführen kann. Mit diesem Instrument wird der Entwickler bei der Überprüfung logischer Schlüsse unterstützt und auf Schwachstellen in einer Ontologie hingewiesen.²¹³

3.2 Erstellung der Projektmanagement-Ontologie

Die Erstellung der Projektmanagement-Ontologie beginnt – gemäß den sieben Schritten im Vorgehensmodell von NOY/MCGUINESS – mit der Erstellung eines Grundgerüsts, das im weiteren Verlauf mit Details verdichtet wird.²¹⁴ Der Konstruktionsprozess folgt keiner streng festgelegten Abfolge von Schritten, sondern wird im gesamten Prozess wiederholt durchgeführt. Das mehrfache Durchlaufen der sieben Schritte ermöglicht dem Entwickler, anfängliche Konstruktionsmängel zu beheben. Die erstellte Projektmanagement-Ontologie dient lediglich als Prototyp und kann durch seine Nutzer erweitert und modifiziert werden.

Schritt 1: Festlegung des Anwendungsbereichs

Bei der betrachteten Domäne handelt es sich um das betriebliche Projektmanagement.

Neben der Klasse „Projekt“ soll die Projektmanagement-Ontologie die Klassen „Projektbeschreibung“, „Projektlösung“ und „Projektbewertung“ umfassen.²¹⁵ Die Bündelung dieser Aspekte innerhalb einer Projektmanagement-Ontologie kann für Unternehmen von Vorteil sein, weil sie in der Lage sind, bereits vorhandenes Wissen früherer Projekte auf neue Projekte anzuwenden. Daher erstrecken sich die Anwendungsmöglichkeiten einer Projektmanagement-Ontologie vor allem auf die strukturierte Wissensspeicherung und die systematische Wiederverwendung von (Erfahrungs-)Wissen über Projekte. Mittels der erstellten Projektmanagement-Ontologie soll eine gemeinsame Wissensbasis für das betriebliche Projektmanagement vorbereitet werden, indem die sprachlichen Ausdrucksmittel für die projektbezogene Wissensrepräsentation bereitgestellt werden.

Ein ressourcenschonendes Vorgehen bei der Projektplanung und -durchführung – unter Verwendung der Erfahrungen früherer Projekte – bietet dem Unternehmen selbst und allen seinen Stakeholdern Vorteile. Vergangene Fehler können durch die Wiederverwendung von Erfahrungswissen vermieden und Erfolgsfaktoren aus den bereits durchgeführten Projekten berücksichtigt werden. Die Ontologie kann als ein Kommunikationsinstrument eingesetzt werden und so z. B. das Wissen verschiedener Unternehmensstandorte vereinen. Darüber hinaus verringert dieses Vorgehen den Verlust von Hu-

212) Vgl. STUCKENSCHMIDT (2011), S. 156 f. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

213) Das Aufzeigen von Schwachstellen in einer Ontologie erfolgt – nach dem Starten des Reasoners – mittels einer Fehlermeldung, die jedoch keinen Aufschluss darüber gibt, wo sich der Fehler in der Ontologie befindet. Daher ist es ratsam, den Reasoner nach jeder Veränderung der Ontologie neu zu starten, um so einen möglichen Fehler direkt finden zu können.

214) An dieser Stelle gilt es zu erwähnen, dass es keine „richtige“ Art und Weise zur Erstellung einer Ontologie gibt. Die Anzahl an Alternativen bietet eine Vielzahl an Lösungsmöglichkeiten. Es bedarf einer Abwägung aller Vor- und Nachteile zur Auswahl einer speziellen Vorgehensweise bei der Ontologiekonstruktion, die der Entwickler treffen muss; vgl. NOY/MCGUINESS (2001), S. 4.

215) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 516.

mankapital, weil beim Ausscheiden eines langjährigen Mitarbeiters sein Wissen weiterhin im Unternehmen bestehen bleibt und von neuen Mitarbeitern genutzt werden kann. Die neuen Mitarbeiter müssen nicht erst – gemäß dem Trial-and-Error-Prinzip – ihre eigenen Fehler begehen, sondern können bewusst Entscheidungen anhand von bereits realisierten und bewerteten Vorgehensweisen in früheren Projekten treffen.

Um den Umfang einer Ontologie zu bestimmen, werden zunächst eine Reihe von Kompetenzfragen²¹⁶ formuliert:

- Welche Kompetenzen erfordert ein Projekt?
- Welche Risiken können auftreten?
- In welchem Budgetrahmen bewegen sich Vergleichsprojekte?
- Welche Aspekte sind notwendig, um ein Projekt zu strukturieren?
- Welche Projekte wurden bereits in einem Bereich durchgeführt?
- Welche Aspekte müssen bei einer Projektbeschreibung berücksichtigt werden?
- Welche Akteure sind am Projekt beteiligt?
- Mit welchem Tool kann ein Projekt kontrolliert werden?

Schritt 2: Wiederverwendung bestehender Ontologien

In diesem Schritt geht es um eine Recherche, welche Ontologien innerhalb des Themengebiets des Projektmanagements bereits zur Verfügung stehen.²¹⁷ Für die Suche geeigneter Ontologien wurden diverse Datenbanken und Suchmaschinen eingesetzt. Eine Auflistung der aufgefundenen Ontologien befindet sich im Anhang 1 zu diesem Projektbericht.²¹⁸

Im Rahmen der Ontologierstellung dieses Projektberichts werden unterschiedliche Aspekte des Projektmanagements – unter Berücksichtigung der Recherchen – in einer möglichst umfangreichen Ontologie dargestellt, sodass die Erfassung, Speicherung, Modifikation und Wiederverwendung von (Erfahrungs-)Wissen über Projekte für die Erfüllung von Aufgaben des Projektmanagements genutzt werden können.

Schritt 3: Aufzählung relevanter Begriffe

Dieser Schritt umfasste eine Liste aller relevanten Begriffe der Projektmanagement-Ontologie. Die Aufzählung ist unabhängig davon, ob es sich bei den Begriffen um Klassen, Instanzen, Attribute oder Relationen handelt. Erst in den nächsten Schritten werden diese Begriffe den vorgenannten Kategorien sprachlicher Ausdrucksmittel zugeordnet.

Aufgrund des umfangreichen Spektrums an Begrifflichkeiten, die im betrieblichen Projektmanagement Verwendung finden, beinhaltet die Aufzählung nur eine Auswahl relevanter Begriffe. Die Grundlage dieser Zusammenstellung basiert auf Informationen aus Literaturrecherchen und diversen Interviews mit Experten aus der betrieblichen Praxis.

Im Bereich des Projektmanagements sind beispielsweise folgende Begriffe relevant: „Projektbeschreibung“, „Projektlösung“, „Projektbewertung“, „analytisches Wissen“, „deskriptives Wissen“,

216) Vgl. NOY/MCGUINNESS (2001), S. 5; GRÜNINGER/FOX (1995), S. 3.

217) Der Aufbau der Recherche erfolgt in Anlehnung an DICKE (2007), S. 414 ff.

218) Die im Anhang 1 befindliche Marktrecherche umfasst nicht nur eine umfangreiche Sichtung von wissenschaftlicher Fachliteratur, sondern auch eine ausführliche Suche im Internet.

„evaluatives Wissen“, „konstruktives Wissen“, „organisatorisches Wissen“, „Projekttyp“, „Projektziel“, „Projektaufgabe“, „Stakeholder“, „Umsatz“, „Kosten“, „Kompetenzen“, „Geschäftsprozesse“, „Ressourcen“, „Erfolgsgrößen“ sowie „Projektkennzahlen“.

Schritt 4: Definition der Klassen und der Klassenhierarchie

In diesem Schritt werden Klassen identifiziert, die als Oberbegriffe verstanden werden können. Diese Klassen werden im Folgenden konkretisiert.²¹⁹ Das Ergebnis dieser Konkretisierung ist eine Unterteilung von Klassen in Subklassen und Instanzen, die mittels der taxonomischen „ist_ein“-Relation verbunden werden.²²⁰ Die anschließende Abbildung 6 stellt einen Ausschnitt aus dieser Klassenhierarchie dar, der mittels des Tools OntoGraf erstellt wurde.²²¹

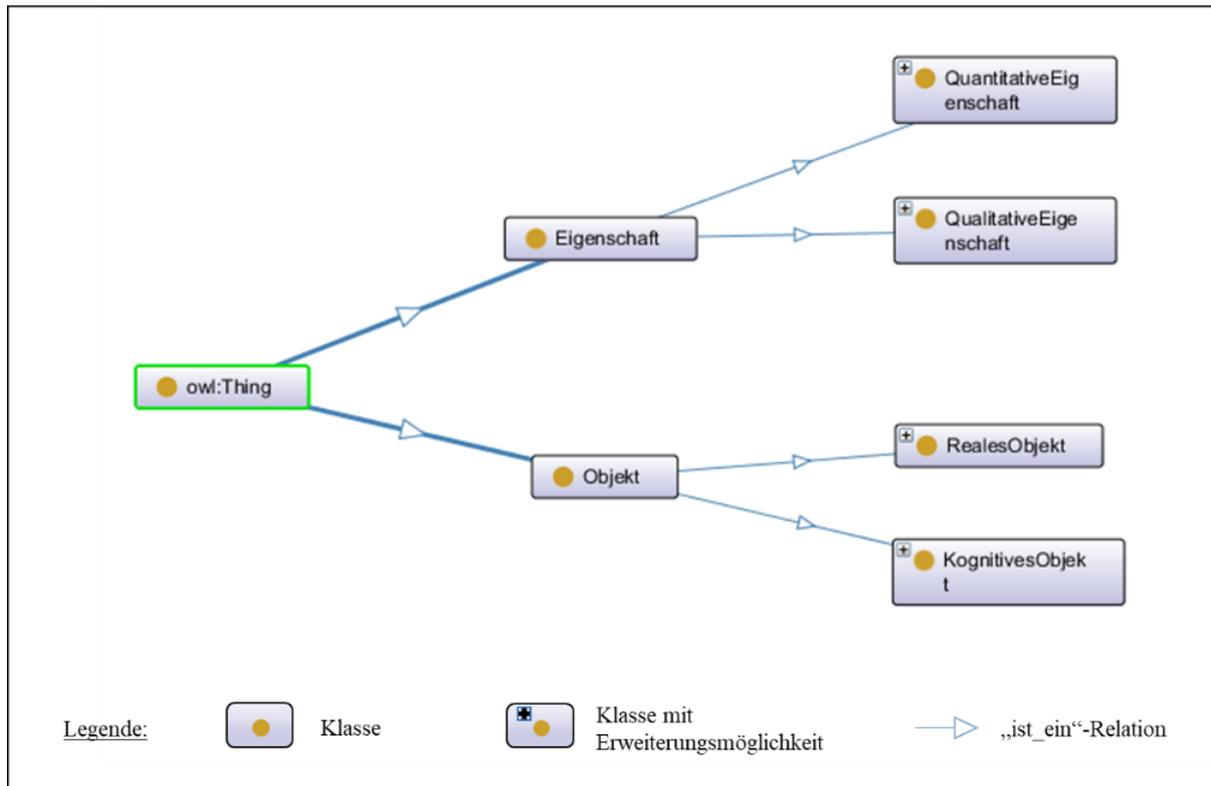


Abbildung 6: Subklassen der Klasse „Thing“²²²

Die oberste (Super-)Klasse wird in der OWL-Klassenhierarchie als „Thing“²²³ bezeichnet und stellt die allgemeinste Klasse dar. Alle weiteren Begriffe, die sich auf den Ebenen unter „Thing“ befinden, sind – gemäß dieser Ontologie – Begriffe aus der Projektmanagementdomäne. Die Klassen „Eigenschaft“ und „Objekt“ sind der Klasse „Thing“ untergeordnet und stellen daher Subklassen der Klasse „Thing“ dar. Die Klasse „RealesObjekt“ bildet wiederum eine Subklasse der Klasse „Objekt“. Betrachtet man beispielsweise die Klasse „Eigenschaft“, so enthält der Nachbereich der „ist_ein“-Relation alle relevanten Elemente, welche die Eigenschaften von Projekten konzeptualisieren. In diesem Fall bilden sowohl die Klasse „QuantitativeEigenschaft“ als auch die Klasse „QualitativeEigenschaft“ Subklassen der Klasse „Eigenschaft“. Die beiden Klassen „Objekt“ und „Eigenschaft“ können

219) Eine eindeutige Trennung zwischen Klassen und Instanzen ist jedoch nicht immer eindeutig möglich; vgl. NOY/MCGUINNESS (2001), S. 18.

220) Vgl. KOWALSKI/ZELEWSKI (2015), S. 604.

221) OntoGraf ist ein in dem Ontologie-Editor Protégé integrierbares Tool zur Darstellung von Ontologien.

222) In der Abbildung 6 wird als Folge des eingesetzten Tools OntoGraf die Richtung der „ist_ein“-Relation „verkehrt herum“ dargestellt.

223) Die Klasse „Thing“ und die Klasse „Ding“ werden in dieser Arbeit synonym verwendet.

nicht als Subklassen voneinander konstruiert werden, da sie beide jeweils separat als grundlegende ontologische Kategorien, die alles Existierende – jede Entität²²⁴ – ihres Gegenstandsbereichs umfassen, betrachtet werden können.²²⁵ Sie sind disjunkt voneinander.

Eine Unterteilung der Subklassen in weitere Subklassen wird in der Darstellungsform der nachfolgenden Abbildung 7 durch die schwarzen Dreiecke neben dem jeweiligen Begriff angedeutet.

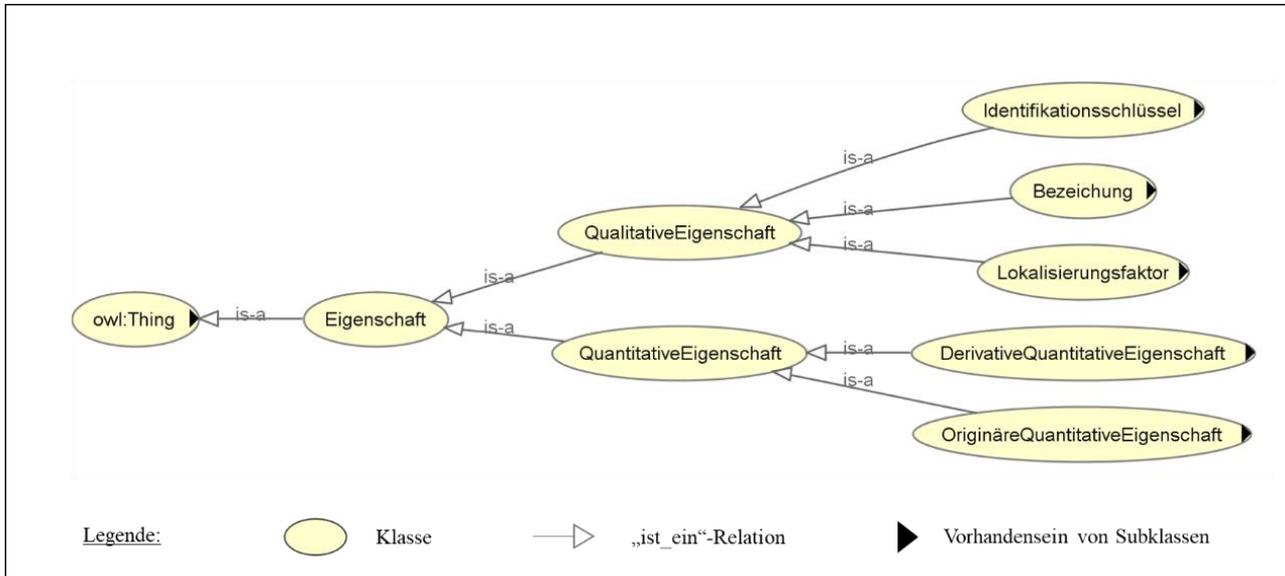


Abbildung 7: Subklassen der Klasse „Eigenschaft“²²⁶

In Protégé gibt es die Möglichkeit, Klassen als disjunkt festzulegen und sie so voneinander abzugrenzen.²²⁷ In diesem Fall kann eine Instanz nur zu genau einer von diesen Klasse gehören. Die Klassen „QualitativeEigenschaft“ und „QuantitativeEigenschaft“ sind disjunkt voneinander, weil eine Instanz nicht gleichzeitig zu beiden Klassen gehören kann. Eine automatische Schlussfolgerungskomponente (Reasoner) kann durch diese Kennzeichnung erkennen, dass Instanzen der Klasse „Eigenschaft“ nur einer dieser beiden disjunkten Klassen zugeordnet werden können.

Schritt 5: Definition der Slots

Nachdem die relevanten Begriffe recherchiert, zu Klassen zugeordnet und hierarchisiert wurden, gilt es die Relationen und Attribute in Bezug auf diese Klassen zu spezifizieren. In Protégé werden die Relationen und Attribute als *Slots*²²⁸ oder *Properties* bezeichnet, die in *Object Properties* für Relationen und *Data Properties* für Attribute unterschieden werden.

Object Properties zeigen Relationen an, die zwischen jeweils zwei Klassen bestehen. Beispiele für *Object Properties* sind: „hat“, „erfordert“, „bestehtAus“ und „betrifft“. Bei der Schreibweise wird auf die Verwendung von Leerzeichen zwischen den Wörtern verzichtet.

Die Relation zwischen den Klassen „Projektbeschreibung“ und „KompetenzSprachen“ lautet beispielsweise „erfordertKompetenzSprachlichPlan“. Die Domain-Klasse drückt den Vorbereitung „Projektbeschreibung“ der Relation aus und die Range-Klasse den Nachbereich „KompetenzSprachen“

224) Vgl. JANSEN (2008), S. 102.

225) Vgl. STUCKENSCHMIDT (2011), S. 11 f.

226) Die Abbildung wurde mittels des Tools OWLViz erstellt. OWLViz ist ein Tool zur Visualisierung von Ontologien, das sich in den Ontologie-Editor Protégé integrieren lässt.

227) Dieser Sachverhalt wird in Protégé als „Disjoint Classes“ bezeichnet.

228) Der Begriff „Slot“ wird im Ontologie-Editor Protégé ab der Version 5 nicht mehr verwendet.

der Relation.²²⁹ Die Darstellungsform von Relationen beginnt in der Regel mit einem kleingeschriebenen Verb, unter Umständen gefolgt von einem (oder mehreren) Substantiv(en) als Hinweis(en) auf die Klassen aus den Nachbereichen der Relationen.²³⁰ Häufig werden auch mehrere Wörter verwendet, um eine Relation auszudrücken.

Data Properties sind in OWL vordefinierte Datentypen, wie z. B. *Integer*, *Symbol*, *Float*, *String*, *Instance* und *Class*.²³¹ Sie werden benötigt, um die einzelnen Attributausprägungen der Instanzen von Klassen zu spezifizieren. In der Projektmanagement-Ontologie können Attribute beispielsweise Währungen (Euro, Dollar), Umsatzwerte (1 Million, 500 Tausend) oder auch Techniken zur Risikoidentifikation (Erfahrungssichtung, Risikochecklisten) darstellen.

Schritt 6: Definition der Sloteseigenschaften

Die Eigenschaften von Slots lassen sich nicht nur durch Relationen und Attribute beschreiben, sondern auch durch slotspezifische Eigenschaften, die in Protégé als *Property Characteristics* bezeichnet werden. Hierzu gehören unter anderem Eigenschaften der Symmetrie, der Transitivität und der Funktionalität.²³² Außerdem können Slots verschiedene Facetten besitzen, mit denen sich diverse Wertetypen und die Kardinalitäten von Slots charakterisieren lassen.²³³

Der Wertetyp gibt an, welche Werte ein Slot annehmen kann.²³⁴ Dazu zählen u. a. die Wertetypen *String*, *Symbol*, *Integer*, *Float*, *Boolean*, *Class* und *Instance*. Der Wertetyp *String* bezeichnet einfache Zeichenketten, die für Attributausprägungen verwendet werden. In der Projektmanagement-Ontologie fallen hierunter z. B. die Buchstaben- und Zahlenkombinationen der Projekt-IDs. Der Wertetyp *Symbol*, der auch als *Enumerated* bezeichnet wird, wird für Attribute mit genau einem Wert verwendet. Beispielsweise kann das Attribut „Region“ der Klasse „Land“ nur jeweils einen festgelegten Wert annehmen. Der Wertetyp *Float* gilt für diejenigen Attribute, deren Werte Dezimalzahlen („Gleitkommazahlen“) darstellen. Darunter fallen z. B. Maßzahlen für Höhe, Breite, Länge und Gewicht. Im Gegensatz zum Wertetyp *Float* beschreibt der Wertetyp *Integer* ganzzahlige Werte. Diejenigen Attribute, die nur die Werte *true* oder *false* annehmen können, werden mit dem Wertetyp *Boolean* gekennzeichnet. Der Wertetyp *Class* wird in der Ontologie für die Relation „ist_ein“²³⁵ verwendet und verknüpft Klassen mit Oberklassen oder eine Instanz mit der jeweiligen Klasse.²³⁶

Die Kardinalität von Slots definiert, wie viele Werte ein Slot haben kann.²³⁷ Dies erstreckt sich von einer Kardinalität mit höchstens einem erlaubten Wert bis hin zu einer Kardinalität mit einer uneingeschränkten Anzahl an Werten. Beispielsweise weist die Relation „hatProjekt-ID-ImEigenenUnternehmen“ hinsichtlich der Klasse „Projektbezeichnung“ in ihrem Nachbereich die Kardinalität 1 auf. Der Grund dieser Bezifferung ist, dass jedes Projekt eine eindeutige Bezeichnung (ID) zur Projektidentifikation besitzt. Ist die Kardinalität mit „0-1“ gekennzeichnet, so kann ein Slot maximal einen Wert annehmen; es kann aber auch keinen Wert („0-Wert“) aufweisen.²³⁸

229) Vgl. NOY/MCGUINNESS (2001), S. 10.

230) Vgl. KOWALSKI/ZELEWSKI (2015), S. 607.

231) Vgl. Schritt 6 des Kapitels 3.2.

232) Vgl. KOWALSKI/ZELEWSKI (2015), S. 608.

233) Der Ontologie-Editor Protégé ermöglicht die Restriktionen „only“, „some“ etc. Diese Restriktionen werden bei der Erstellung der Domänenontologie im Rahmen dieses Projektberichts nicht verwendet, weil sie im Rahmen des gewählten Anwendungsbeispiels für die Implementierung eines ontologiegestützten CBR-Systems nicht erforderlich sind.

234) Vgl. NOY/MCGUINNESS (2001), S. 9 f. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

235) Die Relation „ist_ein“ wird synonym zu der Relation „is_a“ verwendet.

236) Vgl. BEIBEL (2011), S. 147.

237) Vgl. NOY/MCGUINNESS (2001), S. 9. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

238) In Protégé wird dies mit den Kennzeichnungen „max 1“ und „exactly 1“ dargestellt.

Die nachfolgenden Tabelle 2 stellt einen Ausschnitt aus der Charakterisierung der Klasse „Projekt“ in der Projektmanagement-Ontologie dar. Eine umfangreiche Auflistung der Klassen der Projektmanagement-Ontologie mit ihren Slots und Sloteseigenschaften befindet sich im Anhang 3.

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projekt	hatProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	hatProjektbewertung	Projektbewertung	Relation	Instance	1
	hatProjektlösung	Projektlösung	Relation	Instance	1
	hatAuftraggeber	Unternehmen	Relation	String	1
	hatUnmittelbareKooperationspartner	Unternehmen	Relation	String	0-1
	wirdImEigenenUnternehmenDurchgeführtVon	Person	Relation	String	1
	umfasstEinsatzmittelplanung	Einsatzmittelplanung	Relation	Instance	1
	wirdGesteuertMitMethode	Steuerungsmethode	Relation	Instance	1

Tabelle 2: Klasse „Projekt“ mit ihren Slots und Sloteseigenschaften

Schritt 7: Spezifizierung von Instanzen

Im abschließenden siebten Schritt werden die Instanzen der Projektmanagement-Ontologie erstellt. Die Abgrenzung zwischen Klassen und Instanzen lässt sich nicht immer scharf vornehmen.²³⁹

Die Abbildung 8 auf der nächsten Seite zeigt einen Ausschnitt aus der Gesamtheit der Instanzen der Subklasse „OriginäreQuantitativeEigenschaft“. Die Instanzen erlauben weitere Variationsmöglichkeiten bezüglich der Ausgestaltung. Eine Konkretisierung dieser Instanzen erfolgt mithilfe von Attributen und ihren Ausprägungen (Werten). Diese Attribute können beispielsweise konkrete Zeitpunkte und Mengenangaben umfassen.

239) Vgl. NOY/MCGUINNESS (2001), S. 18. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

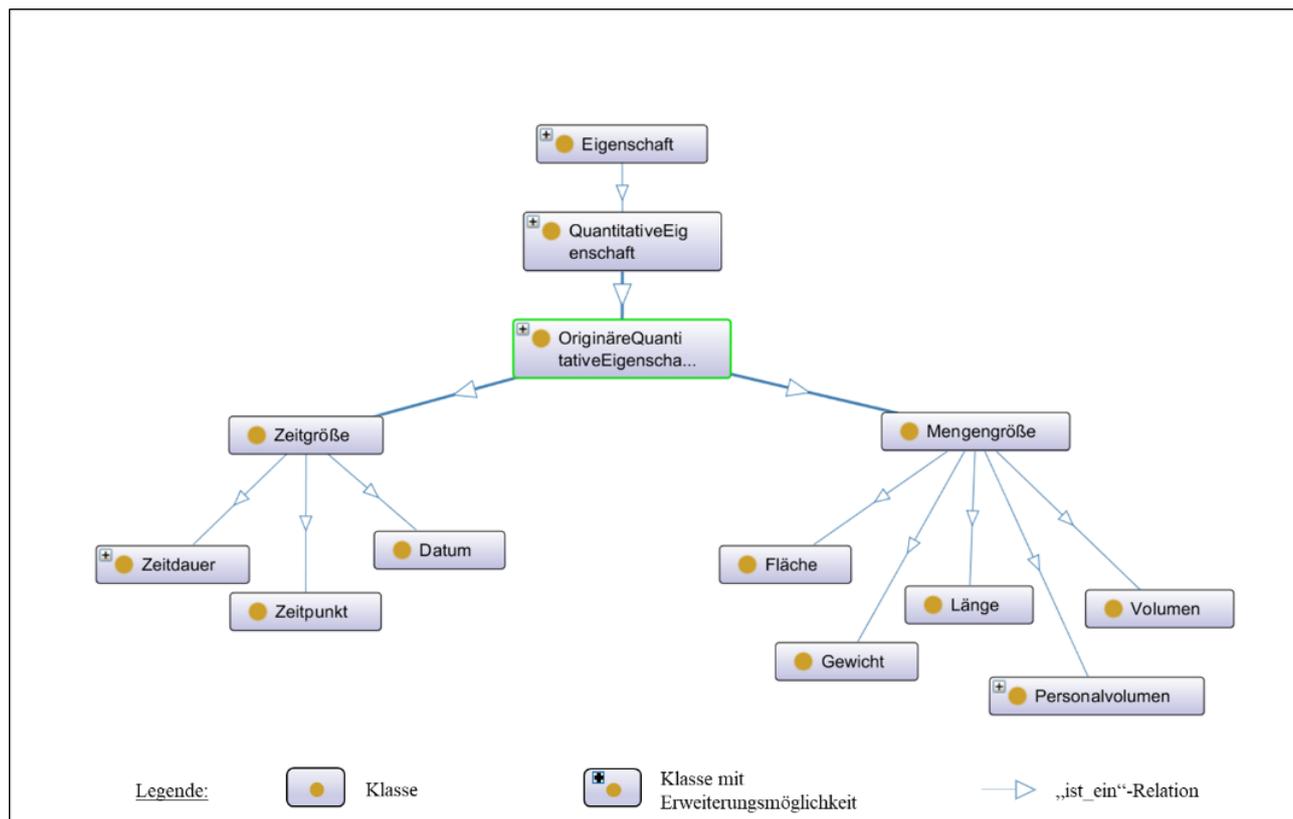


Abbildung 8: Instanzen der Subklasse „OriginäreQuantitativeEigenschaft“²⁴⁰

Die nachstehende Tabelle 3: Beispiele für Relations- oder Attributwerte einer Instanz

liefert einen Überblick über die konkrete Ausgestaltung von Instanzen. Bei diesen Relations- oder Attributwerten handelt es sich jeweils um fiktive Werte.

Instanz „Projekt Softwareentwicklung“	
Relationen/Attribute	Relations- oder Attributwerte
betrifftLand	Deutschland
betrifftRegion	Essen
benötigtZeitPlan	5 Monate
beginntAm	01.05.2018
endetAm	01.09.2018
erfordertKompetenzen	Programmieren, Vertrieb
hatProjektvolumenKostenIst	300.000 €
hatProjektvolumenUmsatzIst	600.000 €
hatProjektvolumenPersonalAnzahlIst	2
hatAuftraggeber	XY Food GmbH

Tabelle 3: Beispiele für Relations- oder Attributwerte einer Instanz

240) In der Abbildung 8 wird als Folge des eingesetzten Tools OntoGraf die Richtung der „ist_ein“-Relation „verkehrt herum“ dargestellt.

4 Prototypische Implementierung des ontologiegestützten CBR-Systems jCORA

4.1 Struktur der Implementierung

Das Erfahrungswissen innerhalb eines Unternehmens stellt eine wichtige Unternehmensressource dar.²⁴¹ Mit Hilfe ontologiebasierter CBR-Systeme kann dieses Wissen wiederverwendet werden. Um die KI-Technik des Case-based Reasonings hinsichtlich der Phasen Retrieve, Reuse, Revise, Retain des typischen CBR-Zyklus anwenden zu können, existieren mehrere Prototypen für ontologiebasierte CBR-Systeme. Einer dieser Prototypen ist das CBR-System jCORA²⁴², das als Werkzeug zur Wissenswiederverwendung im betrieblichen Projektmanagement eingesetzt werden kann.²⁴³

Zur Implementierung eines CBR-Systems wird zunächst ein Web Ontology Language (OWL) Framework (kurz: OWL-Framework) ausgewählt.²⁴⁴ Nachfolgend werden einige Funktionen²⁴⁵ eines solchen OWL-Frameworks aufgeführt:

- *Aufbewahrung*: das Speichern von Ontologien im Hauptspeicher, auf Festplatten und in Datenbanken,
- *Wachstum*: das Instanzieren von Ontologien aus Internetquellen, Dateien und durch manuelles Hinzufügen,
- *Verknüpfung*: das Bilden von Vereinigungs- und Schnittmengen in Bezug auf Klasseneigenschaften,
- *Lokation*: die Navigation innerhalb von Ontologien,
- *Anfrage*: das Stellen komplexer Anfragen an das System,
- *Schlussfolgerung*: das Bilden eines Inferenzmodells²⁴⁶ mittels eines OWL-Reasoners,
- *Export und Import*: Export und Import von Ontologien in unterschiedliche Datei-Formate.

Das in der Fachliteratur am häufigsten anzutreffende OWL-Werkzeug ist „OWL API“, unterstützt vom Framework „APACHE Jena“.²⁴⁷ Für die Integration der in Kapitel 3 erstellten Projektmanagement-Ontologie in das ontologiegestützte CBR-System jCORA wurde das Werkzeug „OWL API“ ausgewählt, weil es u. a. von Protégé²⁴⁸ verwendet wird und sowohl OWL-Ontologien als auch diverse OWL-Reasoner unterstützt.²⁴⁹

241) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 475.

242) Das Akronym steht für „java based Case- and Ontology-based Reasoning Application“.

243) Das CBR-System jCORA wurde im Rahmen des Verbundprojekts OrGoLo erstellt. Es sollte u. a die Anforderungen erfüllen, Case-based Reasoning mit unterschiedlichen, zeitlich variablen Fallstrukturen durchzuführen und einen leistungsfähigen Algorithmus zur Ähnlichkeitsbestimmung zu implementieren. Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 480.

244) Vgl. BERGENRODT/KOWALSKI (2015), S. 34.

245) Vgl. HEBELER/FISHER/BLACE et al. (2009), S. 268 f.

