

Ansatz zur regelbasierten mehrskaligen Analyse und Prüfung von Systemmodellen im Kontext der integralen Planung

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

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

genehmigte

Dissertation

von

Sebastian Ebertshäuser

aus Stuttgart

Tag der mündlichen Prüfung: 23. Juni 2023

Erste Gutachterin: Prof. Dr.-Ing. Petra von Both

Zweiter Gutachter: Prof. Dr.-Ing. Gunnar Grün

Kurzfassung

Einleitung – Integrale Planung ist als ein anforderungsorientierter Regelmechanismus zu verstehen, welcher eine frühzeitige prozessbegleitende Evaluierung und Prüfung der Planungslösungen beinhaltet. Ein besonderes Augenmerk ist dabei auf eine ganzheitliche fachübergreifende Herangehensweise zu legen. Neben den Expertenwerkzeugen zur Simulation und Berechnung einzelner fachspezifischer Belange, wie bauphysikalischer, energetischer oder statischer Belange, erscheint daher eine fachübergreifende ganzheitliche Bewertung und Prüfung erforderlich. Bezugnehmend zu dieser aktuellen Diskussion hat das Thema der Analyse und Qualitätssicherung kooperativ bearbeiteter Informationen an Bedeutung gewonnen insbesondere auch bei langandauernden und vielschichtigen kommunalen Planungs- und Ausführungsprozessen mit der zunehmenden Etablierung entsprechender offener modellbasierter Austauschformate auf Gebäude- und Stadtebene (vgl. BIM, GIS).

Zielsetzung – Diese Dissertation ist im Kontext der Entwicklung und Bereitstellung integraler Planungshilfsmitteln für die Planer- und Entscheider*innenebene verortet. Sie beschäftigt sich mit der Fragestellung, wie Ergebnisse, welche aus Simulation und Berechnung in den Planungsprozess zurückgeführt wurden, fach- und aspektübergreifenden Analyse- und Auswertungsprozessen zugänglich gemacht werden können.

Gemäß einem Lösungsansatz der auf Erkenntnissen und Vorarbeiten aus zwei Forschungsprojekten am Fachgebiet Building Lifecycle Management des KIT (Integriertes systemisches Informationsmodell Stadt – ISIS sowie EnOB EnEff-BIM) aufbaut, soll hierzu ein regelbasiertes Planungswerkzeug entwickelt und anhand eines Beispielszenarios evaluiert werden.

Lösungsansatz – Der Ansatz der modellbasierten Planung bildet die strukturelle Basis. Erst eine kohärente gemeinsame Informationsbasis des verteilten Arbeitens über den ganzen Verlauf kooperativer Planungsprozesse ermöglicht die Generierung von Mehrwert im Sinne der Integralen Planung. Aber auch diese, sich in der Gebäudeplanung bereits etablierende ganzheitliche Methodik

des modellbasierten Planens (vgl. Building Information Modeling – BIM) fordert in ihrer konsequenten Anwendung das Aufgreifen von begleitenden Maßnahmen zur Qualitätssicherung der Datengrundlage.

Auf der Ebene des Bauwerks existieren bereits erste Ansätze von Theoriebildung und von Werkzeugen zur inhaltlichen Modellprüfung (vgl. Solibri Model Checker), zumeist für den internationalen herstellerneutralen Modellstandard IFC (Industry Foundation Classes). Kontextspezifische Prüflogik, wie beispielsweise für energetische Fragestellungen, sind in diesen allgemeinen Ansätzen allerdings nicht verfügbar. Diese Entwicklung ist Inhalt aktueller Forschungsprojekte, wie das vom Wirtschaftsministerium geförderte Verbundvorhaben EnEff-BIM, an dem das Fachgebiet aktiv beteiligt war.

Trotz der sich zunehmend etablierenden Modellbasis CityGML zum Austausch kooperativ bearbeiteter kommunaler Informationen, fehlen für planerische Fragestellungen auf dem urbanen Maßstab bisher sowohl modellbasierte Werkzeuge wie auch grundlegende Methoden. Durch das Aufzeigen gezielter Unterstützung qualitätsgesicherter Austausch- und Auswertungsprozesse modellbasierter Inhalte kann für die kommunale Ebene – wie sich auf der Gebäudeebene bereits zeigt – eine Einstiegshürde in das modellbasierte Arbeiten signifikant gesenkt werden. Relevant erscheint hierbei neben dem Bedienen der im Quartierskontext verwendeten Stadtmodellstandards auch die Möglichkeit des Einbeziehens spezifischer Detailinformationen beispielsweise aus dem Gebäudebereich.

Daher wird im Lösungsansatz ein mehrskaliges Konzept vorgeschlagen. Forschungsseitig sind bereits Ansätze bezüglich spezifischer Modellstandards erkennbar, die in ausgewählten szenariobasierten Anwendungskontexten Analysewerkzeuge als Planungshilfsmittel bereitstellen und als Ausgangsbasis einer weiterführenden Untersuchung herangezogen werden können. (ISIS, EnEff-BIM, Simstadt, Berlin Energy Atlas usw.).

Prinzipiell ermöglicht der Lösungsansatz für die Planer- und Entscheider*inenebene als Planungshilfsmittel eine Prüfung und Analyse der modellbasierten Planungsgegenstände zum Aufbereiten von planungsbegleitenden Entscheidungsgrundlagen.

Vorgehen, Rahmenbedingungen und Arbeitsumfeld – Im Rahmen der Dissertation werden für den verfolgten mehrskaligen Lösungsansatz modellübergreifende, schemaneutrale Methoden untersucht. Diese werden konzeptionell ausformuliert und prototypisch in einer Softwareanwendung umgesetzt. Im Kontext eines Szenarios, das in der Gebäudeplanung ansetzt, werden die Werkzeuge des Planungshilfsmittels exemplarisch auf eine spezifische Entscheidungssituation angepasst. Anhand des Einsatzes der Softwareanwendung wird der Lösungsansatz evaluiert.

Bereits erfolgte Vorarbeiten des Autors hatten die Entwicklung eines regelbasierten Prüfwerkzeugs und dessen Anpassung an verschiedene Anwendungskontexte (Gebäude, Stadt) zum Inhalt. Auf dieser technologischen Basis soll für die Ergebnisse der Untersuchung prototypisch die Machbarkeit im festgelegten Szenario evaluiert werden.

Generell boten die am Fachgebiet BLM laufenden Lehr- und Forschungstätigkeiten im Bereich Systems Engineering, der Bauinformatik und integrativen Planungsmethodik dem Autor ein geeignetes fachliches Arbeitsumfeld zur Erstellung seiner Dissertation. Zum Themenfeld der modellbasierten Planung in frühen Projektphasen und der bauspezifischen Produktmodellierung lagen am BLM bereits geeignete Vorarbeiten und Informationen vor, auf die zurückgegriffen werden konnte.

Inhaltsverzeichnis

Kurzfassung	i
Inhaltsverzeichnis	vii
Abbildungsverzeichnis	xi
Tabellenverzeichnis	xvii
Abkürzungsverzeichnis	xix
Danksagung	xxvii
1 Einleitung	1
1.1 Problemstellung und Motivation	2
1.2 Zielsetzung	10
1.3 Gliederung und Aufbau der Arbeit	12
2 Grundlagen	15
2.1 Integrale Planung	16
2.1.1 Aktualität und Hintergrund	16
2.1.2 Paradigma und Voraussetzung	20
2.1.3 Grundlegende Anforderungen	22
2.1.4 Baustein (Building) Information Modeling	25
2.1.5 Methodische und technische Hilfsmittel	28
2.2 Modellierung	35
2.2.1 Begriffsklärung im Sinne der Arbeit	35
2.2.2 Gegenstand Daten	39
2.2.3 Produktmodellierung	48
2.3 Modelldatenverarbeitung	51
2.3.1 Objektorientierte Programmierung	52
2.3.2 Standardisierte Datenhandhabung	54
2.3.3 Datenqualitätsmanagement	59
3 Lösungsansatz und weiterführende Anforderungen	73
3.1 Lösungsansatz	74

3.2	Konkretisierung des Lösungsansatzes anhand von Anforderungen	79
3.2.1	Was ist nötig für das Konzept?	80
3.2.2	Konzeptionelle Festlegungen zur Werkzeugentwicklung ...	97
3.2.3	Konzeptionelle Festlegungen bezüglich des Kontextes.....	105
4	Anwendungsszenario.....	112
4.1	Herleitung objektbezogener Anforderungen der Objektplanung ..	115
4.2	Anwendungsfall des Lösungsansatzes zur Unterstützung einer Entwurfsentscheidung.....	123
4.3	Weiterführende informationstechnische und methodische Aspekte zum Anwendungsszenario	133
4.4	Wiederkehrende (vorbereitende) Arbeitsschritte zur formalstrukturellen Qualitätssicherung der Modellinformationen	142
4.4.1	Prüfen der Vollständigkeit von Modellinhalten der Energieplanung vorbereiten.....	145
4.4.2	Analyse zur Zieleinhaltung der Energieplanung vorbereiten	148
4.4.3	Potentielle Backoffice Routinen: Aspekte des Modellhandling verstetigt als Bürostandard.....	151
4.5	Zusammenfassung modellbezogener und methodischer Anforderungen im Lastenheft.....	153
5	Umsetzung der Ebene des dynamischen Anwendungsansatzes mit Werkzeugen für modellbasierte Analyseprozesse	159
5.1	Komponentenentwicklung als Grundlage modularer Softwarearchitektur	160
5.1.1	Basiskomponenten des BLM ToolKIT.....	162
5.1.2	Weitere (eigenständige) der Desktop-Softwareanwendung zugrundeliegende Komponenten	165
5.2	Integration der Komponenten in die Architektur der Desktop-Anwendungssoftware	175
5.2.1	Benutzeroberfläche (UI) Ebene – Hauptfenster und enthaltene Werkzeugansichten (Views)	178
5.2.2	Business Logic (BL) Ebene – Steuerung der Nutzerinteraktion und Datenhandhabung	180

6 Umsetzung der Ebene der Methoden und Hilfsmittel für die Analyse- bzw. Prüflogik.....	198
6.1 Umsetzungsstrategie zur gestuften Anpassung der Softwareanwendung an den Anwendungsfall.....	200
6.1.1 Entwicklungsstufe des Regel-Grundstocks	204
6.1.2 Entwicklungsstufe der Projektregeln	206
6.1.3 Entwicklungsstufe der Situationsregeln	208
6.2 Konzipieren und Umsetzen des exemplifizierten Pflichtenhefts für Prüf- und Analysepläne	210
6.2.1 Regelbasis „Bürostandard“	213
6.2.2 Regelbasis „Planungsprojekt“	221
6.2.3 Regelbasis „Planungssituation“	240
7 Beispielhafte Anwendung und Evaluierung der erstellten Softwarelösung	251
7.1 Bedienung der Benutzeroberflächen.....	252
7.2 Anwendungsbeispiele aus weiteren Projekten	270
7.2.1 BBR Model Check	270
7.2.2 Verbundprojekt EnEff-BIM	273
7.2.3 Integriertes Semantisches Informationsmodell für den Kontext der energieeffizienten Stadt	278
8 Schlussbetrachtung.....	287
8.1 Zusammenfassung	287
8.2 Reflexion und Ausblick	294
Literaturverzeichnis.....	303

Abbildungsverzeichnis

Abbildung 1 Beeinflussbarkeit der Wirtschaftlichkeit (von Both, 2011 nach Müller 1999)	4
Abbildung 2 Flussdiagramm zum Aufbau der Arbeit	13
Abbildung 3 Übersicht Ergebnisse Gutachten Aufwandssteigerung in Planungsprozessen des Bauwesens (Motzko et al., 2012). 17	
Abbildung 4 BIM – Querschnittsdisziplin (Quelle: May, 2014; Folie 28)....	25
Abbildung 5 In welcher Struktur organisieren Sie Ihre Softwareumgebung? (Quelle: Herrmann & Westphal, 2017)	27
Abbildung 6 Konzeption der Nachhaltigkeit von Gebäuden (Quelle: BMUB, 2016 S. 17)	30
Abbildung 7 Schaubild Relation Modellbildung	35
Abbildung 8 XML-Sprachlandschaft (erweitert nach Vossen, 2008)	47
Abbildung 9 Geodatenmodellstandards (Quelle: Brüggemann u. a., 2020, Abb. 10.1)	49
Abbildung 10 Schematische Struktur ausgewählter IFC Schemateile	50
Abbildung 11 Informationsqualitäts-Dimensionen in 4 Kategorien (Hildebrand u. a., 2015, S. 30).....	62
Abbildung 12 Untersuchungsgegenstände der Informationsqualität (Hildebrand u. a., 2015, S. 30)	63
Abbildung 13 Erweiterte Datenqualitätspyramide (Hildebrand u. a., 2015, S. 92).....	67
Abbildung 14 Zusammenstellung der für die Arbeit wichtigen DQ-Dimensionen	72
Abbildung 15 Kernprozess der Entscheidungsunterstützung (Quelle: von Both & Ebertshäuser, 2018)	76
Abbildung 16 Einzelne Planung/Simulation von Fachaspekten.....	77

Abbildung 17 Lösungsraum der Arbeit zur Unterstützung integrierter Entscheidungsprozesse.....	78
Abbildung 18 Schematischer Regelaufbau zur Verdeckung von Komplexität mittels gestuft vorausgesetzten Anwendungswissen	92
Abbildung 19 Schema des Repository Entwurfsmusters ©SpringPeople ...	102
Abbildung 20 Zustimmungsverteilung 'Die formale und inhaltliche Qualität von digitalen Gebäudemodellen ist für eine sichere Vertragsgestaltung nicht ausreichend normiert.' (von Both, 2011, S. 159)	107
Abbildung 21 Zustimmungsverteilung 'Die verfügbaren Austauschformate für digitale Gebäudemodelle sind nicht durchgängig nutzbar' (von Both u. a., 2013, S. 150).....	108
Abbildung 22 Schematische Übersicht zur Gesamtkonzeption.....	111
Abbildung 23 Iterativer Prozessschritt Optimierung des Planungsgegenstandes.....	116
Abbildung 24 Schematische Darstellung der planerischen Entscheidungssituation.....	124
Abbildung 25 Modellabbildung massives Wandelement	125
Abbildung 26 Modellabbildung Wandelement in Holzrahmenbauweise	126
Abbildung 27 Prozess des Anwendungsfalls.....	128
Abbildung 28 Detailprozess zur Erstellung der Entscheidungsgrundlage...	131
Abbildung 29 Vorbereiten Modellprüfungsplan	145
Abbildung 30 Vorbereiten Modellanalyseplan.....	149
Abbildung 31 Interface der Basiskomponente DynamicExecution.....	163
Abbildung 32 DomainObjects Komponente	164
Abbildung 33 ErrorHandling Komponente	166
Abbildung 34 Klasse zum gestaffelten Sammeln von Ereignissen	167
Abbildung 35 Projekt Management Komponente	168
Abbildung 36 Modell-diagnostisch vorgehaltener Quelltext für Hilfsmethode	169

Abbildung 37 ModelInfo Klasse und implementiertes Interface zur Basiskomponente DynamicExecution.....	171
Abbildung 38 RuleManager Komponente sowie durch die RuleHandling Komponente implementierte Basiskomponente DynamicExecution.....	174
Abbildung 39 Verwendete bzw. (weiter-)entwickelte BLM ToolKIT Komponenten	175
Abbildung 40 Architektur- und Abhängigkeitsdiagramm der Desktop- Anwendung	177
Abbildung 41 Klassen der wesentlichen Ansichten der Benutzeroberfläche	179
Abbildung 42 Workflow Komponente - Realisierung der Business Logik .	182
Abbildung 43 Realisiertes MVC-Muster am Beispiel vom Berichtsmodul	183
Abbildung 44 Anbindung der Datenzugangsebene durch Manager Komponenten	185
Abbildung 45 Wesentliche Bestandteile der RuleRepository Komponente	187
Abbildung 46 An XML-Repräsentation gebundene (grau) und zur Laufzeit erweiterte Eigenschaften und Methoden der RuleRepository Komponenten Klassen	188
Abbildung 47 Zusätzliche Felder, Eigenschaften bzw. Methoden der um Bearbeitungsmöglichkeiten von Regeldaten erweiterten RuleRepository Klasse	190
Abbildung 48 Schematisches Regeldateikonzept (Ebertshäuser & von Both, 2013)	192
Abbildung 49 Strukturierung Nutzerdaten der Softwareanwendung.....	193
Abbildung 50 Skizze der Umsetzungsstrategie einer gestuften Regelentwicklung.....	203
Abbildung 51 Regel Container "Bürostandard (IFC)"	206
Abbildung 52 Container "Planungsprojekt": Gliederung und Regelbezeichnungen	207
Abbildung 53 Container "Planungssituation" (Struktur des Pflichtenhefts)	209

Abbildung 54 Logische Abfolge zum Filtern von Typobjekten.....	217
Abbildung 55 Logische Abfolge zum Filtern von Eigenschaftsätzen	218
Abbildung 56 Filtern der Eigenschaften eines Satzes	219
Abbildung 57 Logik einer Schablone für zusammengeführten Teilmethodenaufruf.....	220
Abbildung 58 Beispielhafte Entscheidungsgrundlage.....	249
Abbildung 59 Menüs und Informationsdialog im Hauptfenster des BLM Model Check	253
Abbildung 60 Modellübersichtswerkzeug - Informationen zum geladenen Modell	255
Abbildung 61 Regelauswahlwerkzeug	256
Abbildung 62 Regelausführungswerkzeug.....	257
Abbildung 63 Berichtswerkzeug	258
Abbildung 64 Administrationswerkzeug.....	260
Abbildung 65 Regelbearbeitungswerkzeug	262
Abbildung 66 Regelbearbeitungswerkzeug - Anlegen spezifischer Einträge im Modell Abfragen Pool einer Regel	267
Abbildung 67 Modellexplorerwerkzeug.....	269
Abbildung 68 Modellansicht des Berliner Humboldt Forum (Quelle: M. Reif, BBR)	270
Abbildung 69 BBR ModelCheck (überarbeitete, erweiterte Version 0.5 – 2016)	271
Abbildung 70 Schematischer EnEff-BIM Ansatz mit KIT Tools (ergänzte Grafik nach v. Both).....	274
Abbildung 71 Softwareprototyp EneffBIM ModelCheck	275
Abbildung 72 EneffBIM ModelCheck - Exemplifizierte TGA Regelbasis. 277	
Abbildung 73 Schematisches Inhaltskonzept des integrierten Stadtmodells (Quelle: Brüggemann & Both, 2014).....	279
Abbildung 74 Skizze des mehrskaligen Werkzeugansatzes zum integrierten Stadtmodell (Quelle: Vorhabensantrag, P. von Both)	280

Abbildung 75 Stadtmodellplattform ISISAgent – exemplifiziertes Modul des Prozessnavigators	282
Abbildung 76 ISISAgent – Projektmodul mit eingelesenem Stadtquartiersmodell	283
Abbildung 77 ISISAgent - Funktionsmodul für mehrskalige Analysen.....	284

Tabellenverzeichnis

Tabelle 1 Beispiel Schema-basierter bzw. „selbstbeschreibender,“ XML-basierter Datensatz	43
Tabelle 2 Vergleich zwischen OCL und C# LINQ Ausdrücken (Quelle: Akehurst u. a., 2008)	59
Tabelle 3 Exemplifizierter Ausdruck mit Schlüsselwörtern der Spezifizierungssprache.....	89
Tabelle 4 Zusammenhang Ausdrücke der Spezifikationsprache zur logischen Regelstruktur	90
Tabelle 5 Übersicht der gestuften Anforderungen bezüglich der Anwendungsebenen des Regelkonzepts.....	96
Tabelle 6 Soll-Referenzwerte	122
Tabelle 7 Beispiel IFC elementierte Wandinstanz	137
Tabelle 8 Beispiel Typobjekte der IFC Wandinstanz.....	138
Tabelle 9 Lastenheft Unternehmensspezifische Modellprüfung und -analyse	154
Tabelle 10 Lastenheft Projektspezifische Modellprüfung und -analyse.....	154
Tabelle 11 Lastenheft der situationsspezifischen Modellprüfung und -analyse	157
Tabelle 12 Strukturierungseinheiten der Anwenderdaten	194
Tabelle 13 Pflichtenheft Inhalte einer designierten Hilfsfunktion.....	214
Tabelle 14 Pflichtenheft Klassifizierung und Grundstruktur.....	223
Tabelle 15 Pflichtenheft Spezifische Modellierungsvorgaben	230
Tabelle 16 Pflichtenheft Energetische (Basis-) Werte.....	233
Tabelle 17 Pflichtenheft Energetische (Basis-) Werte (Fortsetzung)	235
Tabelle 18 Pflichtenheft Analysen zu energetischer Zielstellung.....	240
Tabelle 19 Pflichtenheft Modellierungsqualität Wandbauteil	241

Tabelle 20 Pflichtenheft Ermittlung Energetischer Kennwerte	245
Tabelle 21 Pflichtenheft Analyse Zielerreichung	248

Abkürzungsverzeichnis

Abb.	Abbildung
Abs.	Absatz
ADE	Application Domain Extensions
AEC	Architecture Engineering Construction
AIA	Auftraggeber-Informationsanforderungen
AMT	Allgemeine Modelltheorie
Anm. des A.	Anmerkung des Autors
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BAP	BIM-Projektentwicklungspläne
BBR	Bundesamt für Bauwesen und Raumordnung
BCF	BIM Collaboration Format
BIM	Building Information Modeling
BKI	Baukostenindex
BL	Business Logic
BLM	Building Lifecycle Management
BMU	Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit

BMWi	Bundesministerium für Wirtschaft und Energie
BNB	Bewertungssystem Nachhaltiges Bauen
BOM	Byte Order Mark
BPMN	Business Process Model and Notation
bspw.	beispielsweise
bzw.	beziehungsweise
CAD	Computer Aided Design
CMIS	Content Management Interoperable Standards
CMS	Content Management System
CRUD	Create, Read, Update, and Delete
CSV	Comma-separated values
d.h.	das heißt
DA	Data Access
DFG	Deutsche Forschungsgemeinschaft
DGNB	Deutsche Gesellschaft für Nachhaltiges Bauen
DGQI	Deutsche Gesellschaft für Informationsqualität
DIN	Deutsche Institut für Normung
DLL	Dynamic Link Library
DOM	Domain Object Model
DQ	Datenqualität

DQM	Qualitätsmanagement
dt.	deutsch
ebd.	ebenda
EDV	Elektronische Datenverarbeitung
EN	Europäische Normen
ENEV	Energieeinsparverordnung
engl.	englisch
erw.	erweitert
et al.	und andere
FM	Facility Management
FMI	Functional Mockup Interface
ggf.	gegebenenfalls
GIS	Geographic Information Systems
GML	Geographic Markup Language
GUID	Globally Unique Identifier
GÜ	Generalübernehmer
HOAI	Honorarordnung für Architekten und Ingenieure
HTML	Hypertext Markup Language
i. d. R.	in der Regel
IAI	Institut für angewandte Informatik

IBP	Fraunhofer Institut für Bauphysik
IBPSA	International Building Performance Association
IDM	Information Delivery Manual
IDX	Index
IEA	International Energy Agency
IEEE	Institute of Electrical and Electronics Engineers
IFC	Industry Foundation Classes
IFC-SPF	STEP Physical File
IP	Integrale Planung
IPM	Integrierten Produktmodells
IQ	Informationsqualität
IS	Informations-Systems
ISIS	Integriertes systemisches Informationsmodell Stadt
ISO	International Organization for Standardization
IT	Informationstechnik
JIT	Just in Time
Kap.	Kapitel
KfW	Kreditanstalt für Wiederaufbau
KIT	Karlsruher Instituts für Technologie
LCA	Life Cycle Assessment

LINQ	Language Integrated Query
LOD	Level of Detail
MIT	Massachusetts Institute of Technology
MVC	Model-View-Controller
MVD	Model View Definition
MVVM	Model View View-Model
NR	Namensraum
OCL	Object Constraint Language
OGC	Open Geospatial Consortium
OMG	Objekt Management Group
OOP	objektorientierten Programmierung
OWL	Ontology Web Language
QL	Query Language
QP	Jahres-Primärenergiebedarf
RDF	Resource Description Framework
REPL	Read, Eval, Print and Loop
RFC	Request for Comments
S.	Seite
s.o.	siehe oben
s.u.	siehe unten

SAX	Simple API to XML
SGML	Standard Generalized Markup Language
sic	wirklich so
sog.	sogenannt
SPC	statische Prozesskontrolle
SQL	Structured Query Language
STEP	ISO Standard for the Exchange of Product model
TDQM	Total Data Quality Management
TGA	Technische Gebäudeausrüstung
TQM	Total Quality Management
U-Wert	Wärmedurchgangskoeffizient
u.a.	unter anderem
UI	User Interface
UML	Unified Modeling Language
URI	Unified Ressource Identifier
usw.	und so weiter
UTC	Universal Time Coordinated
UTF-8	8-Bit UCS Transformation Format
VDI	Verein deutscher Ingenieure
vgl.	vergleiche

W3C	World Wide Web Consortium
WPF	Windows Präsentation Framework
WSDL	Web Services Description Language
XHTML	XML-basierte HTML
XML	eXtensible Markup Language
XSD	XML Schema Definition
XSI	XSD Schema Instance
z. T.	zum Teil
z.B.	zum Beispiel

Danksagung

An dieser Stelle möchte ich allen Menschen meinen großen Dank aussprechen, die mich bei der Anfertigung meiner Dissertation unterstützt haben.

Besonders danken möchte ich Frau Prof. Dr.-Ing. von Both für die ausgezeichnete Unterstützung und Betreuung bei der Umsetzung der gesamten Arbeit.

Außerdem möchte ich mich beim gesamten Team des Fachgebiets Building Lifecycle Management am KIT Karlsruhe bedanken, das mich bereits im Rahmen meines Studiums und über die Jahre während meiner Anstellung kollegial mit produktiven Gesprächen begleitet hat.

Für die Durchsicht der Arbeit sowie das Zweitgutachten danke ich Herr Prof. Dr.-Ing. Grün.

Meiner Familie und Freunden danke ich für ihre Geduld und Zusprüche während der Arbeit an dieser Dissertation.

Karlsruhe, im Januar 2024

Sebastian Ebertshäuser

1 Einleitung

Zunehmend komplexer werdende Herausforderungen konfrontieren die Planungsbeteiligten im Bereich der gebauten Umwelt. Im Denkansatz der integralen Planung werden daher Wege aufgezeigt, wie dieser inhärenten Komplexität der Planungsprobleme entgegengewirkt werden kann. Ein Übertragen der theoretischen Ansätze in die planerische Praxis setzt bei einer ganzheitlichen Sicht auf die spezifische Problemsituation an. Dabei verstanden als anforderungsorientierter Regelmechanismus, vermag eine integrale Planung durch die frühzeitige Prüfung und Evaluierung von Planungslösungen gemäß der auf dem Problemverständnis angesetzten Zielvorstellung einen qualitätssteigernden Einfluss auf den Planungsprozess zu leisten. Aufgrund einer besseren Abgrenzbarkeit spezifischer fachlicher Problemkontexte konzentriert sich eine Optimierung auf Ebene des Gesamtprozesses noch vornehmlich auf a posteriori eingesetzte Expertenwerkzeuge, die an einzelnen (energetischen, bauphysikalischen, statischen, usw.) Fachaspekten bereits fortgeschrittener Planungslösungen mit Möglichkeiten einer Simulation und Berechnung ansetzen. Da formelhafte Ableitungen integraler Planungsansätze jedoch zu einer parallelen Bearbeitung - letztendlich zu einer Entkoppelung vom eigentlichen Planungsablauf - führen können, sollte ein besonderes Augenmerk auf den ganzheitlichen fachübergreifenden Herangehensweisen liegen. So erscheint neben diesen, auf eine fachlich spezialisierte Optimierung zielenden Hilfsmitteln, eine fachübergreifende, ganzheitliche Bewertung und Prüfung auf Ebene des Gesamtprozesses sinnvoll. In Bezug zur aktuellen Diskussion, insbesondere die, der Bauwerksplanung übergeordneten, lang andauernden, vielschichtigen kommunalen Planungs- und Ausführungsprozessen betreffend, steht auch das Thema Analyse und Qualitätssicherung kooperativ bearbeiteter Information, das im Rahmen von sich zunehmend etablierenden, offenen modellbasierten Austauschformaten auf Gebäude- und Stadtebene (vgl. Building Information Modeling – BIM, Geographic Information System – GIS) an Bedeutung gewonnen hat.

1.1 Problemstellung und Motivation

Der in diesem Kapitel beschriebene Problemkontext verortet sich im Feld des Bauwesens in einer zunehmenden Komplexität bei der Bewältigung von Planungsaufgaben (vgl. Abs. Entwicklung im Bauwesen). Diese äußert sich u. a. in einer, durch stetig steigende Anzahl zu berücksichtigender Fachaspekte, verschärfenden Abstimmungsproblematik zwischen einer Vielzahl beteiligter Disziplinen. Von Seiten der Planungsmethodik wird versucht, mit dem fundierten Denkansatz der *integralen Planung* (IP) diesen komplexen Planungssituationen in ihrer Entstehung entgegenzuwirken (vgl. Abs. Paradigma der Integralen Planung sowie Potentiale integralen Planung). Auf den dabei aufgezeigten Potentialen zur Unterstützung von Entscheidungssituationen gründet sich die Motivation für die Arbeit. Somit fokussiert sie diese theoretisch fundierten Hilfestellungen der integralen Planung und beschäftigt sich insbesondere mit den darauf aufbauenden modellbasierten Arbeitsweisen im Rahmen der Digitalisierung. Dabei stellt sich die Frage, ob das volle Potential der gemäß dem theoretischen Denkansatz formulierten Hilfsmittel schon vollständig entfaltet ist. Beispielsweise eine integrierte Datenbasis betreffend, die als durchgängige Informationsgrundlage den Ablauf von Planungsprozessen unterstützt (vgl. Abs. Integrale Planung in der Forschungsförderung). Der letzte Abschnitt des Kapitels konkretisiert die Motivation zur vorliegenden Arbeit, sich diesen Fragestellungen zu widmen und dabei beim Schöpfen von Potentialen integraler Planungshilfsmittel, an deren gewichtigen planungsmethodischen Anforderung bezüglich einer ganzheitlichen, integrierten Datenbasis anzusetzen.