246) Ein Inferenzmodell umfasst nicht nur explizite Aussagen, sondern auch Aussagen, die mittels Transitivitätsregeln abgeleitet werden können. Beispielsweise lässt sich aus den Aussagen „Golf ist ein Auto“ und „Auto ist ein Fahrzeug“ mittels transitiver Inferenz die Aussage „Golf ist ein Fahrzeug“ herleiten; vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 515.

247) Vgl. GANZHA/MESJASZ/PAPRZYCHI et al. (2014), S. 6 ff.; PÖHLAND/KORETZ/SCHLEGEL (2013), S. 75 f.; HEBELER/FISHER/BLACE et al. (2009), S. 154 ff.

248) Vgl. WATERFELD/WEITEN/HAASE (2011), S. 64.

249) Vgl. HORRIDGE/BECHHOFFER (2011), S. 11 f.

Die Architektur der Fallbasis des CBR-Systems jCORA besteht aus globalen Domänenontologien und lokalen Fallontologien.²⁵⁰ Als Grundlage der Domänenontologie²⁵¹ dienen Klassen, Instanzen, Attribute und Relationen sowie Inferenz- und Integritätsregeln. Projektspezifische Fälle werden innerhalb einer lokalen Fallontologie abgelegt.

Die Bearbeitung eines Falls erfolgt in vier Schritten.²⁵² Im ersten Schritt wird die Domänenontologie von einem Datenträger geladen.²⁵³ Anschließend wird die lokale Fallontologie aus einem Tripel-Speicher geladen. Im dritten Schritt wird die Vereinigungsmenge dieser beiden Ontologien erstellt, die als Fallontologie bezeichnet wird. Zum Schluss wird mit Hilfe des „Pellet Reasoners“ ein Inferenzmodell²⁵⁴ konstruiert. Jegliche Veränderungen eines bearbeiteten Falls werden in der lokalen Fallontologie gespeichert.

Um das Projektwissen eines Unternehmens mittels des CBR-Systems jCORA wiederverwenden zu können, gilt es bereits in Protégé auf die verwendeten sprachlichen Ausdrucksmittel zu achten. Die Basis von jCORA und Protégé bildet eine globale Domänenontologie.

Drei wesentliche Anforderungen werden an die globale Domänenontologie gestellt. Erstens wird eine dreidimensionale Fallstruktur benötigt.²⁵⁵ Die Projektmanagement-Ontologie enthält entsprechend der Fallstruktur die Klassen „Projektbeschreibung“²⁵⁶, „Projektlösung“ und „Projektbewertung“.²⁵⁷ Zweitens werden diese drei Klassen mittels Relationen der Klasse „Projekt“ als Nachbereiche dieser Relationen zugeordnet. Drittens dürfen keine Instanzen der Unterklassen „Projekt“, „Projektbeschreibung“, „Projektlösung“ und „Projektbewertung“ in der Projektmanagement-Ontologie vorhanden sein, weil ansonsten bei einem Vergleich zweier Fälle (Projekte) der Ausgangspunkt für z. B. eine Ähnlichkeitsberechnung nicht eindeutig diesen Instanzen (Fällen, Projekten) zugeordnet werden kann.²⁵⁸

Ein grundlegendes Bedienungskonzept des CBR-Systems jCORA, beginnend mit der Eingabe eines neuen Falls bis hin zur Erweiterung der Fallbasis mittels eines bearbeiteten neuen Falls, wird in den Anhängen 4 bis 6 dargestellt.

250) Vgl. BERGENRODT/KOWALSKI (2015), S. 36. Der Inhalt der nächsten beiden Sätze bezieht sich ebenfalls auf diese Quelle.

251) Die Domänenontologie selbst kann mithilfe des Ontologie-Editors Protégé bearbeitet werden.

252) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 514 f. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

253) Es gilt zu beachten, dass der Datenträger des Computers genutzt werden muss, auf dem das CBR-Tool installiert wurde.

254) Ein Inferenzmodell umfasst nicht nur die Ontologie selbst, sondern auch alle Aussagen in der RDF-Standard-Form (Subjekt, Prädikat, Objekt), die aus einer Ontologie mittels einer Inferenzkomponente („Reasoner“) abgeleitet werden können. Das Ableiten von Aussagen erfolgt gemäß Transitivitätsregeln; vgl. GLEICH (2008), S. 105.

255) Vgl. ASSALIM/LENNE/DEBRAY (2010), S. 102 f.

256) Dieser Begriff wird synonym zum Begriff „Fallbeschreibung“ verwendet.

257) Vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 516. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

258) Vgl. BERGENRODT/KOWALSKI (2015), S. 38.

4.2 Implementierung des rekursiven Ähnlichkeitsalgorithmus

4.2.1 Ein Anwendungsbeispiel

4.2.1.1 Überblick über das Anwendungsbeispiel

Im Folgenden wird anhand fiktiver Beispiele der Vorgang des Auffindens eines ähnlichsten Falls (Projekts) mittels des rekursiven Ähnlichkeitsalgorithmus beschrieben, der im CBR-System jCORA implementiert ist. Hierzu erfolgt zunächst eine manuelle Berechnung der Ähnlichkeit zweier Fälle. Anschließend wird das Auffinden (mindestens) eines ähnlichsten Falls mithilfe des ontologiegestützten CBR-Systems jCORA exemplarisch demonstriert. Um Ergebnisse für die Ähnlichkeitsberechnung erhalten zu können, werden bereits durchgeführte Fälle benötigt.²⁵⁹ Diese Fälle bedienen sich ebenfalls fiktiver Beispiele, die in dem Kapitel 4.2.1.2 skizziert werden.

Für die Berechnung der Ähnlichkeit dieser fiktiven Fälle werden die Falllösung und die Fallbewertung vernachlässigt, da sie für die Ähnlichkeitsberechnung nicht ausschlaggebend sind.²⁶⁰

4.2.1.2 Faldefinitionen

Alter Fall I: Das Sortimentsoptimierungs-Unternehmen UDE-Assortment Optimization GmbH (kurz: UDE-AO GmbH) entwickelte im Zeitraum vom 01.05.2018 bis zum 01.09.2018 für den in Essen ansässigen Großhändler XY Food GmbH eine Software zur Regaloptimierung. Der primäre Verwendungszweck dieser Regaloptimierungssoftware lag in einer Ausschöpfung der Regalplätze und in der Optimierung der „Facings“²⁶¹ innerhalb eines Regals.



Abbildung 9: Planogramm zur Sortimentsoptimierung²⁶²

Das für dieses Projekt beauftragte Projektteam bestand aus zwei ausgewählten Mitarbeitern. Die Projektleiterin Frau C. MEIER wurde ausgewählt, da sie mit ihren Vertriebskompetenzen nicht nur das Projekt strukturiert leiten konnte, sondern auch, weil sie ihre Kompetenzen im Bereich Vertrieb zur Vermarktung dieses Produktes einsetzen konnte. Frau U. BOHRT war ebenfalls an dem Projekt als Mitarbeiterin beteiligt und übernahm die programmierbedingten Aufgaben.

259) Vgl. BERGENRODT/KOWALSKI (2015), S. 55.

260) Vgl. BERGENRODT/KOWALSKI (2015), S. 56.

261) Vgl. HÜBNER/SCHAAL/DÜSTERHÖFT (2019), S. 5.

262) Die Abbildung von Validoo befindet sich im Internet unter der URL: <https://validoo.se/en/tjanst/validoo-planogramm/> (letzter Zugriff am: 13.03.2022).

Alter Fall II: Das Unternehmen UDE-AO GmbH hat im Zeitraum vom 01.01.2018 bis zum 18.07.2019 ein umfangreiches Projekt für den Großhändler XY Food GmbH durchgeführt. Es wurde ein neues Sicherheitssystem zum Schutz der Unternehmensdaten entwickelt.



Abbildung 10: IT-Sicherheit im Unternehmen²⁶³

Für dieses Projekt wurden vor allem eine Person mit ausgeprägten IT-Security-Kenntnissen benötigt. Das Unternehmen hat sich für den Mitarbeiter J. BORNEMANN entschieden, weil er langjährige Erfahrungen in diesem Bereich vorweisen kann.

Neuer Fall: Das Unternehmen UDE-AO GmbH hat einen weiteren Auftrag von dem Großhändler XY Food GmbH erhalten. Es soll eine Software entwickelt werden, mit der die Preise des Großhändlers berechnet und automatisch an elektronische Preisschilder gesendet werden.



Abbildung 11: Elektronische Preisschilder²⁶⁴

Für die Entwicklung dieses Pricingtools wurde der Mitarbeiter Herr C. DIETRICH engagiert, weil er bereits in der Vergangenheit Erfahrungen mit der Entwicklung von elektronischen Preisschildern gesammelt hat und über die benötigten Programmierkenntnisse verfügt.

263) Die Abbildung von Unsplash befindet sich unter der URL: <https://unsplash.com/photos/npxXWgQ33ZQ> (letzter Zugriff am: 13.03.2022).

264) Die Abbildung von Theamericangenius befindet sich unter der URL: <https://theamericangenius.com/wp-content/uploads/2015/07/electronic-price-tags.png> (letzter Zugriff am: 13.03.2022).

4.2.2 Manuelle Ähnlichkeitsberechnung

4.2.2.1 Ausgangspunkt der Berechnung

Die Implementierung des in Kapitel 2.4.2.2 vorgestellten rekursiven Ähnlichkeitsalgorithmus, der im CBR-System jCORA implementiert ist, lässt sich in drei Komponenten unterteilen:²⁶⁵ Die erste Komponente beinhaltet eine Ähnlichkeitsfunktion (ksim) zur Berechnung der Ähnlichkeit von Klassen. Die zweite Komponente umfasst Funktionen (sim_{instanz}) für die Ähnlichkeitsberechnung von Instanzen. Die Menge N der verfügbaren Ähnlichkeitsfunktionen ist durch $N = \{\text{sim_instanz}\}$ definiert. Die dritte Komponente dient dazu, die Ähnlichkeiten von Attributswerten zu berechnen.

Im Folgenden wird mit Hilfe des in Kapitel 2.4.2.2 vorgestellten Ähnlichkeitsalgorithmus die Ähnlichkeit der in Kapitel 4.2.1.2 dargestellten Fälle demonstriert. Für alle Gewichte w_r von Relationen und Attribute aus der Projektbeschreibung wird der Einfachheit halber der Wert 1,0 festgelegt.

Ausgangspunkt für die Berechnungen sind die Instanzen:

- „Fallbeschreibung_Regaloptimierung_XY_Food_GmbH“²⁶⁶ (alter Fall I)
- „Fallbeschreibung_Sicherheit_XY_Food_GmbH“²⁶⁷ (alter Fall II)
- „Fallbeschreibung_Pricingtool_XY_Food_GmbH“²⁶⁸ (neuer Fall)

Alle drei Instanzen gehören zur Klasse „Projektbeschreibung“. Den LCS dieser drei Instanzen bildet daher die Klasse „Projektbeschreibung“.

4.2.2.2 Ähnlichkeitsberechnung „Neuer Fall“ in Bezug auf „Alter Fall I“

Die Distanz zwischen den (identischen) Klassen „Projektbeschreibung“ und „Projektbeschreibung“ ergibt sich wie folgt:

$$\begin{aligned} & \text{dist}(\text{Projektbeschreibung}, \text{Projektbeschreibung}) \\ &= \max(|\text{pfad}(\text{Projektbeschreibung}, \text{Projektbeschreibung})|; \\ & \quad |\text{pfad}(\text{Projektbeschreibung}, \text{Projektbeschreibung})|) \\ &= \max(1,0; 1,0) \\ &= 1,0 \end{aligned}$$

Die Klasse „Projektbeschreibung“ besitzt drei Relationen („betrifftProjektart“, „wirdDurchgeführtVon“ und „wirdBearbeitetVon“). Die Menge $\text{kr}(\text{Projektbeschreibung})$ der auf dieser Klasse definierten Relationen²⁶⁹ entspricht der Menge:

$$\text{kr}(\text{Projektbeschreibung}) = \{\text{betrifftProjektart}; \text{wirdDurchgeführtVon}; \text{wirdBearbeitetVon}\}$$

265) Vgl. BERGENRODT/KOWALSKI (2015), S. 38. Der Inhalt dieser Quelle bezieht sich auf den gesamten Absatz.

266) Die Kurzversion „Regaloptimierung“ wird synonym zur Bezeichnung „Fallbeschreibung_Regal-optimierung_XY_Food_GmbH“ verwendet.

267) Die Kurzversion „Sicherheit“ wird synonym zur Bezeichnung „Fallbeschreibung_Sicherheit_XY_Food_GmbH“ verwendet.

268) Die Kurzversion „Pricingtool“ wird synonym zur Bezeichnung „Fallbeschreibung_Pricingtool_XY_Food_GmbH“ verwendet.

269) Zu Illustrationszwecken wird dieses Beispiel simpel gehalten. Daher werden keine Attribute für die Klasse „Projektbeschreibung“ definiert.

Die Klassenähnlichkeit der Klassen „Projektbeschreibung“ und „Projektbeschreibung“ ergibt:

$$\begin{aligned} & \text{ksim}(\text{Projektbeschreibung}, \text{Projektbeschreibung}) \\ &= \frac{1}{\text{dist}(\text{Projektbeschreibung}, \text{Projektbeschreibung})} \\ &= \frac{|\text{kr}(\text{Projektbeschreibung}) \cap \text{kr}(\text{Projektbeschreibung})|}{|\text{kr}(\text{Projektbeschreibung}) \cup \text{kr}(\text{Projektbeschreibung})|} \end{aligned}$$

Da die beiden Klassen identisch sind, lässt sich der Ausdruck vereinfachen:

$$\begin{aligned} & \text{ksim}(\text{Projektbeschreibung}, \text{Projektbeschreibung}) \\ &= \frac{1}{\text{dist}(\text{Projektbeschreibung}; \text{Projektbeschreibung})} * \frac{|\text{kr}(\text{Projektbeschreibung})|}{|\text{kr}(\text{Projektbeschreibung})|} \end{aligned}$$

Nach Einsetzen der zuvor berechneten Werte ergibt sich eine Klassenähnlichkeit von:

$$\begin{aligned} & \text{ksim}(\text{Projektbeschreibung}; \text{Projektbeschreibung}) \\ &= \frac{1}{1} * \frac{3}{3} \\ &= 1,0 \end{aligned}$$

Die Ähnlichkeit der Klassen „Projektbeschreibung“ und „Projektbeschreibung“ beträgt daher 1,0.

Um die Ähnlichkeit der Instanzen „Fallbeschreibung_Regaloptimierung_XY_Food_GmbH“ und „Fallbeschreibung_Pricingtool_XY_Food_GmbH“ berechnen zu können, müssen zunächst die Ähnlichkeiten aller Relations- und Attributswerte dieser beiden Instanzen bekannt sein.

Die Instanz „UDE-AO_GmbH“ ist eine für beide Fälle in der Domänenontologie definierte Instanz, sodass die Ähnlichkeit aufgrund der vorliegenden Reflexivitätseigenschaft der Ähnlichkeitsfunktion $\text{sim}_{\text{instanz}}$ vereinfacht werden kann:

$$\text{sim}_{\text{instanz}}(\text{wirdDurchgefuehrtVon}; \text{UDE} - \text{AO_GmbH}; \text{UDE} - \text{AO_GmbH}) = 1,0$$

Der Nachbereich der Relation „betrifftProjektart“ der beiden zu vergleichenden Fälle weist nur die gemeinsame Instanz „Software-Entwicklung“ auf. Analog zur vorherigen Ähnlichkeitsbestimmung kann aufgrund der Reflexivitätseigenschaft der Ähnlichkeitsfunktion $\text{sim}_{\text{instanz}}$ die Ähnlichkeit wie folgt bestimmt werden:

$$\text{sim}_{\text{instanz}}(\text{betrifftProjektart}; \text{Software} - \text{Entwicklung}; \text{Software} - \text{Entwicklung}) = 1,0$$

Im Gegensatz zu den zuvor gewählten Relationen weist die Relation „wirdBearbeitetVon“ im Nachbereich jeweils fallspezifische Instanzen auf. Zunächst müssen die Ähnlichkeiten aller Relationswerte ermittelt werden. Hierfür bedarf es auf der einen Seite einer Ähnlichkeitsberechnung zwischen den Instanzen „Projektteam_Regaloptimierung“ und „Projektteam_Pricingtool“. Beide Instanzen gehören zur Klasse „Projektteam“. Durch die Reflexivitätseigenschaft der Funktion ksim lässt sich die Klassenähnlichkeit $\text{ksim}(\text{Projektteam}; \text{Projektteam})$ mit 1,0 bestimmen.

Bevor die Ähnlichkeitsberechnung vervollständigt werden kann, müssen die Ähnlichkeiten aller Relationswerte der Instanzen ermittelt werden. Hierfür wird die Ähnlichkeit zwischen den Instanzen „Mitarbeiter_U.Bohrt“ und „Mitarbeiter_C.Dietrich“ auf der einen Seite und den Instanzen „Projektleiter_C.Meier“ und „Mitarbeiter_C.Dietrich“ auf der anderen Seite berechnet.

Zunächst wird die Ähnlichkeit der Instanzen „Mitarbeiter_U.Bohrt“ und „Mitarbeiter_C.Dietrich“ berechnet. Beide Instanzen gehören zur Klasse „Mitarbeiter“, sodass die Klassenähnlichkeit $\text{ksim}(\text{Mitarbeiter}; \text{Mitarbeiter})$ gleich 1,0 ist. Die Instanzenähnlichkeit lautet wie folgt:

$$\text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_U.Bohrt}; \text{Mitarbeiter_C.Dietrich}) = 1,0$$

Die Instanz „Mitarbeiter_U.Bohrt“ besitzt die Relation „hatProjektkompetenz“ mit der Instanz „SoftwareProgrammierung“ im Nachbereich. Die Instanz „Mitarbeiter_C.Dietrich“ besitzt ebenfalls

die Relation „hatProjektkompetenz“ mit der Instanz „SoftwareProgrammierung“ im Nachbereich. Daher kann auch an dieser Stelle die Ähnlichkeitsberechnung vereinfacht werden:

$$\text{sim}_{\text{Instanz}}(\text{hatProjektkompetenz}; \text{SoftwareProgrammierung}; \text{SoftwareProgrammierung}) = 1,0$$

Im Folgenden wird die Ähnlichkeit der Instanzen „Projektleiter_C.Meier“ und „Mitarbeiter_C.Dietrich“ berechnet.

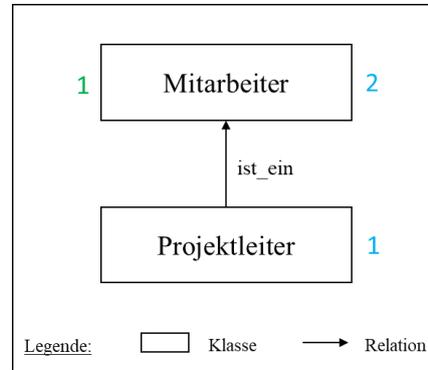


Abbildung 12: Klassenhierarchie der Klassen „Mitarbeiter“ und „Projektleiter“

Die Klassendistanz ergibt sich wie folgt:

$$\begin{aligned} \text{dist}(\text{Mitarbeiter}; \text{Projektleiter}) &= \max(|\{\text{Mitarbeiter}; \text{Person}\}|; |\{\text{Projektleiter}; \text{Person}\}|) \\ &= \max(1; 2) \\ &= 2 \end{aligned}$$

Die Klassenähnlichkeit ergibt sich wie folgt:

$$\begin{aligned} \text{ksim}(\text{Mitarbeiter}; \text{Projektleiter}) &= \frac{1}{2} * \frac{1}{1} \\ &= 0,5 \end{aligned}$$

Die Instanzen „Projektleiter_C.Meier“ und „Mitarbeiter_C.Dietrich“ besitzen beide Kompetenzen im Nachbereich. „Projektleiter_C.Meier“ besitzt die Relation „hatProjektkompetenzen“ mit der Instanz „Vertriebskompetenz“ im Nachbereich, „Mitarbeiter_C.Dietrich“ besitzt die Relation „hatProjektkompetenzen“ mit der Instanz „SoftwareProgrammierung“ im Nachbereich. Die Instanzen „Vertriebskompetenz“ und „SoftwareProgrammierung“ gehören zu unterschiedlichen Klassen. Den LCS bildet die Klasse „Kompetenztyp“; vgl. die nachfolgende Abbildung 13.

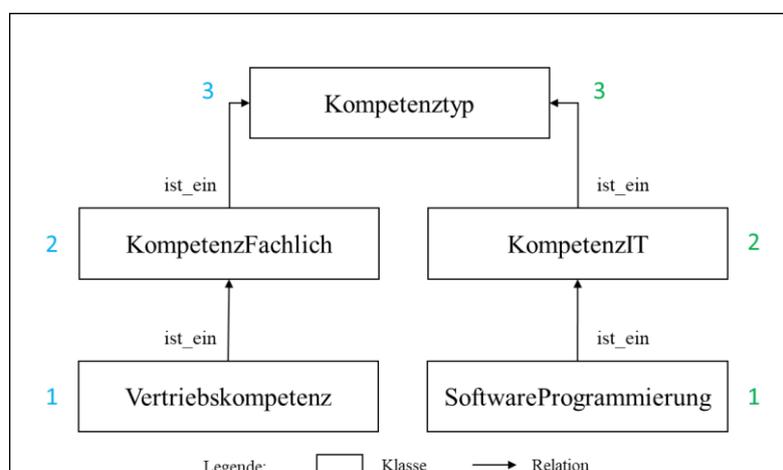


Abbildung 13: Klassenhierarchie für die Klassen „Vertriebskompetenz“ und „SoftwareProgrammierung“

Die Instanzendistanz zwischen „Vertriebskompetenz“ und „SoftwareProgrammierung“ lässt sich wie folgt bestimmen:

$$\begin{aligned} & \text{dist}(\text{Vertriebskompetenz}; \text{SoftwareProgrammierung}) \\ &= \max(3; 3) \\ &= 3 \end{aligned}$$

Für die zu vergleichenden Instanzen sind keine gemeinsamen Relationen oder Attribute definiert ($|\text{kr}(k_a) \cup \text{kr}(k_b)| = 0$). Demnach entspricht die Ähnlichkeit dem Reziprokwert der Klassendistanz:

$$\text{ksim}(\text{Vertriebskompetenz}; \text{SoftwareProgrammierung}) = \frac{1}{3} * 1,0 = \frac{1}{3}$$

Die Instanzen besitzen keine Relationen oder Attribute:

$$|\text{R}(\text{Vertriebskompetenz}) \cup \text{R}(\text{SoftwareProgrammierung})| = 0,0$$

Damit ergibt sich folgende Instanzenähnlichkeit:

$$\begin{aligned} & \text{sim}_{\text{instanz}}(\text{hatProjektkompetenzen}; \text{Vertriebskompetenz}; \text{SoftwareProgrammierung}) \\ &= \frac{1}{3} * 1,0 = \frac{1}{3} \end{aligned}$$

Die kumulierte Ähnlichkeit ergibt sich nach der Formel:

$$\text{relsim}(\text{Projektleiter_C. Meier}; \text{Mitarbeiter_C. Dietrich}; \{\text{hatProjektkompetenzen}\}; \text{sim}_{\text{instanz}})$$

$$= W_{\text{hatProjektkompetenzen}}$$

$$\begin{aligned} & \max\left(\text{sim}_{\text{instanz}}\left(\begin{array}{c} \text{hatProjektkompetenzen}; \text{Vertriebskompetenz}; \\ \text{SoftwareProgrammierung} \end{array}\right)\right) + \\ & \max\left(\text{sim}_{\text{instanz}}\left(\begin{array}{c} \text{hatProjektkompetenz}; \text{Vertriebskompetenz}; \\ \text{SoftwareProgrammierung} \end{array}\right)\right) \\ & * \frac{1}{|\{\text{Vertriebskompetenz}\}| + |\{\text{SoftwareProgrammierung}\}|} \\ & = 1,0 * \frac{\frac{1}{3} + \frac{1}{3}}{1 + 1} \\ & = \frac{1}{3} \end{aligned}$$

Damit lässt sich die Ähnlichkeitsberechnung der Instanzen „Projektleiter_C.Meier“ und „Mitarbeiter_C.Dietrich“ vervollständigen. Es ergibt sich eine Instanzenähnlichkeit von:

$$\text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Projektleiter}_C. \text{ Meier}; \text{Mitarbeiter}_C. \text{ Dietrich})$$

$$= 0,5 * \frac{\text{relsim}(\text{Projektleiter_C. Meier}; \text{Mitarbeiter_C. Dietrich}; \{\text{hatProjektkompetenzen}\}; \text{sim}_{\text{instanz}})}{W_{\text{hatProjektkompetenz}}}$$

$$= 0,5 * \frac{\frac{1}{3}}{1,0}$$

$$= \frac{1}{6}$$

Die Instanzen „Projektleiter_C.Meier“ und „Mitarbeiter_C.Dietrich“ besitzen also eine Ähnlichkeit von rund 0,167.

Aufgrund der Symmetrieeigenschaft beträgt sie Instanzenähnlichkeit der Mitarbeiter U. BOHRT und Projektleiter C. MEIER ebenso rund 0,167.

$$\text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Projektleiter_C. Meier}; \text{Mitarbeiter_U. Bohrt}) = \frac{1}{6}$$

An dieser Stelle kann die Berechnung der Ähnlichkeit der Instanzen „Projektteam_Regaloptimierung“ und „Projektteam_Pricingtool“ bezüglich der Relation „hatMitglied“ fortgesetzt werden. Hierfür wird für jedes Element der ersten Wertemenge {Projektleiter_C.Meier; Mitarbeiter_U.Bohrt} das ähnlichste Element der zweiten Wertemenge {Mitarbeiter_C.Dietrich} gesucht. Durch Einsetzen der Elemente in den Term:

$$\sum_{a \in \text{wert}(i_a, r)} \max_{b \in \text{wert}(i_b, r)} (\text{sim}_n(r, a, b))$$

ergibt sich:

$$\begin{aligned} & \max \left(\begin{array}{l} \text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Projektleiter_C.Meier}; \text{Mitarbeiter_U.Bohrt}); \\ \text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_C.Meier}; \text{Mitarbeiter_C.Dietrich}) \end{array} \right) + \\ & \max \left(\begin{array}{l} \text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_U.Bohrt}; \text{Projektleiter_C.Meier}); \\ \text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_U.Bohrt}; \text{Mitarbeiter_C.Dietrich}) \end{array} \right) \end{aligned}$$

Durch Einsetzen der berechneten Werte ergibt sich:

$$\begin{aligned} & \max \left(\frac{1}{6}; \frac{1}{6} \right) + \max \left(\frac{1}{6}; 1,0 \right) \\ & = \frac{1}{6} + 1,0 = \frac{7}{6} \approx 1,167 \end{aligned}$$

Analog wird für jedes Element aus der zweiten Wertemenge das ähnlichste Element aus der ersten Wertemenge ermittelt:

$$\begin{aligned} & \max \left(\begin{array}{l} \text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_C.Dietrich}; \text{Mitarbeiter_U.Bohrt}); \\ \text{sim}_{\text{instanz}}(\text{Mitarbeiter_C.Dietrich}; \text{Projektleiter_C.Meier}) \end{array} \right) \\ & \max \left(1,0; \frac{1}{6} \right) \\ & = 1,0 \\ & = 1,0 \end{aligned}$$

Durch Einsetzen der berechneten Werte ergibt sich folgende kumulierte Ähnlichkeit der Instanzen „Projektteam_Regaloptimierung“ und „Projektteam_Pricingtool“ bezüglich der Relation „hatMitglied“:

$$\begin{aligned} & \text{relsim}(\text{Projektteam_Regaloptimierung}; \text{Projektteam_Pricingtool}; \{\text{hatMitglied}\}; \text{sim}_{\text{instanz}}) \\ & = w_{\text{hatMitglied}} * \frac{\frac{7}{6} + 1,0}{|\{\text{Projektleiter_C.Meier}; \text{Mitarbeiter_U.Bohrt}\}| + |\{\text{Mitarbeiter_C.Dietrich}\}|} \\ & = 1,0 * \frac{\frac{7}{6} + 1,0}{2,0 + 1,0} \\ & = \frac{13}{18} \approx 0,722\bar{2} \end{aligned}$$

Die finale Instanzenähnlichkeit berechnet sich wie folgt:

$$\begin{aligned} & \text{sim}_{\text{instanz}}(\text{wirdBearbeitetVon}; \text{Projektteam_Regaloptimierung}; \text{Projektteam_Pricingtool}) \\ & = 1,0 * \frac{\text{relsim}(\text{Projektteam_Regaloptimierung}; \text{Projektteam_Pricingtool}; \{\text{hatMitglied}\}; \text{sim}_{\text{instanz}})}{w_{\text{hatMitglied}}} \\ & = 1,0 * \frac{\frac{13}{18}}{1,0} \\ & = \frac{13}{18} \end{aligned}$$

Die kumulierte Ähnlichkeit beträgt nach Einsetzen der zuvor berechneten Werte:

$$\begin{aligned}
& \text{rehsim} \left(\begin{array}{c} \text{Fallbeschreibung_Pricingtool; Fallbeschreibung_Regaloptimierung;} \\ \text{betrifftProjektart;} \\ \left\{ \text{projektWirdDurchgefuehrtVon; projektWirdBearbeitetVon} \right\}; \\ \text{sim}_{\text{instanz}} \end{array} \right) \\
&= 1,0 * \frac{\max\left(\frac{13}{18}\right) + \max\left(\frac{13}{18}\right)}{1,0 + 1,0} \\
&+ 1,0 * \frac{\max(1,0) + \max(1,0)}{1,0 + 1,0} \\
&+ 1,0 * \frac{\max(1,0) + \max(1,0)}{1,0 + 1,0} \\
&= \frac{13}{18} + 1,0 + 1,0 \\
&= \frac{49}{18} \\
&\approx 2,7222
\end{aligned}$$

Die abschließende Berechnung lautet:

$$\begin{aligned}
& \text{sim}_{\text{instanz}} \left(\begin{array}{c} \text{bestehtAusProjektbeschreibung; Fallbeschreibung_Pricingtool;} \\ \text{Fallbeschreibung_Regaloptimierung} \end{array} \right) \\
& \text{rehsim} \left(\begin{array}{c} \text{Fallbeschreibung_Pricingtool; Fallbeschreibung_Regaloptimierung;} \\ \text{betrifftProjektart;} \\ \left\{ \text{projektWirdDurchgefuehrtVon; projektWirdBearbeitetVon} \right\}; \\ \end{array} \right) \\
&= 1,0 * \frac{W_{\text{betrifftProjektart}} + W_{\text{projektWirdDurchgefuehrtVon}} + W_{\text{projektWirdBearbeitetVon}}}{49} \\
&= 1,0 * \frac{\frac{49}{18}}{1,0 + 1,0 + 1,0} \\
&= \frac{49}{54} \\
&\approx 0,9074
\end{aligned}$$

Der Ähnlichkeitswert der Fallbeschreibungen des neuen Falls „Pricingtool“ und des alten „Regaloptimierung“ beträgt also 0,9074. Dies entspricht einer Ähnlichkeit von ca. 91 %.