Entwicklung im Bauwesen

Der Bereich des Bauwesens – von der Gegenstandsebene des einzelnen Bauwerks, seiner Bestandteile und technischen Systeme bis zur übergeordneten Ebene urbaner Strukturen und Systeme – ist von immanenter Komplexität geprägt. Das Wirken in diesem Bereich lässt sich unabhängig von der Skaligkeit des Gegenstands, generell drei (phasen-orientierten) Prozessbereichen zuordnen – Gestaltungs-, Umsetzungs- und Verwaltungsprozesse. Am Beispiel eines typischen Lebenszyklus des Bauwerks, von der initialen Idee, der Planung,

über den Betrieb bis zum Abbruch/Umbau, teilen sich so Planungs-, Ausführungs- und Betriebsprozesse in diese drei Prozessbereiche auf. Dabei bilden die Prozesse einer jeweiligen Phase eine bedeutsame Perspektive auf den Gegenstand in seinem komplexen Problemkontext. Für die Prozesse der Folgephase ist diese Sicht entscheidend, da sie die Basis der Entscheidungen der vorhergehenden Phase reflektiert, und somit die Annahmen transparent machen, die zum gegenwärtigen Gegenstand geführt haben. Je vollständiger beispielsweise ein Planungsgegenstand gemäß den Anforderungen der Planungssituation gefasst ist, desto optimaler ist dieser als Ausgangspunkt der Ausführung aufgestellt (d.h. alle entscheidenden Aspekte sind vollständig in der Entwurfsplanung berücksichtigt und ergeben somit für die Ausführungsplanung eine lückenlose Dokumentation des Lösungsraums).

In der Gestaltungsphase der Gebäudeplanung liegt die dabei zu bewältigende Komplexität (vgl. für diesen Abschnitt von Both, 2011) beispielsweise in einer hohen Anzahl unterschiedlich beteiligter Akteure, zumeist unscharf und dynamisch gehandhabter Zieldefinitionen und dem Planungsgegenstand selbst, der als Unikatentwicklung ein Produkt mit hoher Lebensdauer zum Ziel hat.

Dieser initialen Phase des Gesamtprozesses kommt eine entscheidende Weichenfunktion für die weiteren Folgeprozesse zu. Gezeigt wird dies auch in Abbildung 1, wonach sich der Lösungsraum nach den initialen Entscheidungen sukzessive fokussiert. Oder anders ausgedrückt – im Frühstadium getroffene (Fehl-) Entscheidungen werden im Prozessverlauf für die Anzahl möglicher Lösungswege immer bedeutender (d.h. einschränkender). Zu diesen „vertikalen“ Interdependenzen kommen starke „horizontale“ (inhaltliche) Verflechtungen der verschiedenen Planungsbereiche hinzu (ebd.).

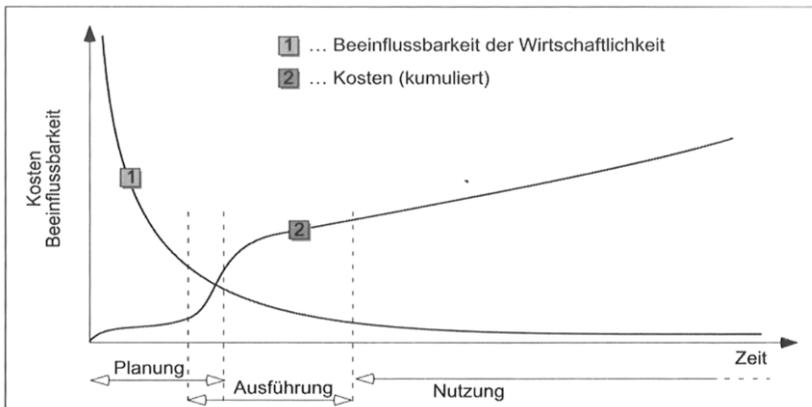


Abbildung 1 Beeinflussbarkeit der Wirtschaftlichkeit (von Both, 2011 nach Müller 1999)

Dabei haben sich in den letzten Jahrzehnten in den einzelnen Planungsbereichen die Zahl einzubeziehender Aspekte vervielfacht, beispielsweise durch steigende bau- bzw. bauordnungsrechtliche Anforderungen. Dies wiederum resultiert in steigenden Anforderungen an die Planungs- und Koordinationsleistungen der Prozessbeteiligten (ebd.).

Paradigma der Integralen Planung

Um in diesem durch starke Wechselwirkungen und Verflechtungen geprägten Bereich des Bauwesens komplexe Planungsaufgaben zielführend zu lösen, bedarf es der Anwendung ganzheitlicher integraler Methoden (Lützkendorf u. a., 2009) Das Paradigma der Integralen Planung bietet hierfür eine vielversprechende Grundlage mit einem idealtypischen Ansatz für die Lösung komplexer Probleme.

Da sich die Komplexität oftmals in sich widersprechenden Zielen und Anforderungen niederschlägt, bildet das „Durchdenken“ des gesamten Projektes von Beginn an sowie in allen Details bis zum Ende im Sinne der Integralen Planung einen fundierten Lösungsansatz. Insbesondere durch ihre ganzheitliche Perspektive erscheint sie hervorragend geeignet, kooperativ planende Akteure

beim Entwickeln wohlgeformter Gesamtleistungen zu unterstützen. Wobei das finale Produkt „Bauwerk“ dann mehr ist als die Summe der einzelnen Planungs- und Ausführungsleistungen.

Obwohl schon mehrere Jahrzehnte in Wissenschaft und Praxis verwendet, unterliegen dem Begriff der Integralen Planung verschiedene Bedeutungen. Oftmals falsch verwendet – integriert nur als „inbegriffen“ verstanden und auf Teilausschnitte fokussiert – wird mittels dieser isolierten Sichtweisen das Lösungsmoment des Ansatzes bezüglich der Komplexitätsbewältigung unterminiert (denn: isolierte Sichten auf bzw. „Vereinfachen“ von Komplexität zerstört diese! Vgl. Dörner, 2003). Auf wissenschaftlicher Seite versuchen neuere Ansätze (vgl. Rexroth & von Both, 2016) den bisher kaum geklärten Begriff grundlegend zu definieren und damit die zahlreichen Begriffsbedeutungen im Sinne eines einheitlichen, kohärenten Begriffsfeldes zu harmonisieren.

In dieser Arbeit wird eine Auslegung des Begriffes im Sinne der Ganzheitlichkeit aufgegriffen – integriert meint auch „übergreifend“. So definiert N. Kohler schon Ende der 1980er IP als **ganzheitliche Planungsmethodik**, der ein systemischer Denkansatz unterliegt (vgl. Suter u. a., 1986). Hieraus leitet sich auch ein mehrdimensionaler Integrationsraum der Integralen Planung ab: Zum einen die inhaltliche Vernetzung über die verschiedenen Fachdomänenaspekte hinweg – eine horizontale Integration. Und zum anderen den Lebenszyklusansatz implizierend, eine vertikale Integration, d.h. Integration phasenübergreifender (bzw. Prozess-durchgängiger) Aspekte.

Potentiale integraler Planung

Aus dieser ganzheitlichen Auslegung der Integralen Planung erwachsen große Potentiale. In der Praxis zeichnet sich beispielsweise ab, dass durch einen Mehraufwand von 2% in der (ausführlicheren) Planungsphase bis zu 40% Kosteneinsparung im Betrieb des Bauwerkes möglich sind (Kovacic u. a., 2012).

Auf Seiten der Wissenschaft ermöglicht die breite Spannweite des Ansatzes („Betrachtungshorizonts“), von der strategischen bis zur operativen Ebene, be-

züglich eines Gesamtvorhabens konsistent (beschreibbare) Anforderungskontexte abzuleiten, auf denen ganzheitliche Entscheidungshilfen aufbauen können.

Hierdurch kann beispielsweise ein Ausblenden von Reintegrationsproblemen der (fachspezifisch) erbrachten Erkenntnisse in die „Wissenswelt“ einer Entscheider*innenebene adressiert werden, was bei vielen, nur einen Teilausschnitt des Gesamtprozesses betrachtenden Ansätzen festgestellt werden kann. Im Gegensatz dazu beinhaltet die holistische Perspektive der Integralen Planung sowohl die „anfragenden“ Entscheider*innen, wie auch die prognostizierenden, simulierenden, berechnenden Fachdisziplinen mit ihren Informationsbedarfen, die sich dann konsistent in entsprechend integrale Hilfsmittelansätze einbeziehen lassen.

Durch diesen „inkluisiven“ Betrachtungshorizont wird auch eine Grundlage für eine wesentliche informationslogistische Funktion – aktuelle und richtige Information zur richtigen Zeit am richtigen Ort bzw. bei der richtigen Person/Rolle – geschaffen. Darüber hinaus können auf dieser Grundlage, durch ein Transparentmachen des Bezugs der einzelnen Information zum Gesamtkontext, auch Einstiegshürden beim Ergreifen dieser methodischen Hilfsmittel für darauf aufbauende Entscheidungshilfen gesenkt werden.

Zudem erlaubt ein konsistent den gesamten Verlauf adressierender Ansatz die erforderliche planungsbegleitende Evaluierung, beispielsweise zur laufenden Revision des Grades an Zielerreichung bzw. Abweichung. Auf dieser Grundlage können dann weitere Generalisierungen von Systematiken ansetzen, wie beispielsweise ein „iterativer Prozess der Schritte Problemanalyse, Systemsynthese und Systemanalyse“ (von Both, 2011) als (generischer) Kern einer Revisionsmethodik. Worauf wiederum eine weiterführende (kohärente) Unterstützung mit technischen Hilfsmitteln sinngemäß angesetzt werden kann.

Durch diese methodisch konzeptionellen Ansatzpunkte zur Integration und Schaffung von Kompatibilität wird mittels Theoriebildung im Feld der Integralen Planung ein wichtiger Faktor für die verteilte Zusammenarbeit, sowie die fach- und applikationsübergreifende Interaktion weiter vorangetrieben. Jedoch

bedürfen die zumeist nur theoretisch fundierten integrierten Entscheidungshilfen einer Unterstützung von Seiten technischer Werkzeuge, die bei der Umsetzung der Methodik in eine schwer generalisierbare bzw. formalisierbare planerische Praxis Hilfestellung leisten.

Umgekehrt, erscheint für eine solche (bisher heterogen gewachsene, technische) Werkzeugebene eine ganzheitliche theoretisch fundierte Grundlage als „zukunftssichere“ (lösungsneutrale Basis-) Leitlinie/Metrik von Bedeutung zu sein. Gerade auch als eine gemeinsame „Bezugssystematik“ zur Harmonisierung von Umsetzung derzeit laufender disziplin- und skalenübergreifender Digitalisierungsprozesse (in verschiedenen Geschwindigkeiten) im gesamten Bauwesen. So kann in dieser Verknüpfung der Integralen Planung und deren Unterstützung durch moderne Informations- bzw. Kommunikationstechnologie ein nachhaltiger Mehrwert geschaffen werden.

Integrale Planung in der Forschungsförderung

Die Erkenntnis diesbezüglicher Potentiale zeichnet sich auch in den ausgeschriebenen Förderungen in diesen Themenbereichen ab. So ist nach langjähriger Förderung themenspezifischer Förderinitiativen im Bereich Energie und gebauter Umwelt auf Seiten des Bunds eine erkennbare Zuwendung zu integralen Themen erkennbar. In der aktuellen Strategie des Bundesministerium für Wirtschaft und Energie (BMWi) zeigt sich dies beispielsweise auch in einer engeren Vernetzung der Initiativen EnEff:Stadt, EnEff:Wärme und EnOB.

Darüber hinaus werden konkrete Werkzeuge zur Unterstützung integraler Ansätze gefördert (vgl. BMWi, 2015). Mit verstärkter Förderung im Themenbereich der Ko-Simulation wird so beispielsweise eine stärkere Durchgängigkeit bei den bisher eher isoliert entwickelten (und geförderten) Simulationsansätzen auf Ebene der Business-Logik vorangetrieben.

Konkret werden hierbei innovative Fazilitäten zur Schaffung herstellerneutraler Kopplungsebenen für die eigenständigen Simulationsansätze untersucht. Neben dabei geförderten Ansätzen zum grundlegend neuen Aufbau von Schnittstellen, ist das Aufgreifen vielversprechender Standardisierungsan-

sätze, wie beispielsweise das *Functional Mockup Interface* (FMI¹) in den neueren Untersuchungen in diesem Bereich erkennbar (vgl. Wetter & Treeck, 2017). Im Prinzip werden hierbei, zum Schutz des geistigen Eigentums (engl. intellectual property) proprietärer Simulationsumgebungen, jeweils eine „Blackbox“ mit der ausführbar gekapselten Business-Logik des Simulationsmodells über standardisierte Ein- und Ausgangsschnittstellen in einem übergeordneten Vorgang (Ko-Simulation) verknüpft.

Der dabei implizierte Integrationsansatz auf Ebene der Methoden bzw. Business-Logik (vgl. run-time coupling) scheint jedoch zu kurz zu greifen. Zwar können Teilaspekte, für die bereits Simulationslösungen vorliegen, damit verknüpft. Existieren jedoch Interrelationen zwischen den einzelnen (Teil-) Simulationen, steigt indessen häufig die Komplexität der Gesamtsimulation und erschwert ein Validieren. Hinzu kommt, dass die gekoppelten dynamischen Modelle (vgl. FMI for Co-Simulation) oftmals nicht von einer einheitlichen Datenmodellebene mit den Informationen der Planung parametrisiert, sondern zusätzlich jeweils eigene Parameterdatensätze (vgl. FMI for Model Exchange) benötigen. Eine durch starke disziplinübergreifende Zusammenhänge geprägte Planungsinformation als Datengrundlage dieses Integrationsansatzes der Simulation auf Ebene der Business Logic einzelner Werkzeuge wirft freilich die Frage auf, wie damit in Bezug auf eine Skalierbarkeit durch die sich weiterentwickelnden Anforderungen an die Planungsgegenstände sowie durch hinzukommende (fachübergreifende) Aspekte, zielführend einer steigenden Komplexität von Planungsaufgaben begegnet werden kann.

Ansatzpunkt der Arbeit

Um die Gesamtheit aller möglichen Fachaspekte, auch im Sinne möglicher zukünftiger (Weiter-)Entwicklungen, anforderungsgerecht einbeziehen zu können, erscheint ein Integrationsansatz auf Ebene der Daten am sinnvollsten zu sein. Gerade eine große Vielfalt an (z. T. affinen) Aspekten, wie sie beispielsweise bei planerischen Fragestellungen im Bereich Energie und Nachhaltigkeit

¹ FMI wird von der Modelica Association entwickelt und ist simulationswerkzeug-unabhängiger Standard einer (Simulations-) Modellschnittstelle; vgl. <https://fmi-standard.org/> bzw. <https://www.modelica.org/mastandards>

zu berücksichtigen sind, scheint ohne Harmonisierung und Integration auf Datenebene kaum mehr „kohärent“ handhabbar zu sein.

Hierzu wird auch von planungsmethodischer Seite bezüglich der Handhabung heterogener, weitgestreuter Gesichtspunkte als Basis der Entscheidungsfindung ein integriertes Produkt- bzw. Systemmodell gefordert (von Both, 2006). Als fundamentaler Baustein der Integralen Planung ermöglicht erst eine integrierte, alle Zusammenhänge berücksichtigende sowie durchgängig im Prozess verfügbare Datenbasis das zielführende Ansetzen integraler Methoden und Konzepte zur Optimierung der Planungsgegenstände (vgl. auch von Both & Ebertshäuser, 2018).

Die Anforderung, eine datenbasierte Integrationsebene als Hilfsmittel der Integralen Planung zu schaffen, zielt dabei auf das (Vor-)Bereiten geeigneter allgemein gültiger Grundlagen ab. Schon vor den ersten Schritten zur Konkretisierung der Problemsituation, soll für die sich erst später abzeichnenden (prozessspezifischen) Fragestellungen auf „generischen“ Ansätzen zur Prozessunterstützung aufgebaut werden. Mit dem Umsetzen dieser integrierten Hilfsmittel werden somit, schon vor ihrem Entstehen, im Allgemeinen erkannte übliche Problemfelder adressiert. Der Mehrwert einer ganzheitlichen, kohärenten und transparenten Datenbasis ist neben der durchgängigen Unterstützung einer effektiven Handhabung und Haltung der Daten im Prozess, auch das Eliminieren von Medienbrüchen, die sich beim Austausch zwischen verschiedenen Datenbasen oftmals als technisches Problem zeigen.

Auf diesen im Ansatz einer Integralen Planung theoretisch fundierte Anforderungen, wie der Bedarf einer einheitlichen Datenbasis können konkrete Zielstellungen für die vorliegende Arbeit aufgebaut werden. Dabei nimmt die im Kapitel 1.2 zusammengestellte Zielsetzung auch Bezug auf sich bereits erkennbar etablierende methodische Grundlagen in der Praxis, wie beispielsweise einer BIM Methode zur Bewerkstellung eines absprachelosen Datenaustausches. Allerdings zeichnet sich ein einheitlicher Umgang zur Anwendung der Methodik im Sinne einer vereinheitlichten Datengrundlage für den Planungsprozess erst ab, z. B. in der derzeit laufenden Normungsarbeit zu BIM. Daher werden entsprechende Entwicklungen zwar mit bedacht, die Ziel-

stellung selbst aber baut auf einer idealtypisch etablierten modellbasierten Arbeitsweise auf, die stellvertretend für eine durch einen starken Wandel geprägte, (noch) schwer formalisierbare Praxis steht. Dabei steht insbesondere das Aufgreifen offener, herstellernerutraler Produktdatenstandards im Zentrum der idealisierten Arbeitsweise, auf der die Zielstellung aufbaut.

Einordnung in den Institutskontext

Der Dissertation liegt eine langjährige Forschungstätigkeit des Autors am Fachgebiet Building Lifecycle Management (BLM) des Karlsruher Instituts für Technologie (KIT) zugrunde. Erkenntnisse und Vorarbeiten aus verschiedenen Forschungsprojekten am BLM bilden somit eine der wesentlichen Grundlagen der vorliegenden Arbeit. Thematisch stellte das Handlungsfeld der Energieeffizienten Planung vor dem Hintergrund klimapolitischer Zielsetzungen auf verschiedenen Skalenebenen im Bereich der gebauten Umwelt einen wichtigen Fokus der Tätigkeiten dar. Auf Seiten der Methoden standen dabei insbesondere auch ein Übertragen von planungsmethodischen Konzeptionen in IT-gestützte Hilfsmittel im Vordergrund einer disziplinübergreifenden Forschungsarbeit.

1.2 Zielsetzung

Diese Dissertation ist im Kontext der Entwicklung und Bereitstellung integrierter Planungshilfsmittel für die Ebene der Planenden und Entscheider*innen verortet. Sie beschäftigt sich mit der Fragestellung, wie Ergebnisse, welche aus Simulation und Berechnung in den Planungsprozess zurückgeführt wurden, fach- und aspektübergreifenden Analyse- und Auswertungsprozessen zugänglich gemacht werden können. Einen wesentlichen Rahmen, auf den die Zielstellung der vorliegenden Arbeit dabei aufbaut, bilden die von den theoretisch fundierten Methoden und Hilfsmitteln der integralen Planung abgeleiteten Anforderungen. Anstelle der Unterstützung der Planenden allein in einer spezifischen Planungssituation, wird ausgehend von wiederkehrenden Entscheidungssituationen im Planungsprozess das Ziel verfolgt, Methoden und Hilfsmittel für eine auf Basis integrierter Produktmodelle generalisierbaren

Entscheidungsunterstützung aufzuzeigen. Generell zielt dies dabei auf die ganzheitliche Optimierung des Planungsprozesses ab. Im Speziellen sollen mit der Suche nach Möglichkeiten der Qualitätsverbesserung der Grundlagen von Entscheidungssituationen dynamisch anpassbare Werkzeugansätze erarbeitet werden.

Ausgehend von den auf Seiten der integralen Planung geforderten Hilfsmittel, wie beispielsweise einer durchgängigen digitalen Beschreibung der Planungslösungen mittels virtueller Modelle (von Both, 2011, S. 235) sowie von Methoden wie beispielsweise eines anforderungsorientierten Regelmechanismus zur Entscheidungsunterstützung (siehe Abbildung 15; vgl. von Both & Ebertshäuser, 2018) sollen generische Ansätze zur Unterstützung des modellbasierten Arbeitens untersucht werden. Dabei liegt der Ansatzpunkt der Zielsetzung in einer derzeit durch einen starken Wandel geprägten planerischen Praxis. Es erscheint daher sinnvoll, mit wesentlichen Rahmenbedingungen eines zukünftigen modellbasierten Arbeitens dazu eine klare Stellung zu beziehen. Eine (noch) schwer zu formalisierende Praxis kann damit abgegrenzt werden, um sich voll auf die verfolgten Ziele im Rahmen der Arbeit konzentrieren zu können.

Ohne dabei die sich beispielsweise auf der Bauwerksebene unter dem Thema BIM (vgl. Kap. 2.1.4) abzeichnenden Vorgehensweisen eines modellbasierten Arbeitens vorzugeben, sollen generalisierbare Konzepte und Hilfsmittel entwickelt werden, die die Entscheider*innenebene beim Bewältigen ihrer skalen- sowie fachübergreifenden Fragestellungen durch inhaltlich-logische Modelnanalyse und -prüfungen unterstützen. Im Fokus der diesbezüglich verfolgten Entwicklung von IT-gestützten Hilfsmittel stehen daher etablierte offene (herstellerneutrale) Produktmodellstandards zur Abbildung der Planungsgegenstände, wie z.B. die Industry Foundation Classes (IFC). Entgegen der derzeitigen Unterstützung dieser Modellstandards in ihren neuesten Versionen durch entsprechende Autorensysteme zur Erstellung und Handhabung der Planungsinformationen, wird im Rahmen dieser Arbeit davon ausgegangen, dass den Planenden bei der Erstellung der modellbasierten Planungsgegenstände sämtliche Beschreibungsmöglichkeiten der Standards zur Verfügung stehen. Erst vor diesem Hintergrund kann im Rahmen einer übergeordneten Zielstellung

zur kollaborativen Lösung komplexer planerischer Problemstellungen ein konkretes Ziel für eine Entscheidungshilfestellung formuliert werden: Zur Unterstützung der Entscheider*innen in planerischen Fragestellungen soll ein regelbasiertes Planungswerkzeug entwickelt und anhand eines konkretisierten Beispielszenarios evaluiert werden. Dabei sollen die Planenden, die durch ihr Wissen zur konkreten Entscheidungslage die höchste Problemlösekompetenz besitzen und denen in der Praxis die Kontrollpflicht zur Planungslösung obliegt, selbständig die Prüfungen und Analyse der modellbasierten Planungsgegenstände gestalten. Dazu soll als begleitende Zielstellung ein adäquater Umgang mit einer erforderlichen technischen Vorkenntnis zur Bedienung der IT-gestützten Entscheidungshilfe untersucht und entwickelt werden. Idealerweise soll die Hilfestellung in einer Entscheidungssituation mit möglichst geringem technischen Vorwissen anwendbar sein. Mit einer geeigneten Abstufung – von einem einfach zu bedienenden Werkzeug hin zu einem Expertenwerkzeug – soll beim Einsatz der avisierten Hilfestellung eine rollenbasierte Arbeitsteilung ermöglicht werden; eine benötigte Vorbereitung sowie Anpassung des Hilfsmittels an die Entscheidungssituation auf der einen Seite und die eigentliche Anwendung des Hilfsmittels als Grundlage der Entscheidungsunterstützung auf der anderen Seite.

1.3 Gliederung und Aufbau der Arbeit

Um die Inhalte des vorliegenden Manuskriptes in ihrem Gesamtzusammenhang besser einordnen zu können, werden in diesem Kapitel deren Struktur zusammen mit den arbeitsmethodischen Überlegungen bei der Entstehung vorgestellt. Leser*innen, die nur an bestimmten Inhalten interessiert sind, soll durch den Überblick der wesentlichsten inhaltlichen Zusammenhänge dabei auch ein Querlesen ermöglicht werden.

Die im weitgespannten Feld der Digitalisierung des Bauwesens ansetzende Arbeit erstreckt sich, „verkürzt“ dargestellt, von den methodischen Bereichen in die umsetzungsorientierten Bereiche. Diese Spannweite adressiert dabei sowohl theoretische und praktischen Disziplinen – Planungsmethodik sowie Planung und Entscheidung auf der einen Seite und Informationstechnologie als

auch Entwicklung auf der anderen Seite. Abbildung 2 stellt eine schematische Übersicht der in der Arbeit behandelten Themenbereiche in zusammenhängender Form als Flussdiagramm dar. Die wesentlichen Arbeitsteile sind dabei jeweils thematisch zusammengefasst.

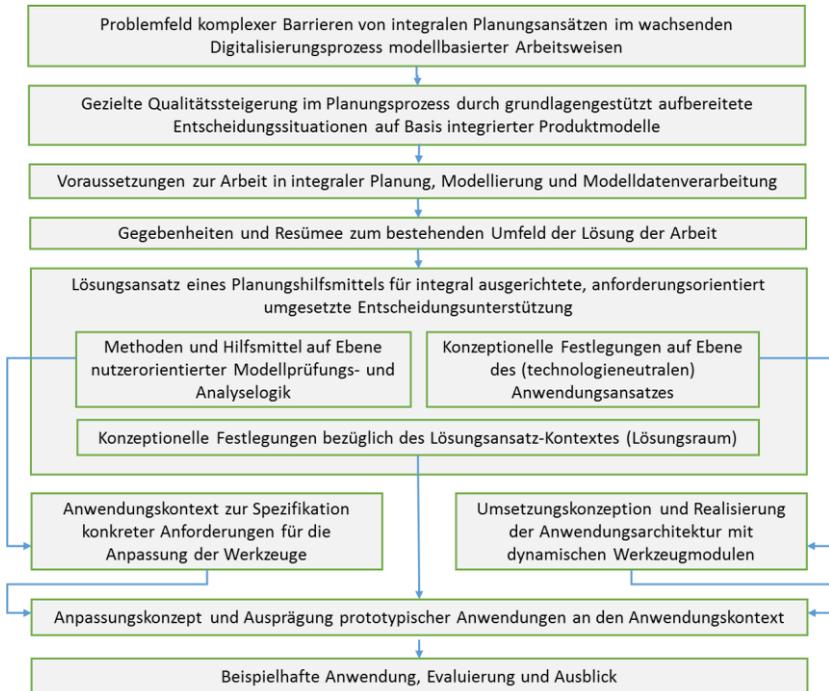


Abbildung 2 Flussdiagramm zum Aufbau der Arbeit

2 Grundlagen

Um den thematischen Kontext der beabsichtigten Lösungssuche (Kap. 1.2) sowie den daraufhin beschriebenen Lösungsweg (Kap. 3) in den wissenschaftlichen bzw. praktischen Themenfeldern einordnen zu können, bedarf es der Aufbereitung der einbezogenen Grundlagen in diesem Kapitel, zum Beispiel einer stringenten Klärung existierender, unterschiedlichen Deutungen bezüglich *integrierter Planungshilfsmittel* für den Kontext der Arbeit. So setzt ein Grundlagenkapitel mit der zum Einordnen der Zielstellung wichtigen Begriffsklärung zum (theoretischen) Thema integrale Planung an (Kap. 2.1). Neben einer Zusammenfassung der methodischen Aspekte des Planungsparadigmas zur Verortung eines Ansatzpunktes der Arbeit werden Hilfsmittel zusammengestellt, die im planungsmethodischen Kontext stehen (vgl. Kap. 2.1.5). Die beiden darauffolgenden Kapitel spannen übergeordnet einen Bogen zwischen den zugrundeliegenden Kontexten der thematischen Grundlagen der Arbeit – von der Theorie in die Praxis. Damit soll der, das proklamierte Ziel der Arbeit abgestuft verfolgende, thematisch weitgespannte Lösungsweg auf eine fundierte (Wissens-) Basis gestellt werden, die alle für das Nachvollziehen der Herangehensweise relevanten Themenbereiche abdeckt. So werden zunächst weitere theorieorientierte Themen angeführt, die dem Rahmen der Arbeit von Seiten der (Daten-) Modellierung zugrunde liegen (Kap. 2.2). Auf der Seite der Modelldatenverarbeitung (Kap. 2.3) werden dann für die Arbeit bedeutsame (informations-) technologische Themen aufbereitet. In diesem Kapitel geht es generell darum, ein der Arbeit zugrunde gelegtes Verständnis zu den dargestellten Aspekten zu vermitteln.

2.1 Integrale Planung

2.1.1 Aktualität und Hintergrund

Vom Planungsproblem als Symptom wesentlicherer Planungsprobleme?

Die heutigen Herausforderungen an die gesellschaftlichen Entwicklungen – Klimawandel, Umweltverschmutzung, Artensterben, Digitalisierung, Globalisierung, demographischer Wandel u. a. – spiegelt sich, in komplementärer Form, in einer kontinuierlich wachsenden Komplexität bei der Bewältigung der Aufgaben im Bereich der gebauten Umwelt. Dies zeichnet sich beispielsweise in Aufwandsveränderungen ab, den Motzko et al. (2012) über zwei Dekaden qualitativ für untersuchte Planungsprozesse in ihrem Gutachten herausarbeiten. Thematisch wurden dabei Ingenieurbau, Flächenplanung, Infrastruktur- sowie Hochbauprojekte betrachtet und aus Experteninterviews bezüglich folgender Kriterien die gesteigerten Aufwände evaluiert:

- Technischer Baubestimmungen
- Recht und Rechtsprechung
- Kommunikation
- Koordination
- Psychische Belastung
- Wirtschaftlichkeit
- Nachhaltigkeit
- Umwelt
- Qualitätsmanagement
- soziokulturelle Belange
- elektronische Medien, Software und Computer
- Organisation und Verwaltung

Abbildung 3 gibt in Form einer Matrix die untersuchten Aufwandssteigerungen entlang der thematischen Untersuchungsitems wieder. Für den Kontext dieser Arbeit sind die Bereiche a) Qualitätsmanagement sowie b) elektronische Medien, Software und Computer dabei von besonderem Interesse. So führte eine gestiegene Bedeutung der Qualitätssicherung auf Seiten des Bauherrn bezüglich der Baudokumentation zur Anforderung „... den Planungsprozess auch für Dritte transparent zu gestalten“. Am meisten führte dies im Beobachtungszeitraum zu Mehraufwänden an der Nahtstelle der Bauvorhaben zur Umwelt. Demnach verzeichnen Bodenmechanik, Erd- und Grundbau bzw. Vermessung sowie Technische Gebäudeausrüstung die größten Zunahmen bei den Aufwänden (Motzko u. a., 2012).