4.2.2.3 Ähnlichkeitsberechnung „Neuer Fall“ in Bezug auf „Alter Fall II“

Die Klassenähnlichkeit der beiden zu vergleichenden Fälle „Fallbeschreibung_Pricing-tool_XY_Food_GmbH“ und „Fallbeschreibung_Sicherheit_XY_Food_GmbH“ weist aufgrund der Symmetrieeigenschaft zu den vorherigen Fallbeschreibungen folgenden Wert auf:

$$\begin{aligned}
& \text{ksim}(\text{Projektbeschreibung; Projektbeschreibung}) \\
&= \frac{1}{1} * \frac{3}{3} \\
&= 1,0
\end{aligned}$$

Die Klasse „Projektbeschreibung“ besitzt ebenfalls drei Relationen („betrifftProjektart“, „wirdDurchgeführtVon“ und „wirdBearbeitetVon“). Die Menge der auf dieser Klasse definierten Relationen²⁷⁰ $kr(\text{Projektbeschreibung})$ entspricht folgender Menge:

$$kr(\text{Projektbeschreibung}) = \{\text{betrifftProjektart}; \text{wirdDurchgeführtVon}; \text{wirdBearbeitetVon}\}$$

Um die Ähnlichkeit der Instanzen „Fallbeschreibung_Sicherheit_XY_GmbH“ und „Fallbeschreibung_Pricingtool_XY_GmbH“ berechnen zu können, muss zunächst die Ähnlichkeit der Relations- und Attributswerte aller Instanzen bekannt sein. Die Instanz „UDE-AO_GmbH“ ist eine in der Domänenontologie gemeinsam definierte Instanz:

$$\text{sim}_{\text{instanz}}(\text{wirdDurchgeführtVon}; \text{UDE} - \text{AO_GmbH}; \text{UDE} - \text{AO_GmbH}) = 1,0$$

Der Nachbereich der Relation „betrifftProjektart“ der beiden zu vergleichenden Fälle weist zwei unterschiedliche Instanzen auf. Beide Instanzen gehören zur Klasse „Projektart“. Durch die Reflexivitätseigenschaft der Funktion k_{sim} lässt sich die Klassenähnlichkeit $k_{\text{sim}}(\text{Projektart}; \text{Projektart})$ mit 1,0 bestimmen. Die Ähnlichkeitsfunktion $\text{sim}_{\text{instanz}}$ ergibt nach Einsetzen der berechneten Werte:

$$\begin{aligned} &\text{sim}_{\text{instanz}}(\text{betrifftProjektart}; \text{Softwareentwicklung}; \text{EntwicklungSicherheitssystem}) \\ &= 1 * 1 = 1,0 \end{aligned}$$

Die Relation „wirdBearbeitetVon“ weist im Nachbereich ebenfalls fallspezifische Instanzen auf. Zunächst müssen die Ähnlichkeiten aller Relationswerte einer Instanz ermittelt werden. Hierfür bedarf es auf der einen Seite einer Ähnlichkeitsberechnung zwischen den Instanzen „Projektteam_Sicherheit“ und „Projektteam_Pricingtool“. Beide Instanzen gehören zur Klasse „Projektteam“. Durch die Reflexivitätseigenschaft der Funktion k_{sim} lässt sich die Klassenähnlichkeit $k_{\text{sim}}(\text{Projektteam}; \text{Projektteam})$ mit 1,0 bestimmen.

Bevor diese Ähnlichkeitsberechnung vervollständigt werden kann, müssen die Ähnlichkeiten aller Relationswerte der Instanzen ermittelt werden. Hierfür wird die Ähnlichkeit zwischen den Instanzen „Mitarbeiter_J.Bornemann“ und „Mitarbeiter_C.Dietrich“ berechnet.

Beide Instanzen gehören zur Klasse „Mitarbeiter“, sodass die Klassenähnlichkeit $k_{\text{sim}}(\text{Mitarbeiter}; \text{Mitarbeiter})$ gleich 1,0 ist. Die Instanzenähnlichkeit lautet wie folgt:

$$\text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_J. Bornemann}; \text{Mitarbeiter_C. Dietrich}) = 1,0$$

Die Instanz „Mitarbeiter_J.Bornemann“ besitzt die Relation „hatProjektkompetenz“ mit der Instanz „IT-Security-Kenntnisse“ im Nachbereich. Die Instanz „Mitarbeiter_C.Dietrich“ besitzt ebenfalls die Relation „hatProjektkompetenz“ mit der Instanz „SoftwareProgrammierung“ im Nachbereich. Die Instanzen „IT-Security-Kenntnisse“ und „SoftwareProgrammierung“ gehören beide zur Klasse „KompetenzIT“. Demnach beträgt die Klassenähnlichkeit $k_{\text{sim}}(\text{KompetenzIT}; \text{KompetenzIT})$ 1,0. Damit ergibt sich folgende Instanzenähnlichkeit:

$$\begin{aligned} &\text{sim}_{\text{instanz}}(\text{hatProjektkompetenz}; \text{IT} - \text{Security} - \text{Kenntnisse}; \text{SoftwareProgrammierung}) \\ &= \frac{1}{1} * 1,0 \\ &= 1,0 \end{aligned}$$

Die kumulierte Ähnlichkeit der einzigen Relation „hatProjektkompetenz“ ergibt:

$$\begin{aligned} &\text{relsim}(\text{Mitarbeiter_J. Bornemann}; \text{Mitarbeiter_C. Dietrich}; \{\text{hatProjektkompetenz}\}; \text{sim}_{\text{instanz}}) \\ &= 1,0 * \frac{1,0 + 1,0}{1 + 1} \\ &= 1,0 \end{aligned}$$

270) Zu Illustrationszwecken wird dieses Beispiel simpel gehalten. Daher werden keine Attribute für die Klasse „Projektbeschreibung“ definiert.

Damit lässt sich die Ähnlichkeitsberechnung der Instanzen „Mitarbeiter_J.Bornemann“ und „Mitarbeiter_C.Dietrich“ vervollständigen. Es ergibt sich eine Instanzähnlichkeit von:

$$\begin{aligned} & \text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_J. Bornemann}; \text{Mitarbeiter_C. Dietrich}) \\ &= 1,0 * \frac{\text{relsim}(\text{Mitarbeiter_J. Bornemann}; \text{Mitarbeiter_C. Dietrich}; \{\text{hatProjektkompetenz}\}; \text{sim}_{\text{instanz}})}{W_{\text{hatProjektkompetenz}}} \\ &= 1,0 * \frac{1,0}{1,0} \\ &= 1,0 \end{aligned}$$

Die Instanzen „Mitarbeiter_J.Bornemann“ und „Mitarbeiter_C.Dietrich“ besitzen folglich die maximale Ähnlichkeit 1,0.

An dieser Stelle kann die Berechnung der Ähnlichkeit der Klassen „Projektteam_Sicherheit“ und „Projektteam_Pricingtool“ bezüglich der Relation „hatMitglied“ fortgesetzt werden. Hierfür wird für jedes Element der ersten Wertemenge {Mitarbeiter_J.Bornemann} das ähnlichste Element aus der zweiten Wertemenge {Mitarbeiter_C.Dietrich} gesucht. Durch Einsetzen der Werte ergibt sich:

$$\begin{aligned} & \max(\text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_C. Dietrich}; \text{Mitarbeiter_J. Bornemann})) \\ & \quad \max(1,0) \\ & = 1,0 \end{aligned}$$

Analog wird für jedes Element aus der zweiten Wertemenge das ähnlichste Element aus der ersten gefunden:

$$\begin{aligned} & \max(\text{sim}_{\text{instanz}}(\text{hatMitglied}; \text{Mitarbeiter_J. Bornemann}; \text{Mitarbeiter_C. Dietrich})) \\ & \quad \max(1,0) \\ & = 1,0 \end{aligned}$$

Durch Einsetzen der berechneten Werte ergibt sich eine kumulierte Ähnlichkeit der Instanzen „Projektteam_Sicherheit“ und „Projektteam_Pricingtool“ bezüglich der Relation „hatMitglied“:

$$\begin{aligned} & \text{relsim}(\text{Projektteam}_{\text{Sicherheit}}; \text{Projektteam}_{\text{Pricingtool}}; \{\text{hatMitglied}\}; \text{sim}_{\text{instanz}}) \\ &= W_{\text{hatMitglied}} * \frac{1,0 + 1,0}{|\{\text{Mitarbeiter}_J. \text{Bornemann}\}| + |\{\text{Mitarbeiter}_C. \text{Dietrich}\}|} \\ &= 1,0 * \frac{2,0}{1,0 + 1,0} \\ &= 1,0 \end{aligned}$$

Die finale Instanzenähnlichkeit berechnet sich wie folgt:

$$\begin{aligned} & \text{sim}_{\text{instanz}}(\text{wirdBearbeitetVon}; \text{Projektteam}_{\text{Sicherheit}}; \text{Projektteam}_{\text{Pricingtool}}) \\ &= 1,0 * \frac{\text{relsim}(\text{Projektteam}_{\text{Sicherheit}}; \text{Projektteam}_{\text{Pricingtool}}; \{\text{hatMitglied}\}; \text{sim}_{\text{instanz}})}{W_{\text{hatMitglied}}} \\ &= 1,0 * \frac{1,0}{1,0} \\ &= 1,0 \end{aligned}$$

Die kumulierte Ähnlichkeit beträgt nach Einsetzen der zuvor berechneten Werte:

$$\begin{aligned} & \text{relsim} \left(\begin{array}{c} \text{Fallbeschreibung}_{\text{Sicherheit}}; \text{Fallbeschreibung}_{\text{Regaloptimierung}}; \\ \text{betrifftProjektart}; \\ \{\text{projektWirdDurchgefuehrtVon}; \text{projektWirdBearbeitetVon}\}; \end{array} \right) \\ & \quad \text{sim}_{\text{instanz}} \\ &= 1,0 + 1,0 + 1,0 \\ &= 3,0 \end{aligned}$$

4.2 Kritische Würdigung der Implementierung des Prototyps

Im Kapitel 2.4.2.2 wurde der Ähnlichkeitsalgorithmus von BERGENRODT/KOWALSKI vorgestellt, mit dem auch heterogene Fallbeschreibungen miteinander verglichen werden können.

Um eine kritische Würdigung dieses Algorithmus vorzunehmen, bedarf es zunächst einer Konzeptualisierung des Begriffs „Ähnlichkeit“.²⁷² LADYMAN verwendet den Begriff „Ähnlichkeit“ im Sinne einer Kategorisierung der Welt in verschiedene Arten.²⁷³ Der Autor verweist auf den Fortschritt diverser Wissenschaften, deren Geschichte aus einer Reihe von Neuklassifizierungen besteht.²⁷⁴ Ein System zur Einteilung von diversen Dingen²⁷⁵ in Arten²⁷⁶ bezeichnet er als eine Taxonomie.

BERGENRODT/KOWALSKI vergleichen mittels ihres Ähnlichkeitsalgorithmus verschiedene Fallbeschreibungen, die aus diversen Klassifizierungen von Dingen bestehen.²⁷⁷

In einem weiteren Schritt gilt es zu bewerten, ob der hier eingesetzte Ähnlichkeitsmaßstab als „gut“ bezeichnet werden kann. Da die Interpretation des Begriffs „gut“ überwiegend subjektiv festgelegt wird, ist eine eindeutige Bewertung des Ähnlichkeitsmaßstabs schwierig. Die Autoren verweisen auf eine Interpretationsweise, die einen Ähnlichkeitsmaßstab als „gut“ bewertet, wenn die Ergebnisse seiner Anwendung keine überraschenden Werte aufzeigen, die intuitiven Einschätzungen den Ergebnissen entsprechen und sich dieser Zusammenhang plausibel nachvollziehen lässt.²⁷⁸ Dies ist kritisch zu beurteilen, da die Autoren nicht erwähnen, wie im Falle einer „starken“ Abweichung der Ergebnisse von den intuitiven Einschätzungen vorgegangen werden soll.²⁷⁹ Wo die Grenze zwischen plausibel nachvollziehbaren und von der eigenen Intuition abweichenden Ergebnissen liegt, wird ebenso nicht festgelegt.

BERGENRODT/KOWALSKI verwenden den Ähnlichkeitsalgorithmus von BEIBEL als Grundlage für ihre eigene Implementierung eines ontologiegestützten CBR-Systems. Die Autoren erweitern den Algorithmus um die Ähnlichkeit von Projektbeschreibungen, die überwiegend auf natürlichsprachlicher Weise („qualitativ“) verfasst wurden. Sie lösen mittels ihres Ähnlichkeitsalgorithmus die Kardinalitäts-²⁸⁰ und Rekursivitätsprobleme²⁸¹ des Ähnlichkeitsalgorithmus von BEIBEL.

Die Berechnung der Ähnlichkeit zweier Fallbeschreibungen birgt jedoch noch einige Defizite. Die Anzahl der Relationen und Attribute im Nachbereich einer Instanz wird nicht in die Ähnlichkeitsberechnung einbezogen. Aussagen mit unterschiedlichem Wissensgehalt können mit den gleichen

272) Es gilt anzumerken, dass der Begriff „Ähnlichkeit“ von Person zu Person unterschiedlich definiert werden kann.

273) Vgl. LADYMAN (2007), S. 304. Der Inhalt des gesamten Absatzes bezieht sich auf diese Quelle.

274) Beispielsweise lässt sich in der Chemie eine „Säure“ der Klasse der „Flüssigkeiten“ zuordnen. Im weiteren Verlauf wurde die Säure weiter klassifiziert; vgl. LADYMAN (2007), S. 304.

275) Unter dem Begriff „Ding“ werden einzelne Subjekte (z. B. Tiere, Individuen), Objekte (z. B. Gegenstände) sowie Eigenschaften (z. B. basisch, sauer) verstanden.

276) Der Begriff „Arten“ ist synonym zu den Begriffen „Sorten“, „Klassen“ und „Kategorien“ zu verstehen.

277) Beispielsweise werden „Englischkenntnisse“ und „Niederländischkenntnisse“ unter dem Begriff „Linguistische Kompetenz“ vereinigt; vgl. BERGENRODT/KOWALSKI (2015), S. 64.

278) Vgl. BERGENRODT/KOWALSKI (2015), S. 108.

279) Einem unerfahrenen Mitarbeiter, der neu in einem Unternehmen ist und zuvor keine Projekte durchgeführt hat, gelingt beispielsweise eine intuitive Einschätzung der Ergebnisse nicht vollständig, sodass seine Bewertung der Ergebnisse kein „guter“ Anhaltspunkt für eine Bewertung des Ähnlichkeitsmaßstabs darstellt. Daher ist eine Evaluation durch Experten unter der Annahme, dass sie genügend Erfahrungen besitzen, ratsam.

280) Die Limitation hinsichtlich der Kardinalitäten von Attributen und Relationen stellen ein Problem im Anwendungsfall „Transportlogistik“ dar; vgl. BERGENRODT/KOWALSKI (2015), S. 26 f.

281) BEIBELS Algorithmus sieht keine Rekursion vor. Relationen, die nicht unmittelbar zur Fallbeschreibung gehören, werden nicht berücksichtigt; vgl. BERGENRODT/KOWALSKI (2015), S. 26.

Gewichten versehen werden.²⁸² Um den unterschiedlichen Wissensgehalt von Attributen und Relationen anzupassen, ist eine Verwendung von Gewichten denkbar. Die Autoren weisen zwar auf eine solche Methode hin (entropiebasierte Gewichte), verzichten jedoch aufgrund einer gesteigerten Komplexität des Algorithmus darauf.²⁸³

Der vorgestellte Ähnlichkeitsalgorithmus wurde im ontologiegestützten CBR-System jCORA implementiert. BERGENRODT/KOWALSKI zeigen anhand der Berechnungsbeispiele in den Kapiteln 6.3 und 6.4, dass ein Vergleich zwischen komplexen Fallbeschreibungen funktioniert. Jedoch fehlt ein umfangreicher Vergleich verschiedenster Fälle.²⁸⁴

Grundlage der Vergleiche bilden Ontologien. Der Aufbau einer heterogenen Fallbasis wird dadurch ermöglicht, dass einzelne Fälle mithilfe von lokalen Ontologien separat gespeichert werden.²⁸⁵ Das CBR-System jCORA ist zwar unabhängig von der Domäne, jedoch hängt seine Nützlichkeit von der verwendeten Domänenontologie ab. Der Aufbau einer Domänenontologie ist in jCORA leider nicht möglich.²⁸⁶ Stattdessen ist ein externer Ontologie-Editor zur Ontologierstellung notwendig.²⁸⁷ Es gilt: „Je vollständiger eine Domänenontologie ist, desto mehr Aussagen können vom eingesetzten OWL-Reasoner automatisch inferiert werden – die Anzahl notwendiger Nutzereingaben verringert sich.“²⁸⁸ Lediglich die spezielle Falleingabe (für ein konkretes Projekt) erfolgt in jCORA.²⁸⁹ Werden in der Domänenontologie Änderungen vorgenommen, so muss das CBR-System erneut gestartet werden. Ein automatisches Laden der Modifizierungen, ohne das vorherige Schließen des CBR-Systems jCORA, würde seine Bedienung wesentlich erleichtern.

Die Performance des CBR-Systems erweist sich beim Finden ähnlicher Fälle als unbefriedigend. Aufgrund des hohen Rechenaufwands beim Vergleich komplexer Fallbeschreibungen kann es zu längeren Wartezeiten kommen. Das CBR-System benötigt mehrere Sekunden zur Ergebnisausgabe. Die Akzeptanz der Wartezeiten auf der Nutzerseite erscheint fraglich.²⁹⁰

Neben dem hohen Rechenaufwand gilt es auch, die benötigte Zeit für die Falleingabe zu bemängeln. BERGENRODT/KOWALSKI weisen darauf hin, dass eine Beschleunigung der Falleingabe notwendig ist und führen verschiedene Möglichkeiten zur Erhöhung der Praxistauglichkeit des CBR-Systems jCORA an.²⁹¹

282) Liegen zwei Fälle „Projekt A“ und „Projekt B“ in der Fallbasis vor, führt ein Vergleich dieser Instanzen der Klasse „Projekt“ beispielsweise bezüglich der Relation „betrifftTransport“ zum Ähnlichkeitswert 0,9 und hinsichtlich der Relation „hatProjektvolumen“ zum Ähnlichkeitswert 0,2. Die Gesamtähnlichkeit ergibt sich aus dem Durchschnitt der zwei Ähnlichkeitswerte und beträgt 0,55. Der Wissensgehalt ist jedoch bezüglich der Relation „betrifftTransport“ höher, da mehr Instanzen (Transportabschnitt 1, Transportabschnitt 2, Transportmittel 1 und Transportmittel 2) subsumiert werden als bei der Relation „hatProjektvolumen“. Intuitiv betrachtet, ist der Wissensgehalt – und dadurch die Aussagekraft – höher bei „betrifftTransport“ als bei „hatProjektvolumen“; vgl. BERGENRODT/KOWALSKI/ZELEWSKI (2015), S. 537 ff.

283) Vgl. BERGENRODT/KOWALSKI (2015), S. 109.

284) Vgl. BERGENRODT/KOWALSKI (2015), S. 108.

285) Vgl. BERGENRODT/KOWALSKI (2015), S. 110. Der Inhalt des nächsten Satzes bezieht sich ebenfalls auf diese Quelle.

286) Eine Ausnahme bildet die Erstellung weiterer Relationen.

287) Vgl. BERGENRODT/KOWALSKI (2015), S. 111.

288) BERGENRODT/KOWALSKI (2015), S. 110.

289) Bei der Falleingabe gilt es zu beachten, eine möglichst aussagekräftige und eindeutige Bezeichnung für die Fall-IDs zu verwenden, um im Falle einer umfangreichen Datenbank das Wiederfinden von Projekten zu erleichtern.

290) Vgl. BERGENRODT/KOWALSKI (2015), S. 110.

291) Vgl. BERGENRODT/KOWALSKI (2015), S. 112.

Außerdem finden im CBR-System nach einer Falleingabe weder eine automatische Speicherung noch eine Speichererinnerung statt. Eine Notifikation für den Nutzer würde den Vorgang einer erneuten Falleingabe, aufgrund einer versäumten Speicherung, verhindern.

Zu bemängeln gilt es ebenfalls, dass es keine „Return“-Möglichkeit gibt, mit der eine Aktion zurückgesetzt werden kann. Bereits hinzugefügte Klassen und Instanzen können nicht mehr entfernt oder geändert werden.

Insgesamt bedürfen die verschiedenen Bearbeitungsfunktionen des CBR-Systems jCORA weiterer Forschungs- und Entwicklungsarbeiten, um seine Praxistauglichkeit nachhaltig zu steigern.

Bezüglich des Anwendungsbeispiels gilt es kritisch zu hinterfragen, ob die Fallbeschreibungen „Pricingtool“ und „Sicherheit“ tatsächlich vollkommen ähnlich sind. Grund dieser Kritik ist der Ähnlichkeitswert in der Höhe von 100 %, der sich sowohl aus den manuell durchgeführten als auch aus den maschinellen Berechnungen ergibt. Die Fälle beziehen sich nicht nur auf unterschiedliche Projektarten, sondern umfassen auch verschiedene Mitarbeiter, die darüber hinaus unterschiedliche Kompetenzen vorweisen. Dennoch gibt das System einen Ähnlichkeitswert von 100 % wieder. Dies mutet prima facie keineswegs als überzeugend an.

5 Fazit und Ausblick

Der Verlust vorhandenen Erfahrungswissens stellt ein fundamentales Problem für Unternehmen verschiedenster Branchen dar. Für bereits durchgeführte Projekte liegt oftmals umfangreiches Erfahrungswissen vor. Dieses Erfahrungswissen ist häufig personengebunden oder liegt in Form von nicht oder kaum strukturierten Textdokumenten vor. Es bedarf einer Technik, mit der dieses Erfahrungswissen aus Projekten strukturiert aufbereitet, archiviert und wiederverwendet werden kann, um nach der Beendigung eines Projekts das erlangte Erfahrungswissen nicht zu verlieren.

Das intendierte Ergebnis dieses Projektberichts war zum einen die Entwicklung einer Ontologie für die Domäne „Projektmanagement“ und zum anderen die Integration dieser Ontologie in ein ontologiegestütztes CBR-System wie jCORA.

Die Grundlage des CBR-Systems jCORA bildet eine Projektmanagement-Ontologie, die mittels der sieben Schritte zur Ontologieerstellung nach NOY/MCGUINNESS konstruiert wurde. Zunächst wurde der Anwendungsbereich erfasst. Anschließend wurden relevante Begriffe der Projektmanagementdomäne gesammelt²⁹², den jeweiligen Klassen zugeordnet und hierarchisiert. Danach wurden die einzelnen Sloteseigenschaften und Instanzen spezifiziert. Dieser Prototyp der Projektmanagement-Ontologie wurde anschließend in das CBR-System jCORA integriert. Um das Auffinden von ähnlichen Fällen zu demonstrieren, wurde die Fallbasis in jCORA mit drei exemplarischen Fällen „gefüllt“. Ein Vergleich eines neuen Falls mit zwei bereits in der Fallbasis vorhandenen alten Fällen demonstrierte die Funktionsfähigkeit des prototypischen CBR-Systems jCORA. Anhand eines manuell durchgeführten Berechnungsbeispiels wurde das maschinell berechnete Ergebnis validiert.

Zum umfangreichen Einsatz des prototypischen CBR-Systems jCORA in der Praxis müsste eine Verbesserung seiner Anwendungsperformance erfolgen.²⁹³ Ein weiterer Punkt zur Steigerung der Praxis-tauglichkeit umfasst die Internationalisierung der Software. Durch die wachsende Globalisierung sind viele Unternehmen international tätig. Dies beinhaltet die Durchführung von internationalen Projekten mit verschiedensprachigen Mitarbeitern. Daher wird die Erstellung der Projektmanagement-Ontologie und die Falleingabe in jCORA in englischer Sprache für einen höheren Praxisbezug empfohlen.

292) Zur Ermittlung der relevanten Begriffe wurden u. a. bereits bestehende Ontologien herangezogen.

293) Vgl. Kapitel 4.3.

Literaturverzeichnis

Vorbemerkungen:

- Alle Quellen werden im Literaturverzeichnis wie folgt aufgeführt: In der ersten Zeile wird der *Referenztitel* der Quelle angegeben. Er entspricht der Form, die im Text Verwendung findet, wenn auf die Quelle hingewiesen wird.
- Bei der Vergabe der Referenztitel wird bei *einem* Autor dessen Nachname, gefolgt von dem Erscheinungsjahr der Quelle in Klammern, verwendet. Existieren *zwei* oder *drei* Autoren, werden diese getrennt von einem Schrägstrich („/“) aufgeführt. Bei mindestens *vier* Autoren werden nur die ersten drei Autoren mit dem Zusatz „et al.“ aufgeführt.
- Die Quellen werden *lexikografisch* nach Maßgabe der Namen ihrer Autoren geordnet.
- Bei Quellen mit *gleichen Autoren* werden Quellen mit früheren Erscheinungsdaten vor Quellen mit neueren Erscheinungsdaten angeführt.
- Zwischen Quellen, die sich hinsichtlich ihrer Autoren und Erscheinungsdaten nicht unterscheiden, wird durch *Zusätze* wie „a“ und „b“ unterschieden.
- Hinsichtlich der *Schreibweise* von Beitragstiteln wird die Groß- und Kleinschreibung so übernommen, wie sie sich im Original befindet. Ausnahmen sind Majuskeln in und Fettdruck von Beitragstiteln, die nicht übernommen, sondern durch normal formatierte Schrift ersetzt werden.
- Zu *Internetquellen* wird die dafür verantwortliche Instanz aufgeführt. Dies können sowohl natürliche als auch juristische Personen sein. Für Internetquellen werden die zum Zugriffsdatum gültige Internetadresse (URL) und das Zugriffsdatum angegeben.

AAMODT/PLAZA (1994)

Aamodt, A.; Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and Systems Approaches. In: AI Communications, IOS Press, Vol. 7 (1994), No. 1, S. 39-59.

ABELS/AHLEMANN/HAHN et al. (2006)

Abels, S.; Ahlemann, F.; Hahn, A.; Hausmann, K.; Strickmann, J.: PROMONT – A Project Management Ontology as a Reference for Virtual Project Organizations. In: Meersman, R.; Tari, Z.; Herrero, P. et al. (Hrsg.): On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, 29.10.-03.11.2006 in Montpellier, Proceedings, Part I. Berlin - Heidelberg 2006, S. 813-823.

ACHANANUPARP/HU/ZHOU et al. (2008)

Achananuparp, P.; Hu, X.; Zhou, X.; Zhang, X.: Utilizing Semantic, Syntactic, and Question Category Information for Automated Digital Reference Services. In: Buchanan, G.; Masoodian, M.; Chunningjam, S.J. (Hrsg.): Digital Libraries: Universal and Ubiquitous Access to Information. Berlin - Heidelberg 2008, S. 203-214.

AHLEMANN/TEUTEBERG/BRUNE (2006)

Ahlemann, F.; Teuteberg, F.; Brune, G.: Ontologie-basierte Attributierung von Informationsmodellen – Grundlagen und Anwendungsgebiete. ISPRI-Arbeitsbericht, Nr. 01/2006. ISPRI – Forschungszentrum für Informationssysteme in Projekt- und Innovationsnetzwerken, Universität Osnabrück. Osnabrück 2006.

AMBÜHL (1994)

Ambühl, H.: Metaphysik und Ontologie bei Aristoteles. In: Freiburger Zeitschrift für Philosophie und Theologie, Jg. 41 (1994), Heft 1-2, S. 223-228.

ARAMO-IMMONEN (2009)

Aramo-Immonen, H.: Project Management Ontology – The Organizational Learning Perspective. Dissertation, Tampere University of Technology, Publication 836. Tampere 2009.

ASSALIM/LENNE/DEBRAY (2010)

Assalim, A.A.; Lenne, D.; Debray, B.: Heterogeneity in Ontological CBR Systems. In: Montani, S.; Jain, L.C. (Hrsg.): Successful Case-based Reasoning Applications – I. Berlin - Heidelberg 2010, S. 97-116.

BEIERLE/KERN-ISBERNER (2019)

Beierle, C.; Kern-Isberner, G.: Methoden wissensbasierter Systeme – Grundlagen, Algorithmen, Anwendungen. 6. Aufl., Wiesbaden 2019.

BEIBEL (2011)

Beißel, S.: Ontologiegestütztes Case-Based Reasoning – Entwicklung und Beurteilung semantischer Ähnlichkeitsindikatoren für die Wiederverwendung natürlichsprachlich repräsentierten Projektwissens. Dissertation, Universität Duisburg-Essen 2011. Wiesbaden 2011.

BENJAMIN/AKELLA/MALEK et al. (2005)

Benjamin, P.; Akella, K.V.; Malek, K.; Fernandes, R.: An ontology-driven framework for process-oriented applications. In: Kuhl, M.E.; Steiger, N.M.; Armstrong, F.B.; Joines, J.A. (Hrsg.): Proceedings of the 2005 Winter Simulation Conference, WSC'05, 04.-07.12.2005 in Orlando. New York 2005, S. 2355-2363.

BERGENRODT/KOWALSKI (2015)

Bergenrodt, D.; Kowalski, M.: Konzipierung, Implementierung und kritische Evaluierung einer Projektwissensbank auf Basis von semantischen Methoden der Künstlichen Intelligenz. OrGoLo-Projektbericht Nr. 31. Institut für Produktion und Industrielles Informationsmanagement, Universität Duisburg-Essen (Campus Essen). Essen 2015.

BERGENRODT/KOWALSKI/ZELEWSKI (2015)

Bergenrodt, D.; Kowalski, M.; Zelewski, S.: Prototypische Implementierung des ontologiegestützten CBR-Tools jCORa. In: Zelewski, S.; Akca, N.; Kowalski, M. (Hrsg.): Organisatorische Innovationen mit Good Governance und Semantic Knowledge Management in Logistik-Netzwerken – Wissenschaftliche Grundlagen und Praxisanwendungen. Berlin 2015, S. 475-553.

BERGMANN (2002)

Bergmann, R.: Experience Management – Foundations, Development Methodology, and Internet-Based Applications. Berlin - Heidelberg - New York 2002.

BIESALSKI/ABDECKER (2005)

Biesalski, E.; Abdecker, A.: Human Resource Management with Ontologies. In: Althoff, K. D.; Dengel, A.; Bergmann, R.; Nick, M.; Roth-Berghofer, T. (Hrsg.): Professional Knowledge Management – Third Biennial Conference, WM 2005, 10.-13.04.2005 in Kaiserslautern, Revised Selected Papers. Berlin - Heidelberg 2005, S. 499-507.

BODEA (2007)

Bodea, C.-N.: An Innovative System for Learning Services in Project Management. In: Park, M.; Park, J. (Hrsg.): IEEE International Conference on Service Operations and Logistics, and Informatics – Perceived Interactivity in Online Retail Services: Approach from the Features, 27.-29.08.2007 in Philadelphia. o. O. 2007, S. 1-5 (eigene Paginierung).

BODEA/ELMAS/TANASESCU et al. (2010)

Bodea, C.-N.; Elmas, C.; Tanasescu, A.; Dascalu, M.: An Ontological-Based Model for Competences in Sustainable Development Projects: A Case Study for Projekt's Commercial Activities. In: Economic Interferences, Vol. 12 (2010), No. 27, S. 177-189.

BRANDT/EHRENBERG/ALTHOFF et al. (2001)

Brandt, M.; Ehrenberg, D.; Althoff, K.-D.; Nick, M.: Ein fallbasierter Ansatz für die computergestützte Nutzung von Erfahrungswissen bei der Projektarbeit. In: Buhl, H.U.; Huther, A.; Reitwiesher, B. (Hrsg.): Information Age Economy – 5. Internationale Tagung Wirtschaftsinformatik 2001. Berlin - Heidelberg 2001, S. 251-264.

BURGHARDT (2013)

Burghardt, M.: Einführung in Projektmanagement – Definition, Planung, Kontrolle, Abschluss. 6. Aufl., Erlangen 2013.

CAMBRIA/WHITE (2014)

Cambria, E.; White, B.: Jumping NLP Curves: A Review of Natural Language Processing Research. In: IEEE Computational Intelligence Magazine, Vol. 9 (2014), No. 2, S. 48-57.

CHANGCHIEN/LIN (2005)

Changchien, S.W.; Lin, M.-C.: Design and implementation of a case-based reasoning system for marketing plans. In: Expert Systems with Applications, Vol. 28 (2005), No. 1, S. 43-53.

CHOI/PARK (2005)

Choi, S.; Park, H.: Extracting Semantic Taxonomies of Nouns from a Korean MRD Using a Small Bootstrapping Thesaurus and a Machine Learning Approach. In: Montoyo, A.; Munoz, R.; Métais, E. (Hrsg.): Natural Language Processing and Information Systems – 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, 15.-17.06.2005 in Alicante, Proceedings. Berlin - Heidelberg 2005, S. 1-9.