Objekt	Hochbau	Ingenieur- bau	Verkehrs- anlage	Flächen- planung	Ver- messung	TWP Hochbau	TWP Ing.Bau	Bau physik	Boden mechanik	TGA	Frei anlagen
Projekt - beteiligte	↗ Sehr Hoch	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz
Komplexität Planung	↗ Sehr Hoch	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Gering	↗ Tendenz	↗ Tendenz
Technische Regelwerke	↗ Mittel	↗ Sehr Hoch	↗ Gering	↗ Mittel	↔ Neutral	↗ Mittel	↗ n.B.	↗ Sehr Hoch	↔ Neutral	↗ Sehr Hoch	↗ Hoch
Rechtl. Belange	↗ Hoch	↗ Sehr Hoch	↗ Gering	↗ Sehr Hoch	↗ Hoch	↗ Gering	↗ Mittel	↗ In Hoch. Reg. Sehr Hoch	↗ Gering	↗ Mittel	↗ Hoch
Umweltrechtl. Belange	↗ Hoch	↗ Gering	↗ Gering	↗ Mittel	↔ Neutral	↗ Gering	↗ Gering	↗ In Hoch. Reg. Sehr Hoch	↔ Neutral	↗ In Hoch. Reg. Sehr Hoch	↗ In Hoch. Reg. Hoch
Ökonomie	↗ Hoch	↗ Mittel	↗ Gering	↗ Mittel	↔ Neutral	↗ Gering	↗ Gering	↗ In Hoch. Reg. Sehr Hoch	↔ Neutral	↗ Gering	↗ Gering
Soziokulturelle Belange	↗ Mittel	↔ Neutral	↗ Gering	↗ Sehr Hoch	↔ Neutral	↔ n.A.	↔ Neutral	↗ In Hoch. Reg. Sehr Hoch	↔ Neutral	↔ Neutral	↗ In Hoch. Reg. Gering
Kommunik. & Koordination	↗ Mittel	↗ Mittel	↗ Mittel	↗ Hoch - Sehr Hoch	↗ Sehr Hoch	↗ Gering	↗ Mittel	↗ Mittel	↗ Mittel	↗ Mittel	↗ Mittel
Qualitäts- management	↗ Mittel	↗ Gering	↗ Deutlich Mittel	↗ Gering	↗ Mittel	↔ Neutral	↗ Mittel	↔ Neutral	↗ Hoch	↗ Mittel	↗ Gering
Nachhaltigkeit	↗ Mittel	↔ Neutral	↗ Gering	↗ Mittel	↔ Neutral	↗ Gering	↔ Neutral	↗ In Hoch. Reg. Sehr Hoch	↔ Neutral	↗ Gering	↗ In Hoch. Reg. Gering
EDV/IT/ Software	↗ Mittel	↗ Gering	↗ Tendenz	↗ Sehr Hoch	↔ Neutral	↗ Mittel	↗ Gering	↔ Neutral	↗ Gering	↗ Gering	↔ Neutral
Mitwirken Auftraggeber	↗ Hoch	↗ projektspezifisch	↗ Mittel	↗ Gering	↗ Tendenz	↗ Tendenz	↔ Neutral	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Sehr Hoch
Sonstiges	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↗ Tendenz	↔ n.A.	↗ Tendenz	↗ Tendenz	↗ Tendenz

Abbildung 3 Übersicht Ergebnisse Gutachten Aufwandssteigerung in Planungsprozessen des Bauwesens (Motzko et al., 2012)

Für die technische Ebene der im Gutachten untersuchten Planungsprozesse, brachte der Einsatz neuer Technologien „Modellierungen und Prognosen und somit Konstruktionen“ hervor, die am Beginn des Betrachtungszeitraums (noch) nicht umsetzbar waren. Allerdings folgern die Autoren, dass (inzwischen mögliche) Vorteile dieses technischen Fortschritts durch a) zunehmend

schwierigere IT-Bedienungsschnittstellen, b) aufwändigere Eingabe (von immer komplexer werdenden Planungsgegenständen) sowie c) die gestiegenen Qualitätsansprüche seitens der Auftraggeber relativiert werden (Motzko u. a., 2012).

Diese Art von Rebound Effekten, trotz erweiterter Möglichkeiten durch technischen Fortschritt (innovative Materialien, Verarbeitungsweisen, Konstruktionsmöglichkeiten usw.) und zunehmende Digitalisierung (Beherrschung komplexer fachlicher Systeme durch virtuelle Modellierung usw.), kann als Symptom auch für darunterliegende Ursachen in der Art wie wir (immer noch traditionell) planen gedeutet werden. Im Forschungsfeld der Planungsmethodik beschreibt Rittel (1972) schon Anfang der 1970er in seiner Diskussion zur „Planung in der Krise“ eine den Planungsproblemen selbst inhärente Komplexität, die nicht mehr mit tradierten, einem Taylor'schen Verständnis folgenden, konsekutiven Planungsprozessen beherrschbar erscheint. Mit folgenden 11 Eigenschaften beschreibt er diese „böartigen“ Planungsprobleme:

- haben keine definitive Formulierung – jeder Schritt hin zu einer Lösung wirft weitere Fragen bezüglich der Problemstellung auf
- alle Formulierungen korrespondieren jeweils zu Ausdrücken ihrer Lösung und vice versa – Problemverständnis ist identisch zur Problemlösung
- haben [wegen der beiden ersten Eigenschaften, Anm. des A.] keine natürliche „Stopregel“ bezüglich ihrer Lösung – es kann endlos weiter nach besseren Lösungen gesucht werden
- ihre Lösungen können nicht mit RICHTIG oder FALSCH bewertet werden – unter variierenden Graden kann lediglich gut oder schlecht bewertet werden

- haben keine endliche, auflistbare Menge erlaubter (Lösungs-) Operationen – „[sic] everything goes as a matter of principles and fantasy“
- ihre Lösungsrichtung wird vom initialen Schritt der Problemdeutung („Wieso gibt es ein Problem?“) bestimmt – der Erste entspricht dem am entscheidendsten, eine Behandlung des Problems bestimmenden Handlungsschritt
- können immer als Symptom weiterer Probleme betrachtet werden – jeder mögliche Lösungsansatz kann daher an der eigentlichen Ursache des Problems vorbeigehen
- können weder unmittelbar noch ultimativ getestet werden – jede Aktion als Reaktion auf das Problem kann unterschiedliche Konsequenzen im Zeitablauf haben, zeitversetzt ggf. zu anderen Rückschlüssen eines Lösungsansatzes führen
- sind jeweils einmalige (nicht wiederholbare) Operationen – ob potentielle Konsequenzen einer Reaktion auf das Problem gut sind oder nicht, kann nur zu einem gewissen Grad vorhergesehen bzw. simuliert werden
- sind im Wesentlichen jeweils einzigartig – es kann bei der Problemlösung nicht für das nächste Mal gelernt werden, Strategien können nur sehr bedingt von der Vergangenheit in die Zukunft übertragen werden
- deren Problemlöser*innen haben kein Anrecht darauf falsch zu liegen – Eigenverantwortung bestimmt jedes Tun beim Problemlösen

(Rittel, 1972, S. 392–393)

2.1.2 Paradigma und Voraussetzung

Unter dem Begriff der Integralen Planung formierte sich in den 1980er ein ganzheitlicher bzw. systemischer Ansatz, der einer inhärenten Komplexität der Planungsprobleme Rechnung trägt. Von Both (2011) gibt ausgehend hiervon ein zusammenfassendes Resümee zum Stand der integralen Planung im Bereich der Planungsmethodik des Bauwesens, dem die vorliegende Arbeit folgt. Wesentliche Deutung und Einordnung des Begriffes werden daraus im Folgenden ergänzend durch weitere Quellen dargelegt.

Die optimale, umfänglich den Stand der Technik spiegelnde Aspekte berücksichtigende, Verwirklichung eines Bauvorhabens soll mit der, als Planungsphilosophie durch Kohler und Stulz definierten IP ermöglicht werden (vgl. Suter u. a., 1986). Zur ganzheitlichen Erfassung relevanter Aspekte betrachtet die IP dabei den langfristigen Lebenszyklus des Bauwerkes. Dazu werden auf organisatorischer Seite dazu kooperative Arbeitsweisen adressiert, da von einem in Kooperation integral planenden Team ausgegangen wird. Für eine IP werden folgerichtig, neben einer lediglich subjektiv wertbaren Lösung des Planungsproblems (Rittel, 1972), daher die Beteiligung einer Vielzahl Wissens- und Entscheidungsträger an den bauplanerischen Aktivitäten vorausgesetzt. Des Weiteren geht die IP von der Voraussetzung aus, dass die zumeist komplexen Probleme im Planungs- und Baubereich zur ganzheitlichen Lösung auch komplexer Ansätze bedürfen. Erkannte systemische Wirkzusammenhänge verdeutlichen, dass sich eine nicht sachgemäße Herangehensweise, wie z.B. am Anfang der Planung eine inhärente Komplexität des Planungsproblems durch Vereinfachung auszublenden und im weiteren Planungsverlauf dann symptomatisch auf daraus resultierende Planungslücken zu reagieren, prinzipiell nicht rechnet (vgl. von Both, 2011).

So manifestiert sich mit dem Ansetzen der ersten Lösungsideen ein Lösungsraum, der sich sukzessive mit der Konkretisierung der Planungslösung verengt. Daher nehmen nach der Planungsinisierung auch die Möglichkeiten eine Planungsqualität zu beeinflussen im Planungsverlauf rapide ab. Umgekehrt dazu stehen Planungskosten im Gesamtumfang mit ca. 2-4% Prozent dem Groh der Kosten ab der Herstellung des Bauwerkes gegenüber (vgl. Abbildung 1). In

der frühen Planung relativ preisgünstig behebbare Planungsfehler führen, je später sie im Prozess erkannt und behoben werden müssen, zu großen Kostensteigerungen, wie beispielsweise häufig in der Ausführungspraxis anzutreffenden Nachträgen. Diese „(...) steigenden Kosten einer notwendigen Planungsanpassung führen im Planungsverlauf nicht selten zu starken Abstrichen hinsichtlich der Planungsqualität.“(Kovacic u. a., 2012; Zitatquelle: von Both, 2011, S. 234, vgl. Both, 2011)

Die frühen (strategischen) Planungsphasen sind für eine IP von größter Wichtigkeit. Hierzu hat sich unter dem Begriff einer Horizontalen Integration, der Ansatz etabliert, alle beteiligten Akteure und Kompetenzen, die im Lebenszyklus des Bauwerks stehen, einzubinden. Somit können die in frühen Phasen noch möglichen Planungsspielräume zur Abstimmung eines ganzheitlichen Lösungsansatzes genutzt werden (vgl. von Both, 2011). Der Benefiz einer durch wenig (ca. 2-4%) Mehrkosten in diesen für die weitere Planung, die Herstellung sowie den Betrieb so bedeutsamen (weichenstellenden) Phasen wichtigen Integration wird häufig hervorgehoben (Kovacic u. a., 2012). Auf Seiten der Theorie deckt sich der Ansatz der IP dabei auch mit Ideen des System Engineerings, wonach auf der konzeptionellen Ebene frühzeitig eine ganzheitliche Optimierung und Abstimmung stattfindet. Damit diese methodisch in den frühen Phasen einbezogenen Belange im Planungsverlauf durchgängig konsistent gehalten werden, gilt es die fachlichen Konzepte und Planungslösungen bezüglich nachhaltiger Zielstellungen und Anforderungen, durch stetiges Prüfen ihrer Auswirkungen auf das spätere Gebäudeverhalten, planungsbegleitend zu validieren. Bezüglich einer Sicherung von Planungsqualität, von gestalterischen Aspekten bis hin zu ökologischen, ökonomischen, funktionalen, technischen sowie sozialen Fragestellungen, kann IP als anforderungsgetriebener Regelmechanismus verstanden werden. Im Planungsverlauf optimiert die IP dabei durchgängig die Planungsqualität als **iterativer Prozess**, indem jeweils aus den unterschiedlichen Sichten auf das Planungsproblem (Problemanalyse) – durch Integration daraus entspringender Lösungsansätze und fachlicher Konzepte (Systemsynthese) eine ganzheitlich validierte Planungslösung (Systemanalyse) – zusammengeführt wird (von Both, 2011).

2.1.3 Grundlegende Anforderungen

Freilich bedarf es zur generellen Unterstützung von verschiedenartigen Planungsprozessen einer Bestimmung neuralgischer Ansatzpunkte, die zu klären vermögen wo dieser iterative Prozess im spezifischen Planungsverlauf (bzw. mit welchen Inhalten und Beteiligten) zur Validierung der Planungsqualität anzusetzen ist. Dies stellt als bisher noch ungelöste Forschungsfrage eine erste Anforderungsebene der IP dar. Die lässt sich jedoch nur schwer wegen der situationsbedingten Eigenständigkeit unterschiedlicher Planungsprojekte generalisieren. Jedoch zeigen theoretische Ansätze diesbezüglich – wie beispielsweise am Modell einer generalisierten Planungssituation – die Wichtigkeit einer Unterscheidung zwischen der Beschreibung der Prozess- und Produktmodelle – auf der Ebene der Prozesse bzw. der (Planungs-) Objekte – für das planungsmethodische Gelingen einer IP auf (vgl. von Both, 2006). In einem holistischen Ansatz der Modellierung hierzu stellt von Both (2006) ein systemisches Projektmodell vor, das die eigenständig modellierten Teilsysteme – Prozessschicht bzw. integriertes Produktmodell – in Beziehung setzt. Konzeptuell wird der Projekthinhalte von der Projektumgebung abgegrenzt. Ein kooperativer Planungsrahmen ermöglicht dabei eine Koordination der Informationsflüsse durch rahmengebende, kommunikationsunterstützende und informationslogistische Strukturen, in denen dann (im Sinne eines Informationsträgers) die Inhalte der, vom Rahmen unabhängig modellierten, strukturierten Informationsbasis orchestriert werden können (ebd.). Der Ansatz unterteilt folgerichtig den Primärprozess der Objektplanung, von den, diesen unterstützenden Sekundärprozessen (projektinterne Leistung – Prozesse des Projektmanagements). Horizontale (Lebenszyklus-phasenübergreifender Aspekte) und vertikale (disziplinübergreifende Aspekte) Integration, als Anforderung an die durchgängige Informationsbasis der zu unterstützenden integralen Planungsmethodik, werden somit kohärent (im Rahmen des übergreifenden Projektmodells) in Beziehung zueinander gesetzt. Im Rahmen des ganzheitlichen Projektmodells finden dann, in Teilsystemen beschriebene, Sekundärprozesse „Inhaltliche Synchronisation“ und „Analyse“ (Kernprozess

der Entscheidungsfindung; vgl. "iterativer Prozess der Schritte Problemanalyse, Systemsynthese und Systemanalyse" von Both, 2011), als eigentliche Unterstützung des Primärprozesses statt (vgl. ebd.).

Neben situationsbedingt ausprägenden Entscheidungsunterstützungen (z.B. Sekundärprozess zum Validieren eines Planungszieles am spezifischen Stand der Planungslösung im Prozess) erscheint eine weitere Untergliederung von Bestandteilen dieser generalisierbaren Sekundärprozesse im Sinne von tertiären Prozessen (vgl. zur Definition „tertiärer Prozess“ Biehal, 1994; sowie Bringel u. a., 2004) möglich, die in verschiedenen Planungssituationen wiederkehrende Fragestellungen zur Planungslösung beantworten (z.B. regulatorische Vorgaben, energetische Zielstellungen usw. betreffend).

Da die vorliegende Arbeit an diesem (hinreichend generalisierbaren) iterativen Prozess zur Validierung der Planungsqualität als Anforderung der IP ansetzt, wird an dieser Stelle nicht weitergehend auf theorie- bzw. praxisorientierte Diskussionen zu generellen prozessbezogenen Anforderungen einer IP eingegangen. Jedoch sei diesbezüglich auf Arbeiten im Fachgebiet des Autors verwiesen, die sich mit synthetischen Metamodellen auf der Ebene des Prozesses sowie einer phasenorientierten Gliederung gewichtiger Aspekte der IP auseinandersetzen (vgl. Rexroth & Both, 2014; Rexroth & von Both, 2016; Rexroth u. a., 2018). Diese stehen vor dem Hintergrund der Schaffung weiterer theoretischer, prozessbezogener Grundlagen der IP, auf denen auch neuere Forschungsvorhaben generalisierende Hilfestellung für die praktische Umsetzung der IP untersuchen (vgl. Kap. 8.1). Im Folgenden werden die Anforderungen auf Ebene des Planungsgegenstandes bezüglich des integralen Planungsansatzes weiter ausgeführt.

Eine durchgängige Sicherung der Planungsqualität bedarf einer einheitlichen Dokumentation der Planungslösung im Verlauf des Planungsprozesses. Diese wichtige Anforderung einer iterativen Validierung der Planungsqualität impliziert die disziplinübergreifende Zusammenführung sämtlicher (Teil-) Lösungen zu den fachlichen Fragestellungen, wie auf Ebene der Bauwerksplanung, z.B. zu architektonischen Raumkonzepten, Standsicherheit, Brandschutz, technischem Ausbau usw., um auf dieser umfangreichen Basis das gesamtheitliche konzeptionelle, räumliche bzw. technische Gefüge der Planung bewerten zu

können (vgl. Rexroth & von Both, 2016). Eine Bedienung dieser Anforderung stellt große Herausforderungen an die Art und Weise einer gemeinsamen Beschreibung der Planungslösung dar, insbesondere durch die Vielzahl der zu beachtenden Aspekte sowie der erkannten komplexen Wechselwirkungen zwischen diesen. Zum Beispiel innerhalb von Fachdisziplingrenzen, wie bauphysikalischen dreidimensional wirkenden Wärmebrücken, aber auch disziplinübergreifenden, wie räumlichen Kollisionen technischer Gebäudeinfrastrukturen mit Tragwerksystemen. Mit tradierten, planbasierten Methoden lassen sich viele dieser Aspekte nicht mehr beschreiben und planerische Konflikte daher auch nicht durch Linienzeichnungen davon ableiten. Dabei angewendete visuelle Kontrolle reicht selbst bei dreidimensionalen Geometrieinformationen nicht mehr aus, so dass es allein zur räumlichen Kohärenzwahrung darüberhinausgehender algorithmischer Kollisionsdetektion bedarf. Allein auf Basis einer, mit disziplinübergreifenden semantischen Metainformationen ergänzten, einheitlich repräsentierten Planungslösung können oftmals erst durch logisch-funktionale Prüfung und multi-kriterielle Simulation Probleme erkannt bzw. diesen mit gezielten Planungsoptimierungen entgegenwirkt werden (siehe sowie vgl. für den Absatz von Both, 2011). Elementar hierfür ist eine umfängliche (über den Lebenszyklus eines Bauwerks) und einheitliche (disziplinübergreifenden) Repräsentation der Planungslösungen. Als Integrationsgrundlage zur Zusammenführung dieser heterogenen Informationen erfüllen virtuelle Produktmodelle diese Voraussetzung durch ihre durchgängige digitale und damit validierbare Beschreibungsmethode. Auf den damit datentechnisch abbildbaren, unterschiedlichen konstruktiven, funktionalen sowie technischen Zusammenhängen einer Planungslösung können dann die Simulations- und Evaluierungswerkzeuge aufbauen (ebd.). Für die Bauwerksplanung impliziert dies einen durchgängigen rechnergestützten Entwurfsprozess, der als fundamentales *Ökosystem* eine Entwicklung und Anwendung innovativer AEC Lösungen erst ermöglicht. Eine „essentielle Voraussetzung zur Erreichung qualitativ hochwertiger und nachhaltiger Planungslösungen“ (ebd. S.236) kann dann im Rahmen dieser umfänglich digitalisierten Prozesse mit den in virtuellen Bauwerksmodellen repräsentierten Planungsgegenständen bzw. darin verorteten intelligenten bauteilorientierten Beschreibungen im Zusammenspiel mit entsprechenden Evaluierungsmöglichkeiten erfüllt werden (vgl. von Both & Ebertshäuser, 2018).

2.1.4 Baustein (Building) Information Modeling

Für die Gebäudeebene formiert sich zunehmend unter dem Begriff *Building Information Modeling* eine Methodik zur Gestaltung abspracheloser Datenflüsse im Prozess auf Basis vereinheitlichter Informationsstrukturen. Die Methode kann dabei als datentechnischer Baustein einer ganzheitlich integrierten Planungsmethodik verstanden werden (vgl. von Both & Ebertshäuser, 2018). Eine Anwendung der Methode impliziert gegenüber der herkömmlichen planbasierten Methode neben der modellbasierten Form der Abbildung des Planungsgegenstands auch Arbeitsweisen einer grundlegend veränderten Planungskultur (vgl. Abbildung 4).

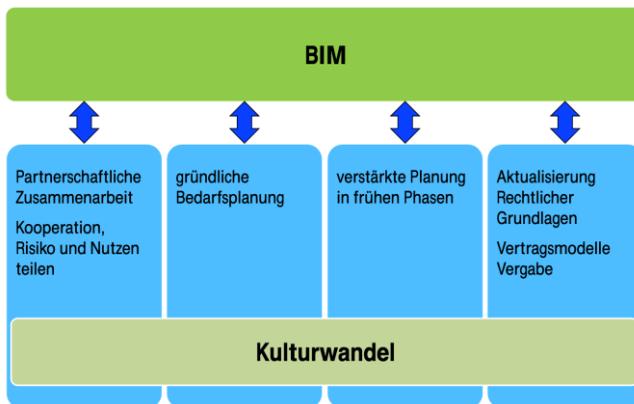


Abbildung 4 BIM – Querschnittsdisziplin (Quelle: May, 2014; Folie 28)

Mehrere Ansätze sehen durch diese Methodik schon die „nahtlose“ Verwirklichung des, von der integralen Planung geforderten Hilfsmittels einer durchgängigen Datenbasis (Achammer & Salzburger, 2014; Lützkendorf u. a., 2009). Im Sinne einer ganzheitlichen Planungsmethodik, die sich explizit nicht auf eine Skalierungsebene „Gebäude“ einschränkt, sondern als ganzheitlicher Regelmechanismus (vgl. von Both & Ebertshäuser, 2018) neben einer Ebene der Fachdisziplinen ebenso eine räumliche Planung einbeziehen kann, erscheint diese Sicht jedoch verkürzt. Zudem steuert eine vornehmlich technologiegetriebene Umsetzung der BIM Methode, wie sie derzeit auf Seiten großer

AEC/CAD-Hersteller vorangetrieben wird, eher mit einzelnen datentechnischen Lösungsiseln um deren proprietäre Beschreibungsformate auf ein ‚Turmbau-zu-Babylon-Problem‘¹ zu. Fraglich bleibt bei diesen unter dem Begriff *Closed BIM* summierten Ansätzen, ob sie die eigentliche Anforderung einer integralen Planung an eine durchgängige, transparente sowie kohärente Datengrundlage – idealerweise auf Basis einer herstellerneutral und offen standardisierten, integrierten Beschreibung der Planungslösung – problemorientiert und skalenunabhängig adressieren können. Durch die historisch in dem oftmals noch von konsekutiven Planungsvorgehen sowie vom Leanmanagement großer Generalübernehmer (GÜ) geprägten, nordamerikanisch bzw. angelsächsischen Raum umgesetzten BIM Methodik ist darüber hinaus noch wenig geklärt, wie bisher darin vorgeschlagene neue Rollen, z.B. inflationär benutzte Begriffe „BIM-Manager“ oder „BIM-Koordinator“ allgemeingültig definiert werden können, sodass sie in Bezug zu hiesigen Planungssituationen (bspw. bezüglich der Honorarordnung für Architekten und Ingenieure – HOAI) zweckdienlich in das Planungsvorgehen eingeordnet werden können (Wildenauer, 2016). Die vorliegende Arbeit folgt daher im Sinne einer integralen Planung einem Verständnis bezüglich der BIM Methodik, welches im Rahmen von den Begriffen *Open BIM* oder *Big BIM* diskutiert wird. Zentraler Angelpunkt ist dabei ein herstellerneutrales Austauschformat, wobei sich entsprechende konkrete Ausgestaltungen einer modellbasierten Arbeit, insbesondere im Bereich der Öffentlichen Hand (vgl. Bramann & May, 2015), um den etablierten Produktmodellstandard Industry Foundation Classes (IFC; ISO 2018; vgl. Kap. 2.2.3) formieren. Generell zeigt sich in der Praxis im deutschen Baubereich – wie es bezüglich der Struktur der Softwareumgebung in Abbildung 5 dargestellt ist – eine klare Präferenz zur Arbeit mit dem offenen IFC Standard im Rahmen der (kooperativen) Umsetzung der BIM Methode.

¹ vgl. D. Probst (2012) „Buildings' 'Tower of Babel' problem: They can't share data“ (Quelle: <https://www.greenbiz.com/blog/2012/03/16/buildings-tower-babel-problem-they-cant-share-data>)

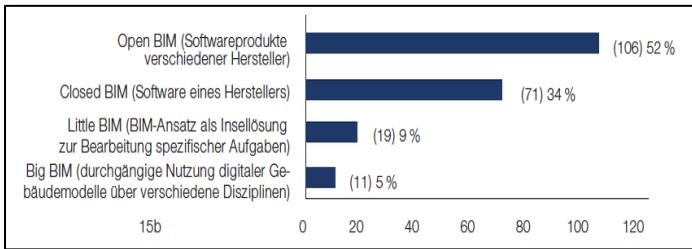


Abbildung 5 In welcher Struktur organisieren Sie Ihre Softwareumgebung? (Quelle: Herrmann & Westphal, 2017)

Vor diesem Hintergrund werden im Rahmen der Richtlinien und Normungsarbeit derzeit Grundlagen für ein offenes modellbasiertes Arbeiten geschaffen. Im Gremium des Vereins Deutscher Ingenieure (VDI) wird zum VDI 2552 Blatt 10 dazu beispielsweise verbindliche Definitionen und weiterführende Handreichungen für „Auftraggeber-Informationsanforderungen“ (AIA) und „BIM-Projektentwicklungspläne“ (BAP) erarbeitet. Im Wesentlichen geht es dabei um organisatorische Mittel, mit denen sich beispielsweise in festzulegenden sogenannten Datenübergabepunkten eine bestimmte Qualität der Abbildung der Planungslösung festlegen lässt (vgl. auch im engl. verwendeten Begriff *quality gate*). Diese rahmengebenden Grundlagen zur Ausgestaltung einer verbindlichen modellbasierten kooperativen Arbeitsweise bedienen sich dabei bereits normierter Rahmenwerke, wie dem formalisierten Informationslieferungshandbuch (Information Delivery Manual – IDM; ISO, 2016; vgl. Beetz u. a., 2015). Ergänzend zu dem den Planungsgegenstand mit lebenszyklisch-orientierten Datenstrukturen beschreibbar machenden IFC Standard, werden von der Standardisierungsorganisation buildingSMART begleitende Standards und Verfahren zur Unterstützung von dessen Einbindung als Austauschformat in den kooperativen Planungsprozess (weiter-) entwickelt. So bilden die im Rahmen einer IDM formal abgebildeten sowohl prozess- wie auch objektbezogenen Informationen die Grundlage zum Abbilden einer formalisierten Modellsicht auf die IFC als (fachliche) Teilmenge. Mit dem *Model View Definition*

(MVD²) Spezifikationsstandard können mit diesen Modellsichten die Beschreibungsmöglichkeiten der IFC sowohl beschränkt als auch erweitert werden. Zahlreiche offizielle Teilbilder werden in den Fachgruppen der buildingsSMART erarbeitet, und der Standard dadurch an die Anforderungen in der Praxis angepasst. Im Gegensatz zum in der Stadtplanung etablierten CityGML Standard (vgl. Kap. 2.2.3) ist bezüglich der modellbasierten Arbeit im Rahmen offener Modellstandards im Themenbereich BIM bereits das arbeitsmethodische Umfeld weiter ausgereift. Daher bietet sich dieses um die herstellereutralen IFC gebildete normen- bzw. standardbasierte „Ökosystem“ einen guten Ansatzpunkt für die vorliegende Arbeit bezüglich der Veranschaulichung einer durchgängigen Datenbasis im Kontext einer Integralen Planung. Durch Bezügen zu einem modellbasierten Arbeiten mit dem IFC Produktmodell zur Abbildung des Planungsgegenstandes, können die von der Arbeit verfolgten, darauf aufbauenden Entscheidungsunterstützungen im Planungsprozess gezeigt werden (vgl. Kap. 4).

2.1.5 Methodische und technische Hilfsmittel

Im Kontext einer planungsmethodischen Vorgehensweise zur Bewältigung der inhärenten Komplexität von Planungsaufgaben zeichnen sich Hilfestellungen sowohl auf der Ebene des Prozesses wie auch auf der Ebene der Planungsgegenstände ab. Im Weiteren werden ausgewählte Hilfsmittel zur Verdeutlichung eines diesbezüglichen Instrumentariums der Planenden auf diesen beiden Ebenen dargestellt.

Leitfäden, Anleitungen und Vorschriften

In Handlungsanleitungen sind bereits früh Kernelemente einer Integralen Planung bei Hilfestellungen zum methodischen Vorgehen zu finden. In der kommunalen Planung werden beispielsweise im Handlungsfeld der energieeffizienten Stadtentwicklung von der Landesebene Vorgehenshilfen zur

² vgl. <https://technical.buildingsmart.org/standards/mvd/>

Erstellung von Energienutzungsplänen an die Kommunen gegeben (Hausladen, u. a., 2011). Damit werden ehemals allein für eine geregelte Ordnung räumlicher Verhältnisse entwickelte Instrumente der Raumordnung weiterentwickelt. In dem die integralen Methoden, wie eine frühe Beteiligung sowohl der Akteure als auch der Betroffenen einer Planung, in die Verfahren aufgenommen werden, können sie für die neuen Herausforderungen ertüchtigt werden, wie beispielsweise der Ausbau der regenerativen Energieträger. Dabei werden in der Stadtentwicklung insbesondere erkannte hemmende Wirkungen durch ein Nichteinbeziehen der Bevölkerung mittels integrierter energetischer Leitbild- und Konzeptbildungen entgegengewirkt. Generell spielt das transparente Vorgehen eine wichtige Rolle. So haben Informationsportale zur Kommunikation mit bzw. der Einbeziehung und Mitnahme der von den Maßnahmen oftmals betroffenen Bevölkerung bezüglich der Planungsprozesse zur Darstellung und Kommentierung von Planungsprozessen oder andere Mittel zur Bürgerbeteiligung an Bedeutung für eine reibungslosere kommunale Entwicklungsplanung an Bedeutung gewonnen.

Auch bei der Gebäudeplanung existieren planungsmethodische Anleitungen, um mittels der Umsetzung integraler Methoden die Planungslösung bezüglich strategischer Zielsetzungen zu optimieren und somit insgesamt die Qualität des Bauens steigern zu können. Für das Handlungsfeld Nachhaltigen Bauen stellt die Öffentliche Hand auf Bundesebene diesbezügliche Leitfäden für den Rahmen von öffentlichen Bauvorhaben zur Verfügung (vgl. BMUB, 2016). Darin sind zweckmäßige Herangehensweisen phasenorientiert für die Planungs- und Bauprojekte aufbereitet und können auch über den Rahmen des öffentlichen Bauens hinaus als Hilfestellung zum Erstellen nachhaltiger Bauwerke verwendet werden. Bezüglich gewichtiger Entscheidungen in den frühen Planungsphasen mit immenser Auswirkung auf die Umweltwirkung des späteren Bauwerks wird darin beispielsweise auf ein frühzeitiges Einbinden entsprechender Fachexpertise hingewiesen (vgl. Abbildung 6).

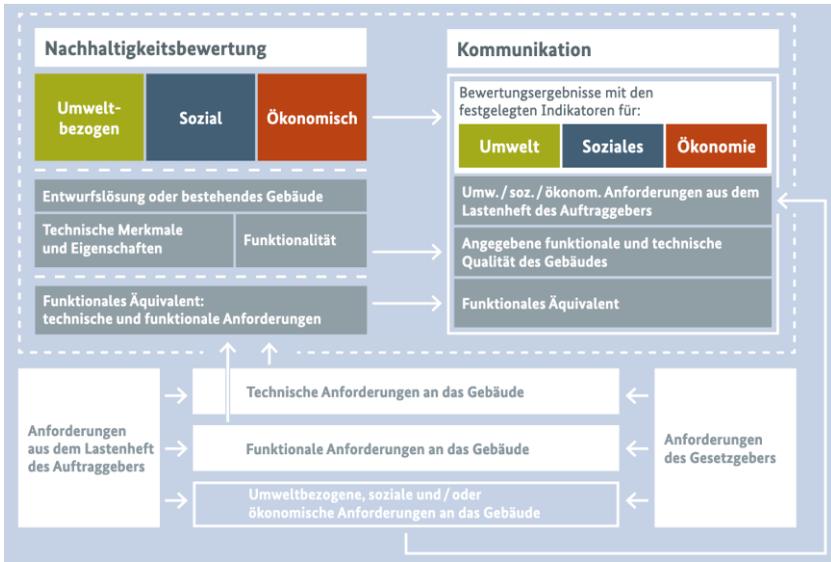


Abbildung 6 Konzeption der Nachhaltigkeit von Gebäuden (Quelle: BMUB, 2016 S. 17)

Ergänzend zu diesen methodischen Hilfestellungen werden auch entsprechende Methoden zur einheitlichen Bewertung der durch ein Befolgen der Anleitungen möglichen Nachhaltigkeitsqualitäten bereitgestellt, wie das Bewertungssystem Nachhaltiges Bauen (BNB). Das Vorhandensein dieser Art von Instrumenten zur Bestimmung eines messbaren Erfolges bei den im Planungsprozess verfolgten (bau-) fachlichen Zielstellungen bezüglich der baulichen Qualität bildet einen gewichtigen Aspekt in der Anwendung methodischer Hilfestellungen im Sinne der Integralen Planung. So können die darin verorteten Kriterien, neben einem Messen der Zielerreichung im Nachhinein, sowohl bereits zu Prozessbeginn bei der Zieldefinition, als auch bei den iterativen, planungsbegleitenden Überprüfungen der Planungslösungen eine wichtige Rolle spielen. Eine Zuwendung zu den (bereits für die Bewertung erstellter Bauwerke sowie existierender Quartiere) bestehenden Systemen zur Entwicklung von Ansätzen einer frühzeitigen Evaluierung von Planungsalternativen, kann

bei den im Weiteren exemplifizierten Methoden und Hilfsmitteln bezüglich des Planungsgegenstandes festgestellt werden.

Integrierte Entscheidungshilfen zum Planungsgegenstand

Wie im Kapitel 2.1.4 zusammenfassend dargestellt, erfordert ein systematischer, planungsbegleitender Einsatz von Hilfsmitteln im Sinne einer integralen Planung als Basis einer einheitlichen durchgängigen Informationsgrundlage im Planungsprozess. Daher können Lösungen zur Ermöglichung abspracheloser Datenflüsse mittels offener Modellstandards, sowohl auf Stadt- wie auch auf Gebäude- bzw. Fachebene, eine gewichtige Grundlage zur zweckmäßigen Anwendung der Hilfsmittel im Planungsprozess bilden. Hierzu haben sich verschiedene Produktmodellstandards als Grundlage eines modellbasierten Arbeitens in der Stadt- sowie Bauwerksplanung etabliert (vgl. Kap. 2.2.3). Darüber hinaus werden für eine vereinheitlichte und transparente Handhabung der Modelldaten auch modellspezifische Spezifikationssprachen gefordert (vgl. Eastman u. a., 2009). Hierzu sind zum Beispiel neben generellen Standards für Spezifikationssprachen (vgl. Kap. 2.3.2) zu dem IFC Standardisierungsansätze zu erkennen, die auf eine einheitliche Art des Zugriffs auf die modellbasiert abgelegten Informationen abzielen.

Vor diesem Hintergrund sich etablierender Produktmodelle gewinnen bei einer zunehmenden Komplexität in den Planungsaufgaben im Bereich der gebauten Umwelt (IT-gestützte) Simulationen und Planungshilfsmittel an Bedeutung. Damit lassen sich schon vor der Umsetzung die Planungslösungen in kostengünstiger Weise überprüfen und ggf. bezüglich der designierten Eigenschaften wie Energieeffizienz oder Nachhaltigkeit optimieren. Ohne bereits gewichtige Planungsentscheidungen treffen zu müssen, ermöglichen sie darüber hinaus eine fundierte Entscheidungsfindung durch das virtuelle Durchspielen von Planungsvarianten, bzw. das Durchleuchten bestimmter Aspekte, wie den möglichen Umweltwirkungen. Allerdings werden planungsbegleitende Anwendungen dieser digitalen Methoden und Werkzeuge jedoch durch den erforderlichen, (aufgrund fehlender standardisierter Schnittstellen im Datenaustausch oftmals bedingten) enormen Vorbereitungsaufwand gehemmt. Dabei fehlt beispielsweise zur Anwendung energetischer Simulationswerkzeuge

allein eine konkrete Spezifizierung anhand der Elemente des Produktmodellstandards zur vereinheitlichten Abbildung energierelevanter Informationen, um zahlreiche Energieplanungstools tatsächlich mittels standardisierter Schnittstelle an den Planungsprozess koppeln zu können. Ansätze hierfür, entsprechende IFC-basierte Modellsichten (MVD) zur Spezifizierung der energierelevanten Informationen in den Planungsmodellen zu entwickeln, wurden in Forschungsprojekten untersucht (Remmen u. a., 2015), sind jedoch noch Gegenstand laufender Standardisierungsarbeit. Einige der am Markt erhältlichen Werkzeuge besitzen neben den meist formularhaft ausgebildeten Eingabeschnittstellen zum manuellen Aufbereiten der energierelevanten Informationen bereits vereinfachte standardbasierte Schnittstellen, um beispielsweise die Bauwerksgeometrie aus IFC-basierten Planungsgegenständen lesen zu können. Diese bei den Werkzeugen zum Erstellen des bundesweit geltenden Energieausweises zu beobachtende Entwicklung hin zu standardbasierten Schnittstellen, kann auch beispielsweise bei der im US-amerikanischen energetischen Nachweisverfahren eingesetzten offenen Simulationsplattform EnergyPlus beobachtet werden.

Auf Forschungsseite beschäftigen sich zahlreiche Ansätze mit der Fragestellung, wie sich mit der dynamischen Modellierung numerische Simulationen zu verschiedenen bauphysikalischen (z.B. thermodynamische) Aspekten bewerkstelligen lassen. Dabei kommt der Frage, wie sich eine sehr aufwändige (manuelle) Parametrisierung der Simulationsmodelle durch Schnittstellen zu modellbasierten Planungsdaten drastisch vereinfachen lässt, eine an Bedeutung gewinnende Rolle zu. Denn obwohl es bereits Ansätze zur kommerziellen Anwendung dieser forschungsnahen Methoden und Werkzeuge in der Praxis gibt, werden diese jedoch zumeist wegen des aufwändigen Vorbereitungsaufwands nur in sehr umfangreichen Großprojekten mit entsprechenden energiebezogenen Schwerpunkten im Gebäudeentwurf eingesetzt. Unter dem Dach der International Building Performance Association (IBPSA³) werden derzeit Projekte⁴ zur Bündelung der in Forschungsprojekten zum Thema Gebäudesimulation

³ <http://www.ibpsa.org/>

⁴ vgl. ursprünglich International Energy Agency (IEA) Annex 60 Projekt; <http://iea-annex60.org/>

untersuchten (IT-gestützten) Lösungen fortgeführt. Darin soll neben der forschungsseitigen Verbreitung offener Simulationsansätze mittels der objektorientierten Modelica Simulationssprache auch generell die Forschungsarbeit einer breiten Praxis zugänglich gemacht werden (vgl. Wetter u. a., 2019; Remmen u. a., 2015; Wetter & Treeck, 2017).

Von einer unzureichenden standardisierten Spezifikation entsprechender Fachinformationen sind neben dem bereits gut mittels IT-basierter Hilfsmittel unterstützten Handlungsfeld der Energieplanung noch weitere Handlungsfelder betroffen. Im Rahmen der oben beschriebenen Bewertungssysteme werden beispielsweise im Handlungsfeld des Nachhaltigen Bauens von der öffentlichen Hand entsprechende IT-gestützte Methoden und Werkzeuge entwickelt. Für die Zielplattform *elektronische Bewertungssystem Nachhaltiges Bauen* (eBNB), in der die bewerteten Bauwerksinformationen zu allen Bundesbauten in Deutschland zusammenfließen, wird mit dem *eLCA* Werkzeug dabei ein Ökobilanzhilfsmittel bereitgestellt. In laufenden Forschungsprojekten werden sowohl zu diesem Werkzeug als auch zu den im Rahmen der Deutsche Gesellschaft für Nachhaltiges Baues (DGNB) Zertifizierung eingesetzten, vom Fraunhofer Institut für Bauphysik (IBP) in der Abteilung Ganzheitliche Bilanzierung entwickelten GENERIS Tool entsprechende Schnittstellen zum modellbasierten Arbeiten auf Basis der IFC untersucht (vgl. Kap. 8.1). Bezüglich einer Unterstützung der Planung in den frühen Phasen mit den für die Nachhaltigkeit des Bauwerks gewichtigen Entscheidungen werden zudem in der Forschungsarbeit dabei Grundlagen untersucht (vgl. Ebertshäuser u. a., 2019). Zwar gibt es aus dem Forschungsbereich entstammende erste Softwareanwendungen für den planungsbegleitenden Einsatz, wie das OneClickLCA⁵ Werkzeug, jedoch verbleibt eine produktive Anwendung der Ansätze bisher oftmals als Aufsatz proprietärer CAD-Systeme. Offene Standards werden dabei nur unzureichend unterstützt oder erfordern ein aufwendiges Nachmodellieren. Aufgrund von speziellen Informationsbedarfen und Strukturierungen können

⁵ <https://www.oneclicklca.com/>

die dabei nachmodellierten IFC Informationen dann oftmals nach ihrer Aufbereitung für eine konkrete Planungsunterstützung nicht mehr in den Planungsprozess zurückgekoppelt werden.

Weitere forschungsseitig untersuchte Ansätze bezüglich einer Parametrik zielen durch ein direktes Anzeigen beispielsweise einer Verschattung auf eine Rückkopplung von Auswirkungen der Entwurfsentscheidungen schon bei ersten Skizzen. Dabei werden Lösungen auf Basis des Modellierungswerkzeug Rhinoceros⁶ sowie seinem Aufsatz Grasshopper⁷ zur visuellen parametergestützten Programmierung untersucht (vgl. Plotnikov u. a., 2016). Darüber hinaus untersuchen weitere Ansätze mittels des Honeybee⁸ Werkzeuges eine Kopplung von einfachen bauphysikalischen Berechnungen an die Planungsinformationen (Ghobad, 2018). Einer Unterstützung des modellbasierten Arbeitens mit offenen Standards steht bei diesen Ansätzen allerdings entgegen, dass sie auf proprietäre Werkzeugen basieren oder als Aufsatz auf diese konzipiert sind und daher auf proprietären Modelldatenstrukturen aufbauen bzw. sie sich oftmals allein auf die Geometrie der offenen Modellstandards beschränken.

In der Praxis wird jedoch im Rahmen der Arbeit auf Basis offener Modellstandards bezüglich der Nachhaltigkeitsbetrachtung eines Bauwerkes und einer dazu angestrebten Ökobilanz (LCA) ein Verknüpfen der Planungsentscheidungen zu belastbaren Werten einer möglichen Umweltwirkung benötigt. Dadurch können die gewichtigen Entscheidungen im (frühen) Planungsprozess unterstützt werden, die entscheidenden Einfluss auf die Nachhaltigkeit haben. Diesbezüglich vielversprechende Ansätze lassen sich beim LCA-basierten Untersuchen von Benchmarks auf Basis bereits bewerteter Bauwerke bzw. typologisch zusammengestellter Referenzwerte als Grundlage der Entscheidungsunterstützung in den frühen Phasen finden (vgl. Rexroth u. a., 2018; S. Ebertshäuser u. a., 2019). Die methodische Herangehensweise ist dabei ähnlich dem Prinzip der Baukostenindexe (BKI⁹) bei der Unterstützung von Ent-

⁶ <https://www.rhino3d.com/de/>

⁷ <https://www.grasshopper3d.com/>

⁸ <https://www.ladybug.tools/honeybee.html>

⁹ <https://www.bki.de/>

scheidungen in wirtschaftlichen (Vor-) Überlegungen. Dieser Forschungsbereich kann als weiteres Handlungsfeld von den in dieser Arbeit zunächst für die Energieplanung dargestellten Lösungen vor dem Hintergrund eines offenen, modellbasierten Arbeitens dienen (vgl. 8.1).

2.2 Modellierung

Im Folgenden wird im weiten Feld der *Modellierung* der Bezug zu dem Strukturieren von Daten als einem Teilbereich der Informatik hergestellt, der eine Grundlage der vorliegenden Arbeit bildet. In Abgrenzung zu im engeren Sinn unmittelbar zur Software gehörenden Daten (Quelltext usw.), wird der Modellierungsgegenstand *Daten* dabei als Begriff im Sinne der Arbeit für diejenigen verwendet, die zum fachlich-inhaltlichen Zweck der (IT-) Systeme gehören.

2.2.1 Begriffsklärung im Sinne der Arbeit

Modellbegriff

Generell kann ein Modell als abstraktes Abbilden von (Phänomenen) der Realität bzw. einer Idee verstanden werden. Dabei bildet ein Subjekt gemäß eines angesetzten Zwecks ein Original in Form eines Modells ab (vgl. Abbildung 7, Bildquelle: Claus & Schwill, 1997). Die sich ergebende Relation R (Subjekt, Zweck, Original, Modell) ist bestimmend für das sinngemäße Verständnis der Abstraktion in der durch das Modell intendierten Sicht auf das Original. Auf

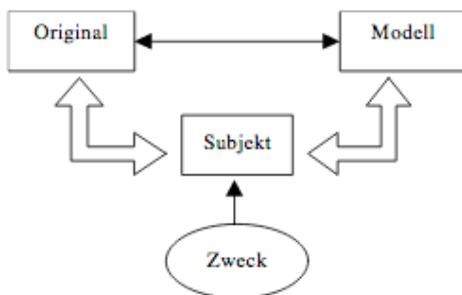


Abbildung 7 Schaubild Relation Modellbildung

Basis dieser generalisierten Repräsentation einer Modellbildung liefert der Bereich der Modelltheorie weiterführende Ansätze zur allgemeingültigen Definition eines *Modells* (vgl. Claus & Schwill, 1997).

So definiert Stachowiak (1973) drei wesentliche Merkmale eines Modells in seiner Allgemeinen Modelltheorie (AMT) als

- Abbildungsmerkmal «Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können» (AMT, S. 131)
- Verkürzungsmerkmal «Modelle erfassen im Allgemeinen nicht alle Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant scheinen» (AMT, S.132)
- Pragmatisches Merkmal «Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion:
 - für bestimmte - erkennende und/ oder handelnde, modellbenutzende - Subjekte;
 - innerhalb bestimmter Zeitintervalle und
 - unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen» (AMT, S. 132f)

Eine weitere Vertiefung der allgemeinen Modelltheorie übersteigt den Rahmen der vorliegenden Arbeit, daher wird im Folgenden, ausgehend von konkreten Beispielen von Modellen im Bereich der räumlichen Planung (Architektur, Stadtplanung), auf Modelle in der Informatik eingegangen. Auf Seiten der Modelltheorie wird jedoch – da jedes Modell auch als System verstanden werden kann – im folgenden Kapitel dieser, für die vorliegende Arbeit bedeutsame, systemorientierte Bereich des Modellbildens (zusammen mit einem dabei unterlegten Vorgehen) weiter erörtert.

Ein (materielles) Modell bildet den Planungsentwurf bzw. die räumliche Situation der gebauten Umwelt ab. Ein (immaterielles) Entwicklungsmodell bildet historische bzw. prognostizierte Veränderungen mit alphanumerischen oder grafischen Darstellungen ab. Mit dem Einzug der Digitalisierung in dieses Feld der Planung, wird auch eine virtuelle maßstäbliche Abbildung eines Entwurfes als digitales Architektur-, Bauwerks- oder Stadtmodell bezeichnet (Janke, 1978).

Wichtig für das Verständnis dieser Modelle ist, dass jedem Modellansatz ein Beschreibungsansatz zum Abbilden der Realität zugrunde liegt, der, als Sprache veranschaulicht, ein Vokabular bereitstellt, mit dem die realen Phänomene abstrahiert werden können, z.B. ein zu räumlichen Miniaturen formbares Material, wie Holzwerkstoffe für ein physikalisches Architekturmodell, eine replizierbare Berechnungsvorschrift, bzw. eine grafische Formensammlung zum Darstellen von Entwicklungen. Bei digitalen Modellen bildet eine (zumindest binäre) Kodierung das (maschinen-interpretierbare) Vokabular ab, in dem das eigentliche Modell verfasst ist. Die mittels Datenschemata weiterentwickelte, spezialisiertere Beschreibungsansätze einer zugrundeliegenden – im einfachsten „binären“ Fall aus Digits (Zeichenfolgen) bestehenden – Modellsyntax werden im Kapitel 2.2.3 für die in der vorliegenden Arbeit behandelten Produktmodelle weiter konkretisiert.

Systembegriff

Ähnlich zum Modellbegriff wird in der vorliegenden Arbeit nur ein Ausschnitt der Bedeutung, die sich hinter dem Begriff System verbirgt, herausgegriffen. Die Systemwissenschaften liefern – mit dem Anspruch auf universelle Verwendbarkeit – Theorien zu dem Begriff System. Wobei trotz seiner mannigfaltigen Verwendung in unterschiedlichsten Bereichen kaum Ansätze einer Systematisierung existieren, und es daher eine „Vielzahl von sich überschneidenden Ansätzen, Aussagen, Definitionen und Methoden gibt, was die praktische Arbeit erschwert und unübersichtlich macht.“ (Händle & Jensen, 1974, S. 9). Da sich die vorliegende Arbeit im Sinne einer „Anwendbarkeit“ mit diesem Themenfeld auseinandersetzt, kann sie nur einen kleinen Teilbereich von dem bis heute kontrovers diskutierten Wissenschaftsbereich anschneiden. Die oben

angeführte Quelle kann jedoch als guter Ausgangspunkt zur weitergehenden Beschäftigung mit Grundlagen dieser Thematik empfohlen werden.

Für eine ganzheitliche Betrachtung verschiedener Teile in ihrem Zusammenspiel beschreibt der Begriff System ein übergeordnetes Gefüge. Innerhalb der bestimmten Grenzen dieses Gefüges lassen sich Wirkmechanismen beschreiben. Für den Bereich der Naturwissenschaften wird der Begriff im Duden folgendermaßen definiert:

„Gesamtheit von Objekten, die sich in einem ganzheitlichen Zusammenhang befinden und durch die Wechselbeziehungen untereinander gegenüber ihrer Umgebung abzugrenzen sind“ (Quelle: <https://www.duden.de/rechtschreibung/System>).

Generell als wissenschaftliches Schema bzw. „Prinzip, nach dem etwas gegliedert, geordnet wird“ bezeichnet, kann als System auch „eine Einheit aus technischen Anlagen, Bauelementen, die eine gemeinsame Funktion haben“ sowie eine „Menge von Elementen [bzw. Zeichen] zwischen denen bestimmte Beziehungen bestehen [bzw. die nach bestimmten Regeln zu verwenden sind]“ verstanden werden (ebd.). Zu der Bedeutung, die im Sinne der vorliegenden Arbeit, dem Begriff eines *Systemmodells* unterliegt, können diese allgemeinen Erläuterungen bereits die Grundlage darstellen. Das Systemmodell steht dabei stellvertretend für den Gegenstand der in der Arbeit verfolgten Prüfungen und Analysen von Planungsdaten. Ohne einen bestimmten Maßstab vorzugeben, soll damit ausgedrückt werden, dass der Lösungsansatz auf eine beliebige Zusammenstellung der Daten angewendet werden kann. Eine generelle Definition eines virtuellen Systemmodells ist nach Gausemeier et. al. (1995): „Das virtuelle Systemmodell ist ein einheitliches, logisches Arbeitsmodell des Systems. Es dient als Basis der Systementwicklung, -anpassung (...)“. Im Sinne der Arbeit steht konkretisierend hierzu das „einheitliche“ für eine einzige Modellsyntax XML, die den unter dem „logischen Arbeitsmodell“ verstandenen verschiedenen bedarfsorientiert zusammengestellten Planungsdatenmodellen unterliegt. Dem Lösungsansatz der vorliegenden Arbeit dient das derart gedeutete *Systemmodell* dann als einheitliche Datenbasis, auf der skalenübergreifende Analysen angesetzt werden können. Das Systemmodell trägt dabei die über die inhaltliche Information in den zusammengestellten Einzelmodellen

hinausgehende Information zu den Beziehungen der auf den unterschiedlichen Ebenen von der Stadt, des Bauwerks oder der (Fach-)Details befindlichen Modellelemente untereinander.

2.2.2 Gegenstand Daten

In den vorangegangenen Grundlagenkapiteln sind die Theorie und einhergehenden Bedeutungen von zentralen, im Rahmen der Arbeit verwendeten Begriffen allgemein eingeführt worden. Anhand des Datengegenstands, der die Informationsbasis für die Analysen und Prüfungen im Rahmen des Lösungsansatzes der vorliegenden Arbeit bildet, wird in diesem Kapitel das Feld der (Daten-) Modellierung näher beleuchtet. Dabei werden grundlegende (technische) Begriffe dargestellt, mit denen die weitere thematische Eingrenzung des Gegenstands der Prüfungen und Analysen auf die Produktmodellierung im folgenden Kapitel besser eingeordnet werden kann. Generell soll ein einheitliches Verständnis der verwendeten Begriffe und Konzepte zu datentechnischen Ansätzen im Bereich des Bauwesens mit ihrem Bezug zur in der Arbeit verfolgten Lösung ermöglicht werden.

Unstrukturierte Daten

Im Zuge der Digitalisierung steigt die Anzahl Daten stetig durch (neu hinzukommende bzw. digitalisierte, vormals analoge) Informationen, die elektronisch generiert, verarbeitet und gehandhabt werden. Eine Vielzahl dabei verwendeter Dokumente (als Träger der Information) folgen dabei keiner formalen Struktur. Zu beachten ist dabei, dass durch die, dem Dokument zugrundeliegende Binärcodierung (Kodierung mittels Codealphabet, z.B. ASCII, UTF-8) eine maschinelle Verwertbarkeit ermöglicht wird, und dies in einem weiteren Sinne auch als (technisch) formalisiert gilt. Da den dabei verwendeten Zeichen(-ketten) jedoch keine Vorgaben zu dem Vorkommen in einer Reihenfolge, wie z.B. eines formal-strukturierten Dokuments unterliegen, gibt es neben der Bedingung „Zeichen muss Element aus Codealphabet sein“ keine weiteren strukturellen Vorgaben. Zusammengefasst im Sinne der Arbeit wird diese Form der elektronischen Information als unstrukturierte Daten bezeichnet.

net. Zu diesen zählen beispielsweise Textdokumente, Bitmap Bilder. Im Rahmen dieser Arbeit wird dabei eine Sichtweise „von außen“ angenommen, das heißt, dass sehr wohl eine Strukturvorgabe für die Inhalte eines unstrukturiert abgelegten Dokuments existieren kann, diese sich jedoch nicht physikalisch auf dem Datenträger abzeichnet, da sie beispielsweise proprietär ist. So werden im engeren Sinne dieser Arbeit zu den unstrukturierten Daten auch – nicht offen gelegte – proprietäre Formate gezählt, da sie aus Sicht eines verteilten Arbeitens ohne das passende, sie unterstützende Programm zur Hand zu haben – unabhängig ob ihnen eine formalisierte Struktur unterliegt oder nicht – als nicht zu bearbeitende Blackbox darstellen.

Im Sinne des verteilten Arbeitens gelten unstrukturierte Daten dabei als nicht interoperabel, obwohl sie von verschiedenen Softwarelösungen interpretiert werden können (Fallon & Palmer, 2007). Dies liegt daran, dass zur letztendlichen Interpretierung bzw. Einordnung des Inhalts immer noch manuelle, d.h. von Menschen vorgenommene Schritte, und somit potentielle Fehlerquellen, Teil ihrer Verarbeitungsprozesse sind. Ein gutes Beispiel ist dabei die noch weitverbreitete Ebenen-basierte (engl. Layer) Festlegung beim Austausch von unstrukturierten CAD-Zeichnungen. Dadurch wird zwar eine Strukturierung der Inhalte angedeutet, legen Bearbeiter*innen jedoch eine Außenwand fälschlicherweise auf die Ebene der Innenwände, führen auf der empfangenden Seite auf Basis der festgelegten Ebenen Struktur erstellte Mengenermittlungen zu falschen Ergebnissen (vgl. ebd.).

Strukturierte Daten

Um die Zielstellung des modellbasierten Arbeitens zu realisieren, wie zum Beispiel eine vollständige Eliminierung von Medienbrüchen im Rahmen kooperativer Datenhandhabung, bedarf es folglich strukturierter Daten. Diese zeichnen sich in einem engeren Sinne durch offen beschriebene Strukturvorgaben aus, die unabhängig von dem verwendeten System, allen beteiligten Seiten bekannt und idealerweise als Standard festgeschrieben sind. Erst auf dieser Basis lassen sich Aspekte wie formal-strukturelle oder fachliche Qualitätssicherung zu den Planungsdaten bzw. absprachelose Austauschszenarien, vollkommen unabhängig von der Form der Datenhaltung und somit fokussiert auf

die spezifischen Gegebenheiten der jeweiligen Situation des verteilten Arbeitens – sachgerecht, zielorientiert und bezüglich übergeordneter Strategie nachhaltig – konzipieren und umsetzen.

Ein generell strukturierten Daten zugrundeliegender Modellansatz wird als **Datenmodell** bezeichnet. Dabei sind dem Begriff zwei Bedeutungen zugeordnet. So bedeutet er zum einen die **Datenmodell-Theorie**, die Ebene der Modellierung (dem Bilden benötigter (Sprach-) Konstrukte) selbst, bestehend aus den Teilen zur strukturellen, integritäts- sowie manipulationsbezogenen Definitionen. Die Form, in der diese Festlegungen beschrieben werden, wird als **Schema** bezeichnet. Auf Basis der in der Theorieebene festgelegten Vorgaben, leitet sich dann in deren konkrete Anwendung (beispielsweise virtuelles Modell eines spezifischen, geplanten bzw. existierenden Gebäudes) die zweite Bedeutung als **Datenmodell-Instanz** ab. Für eine in der Modellsprache (Theorie – bspw. dem objekt-orientierten Ansatz folgend – bestehend aus Klassen, wie einer „Fensterklasse“) verfasste Beschreibungskonzeption ist hiermit also eine entsprechende Ausprägung (bspw. entsprechende Exemplare der Fenster, d.h. Instanzen der Fensterklasse) gemeint (vgl. ebd.).

Neben der Definition und Vereinheitlichung der zu verwendenden Datenobjekte (Ausprägung), schafft die konzeptionelle Theorie des Datenmodells auch die Grundlage zur Maschinenverwertbarkeit der in den Datenobjekten abgelegten Informationen. Auf Basis der Strukturvorgabe kann dazu eine entsprechende Interpretierungslogik (z.B. durch Ableitung einer entsprechenden Klassenbibliothek für eine bestimmte Programmiersprache; vgl. Kap. 2.3.2) für einen sogenannten Parser zum Lesen und Schreiben der Datenobjekte abgeleitet werden.

Standardisierte, strukturierte Daten

Wird die Datenmodell Theorie im Sinne der Harmonisierung über einen spezifischen Anwendungskontext hinaus verwendet, um eine einheitliche definitorische Grundlage eines erweiterten Anwendungsgebietes zu erreichen, wird sie als Standard bezeichnet.

Dabei können zwei Arten von Standards unterschieden werden:

„De facto“ Standard – bspw. DXF: durch die Offenlegung dieses proprietären Formates, können trotz der beim Hersteller verbleibenden Besitz- und Änderungsrechte, Lösungen von Dritten entwickelt werden, die das Format bedienen (Fallon & Palmer, 2007, S. 24)

„De jure“ Standard – werden i. d. R. durch Standardisierungsinitiativen wie ISO, World Wide Web Consortium (W3C), Objekt Management Group (OMG) oder OGC bzw. OpenGIS entwickelt und fortgeschrieben (vgl. Fallon & Palmer, 2007, S. 24).

Vor dem Hintergrund abspracheloser Datenflüsse im Planungsprozess impliziert die in der vorliegenden Arbeit verfolgte Lösung mit offenen Datenmodellstandards, die Planungsinformationen dokumentenbasiert zu adressieren. Bei den im Weiteren dargestellten Prinzipien von Datenstandards, wird nicht weiter auf das Thema von Datenbankschema bzw. -technologie eingegangen, da die Frage der Persistierung der Daten dabei nachgelagert behandelt und im Sinne der Arbeit technologieneutral gehalten ist.

Auszeichnungssprache (Notation, Basissyntax bzw. -grammatik)

Zum Beschreiben einer Datenmodell-Theorie existiert eine Vielzahl von Notationen, mittels derer ein Schema als Grundlage einer entsprechenden Datenmodell-Instanz definiert werden kann. Am Beispiel der unter dem Dach der Standardisierungsorganisation W3C entwickelten Basissyntax *eXtensible Markup Language* (XML) wird im Weiteren die für den Lösungsansatz der Arbeit wichtige einheitliche Notation der Daten (vgl. Abschnitt *Systemmodell* oben) dargestellt. Dabei greifen die XML einen „völlig anderen Zugang zur Modellierung von Daten“ als den „aus dem Relationsmodell und seiner Erweiterung um Objekte“ (Vossen, 2008, S. 337) bekannten Ansatz auf.

Ein grundlegendes Konzept der XML besteht darin, sich selbst beschreibende Daten zu gestalten. Damit wird der Ansatz einer sowohl menschen- wie auch maschinenlesbaren Form von Daten umgesetzt. Im Gegensatz zu dem in Ta-

belle 1 illustrierten Beispiel einer EXPRESS-Schema Definition mit entsprechender Ausprägung¹⁰, ist die Datendarstellung der XML per se aussagekräftiger. Ohne die Kenntnis eines ggf. hinterlegten XML-Schemas (siehe „XSD“ unten) kann der vollständige Bedeutungsgehalt des XML-Datensatzes allein durch die sich selbstbeschreibende Notation von Leser*innen erfasst werden. Ersichtlich wird einerseits allerdings auch, dass die in der Ausprägung mitgeführte Semantik, mit Einbußen von deren kompakten Ausdrucksweisen einhergeht. In XML notierte Daten, wie ifcXML Dokumente, sind daher typischerweise umfangreicher als entsprechend „kryptisch“ notierte Daten, wie beispielsweise in IFC-SPF.