CURTS/CAMPBELL (2005)

Curts, R.J.; Campbell, D.E.: Building An Ontology For Command & Control – 10th International Command and Control Research and Technology Symposium – The Future of C2. Online-Quelle vom 17.03.2005 unter der URL „<https://apps.dtic.mil/dtic/tr/fulltext/u2/a464304.pdf>“, letzter Zugriff am 08.05.2020.

DARPA (2003)

Darpa: The Darpa Agent Markup Language Homepage. Online-Quelle vom 05.03.2003 unter der URL „<http://www.daml.org/about.html>“, letzter Zugriff am 09.09.2020.

DEY (2002)

Dey, P.K.: Project Risk Management: A Combined Analytic Hierarchy Process and Decision Tree Approach. In: Cost Engineering, Vol. 44 (2002), No. 3, S. 13-26.

DICKE (2007)

Dicke, R.: Strategische Unternehmensplanung mit Hilfe eines Assumption-based-Truth-Maintenance-Systems (ATMS) – Formalisierung eines Kontingenzansatzes in Prädikatenlogik und Anpassungsplanung nach dem Net-Change-Prinzip. Dissertation, Universität Duisburg-Essen (Campus Essen) 2006. Wiesbaden 2007.

DIN 69901-5 (2009)

DIN 69901-5: Projektmanagement – Netzplantechnik; Beschreibungen und Begriffe. Berlin 2009.

DING/LONSDALE/EMBLEY et al. (2007)

Ding, Y.; Lonsdale, D.; Embley, D.W.; Hepp, M.; Xu, L.: Generating Ontologies Via Language Components and Ontology Reuse. In: Kedad, Z.; Lammari, N.; Métais, E.; Meziane, F.; Rezgui, Y. (Hrsg.): Natural Language Processing and Information Systems – 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, 27.-29.06.2007 in Paris, Proceedings. Berlin - Heidelberg 2007, S. 131-142.

DONG/HUSSAIN/CHANG (2011)

Dong, H.; Hussain, F.K.; Chang, E.: ORPMS: An Ontology-based Real-time Project Monitoring System in the Cloud. In: Journal of Universal Computer Science, Vol. 17 (2011), No. 8, S. 1161-1182.

DORN/NAZ/PICHLMAIR (2007)

Dorn, J.; Naz, T.; Pichlmair, M.: Ontology Development for Human Resource Management. In: Sary, C.; Barachini, F.; Al-Hawamdeh, S. (Hrsg.): Knowledge Management: Innovation, Technology and Cultures – Proceedings of the 2007 International Conference on Knowledge Management, 27.-28.08.2007 in Wien. Singapore 2007, S. 109-120.

ENGEL/TAMDJIDI/QUADEJACOB (2008)

Engel, C.; Tamdjidi, A.; Quadejacob, N.: Ergebnisse der Projektmanagement Studie 2008 – Erfolg und Scheitern im Projektmanagement. Online-Quelle vom 01.12.2008 unter der URL „https://www.gpm-ipma.de/fileadmin/user_upload/GPM/KnowHow/Ergebnisse_Erfolg_und_Scheitern-Studie_2008.pdf“, letzter Zugriff am 09.03.2022.

FAYYAD/PIATETSKY-SHAPIRO/SMYTH (1996)

Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: From Data Mining to Knowledge Discovery in Databases. In: AI Magazine, Vol. 17 (1996), No. 3, S. 37.

FELBERT (1998)

Felbert, D.: Wissensmanagement in der unternehmerischen Praxis. In: Pawlowsky, P. (Hrsg.): Wissensmanagement – Erfahrungen und Perspektiven. Wiesbaden 1998, S. 119-141.

FENSEL (2004)

Fensel, D.: Ontologies – A Silver Bullet for Knowledge Management and Electronic Commerce. 2. Aufl., Berlin - London 2004.

FINNIE/SUN (2002)

Finnie, G.; Sun, Z.: Similarity and Metrics in Case-Based Reasoning. In: International Journal of Intelligent Systems, Vol. 17 (2002), No. 3, S. 273-287.

FONG/KWOK (2009)

Fong, P.S.W.; Kwok, C.W.C.: Organizational Culture and Knowledge Management Success at Project and Organizational Levels in Contracting Firms. In: Journal of Construction Engineering and Management, Vol. 135 (2009), No. 12, S. 1348-1356.

FRANK/SCHÖNERT (2001)

Frank, U.; Schönert, S.: Wissensmanagement in Projekten – Status quo und informationstechnische Unterstützungspotentiale. In: Projektmanagement Aktuell, Jg. 12 (2001), Heft 4, S. 25-33.

FREUDENTHALER (2008)

Freudenthaler, B.: Case-based Reasoning (CBR) – Grundlagen und ausgewählte Anwendungsgebiete des fallbasierten Schließens. Saarbrücken 2008.

FRIEDMANN (2006)

Friedmann, K.: Deutsche IT-Manager planen schlecht. Gemeinschaftsstudie der Technischen Universität München (TUM) und der (Fachzeitschrift) Computerwoche zum Thema IT-Management. Online-Quelle vom 02.02.2006 unter der URL „<https://www.tcw.de/uploads/html/news/pressespiegel/IT-Management.pdf>“ und „<https://www.computerwoche.de/a/deutsche-it-manager-planen-schlecht,571685>“, letzter Zugriff (zweite URL) am 09.03.2022.

GANZHA/MESJASZ/PAPRZYCHI et al. (2014)

Ganzha, M.; Mesjasz, M.M.; Paprzychi, M.; Ouedraogo, M.: Inserting „Brains“ into Software agents – preliminary considerations. In: Fortino, G.; Di Fatta, G.; Li, W.; Ochoa, S.; Cuzzocrea, A. (Hrsg.): Internet and Distributed Computing Systems, IDCS 2014, 7th International Conference, 22.-24.09.2014 in Calabria. Cham 2014, S. 3-14.

GLEICH (2008)

Gleich, B.: Ontologie-Methodologien und Ontologie Design. In: Bauer, B.; Fischer, W.; Lautenbacherm, F.; Roser, S. (Hrsg.): Konzepte und Techniken für das Semantic Web – WS 2007/2008. Seminarband, Universität Augsburg, Institut für Informatik, Programmierung verteilter Systeme. Augsburg 2008, S. 105-124.

GÓMEZ-PÉREZ/CORCHO/FERNÁNDEZ-LÓPEZ (2004)

Gómez-Pérez, A.; Corcho, O.; Fernández-López, M.: Ontological Engineering – with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. 2. Aufl., London 2004.

GRIMES (2008)

Grimes, S.: Unstructured Data and the 80 Percent Rule – Breakthrough Analysis – Seth Grimes on NLP, text analytics, sentiment analysis, BI, visualization and more. Online-Quelle vom 01.08.2008 unter der URL „<http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/>“, letzter Zugriff am 09.03.2022.

GRUBER (1993)

Gruber, T.R.: A translation approach to portable ontology specifications. In: Knowledge Acquisition, Vol. 5 (1993), No. 2, S. 199-220.

GRÜNINGER/FOX (1995)

Grüniger, M.; Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. Online-Quelle vom 13.04.1995 unter der URL „<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=E470A23FC15195E27EF3A5B2F18D2952?doi=10.1.1.44.8723&rep=rep1&type=pdf>“, letzter Zugriff am 09.03.2022.

HAARMANN (2014)

Haarmann, B.: Ontology On Demand – Vollautomatische Ontologieerstellung aus deutschen Texten mithilfe moderner Textmining-Prozesse. Dissertation, Universität Bochum 2014. Berlin 2014.

HEBELER/FISHER/BLACE et al. (2009)

Hebeler, J.; Fisher, M.; Blace, R.; Perez-Lopez, A.: Semantic Web Programming, Indianapolis 2009.

HORRIDGE/BECHHOFFER (2011)

Horridge, M.; Bechhofer, S.: The OWL API: A Java API for OWL Ontologies. In: Semantic Web, Vol. 2 (2011), No. 1, S. 11-21.

HORROCKS/PATEL-SCHNEIDER/VAN HARMELEN (2003)

Horrocks, I.; Patel-Schneider, P.; van Harmelen, F.: From SHIQ and RDF to OWL: the Making of a Web Ontology Language. In: Journal of Web Semantics, Vol. 1 (2003), No. 1, S. 7-26.

HORST/BACHMANN/HESSE (2012)

Horst, B.; Bachmann, A.; Hesse, W.: Ontologien als ein Mittel zur Wiederverwendung von Domänen-spezifischem Wissen in der Software-Entwicklung. In: Sinz, E.J.; Schür, A. (Hrsg.): Modellierung 2012. Bamberg 2012, S. 171-186.

HÜBNER/SCHAAL/DÜSTERHÖFT (2019)

Hübner, A.; Schaal, K.N.; Düsterhöft, T.: Regalplatzoptimierung im Einzelhandel. In: Wirtschaftswissenschaftliches Studium, Jg. 48 (2019), Heft 9, S. 4-10.

HUGHES (2010)

Hughes, R.: Project Management Process Ontologies: A Proof of Concept. In: UK Academy for Information Systems Conference Proceedings 2010, Beitrag 30, S. 1-20 (gemäß eigener Paginierung).

HUMPL (2004)

Humpl, B.: Transfer von Erfahrungen – Ein Beitrag zur Leistungssteigerung in projektorientierten Organisationen. Dissertation, Technische Universität Graz 2002. Wiesbaden 2004.

ISO/IEC/IEEE 29148 (2011)

ISO/IEC/IEEE 29148: Systems and software engineering – Life cycle processes – Requirements engineering. Schweiz 2011.

JACKSON/MOULINIER (2002)

Jackson, P.; Moulinier, I.: Natural Language Processing for Online Applications – Text Retrieval, Extraction and Categorization. 5. Aufl., Amsterdam - Philadelphia 2002.

JAKUS/MILUTINOVIĆ/OMEROVIĆ (2013)

Jakus, G.; Milutinović, V.; Omerović, S.: Concepts, Ontologies, and Knowledge Representation. New York et al. 2013.

JANSEN (2008)

Jansen, L.: Kategorien: Die *top level* Ontologie – Wissen strukturieren für den Informatik-Einsatz. In: Jansen, L.; Smith, B. (Hrsg.): Biomedizinische Ontologie – Wissen strukturieren für den Informatik-Einsatz. Zürich 2008, S. 85-112.

KABEL (2020)

Kabel, P.: Dialog zwischen Mensch und Maschine – Conversational User Interfaces, intelligente Assistenten und Voice-Systeme. Wiesbaden 2020.

KAMARA/AUGENBROE/ANUMBA et al. (2002)

Kamara, J.M.; Augenbroe, G.; Anumba, C.J.; Carillo, P.M.: Knowledge management in the architecture, engineering and construction industry. In: Construction Innovation: Information, Process, Management, Vol. 2 (2002), No. 1, S. 53-67.

KOLODNER (1993)

Kolodner, J.: Case-based Reasoning. San Mateo 1993.

KOSKINEN (2004)

Koskinen, K.U.: Knowledge management to improve project communication and implementation. In: Project Management Journal, Vol. 35 (2004), No. 1, S. 13-19.

KOWALSKI/KLÜPFEL/ZELEWSKI et al. (2013)

Kowalski, M.; Klüpfel, H.; Zelewski, S.; Bergenrodt, D.; Saur, A.: Integration of Case-Based and Ontology-Based Reasoning for the Intelligent Reuse of Project-Related Knowledge. In: Clausen, U.; ten Hompel, M.; Klumpp, M. (Hrsg.): Efficiency and Logistics. Berlin - Heidelberg 2013, S. 289-299.

KOWALSKI/ZELEWSKI (2015)

Kowalski, M.; Zelewski, S.: Erstellung einer Verpackungsontologie mithilfe des Ontologie-Editors Protégé. In: Zelewski, S.; Akca, N.; Kowalski, M. (Hrsg.): Organisatorische Innovationen mit Good Governance und Semantic Knowledge Management in Logistik-Netzwerken – Wissenschaftliche Grundlagen und Praxisanwendungen. Berlin 2015, S. 593-614.

KULICHIKHINA/RUBAN (2013)

Kulichikhina, M.; Ruban, N.: Semantisches Wörterbuch der deutschen Sprache für maschinelle Sprachverarbeitungssysteme. In: Bergerová, H.; Schmidt, M.; Schuppener, G. (Hrsg.): Lexikologie und Lexikografie – Aktuelle Entwicklungen und Herausforderungen. Aussiger Beiträge, Germanistische Schriftenreihe aus Forschung und Lehre 7. Ústí nad Labem 2013, S. 103-118.

LADYMAN (2007)

Ladyman, J.: Ontological, Epistemological, and Methodological Positions. In: Kuipers, T.A.F. (Hrsg.): General Philosophy of Science – Focal Issues. Amsterdam 2007, S. 303-376.

LAVBIC/VASILECAS/RUPNIK (2010)

Lavbic, D.; Vasilecas, O.; Rupnik, R.: Ontology-Based Multi-Agent System to Support Business Users and Management. In: Technological and Economic Development of Economy, Vol. 16 (2010), No. 2, S. 327-347.

LEAKE (1996)

Leake, D.B.: CBR in Context: The Present and Future. In: Leake, D.B. (Hrsg.): Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park 1996, S. 3-30.

LEE/NG (2006)

Lee, S.-G.; Ng, Y.C.: Hybrid case-based reasoning for on-line product fault diagnosis. In: The International Journal of Advanced Manufacturing Technology, Vol. 27 (2006), Nos. 7-8, S. 833-840.

LEITNER (2006)

Leitner, J.: Extraktion von Ontologien aus natürlichsprachlichen Texten. In: Witte, R.; Mülle, J. (Hrsg.): Text Mining: Wissensgewinnung aus natürlichsprachigen Dokumenten. Karlsruhe 2006, S. 113-134.

LENZ/HÜBNER/KUNZE (1998)

Lenz, M.; Hübner, A.; Kunze, M.: Question Answering with Textual CBR. In: Andreasen, T.; Christiansen, H.; Larsen, H.L. (Hrsg.): Flexible Query Answering Systems – Third International Conference, FQAS'98, 13.-15.05.1998 in Roskilde, Proceedings. Berlin - New York 1998, S. 236-247.

LEPRATTI(2006)

Lepratti, R.: Advanced human-machine system for intelligent manufacturing – Some issues in employing ontologies for natural language processing. In: Journal of Intelligent Manufacturing, Vol. 17 (2006), No. 6, S. 653-666.

LIM/LIU/LEE (2011)

Lim, E.H.Y.; Liu, J.N.K.; Lee, R.S.T.: Knowledge Seeker – Ontology Modelling for Information Search and Management – A Compendium. Berlin - Heidelberg 2011.

MANNING/RAGHAVAN/SCHÜTZE (2009)

Manning, C.D.; Raghavan, P.; Schütze, H.: An Introduction to Information Retrieval. 4. Aufl., Cambridge 2009.

MCDANIEL/STOREY/SUGUMARAN (2018)

McDaniel, M.; Storey, V.C.; Sugumaran, V.: Assessing the quality of domain ontologies: Metrics and an automated ranking system. In: Data & Knowledge Engineering, Vol. 115 (2018), S. 32-47.

MERTENS (1995)

Mertens, P.: Wirtschaftsinformatik – Von den Moden zum Trend. In: König, W. (Hrsg.): Wirtschaftsinformatik '95 – Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit. Berlin - Heidelberg 1995, S. 25-64.

MERTENS/BODENDORF/KÖNIG et al. (2017)

Mertens, P.; Bodendorf, F.; König, W.; Schumann, M.; Hess, T.; Buxmann, P.: Grundzüge der Wirtschaftsinformatik. 12. Aufl., Berlin - Heidelberg 2017.

MUSEN (1992)

Musen, M.A.: Dimensions of Knowledge Sharing and Reuse. In: Computers and Biomedical Research, Vol. 25 (1992), No. 5, S. 435-467.

NAGEL (2009)

Nagel, R.: Technische Herausforderungen modellgetriebener Beherrschung von Prozesslebenszyklen aus der Fachperspektive: Von der Anforderungsanalyse zur Realisierung. Dissertation, Technische Universität Dortmund, Fakultät für Informatik 2009. Dortmund 2009.

NONAKA/TAKEUCHI (2012)

Nonaka, I.; Takeuchi, H.: Die Organisation des Wissens – Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen. 2. Aufl., Frankfurt - New York 2012.

NOY/CRUBÉZY/FEGERSON et al. (2003)

Noy, N.; Crubézy, M.; Fegerson, R. W.; Knublauch, H.; Tu, S. W.; Vendetti J.; Musen, M. A.: Protégé-2000: An Open-Source Ontology-Development and Knowledge-Acquisition Environment – AMIA 2003 Open Source Expo. In: AMIA Annual Symposium Proceedings Archive, Vol. 2003 (2003), S. 953.

NOY/MCGUINNESS (2001)

Noy, N.; McGuinness, D.: Ontology Development 101: A Guide to Creating Your First Ontology. Online-Quelle vom 01.01.2001 unter der URL „https://protege.stanford.edu/publications/ontology_development/ontology101.pdf“, letzter Zugriff am 09.03.2022.

NÜNNING/JÄGER (2015)

Nünning, M.; Jäger, S.: Schublade oder Praxis? – Eine Analyse zu den Verwendungsmöglichkeiten der Projektergebnisse. In: Zelewski, S.; Akca, N.; Kowalski, M. (Hrsg.): Organisatorische Innovationen mit Good Governance und Semantic Knowledge Management in Logistik-Netzwerken – Wissenschaftliche Grundlagen und Praxisanwendungen. Berlin 2015, S. 659-688.

O. V. (2004)

O. V.: OWL Web Ontology Language Overview – W3C Recommendation 10 February 2004. Online-Quelle vom 10.02.2004 unter der URL „<https://www.w3.org/TR/owl-features/>“, letzter Zugriff am 09.03.2022.

O. V. (2021)

O. V.: W3C Mission. Online-Quelle, © 2021, unter der URL „<https://www.w3.org/Consortium/mission>“, letzter Zugriff am 09.03.2022.

PAL/SHIU(2004)

Pal, S.K.; Shiu, S.C.K.: Foundations of Soft Case-Based Reasoning. New York 2004.

PFUHL (2003)

Pfuhl, M.: Case-Based Reasoning auf der Grundlage Relationaler Datenbanken – Eine Anwendung zur strukturierten Suche in Wirtschaftsnachrichten. Dissertation, Universität Marburg 2003. Wiesbaden 2003.

PÖHLAND/KORZETZ/SCHLEGEL (2013)

Pöhländ, R.; Korzetz, M.; Schlegel, T.: Semantische Modellierung und Klassifikation von Touch-Interaktionen. In: Schlegel, T. (Hrsg.): Multi-Touch – Interaktion durch Berührung. Berlin - Heidelberg 2013, S. 69-88.

PROBST/RAUB/ROMHARDT (2010)

Probst, G.; Raub, S.; Romhardt, K.: Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen. 6. Aufl., Frankfurt - Wiesbaden 2010.

PROJECT MANAGEMENT INSTITUTE (2013)

Project Management Institute, Inc.: A Guide to the Project Management Body of Knowledge – (PMBOK® Guide). 5. Aufl., Pennsylvania 2013.

RAJMAN/BESANÇON (1998)

Rajman, M.; Besançon, R.: Text Mining – Knowledge extraction from unstructured textual data. In: Rizzi, A.; Vichi, M.; Bock, H.-H. (Hrsg.): Advances in Data Science and Classification, Proceedings of the 6th Conference of the International Federation of Classification Societies (IFCS-98), 21.-24.07.1998 in Rom. Berlin et al. 1998, S. 473-480.

RECIO-GARCÍA et al. (2007)

Recio-García, J.A.; Díaz-Agudo, B.; Gonzáles-Calero, P.A.; Sánchez-Ruiz-Granados, A.: Ontology based CBR with jCOLIBRI. In: Ellis, R.; Allen, T.; Tuson, A. (Hrsg.): Applications and Innovations in Intelligent Systems XIV, Proceedings of AI-2006, the Twent-sixth SGA International Conference on Innovative Techniques and Applications of Artificial Intelligence. London 2007, S. 149-162.

RECIO-GARCÍA/GONZÁLES-CALERO/DÍAZ-AGUDO (2014)

Recio-García, J.A.; Gonzáles-Calero, P.A.; Díaz-Agudo, B.: jCOLIBRI2: A framework for building Case-based reasoning systems. In: Science of Computer Programming, Vol. 79 (2014), S. 126-145.

REICH/WEE (2006)

Reich, B.H.; Wee, S.Y.: Searching for Knowledge in the Pmbok® Guide. In: Project Management Journal, Vol. 37 (2006), No. 3, S. 11-26.

RICHTER (2008)

Richter, M.: Similarity. In: Perner, P. (Hrsg.): Case-Based Reasoning on Images and Signals. Berlin - Heidelberg - New York 2008, S. 2-90.

RICHTER/BENDER/KLINGER et al. (2008)

Richter, G.; Bender, K.; Klinger, M.; Herbolzheimer, C.: Projekte mit Launch Management auf Kurs halten – Warum IT-Großprojekte häufig kentern und Projekterfolg kein Glücksspiel ist. Studie 07/2008 Roland Berger Strategy Consultants. Asterdam et al. 2008. Online-Quelle unter der URL „<https://docplayer.org/3962333-Projekte-mit-launch-management-auf-kurs-halten.html>“, letzter Zugriff am 09.03.2022.

RICHTER/WEBER (2013)

Richter, M.; Weber, R.: Case-based reasoning – A Textbook. Heidelberg - New York - Dordrecht et al. 2013.

RICKLEFS/BLOMQVIST (2008)

Ricklefs, M.; Blomqvist, E.: Ontology-Based Relevance Assessment: An Evaluation of Different Semantic Similarity Measures. In: Meersman, R.; Tari, Z. (Hrsg.): On the Move to Meaningful Internet Systems: OTM 2008, 09.-14.11.2008 in Monterrey, Proceedings, Part II. Berlin - Heidelberg 2008, S. 1235-1252.

RUIZ-BERTOL/RODRÍGUEZ/DOLADO (2011)

Ruiz-Bertol, F.J.; Rodríguez, D.; Dolado, J.: Applying Rules to an Ontology for Project Management. In: Muñoz, C.C.; Places, Á.S. (Hrsg.): Jornadas de Ingeniería del Software y Bases de Datos (JISBD), 05.-07.09.2011 in A Coruña. A Coruña 2011, S. 257-262

RUPP (2014)

Rupp, C.: Requirements Engineering und -management – Aus der Praxis von klassisch bis agil. 6. Aufl., München 2014.

SCHELLE (2014)

Schelle, H.: Projekte zum Erfolg führen – Projektmanagement systematisch und kompakt. 7. Aufl., München 2014.

SCHUHBAUER/FUHR/WITTMANN (2008)

Schuhbauer, H.; Fuhr, T.; Wittmann, S.: Flexible Informationsstrukturen mit Ontologien. In: HMD Praxis der Wirtschaftsinformatik, Jg. 45 (2008), Heft 4, S. 97-105.

SCHWARZ/SMITH (2008)

Schwarz, U.; Smith, B.: Ontologische Relationen. In: Jansen, L.; Smith, B.; Bittner, T. (Hrsg.): Biomedizinische Ontologie – Wissen strukturieren für den Informatik-Einsatz. Zürich 2008, S. 155-172.

SHEEBA/KRISHNAN/BERNARD (2012)

Sheeba, T.; Krishnan, R.; Bernard, M.J.: An Ontology in Project Management Knowledge Domain. In: International Journal of Computer Applications, Vol. 56 (2012), No. 5, S. 1-7.

SNIDER/NISSEN (2003)

Snider, K.F.; Nissen, M.E.: Beyond the Body of Knowledge: A Knowledge-Flow Approach to Project Management Theory and Practice. In: Project Management Journal, Vol. 34 (2003), No. 2, S. 4-12.

STAAB (2011)

Staab, S.: Ontologies and similarity. In: Ram, A.; Wiratunga, N. (Hrsg.): Case-Based Reasoning Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011, 12.-15.09.2011 in London, Proceedings. Berlin - Heidelberg 2011, S. 11-16.

STUCKENSCHMIDT (2011)

Stuckenschmidt, H.: Ontologien – Konzepte, Technologien und Anwendungen. 2. Aufl., Dordrecht et al. 2011.

STUDER/BENJAMINS/FENSEL (1998)

Studer, R.; Benjamins, V.; Fensel, D.: Knowledge Engineering: Principles and methods. In: Data & Knowledge Engineering, Vol. 25 (1998), Nos. 1-2, S. 161-197.

TAN (1999)

Tan, A.-H.: Text Mining: The state of the art and the challenges. In: Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining PAKDD'99, Workshop on Knowledge Discovery from Advanced Databases (KDAD'99). o. O., S. 65-70.

THE STANDISH GROUP INTERNATIONAL (2015)

The Standish Group International, Inc.: CHAOS Report 2015. Online-Quelle unter der URL „https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf“, letzter Zugriff am 09.03.2022.

USCHOLD/KING/MORALEE et al. (1998)

Uschold, M.; King, M.; Moralee, S.; Zorgios, Y.: The Enterprise Ontology. In: The Knowledge Engineering Review, Vol. 13 (1998), No. 1, S. 31-89.

VAN HARMELEN/HORROCKS (2000)

van Harmelen, F.; Horrocks, I.: FAQs on OIL: the Ontology Inference Layer. In: IEEE Intelligent Systems, Vol. 15 (2000), No. 6, S. 69-72.

VAN LAARSCHOT (2005)

van Laarschot, R.: Ontology-based knowledge modelling in Dutch civil law. Report August 2005, Department of Artificial Intelligence, Vrije Universiteit Amsterdam. Amsterdam 2005 (auch als Online-Quelle unter der URL „https://www.researchgate.net/publication/228733348_Ontology-based_knowledge_modeling_in_dutch_civil_law“, letzter Zugriff am 09.03.2022).

VASIL/XHINA/NINKA et al. (2018)

Vasili, R.; Xhina, E.; Ninka, I.; Souliotis, T.: A Comparative Review of Text Mining & Related Technologies. In: Xhina, E.; Hoxha, K. (Hrsg.): RTA-CSIT 2018 – Recent Trends and Applications in Computer Science and Information Technology, Tirana 2018, S. 1-10.

WATERFELD/WEITEN/HAASE (2011)

Waterfeld, W.; Weiten, M.; Haase, P.: Ontology Management Infrastructures. In: Hepp, M.; de Leenheer, P.; de Moor, A.; Sure, Y. (Hrsg.): Ontology Management – Semantic Web, Semantic Web Services, and Business Applications. New York 2011, S. 59-87.

WATSON (2003)

Watson, I.: Applying Knowledge Management – Techniques for Building Corporate Memories. San Francisco 2003.

WITTE/GITZINGER (2008)

Witte, R.; Gitzinger, T.: Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: Domingue, J. (Hrsg.): The Semantic Web, 3rd Asian Semantic Web Conference, ASWC 2008, 08.-11.12.2008 in Bangkok, Proceedings. Berlin - Heidelberg 2008, S. 360-374.

WONG/LAFAYE/BOURSIER (2010)

Wong, S.M.; Lafaye, J.-Y.; Boursier, P.: Developing PMIS for Business Projects Based on Social Science Research Findings and Ontology Modelling. In: Nunes, M.B.; Isaías, P.; Powell, P. (Hrsg.): Proceedings of the IADIS International Conference on Information Systems, IADS'10, 18.-20.03.2010 in Porto. o. O. 2010, S. 283-291.

ZELEWSKI (1999)

Zelewski, S.: Ontologien zur Strukturierung von Domänenwissen – Ein Annäherungsversuch aus betriebswirtschaftlicher Perspektive. Arbeitsbericht Nr. 3, Institut für Produktion und Industrielles Informationsmanagement, Universität Duisburg-Essen, Campus Essen. Essen 1999.

ZELEWSKI (2005)

Zelewski, S.: Einführung in das Themenfeld „Ontologien“ aus informations- und betriebswirtschaftlicher Perspektive. In: Zelewski, S.; Alan, Y.; Alparslan, A.; Dittmann, L.; Weichelt, T. (Hrsg.): Ontologiebasierte Kompetenzmanagementsysteme – Grundlagen, Konzepte, Anwendungen. Berlin 2005, S. 115-228.

ZELEWSKI (2015)

Zelewski, S.: Ontologien als Grundlage für das Semantische Knowledge Management von projektbezogenem Erfahrungswissen. In: Zelewski, S.; Akca, N.; Kowalski, M. (Hrsg.): Organisatorische Innovationen mit Good Governance und Semantic Knowledge Management in Logistik-Netzwerken – Wissenschaftliche Grundlagen und Praxisanwendungen. Berlin 2015, S. 81-228.

ZELEWSKI/ALAN/ALPARSLAN et al. (2005)

Zelewski, S.; Alan, Y.; Alparslan, A.; Dittmann, L.; Weichelt, T. (Hrsg.): Ontologiebasierte Kompetenzmanagementsysteme – Grundlagen, Konzepte, Anwendungen. Berlin 2005.

ZELEWSKI/SCHÜTTE/SIEDENTOPF(2001)

Zelewski, S.; Schütte, R.; Siedentopf, J.: Ontologien zur Repräsentation von Domänen. In: Schreyögg, G. (Hrsg.): Wissen in Unternehmen – Konzepte, Maßnahmen, Methoden. Berlin 2001, S. 183-221.

Anhang 1: Quellen der Marktrecherche

Um einen Überblick über den derzeitigen Stand der Forschung – im Bereich von Projektmanagement-Ontologien – zu erhalten, wurde zunächst eine umfangreiche Marktrecherche durchgeführt. Ziel dieser Marktrecherche ist es, durch den Einsatz verschiedenster Quellen eine möglichst umfangreiche Übersicht über die Begrifflichkeiten und deren Zusammenhänge im Bereich des Projektmanagements zu erhalten. Die hier zugrunde liegenden Medien belaufen sich auf wissenschaftliche Fachliteratur und Internetquellen. Die Recherche mit Hilfe wissenschaftlicher Fachliteratur erstreckt sich von Fachbüchern bis hin zu Fachzeitschriften und ermöglicht den Einsatz von Suchmaschinen diverser Online-Bibliotheken. Auch Zeitschriften-Datenbanken unterschiedlicher Anbieter werden für eine umfangreiche Recherche herangezogen. Die Suche im Internet erstreckt sich auf Suchmaschinen, wie zum Beispiel Google oder Swoogle.

Die folgende Abbildung 15 gibt einen Überblick über die zur Marktrecherche verwendeten Quellen.

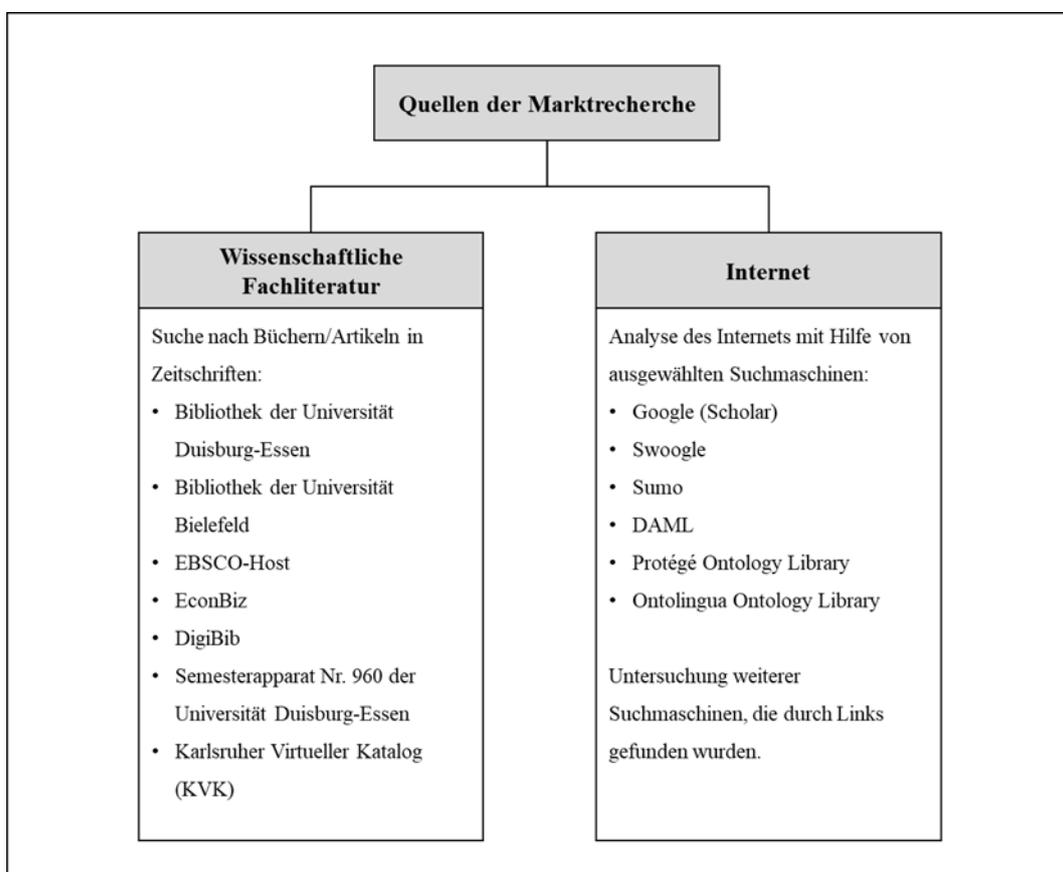


Abbildung 15: Quellen der Marktrecherche

Eine Auswahl der Ontologien, die anlässlich der Recherche wissenschaftlicher Fachliteratur gefunden wurden, werden in der Tabelle 4 auf den nächsten Seiten aufgelistet.

Autor(en) und ggf. Name der Ontologie	Schwerpunkt	Kurzbeschreibung
ABELS/AHLEMANN/HAHN et al. ²⁹⁴ PROMONT	Projektmanagement in virtuellen Organisationen	Die PROMONT-Ontologie formalisiert die typischen Elemente, die für die Projektstrukturierung verwendet werden, und bietet so die Grundlage für das gemeinsame Verständnis zwischen projektbezogenen Begriffen und Methoden. Dieses Vorgehen erleichtert das Managen von Projekten in dynamischen, virtuellen Organisationen.
AHLEMANN/TEUTEBERG/ BRUNE ²⁹⁵	Informationsmodelle	Mittels der Ontologie der ISPRI erfolgt eine ontologiebasierte Attribuierung von Informationsmodellen.
ARAMO-IMMONEN ²⁹⁶	PMBOK®	Diese Ontologie bietet eine Klassifizierung der Projektmanagementkonzepte, basierend auf dem PMBOK® Guide. Ziel ist die Steuerung der Projekte während ihres Lebenszyklus. Die betrachteten Disziplinen sind: Projektintegrationsmanagement, Projektumfangsmanagement und Projektzeitmanagement.
BENJAMIN/AKELLA/MALEK et al. ²⁹⁷	prozessorientierter Rahmen	Die Autoren beschreiben einen ontologiegestützten Rahmen für prozessorientierte Anwendungen zur Vereinfachung des Informationsaustauschs.
BERGENRODT/KOWALSKI ²⁹⁸ Casablanca	Logistik	Hierbei handelt es sich um eine Transportontologie.

294) Vgl. ABELS/AHLEMANN/HAHN et al. (2006).

295) AHLEMANN/TEUTEBERG/BRUNE (2006).

296) Vgl. ARAMO-IMMONEN (2009).

297) Vgl. BENJAMIN/AKELLA/MALEK et al. (2005).

298) Vgl. BERGENRODT/KOWALSKI (2015).

BIESALSKI ²⁹⁹	Kompetenzmanagement	Diese Ontologie bietet einen Überblick über die Fach-, Methoden-, Sozial- und IT-Kenntnisse des Managements, die für das strategische Planen hilfreich sein können.
BODEA ³⁰⁰ SinPers System	E-Learning	Das entwickelte SinPers-System ist ein webbasiertes, personalisiertes E-Learning-System, das den Fokus auf das Projektmanagement legt. Das verwendete Lernmaterial wird mittels einer Ontologie strukturiert.
BODEA/ELMAS/TANASESCU et al. ³⁰¹ PMCatalog	Kompetenzmanagement	Innerhalb des Projekts CONTO wurde ein Modell für die Kompetenzen des Projektmanagements erstellt. Hierbei wurden verschiedene Level der Kompetenzausprägungen für eine Reihe von Kompetenz-Kategorien (technische, kontextbezogene und Verhaltens-Kompetenzen) definiert.
DONG/HUSSAIN/CHANG ³⁰² ORPMS	allgemeiner Ansatz zum Projektmanagement	Die Ontologie dient der Erfassung und Speicherung von Wissen und der gemeinsamen Nutzung von Projektüberwachungssystemen sowie -prozessen.
DORN/NAZ/PICHLMAIR ³⁰³	Kompetenzmanagement	Diese Ontologie stuft relevante Kompetenzen in Level und Noten ein.
HORST/BACHMANN/HESSE ³⁰⁴ OBSE	Software-Entwicklungsprozesse	Die Grundidee dieser Ontologie basiert auf der Formalisierung von Fachwissen bereits beendeter Projekte. Es wird versucht, das Fachwissen aus bereits umgesetzten Projekten in neuen Projekten innerhalb der gleichen Domäne wiederzuverwenden.

299) Vgl. BIESALSKI/ABDECKER (2005).

300) Vgl. BODEA (2007).

301) Vgl. BODEA/ELMAS/TANASESCU et al. (2010).

302) Vgl. DONG/HUSSAIN/CHANG (2011).

303) Vgl. DORN/NAZ/PICHLMAIR (2007).

304) Vgl. HORST/BACHMANN/HESSE (2012).

HUGHES ³⁰⁵ Prinny	Prince2®	Der Prototyp wird genutzt, um Projektmanagementprozesse anzupassen und zu modellieren. Zudem wird eine Überprüfung der Übereinstimmung der von den Lieferanten angegebenen Methoden mit den Standards ermöglicht.
RUIZ-BERTOL/ RODRÍGUEZ/DOLADO ³⁰⁶	allgemeiner Ansatz zum Projektmanagement	Dieser Ansatz zeigt, wie Wissen abgeleitet werden kann, um die Entscheidungsfindung für Projekte des Managements zu unterstützen und zu verbessern.
SHEEBA/KRISHNAN/BERNARD ³⁰⁷	PMBOK®	Die entwickelte Ontologie wird für die automatische Klassifizierung von Lernmaterialien des Projektmanagements genutzt.
USCHOLD et al. ³⁰⁸ Enterprise Ontology	Unternehmensmodellierung und Software-Anwendungen	Diese Ontologie liefert einen Überblick über die relevanten Begrifflichkeiten für Unternehmen. Sie beinhaltet Methoden und ein Set an Computer-Tools zur Unternehmensmodellierung. Zudem sind für Projekte erforderliche Kernprozesse vorzufinden.
WONG/LAFAYE/BOURSIER ³⁰⁹ BProjM	allgemeiner Ansatz zum Projektmanagement	Die Ontologie bietet eine Grundlage für die Entwicklung eines integrierten Projektmanagement-Informationssystems (PMIS).
ZELEWSKI/ALAN/ALPARSLAN et al. ³¹⁰ KOWIEN	Kompetenzmanagementsysteme	Die Ontologie KOWIEN dient der Strukturierung und Unterstützung bei der Wissensakquisition und Wissenswiederverwendung innerhalb ontologiebasierter Kompetenzmanagementsysteme.

Tabelle 4: Recherche wissenschaftlicher Fachliteratur zu Ontologien

305) Vgl. HUGHES (2010).

306) Vgl. RUIZ-BERTOL/RODRÍGUEZ/DOLADO (2011).

307) Vgl. SHEEBA/KRISHNAN/BERNARD (2012).

308) Vgl. USCHOLD/KING/MORALEE et al. (1998).

309) Vgl. WONG/LAFAYE/BOURSIER (2010).

310) Vgl. ZELEWSKI/ALAN/ALPARSLAN et al. (2005).

Die gesamte Marktrecherche wurde im Zeitraum von Februar 2020 bis August 2020 durchgeführt. Für die Suche wurden u. a. die in der Abbildung 16 skizzierten Begriffe in den oben aufgelisteten Quellen sowohl in deutscher als auch in englischer Sprache recherchiert.



Abbildung 16: „Wortwolke“ der Internetrecherche

Anhang 2: Erstellung der Domänenontologie in Protégé

Anhang 2.1: Installationsanleitung von Protégé

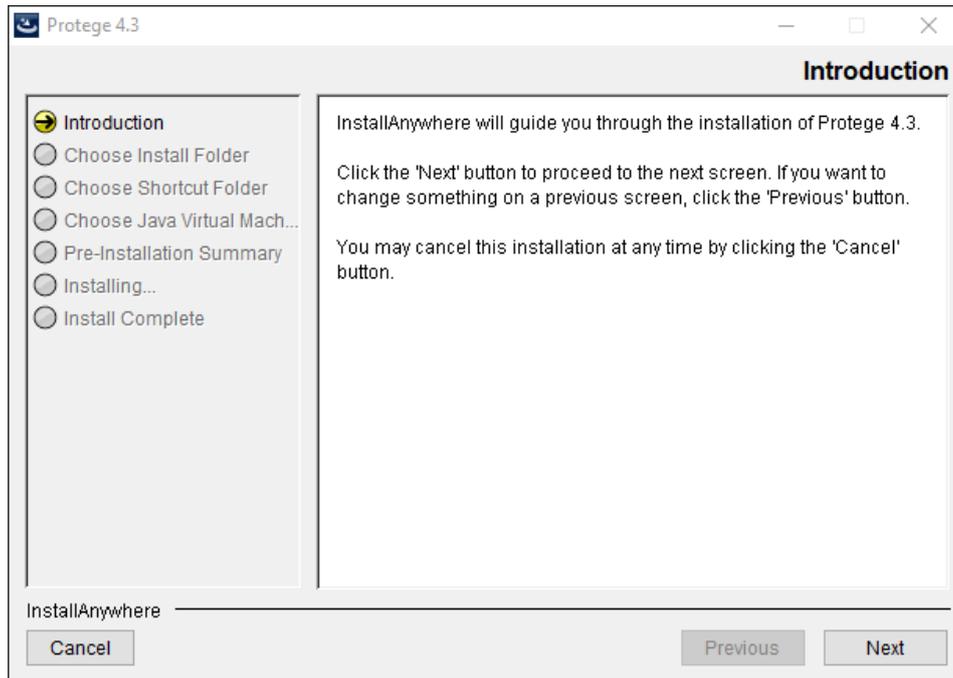


Abbildung 17: „Introduction“

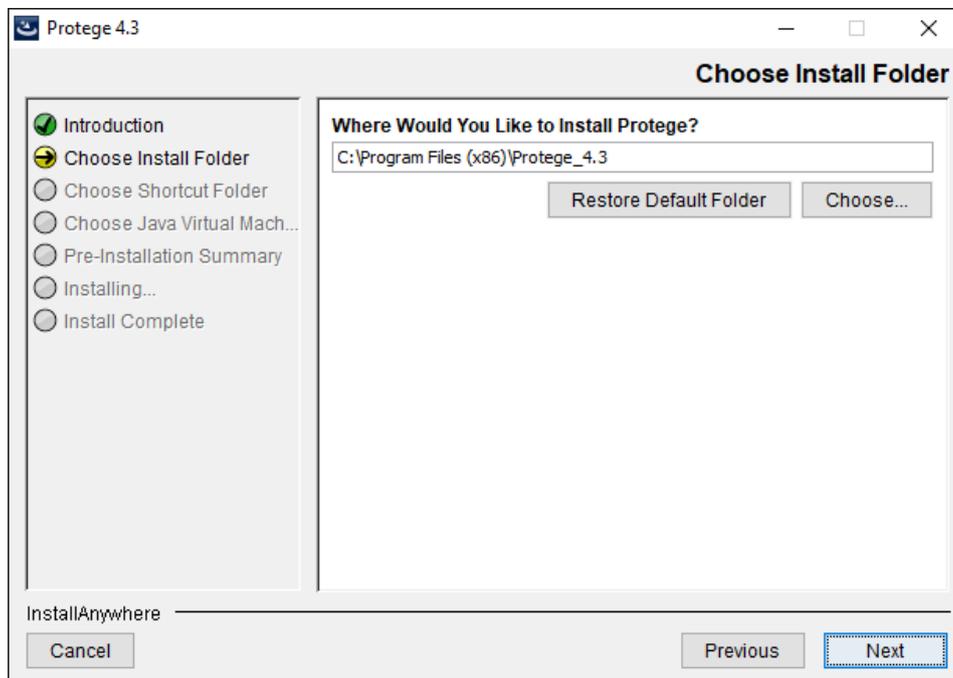


Abbildung 18: „Choose Install Folder“

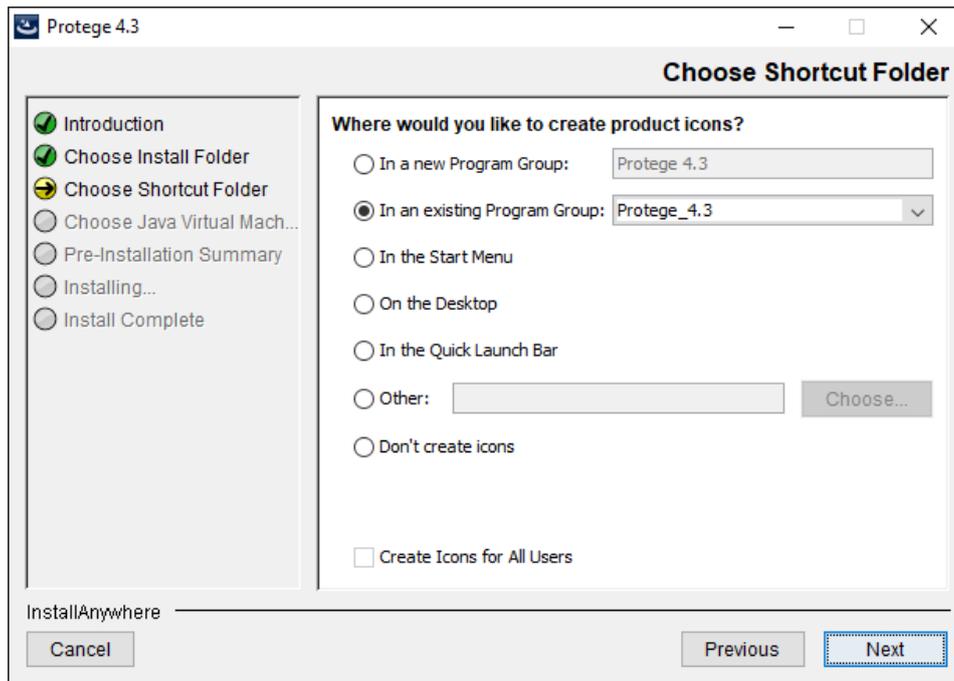


Abbildung 19: „Choose Shortcut Folder“

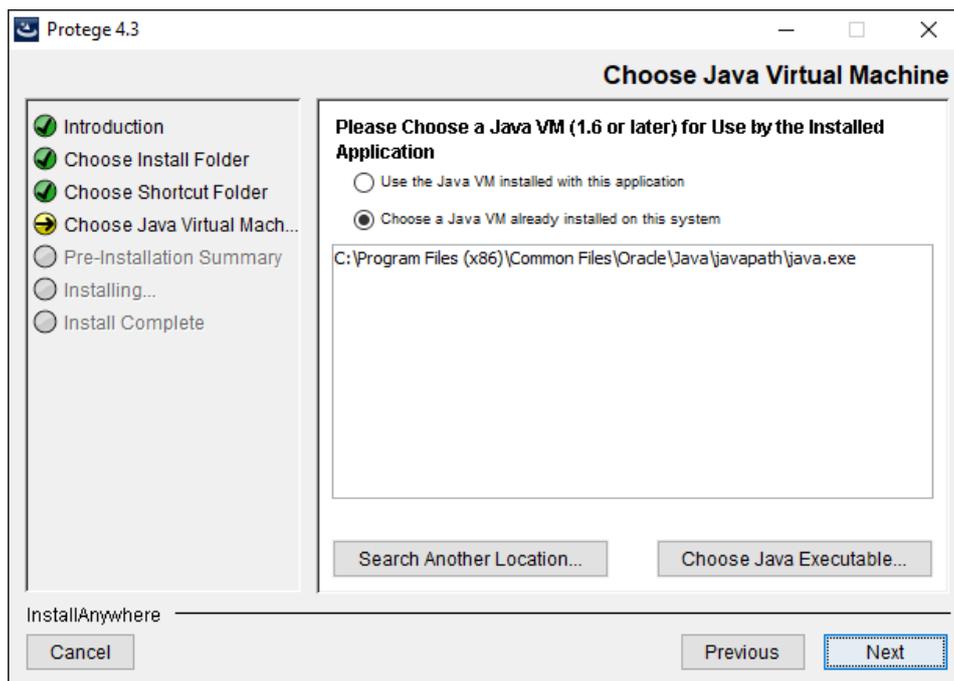


Abbildung 20: „Choose Java Virtual Machine“

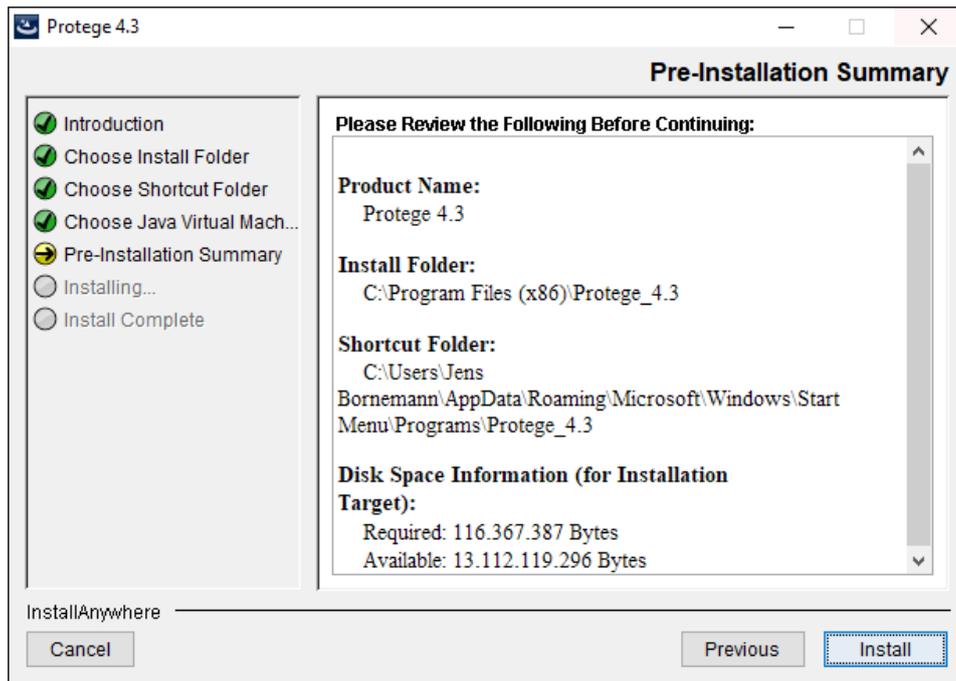


Abbildung 21: „Pre-Installation Summary“

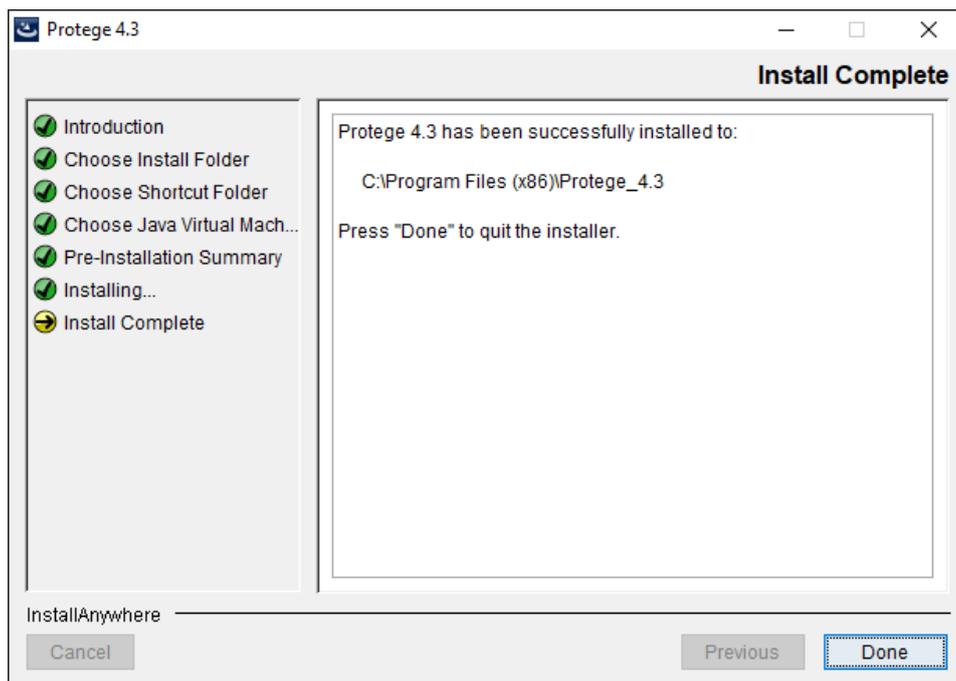


Abbildung 22: „Install Complete“

Anhang 2.2: Erstellen von Klassen

Eine Einführung in den Umgang mit Protégé befindet sich auf der Protégé-Website.³¹¹

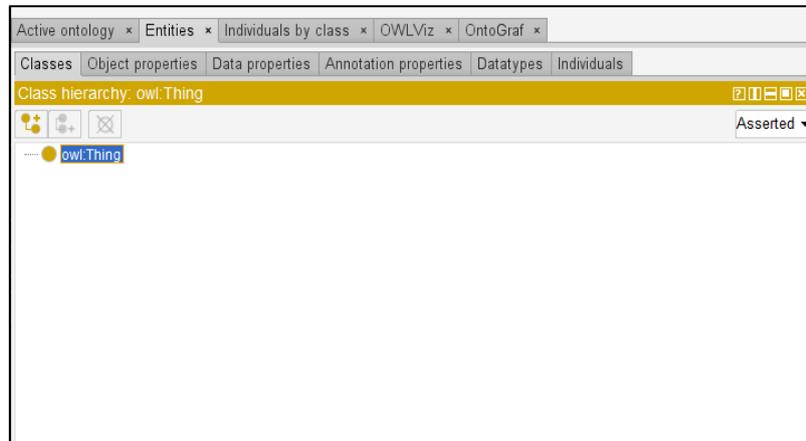


Abbildung 23: Hinzufügen einer neuen Klasse

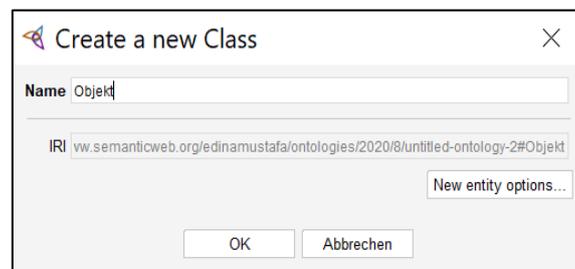


Abbildung 24: Festlegung des Namensraums einer neuen Klasse

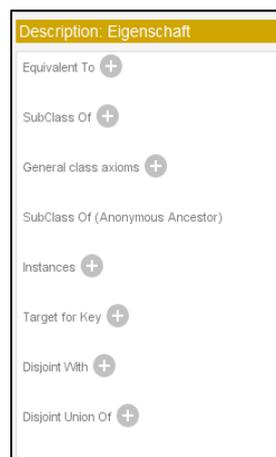


Abbildung 25: Festlegung der Klasseneigenschaften

311) Die Protégé-Website befindet sich unter der URL: <http://protegeproject.github.io/protege/getting-started/> (letzter Zugriff am: 10.03.2022).