Tabelle 1 Beispiel Schema-basierter bzw. „selbstbeschreibender,“ XML-basierter Datensatz

	Schema	Ausprägung/Instanz
EXPRESS Notation (komma- sepa- riert)	ENTITY Adresse Straßennamen : string Hausnummer : integer Postleitzahl : integer END_ENTITY	Adresse(Muster- straße,17,10800)
XML No- tation	<xs:element name="Adresse"> <xs:complexType> <xs:sequence> <xs:element name="Straßennamen" type="xs:string"/> <xs:element name="Hausnummer" type="xs:int"/> <xs:element name="Post- leitzahl" type="xs:int"/> </xs:complexType> </xs:element>	<Adresse> <Straßennamen>Muster- straße </Straßennamen> <Hausnummer>17 </Hausnummer> <Postleitzahl>10800 </Postleitzahl> </Adresse>

¹⁰ technisch werden die entsprechenden Instanzdokumente STEP Physical File (SPF) beispielsweise für das IFC Produktmodell IFC-SPF genannt

	<pre></xs:sequence> </xs:complexType> </xs:element></pre>	
--	---	--

Der Notationsansatz der XML eignet sich sehr gut als formale Integrationsbasis in **Datenintegrationsszenarien**, in denen semistrukturierte¹¹ Daten aus verschiedenen Quellen zusammengeführt werden sollen. Im Zuge seiner starken Verbreitung wird XML zudem extensiv als **universales Datenaustauschformat** eingesetzt bzw. als **Ein- und Ausgangsschnittstelle** bei dem Datenaustausch zwischen verschiedenen Systemen genutzt. Insbesondere wird die XML als Basissyntax zum **Definieren neuer Sprachen** für unterschiedlichste Anwendungskontexte verwendet (ebd.). Eine den flexiblen Erweiterungsmechanismus der XML anwendende Sprachdefinition kann hierbei auf den im Folgenden skizzierten XML-Sprachelementen ansetzen.

Die XML baut auf der zugrundeliegenden Standard Generalized Markup Language (SGML) auf. Diese definiert zur Auszeichnung von Dokumenten als Grundkonzept die Beschreibung von Elementen mittels sogenannter *Tags*. Dabei wird der Inhalt eines Elements durch ein in spitzen Klammern gefasstes öffnendes *Tag* (`<Elementname>`) am Anfang sowie ein entsprechendes schließendes *Tag* (`</Elementname>`) am Ende der Elementbeschreibung eingeschlossen. Durch eine verkürzte Form (`<Elementname/>`) lassen sich darüber hinaus inhaltsleere Elemente auszeichnen. Die Einhaltung dieser Elementauszeichnung auf jeder Hierarchiestufe beschreibt zugleich die wichtigste Vorgabe beim Verschachteln beliebiger, weiterer Elemente. Ein Dokument kann dabei dann gebildet werden, indem alle Elemente in ein Wurzelement verschachtelt sind. Werden dabei alle beinhalteten Elemente korrekt – mit öffnenden und schließenden bzw. leeren *Tags* – abgebildet, handelt es sich um ein wohlgeformtes Dokument. Eine weitere Abbildungskonzeption der XML-Elemente sieht neben dem Verschachteln von weiteren Kindselementen in beliebiger Tiefe die Vergabe von Attributen innerhalb vom öffnenden (bzw. im

¹¹ „Daten, die gleiche oder ähnliche Sachverhalte beschreiben (z.B. Bücher), unterschiedliche Auswahlen an Attributen besitzen können.“ (Vossen, 2008, S. 337)

leeren) *Tag* vor. In der Form „Name ist gleich Wert“ (z.B. Attributname=“Attributwert“) können damit beliebige primitive Werttypen (Zeichenketten, numerische Werte) dem Element als XML-Attributwerte zugeordnet werden (vgl. ebd.).

Dieses syntaktische (Grund-) Prinzip der SGML wird bei der Anwendungssprache Hypertext Markup Language (HTML - Schema) zum Beschreiben von sowohl Inhalt als auch dem Layout (die Art wie der Inhalt – mehr oder weniger plattform-unabhängig – im Webbrowser dargestellt werden soll) beim Erstellen von HTML-Dokumenten (Instanzen) genutzt. Bei XML hingegen ist das Layout nicht Teil der syntaktischen Information, die den Aufbau und die Struktur eines Dokuments beschreibt (vgl. Vossen, 2008, S. 338). Jedoch wird, gemäß des XML-Prinzips für spezielle Anwendungen eine entsprechende Anwendungssprache zu definieren, dieser Anwendungsfall (Inhalt+Layout) von XHTML bedient.

Ein weiteres wichtiges Konzept zur Gestaltung von XML-Dokumenten stellt die Möglichkeit dar, einem Dokument verschiedene Namensräume (NR, engl. namespace – NS) zu definieren. Damit lassen sich in einem Dokument gleichlautende Elementbezeichnungen durch Voranstellen eines entsprechenden Platzhalter *Präfix* für den spezifischen Namensraum (bspw. <NR-präfix:Elementname/>) verschiedenen Ursprungskontexten zuordnen. Der Hauptnamensraum eines Dokuments wird dabei mit dem XML-Attribut „xmlns“ im Wurzelement des Dokuments angegeben und gilt für alle nicht mit dem Präfix eines spezifischen Namensraumes ausgezeichneten Kindselemente. Allgemeine (Sub-)Namensräume können einerseits (dokumentweit) im Wurzelement als mit entsprechendem Platzhalter Suffix versehenen xmlns-Attribut definiert werden (bspw. xmlns:Namensraumkürzel=“Namensraumadresse“). Andererseits kann der Namensraum abschnittsweise durch Vergabe entsprechender (Sub-) Namensräume im xmlns-Attribut auch im spezifischen Element selbst (d.h. für alle seine Kindselemente gültig) definiert werden. Generell können hinter der Angabe von Namensräumen mittels einer Unified Resource Identifier (URI) die entsprechend dem Namensraum definierten Elementnamen vorgehalten werden.

Anwendungssprachdefinition (Schemasprachen)

Auf diesen exemplifizierten, syntaktischen Grundbausteinen, lassen sich beliebig strukturierte Dokumente (Datenmodell Instanzen) erstellen, die generell als semistrukturierte Daten klassiert werden. Um mit der XML-Basissyntax wohlgeformte strukturierte Daten erstellen zu können, bedarf es, neben dem oben angeführten Namensraumkonzept, einer abstrakten (d.h. von außerhalb auf die Basissyntax bezugnehmenden) Beschreibung der Struktur der Elemente im Sinne eines Schemas.

Van der Vlist (2002) unterscheidet diesbezügliche Ansätze von Schemasprachen für XML in die folgenden drei Kategorien:

- 1) Regelbasierte XML Schemasprachen – mittels einer Kollektion von Regeln können alle möglichen Einschränkungen (engl. constraints) beschrieben werden. Dabei werden diese als „nachrangige“ (eng. „lower-level“) Sprachen betrachtet, auf welchen andere Schemasprachen aufbauen können.
- 2) Grammatikbasierte XML Schemasprachen – beschreiben die Struktur (d.h. die erwartete Topologie) des Instanz-Dokuments. Können dabei mit nicht-strukturellen Regeln kombiniert werden, um eine ganzheitlichere Ausdruckskraft zu erreichen.
- 3) Objekt-orientierte XML Schemasprachen – versuchen, möglichst nah an den Ansatz des objekt-orientierten Designs (Klassen, Methoden, Relationen) angelehnt, die Strukturen eines Instanz-Dokuments zu beschreiben. Einen wichtigen Stellenwert wird dabei dem (Objekt-) Typ in diesem Ansatz beigemessen, um Ähnlichkeiten unter (davon ableitenden) Objekten zu ermöglichen. Zudem beinhalten diese Ansätze Mechanismen zur Ableitung und Substitution und Grundlagen für die sogenannte Post Schema Validation Infoset (PSVI), welche eine Schlüsselfunktion zur Kommunikation zwischen dem Schema-Prozessor und den Anwendungen darstellt.

Im Rahmen von Standardisierungen, vorangetrieben von den Entwicklungen im Bereich des Internets und zunehmend zum thematischen Anwendungskontext der Arbeit, insbesondere auch im Bereich der Stadtplanung angewendet, existieren zahlreiche Anwendungssprachen auf Basis des Typdeklarationskonzepts **XML Schema Definition (XSD)**, einer XML-basierten, objektorientierten Schemasprache. Ebenso existiert hierzu eine Vielzahl an Literatur (verwiesen sei hierbei auf Van der Vlist, 2002). Im Rahmen der Arbeit wird bezüglich der Schemasprache XSD beispielhaft auf die von der Standardisierungsinitiative Open Geospatial Consortium (OGC) entwickelten Geographic Markup Language (GML) als Grundstruktur (Basisschema) für darauf aufbauende spezifische, geodatenbasierte Anwendungssprachen eingegangen (siehe Kap. 2.2.3). Zudem werden weitere Ansätze aus der XML-Technologiefamilie im Teil Modelldatenverarbeitung der Grundlagen vorgestellt (vgl. Kap. 2.3.2), zu denen folgendes von Vossen (2008) aufgegriffenes und ergänztes Schaubild eine grobe Übersicht ermöglicht (Abbildung 8).

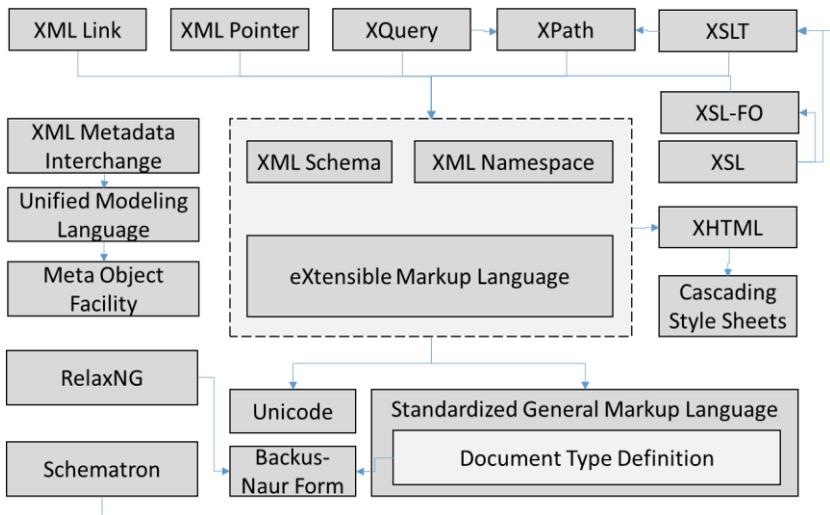


Abbildung 8 XML-Sprachlandschaft (erweitert nach Vossen, 2008)

2.2.3 Produktmodellierung

Für die Abbildung der Planungsgegenstände haben sich sowohl in der Stadt- als auch in der Gebäudeplanung standardisierte Produktmodelle etabliert. Vor dem Hintergrund einer objektorientierten Repräsentation der behandelten Gegenstände *Stadt* bzw. *Bauwerk* ermöglichen die Modellstandards jeweils als Objekt die beinhaltenen Einzelteile sowohl geometrisch wie auch semantisch zu beschreiben. Hierfür können den in einer Klasse im Schema der Standards gekapselten Modellentitäten neben Geometriebeschreibungskonzepten für visuelle Darstellungen weitere alphanumerische Beschreibungskonzepte zugeordnet werden.

Damit ermöglichen die Standards begleitend zur fortschreitenden Digitalisierung in Staat, Wirtschaft und Gesellschaft eine geforderte sukzessive, bedarfsgerechte Bereitstellung an Fachinformationsmodellen, die neben reinen geometrischen bzw. raumbezogenen Daten auch die Konnotation semantischer Eigenschaften bis hin zu thematischen Ontologien ermöglichen. Um auf Seiten der Stadtplanung die Interoperabilität zunächst bezüglich der Abbildung von geografischer Information heterogen entwickelter Geographic Information Systems (GIS) langfristig zu verbessern, wurde erstmals in den Jahren 2002 bis 2004 die Normenreihe ISO 191xx zur harmonischen Repräsentation und Modellierung von Geoinformationen spezifiziert (ISO 2005, 2006, 2007). Mit der Geographic Markup Language (GML) wird darin, basierend auf der XML-Notation, u.a. eine Sprache zum Beschreibbarmachen geografischer Gebilde normiert. Die GML ist dabei in einem XSD Schema definiert und bildet ein Basisschema auf dem verschiedene (fachliche) Anwendungsschemata aufbauen (vgl. Abbildung 9). Als Produktmodell stellt dabei der CityGML Standard für die Stadtplanung zur Abbildung von Artefakten einer Stadt entsprechende Elemente zur Verfügung (vgl. Brüggemann u. a., 2020). Ein im Standard vorgesehener Erweiterungsmechanismus ermöglicht es, mittels sogenannter Application Domain Extensions (ADE) eigene Abbildungskonzepte mit Bezug zu den CityGML bedarfsorientiert in einem ergänzenden Schema zu definieren. So werden beispielsweise für den Bereich der energetischen Stadtsimulation neben entsprechenden Forschungsarbeiten (vgl. Kap. 8.1)

auch von offizieller Seite der Standardisierungsorganisation OGC solche ADE Entwicklungen vorangetrieben.

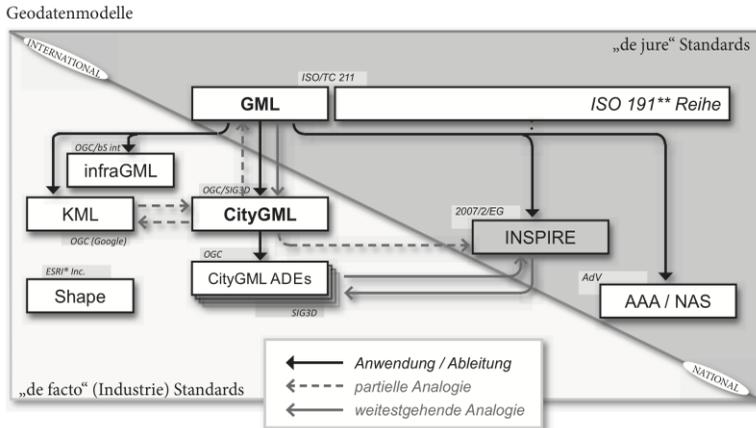


Abbildung 9 Geodatenmodellstandards (Quelle: Brüggemann u. a., 2020, Abb. 10.1)

Somit lassen sich die in verschiedenen Disziplinen der Stadtplanung benötigten Fachinformationen (z.B. Lärmkartierung, usw.) im Rahmen eines zentralen Produktmodells zusammenstellen, um eine ganzheitliche Sicht auf den Planungsgegenstand generieren zu können. Viele Kommunen haben inzwischen für ihre kommunalen Informationen CityGML Stadtmodelle aufgebaut (vgl. sowie ein guter Überblick Biljecki u. a., 2018).

Eine wesentliche Darstellungssystematik in diesem Stadtmodellstandard stellen verschiedene darin abbildbare Detaillierungsgrade (sogenannte Level of Detail – LOD) dar. Damit ist es möglich, Informationen zu einem Bauwerk je nach Verfügbarkeit der Information in vier verschiedenen Detailtiefen zu beschreiben. Beginnend bei einem in der Geometrie als Umrissfläche zweidimensional repräsentierten LOD0 bis zu einem auch die Innenräume der Stadtmodellgebäude beschreibarmachenden LOD4. Diese detaillierteste Stufe der CityGML wird in der Fachdiskussion auch als Brückenschlag zu den Planungs- und Produktmodellstandards auf Gebäudeebene diskutiert (vgl. ebd.)

Auf dieser Ebene der Objektplanung hat sich im Rahmen einer kooperativen modellbasierten Arbeit der Produktmodellstandard Industry Foundation Classes (IFC) etabliert. Der internationale Standard zur Beschreibung von Bauwerken (sowie in seiner neuesten Version 4.2 auch von Infrastruktur, wie Straßen) wird von der Standardisierungsorganisation buildingSMART (weiter-)entwi-

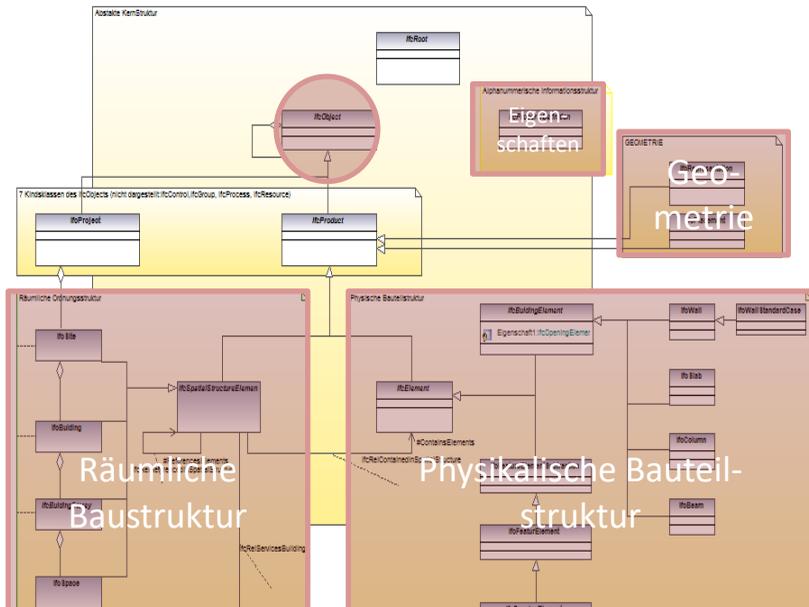


Abbildung 10 Schematische Struktur ausgewählter IFC Schemateile

ckelt. Unter der ISO 16739-1 (2018) ist das Datenschema sowohl auf der Syntaxbasis EXPRESS als Anwendungsschema zum STEP Format als auch auf der Syntaxbasis XML als XSD Schema normiert. Abbildung 10 zeigt in einer schematischen Darstellung die wesentlichen Teile des IFC Schemas. Im Kern des Schemas stehen verschiedene grundlegende Beschreibungskonzepte, die objektorientiert auf einem Wurzelement *IfcObject* (vgl. Kreis in Abbildung 10) in unterschiedlichen hierarchischen Strängen aufbauen. Als wesentliche Grundkonzepte können dabei den in den beiden – räumlichen sowie physika-

lischen – Hauptsträngen untergliederten Bau-(werks-) Elementen über die gemeinsame Basisklasse sowohl geometrische Repräsentation als auch beliebige Eigenschaften zugeordnet werden. Dieses flexible Prinzip erlaubt zum einen auf Seite der Bauwerksstruktur verschiedene bedarfsgerechte Abbildungen der Bauwerkstopologie, wie beispielsweise: Baugrund – Bauwerk – Stockwerk – Raum. Auf der anderen Seite können hiervon losgelöst, die Bauteile sowie ihre Zusammenstellungen aus z.B. einzelnen Komponenten beliebig strukturiert und der räumlichen Bauwerksstruktur zugeordnet werden. Neben Weiterentwicklungen in der geometrischen Repräsentierungsmöglichkeiten, wie beispielsweise durch parametrisierte Beschreibungskonzepte, wird der IFC Standard von buildingSMART auch in der Abbildungsmöglichkeit fachlicher Informationen stetig an die Bedarfe der Praxis angepasst. Im Rahmen der Erweiterung bezüglich der Abbildung von Informationen zur Nachhaltigkeit eines Bauwerkes beteiligt sich das Fachgebiet unter Mitwirkung des Autors aktiv in der Fachgruppe *BIM und Nachhaltigkeit* (vgl. Kap. 8.1).

2.3 Modelldatenverarbeitung

Zu den im Kapitel 2.2 beschriebene Daten werden in diesem Kapitel die für den Rahmen der vorliegenden Arbeit wichtigen technischen Aspekte der Datenverarbeitung dargestellt. Dabei wird die ausgehend von den informatischen Grundlagen zu Programmiersprache bzw. Entwurfsmuster (vgl. Kap. 2.3.1) näher auf die standardisierte Handhabung von Daten mittels entsprechender Technologien eingegangen (vgl. Kap. 2.3.2). Abschließend werden die für die Dissertation bedeutsamen methodischen Aspekte der Datenhandhabung zum Bereich des Datenmanagements zusammengestellt. Prinzipiell geht es dabei aber nicht um eine umfängliche Abhandlung der angeschnittenen informatischen Themen, sondern um ein Klären der aufgegriffenen technischen Begriffe im Sinne der Arbeit.

2.3.1 Objektorientierte Programmierung

Zur Umsetzung von Desktop-Softwareanwendungen bietet es sich, an eine der *hohen* Programmiersprachen zu verwenden. Im Gegensatz zur Verwendung von funktionalen (Skript-) Sprachen ermöglichen die weitverbreiteten, etablierten objektorientierten Hochsprachen, wie .NET C#, Oracle JAVA oder Python, inhärente Mittel zur Gestaltung einer robusten, skalierbaren Softwarearchitektur. Bei der Softwareentwicklung unterstützen sie das wesentliche Prinzip des objektorientierten Entwicklungsparadigmas: dem Kapseln von einzelnen Belangen jeweils in eigenständigen Objekten. Indem im Quelltext entsprechende Klassen definiert werden, können die Objekte mit Eigenschaften und Methoden als Grundbausteine einer designierten Softwarearchitektur ausgeprägt werden. Dieses deklarative Prinzip bildet ein wesentliches Merkmal der Hochsprachen, die – bevor der mit ihnen ausgedrückte Quelltext ausgeführt werden kann – interpretiert werden muss. Anders als direkt mittels einer entsprechenden Engine ausgeführte Ausdrücke von Skriptsprachen, wird der Quelltext mittels der Kompilation in eine sogenannte Intermediate Language übersetzt. Dadurch zusammengestellte Programmdateien bzw. Klassenbibliotheken können, falls eine entsprechende Laufzeit im Betriebssystem integriert ist, direkt ausgeführt werden, wie beispielsweise eine in Microsofts Windows Betriebssystem bezüglich der von C# kompilierten Artefakte. Eine Besonderheit dieser interpretierten Programmiersprachen liegt in der Möglichkeit, mittels sogenannter *Reflection*¹² ohne das vorherige deklarative Definieren im Quelltext direkt auf die Operationen der IL (also dem kompilierten Code) zuzugreifen und diesen zu verändern. In .NET wurde mit der *DynamicObject*¹³ Komponente neben den ansonsten typischeren Eigenschaften von C# ein Mechanismus eines dynamischen Polymorphismus ermöglicht. An einer Instanz dieser Klasse (bzw. einer Ableitungsklasse davon) kann dabei beispielsweise, ohne vorheriges Deklarieren einer Klasseneigenschaft im Quelltext, dynamisch zur Laufzeit der Software – allein durch entsprechenden Aufruf – eine

¹² Siehe <https://docs.microsoft.com/en-us/dotnet/framework/reflection-and-codedom/reflection>. Zugriff: 15.03.2018

¹³ Siehe <https://docs.microsoft.com/en-us/dotnet/api/system.dynamic.dynamicobject?redirectedfrom=MSDN&view=netframework-4.8>. Zugriff: 15.03.2018

beliebige Eigenschaft bzw. Methoden hinzugefügt werden. Diese Möglichkeit, die beispielsweise eine Grundeigenschaft von Objekten der Programmiersprache Python darstellt, ermöglicht zwar flexible Gestaltungen des Objektverhaltens ähnlich einer funktionalen Sprache, ist jedoch auch schwieriger zu lesen bzw. steigert die Komplexität des Codes. Als besser zu wartender bzw. wiederzuverwendender Code gilt daher ein (typischesicheres) programmiertes Verhalten eines Objekts durch entsprechendes Deklarieren von Klasseeigenschaften bzw. -methoden.

Als eine gute methodische Entwicklungsgrundlage bietet es sich prinzipiell an der Softwareentwicklung – gemäß dem designierten Zweck der Software – eine Architektur zu unterlegen, die die Klassen, respektive Objekte beim Programmieren grundlegend in ihrer Beziehung zueinander gliedert. Einer Desktop-Softwareanwendung wird typischerweise eine Schichtenarchitektur zur Entkopplung der Elemente der Benutzeroberfläche von denen der Geschäftslogik- bzw. Datenschicht unterlegt. Damit kann dem im einzelnen verfolgten Trennen der Belange ein übergeordneter Rahmen gegeben werden, der generell die Skalierbarkeit des entwickelten Quelltextes erhöhen kann. In dem darüber hinaus verschiedene in der Softwareentwicklung entstandenen Entwurfsmuster beim Umsetzen der Software einbezogen werden, können diese Eigenschaften, wie Skalierbarkeit bzw. Wiederverwendbarkeit weiter im Rahmen der Architektur geschärft werden. Ein wichtiges Prinzip zur generellen Entkopplung der Abhängigkeit von Objekten im Softwareentwurf bietet dabei das Einbinden von Schnittstellen. Dabei werden anstelle einer direkten Bindung zwischen zwei Objekten mittels ihrer jeweiligen (dann gezwungenermaßen) „öffentlich“ sichtbaren Eigenschaften, wie bei einer Fassade ein Stellvertreter (Schnittstellenobjekt) mit den benötigten Bindungseigenschaften definiert. Ohne zu wissen, welches – tatsächlich die Schnittstellenklasse implementierende – Objekt dahintersteht, kann eine durch die in der Schnittstelle bekannt gemachten Eigenschaften aufgerufen werden. Im sogenannten *Repository Pattern* (vgl. Fowler, 2003) wird dieses Prinzip bezüglich eines Datenzugangs verschiedenartig persistierter Daten weiter verfeinert (vgl. Abbildung 19). Mit diesem Entwurfsmuster können prinzipiell für verschiedene Persistenztechnologien, wie die exemplifizierten Dokumente bzw. datenbankbasier-

ten Zugänge mittels eines Schnittstellenobjekts bedient werden. Bei der Entwicklung im Rahmen der vorliegenden Arbeit wurde mittels Aufgreifens dieser Art von Entwurfsmuster versucht, die entwickelte Software möglichst skalierbar zu gestalten (vgl. Kap. 5).

Programmieren, Arbeitsstrategien & -hilfen

Zum Programmieren bietet es sich an, eine Programmierumgebung zu verwenden. Obwohl der Quelltext prinzipiell auch in einem reinen Textverarbeitungsprogramm erstellt werden kann, liegen die Vorteile einer Umgebung in der begleitenden Hilfestellung, wie den Möglichkeiten zum direkten Ausführen des Codes bzw. zum Setzen sogenannter Sprungmarken, bei denen dann die Ausführung angehalten und die entsprechende Stelle im Quelltext angezeigt wird (vgl. Debug). Diese Rückkopplung zwischen Programmieren und Ausführen wird in einer sogenannten REPL – Read, Eval, Print and Loop – Programmierumgebung auf ein einfaches Prinzip reduziert. Diese ermöglicht aufgrund des direkten Feedbacks eine selbsterklärende Programmierung und ist daher insbesondere für Einsteiger*innen bzw. für einen Ansatz zum Selbsterlernen geeignet. Hierzu werden synonym die Begriffe *interactive toplevel* oder (z.B. Python *Language Shell*) verwendet.

2.3.2 Standardisierte Datenhandhabung

Data Binding

Bezüglich des bereits oben zum Entwurfsmuster *Repository Pattern* dargestellten Datenzugang werden im Weiteren verschiedene standardisierte Daten Bindungsansätze am Bindungsbeispiel XSD basierter XML Instanzdokumente exemplifiziert. So stellt die Standardisierungsorganisation mit dem Domain Object Model (DOM) Standard eine normierte Art des Implementierens einer „internen“ Repräsentation der Inhalte eines XML Dokuments mit den Klassenobjekten einer Programmiersprache bereit. Beispiele für entsprechende Implementierungen des Standards finden sich als inhärente Komponente

*org.w3c.dom*¹⁴ in JAVA. Mit der Komponente ist es möglich, die normierten Eigenschaften und Methoden im Sinne einer API im Programmiercode zu verwenden, um sie als Objektgraf komplett in den Systempeicher geladenen Inhalt des XML Dokuments aufzurufen. Da dies bei großen Dokumenten zu Performanceproblemen führen kann, wird ein Prinzip ohne das vollständige Einlesen in den Systempeicher im sogenannten Simple API to XML (SAX, vgl. z.B. Implementierung als inhärente JAVA Komponente *org.xml.sax*¹⁵) als „de facto“ Standard normiert. Mittels standardisierter Verben einer Application Programming Interface (API) wird dabei auf Basis eines als geöffneten Datenstrom-Objekts direkt auf die im Verzeichnis liegende Datei zugegriffen. Dieser Ansatz ist zwar speicherfreundlich, d.h. selbst große Dokumente können mit wenig Speicherkapazität ausgelesen werden, allerdings sind dadurch die Abfragemöglichkeiten an die Dateninhalte komplizierter zu gestalten. Diese können im Gegensatz zum DOM Ansatz nicht deklarativ formuliert werden. Da dabei jeweils aufgrund des geöffneten Datenstroms rekursiv alle Inhalte iteriert werden, sind die Abfragevorgänge zu den Modellinhalten in SAX prinzipiell komplizierter zu formulieren. Im sogenannten XML Data Binding wird eine dedizierter als das generische DOM Prinzip an die im Schema definierten modellspezifischen Elementnamen orientierte Lösung realisiert. Hierbei werden zu jedem Element des XSD Schema mittels Klassengeneratoren anstelle von generischen Klassen, wie beispielsweise einer *XElement* Klasse beim an den DOM Standard angelehnten .NET *LINQ to XML*¹⁶ Ansatz, eine Klasse mit dem Elementnamen generiert. Das Produkt ist dann eine Klassenbibliothek, die jedem Schemaelement eine Klasse zuordnet, und damit das Implementieren eines Vorgangs zum (De-) Serialisieren der im XML-Dokument binärkodierte Daten ermöglicht. Hierbei wird die jeweils zu einem Element gehörende Zeichenabfolge im Datenstrom gefiltert und mittels einer Deserialisierungsmethode in einem Objekt der entsprechenden Klasse instanziiert. Die Klassenbibliothek stellt somit die Grundlage zum (bidirektionalen) Binden der Inhalte eines XML-Dokuments an Objekte einer Programmiersprache. Für die

¹⁴ <https://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html>

¹⁵ <https://docs.oracle.com/javase/7/docs/api/org/xml/sax/package-summary.html>

¹⁶ <https://docs.microsoft.com/en-us/dotnet/api/system.xml.linq?view=netframework-4.8>

verschiedenen Programmiersprachen existieren verschiedene Klassengeneratoren für das XSD Schema, beispielsweise XSD2CODE¹⁷ für C#, generateDS (Kuhlman, 2020) oder PyXB¹⁸ für Python. Da die Modellstrukturen trotz der vorgegebenen Monohierarchie eines XSD durch verschiedene Schemakopplungen (vgl. *include* bzw. *import* Attribute¹⁹ eines XSD Schema) sehr komplex werden können, sind gegebenenfalls bei den generierten Klassenbibliotheken manuelle Anpassungen zur affinen Abbildung des Schemas erforderlich. Bezüglich eines komplexen Schemas, wie dem über 15 Einzelschemas beinhaltenden Stadtmodellstandard CityGML, haben sich Projekte zur Entwicklung von modellspezifischen Klassenbibliotheken gebildet. XML-Dokumente des CityGML Standards können beispielsweise mittels der offen verfügbaren Parserkomponente JAVA-basierten CityGML4J eingelesen und geschrieben werden. Auch für das über 900 Klassen zählende IFC Produktmodell stehen verschiedene vorgefertigte Komponenten als (plattformunabhängige) Parser zur Nutzung in der eigenen Softwareentwicklung frei zur Verfügung: *IFC Engine DLL*²⁰, *IFC Framework*²¹ oder *IfcOpenShell*²².

Spezifikations- und Abfragesprachstandards

Ein mittels der oben beschriebenen Parserkomponenten realisiertes Binden der Inhalte eines XML-Dokumentes an die Objekte einer Programmiersprache, bildet die Grundlage des Zugangs zu persistierten Daten, auf der verschiedene Standards angesetzt werden können, beispielsweise die Dateninhalte nach bestimmten Sachverhalten abzufragen. Mit dieser Art von „generischen“ Abfragesprachstandards (vgl. engl. *Query Language* – QL), wie z.B. die *Structured Query Language* (SQL) zum Abfragen und Manipulieren von Daten in einer Datenbank, können technologie- und datenmodellstandardübergreifende Lösungen der Datenhandhabung realisiert werden. In Abbildung 8 sind die von der W3C im Rahmen der XML-basierten Datenhandhabung standardisierten

¹⁷ <https://www.xsd2code.com/>

¹⁸ <https://pypi.org/project/PyXB/>

¹⁹ https://www.w3schools.com/xml/el_import.asp

²⁰ <http://rdf.bg/product-list/ifc-engine>

²¹ <http://www.apstex.com/>

²² <http://www.ifcopenshell.org/>

Abfragesprachen dargestellt, dabei basiert der XQUERY Standard bezüglich des Traversierens der Inhalte auf dem XPath Standard. Darüber hinaus ermöglichen Fazilitäten in den Sprachstandards, wie die *Update*²³ Komponente in XQuery, auch das Bearbeiten der Modellinhalte. Zum Einsatz kommt dieser W3C Standard bei den speziellen XML-nativen Datenbanken, wie beispielsweise die Open Source Datenbank *Berkeley DB XML*²⁴ von Oracle, für einen Datenzugang implementieren diese DB-Technologien dabei anstelle eines SQL den XQuery Standard.

Neben dem auf alle XML-basierten Daten anwendbaren XQuery Standard stellen modellspezifische Lösungen, ähnlich zum Unterschied zwischen den oben beschriebenen DOM zu einer modellspezifischen Klassenbibliothek, bereits auf die Schemaelemente bzw. -strukturen von Produktmodellstandards angepasste Abfragesprachen bereit. Mit der disziplin-spezifischen Spezifikationsprache BIMQL kann z.B. in vereinfachter Weise auf die im Schema mehrfach verschachtelten Strukturen der IFC zugegriffen werden (Mazairac & Beetz, 2013). Eine Modellkomplexität wird dabei innerhalb der funktionalen Sprache durch bereits inhärente Traversierungsvorgänge verdeckt; und damit die Anzahl benötigter expliziter rekursiver Ausdrücke verringern. So lässt sich eine über eine eigenständige objektifizierte Assoziationsklasse verknüpfte Eigenschaftsobjekt an einem Bauelementobjekt über die sprachinterne Traversierung allein durch einen Methodenaufruf bewerkstelligen, der in XQuery ausgedrückt mindestens drei Ausdrücke erfordern würde: (1) Finde assoziierte Relationsobjekt, das auf das gesuchte Bauelementobjekt verweist; (2) Lese im Assoziationsende des Relationsobjekts das Verweisobjekt zum Eigenschaftsobjekt aus; (3) Folge dem Verweisobjekt zum Auffinden des gesuchten Eigenschaftsobjekts. Somit ermöglichen die modellspezifischen Spezifikationsprachen eine vereinfachte Handhabung der in den Modellen abgelegten Information, indem die teilweise komplexen modelltechnischen formal-strukturellen Aspekte einer Modellabfrage mit verständlichen Begriffen verdeckt werden.

²³ <https://www.w3.org/TR/xquery-update-10/>

²⁴ vgl. <https://www.oracle.com/technical-resources/articles/database/berkeleydb-xml-xquery.html>

Der Anwendungsgegenstand der bisher dargestellten Abfragesprachenstandards sind die Dateninstanzen der Datenmodellschemata, wie beispielsweise ein XML-Instanzdokument eines IFC-basierten Gebäudemodells. Sie werden daher vornehmlich auf Ebene der Endanwender*innen beim Bearbeiten der konkreten Modellinstanzen angewendet.

Formale Spezifikationsprache Object Constraint Language (OCL)

Die formale Spezifikationsprache der OCL ermöglicht als Teil der in der objektorientierten Softwareentwicklung verwendeten grafischen Notation Unified Modeling Language (UML) bereits in der konzeptionellen Modellierung ein Abgrenzen möglicher Objektausprägungen. Beispielsweise können damit Wertausprägungen von Eigenschaften zu den im Rahmen eines UML-Klassendiagramm definierten Klassen eingeschränkt werden. Der von der Standardisierungsorganisation Object Management Group (OMG) herausgebrachte Standard wird auch in der Programmiersprache JAVA als OCL/P umgesetzt. Dabei wird die Syntax zwar an die der Zielplattform angepasst (vgl. Rumpel, 2011), ermöglicht jedoch gemäß dem Zweck als Abfragesprache, in kompakter Weise Objektanfragen auszudrücken. Auch die dem .NET Framework seit der Version 3.5 inhärente Komponente Language Integrated Query (LINQ) lehnt sich an den OCL Standard an, und ermöglicht in der Programmiersprache C# die dem Standard eigenen Iterationsausdrücke zur Abfrage bestimmter Sachverhalte an den C#-Objekten. Sogenannte Lambda Ausdrücke ermöglichen die sprachinternen kompakten Iterationsanweisungen.

OCL Iterations-Ausdrücke
<pre>let seq = Sequence { 1, 2, 3, 4, 5, 6, 7, 8, 9 }, evenSeq = seq->select(i i mod 2 = 0), allEven = seq->forall(i i mod 2 = 0), existsEven = seq->exists(i i mod 2 = 0), sum = seq->sum()</pre>
Äquivalente in C# integrierte Iterations-Ausdrücke
<pre>var seq = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9 }; var evenSeq = seq.Where(i => i % 2 == 0);</pre>

```
var allEven = seq.All( i => i % 2 == 0 );  
var existsEven = seq.Exists( i => i % 2 == 0 );  
var sum = seq.Sum();
```

Tabelle 2 Vergleich zwischen OCL und C# LINQ Ausdrücken (Quelle: Akehurst u. a., 2008)

2.3.3 Datenqualitätsmanagement

Bei der Unterstützung kooperativer modellbasierter Arbeit spielt die Qualität des verteilt bearbeiteten (Planungs-) Gegenstands eine entscheidende Rolle für den Gesamtprozess. Vor dem Hintergrund von Informationen des Baubereiches bezieht sich die vorliegende Arbeit auf folgende Kontexte einer Qualitätssicherung:

- Integrierte Datenbasis
- Kooperativer Datenaustausch
- Projekt-übergreifende Verwendung der Projektdatenbasis (z.B. *Downstream* Anwendung Planungsdaten im Facility Management)
- Einbindung externer Kataloge
- Regulatorische Ebene bspw. Nachweiswesen

In diesem Kapitel werden hierzu entlang allgemeingültiger methodischer Ebenen der Datenhandhabung – Validieren, Prüfen, Analysieren [sowie Simulation/Prognose] – für das Verständnis der Arbeit wesentliche handlungstechnische Aspekte der Datenqualitätssicherung dargestellt. Dabei steht nicht ein umfängliches Aufbereiten der im Bereich der Datenqualität vorhandenen Literatur im Vordergrund, sondern das Zusammenstellen der als Grundlage des Lösungsansatzes einbezogenen Aspekte. Für eine generelle Übersicht zur Thematik sei auf die im Weiteren angeschnittenen Arbeiten von u.a. Helferts (2002) und Hildebrand et. al. (2015) verwiesen. Die obige Untergliederung in methodische Ebenen sind nicht trennscharf voneinander abgrenzbar, vielmehr

soll durch die gewählte Einteilung ein besserer Überblick der Sachverhalte bezüglich der im Lösungsansatz eingeordneten Bereiche regelbasierter Behandlung von modellbasierten Planungsgegenständen ermöglicht werden. Daher werden methodische Aspekte bezüglich Analysen, die im weiteren Sinn freilich auch den beiden anderen methodischen Ebenen einer Prüfung bzw. Simulation/Prognose zugrunde liegt, beispielsweise im engeren Sinn bezüglich benötigter Ableitungen spezieller *Analysemodelle* abgegrenzt. So kann z.B. ein Zwischenschritt im Rahmen zu überprüfender bzw. zu analysierender (mehrskaliger) Sachverhalte erforderlich sein, aus einem Stadtmodell in Kombination mit einem detaillierten Bauwerksmodell eine Objekt-Menge als Analysemodell abzuleiten.

Die gewählte Strukturierung dient somit als Hilfskonstrukt zum Zusammenstellen von den im disziplinübergreifenden Feld des Datenqualitätsmanagements vorzufindenden Ansätze mit ihren Kriterien bezüglich genereller Datenqualität. Zunächst im Feld neutral formulierte Anforderungen an einen allgemeinen Umgang mit dem Thema Datenqualität werden hierbei in die für diese Arbeit wichtigen Teilbereiche zur Behandlung von Planungsgegenständen eingeordnet. Dabei wird – wie oben beschrieben – kein Anspruch auf Vollständigkeit gestellt. Vielmehr sollen die im Baubereich diskutierten –und dabei oftmals zugunsten fokussierter Lösungen generalisierte (theoretische) Fundierungen der betrachteten Aspekte vernachlässigende – fachspezifischen Ansätze des Baubereichs (welche im Kapitel 2.1.5 vorgestellt werden) besser im Rahmen der übergeordneten Theoriebildung zum Datenqualitätsmanagement verortet werden können. Mit folgendem kurzen Überblick soll hierfür ein grober Rahmen zu Entwicklungen im Bereich des Qualitätsmanagements gegeben werden.

Skizzierte Historische Entwicklung Qualitätsmanagement

Die folgenden Ausführungen wurden von Helferts (2002) aus der Literatur zusammengestellten historischen Entwicklungsschritten in der Theoriebildung des Bereiches Qualität entnommen. Die aktuelle Theorie der „Total Quality Managements“ wird als eine vierte Stufe des Qualitätswesens identifiziert, die

sich historisch aus den darunterliegenden, mit zunehmender Erkenntnis einzubeziehender Aspekte seit den 1950er entwickelt hat. Insbesondere macht sich die Veränderung im Zeitverlauf dabei an den Qualitätssichtweisen bemerkbar, die charakteristischerweise in der ersten Stufe „Qualitätskontrolle“ allein auf das Produkt fokussierte. Mittels der Kontrolle des Endprodukts z.B. durch Einrichten entsprechender Prüfabteilungen. Durch die Stellung am Ende der Fertigungsprozesse im Sinne einer Endkontrolle, ermöglichten diese Ansätze jedoch nur Fehlerbehebung am fertigen Produkt. Die Erkenntnis, dass sich durch den Einbezug der zur Erstellung des Produktes führenden Prozesse und der dadurch möglichen frühzeitigeren (kostengünstigeren) Intervention im Sinne eines qualitätsgesicherten Produktionsprozesses vermeiden lassen, führte schließlich zu der zweiten Stufe des Qualitätswesens – Qualitätssicherung. Hiermit wurde die Fehlersuche und Behebung an ihren Entstehungsort verlagert und mittels einer prozessorientierten Qualitätssicht das Augenmerk auf die das Produkt erzeugenden Abläufe ausgeweitet, mit der Zielstellung fehlerfreie Produktionsprozesse zu etablieren. Trotz der Hinzunahme dieser prozessorientierten Sichtweise, die z.B. den Ansatz zur statischen Prozesskontrolle (SPC) auf Grundlage der Theorie des statischen Testens hervorbrachte (siehe ebd. Seite 97), konnten mit dieser Herangehensweise nicht alle Qualitätsmängel behoben werden. Ähnlich zur in der Theoriebildung der integralen Planung erkannten Notwendigkeit zur Berücksichtigung der Integration von vertikalen Interdependenzen (vgl. *Horizontale Integration* Kap. 2.1.2 sowie Kap. 2.1.3) wurde das Qualitätswesen ein weiteres Mal lebenszyklisch-orientiert ausgeweitet. Dadurch wurden sämtliche „Prozesse und Aktivitäten von der Forschung und Entwicklung über Konstruktion und Vertrieb bis zum Kundendienst“ mit in die Qualitätssicht einbezogen und neben der Verbesserung der Qualität des Produkts an sich auch Methoden für diese ganzheitliche Qualitätsbetrachtung entwickelt und zur Anwendung gebracht. Als dritte Stufe des Qualitätswesens werden diese Ansätze unter dem Begriff „Total Quality Control“ versammelt, dessen wesentliche Kernpunkte Feigenbaum Anfang der 1960er in seiner gleichnamigen Veröffentlichung legte. Mittels einer ganzheitlichen Qualitätskonzeption im Kern einer umfassenden Unternehmensphilosophie bereitet die vierte Stufe des Qualitätswesens – Data Quality Management (DQM) – eine theoretische Grundlage, die sich in der normierten Begriffsdefinition in den Einbezug aller Tätigkeiten der Gesamtführungsaufgaben, also

Qualitätspolitik, -ziele sowie der Verantwortungen für die Qualität, wiederfinden (ebd. Seite 97ff).

Abgrenzung Qualitätswesens (TQM) auf Gegenstand Daten/Information – Qualitätsdimensionen bzw. Kategorien (Dateneigenschaften)

Die Deutsche Gesellschaft für Informationsqualität (DGQI) hat den originären Ansatz, der aus im Kontext des MIT *Total Data Quality Management* (TDQM) Programms 1996 entwickelt wurde (vgl. Wang & Strong, 1996), in die deutsche Sprache überführt bzw. ergänzend angepasst. Die Bewertungssystematik des Ansatzes basiert auf Ergebnissen einer empirischen Erhebung auf Seite der Datennutzer, von denen Datenqualitätsdimensionen abgeleitet wurden. Abbildung 11 zeigt die 15 Qualitätsdimensionen, mittels derer umfänglich und widerspruchsfrei sämtliche Merkmale der Informationsqualität (IQ) aus der Perspektive der Anwender*innen beschrieben werden können (vgl. *fit-for-use* Konzept). (Hildebrand u. a., 2015).



Abbildung 11 Informationsqualitäts-Dimensionen in 4 Kategorien (Hildebrand u. a., 2015, S. 30)

Eine Bewertung der eigentlichen Informationsqualität (IQ) (was nicht Inhalt des Ansatzes ist, vielmehr gibt dieser den Nutzer*innen eine Metrik an die

Hand, um dies disziplinübergreifend, einheitlich zu bewerkstelligen) hängt elementar von der Sicht auf den Untersuchungsgegenstand ab. Daher wurden die 15 Dimension eindeutig in vier Kategorien unterteilt, die mit den vier zusammenhängenden Sichten (vgl. Abbildung 11) auf den Gegenstand *Information* in seinem Kontext korrespondieren (ebd.).

Folgender Zusammenhang zwischen IQ-Kategorie und Untersuchungsgegenstand bestehen:

- Systemunterstützt – System
- Inhärent – Inhalt
- Darstellungsbezogen – Darstellung
- Zweckabhängig – Nutzung

Die explizite Darstellung des Gefüges der Untersuchungsgegenstände (vgl. Abbildung 12) bezieht sich dabei auf die „organische Struktur der Informationsverarbeitung eines Unternehmens“. Neben dem Aufzeigen der gegenseitigen Wechselwirkungen der (gleichwertigen) Untersuchungsgegenstände, soll damit eine „Orientierungshilfe für die Entscheidung im Rahmen der Bewertung von Informationsqualität“ für den zugrundeliegenden anwendungsbezogenen Qualitätsbegriff des Ansatzes gegeben werden (ebd. Seite 31).

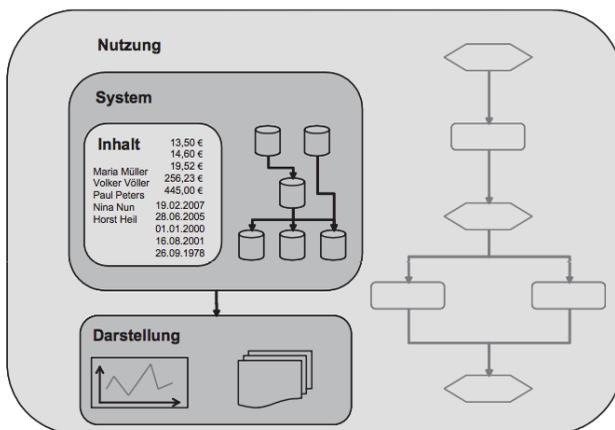


Abbildung 12 Untersuchungsgegenstände der Informationsqualität (Hildebrand u. a., 2015, S. 30)

Kritik fit-for-use Ansatzes (u.a. Hildebrand et al., 2015, S. 52)

Als wesentliches Problem bei der Implementierung des Ansatzes wird die subjektive Einschätzung aus Sicht der Nutzer*innen gesehen, die nicht bzw. schwer generalisierbar bzw. formalisierbar ist. Da diese jedoch a priori erhoben und dann die zentrale Grundlage der situativ geltenden Datenqualitätsmerkmale für die Bewertung der Informationsqualität bildet, resultiert die Schwierigkeit einer rechner-gestützten Umsetzung des Ansatzes in der Praxis (ebd. Seite 52):

Des Weiteren bezieht der Ansatz, dessen theoretische Grundlagen aus der Übertragung von Konzepten aus des ganzheitlichen TQM auf das Thema Daten resultiert, folgerichtig eine Qualitätsplanung und -messung mit ein. Die Zielstellung eines umfassenden Qualitätsmanagements bezüglich des Datengegenstandes wird dabei jedoch nicht vollständig konkretisiert (Helfert, 2002, S. 122).

Als bedeutender Ansatz aus dem nicht akademischen Bereich, der sich an den Ansätzen des TDQM orientiert und die Umsetzung eines qualitätsorientierten Informationsmanagements fokussiert, gilt die Entwicklung einer entsprechenden Vorgehensbeschreibung von Larry English (vgl. Helfert, 2002 bzw. English, 1999 zitiert nach; Hildebrand u. a., 2015). Das in der Praxis angewendete Verfahren berücksichtigt jedoch nicht durchgängig ein theoretisch fundiertes, methodisches Vorgehen (Helfert, 2002) und wird daher im Folgenden anhand eines rezitierenden Beitrages (Hildebrand u. a., 2015) kurz vorgestellt.

Erweiterung des Ansatzes von Larry English: Strukturierung von DQ-Merkmalen (zitiert nach Hildebrand u. a., 2015, S. 89)

Dem Ansatz liegt eine Strukturierung der DQ-Merkmale in drei Hauptmerkmalen zugrunde. Dabei bildet das erste – Qualität der Datendefinition – die Grundlage der beiden anderen Hauptmerkmale, indem damit eine verständliche und vollständige Beschreibung der (Daten-) Objekte gefordert wird. Die Qualitätsbetrachtung der konzeptionellen Definition (Spezifikation der Daten) geschieht hierzu auf drei Ebenen (ebd.).

Bezüglich der Datenspezifikation wird die Beschreibung der Objekte zum einen gemäß ihrer absoluten fachlichen Bedeutung sowie zum anderen in Hinblick auf ihre Modellierung im System betrachtet. Hieraus können dann Aussagen zur DQ abgeleitet werden, wie beispielsweise Existenz, Verfügbarkeit und Aktualität einer Dokumentation. Auf dieser Ebene soll eine Vollständigkeit der Datenobjekte bzw. Verständlichkeit der Beschreibung bewertet werden, „um klarzustellen mit welcher ‚Sprache‘ im System gearbeitet wird“ (ebd. Seite 89). Die zweite Betrachtungsebene – Geschäftsregeln – und Dritte – Integritätsregeln – hängen voneinander ab, da die mit den Geschäftsregeln beschriebenen Beziehungen der Objekte untereinander, sofern sie im konzeptionellen Modell (Schema) abgebildet sind, in Form der *Referenziellen Integrität* geprüft werden können. Da diese nicht vollständig für alle möglichen bzw. relevanten Beziehungen modelliert werden können, finden sich diese implizit in auf dem Modell aufbauender Programmlogik bzw. werden vor dem Aufnehmen von Daten in ein Informationssystem überprüft. Mögliche DQ-Aussagen zu diesen Ebenen behandeln Güte des Datenmodells, Existenz, Vollständigkeit und Richtigkeit von Plausibilitätsprüfungen am Dateneingang (ebd.).

Das zweite Hauptmerkmal – inhaltliche Datenqualität – untergliedert sich in die DQ-Merkmale

- Vollständigkeit
- Eindeutigkeit
- Einhaltung der Geschäftsregeln
- Genauigkeit und Fehlerfreiheit

und betrachtet auf Basis der für die Datenspezifikation bewerteten DQ-Merkmale die Korrektheit des Datenwertes eines spezifischen (Feld-)Inhaltes. Des Weiteren ermöglichen die Geschäftsregeln das Aufdecken von Unplausibilitäten bezüglich anderer (Feld-) Inhalte bzw. die Integritätsprüfung von Unstimmigkeiten zu in Beziehung stehenden weiteren Datensätzen (ebd.).

Das dritte Hauptmerkmal - Qualität der Datenrepräsentation – lässt sich im Gegensatz zu den beiden anderen nur durch direkte Einbeziehung der (End-) Nutzer*innen feststellen. Es ist dabei untergliedert in:

- Rechtzeitige Bereitstellung
- Angemessenheit des Formats
- Verständlichkeit des Formats

Wobei die Datenrepräsentation im Kontext des Ansatzes als nicht kritisch erachtet, was in einem Fokus auf Wichtigkeit einer korrekten Datendefinition zum Ausdruck kommt. Zusammenfassend lässt sich im Vergleich zum stärker auf die Endnutzer*innen der Daten fokussierten sowie Metainformationen zu den Daten im Bereich der Datendefinition hinterlegenden Ansatz des TDQM feststellen, dass es unerheblich ist, nach welchem Modell vorgegangen wird. Entscheidend sei jedoch, bei der praktischen Durchführung alle Bereiche vollständig einer Überprüfung zu unterziehen. Bezüglich einer Operationalisierung, also dem Messbarmachen der Datenqualität, schlägt er vor die „rein technische Vorgehensbeschreibung“ einer bisher nur auf Datenqualitätsmetriken fußenden Datenqualitätspyramide um die Geschäftsregeln zu erweitern (siehe Abbildung 13), da diese eine Definition geeigneter Metriken erst ermöglicht. Dabei wird jedoch auch auf Probleme des Ansatzes mit Geschäftsregeln hingewiesen, die selten explizit dokumentiert vorliegen. So liegen zwar Dokumentationen über bestimmte Datenfelder vor, jedoch wird der fachliche Zweck (Wieso ist das Objekt in dieser Weise definiert? bzw. welche fachlichen Beziehungen bestehen zu anderen Objekten?) in der Regel fehlen. Durch die oftmals nur natürlich-sprachlich vorliegenden, schlecht definierten Geschäftsregeln, seien diese für eine systembezogene Nutzung darüber hinaus nicht direkt einsetzbar. Es bedarf einer „Veredelung“ im Sinne des Datenqualitäts-Managements zur sinngemäßen Nutzung bei der Bestimmung von Datenqualitäten (ebd. Seite 92f.).



Abbildung 13 Erweiterte Datenqualitätspyramide (Hildebrand u. a., 2015, S. 92)

Der von Hildebrand (2015) dargestellte Beitrag von Gebauer bezieht sich auf Datenbanken (Schema, Tabellen, Schlüssel usw.). Dabei stehen in diesem Kontext auch die zur Operationalisierung der Datenqualitätsbetrachtung beschriebenen Metriken bzw. entsprechenden Geschäftsregeln, wie beispielsweise aus Sicht der Anwender*innen welche Daten in „richtige“ und „falsche“ zu differenzieren sind). Von diesen folgend gelisteten Regeltypen erscheint der sich auf die Dateninhalte beziehende Typ sowie die ihm zugeordneten Methoden, auch auf andere Kontexte übertragbar zu sein, wie beispielsweise die Inhaltsprüfung objekt-orientierter Modelle im Rahmen des Lösungsansatzes der vorliegenden Arbeit.

- a) Key-Rules – Messbarer Ausdruck einer Geschäftsregel mit Bezug zu Schlüsseldefinitionen der Datenbank,
- b) Matching Rules – Messbarer Ausdruck einer Geschäftsregel zur Bestimmung des Übereinstimmungsgrades verschiedener Datentabellen und
- c) Content Rules – Überprüfung von Dateninhalten

Bezüglich des Anwendens der inhaltsbezogenen Regeln zur Überprüfung der Inhalte werden folgende methodisch aufeinander aufbauende „Messstufen“ von einzelnen Regeln („... wobei Einschränkungen an die Inhalte [engl. Constraints] mit jeder Stufe schärfer werden“; ebd. S.93) vorgeschlagen:

- *Completeness* – Überprüfung auf korrekte Füllung und Nicht-Füllung
- *Format* – Überprüfung ob Inhalt Konsistenz zu vorgegebenen Muster ist
- *Range* – Überprüfung ob Inhalt im vorgegebenen Wertebereich liegt
- *Plausibility* – Überprüfung von ein- bzw. mehrdimensionalen Relationen zwischen Feldern des Datensatzes
- *Accuracy* – Überprüfung von ein- bzw. mehrdimensionalen funktionalen Abhängigkeiten zwischen Feldern des Datensatzes

Wichtig für den Aufbau der Messstufen ist, dass diese bezüglich der Überprüfung eines Datenwertes nicht vollständig durchlaufen werden müssen. Wenn in einer Stufe ein Fehler festgestellt wird, macht das aufgrund dessen, dass die Stufen aufeinander aufbauen, ein Prüfen der nächsten Stufe obsolet (ebd.).

Helferts (2002) Datenqualitätsansätze

Im Gegensatz zu den im Rahmen der Betrachtung von Datenqualität oben beschriebenen, anwenderbezogenen Ansätzen des TDQM bzw. den technologieorientierten Operationalisierungsansätzen Gebauers, stellt Helfert (2002) eine Unterteilung nach den Qualitätssichten *Design-* und *Ausführungsqualität* auf:

1. Designqualität – Grad an Übereinstimmung zwischen Nutzeranforderung und umgesetzter Spezifikation des Informations-Systems (IS)
2. Ausführungsqualität – Grad der (tatsächlichen) Einhaltung der Spezifikation durch das IS

Diese Unterteilung kann, wenn sie als Strukturierungshilfe verstanden wird, dabei eine Einordnung der in der Praxis vorkommenden Informationssysteme ermöglichen. Wobei die im Ansatz aufgegriffene Unterscheidung für die Messung von DQ durch Hildebrand (2015) als sinnvoll angesehen wird, da eine Trennung zwischen „(subjektiver) Einschätzung der Konformität zwischen dem spezifizierten Datenangebot und dem Datenbedarf des Nutzers von der (objektivierbaren) Analyse der Übereinstimmung von vorhandenem und spezifiziertem Datenangebot“ (ebd., S.53) eingehalten ist. Neben dem Abbilden der Qualitätsanforderungen in Form von überprüfbaren Regelmengen, wird als zentraler Aspekt des Ansatzes vorgeschlagen das DQ-Management in die Metadatenverwaltung zu integrieren. Damit soll weitestgehend eine Automatisierung und Werkzeugunterstützung im Rahmen der Handhabung der DQ gewährleistet werden (vgl. ebd., S.53).

Im Zuge der Operationalisierung werden ähnlich zu den oben dargestellten Regeltypen in den Meßstufen auch die beiden Qualitätssichten weiter mit Kriterien der DQ untergliedert. Aufgrund des erkannten Defizits zahlreicher Ansätze im Bereich der Datenqualität – dem Fehlen von empirischen erfassten Datenqualitätsforderungen sowie entsprechender Eigenschaften qualitativ hochwertiger Daten – folgert Helfert (2002, S. 81) auf Basis einer empirischen Untersuchung folgende wichtige DQ-Eigenschaften (vgl. ebd., S. 92):

- Designqualität
 - Widerspruchsfreiheit (Konsistenz) – inhaltlich und zeitlich konsistente Daten
- -Ausführungsqualität
 - Vollständigkeit
 - Korrektheit
 - Repräsentation fehlender Werte
 - Zeitliche Verfügbarkeit
 - Aktualität
 - Referentielle Integrität

- Syntaktische Korrektheit
- Rückverfolgung der Datenherkunft
- Dokumentation von Abweichungen
- Semantik
- Identifizierbarkeit von Daten (einheitliche, klare und genaue Beschreibung der Datenmodelle und -flüsse)

Diese empirisch untersuchten Aspekte der DQ ermöglichen einen geeigneten Ausgangspunkt im Rahmen der vorliegenden Arbeit für die Einordnung der Qualitätsdimensionen als Grundlage des im Lösungsansatz beschriebenen regelbasierten Handhabens von modellbasierten Planungsdaten. Bei den von Helfert untersuchten Eigenschaften sind unter den meistgenannten auch die folgenden an anderer Stelle bezüglich einer (objektiven) Messbarkeit von Datenqualität oft diskutierten DQ-Dimensionen (vgl. Hildebrand u. a., 2015, S. 49; West, 1999): Vollständigkeit, Fehlerfreiheit, Konsistenz und Aktualität.