Anhang 2.3: Erstellen von Relationen

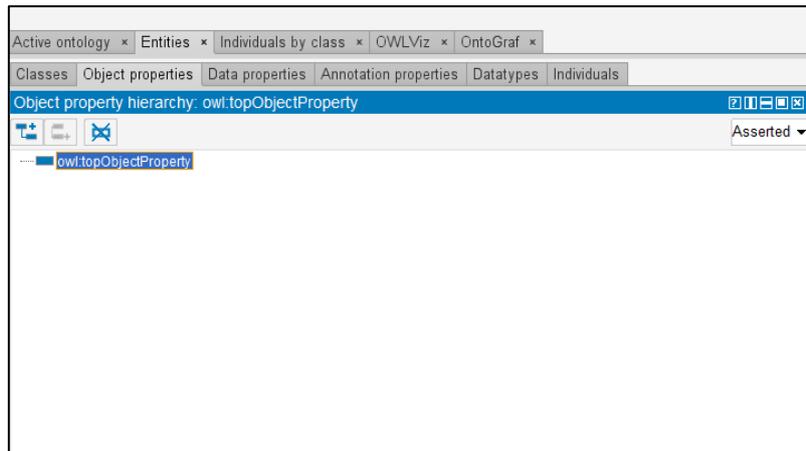


Abbildung 26: Hinzufügen einer neuen Relation

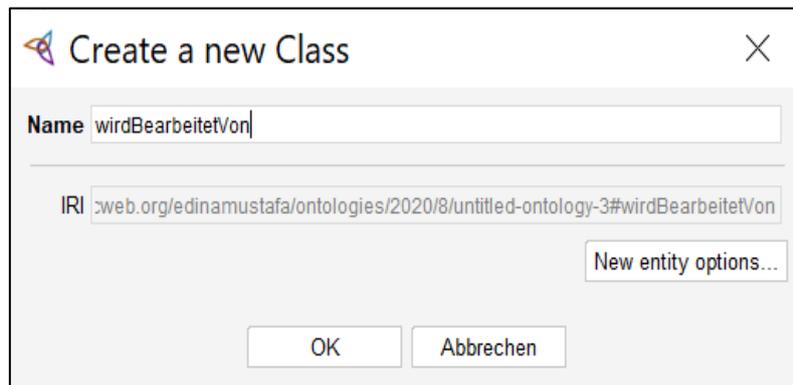


Abbildung 27: Festlegung des Namensraums einer neuen Relation

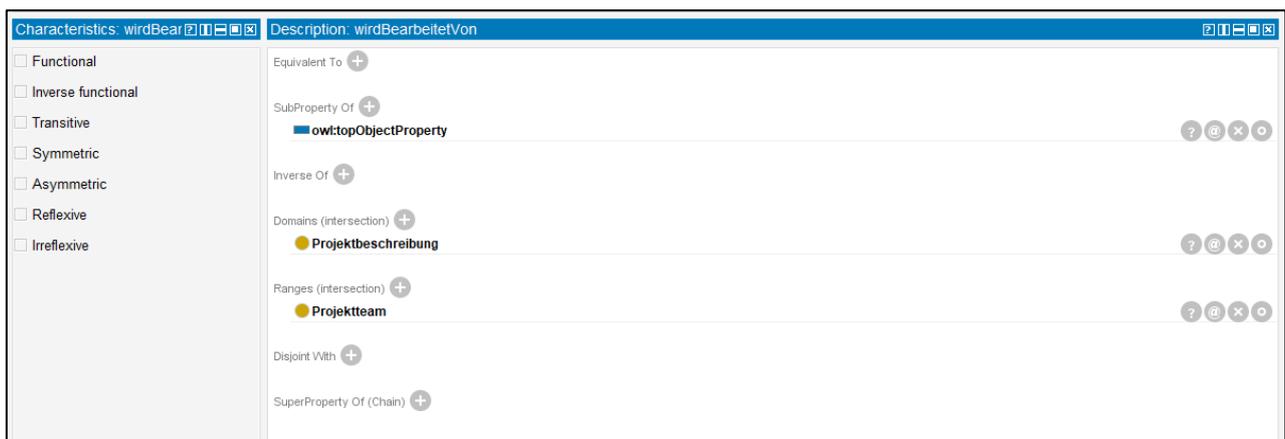


Abbildung 28: Festlegung der Relationseigenschaften

Anhang 2.4 – Prototyp der Domänenontologie

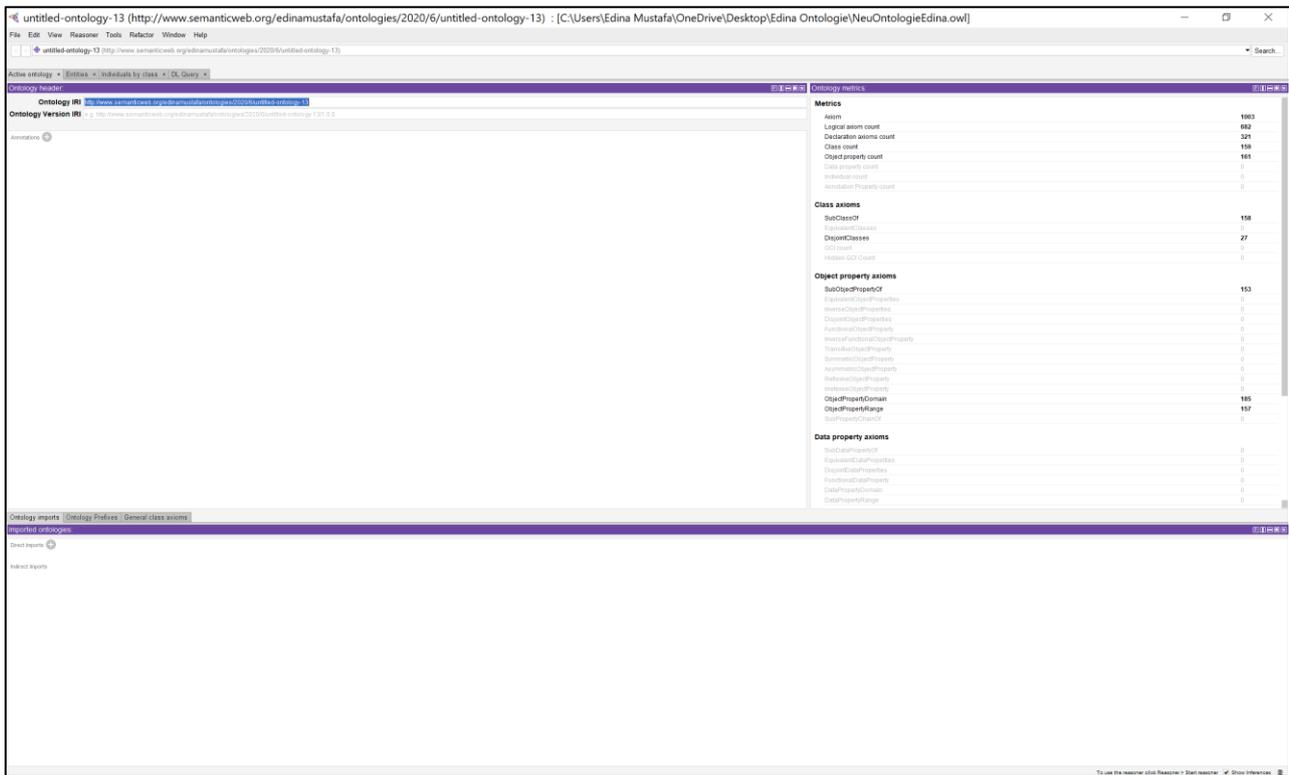


Abbildung 29: Protégé-Oberfläche

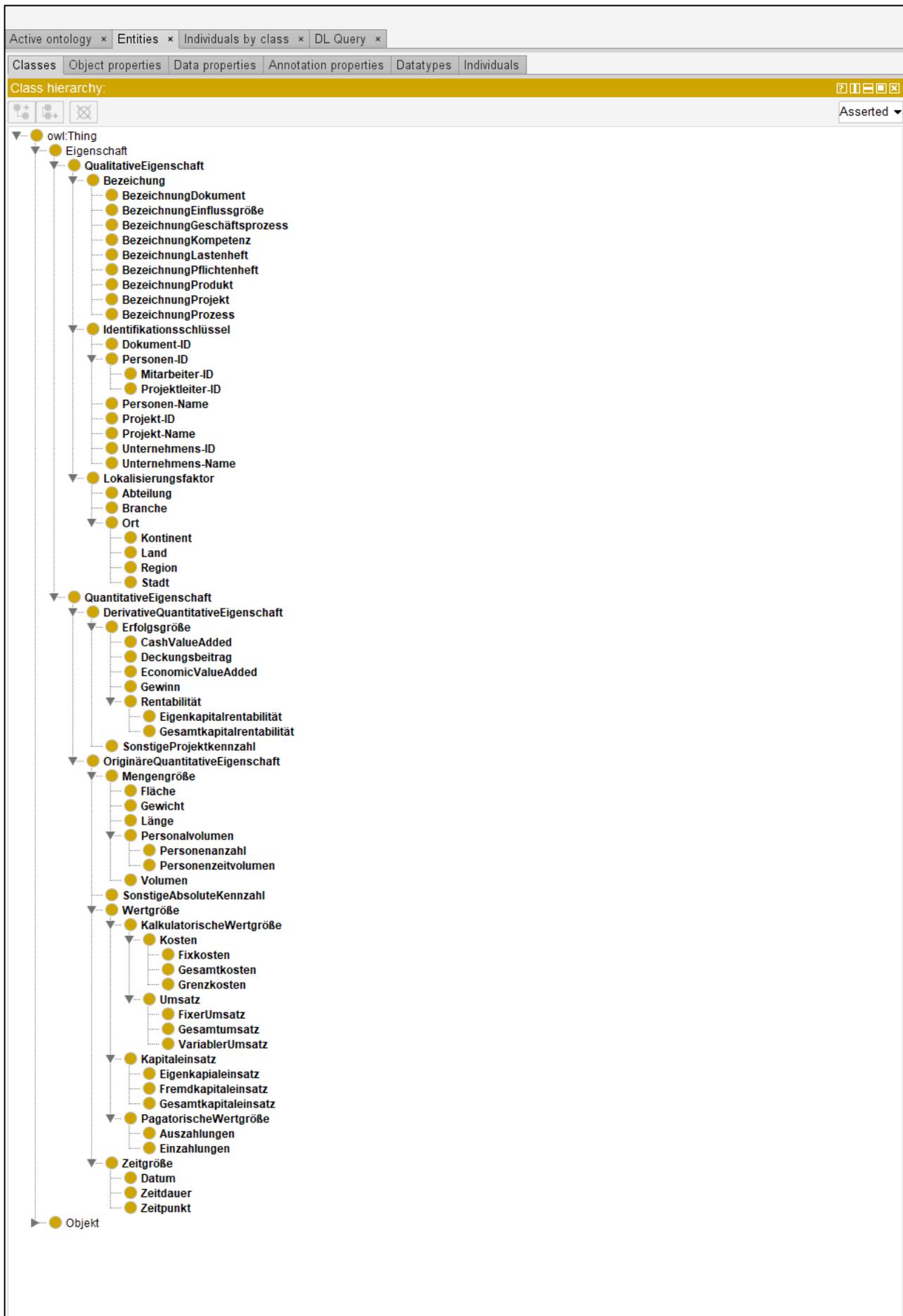


Abbildung 30: Protégé-Klassen Teil 1



Abbildung 31: Protégé-Klassen Teil 2

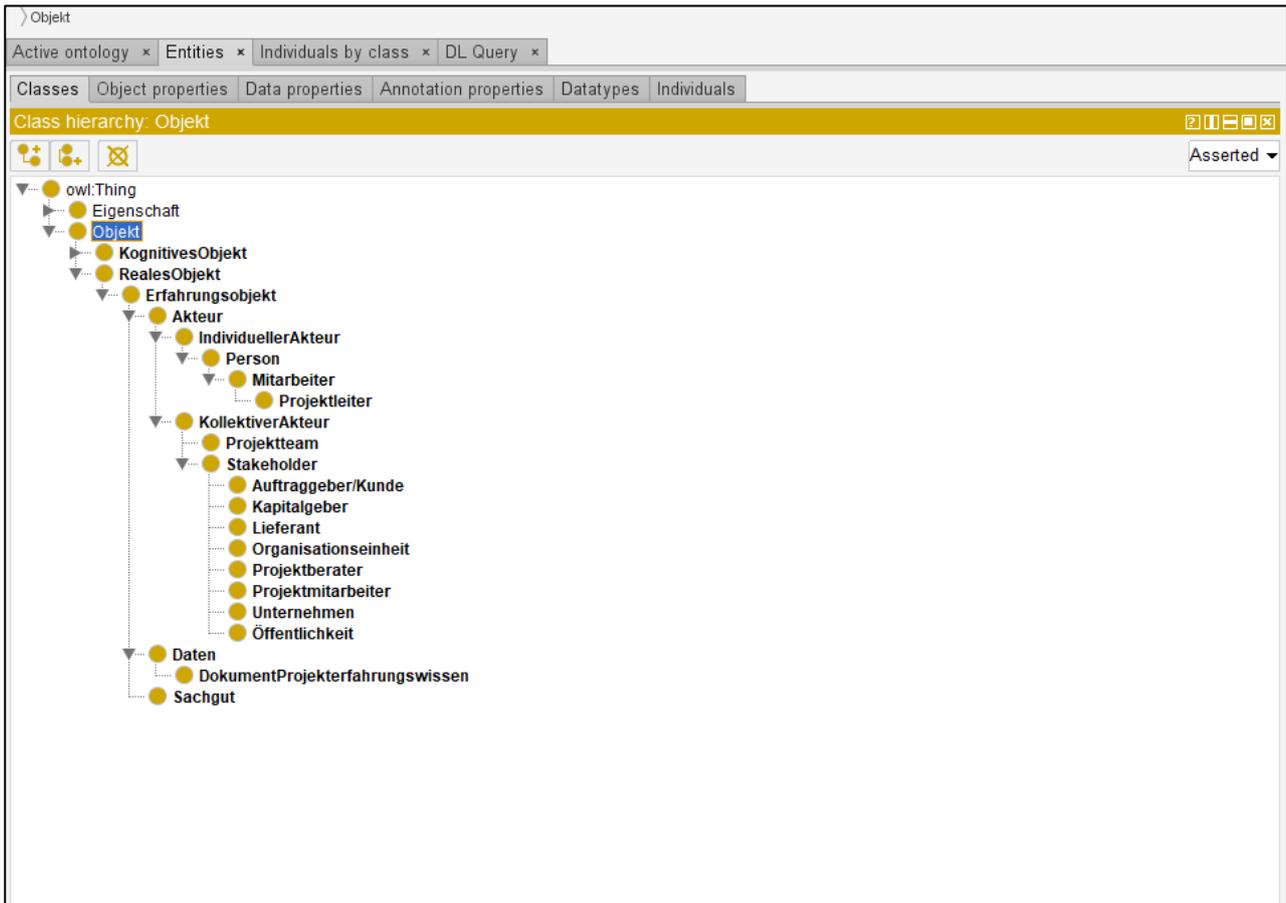


Abbildung 32: Protégé-Klassen Teil 3

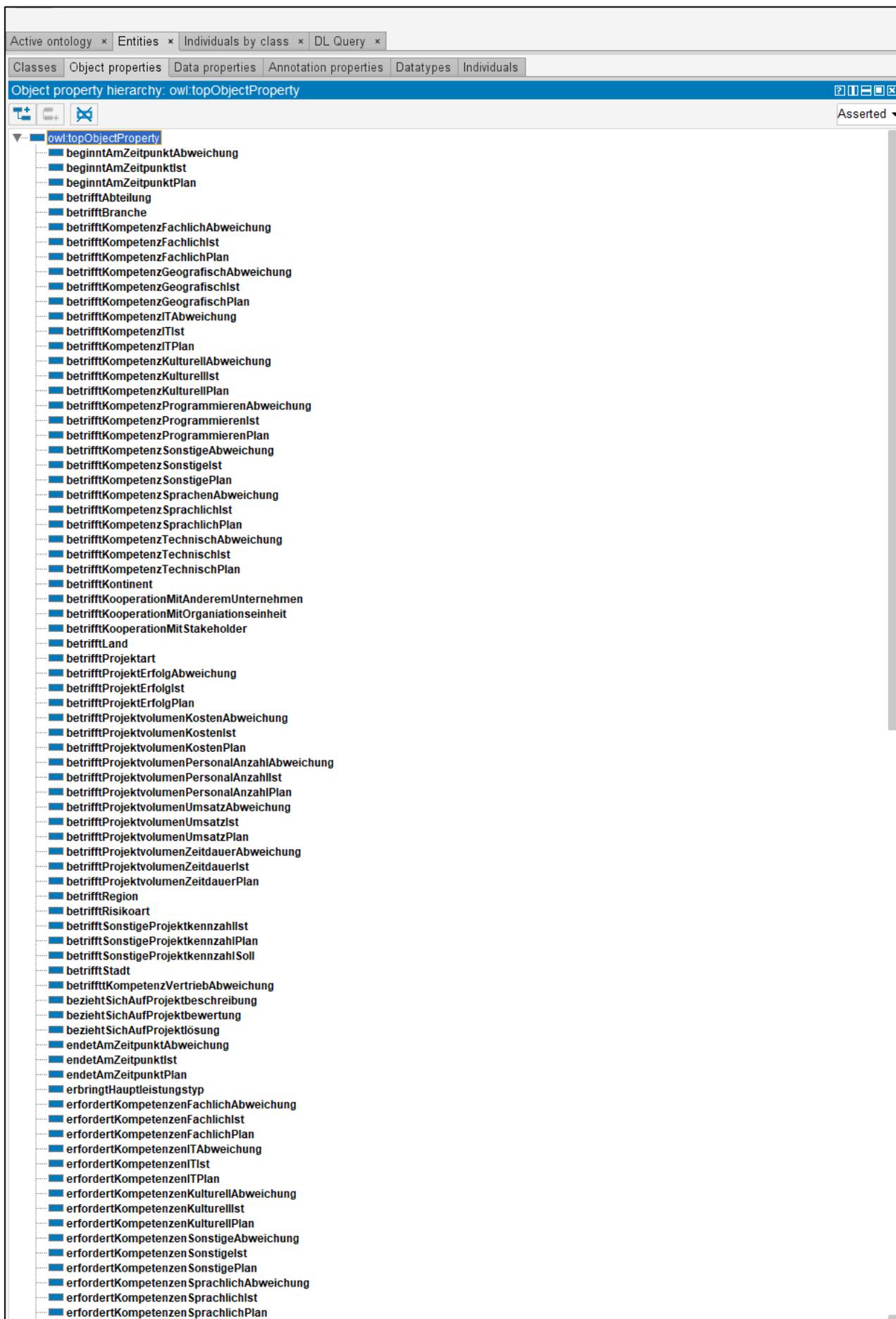


Abbildung 33: Ausschnitt aus den Protégé-Relationen

Anhang 3: Tabelle mit Sloteseigenschaften

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projekt	hatProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	hatProjektbewertung	Projektbewertung	Relation	Instance	1
	hatProjektlösung	Projektlösung	Relation	Instance	1
	hatAuftraggeber	Unternehmen	Relation	String	1
	hatUnmittelbareKooperationspartner	Unternehmen	Relation	String	0-1
	wirdImEigenenUnternehmenDurchgeführtVon	Person	Relation	String	1
	umfasstEinsatzmittelplanung	Einsatzmittelplanung	Relation	Instance	1
	wirdGesteuertMitMethode	Steuerungsmethode	Relation	Instance	1
Projektbeschreibung	betrifftProjektart	Projektart	Relation	Instance	1
	wirdBearbeitetVon	Projektteam	Relation	String	1
	wirdDurchgeführtVon	Unternehmen	Relation	String	1
	betrifftRegion	Region	Relation	Symbol	1
	hatProjekt-ID-ImEigenenUnternehmen	Projekt-ID	Relation	String	1
	hatProjekt-ID-ImAndrenUnternehmen	Projekt-ID	Relation	String	0-1
	gehörtZuProjekt	Projekt	Relation	Instance	1
	hatLastenheft	Lastenheft	Relation	Instance	1
	hatPflichtenheft	Pflichtenheft	Relation	Instance	1
	hatProjekterfolgPlan	Erfolgsgröße	Relation	Instance	0-1

Tabelle 5: Klassen der Domänenontologie mit Slots und Sloteseigenschaften – Teil 1

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projektbeschreibung	hatProjektvolumenKostenPlan	Kosten	Relation	Float	1
	hatProjektvolumenPersonalAnzahlPlan	Personenanzahl	Relation	Integer	1
	hatProjektvolumenUmsatzPlan	Umsatz	Relation	Float	1
	hatProjektvolumenZeitdauerPlan	Zeitdauer	Relation	Float	1
	hatSonstigeProjektkennzahlPlan	SonstigeProjektkennzahl	Relation	Instance	0-1
	wirdGesteuertMitMethode	Steuerungsmethode	Relation	Instance	1
	umfasstEinsatzmittelplanungPlan	Einsatzmittelplanung	Relation	Instance	1
	betrifftRisikoart	Risikoart	Relation	Instance	1
	erfordertKompetenzenFachlichPlan	KompetenzFachlich	Relation	Instance	0-1
	erfordertKompetenzenITPlan	KompetenzIT	Relation	Instance	0-1
	erfordertKompetenzenKulturellPlan	KompetenzKulturverständnis	Relation	Instance	0-1
	erfordertKompetenzenSonstigePlan	KompetenzSonstige	Relation	Instance	0-1
	erfordertKompetenzenSprachlichPlan	KompetenzSprachen	Relation	Instance	0-1
	erfordertKompetenzenTechnischPlan	KompetenzTechnisch	Relation	Instance	0-1
	erfordertKompetenzGeografischPlan	KompetenzGeografisch	Relation	Instance	0-1
	erfordertKompetenzProgrammierenPlan	KompetenzProgrammieren	Relation	Instance	0-1
	betrifftKontinent	Kontinent	Relation	Symbol	1
	betrifftLand	Land	Relation	Symbol	1
	betrifftStadt	Stadt	Relation	Symbol	1
betrifftBranche	Branche	Relation	Symbol	1	

Tabelle 6: Klassen der Domänenontologie mit Slots und Sloteigenschaften – Teil 2

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projektbeschreibung	betrifftAbteilung	Abteilung	Relation	String	1
	hatProjekt-Name	Projekt-Name	Relation	String	1
	hatDokument-ID	Dokument-ID	Relation	String	1
	hatBezeichnungFürDokument	BezeichnungDokument	Relation	String	1
	hatBezeichnungFürEinflussgröße	BezeichnungEinflussgröße	Relation	String	1
	hatBezeichnungFürGeschäftsprozess	BezeichnungGeschäftsprozess	Relation	String	1
	hatBezeichnungFürKompetenz	Kompetenztyp	Relation	String	1
	hatBezeichnungFürLastenheft	BezeichnungLastenheft	Relation	String	1
	hatBezeichnungFürPflichtenheft	BezeichnungPflichtenheft	Relation	String	1
	hatBezeichnungFürProjekt	BezeichnungProjekt	Relation	String	1
	hatBezeichnungFürProdukt	BezeichnungProdukt	Relation	String	1
	hatBezeichnungFürProzess	BezeichnungProzess	Relation	String	1
	beginntAmZeitpunktPlan	Datum	Relation	Float	1
	endetAmZeitpunktPlan	Datum	Relation	Float	1
Projektbewertung	erfordertKompetenzenFachlichAbweichung	KompetenzFachlich	Relation	Instance	0-1
	erfordertKompetenzenITAbweichung	KompetenzIT	Relation	Instance	0-1
	erfordertKompetenzenKulturellAbweichung	KompetenzKulturverständnis	Relation	Instance	0-1
	erfordertKompetenzenSonstigeAbweichung	KompetenzSonstige	Relation	Instance	0-1
	erfordertKompetenzenSprachlichAbweichung	KompetenzSprachen	Relation	Instance	0-1
	erfordertKompetenzenTechnischAbweichung	KompetenzTechnisch	Relation	Instance	0-1

Tabelle 7: Klassen der Domänenontologie mit Slots und Sloteigenschaften – Teil 3

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projektbewertung	erfordertKompetenzGeografischAbweichung	KompetenzGeografisch	Relation	Instance	0-1
	erfordertKompetenzVertriebAbweichung	KompetenzVertrieb	Relation	Instance	0-1
	erfordertKompetenzProgrammierenAbweichung	KompetenzProgrammieren	Relation	Instance	0-1
	hatProjekterfolgAbweichung	Erfolgsgröße	Relation	Instance	0-1
	hatProjektvolumenKostenAbweichung	Kosten	Relation	Float	0-1
	hatProjektvolumenPersonalAnzahlAbweichung	Personenanzahl	Relation	Integer	0-1
	hatProjektvolumenUmsatzAbweichung	Umsatz	Relation	Float	0-1
	hatProjektvolumenZeitdauerAbweichung	Zeitdauer	Relation	Float	0-1
	hatSonstigeProjektkennzahlAbweichung	SonstigeProjektkennzahl	Relation	Instance	0-1
	hatProjekterfahrungswissen	DokumentProjekterfahrungswissen	Relation	Instance	1
	erfordertKooperationMitWeiterenStakeholdern	Stakeholder	Relation	String	0-1
	beginntAmZeitpunktAbweichung	Datum	Relation	Float	1
	endetAmZeitpunktAbweichung	Datum	Relation	Float	1
Projektlösung	erfordertKompetenzenFachlichIst	KompetenzFachlich	Relation	Instance	0-1
	erfordertKompetenzenITIst	KompetenzIT	Relation	Instance	0-1
	erfordertKompetenzenKulturellIst	KompetenzKulturverständnis	Relation	Instance	0-1
	erfordertKompetenzenSonstigeIst	KompetenzSonstige	Relation	Instance	0-1
	erfordertKompetenzenSprachlichIst	KompetenzSprachen	Relation	Instance	0-1
	erfordertKompetenzenTechnischIst	KompetenzTechnisch	Relation	Instance	0-1
	erfordertKompetenzGeografischIst	KompetenzGeografisch	Relation	Instance	0-1

Tabelle 8: Klassen der Domänenontologie mit Slots und Slotigenschaften – Teil 4

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projektlösung	erfordertKompetenzProgrammierenIst	KompetenzProgrammieren	Relation	Instance	0-1
	hatProjekterfolgIst	Erfolgsgröße	Relation	Instance	0-1
	hatProjektvolumenKostenIst	Kosten	Relation	Float	1
	hatProjektvolumenPersonalAnzahlIst	Personenanzahl	Relation	Integer	1
	hatProjektvolumenUmsatzIst	Umsatz	Relation	Float	1
	hatProjektvolumenZeitdauerIst	Zeitdauer	Relation	Float	1
	hatSonstigeProjektkennzahlIst	SonstigeProjektkennzahl	Relation	Instance	0-1
	beginntAmZeitpunktIst	Datum	Relation	Float	1
	endetAmZeitpunktIst	Datum	Relation	Float	1
Projekterfahrungswissen	betrifftProjektErfolgAbweichung	Erfolgsgröße	Relation	Instance	0-1
	betrifftProjektErfolgPlan	Erfolgsgröße	Relation	Instance	0-1
	betrifftProjektErfolgIst	Erfolgsgröße	Relation	Instance	0-1
	betrifftProjektart	Projektart	Relation	Instance	1
	betrifftProjektvolumenKostenAbweichung	Kosten	Relation	Float	0-1
	betrifftProjektvolumenKostenIst	Kosten	Relation	Float	1
	betrifftProjektvolumenKostenPlan	Kosten	Relation	Float	1
	betrifftProjektvolumenPersonalAnzahlAbweichung	Personenanzahl	Relation	Integer	0-1
	betrifftProjektvolumenPersonalAnzahlIst	Personenanzahl	Relation	Integer	1
	betrifftProjektvolumenPersonalAnzahlPlan	Personenanzahl	Relation	Integer	1
	betrifftProjektvolumenUmsatzAbweichung	Umsatz	Relation	Float	0-1

Tabelle 9: Klassen der Domänenontologie mit Slots und Sloteigenschaften – Teil 5

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projekterfahrungswissen	betrifftProjektvolumenUmsatzIst	Umsatz	Relation	Float	1
	betrifftProjektvolumenUmsatzPlan	Umsatz	Relation	Float	1
	betrifftProjektvolumenZeitdauerAbweichung	Zeitdauer	Relation	Float	0-1
	betrifftProjektvolumenZeitdauerPlan	Zeitdauer	Relation	Float	1
	betrifftProjektvolumenZeitdauerIst	Zeitdauer	Relation	Float	1
	betrifftKompetenzFachlichAbweichung	KompetenzFachlich	Relation	Instance	0-1
	betrifftKompetenzFachlichIst	KompetenzFachlich	Relation	Instance	0-1
	betrifftKompetenzFachlichPlan	KompetenzFachlich	Relation	Instance	0-1
	betrifftKompetenzITAbweichung	KompetenzIT	Relation	Instance	0-1
	betrifftKompetenzITIst	KompetenzIT	Relation	Instance	0-1
	betrifftKompetenzITPlan	KompetenzIT	Relation	Instance	0-1
	betrifftKompetenzKulturellAbweichung	KompetenzKulturverständnis	Relation	Instance	0-1
	betrifftKompetenzKulturellIst	KompetenzKulturverständnis	Relation	Instance	0-1
	betrifftKompetenzKulturellPlan	KompetenzKulturverständnis	Relation	Instance	0-1
	betrifftKompetenzSonstigeAbweichung	KompetenzSonstige	Relation	Instance	0-1
	betrifftKompetenzSonstigeIst	KompetenzSonstige	Relation	Instance	0-1
	betrifftKompetenzSonstigePlan	KompetenzSonstige	Relation	Instance	0-1
	betrifftKompetenzSprachlichAbweichung	KompetenzSprachen	Relation	Instance	0-1
	betrifftKompetenzSprachlichIst	KompetenzSprachen	Relation	Instance	0-1
	betrifftKompetenzSprachlichPlan	KompetenzSprachen	Relation	Instance	0-1

Tabelle 10: Klassen der Domänenontologie mit Slots und Slotigenschaften – Teil 6

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Projekterfahrungswissen	betrifftKompetenzTechnischAbweichung	KompetenzTechnisch	Relation	Instance	0-1
	betrifftKompetenzTechnischIst	KompetenzTechnisch	Relation	Instance	0-1
	betrifftKompetenzTechnischPlan	KompetenzTechnisch	Relation	Instance	0-1
	betrifftKompetenzGeografischAbweichung	KompetenzGeografisch	Relation	Instance	0-1
	betrifftKompetenzGeografischIst	KompetenzGeografisch	Relation	Instance	0-1
	betrifftKompetenzGeografischPlan	KompetenzGeografisch	Relation	Instance	0-1
	betrifftKompetenzProgrammierenAbweichung	KompetenzProgrammieren	Relation	Instance	0-1
	betrifftKompetenzProgrammierenIst	KompetenzProgrammieren	Relation	Instance	0-1
	betrifftKompetenzProgrammierenPlan	KompetenzProgrammieren	Relation	Instance	0-1
	betrifftKooperationMitAnderemUnternehmen	Unternehmen	Relation	String	0-1
	betrifftKooperationMitOrganiationseinheit	Organisationseinheit	Relation	String	0-1
	betrifftKooperationMitStakeholder	Stakeholder	Relation	String	0-1
	betrifftSonstigeProjektkennzahlIst	SonstigeProjektkennzahl	Relation	Instance	0-1
	betrifftSonstigeProjektkennzahlPlan	SonstigeProjektkennzahl	Relation	Instance	0-1
	betrifftSonstigeProjektkennzahlSoll	SonstigeProjektkennzahl	Relation	Instance	0-1
	wurdeVerfasstVon	Person	Relation	String	1
	beginntAmZeitpunktIst	Datum	Relation	Float	1
	endetAmZeitpunktIst	Datum	Relation	Float	1
	beginntAmZeitpunktPlan	Datum	Relation	Float	1
	endetAmZeitpunktPlan	Datum	Relation	Float	1
beginntAmZeitpunktAbweichung	Datum	Relation	Float	0-1	
endetAmZeitpunktAbweichung	Datum	Relation	Float	0-1	

Tabelle 11: Klassen der Domänenontologie mit Slots und Sloteseigenschaften – Teil 7

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Lastenheft	beziehtSichAufProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	beziehtSichAufProjektlösung	Projektlösung	Relation	Instance	1
	beziehtSichAufProjektbewertung	Projektbewertung	Relation	Instance	1
	wurdeErstelltVon	Person	Relation	String	1
Pflichtenheft	beziehtSichAufProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	beziehtSichAufProjektlösung	Projektlösung	Relation	Instance	1
	beziehtSichAufProjektbewertung	Projektbewertung	Relation	Instance	1
	wurdeErstelltVon	Person	Relation	String	1
Kosten	wurdenErmitteltVon	Person	Relation	String	1
	beziehtSichAufProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	beziehtSichAufProjektlösung	Projektlösung	Relation	Instance	1
	beziehtSichAufProjektbewertung	Projektbewertung	Relation	Instance	1
Umsatz	wurdenErmitteltVon	Person	Relation	String	1
	beziehtSichAufProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	beziehtSichAufProjektlösung	Projektlösung	Relation	Instance	1
	beziehtSichAufProjektbewertung	Projektbewertung	Relation	Instance	1
Erfolgsgröße	wurdenErmitteltVon	Person	Relation	String	1
	beziehtSichAufProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	beziehtSichAufProjektlösung	Projektlösung	Relation	Instance	1
	beziehtSichAufProjektbewertung	Projektbewertung	Relation	Instance	1
Unternehmen	hatUnternehmens-ID	Unternehmens-ID	Relation	String	1
	hatUnternehmens-Name	Unternehmens-Name	Relation	String	1
	wirdLokalisiertInKontinent	Kontinent	Relation	Symbol	1
	wirdLokalisiertInLand	Land	Relation	Symbol	1
	wirdLokalisiertInRegion	Region	Relation	Symbol	1
	wirdLokalisiertInStadt	Stadt	Relation	Symbol	1

Tabelle 12: Klassen der Domänenontologie mit Slots und Slotigenschaften – Teil 8

Domäne (Vorbereich)	Slot (Relationen)	Range (Nachbereich)	Slottyp	Wertetyp	Kardinalität
Person	hatPersonen-Name	Personen-Name	Relation	String	1
	hatPersonen-ID	Personen-ID	Relation	String	1
	istVerantwortlichFürOrganisationseinheit	Organisationseinheit	Relation	String	0-1
	hatKompetenzenFachlich	KompetenzFachlich	Relation	Instance	0-1
	hatKompetenzenIT	KompetenzIT	Relation	Instance	0-1
	hatKompetenzenKulturell	KompetenzKulturverständnis	Relation	Instance	0-1
	hatKompetenzenSonstige	KompetenzSonstige	Relation	Instance	0-1
	hatKompetenzenSprachlich	KompetenzSprachen	Relation	Instance	0-1
	hatKompetenzenTechnisch	KompetenzTechnisch	Relation	Instance	0-1
	hatKompetenzGeografisch	KompetenzGeografisch	Relation	Instance	0-1
	hatProjektkompetenz	Kompetenztyp	Relation	Instance	1
	hatKompetenzProgrammieren	KompetenzProgrammieren	Relation	Instance	0-1
	istMitarbeiterInAbteilung	Abteilung	Relation	String	1
Netzplan	beziehtSichAufProjektbeschreibung	Projektbeschreibung	Relation	Instance	1
	beziehtSichAufProjektlösung	Projektlösung	Relation	Instance	1
	beziehtSichAufProjektbewertung	Projektbewertung	Relation	Instance	1
	wurdeErstelltVon	Person	Relation	String	1
Projektteam	hatMitglied	Person	Relation	String	1
Mitarbeiter	istProjektleiter	Projektleiter	Relation	String	0-1

Tabelle 13: Klassen der Domänenontologie mit Slots und Sloteigenschaften – Teil 9

Anhang 4: Grundsätzliches Bedienungskonzept von jCORA

Beim erstmaligen Öffnen des Falleditors von jCORA wird im Hauptfenster die „Fallbasis“-Ansicht angezeigt.

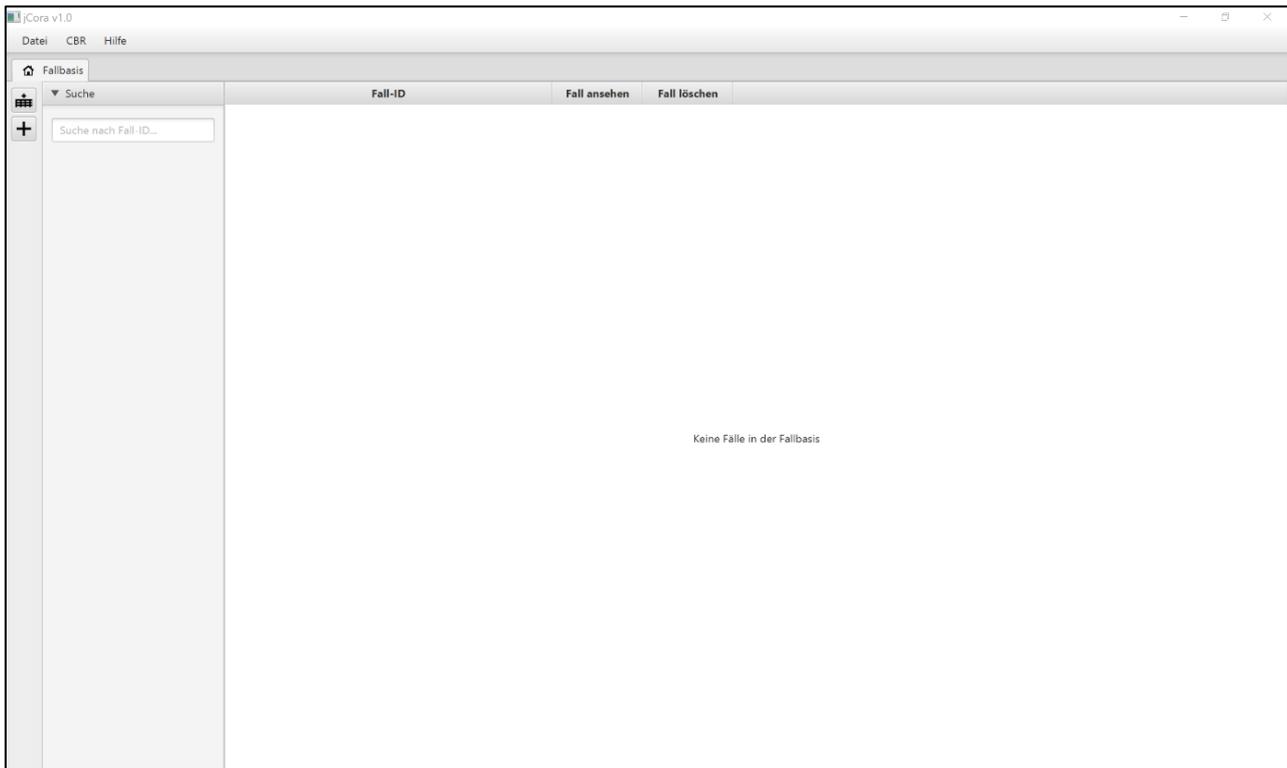


Abbildung 34: jCORA – Ansicht Fallbasis

Die Oberfläche gliedert sich in die drei Spalten „Fall-ID“, „Fall ansehen“ und „Fall löschen“. Am linken Rand ist die Werkzeugliste der Anwendung zu finden. Durch Auswählen des oberen Symbols können externe Fälle in die Fallbasis importiert werden. Mittels des unteren Symbols können neue Fälle angelegt werden. Rechts neben dem Fallimport befindet sich eine Suchfunktion, mit der Fall-IDs innerhalb der Fallbasis gesucht werden können.

Um neue Fälle zu erstellen, bedarf es zunächst als Grundlage einer Domänenontologie. Um dem CBR-Tool Inhalte und Strukturen – in Form einer zuvor erstellten Domänenontologie – zur Verfügung zu stellen, müssen die Einstellungen angepasst werden. Der erste Reiter in den Einstellungen ermöglicht eine Sprachauswahl.