Zusammenfassung von Datenqualitätsmerkmalen für den Rahmen der Arbeit

Im Sinne dieser Arbeit werden zum Abschluss dieses Grundlagenkapitels eine Selektion gewichtiger DQ-Eigenschaften zusammengestellt und den eingangs beschriebenen Kategorien zugeordnet. Dabei wird ein Fokus auf fachliche Datenqualität gelegt, die bezüglich des Datenmodells weitgehend automatisiert und objektiv überprüfbar sein soll, sowie Aspekte der „technischen“ DQ im Sinne des Datenformats und dessen Handhabung. Prinzipiell werden die im allgemeinen Qualitätsmanagement erkannten DQ-Merkmale für eine Einordnung der fachspezifischen Ansätze im Bereich der gebauten Umwelt – Stadt-, Bauwerks- sowie (fachliche) Detailebene – bezüglich der (daten-) technischen Anforderungen zusammengestellt. Die eigene methoden-orientierte Strukturierung – Validieren, Prüfen, Analysieren bzw. Simulieren – dient dabei dem thematischen Fokus einer Modellprüfung und -analyse der vorliegenden Arbeit als Ordnungsrahmen (vgl. Abbildung 14), in dessen Bereiche die verschiedenen im Lösungsansatz vorgeschlagenen Regelarten eingeordnet werden können.

nen (vgl. Abbildung 22). Diesem Rahmen liegt folgende eigens erweiterte Zuordnung der (ausgewählten, auf ein Datenprodukt ausgerichteten) DQ-Dimensionen zu der eigenen Strukturierung von (technischen) Methoden zugrunde. Dabei sind grau hinterlegt die Eigenschaften der original Zusammenstellung von West (1999), unterstrichen Eigenschaften nach Hildebrand u.a. (2015) sowie ausgegraut Ergänzungen nach Helfert (2002):

a) Validieren – bezüglich definitionsbezogener DQ-Eigenschaften:

- Relevanz
- Eindeutigkeit
- Konsistenz (Widerspruchsfreiheit)
- Interpretierbarkeit
- Nützlichkeit (Zweckbezogen)

b) Prüfen – bezüglich Inhaltsbezogene DQ-Eigenschaften:

- Aktualität
- Genauigkeit
- Fehlerfreiheit
- Glaubwürdigkeit
- Zeitlicher Bezug
- Nützlichkeit
- Verfügbarkeit

c) Analyse – bezüglich ganzheitlicher DQ-Eigenschaften

- Vollständigkeit
- Zugänglichkeit
- Aufwand

In Abbildung 14 sind gewichtige Aspekte in den Kategorien Validieren, Prüfen sowie Analysieren zusammengefasst. Das Prüfen der Daten bezüglich sowohl formal-struktureller, als auch (bau-) fachlich inhaltlicher Qualitäten stellt dabei

einen zentralen Bereich der im Rahmen der vorliegenden Arbeit behandelten Aspekte der Datenqualität dar. Essentiell, im Sinne der oben beschriebenen aufeinander aufbauenden Meßstufen, unterliegt dem Bereich des Prüfens der Daten (vgl. mittlere Zeile in Abbildung 14) dabei der Bereich der Validierung. Darin spielen beispielsweise Aspekte zu Konsistenz der Daten eine Rolle, auch im Sinne einer Gewährleistung der Schemakonformität von den Daten. Generell stehen aufgrund einer erforderlichen Datenqualität die im Lösungsansatz der Arbeit verfolgte Entscheidungsunterstützung – basierend auf der Prüfung und Analyse von Planungsmodellen – in einem engen Zusammenhang zum im Kapitel dargelegten Thema des Datenqualitätsmanagements. Eine fundierte Auseinandersetzung mit der Datenqualitätsthematik kann daher den Anwender*innen der in der Arbeit vorgeschlagenen regelbasierten Lösung zur Modellprüfung bzw. -analyse eine zweckdienliche Anwendung ermöglichen.

	Systemunterstützt	Inhärent	Darstellungsbezogen	Zweckabhängig
Validieren (Designqualität - Schema)			Eindeutigkeit, Konsistenz	Relevanz
Prüfen (Ausführungsqualit ät - Instanz)		Fehlerfreiheit	Genauigkeit	Aktualität
Analysieren (hybrid)	Zugänglichkeit			Vollständigkeit, Aufwand

Abbildung 14 Zusammenstellung der für die Arbeit wichtigen DQ-Dimensionen

3 Lösungsansatz und weiterführende Anforderungen

Zur umfänglichen Beschreibung des von der methodischen auf die umsetzungstechnische Ebene spannenden Lösungsweges der vorliegenden Arbeit, ist er in diesem Kapitel abgestuft dargelegt. Der zu Beginn in Kapitel 3.1 beschriebene Lösungsansatz für eine integrierte Entscheidungsunterstützung bezieht den designierten Anwenderkreis als Teil der konzipierten Lösung mit ein. Das inhaltliche Fachwissen zu der spezifischen Planungssituation und den zugrundeliegenden Planungsgegenständen wird dabei herangezogen, um entsprechende Methoden und Hilfsmittel vorbereitend für das Erstellen von Entscheidungsgrundlagen auf Basis der Planungsgegenstände an die Planungssituation anzupassen. Konkret geht es um die Analyse und Prüfung von virtuellen Produktmodellen, zu denen inhaltliche Aspekte problemgerecht aufbereitet die Planenden dann in der Entscheidungssituation unterstützen können. Der im Bereich der Qualitätssicherung von virtuellen Gebäudemodellen erkannten Notwendigkeit einer transparenten Nachvollziehbarkeit beim Prüfen und Analysieren von Inhalten eines kooperativ bearbeiteten Planungsgegenstandes (vgl. Kap. 2.3) folgend, beschreibt der Lösungsansatz die Einbindung des Anwenderkreises auf Basis einer regelbasierten Grundkonzeption. Darin sind in einer Ebene die Hilfsmittel und Methoden für die Produktmodellseite zusammengefasst. Womit die, auf Basis des Fachwissens der Anwender*innen spezifischen, einer Entscheidungssituation zugrundeliegenden Fragestellungen an den Planungsgegenstand ausgeprägt werden. Diese Ebene ist mittels der Regelkonzeption über das Verbindungselement einer formalisierten Regel mit der (technischen) Ebene verbunden. Auf welcher dann Konzeptionen für dynamische Analyseprozesse zusammengefasst werden, die entsprechende Entscheidungsgrundlagen auf Basis der Produktmodelle aufbereiten können. Für die weiteren Unterkapitel des Lösungsansatzes sowie für die Kapitel der Umsetzung bilden diese beiden Ebenen der Grundkonzeption – Methoden und Hilfsmittel für Analyseregeln bzw. Prüflogik bezüglich produktmodellbasierter Planungsgegenstände einerseits und dem dynamischen Anwendungsansatz mit

Werkzeugen für modellbasierte Analyseprozesse andererseits – eine thematische Gliederung bezüglich den fachlichen und (informations-) technischen Inhalten der Arbeit.

Die im Lösungsansatz beschriebene Grundkonzeption wird im Kapitel 3.2 weiter konkretisiert. Wobei die Methoden und Hilfsmittel, die auf dem Lösungsansatz aufbauen, zunächst für die Ebene der Prüf- und Analyselogik anhand von Anforderungen weiter spezifiziert werden (Kap. 3.2.1). Für die Ebene des Anwendungsansatzes werden daran anschließend durch konzeptuelle, technologieneutrale Festlegungen benötigte Spezifizierungen der dynamischen Anteile des Ansatzes weiter konkretisiert (Kap. 3.2.2). Bezüglich der noch unvollständig etablierten virtuellen Produktmodelle sowie den in der Praxis fragmentiert umgesetzten modellbasierten Arbeitsweisen werden dann, die Ausführungen zum Lösungsansatz abschließend, weiterführende kontextuelle Festlegungen bestimmt (Kap. 3.2.3). Mit dem Lösungsprinzip der Modellprüfung und -analyse ermöglichen diese es dann Grundlagen für integrierte Entscheidungsfindungen aufbereiten zu können.

3.1 Lösungsansatz

Um die in Kapitel 1.2 dargelegten übergeordneten Ziele zur kollaborativen Lösung komplexer planerischer Problemstellungen erreichen zu können, stellt sich zunächst die Frage, wie und womit ein **integraler Planungsprozess** realisiert werden kann. Bezug nehmend auf die Ausführungen zur Integralen Planung (vgl. Kap. 2.1) stellt die Entscheidungsfindung einen zentralen Ansatzpunkt dar.

Die hierbei auftretende Herausforderung ergibt sich aus der Qualität und dem Kontext planerischer Problemstellungen (Rittel führt diese auf eine inhärente Börsartigkeit zurück: “Yet all essential planning problems are wicked, ...” Quelle: Rittel, 1972, S. 392). Es geht also um eine **Unterstützung von Entscheidungsfindungen** bei komplexen interdisziplinären planerischen Problemstellungen. Diese zeichnen sich dadurch aus, dass:

- Zumeist eine nicht vollständig erfasste Ausgangssituation – IST Zustand des Planungsgegenstands – vorliegt
- Eine Zielstellung, welche die gewünschte Planungslösung – SOLL Zustand des Planungsgegenstands – mittels problemorientierter Anforderungen bzw. Bewertungskriterien umschreiben, unklar bzw. nicht präzise genug definiert ist
- Eine Vielzahl an Fachaspekten und damit einhergehender organisatorischer Abstimmung zwischen den beteiligten Disziplinen für eine ganzheitliche Entscheidungsfindung relevant sind

Die Abbildung 15 zeigt schematisch eine im Planungsverlauf wiederkehrende idealtypische Entscheidungssituation. Die Schritte Problemanalyse, Systemsynthese und Systemanalyse bilden dabei einen den Entscheidungsprozess unterstützenden Regelmechanismus, welcher auf einer durchgängigen Informationsbasis beruht (von Both, 2011, S. 235). Wichtig für eine zielgerichtete Unterstützung der Entscheidungsträger ist eine Abstimmung der in die Planungslösung einfließenden Fachaspekte, da diese sich zum Teil gegenseitig bedingen können. Mit dem übergeordneten Projektziel, stehen darüber hinaus alle einzelnen Entscheidungssituationen im starken Zusammenhang und können nicht davon isoliert abgehandelt werden.

„Die Realisierung eines solchen ganzheitlichen integralen Planungsprozesses impliziert eine durchgängige planungsbegleitende Evaluierung und Optimierung des Planungsgegenstandes gegenüber dieser integrierten Gesamtzielsetzung.“ (von Both, 2011) Dies führt – als Ergänzung zu bestehenden fachspezifischen Simulations- und Bewertungsansätzen – zu einer Forderung nach einer integrierten, ganzheitlichen, d.h. fachübergreifenden Analyse des Planungsgegenstandes.

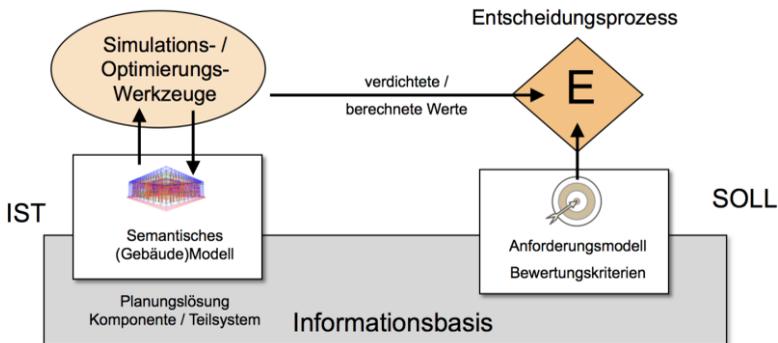


Abbildung 15 Kernprozess der Entscheidungsunterstützung (Quelle: von Both & Ebertshäuser, 2018)

Aktueller Stand

Diesbezüglich kann in der wissenschaftlichen und planerischen Praxis (Treeck u. a., 2014) eine Zuwendung zu simulationsgestützten Hilfsmitteln festgestellt werden. Anstelle der geforderten iterativen, planungsbegleiteten Evaluierung des gesamtheitlichen Planungsgegenstands werden bei den bestehenden Ansätzen jedoch zumeist nur ausgewählte, einzelne Fachaspekte einer isolierten simulationstechnischen Überprüfung unterzogen. Auf diese Weise können Wechselwirkungen zwischen diesen Fachaspekten, die insbesondere für die Gesamtplanung bedeutsam sind, nur schwer bewertet werden. Zwar existieren auf Simulationsseite bereits Ansätze, die Ebene der Wechselwirkungen mittels der Koppelung verschiedener fachspezifischer Simulationswerkzeuge mit einzubeziehen. Aber ihr Einsatz lohnt sich u.a. aufgrund des hohen Aufwands jedoch hauptsächlich bei Fachaspekten mit gut bekannten, starken Wechselwirkungen. Als Beispiel sei die wechselseitige Evaluation einer thermisch-physikalischen Simulation der Bauwerkshülle mit einer Anlagensimulation der Wärmeversorgung (Pazold u. a., 2014) genannt.

Exemplarisch für heutige Planungssituationen stellt Abbildung 16 auf der Bauwerksebene schematisch die bezüglich der Fachplanungen sowie der Simulationsdomäne fragmentierte Evaluation einzelner Fachaspekte dar. Eine an

zentraler Stelle (Architekt*in) avisierte ganzheitliche Optimierung des Planungsgegenstands setzt dabei einen hohen, sowohl organisatorischen als auch fachlichen Abstimmungsbedarf voraus. Was letztendlich auch methodisch oft dazu führt, lediglich die finale Planungslösung zu evaluieren, anstatt planungsbegleiteten Evaluationsiterationen durchzuführen.

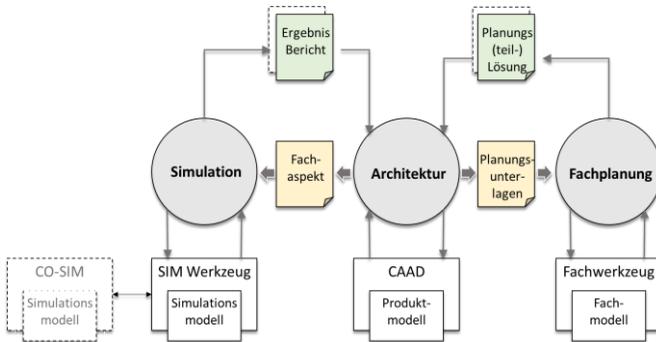


Abbildung 16 Einzelne Planung/Simulation von Fachaspekten

Übergeordneter Lösungsansatz

Der Lösungsansatz der hier vorliegenden Arbeit setzt daher an der Problemstellung der Entscheidungsfindung an und beinhaltet Methoden und Hilfsmittel zur Unterstützung ganzheitlicher Entscheidungen. In dem Lösungsraum, der in Abbildung 17 schematisch dargestellt ist, beschreibt der Lösungsansatz hierzu eine modellbasierte integrierte fachübergreifende Analyse als Basis der Entscheidungsfindung.

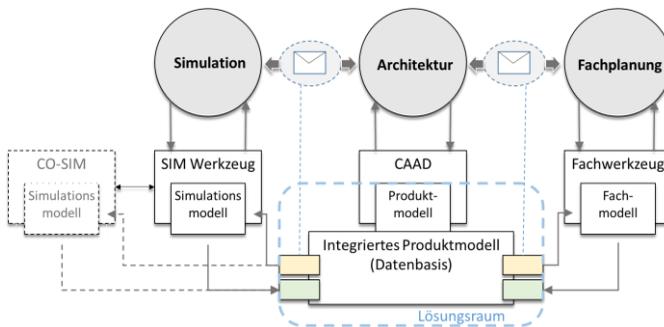


Abbildung 17 Lösungsraum der Arbeit zur Unterstützung integrierter Entscheidungsprozesse

Konkreter Ansatz

Im Folgenden wird zunächst das konzeptionelle „Grundgerüst“ des Ansatzes konkretisiert, auf welches die beinhaltete Entwicklung von Methoden und Werkzeugen zur Unterstützung integrierter Entscheidungsprozesse aufbaut. Die Bestandteile des Ansatzes sind dabei um das zentrale Kernprinzip – durch fachlich-inhaltliche Modellprüfung/-analyse mannigfaltige, anforderungsorientierte Entscheidungsgrundlagen zu ermöglichen – angeordnet. Die Grundlage zur Generalisierung dieses Kernprinzips bilden regelbasierte Konzepte zum formalisierten Dynamisieren des Anwendungskontextes sowie die konsequente Trennung fachlicher (Geschäfts-) Logik von den sie verarbeitenden Mechanismen. Um dies zu erreichen und damit die von unterschiedlichen Seiten einwirkenden Anforderungen (siehe Kap. 3.2.1) durch adäquate konzeptionelle Festlegungen (Kap. 3.2.2) bedienen zu können, ist der Lösungsansatz in zwei grundlegende Ebenen aufgeteilt:

1. Methoden und Hilfsmittel zur inhaltlichen und formalen Spezifikation der entscheidungsrelevanten Fragestellungen als Analyseregeln bzw. Prüflogik

2. Entwicklung dynamisch anpassbarer Werkzeuge, welche kontext- und problemspezifische Prüf- und Abfragelogik in modellbasierten Analyseprozessen umsetzen können

Elementar für das regelbasierte Verbindungskonzept der beiden konzeptuellen Ebenen sind dabei auf Seiten der Werkzeuge Mechanismen zum Erstellen, Anpassen sowie Verwalten der der Analyse- bzw. Prüflöge selbst in Form von Regeln. Damit wird der konzeptionelle „Regelkreislauf“ bezüglich einer dynamischen Weiterentwicklungsstrategie geschlossen und ein designerter Anwenderkreis prinzipiell befähigt, die Werkzeuge – ohne Kenntnis bzw. Änderung ihrer Funktionsweise – gemäß seinem Anwendungskontext inhaltlich auszuprägen. Die erste Ebene bündelt im Weiteren die Fragestellung beim Konkretisieren des Lösungsansatzes, wie die spezifischen Ausprägungen und Weiterentwicklungen der fachlichen Analyse- und Prüflöge den Anwender*innen ermöglicht werden soll, welchen durch Kenntnis der konkreten Situation die höchste Problemlösekompetenz hierfür zukommt. Die zweite Ebene der Werkzeuge ist hiervon durch das formale Verbindungselement *der Regel* strikt getrennt, sodass auf dieser Ebene weiter konkretisierte, (umsetzungs-) technische Konzeptionen des Lösungsansatzes allein auf technischer Seite weiterentwickelt bzw. optimiert werden können.

3.2 Konkretisierung des Lösungsansatzes anhand von Anforderungen

In den folgenden Abschnitten wird die auf der beschriebenen Grundkonzeption aufbauende Entwicklung von Methoden und Hilfsmitteln des Ansatzes zunächst seitens der Prüf- und Analyselogik anhand von Anforderungen weiter spezifiziert (Kap. 3.2.1). Daran schließen sich konzeptuelle Festlegungen an, die für die Konkretisierung des Werkzeugansatzes benötigt werden (Kap. 3.2.2). Um mittels des Lösungsprinzips der Modellprüfung dann Grundlagen für integrierte Entscheidungsfindungen aufbereiten zu können, werden dann, die Ausführungen zum Lösungsansatz abschließend, in dessen Rahmen stehende weiterführende kontextuelle Festlegungen erläutert.

Fachlich-logische multimodale Prüfungen von integrierten Produktmodellen

Durch den Einsatz von zunehmend komplexer werdenden Planungsmodellen stehen selbst erfahrene Planer*innen vor dem Problem, nicht mehr nachvollziehen zu können, ob der in einer vorliegenden Modellinstanz abgebildete Planungsgegenstand den gesetzten Zielstellungen entspricht. Bei den operativen Entscheidungsfindungen im Planungsverlauf kommen – verschärft durch Zeit- und Kostendruck – Abstimmungsbedarfe fachlicher Aspekte hinzu, deren Wirken bezüglich der Gesamtlösung bewertet werden muss. Im Gegensatz zu aufwendigen Evaluationen mittels Simulation von Einzelaspekten sieht der Lösungsansatz hierzu die Modellprüfung des gesamten Modellzustandsraumes vor. Das Umsetzen der Methode der Modellprüfung setzt dabei voraus, dass der Zielstellung entsprechend einzuhaltende Eigenschaftswerte bzw. entscheidungsrelevante Fragestellungen zunächst als temporallogische Formeln spezifiziert werden (Flake & Müller, 2001). Zur inhaltlichen Ausprägung dieser, sich aus den Fragestellungen einer Entscheidungssituation ergebenden, Prüf- bzw. Analyselogik bezieht der Lösungsansatz (a priori) die Anwender*ebene selbst mit ein. Darauf aufbauend ermöglicht der Lösungsansatz dann die fachlich-logischen, multimodalen Prüfungen von integrierten Produktmodellen zur Unterstützung der spezifischen Entscheidungsfindung. Daher stellt sich zunächst die Frage, was für die Gestaltung dieser Interaktion (Eingabe der Prüf- und Analyselogik durch die Anwender*innen) benötigt wird.

3.2.1 Was ist nötig für das Konzept?

Mit der Aufteilung in die zwei grundlegenden Ebenen – Methoden und Hilfsmittel für Analyseregeln bzw. Prüflogik bezüglich produktmodellbasierter Planungsgegenstände einerseits sowie dem dynamischen Softwareansatz mit Werkzeugen für modellbasierte Analyseprozesse andererseits – ist zunächst ein entwicklungstechnischer Rahmen gegeben, der prinzipiell die Weiterentwicklung von technischer und anwendungsbezogener Geschäftslogik unabhängig voneinander ermöglicht. Um dabei übergeordnet die Konsistenz des Gesamtansatzes zu gewährleisten, sind die beiden Ebenen über einen Formalismus verbunden.

Mehrstufige Spezifikation entscheidungsrelevanter Fragestellungen

Auf die konzeptionelle Ausgestaltung des Interaktionsraumes zur Eingabe und Anwendung der Prüf- und Analyselogik wirken demnach abgestufte, aufeinander aufbauende Anforderungen von verschiedenen Seiten:

1. *Stufe der Anwendung*: Bedarfsorientiertes Ausführen der Analyse- bzw. Prüflöge auf den Planungsgegenstand (integriertes Produktmodell) gemäß der Entscheidungsfindung
2. *Stufe der Verwaltung*: Organisieren/Strukturieren von Analyse- bzw. Prüflöge gemäß verschiedener Anwendungskontexte
3. *Stufe der Manipulation*: Eingabe/Anpassung von Inhalten der Analyse- bzw. Prüflöge gemäß verschieden versierter Nutzergruppen
4. *Stufe der Umsetzung*: Maschinenverwertbares, formalisiertes Vehikel zum Transfer der Analyse- bzw. Prüflöge zur technischen Verarbeitung

Zu den genannten Stufen werden im Weiteren kurz die Anforderungen vorgestellt, wobei die entsprechenden Teile des Lösungsansatzes zugeordnet und erläutert werden.

1) Stufe der Anwendung

Konzept „Regel“: formalisierter Träger der Analyse- und Prüflöge

Als Produkt der darunterliegenden Ebenen findet hier die eigentliche Unterstützung der Entscheidungsfindung in der operativen Planung statt. Damit selbst nicht versierte Anwender*innen die Methode der Modellprüfung anwenden können, ist dabei sämtliche Komplexität vollständig verdeckt. Hierzu sieht der Lösungsansatz die Abstraktion sowohl der Modellkomplexität wie auch der entsprechenden Analyse- bzw. Prüflöge in Form von **Regeln** vor. Diese Regeln werden a priori situationsgerecht aufbereitet und bezeichnet (vgl. folgende Abschnitte), so dass sie in dieser Stufe der Anwendung dann auf den

aktuellen Planungsgegenstand ausgeführt werden können. Als Ergebnis liefern ebenfalls in der Regel festgeschriebene Auswertungsfunktionen problemspezifische Berichte zu den überprüften Sachverhalten des aktuellen Modellzustandes. Diese bilden direkt oder in Bezug auf weitere, auszuführende Regeln dann fundierte, nachvollziehbare (d.h. replizierbare) Entscheidungsgrundlagen. Im Planungsverlauf ermöglichen die zu den spezifischen Entscheidungsfindungen ausgeführten Regeln in Bezug auf den stetig verfeinerten Planungsgegenstand idealerweise dann eine transparente (replizierbare) Dokumentation des Hintergrundes der getroffenen Entscheidungen. Zum Beispiel die bei einer Entscheidung zwischen Planungsvarianten zugrunde gelegten technischen Kriterien zur Energieeffizienz des Gebäudes, welche zur Findung der optimalen Planungslösung mit Anforderungen eines übergeordnet angestrebten Planungsziels z.B. Gebäudezertifikat oder energetischer Standard verglichen wurden. Generell soll so eine ganzheitliche Bewertung der in komplexen Planungssituationen nicht ausbleibenden Vor- und Rücksprünge (Planungsvarianten) bzw. in Fällen von grundlegenden Planungsänderungen (Changemanagement) betroffenen Sachverhalte der Planung ermöglicht werden.

2) Stufe der Verwaltung

Konzept „Regelsatz“: als Strukturierungselement referenziert Regelinstanzen (ermöglicht Wiederverwendung in verschiedenen Kontexten)

Neben ad hoc in einer spezifischen Entscheidungssituation auftretenden speziellen Fragestellungen, hängt die Mehrheit der benötigten Analyse- und Prüflö- gik thematisch zusammen mit: A) planungsübergreifenden Anforderungsspezifikationsstandards (z.B. Dokumentationsrichtlinien der Öffentlichen Hand, wie BFR GBestand; Koggelmann u. a., 2012) an den Planungsgegenstand, B) der Art grundlegender Planungsziele (z.B. energetische Performance) sowie C) dem Reifegrad der Planung (z.B. Planungsphase). Dies birgt großes Potential der Wiederverwendung der diesbezüglich planungsspezifisch aufzubereitenden Regeln der Modellprüfung. Darüber hinaus erfordern die Kontexte der I) eigenen Organisationseinheit (z.B. Planungsbüro), II) wiederkehrenden Ko-

operationen (z.B. mit Fachplanern) sowie III) rahmengebenden Bedingungen/Regulatorien (z.B. Baueingaben, Nachweiswesen) langfristige (thematische) Bedarfe an wiederkehrenden Regeln, für die sich ein Aufbau (geteilter) Regelbasen lohnen kann. Der Lösungsansatz sieht mit der **Regelbibliothek** für diesbezügliche verwaltungstechnische Strukturierungen eine flexible Grundlage vor. Das (logische) Hauptelement zur eigenen, bedarfsgerechten Gliederung der Regeln bildet dabei der konfigurierbare **Regelsatz**.

Durch die Festlegung, dass jede Regelinstanz mindestens von einem Regelsatz referenziert werden muss, ist dabei ein einfaches Prinzip zur integren Wiederverwendung von Regeln zugrunde gelegt. Für die oftmals wiederkehrenden, vorgelagerten Prüfungen – ob beispielsweise ein weiter zu untersuchendes Modellelement überhaupt in der Modellinstanz existiert – werden diese „Grundregeln“ in einem Regelsatz der eigentlichen (Haupt-) Regel vorangestellt. Dadurch kann dann bei der Ausführung eines Regelsatzes festgestellt werden, ob beispielsweise die zu überprüfende Elementeigenschaft fehlerhaft ist, oder eine der vorgeschalteten Grundregeln bereits das Fehlen des Elements in der Modellinstanz anzeigt. Diese Konvention ermöglicht es dabei, die einzelnen Regeleinheiten inhaltlich, gemäß der eigentlichen Prüfontention, schlank zu halten. Die höhere Granularität der Regeln, die sich aus der Aufteilung einer Gesamtfragestellung in Einzelteile entsprechend der jeweiligen spezifischen Belange ergibt, erhöht dabei die generelle Wiederverwendbarkeit. So können beispielsweise die Vorbedingungen (Grundregeln) zum Prüfen einer bestimmten Elementeigenschaft für weiterführende, spezifischere Fragestellungen an die Eigenschaft, durch Kopie des Regelsatzes übernommen werden. Dabei muss dann nur die letzte Regel des Satzes, wie im nächsten Abschnitt dargestellt, entsprechend angepasst werden. Darüber hinaus ist durch die feingranulare Aufteilung der Gesamtprüfung in einzelne Schritte gemäß den spezifischen Prüfbelangen auch das bedarfsgerechte Aufdecken geeigneter Synthesen von oftmals wiederkehrend verwendeten Grundregeln in entsprechende Hilfsfunktionen (vgl. unten Abschnitt Hilfsfunktionen) gegeben.

3) Stufe der Manipulation

Im Unterschied zu sogenannten Expertensystemen¹ zeichnen sich regelbasierte Modellprüfsysteme durch formalisierte Regeln als Träger der Analyse- und Prüfllogik aus (vgl. Eastman u. a., 2009). Damit wird neben der generalisierten Gewährleistung einer transparenten Nachvollziehbarkeit der Prüfvorgänge, die Anwendung der Systeme auch einer nicht versierten Nutzergruppe ermöglicht. Dies setzt jedoch voraus, dass in der Konzeption des Regelaufbaus sämtliche, ansonsten nur von erfahrenen Nutzer*innen handhabbare Komplexität – sowohl bezüglich der Kenntnis der Modellsyntax und -semantik, als auch der Umsetzung der Prüfllogik in entsprechend ausführbaren Code – mittels gut verständlicher Abstraktion bewerkstelligt wird.

Als oberste Schicht bzw. Kaskade zum Abstrahieren der Komplexität sieht der Lösungsansatz eine gut verständliche inhaltliche Gliederung der Regellogik in logische **Regelkomponenten** vor. Zur Unterstützung dieses einfachen Aufbaus der eigentlichen Regeln wird die logische Struktur mit einer Systematik von selbst zu vergebenden Schlüsselbegriffen untersetzt. Neben der Möglichkeit, die weiterführende Strukturierung der Regellogik gemäß eigener (Domänen-spezifischer) Begrifflichkeit zu definieren, verdecken diese Schlüsselbegriffe hinter sich die eigentliche funktionale Modellabfragelogik.

Zur Realisierung der Modellabfragelogik sollte darüber hinaus idealerweise eine einfach zu erlernende, standardisierte Spezifikationsprache zum Einsatz kommen, welche durch verständliche Schlüsselworte das Gestalten der funktionalen Anweisungen zur Umsetzung der Prüfllogik ohne Kenntnis komplexer Programmiersprachen ermöglicht (ebd.).

¹ In Expertensystemen werden Analysen des Untersuchungsgegenstandes zumeist direkt in der Programmiersprache abgebildet, was zu deren Eingabe, neben der Kenntnis der Modelle, immer auch entsprechende Programmierkenntnisse voraussetzt (Eastman u. a., 2009).

Ohne eine eigene Spezifikations-sprache hierfür zu definieren, wie beispielsweise neuere Ansätze für bestimmte Produktmodelle vorschlagen (vgl. BIMQL; vgl. Kap. 2.3.2), sondern sich an generalisierten Spezifikations-sprachen-Ansätzen orientierend, setzt der Lösungsansatz folgende differenzierte und gestufte Gesamtkonzeption des Regelaufbaus um.