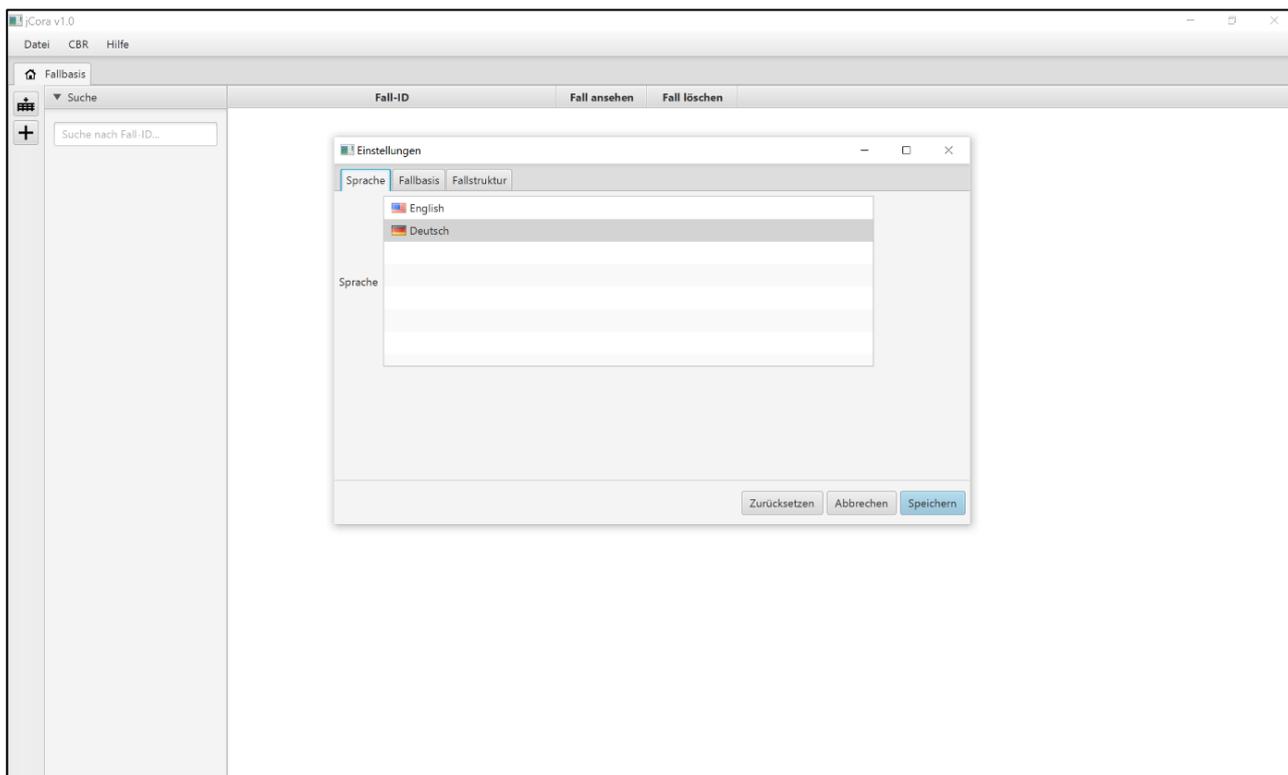


Abbildung 35: jCORA – Spracheinstellungen

Für den hier vorgestellten Projektbericht wurde die Sprache „Deutsch“ gewählt.

Im zweiten Reiter „Fallbasis“ erfolgen eine Auswahl des Ablageorts der Fallbasis, die Auswahl der zugrunde liegenden Domänenontologie und die Eingabe des Namensraums der ausgewählten Ontologie.

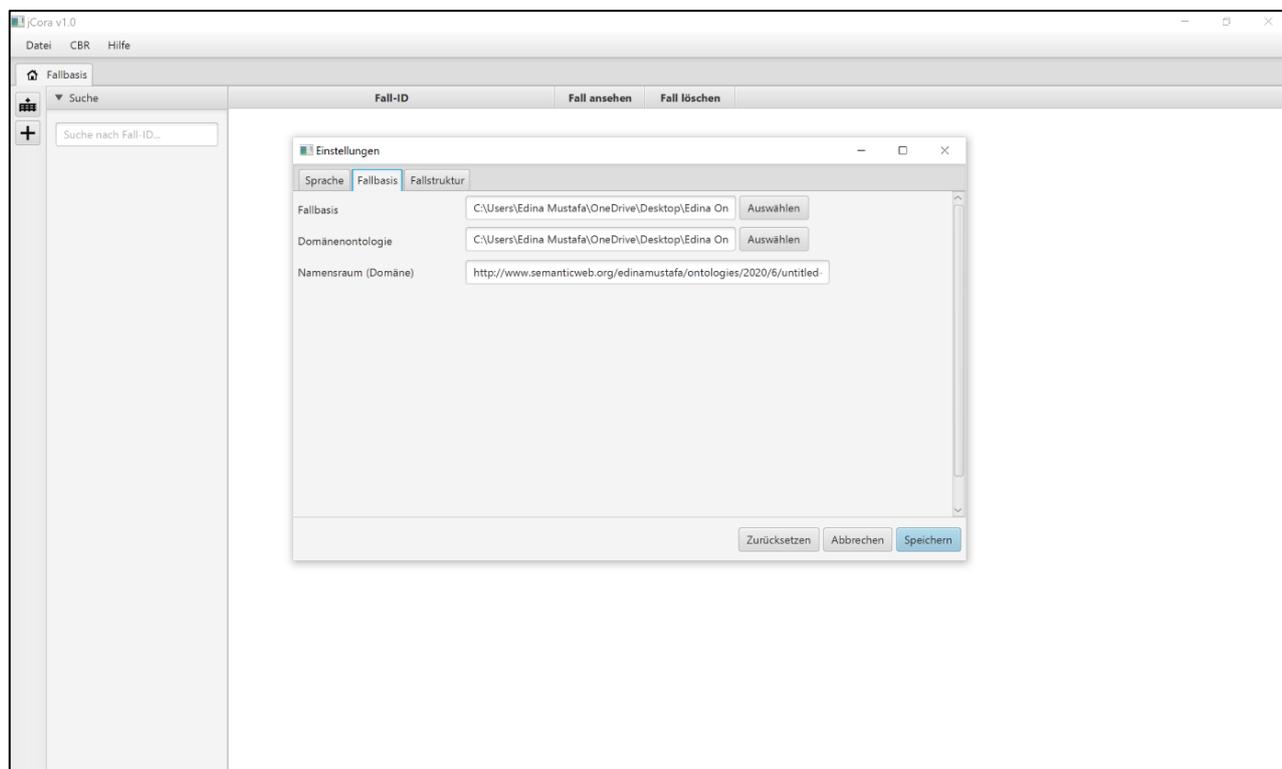


Abbildung 36: jCORA – Fallbasis-Einstellungen

Beim Öffnen der ersten Fallbeschreibung werden die automatisch erzeugten Instanzen „Fallbeschreibung“, „Fallbewertung“ und „Falllösung“ angezeigt.

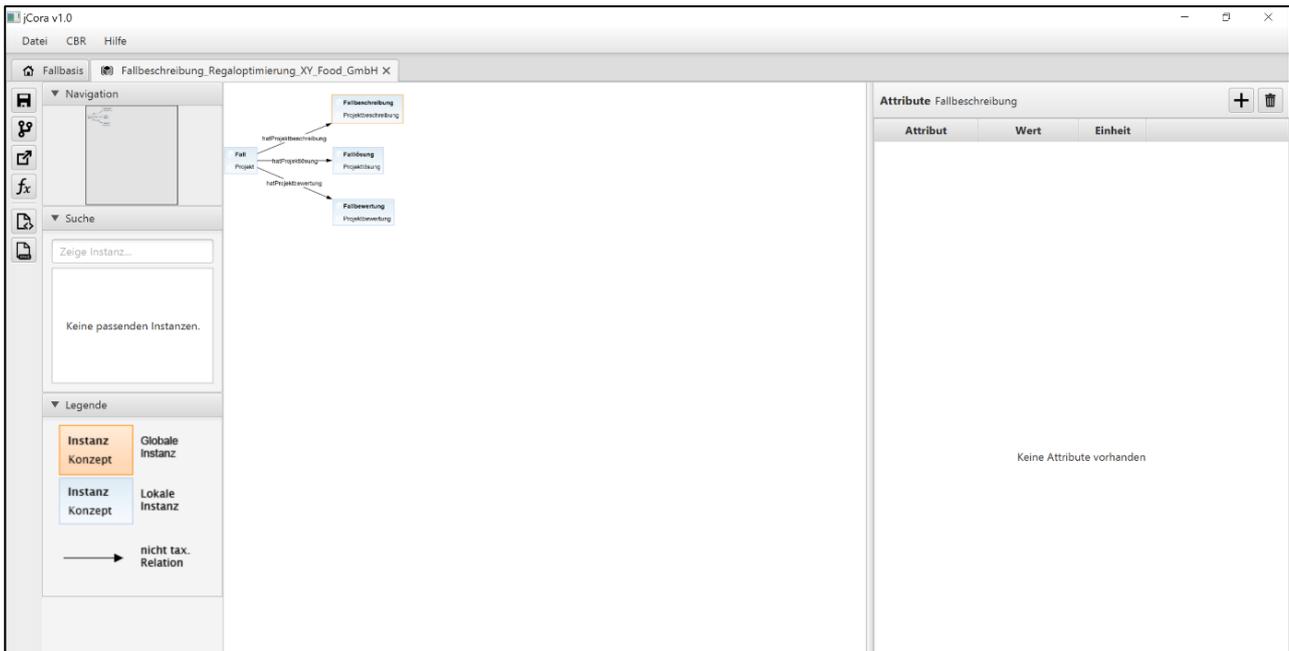


Abbildung 39: jCORA – Falleditor

Mit einem Rechts-Klick auf „Fallbeschreibung“ können Relationen hinzugefügt werden.

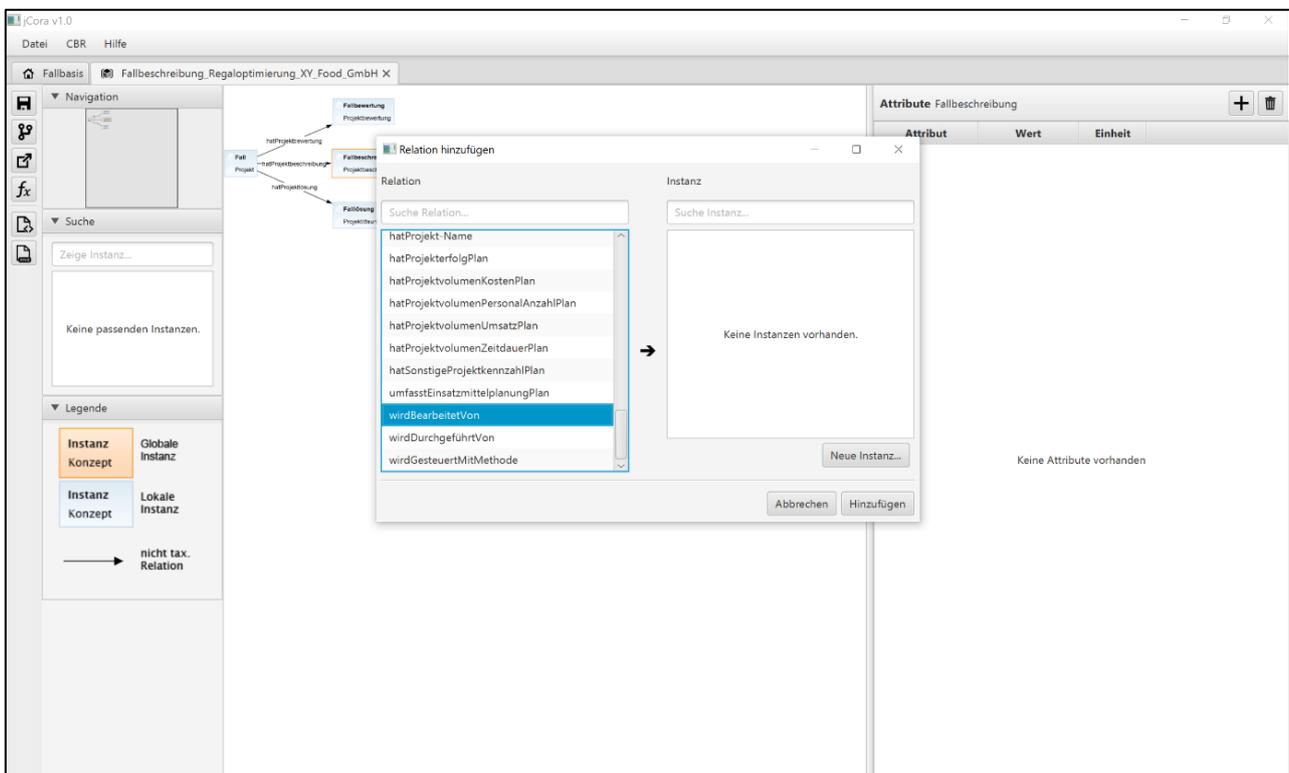


Abbildung 40: jCORA – Relationseditor

In dieser Ansicht sind alle für die Klasse „Projektbeschreibung“ verfügbaren Relationen zu sehen. Mit einem Klick auf „wirdBearbeitetVon“ können auf der rechten Seite unter „Neue Instanz“ fallspezifische Instanzen hinzugefügt werden.

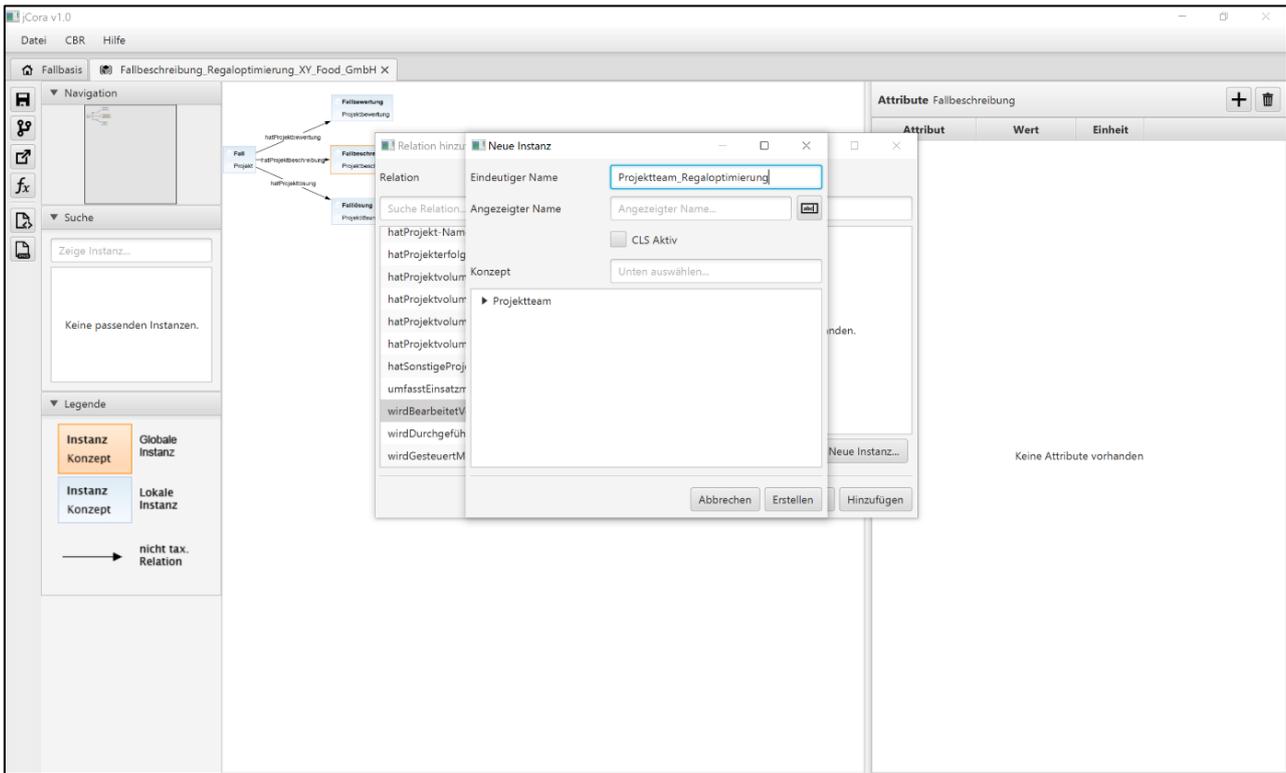


Abbildung 41: Erstellen von Instanzen in jCORA – Teil 1

Hier wird im Feld „Eindeutiger Name“ die Bezeichnung der Instanz eingefügt. Anschließend kann durch Anklicken des auf der rechten Seite befindlichen Texteingabe-Symbols optional ein mehrsprachiger Anzeigename vergeben und mit „Übernehmen“ bestätigt werden.

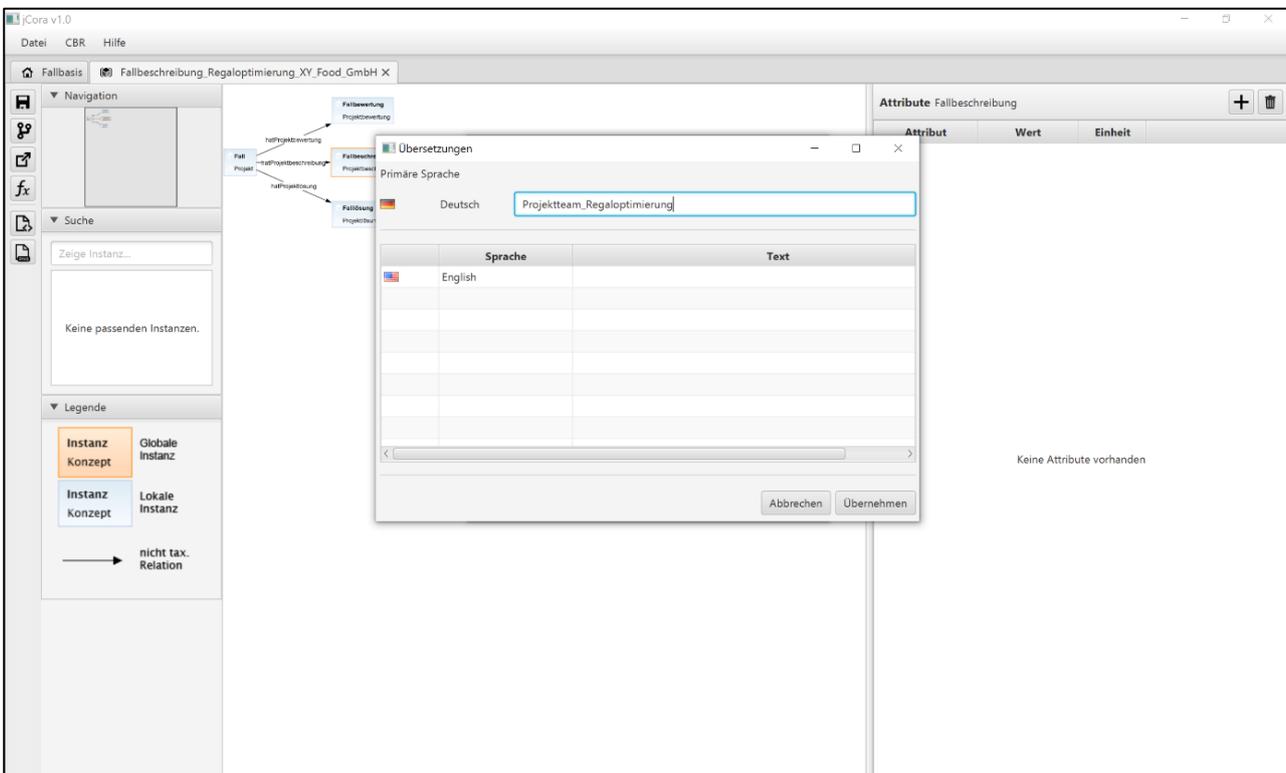


Abbildung 42: Erstellen von Instanzen in jCORA – Teil 2

Im Anschluss wird die jeweilige Klasse ausgewählt, auf die sich die Instanz bezieht, und mit „Speichern“ bestätigt.

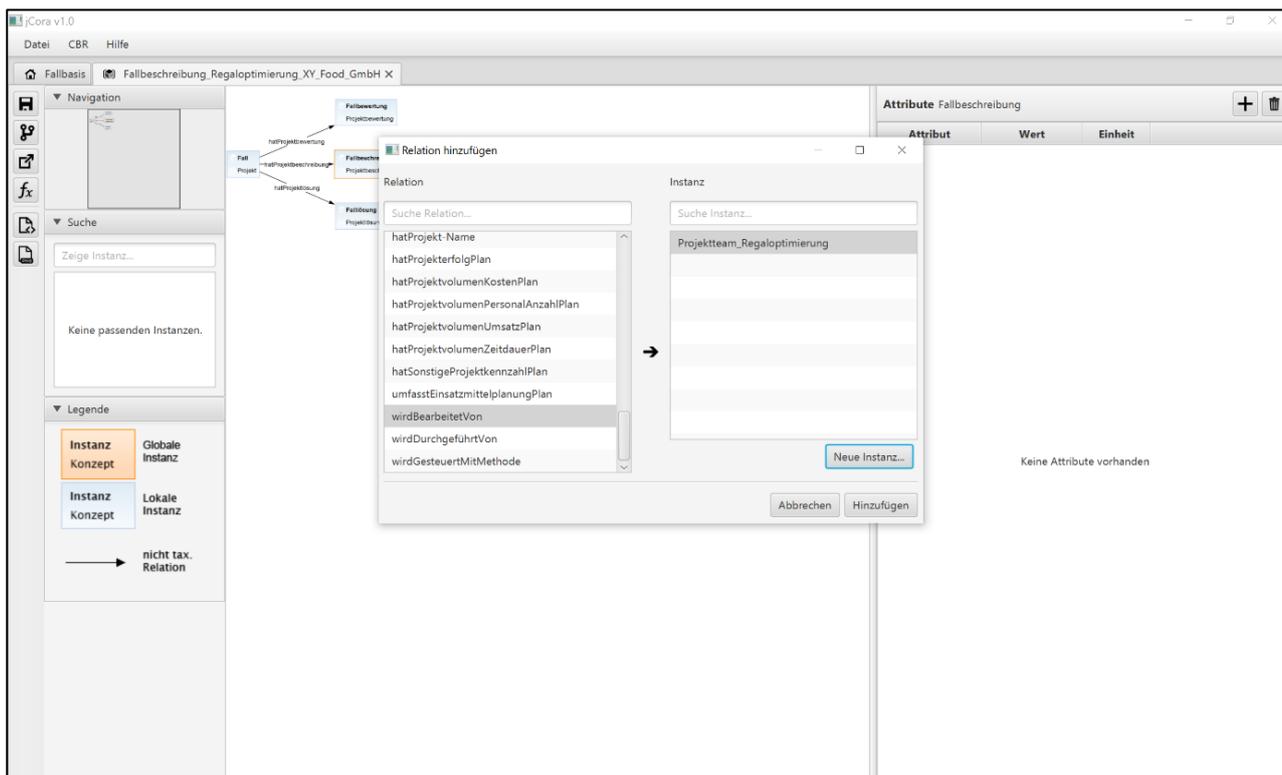


Abbildung 43: Hinzufügen von Instanzen in jCORa

Die neu erstellte Relation ist jetzt im Fallgraphen sichtbar. Nach diesem Schema können weitere Relationen hinzugefügt werden.

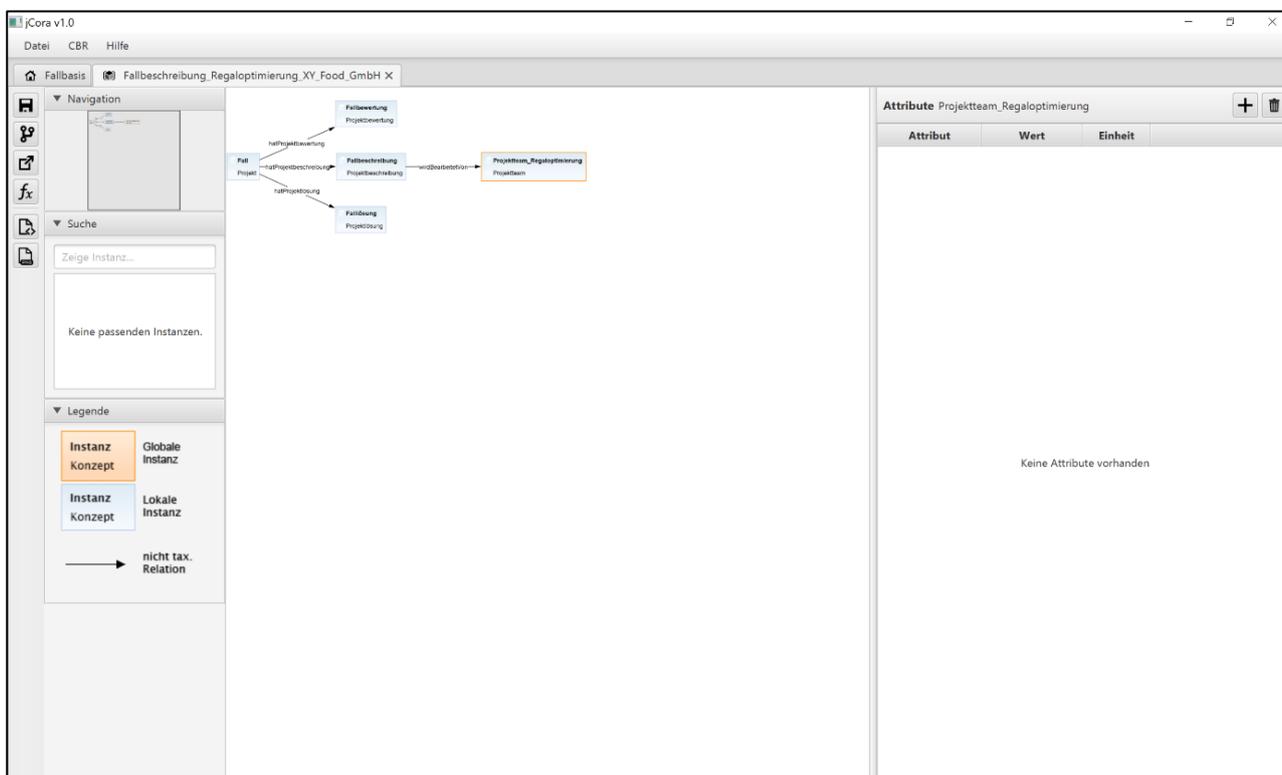


Abbildung 44: Falleditor nach Hinzufügen einer Relation

Die in der Arbeit erstellten und verwendeten Fallgraphen wurden gemäß dem oben beschriebenen Schema erstellt.

Anhang 5: Fallgraphen des Projektberichts

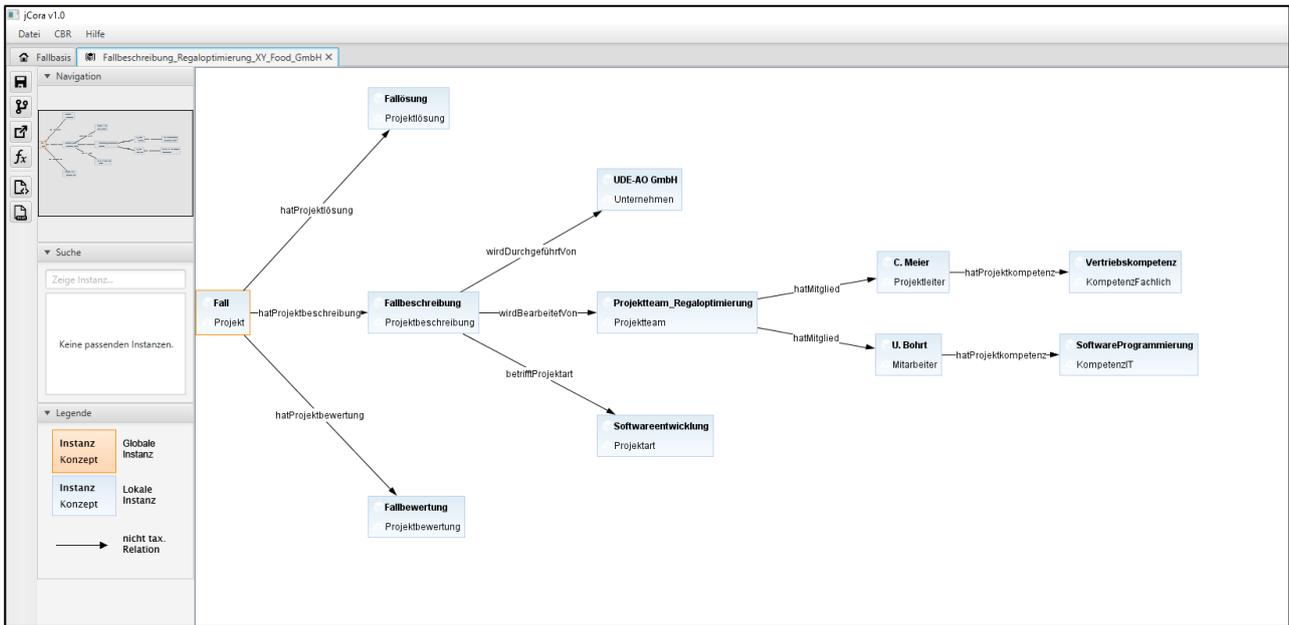


Abbildung 45: Fallgraph des Falls „Regaloptimierung“

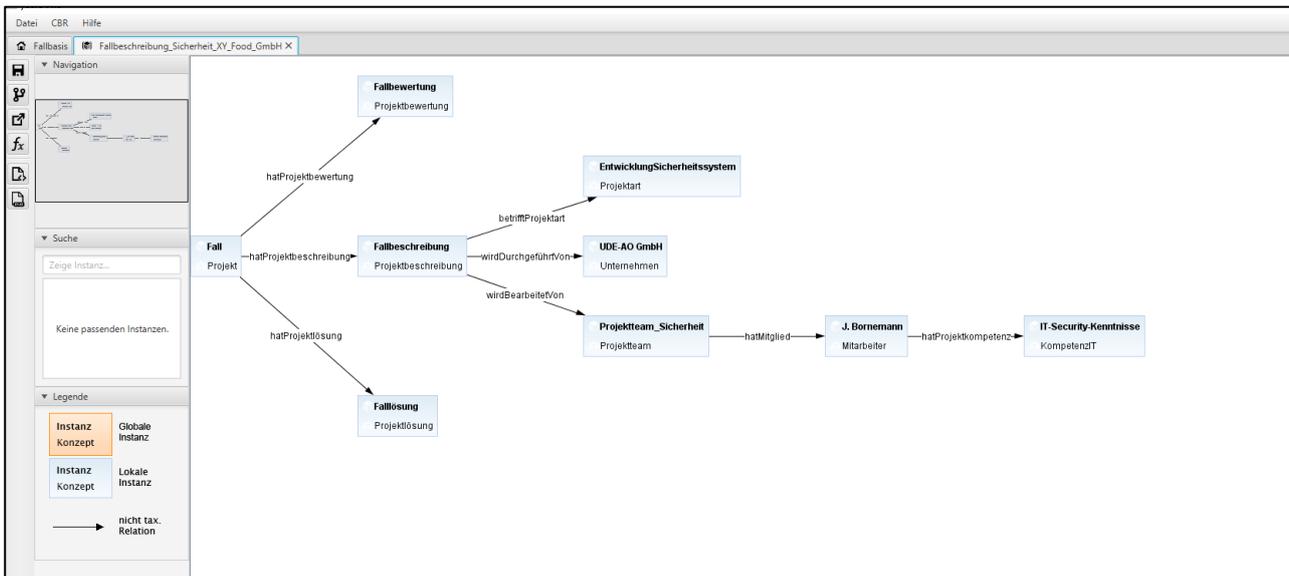


Abbildung 46: Fallgraph des Falls „Sicherheit“

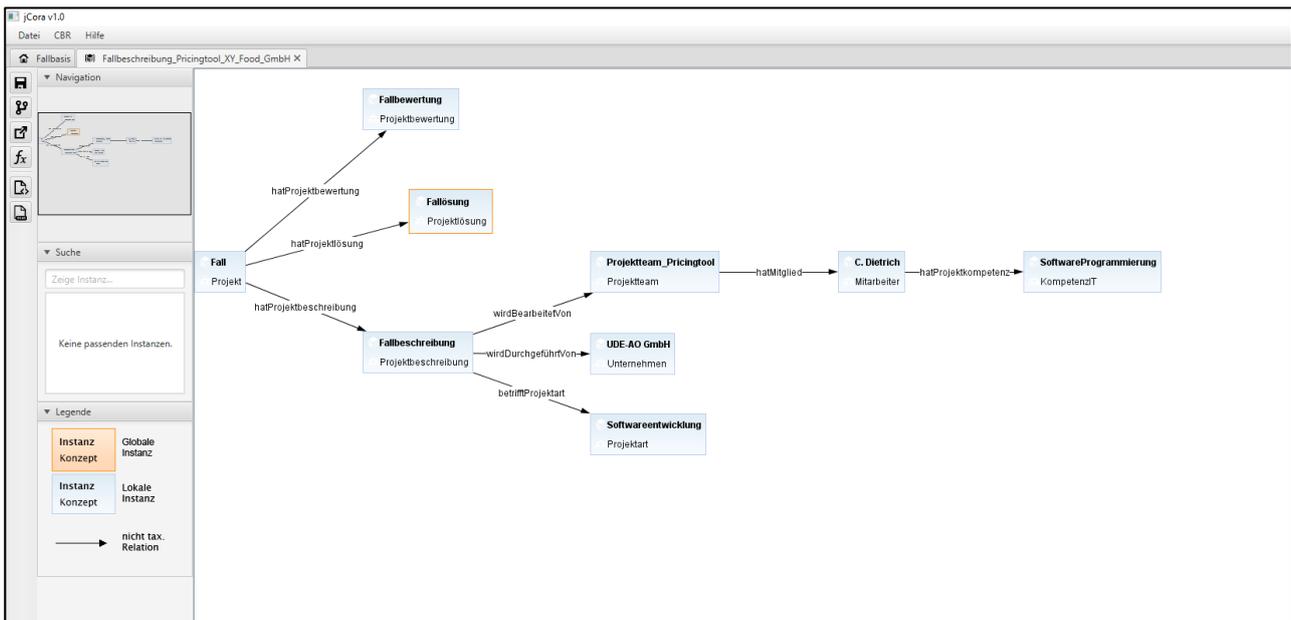


Abbildung 47: Fallgraph des Falls „Pricingtool“

Anhang 6: Neue CBR-Anfrage und Falladaption in jCORA

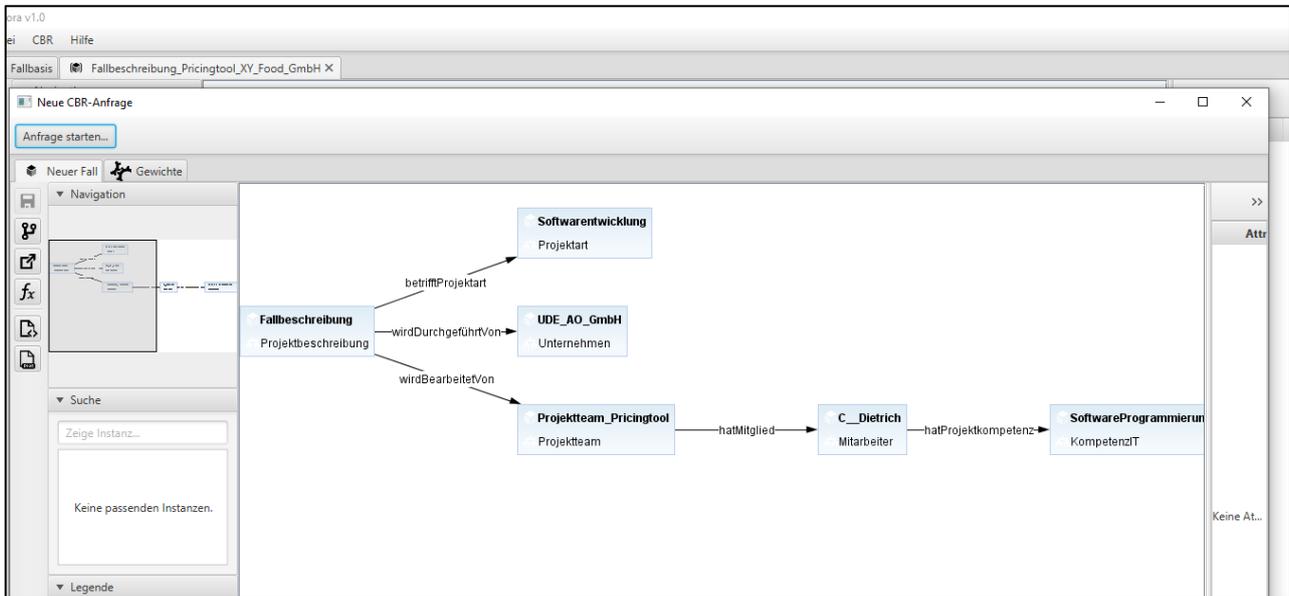


Abbildung 48: CBR-Anfrage in jCORA

Relation	Gewicht	%
beziehtSichAufProjektloesung	100.0	100.0
erfordertKompetenzenFachlichPlan	100.0	100.0
hatPflichtenheft	100.0	100.0
erfordertKompetenzenSprachlichIst	100.0	100.0
hatAuftraggeber	100.0	100.0
betrifftKompetenzKulturellIst	100.0	100.0
hatProjekt-ID-ImEigenenUnternehmen	100.0	100.0
gehörtZuProjekt	100.0	100.0
hatSonstigeProjektkennzahlPlan	100.0	100.0
hatKompetenzenFachlich	100.0	100.0
betrifftKompetenzFachlichAbweichung	100.0	100.0
betrifftProjektvolumenUmsatzPlan	100.0	100.0
betrifftKompetenzITPlan	100.0	100.0
hatMitglied	100.0	100.0
erfordertKompetenzenKulturellPlan	100.0	100.0
istVerantwortlichFürOrganisationseinheit	100.0	100.0
erfordertKompetenzenSonstigeAbweichung	100.0	100.0
erfordertKompetenzenSonstigelst	100.0	100.0
betrifftKooperationMitStakeholder	100.0	100.0
betrifftProjektvolumenUmsatzAbweichung	100.0	100.0
betrifftAbteilung	100.0	100.0

Abbildung 49: Gewichte-Ansicht in jCORA

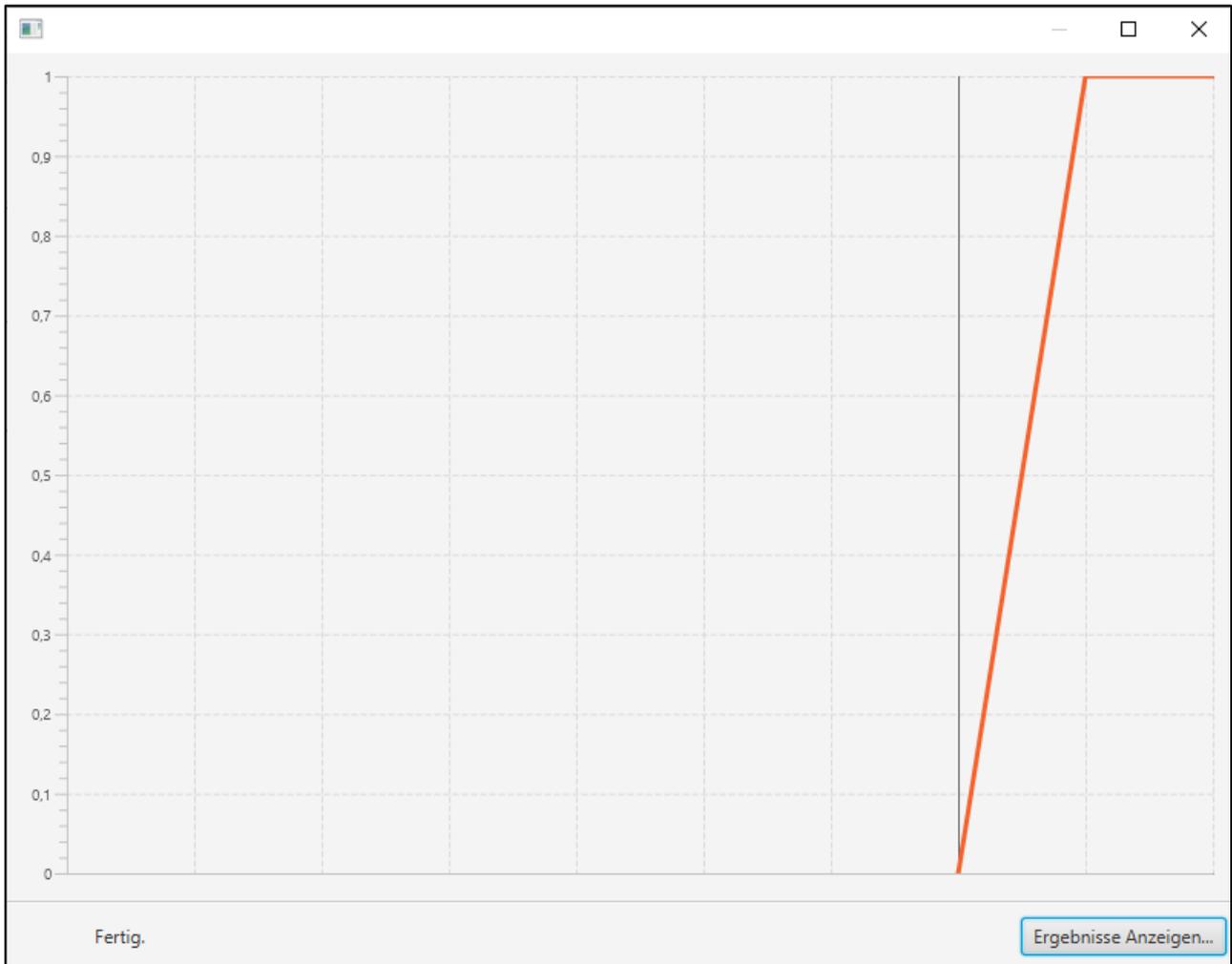


Abbildung 50: Anzeige nach Abschluss der Ähnlichkeitsberechnung

jCora v1.0

Datei CBR Hilfe

Fallbasis Anfrageergebnisse X

Filter

Zeige >= % ...

Fall-ID	Ähnlichkeit	Adaptieren	Anzeigen
Fallbeschreibung_Pricingtool_XY_Food_G...	100%	<input checked="" type="checkbox"/>	Anzeigen
Fallbeschreibung_Sicherheit_XY_Food_G...	100%	<input checked="" type="checkbox"/>	Anzeigen
Fallbeschreibung_Regaloptimierung_XY_...	91%	<input type="checkbox"/>	Anzeigen

Abbildung 51: Anfrageergebnisse in jCORA

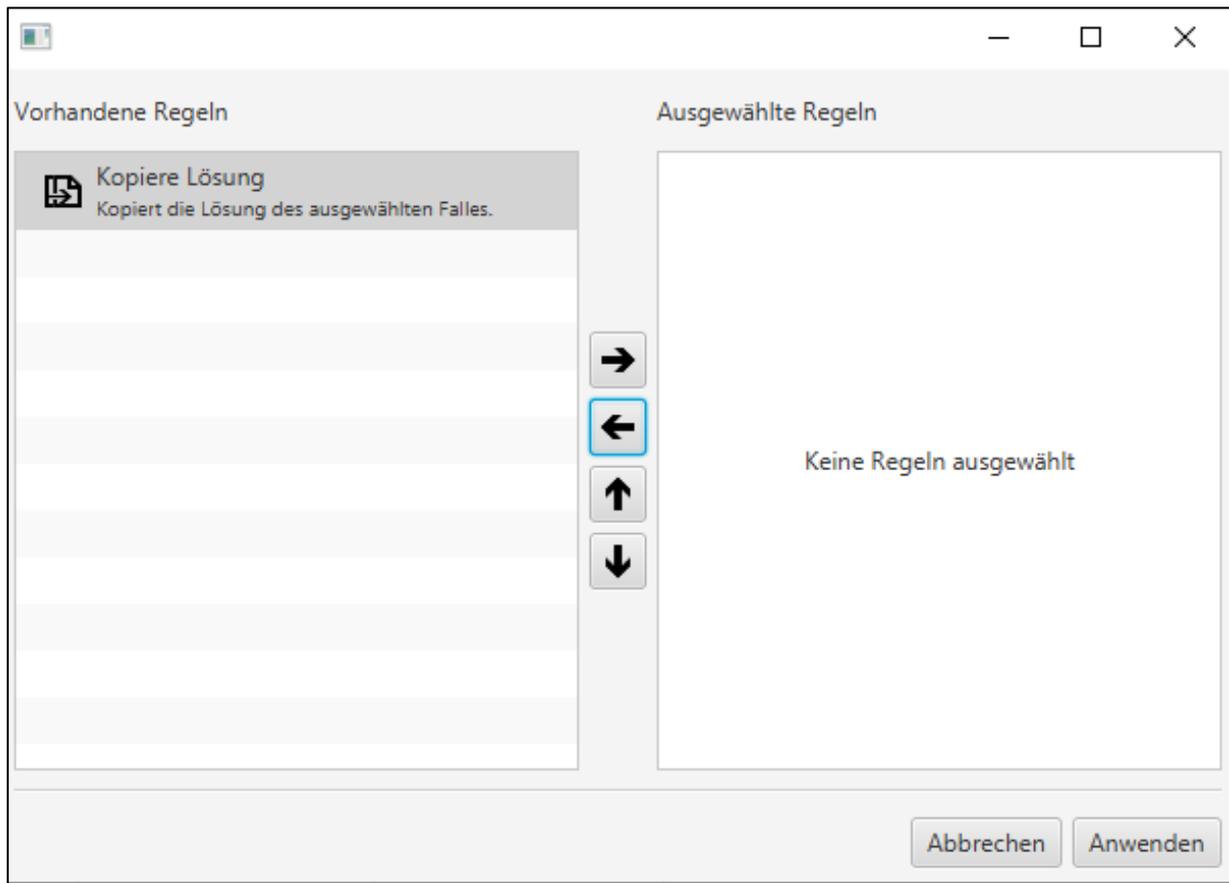


Abbildung 52: Falladaption – Teil 1

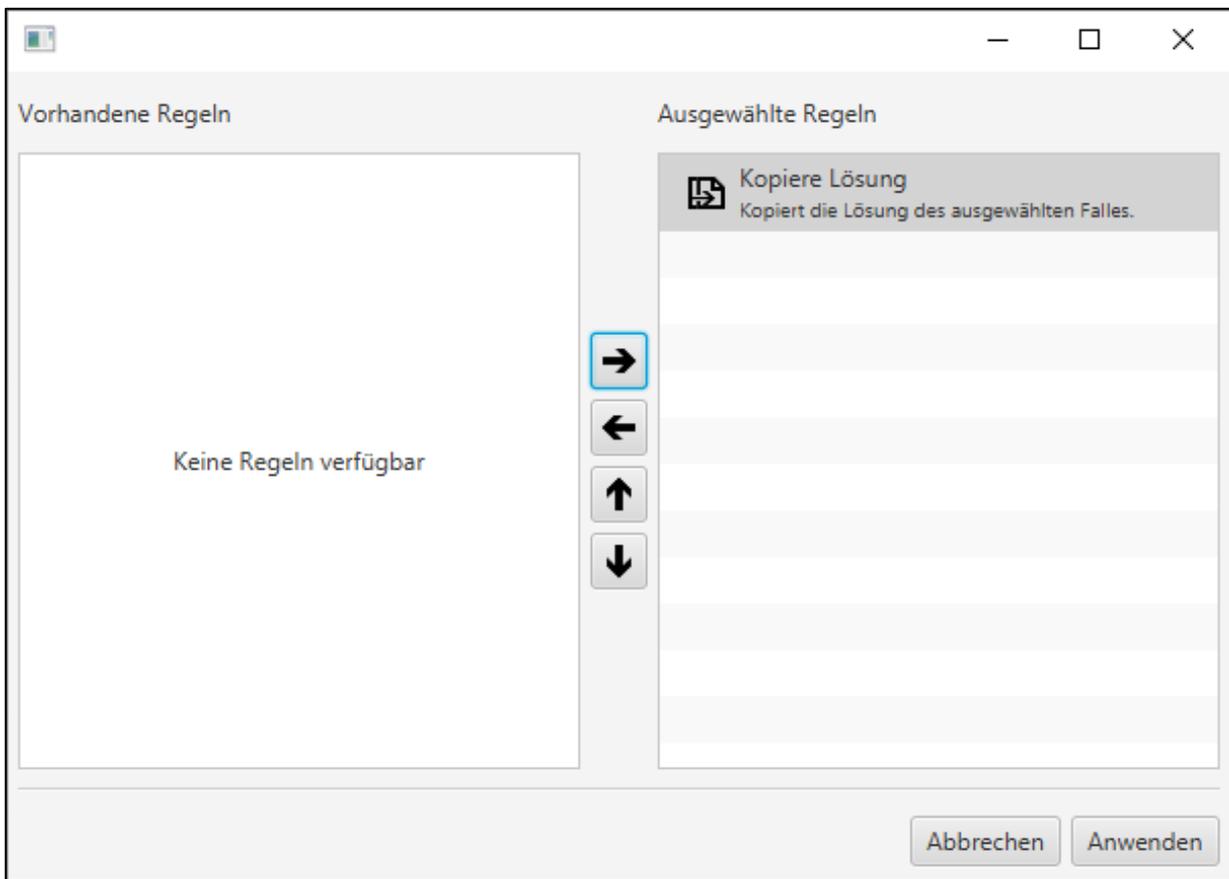


Abbildung 53: Falladaption – Teil 2

**Institut für Produktion und
Industrielles Informationsmanagement
Universität Duisburg-Essen / Campus Essen**

**Verzeichnis der Arbeitsberichte
(ISSN 1614-0842)**

- Nr. 1: Zelewski, S.: Stickels theoretische Begründung des Produktivitätsparadoxons der Informationstechnik. Universität Essen, Essen 1999.
- Nr. 2: Zelewski, S.: Flexibilitätsorientierte Koordinierung von Produktionsprozessen. Universität Essen, Essen 1999.
- Nr. 3: Zelewski, S.: Ontologien zur Strukturierung von Domänenwissen. Universität Essen, Essen 1999.
- Nr. 4: Siedentopf, J.; Schütte, R.; Zelewski, S.: Wirtschaftsinformatik und Wissenschaftstheorie. Universität Essen, Essen 1999.
- Nr. 5: Fischer, K.; Zelewski, S.: Ontologiebasierte Koordination von Anpassungsplanungen in Produktions- und Logistiknetzwerken mit Multi-Agenten-Systemen. Universität Essen, Essen 1999.
- Nr. 6: Weihermann, A. E.; Wöhlert, K.: Gentechnikakzeptanz und Kommunikationsmaßnahmen in der Lebensmittelindustrie. Universität Essen, Essen 1999.
- Nr. 7: Schütte, R.: Zum Realitätsbezug von Informationsmodellen. Universität Essen, Essen 2000.
- Nr. 8: Zelewski, S.: Erweiterungen eines Losgrößenmodells für betriebliche Entsorgungsprobleme. Universität Essen, Essen 2000.
- Nr. 9: Schütte, R.: Wissen, Zeichen, Information, Daten. Universität Essen, Essen 2000.
- Nr. 10: Hemmert, M.: The Impact of Internationalization and Externalization on the Technology Acquisition Performance of High-Tech Firms. Universität Essen, Essen 2001.
- Nr. 11: Hemmert, M.: Erfolgswirkungen der internationalen Organisation von Technologiegewinnungsaktivitäten. Universität Essen, Essen 2001.
- Nr. 12: Hemmert, M.: Erfolgsfaktoren der Technologiegewinnung von F&E-intensiven Großunternehmen. Universität Essen, Essen 2001.
- Nr. 13: Schütte, R.; Zelewski, S.: Epistemological Problems in Working with Ontologies. Universität Essen, Essen 2001.
- Nr. 14: Peters, M. L.; Zelewski, S.: Analytical Hierarchy Process (AHP). Universität Essen, Essen 2002.
- Nr. 15: Zelewski, S.: Wissensmanagement mit Ontologien. Universität Essen, Essen 2002.
- Nr. 16: Klumpp, M.; Krol, B.; Zug, S.: Management von Kompetenzprofilen im Gesundheitswesen. Universität Essen, Essen 2002.
- Nr. 17: Zelewski, S.: Der „non statement view“ – eine Herausforderung für die (Re-) Konstruktion wirtschaftswissenschaftlicher Theorien. Universität Essen, Essen 2002.
- Nr. 18: Peters, M. L.; Zelewski, S.: A heuristic algorithm to improve the consistency of judgments in the Analytical Hierarchy Process (AHP). Universität Duisburg-Essen (Campus Essen), Essen 2003.

- Nr. 19: Peters, M. L.; Zelewski, S.: Fallstudie zur Lösung eines Standortplanungsproblems mit Hilfe des Analytical Hierarchy Process (AHP). Universität Duisburg-Essen (Campus Essen), Essen 2003.
- Nr. 20: Zelewski, S.: Konventionelle versus strukturalistische Produktionstheorie. Universität Duisburg-Essen (Campus Essen), Essen 2003.
- Nr. 21: Alparslan, A.; Zelewski, S.: Moral Hazard in JIT Production Settings. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 22: Dittmann, L.: Ontology-based Skills Management. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 23: Peters, M. L.; Zelewski, S.: Ein Modell zur Auswahl von Produktionsaufträgen unter Berücksichtigung von Synergien. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 24: Peters, M. L.; Zelewski, S.: Ein Modell zur Zuordnung ähnlicher Kundenbetreuer zu Kunden. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 25: Zelewski, S.: Kooperatives Wissensmanagement in Engineering-Netzwerken – (vorläufiger) Abschlussbericht zum Verbundprojekt KOWIEN. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 26: Siemens, F.: Vorgehensmodell zur Auswahl einer Variante der Data Envelopment Analysis. Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 27: Alan, Y.: Integrative Modellierung kooperativer Informationssysteme – Ein Konzept auf der Basis von Ontologien und Petri-Netzen. Dissertation, Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 28: Akca, N.; Ilas, A.: Produktionsstrategien – Überblick und Systematisierung. Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 29: Zelewski, S.: Relativer Fortschritt von Theorien – ein strukturalistisches Rahmenkonzept zur Beurteilung der Fortschrittlichkeit wirtschaftswissenschaftlicher Theorien (Langfassung). Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 30: Peters, M. L.; Schütte, R.; Zelewski, S.: Erweiterte Wirtschaftlichkeitsanalyse mithilfe des Analytic Hierarchy Process (AHP) unter Berücksichtigung des Wissensmanagements zur Beurteilung von Filialen eines Handelsunternehmens. Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 31: Zelewski, S.: Beurteilung betriebswirtschaftlichen Fortschritts – ein metatheoretischer Ansatz auf Basis des „non statement view“ (Langfassung). Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 32: Kijewski, F.; Moog, M.; Niehammer, M.; Schmidt, H.; Schröder, K.: Gestaltung eines Vorgehensmodells für die Durchführung eines Promotionsprojekts am Fachbereich Wirtschaftswissenschaften der Universität Duisburg-Essen, Campus Essen, zum Erwerb des „Dr. rer. pol.“ mithilfe von PETRI-Netzen. Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 33: Peters, M. L.; Zelewski, S.: Effizienzanalyse unter Berücksichtigung von Satisfizierungsgrenzen für Outputs – Die Effizienz-Analysetechnik EATWOS. Universität Duisburg-Essen (Campus Essen), Essen 2006.

- Nr. 34: Häselhoff, I.; Meves, Y.; Munsch, D.; Munsch, S.; Schulte-Euler, D.; Thorant, C.: Anforderung an eine verbesserte Lehrqualität – Qualitätsplanung mittels House of Quality. Universität Duisburg-Essen (Campus Essen), Essen 2007.
- Nr. 35: Zelewski, S.: Das ADL-Modell der Prinzipal-Agent-Theorie für die Just-in-Time-Produktionssteuerung – Darstellung, Analyse und Kritik. Universität Duisburg-Essen (Campus Essen), Essen 2008.
- Nr. 36: Peters, M. L.; Zelewski, S.: Analyse der Effizienzentwicklung von Bankfilialen mithilfe des Operational Competitiveness Ratings (OCRA). Universität Duisburg-Essen (Campus Essen), Essen 2010.
- Nr. 37: Peters, M. L.; Zelewski, S.: Fallstudie zu Porters generischen Wettbewerbsstrategien im Kontext nachhaltigen Wirtschaftens. Universität Duisburg-Essen (Campus Essen), Essen 2010.
- Nr. 38: Peters, M. L.; Zelewski, S.: Erweiterung von EATWOS um die Berücksichtigung von Satisfizierungsgrenzen für Inputs. Universität Duisburg-Essen (Campus Essen), Essen 2012.
- Nr. 39: Bergenrodt, D.; Jene, S.; Zelewski, S.: Implementierung des Tau-Werts. Universität Duisburg-Essen (Campus Essen), Essen 2013.
- Nr. 40: Millan-Torres, J.; Arndt, C.: Erstellung eines Businessplans zur Existenzgründung des Unternehmens Cowdy! – Anwendung des „Fast-Casual“-Konzepts auf ein systemgastronomisch organisiertes Restaurant mit dem Schwerpunkt der Steakzubereitung. Universität Duisburg-Essen (Campus Essen), Essen 2014.
- Nr. 41: Klumpp, M.; Oeben, M.; Zelewski, S.: Evaluation internationaler Bildungstransfer – Konzeptioneller Rahmen und Diskurs zur wissenschaftlichen Bewertung im Forschungs- und Transferprojekt OpporTUNItY. Universität Duisburg-Essen (Campus Essen), Essen 2018.
- Nr. 42: Oeben, M.; Gerlach, A.-T.; Akdogan, D.; Arabaci, T.; Bagbasi, F.; Gudieva, A.; Klumpp, M.: Evaluation von Bildungsleistungen in Deutschland und Tunesien – das Beispiel des Hochschulsektors. Universität Duisburg-Essen (Campus Essen), Essen 2018.
- Nr. 43: Oeben, M.; Klumpp, M.: Die Berufsschulsysteme in Tunesien und Deutschland – Ein systematischer Vergleich im Rahmen der wissenschaftlichen Evaluation des Projektes OpporTUNItY. Universität Duisburg-Essen (Campus Essen), Essen 2018.
- Nr. 44: Peters, M. L.; Zelewski, S.: Adaption der Efficiency Analysis Technique With Input and Output Satisficing (EATWIOS) zur Berücksichtigung von unteren und oberen Satisfizierungsgrenzen. Universität Duisburg-Essen (Campus Essen), Essen 2018.
- Nr. 45: Oeben, M.; Klumpp, M.: Export von Expertise im Bereich der Berufsausbildung – Erfolgsfaktoren und Hemmnisse für den Aufbau und Betrieb eines technischen Berufsschulzentrums in Tunesien im Forschungs- und Transferprojekt OpporTUNItY. Universität Duisburg-Essen (Campus Essen), Essen 2019.
- Nr. 46: Oeben, M.; Klumpp, M.; Zelewski, S.: Internationaler Bildungstransfer – Internationaler Quervergleich als komparativer Ansatz zu Erfahrungen im Bildungstransfer in Richtung Tunesien. Universität Duisburg-Essen (Campus Essen), Essen 2019.

- Nr. 47: Schagen, J. P.; Zelewski, S.; Heeb, T.: Erhebung und Analyse der Anforderungen an ein KI-Tool aus der Perspektive der betrieblichen Praxis – mit Fokus auf der Wiederverwendung von Erfahrungswissen im Bereich des betrieblichen Projektmanagements. Zugleich KI-LiveS-Projektbericht Nr. 1. Universität Duisburg-Essen (Campus Essen), Essen 2020.
- Nr. 48: Schagen, J. P.; Zelewski, S.; Haselhoff, T.; Schmitz, S.; Heeb, T.: Überblick über potenzielle Quellen für Test- und Evaluierungsdaten eines KI-Labors im Rahmen des KI-LiveS-Projekts. Zugleich KI-LiveS-Projektbericht Nr. 2. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 49: Fink, S.; Röhrig, K.; Heeb, T. (Mitarbeit Schagen, J. P.; Zelewski, S.): Konzipierung und Implementierung eines ontologiegestützten Case-based-Reasoning-Systems für die Wiederverwendung von projektbezogenem Erfahrungswissen. Zugleich KI-LiveS-Projektbericht Nr. 3. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 50: Weber, L.; Heeb, T.; Sethupathy, G. (Mitarbeit Schagen, J. P.; Zelewski, S.): „Intelligente“ Wiederverwendung von Erfahrungswissen im betrieblichen Projektmanagement mithilfe von KI-Techniken bei sicherheitskritischen IT-Projekten mit Fokus auf PRINCE2 und Risikomanagement. Zugleich KI-LiveS-Projektbericht Nr. 4. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 51: Allam, S.; Heeb, T.; Zelewski, S.: Konzipierung und Implementierung eines E-Learning-Moduls für ein ontologiegestütztes Case-based Reasoning Tool zur Unterstützung des Projektmanagements im Rahmen des KI-LiveS-Projekts. Zugleich KI-LiveS-Projektbericht Nr. 5. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 52: Weber, L.; Allam, S.; Camgöz, A. (Mitarbeit Heeb, T.; Zelewski, S.): Erstellung eines E-Learning-Moduls für den Ontologie-Editor Protégé. Zugleich KI-LiveS-Projektbericht Nr. 6. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 53: Fink, S.; Hauke, M.; Ye, B. (Mitarbeit Schagen, J. P.; Zelewski, S.): Erstellung und kritische Analyse von Use Cases für Anwendungen von KI-Tools im betrieblichen Projektmanagement – mit Fokussierung auf der „intelligenten“ Wiederverwendung von projektbezogenem Erfahrungswissen. Zugleich KI-LiveS-Projektbericht Nr. 7. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 54: Schagen, T.; Heeb, T.; Zelewski, S. (Mitarbeit Schagen, J. P.): Entwicklung eines E-Learning-Moduls für ein ontologiegestütztes Case-based Reasoning Tool für das betriebliche Projektmanagement. Zugleich KI-LiveS-Projektbericht Nr. 8. Universität Duisburg-Essen (Campus Essen), Essen 2021.
- Nr. 55: Zelewski, S.; Schagen, J. P.: Case-based Reasoning als KI-Technik zur „intelligenten“, computergestützten Wiederverwendung von Erfahrungswissen im Projektmanagement. Zugleich KI-LiveS-Projektbericht Nr. 9. Universität Duisburg-Essen (Campus Essen), Essen 2022.
- Nr. 56: Bornemann, J.; Heeb, T.; Zelewski, S. (Mitarbeit Schagen, J. P.): Ein Vorgehensmodell zur Entwicklung ontologiegestützter Case-based-Reasoning-Systeme. Zugleich KI-LiveS-Projektbericht Nr. 10. Universität Duisburg-Essen (Campus Essen), Essen 2022.
- Nr. 57: Herder, M.-M.; Zelewski, S.; Schagen, J. P.: Evaluation des Prototyps jCORA im Rahmen des KI-LiveS-Projekts hinsichtlich Anforderungen an die „intelligente“ Wiederverwendung von Erfahrungswissen im Projektmanagementbereich. Zugleich KI-LiveS-Projektbericht Nr. 11. Universität Duisburg-Essen (Campus Essen), Essen 2022.

- Nr. 58: Mustafa, E.; Heeb, T.; Zelewski, S.: Ontologiegestütztes Case-based Reasoning im Projektmanagement – Einsatz künstlicher Intelligenz zur Wiederverwendung von projektbezogenem Erfahrungswissen. Zugleich KI-LiveS-Projektbericht Nr. 12. Universität Duisburg-Essen (Campus Essen), Essen 2022.