- i. Logische Regelkomponenten – einfach zu erlernende Grundstruktur einer Regel
- ii. Selbstvergebene Schlüsselbegriffe – Systematik zur Verdeckung eigener bzw. generisch aufbereiteter Modellabfragen
- iii. Standardisierte Spezifikations-sprache – Realisierung des Zuganges zum Modellinhalt durch Aufgreifen der an einen Spezifikations-sprachstandard angelehnten Abfragesyntax
- iv. Hilfsfunktionen – Zusammenfassen oftmals wiederkehrender Abfragen bzw. Traversierungen des Modellinhaltes

i. Logische Regelkomponenten – Um einer nicht versierten Nutzergruppe das Verständnis des Aufbaus der Analyse- und Prüflogik zu erleichtern und damit bereits die Anpassung der Inhalte einer bestehenden Regel zu ermöglichen, ohne die Struktur des zugrundeliegenden Produktmodelles wie bspw. IFC kennen zu müssen, sieht der Lösungsansatz eine einfache übergeordnete Grundstruktur der Regellogik vor. Die Spezifikation des Gegenstands und der entsprechenden Prüflogik einer Regel ist hierzu in vier logische Grundbausteine aufgeteilt (Ebertshäuser & von Both, 2013). Die erste dieser Regelkomponenten – die **Hauptelement-Komponente (H)** – ermöglicht dabei die Auswahl der Modellelemente, auf die die Regel angewandt werden soll. Für eine weitere Spezifizierung der, an den ausgewählten Elementen zu prüfenden Sachverhalte, kann beispielsweise durch Ausweisen einer bestimmten Elementeigenschaft in der **Merkmal-Komponente (M)** festgelegt werden. Die eigentliche Prüflogik wird dann mittels der Angabe des gültigen Wertebereichs, bzw. dessen funktionaler Bestimmung, in der **Prüfkriterium-Komponente (P)** definiert. Darüber hinaus kann ein erwarteter Sollwert in der **Soll-Ausprägung-**

Komponente (S) festgeschrieben werden, gegenüber dem dann die spezifische Ausprägung des zu prüfenden Eigenschaftswerts verglichen wird.

An folgendem Beispiel soll dies verdeutlicht werden:

„Der Wärmedurchgangskoeffizient (U-Wert) der Außendämmung als Bekleidung des Bauteils Wand muss geringer sein als der am Gesamtbau teil Außenwand annotierte Gesamtwärmedurchgangskoeffizient.“

Entsprechend benötigte logische Ausprägung der Regelkomponenten:

H: Objektmenge aller Außenwände

M: U-Wert der Bekleidung (Dämmschicht) der Außenwand

P: ist kleiner als

S: Außenwand U-Wert

Spezifizierung des Bsp. anhand der Ausprägung mit IFC-Syntax:

H: `IfcWall` Elemente bei denen das Attribut `IsExternal` gleich „wahr“ (engl. `true`) ist.

M: Attribut `ThermalTransmittance` des zur `IfcWall` in Beziehung² stehenden `IfcCovering` Elements

P: „ist kleiner als“ ausgedrückt mittels des Symbols „<“

S: `IfcWall`-Attribut `ThermalTransmittance`

Wie im Beispiel gezeigt, werden mittels der ersten vier Komponenten Prüfungen des Modellinhaltes ermöglicht. Das Ergebnis der Prüfung eines Modellelements ist dabei prinzipiell wahr oder falsch gemäß den Festlegungen in der Prüfkriterium-Komponente, ergänzt ggf. durch entsprechende Ausprägung der

² Auflösen der Beziehung mittels verschachtelter Anweisungen (siehe Abschnitte zu Spezifikations sprache bzw. Hilfsfunktion)

Soll-Ausprägung-Komponente. Durch Rückgabe der beanstandeten Modellelemente erfolgt nach dem Prüfungsvorgang deren Auswertung und Aufbereitung in entsprechende Prüfergebnisse. Hierfür ist im Lösungsansatz eine **Ergebnisauswertungs-Komponente (E)** vorgesehen, in der durch Befüllen von Lückentexten mit den entsprechenden Modellelementattributen bzw. -eigenschaften eine bedarfsorientierte, natürlichsprachliche Berichtsausgabe zusammengestellt werden kann. Neben einer Möglichkeit zur programmatischen Weiterverarbeitung der zurückgegebenen Liste von (beanstandeten) Modellelementen durch Zwischenspeicherung, können so für jede Regel aussagekräftige Ergebnisberichte zu den erfolgten Modellprüfungen ausgegeben werden. Für den gesamten ausgeführten Regelsatz bilden die Ausgaben zu den einzelnen Regeln dann die Grundlage eines entsprechend zusammenstellbaren Prüfberichts zu den untersuchten Sachverhalten des Planungsgegenstandes. Zur weiterführenden Erläuterung der darin tabellarisch aufbereiteten Ergebnisse der Prüfung können diese jeweils annotiert bzw. mit hinterlegten Nutzerdaten versehen werden, bevor sie als Tabellendokument zur weiteren Verwendung im Planungsprozess gespeichert werden.

ii. Selbstvergebene Schlüsselbegriffe – Im letzten Abschnitt wurde die logische Grundstruktur einer Regel in Form der Regelkomponenten vorgestellt. Eine Verdeckung der Komplexität von sowohl Modell sowie Prüflogik hinter diesen Komponenten wird dabei mit einem Prinzip zur Ablage von Abfragelogik in Schlüssel-Wert-Paaren untersetzt. Die zugrundeliegende Idee hierzu wurde in einem Entwicklungsprojekt (Ebertshäuser & von Both, 2013) erstmalig vorgestellt, auf das im Kapitel 3.2.2 von Seiten der technischen Konzeption näher eingegangen wird.

Konzeptionell bauen die Regelkomponenten dazu auf einer Systematik von selbst zu vergebenden Schlüsselbegriffen auf. Der Lösungsansatz ermöglicht dadurch die fachspezifische Ausprägung der Regelsemantik gemäß den Anforderungen der Begrifflichkeit einer bestimmten Disziplin. Hinter diesen Schlüsselbegriffen stehen dann die benötigten, ausführbaren Algorithmen bzw. funktionalen Anweisungen zur Überprüfung des Modellinhaltes. Liefert der erstmalige Aufruf eines Schlüsselbegriffes dabei durch das Ausführen der hin-

terlegten funktionalen Anweisung einen konkreten Wert bezüglich der Modellinstanz zurück, so ermöglicht die aufgegriffene Systematik auf der technischen Seite beim wiederholten Aufruf des Schlüssels, direkt den schon berechneten konkreten Wert für die aktuelle Modellinstanz zurückzugeben.

Zum Beispiel besteht der die Hauptelemente-Komponente realisierende Algorithmus aus einer Codeschablone, deren Variablen mit den Schlüsselbegriffen bestückt werden kann. Bei der Ausführung der Regel wird dann die hinter dem ausgewählten Schlüsselbegriff stehende Elementmenge (bspw. „WÄNDE“ – Elementmenge aller Wände einer Bauwerksmodellinstanz), einzeln durchiteriert und dabei das aktuell iterierte Element an die hinter den anderen Komponenten stehenden Schablonen zur Überprüfung spezifischer Sachverhalte weitergereicht.

Durch Kenntnis des Funktionsprinzips einer Regelkomponente – am Beispiel Hauptelement-Komponente: Iterationsschleife über eine angegebene Elementmenge der aktuellen Modellinstanz – wird die Bearbeitung der Inhalte der Regeln ermöglicht, ohne dass der programmatische Aufbau der Schablonen bekannt sein muss. Um mittels der gleichen Regellogik das, beispielsweise allen IFC Elementen gleichermaßen (durch die Basisklasse `IfcBuildingElement`) allozierte Attribut „U-Wert“, zu überprüfen, kann durch das Prinzip der eigenen Schlüsselbegriffe bereits eine nicht versierte Nutzergruppe ein entsprechendes in der Hauptelement-Komponente hinterlegtes Schlüsselwort zum Spezifizieren der Elementmenge z.B. „WÄNDE“ mit „FENSTER“ ersetzen.

Ist für eine Anpassung einer Regel ein avisierter Schlüsselbegriff noch nicht vorhanden, kann neben einer generischen Erstellung von Elementmengen sämtlicher Elemente der aktuellen Modellinstanz (vgl. unten Abschnitt Hilfsfunktionen), nach eigener sinngemäßer Festlegung eines entsprechenden Schlüssels, die dahinterstehende Modellabfrage – wie im folgenden Abschnitt beschrieben – ausgeprägt werden.

iii. Standardisierte Spezifikationssprache – Das oben beschriebene Prinzip von Schlüssel-Wert-Paaren grenzt generell (aus Nutzersicht) die „oberste“ Abstraktionsebene der Analyse- bzw. Prüfllogik von der darunterliegenden funktionalen bzw. programmatischen Umsetzung der Logik ab. Für das Bilden

der Werte zu den Schlüsseln ist bei der rechnerischen Umsetzung der Logik eine weitere Abstraktionsebene im Lösungsansatz vorgesehen. So sind in dieser syntaktisch eigenständigen Zwischenebene, vor der Umsetzung mittels Programmiersprache, die funktionalen Formulierungen von Bedingungen an die zu untersuchenden Modellstrukturen gekapselt. Für die darin festzulegenden, benötigten Zugriffe auf bzw. Bedingungen an die objektorientierten Strukturen der integrierten Produktmodelle, wurde im Ansatz dabei bewusst auf die Definition einer eigenen funktionalen Syntax verzichtet. Indem diese in Anlehnung an eine etablierte Spezifikationssprache (vgl. Kapitel 2.3.2) formuliert werden können, soll die Anwendbarkeit auf multiple Produktmodelle gewährleistet werden. Darüber hinaus sollen Nutzer*innen ohne Programmiererfahrung durch die einfach zu erlernenden funktionalen Ausdrücke beim weiteren Spezifizieren ihrer Prüflöge unterstützt werden. Zum Verfassen der funktionalen Anweisungen (Bedingungen), die für das Berechnen der zu bestimmenden Modellwerte bzw. -mengen benötigt werden, sieht der Lösungsansatz daher - angelehnt an den deklarativen Sprachansatz des verbreiteten OCL Standards (vgl. Kap. 2.3.2; Flake & Müller, 2001) – eine vereinheitlichte Syntax vor.

Die Elementmenge aller Außenwände des obigen Beispiels am Produktmodell IFC kann demnach mit den folgenden erläuterten Teilen eines Ausdrucks (vgl. Tabelle 3), der eine geschlossene Bedingung³ exemplifiziert, aus der Modellinstanz berechnet werden (wobei **modell** selbst schon ein Schlüsselbegriff für die Kollektion aller Elemente der aktuellen Modellinstanz ist):

Tabelle 3 Exemplifizierter Ausdruck mit Schlüsselwörtern der Spezifizierungssprache

Ergebnisvariable	IfcWall ⁴ =
	FORALL (bzw. auch: FROM) element

³ In OCL kann neben der Angabe eines *context* zur Spezifizierung der Elementmenge auf die eine Bedingung angewendet werden soll, eine sogenannte *geschlossene Bedingung* mittels eines Quantors definiert werden (Rumpe, 2011, S. 46).

⁴ Selbstvergebener fachspezifischer Schlüsselbegriff e.g. „Außenwände“ bzw. generischer Schlüsselbegriff gemäß Elementnamen (vgl. nächster Abschnitt) „IfcWall“

Quantor mit Variable, die über qu. Menge variiert	IN	modell
Bedingung 1	WHERE	element.Name == "IfcWall"
Bedingung 2	&&	element.Attribute("IsExternal") == "true"
Spez. Auswahl	SELECT	element

Die in der Beispielkonstruktion einer Anweisung verwendeten Hauptschlüsselwörter sowie das Wissen ihres Funktionsprinzips innerhalb der Gesamtanweisung (z.B. der Quantor *FORALL* . . *IN* als Ausdruck zur Definition der zu untersuchenden Elementmenge über alle Objekte einer angegebenen Klasse bzw. Modellelementmenge) umfassen bereits vollständig die vorausgesetzte Kenntnis zur Anpassung sämtlicher hinter den Schlüssel-Wert-Paaren formulierten funktionalen Logik. Die Aneignung dieses Wissens wird den Nutzer*innen des Weiteren dadurch erleichtert, dass sich die im Lösungsansatz für ein vereinfachtes Beschreiben der Prüflogik definierten, logischen Regelkomponenten in die Struktur einer solchen Anweisung überführen lassen (vgl. Tabelle 4).

Tabelle 4 Zusammenhang Ausdrücke der Spezifikationsprache zur logischen Regelstruktur

Funktionsprinzip	Hauptausdrücke	Log. Regelstruktur
Iterationsschleife	FROM ... IN	Hauptkomponente
Auswahlkriterium	WHERE ... SELECT	Merkmalskomponente
Weitere Bedingungen		Prüfkriterium-Komp.
		Sollausprägung-Komp.

Ein Konzeptschema (siehe Abbildung 18 **Fehler! Verweisquelle konnte nicht gefunden werden.**) hierzu zeigt – die vorhergehenden drei Abschnitte in der Stufe der Manipulation des Regelinhaltes zusammenfassend – den Übergang

von einer natürlichsprachlich verfassten Prüflogik in die, mit der standardisierten Spezifikationsprache untersetzte Konzeption des Regelaufbaus. Auf der rechten Seite sind die drei jeweils seitens der Nutzer*innen vorausgesetzten Kenntnisstufen dargestellt. Mittels Symbolen werden diesen drei Stufen am linken Rand für jede Zeile der im mittleren Hauptbereich tabellarisch exemplifizierten Inhalte die formalen Bestandteile des Lösungsansatzes zugeordnet. In der Kopfzeile ist die natürlichsprachliche Fragestellung abgebildet und wird gefolgt von einer Zeile für die vier Regel-Komponenten sowie einer für selbstvergebene Schlüsselbegriffe. Der Schlüssel dient den Inhalten, die in den beiden darauffolgenden Zeilen die zweite Kenntnisstufe exemplifizieren dabei als eindeutiger Bezeichner, so dass diese aus einer Kollektion dieser Art Regelbausteine abgerufen werden können. Für die obere Zeile symbolisiert ein Blatt mit Stift den bearbeitbaren Ausdruck der Spezifikationsprache und für die untere symbolisiert das Blatt mit Diagramm das entsprechende am Produktmodell ausgewertete Ergebnis des Ausdrucks. In der letzten Zeile, die der dritten Kenntnisstufe zugeordnet ist, symbolisiert ein Bildschirm mit angezeigten Regelkomponenten die Ausführung der Programmiersprache.

	Do all Windows	have an U-Value	less than	one point two?		I
	object selection component	property specification component	operator declaration component	target definition component		Domain Semantic
	„WINDOWS“	„U-Value“	„comparison-less-than“	„numeric-target-value“		
	from <i>obj</i> in model where <i>obj</i> is IfcWindow select <i>obj</i> ;	from <i>prop</i> in object where <i>prop</i> is U-Value select <i>prop</i> ;	<i>[property value]</i> < <i>[target value]</i>	1.2		II Query Language & Domain Models
	collection of all windows	window.uvalue	<	1.2		
	<pre> RuleTemplate () { from item in OBJECT where PROPERTY OPERATOR TARGET select item ; } </pre>					III Code Template & Program Language

Abbildung 18 Schematischer Regelaufbau zur Verdeckung von Komplexität mittels gestuft vorausgesetzten Anwendungswissen

Das zentrale Element der im mittleren Bereich der Abbildung 18 **Fehler! Verweisquelle konnte nicht gefunden werden.** exemplifizierte Inhalte des Regelaufbaus stellt die standardisierte Spezifikationsprache dar, die eine vereinheitlichte und somit generalisierbare Ebene ausbildet. Im Wesentlichen dient sie dabei zur Ausprägung der für die Abfragen an ein jeweiliges Produktmodell benötigten funktionalen Logik gemäß der spezifischen Prüfontention. Zudem fungiert sie als persistier- und wiederverwendbares Verbindungselement der gestuften formalen Abbildung der Prüflogik. Den Regelautor*innen dient dieser Baustein beim leicht zu erlernenden Verfassen von eigenständigen Regelinhalten und ermöglicht ein gezieltes Injizieren von Steuerungslogik in die softwaretechnische Umsetzung (Programmiersprachenebene) dadurch, dass es erst zur Laufzeit der Regelausführung in entsprechende die Berechnungsschritte übersetzt wird. Aus der Perspektive der Softwarekonzeption gewährleistet diese Ebene dabei etablierten Programmiermustern folgend die klare Trennung der Belange und somit Transparenz sowie Wiederverwendung der im Lösungsansatz anwenderbezogen einzubringenden Logik. Bezüglich eines auf dem Lösungsansatz aufbauenden Anwendungskontextes können somit für das beim Erstellen von Regelinhalten einhergehende geistige Eigentum (Intellectual Property) klare Regelungen ermöglicht werden.

iv. Hilfsfunktionen – Dieser letzte Abschnitt zur Stufe der Manipulation befasst sich mit der Konzeption des Lösungsansatzes bezüglich (oftmals) wiederkehrenden funktionalen Anweisungen. Wie am Beispiel der disziplin-spezifischen Spezifikationsprache für das Bauwerksmodell IFC – BIMQL – gezeigt, kann die Modellkomplexität bzw. das benötigte umfangreiche Wissen zu den Konzepten der Modellierung – Modellsyntax und Modellsemantik – innerhalb der funktionalen Sprache verdeckt werden (vgl. Kapitel 2.3.2). Beispielsweise stellt die bidirektionale Kopplung von Modellelementen mittels sogenannter objektifizierter Assoziation ein Grundkonzept der IFC dar. Indem für die durch die objektorientierten Strukturen der Produktmodelle generalisierten Grundkonzepte entsprechende Berechnungsfunktionen in der Sprache hinterlegt sind, können Nutzer*innen die verknüpften Entitäten direkt in ihren

Abfragen aufrufen, ohne die eigentliche Modellierung darin umfänglich auflösen zu müssen.

Mit dem modellübergreifenden Anspruch des Lösungsansatzes, setzt dieser entsprechende Anforderungen zur modellspezifischen Hilfestellung mit dem im Weiteren beschriebenen Prinzip zur (generischen) Bildung von Hilfsfunktionen bzw. Generierung von funktionalen Anweisungen gemäß einer vorliegenden Modellinstanz um.

Das Prinzip greift dabei das Bilden von Methoden - wie aus der Programmiersprache bekannt - auf und erweitert die oben beschriebene Möglichkeit der hinter selbstvergebenen Schlüsselbegriffen stehenden funktionalen Anweisungen. Vor dem Aufruf der Methode über einen entsprechenden Schlüsselbegriff werden deren Argumente als eigene Schlüssel mit den Parameterwerten hinterlegt. Durch die einheitliche Grundkonzeption der Modellierung kann so ein Funktionskörper für wiederkehrende Auflösungen von Modellstrukturen verwendet werden. Am Beispiel des IFC-Modells, welches u.a. die oben angeführten Assoziationsklassen in der Basisklasse der räumlichen Modellelemente (Grundstücks-, Gebäude-, Stockwerksklasse usw.) zur Beschreibung der Topologie alloziert, kann somit eine Methode durch die Angabe der übergeordneten Entität jeweils das topologisch darunterliegende Element zurückgeben. Dies ermöglicht das schlanke, zielorientierte Verfassen eigener Anweisungen durch Aufgreifen der Hilfsfunktionen, die ansonsten im Beispiel mindestens drei weitere Modellelement Abfragen (zur Assoziationsklasse `IfcRelAggregates` und zu deren Assoziationsenden `RelatedObjects` bzw. `RelatingObject`) beinhalten müsste.

Um des Weiteren die Erstellung von funktionalen Anweisungen zu unterstützen, sind im Lösungsansatz spezielle (generische) Hilfsmethoden vorgesehen, die keinen direkten Rückgabewert als Output generieren. Als eine Art Meta-Methode erstellen sie zum Beispiel durch Angabe eines bestimmten Modellelements entsprechende Schlüssel-Anweisung-Paare, die dann beim Aufruf die entsprechende Elementmenge zum angegebenen Element zurückliefern. Hierdurch können Workflows der Erstellung von Prüf- und Analyselogik

unterstützt werden. Beispielsweise ausgehend von einer Referenzmodellinstanz, aus der zunächst eine Hilfsmethode alle relevanten Modellentitäten als Namensliste herausliest, die dann als Parameter-Auswahlliste der Meta-Methode dienen. Dadurch wird es ermöglicht, für die benötigten Modellelementmengen einer avisierten spezifischen Anweisung entsprechende Schlüssel-Anweisung-Paare zu generieren, deren Schlüssel dann nur noch in der Anweisung eingetragen werden. Ein umfängliches Wissen zu den generierten funktionalen Anweisungen ist in der Meta-Methode dadurch kapselbar und somit vor den Anwender*innen verdeckt. Allein auf Basis der generierten Schlüsselbegriffe (im Beispiel wären das die Elementnamen, wie „IfcWall“, welche die Wandelementmenge aus der aktuellen Modellinstanz zurückerliefert) können diese ihre spezifischen Anweisungen formulieren.

Darüber hinaus können durch die Kapselung der modellierungstechnischen Details in den Hilfsfunktionen, bei entsprechender Weiterentwicklung der Produktmodelle, ohne die Funktionalität und Ausprägung der die Hilfsfunktionen aufrufenden Anweisungen zu ändern, benötigte generelle Anpassungen rein auf Ebene der Hilfsfunktionen vollzogen werden. Zum Beispiel sind beim Versionsprung von IFC2x3 zu IFC4.2 auch bestimmte, die Assoziationsklasse betreffende Grundkonzepte geändert worden. Durch entsprechende Änderung auf Ebene der Hilfsfunktionen können darauf aufbauende Prüf- und Analyseregeln dabei „Versionsunabhängig“ gehalten werden.

4) Stufe der Umsetzung

Bisher beschriebene Konzeptionen des Lösungsansatzes bereiten die grundlegende Trennung und eigenständige Erstellung bzw. Manipulation der Prüf- und Analyselogik durch die Anwender*innen vor. Dem Aufgreifen standardisierter Produktmodelle zum Austausch der eigentlichen Planungsgegenstände im Sinne eines modellbasierten Arbeitens wurde damit zur Unterstützung eine offene Methodik zum nachvollziehbaren Handhaben der Modellinhalte beiseitegestellt.

Um das volle Potential des Ansatzes für eine transparente praktische Anwendung zu schöpfen, wird über die Offenlegung der logischen, regelbasierten Konzeption hinaus auch das Kapseln und Ermöglichen eines Austausches der

Prüf- bzw. Analyselogik bei der technischen Umsetzung erforderlich. Hierdurch kann das Benefiz bezüglich der auf Analysen der Produktmodelle basierenden Entscheidungsunterstützung gemäß den erkannten prozessbasierten Bedarfen (beispielsweise einer dem Planungsprozess vorgelagerten Vertragsgestaltung) einem ganzheitlichen Einsatz in kooperativen Planungsprozessen zugänglich gemacht werden.

Konzept: Regelsatz bzw. Regel als eigenständiges Format

Der Lösungsansatz sieht daher das Auslagern der modellspezifisch und anwendungsbezogenen Prüf- und Analyselogik in eigenständige Regeldateien vor. Eine die logischen Konzeptionen beschreibbar machende Auszeichnungssprache auf Basis der weit verbreiteten XML Notation gewährleistet dabei neben der direkten Möglichkeit des dokumentenbasierten Austausches der Regeldateien eine geeignete Grundlage, auch bezüglich der Weiterentwicklung, beziehungsweise einer Standardisierung. Diese auf der Seite der Anwendung des Ansatzes wichtige Erweiterungsebene und offene Persistierung der ‚Geschäftslogik‘ der Nutzer*innen, ermöglicht dabei auch eine externe Bearbeitung in bestehenden XML Editoren oder sich an den Lösungsansatz anschließenden Werkzeugansätzen. Auf Seiten des Lösungsansatzes selbst beschreibt das Format für das Werkzeugkonzept die formalisierte Schnittstelle zwischen den grundlegenden Ebenen – Werkzeugentwicklung und Entwicklung der Anwendungslogik.

Somit ist für die im folgenden Unterkapitel weiter ausgeführten technischen Grundlagen des Lösungsansatzes der konzeptionelle Teil zur Bewältigung der behandelten modellinhärenten bzw. die Modellhandhabung betreffenden Komplexität vollständig spezifiziert und in eine geeignete Form überführt. Die Spezifikationen der Regelkonzeption in Bezug auf die Anwendung des Lösungsansatzes werden in Tabelle 5 zusammengefasst. Dabei sind die abgestuften Anforderungen und der Bezug auf den „vorausgesetzten“ Wissensstand der verschiedenen Nutzer*innen (Anwender/Entscheider, Regeladministrator/-modellierer, Entwickler von Regeln und Hilfsfunktionen) dargestellt. Dies verdeutlicht die Flexibilität der auf dieser Konzeption möglichen Anwendungs-

entwicklung im Zuge der Anpassung auf unterschiedliche Anwendungskontexte. Und sie ermöglicht das Ableiten und die weitere Spezifizierung der für die Umsetzung benötigten technischen Grundlagen.

Tabelle 5 Übersicht der gestuften Anforderungen bezüglich der Anwendungsebenen des Regelkonzepts

Anwendungsebene	Vorausgesetzter Kenntnisstand	Konzeptionelle Anforderung
Ausführen einer Regel	-	[A1] Sequentiell geführter Arbeitsablauf
Verwalten der Regeln	Übergeordnetes Strukturierungsprinzip der Regel	[A2] Zusammengehörige Regeln in Regelsätze einordnen (eine Regel kann von mehreren Sätzen referenziert werden)
Anpassen vorhandener Regeln	Logischer Aufbau der Regel, ggf. Modellsemantik (Namen der Modellelemente, die in der Regel verwendet werden)	[A3] Logische Dekomposition der Regelinhalte in sinngemäße Komponenten, die jeweils in ihrer Wertausprägung geändert werden können
(Re-) Konfigurieren einer vorhandenen Regel	Logischer Aufbau der Regel, Modellsemantik (Namen bzw. Attribute/Inhalt der Modellelemente, die in der Regel verwendet werden), Kenntnis der Hauptverben Spezifikations-sprache	[A4] Möglichkeit des Änderns sowie Erstellens von Anweisungen in designierter Spezifikations-sprache
(Neu-) Erstellen einer Regel	Logischer Aufbau und technische Grundstruktur der Regel, Modellsemantik und Modellstruktur und Kenntnis der Spezifikations-sprache	[A5] Möglichkeit der Anpassung der Regelschablone zur Orchestrierung der einzelnen funktionalen Anweisungen

Erstellen / Zusammenfassen von Hilfsmethoden	Modellsemantik und Modellstruktur, Kenntnis der Spezifikationsprache und rudimentäre Programmierkenntnisse	[A6] Anpassen/Anlegen von eigenständigen Methoden zum Traversieren der Modellstruktur (z.B. Verknüpfung einer Eigenschaft an Element)
--	--	---

3.2.2 Konzeptionelle Festlegungen zur Werkzeugentwicklung

Auf technischer Seite erfordern die dargelegten Konzepte und Methoden des Lösungsansatzes zum einen das Aufgreifen von generischen Methoden aus der Informatik. Darüber hinaus ergänzt zum anderen die Einbeziehung weiterer – neben der vorgesehenen Einbindung einer standardisierten Spezifikationsprache beim Formulieren der Prüf- und Analyselogik – etablierter Standards die Generalisierbarkeit des Lösungsansatzes. Zur besseren Einordnung dieser konzeptionellen Festlegungen, auf denen dann die realisierte Gesamtarchitektur des Ansatzes der Softwareanwendung mit den einzelnen Werkzeugmodulen im Umsetzungskapitel (Kap. 5) beruht, werden folgend gewichtige Aspekte sowie ihr Zusammenhang zu den im vorhergehenden Kapitel konkretisierten Anforderungen auf Ebene der Prüf- und Analyselogik des Lösungsansatzes sowohl an den Gegenstand ‚Produktmodell‘ als auch dessen (kooperative) Handhabung dargestellt. Dabei werden ausgehend von der konzipierten regelbasierten Übergabe der Prüf- bzw. Analyselogik an die in diesem Kapitel weiter konkretisierte Ebene des Anwendungsansatzes, für die eine formalisierte, XML-basierte Schnittstelle vorgesehen ist, entsprechende einbezogene technische Konzepte vorgestellt sowie informatische Entwurfsmuster erläutert, auf denen konzeptionell aufgebaut wurde.

1) Schnittstelle zwischen konzeptionellen Ebenen

XSD Regel/Regelsatz

Als Ausgangspunkt der maschinen-basierten Verwertung der von Seiten der Anwender*innen mittels der beschriebenen Regelkonzeption eingebrachten

funktionalen Logik steht die datentechnische Definition der Schnittstelle zwischen den beiden konzeptionellen Ebenen – der Methoden und Hilfsmittel für Analyseregeln bzw. Prüflogik einerseits sowie dem dynamischen Anwendungsansatz mit Werkzeugen für modellbasierte Analyseprozesse andererseits (vgl. 3.1). In Form von zwei XML Schemata (XSD) – dem Regelsatz-Schema als strukturellem Gruppierungsformat sowie dem Regel-Schema als formalisierte Grundstruktur einer Regel-Logikeinheit – wird diese Schnittstelle im Lösungsansatz realisiert. Zunächst bedarf es zur Interpretierung dieser Formate entsprechende XML-Parser, für deren Erstellung, wie im Kapitel 5 der Umsetzung des Lösungsansatzes weiter ausgeführt wird, Standardtechnologien im Umfeld der Programmiersprachen existieren. Die eigentliche (generische) Herausforderung stellt die weitere Auswertung des mittels entsprechendem Parser eingelesenen Inhaltes der Regelsätze bzw. Regeln dar.

So geht es zum einen generell um die dynamische Verwertung des mit der Regelsemantik beschreibbaren Quelltextes zur Prüfung und Analyse der Produktmodelle – also dessen Kompilieren und Ausführen während der Laufzeit. Dies ermöglicht neben dem Verarbeiten bereits vorhandener Regeln auch die im Konzept der Manipulation der Prüflogik (vgl. Kap. 3.2.1) beinhaltetete Anforderung des Anpassens der funktionalen Anweisungen durch die Anwender*innen (laufzeitbezogen) zu bewerkstelligen.

Darüber hinaus erfordert die vollständige Umsetzung des Prinzips der von den Anwender*innen selbstvergebenen Semantik (Schlüssel-Wert-Paare, siehe Kap. 3.2.1) einen Mechanismus für die Interpretation dieser über die Regelsyntax hinausgehenden Semantik (Schlüsselbegriffe). Die im Konzept beschriebenen Teile dieses Ansatzes – Schlüssel-Wert-Paar sowie dem, entsprechend den Wert berechnenden (funktionalen) Ausdruck – wurden in einem technischen Konzept eines „lernenden“ Datenbehälters (Objektspeicher) zusammengeführt. So ermöglichen – wie in den Grundlagen dargelegt – hohe OO-Programmiersprachen als Basis dieser Art von Objektspeicher im Kontext von Aufrufen von Feldern bzw. Methoden einer Klasse auch ein direktes Erweitern/Überschreiben hierfür bereitgestellter Grundfunktionen (vgl. Kap.2.3.1). Dadurch wird es ermöglicht, der Instanz einer Klasse, ohne vorher-

rige quelltextseitige Deklaration darin, dynamisch einen entsprechenden Feldnamen bzw. Methodennamen hinzuzufügen. Durch Aufgreifen und Erweitern dieser Möglichkeit (vgl. auch Konzept des Polymorphismus), sodass beim erstmaligen Aufruf des neuhinzugefügten Feld- bzw. Methodennamens zunächst ein entsprechendes Skript ausgeführt wird, konnte ein Datenbehälter konzipiert werden, der es ermöglicht, die geforderten (Rückgabe-) Werte der „dynamisch“ hinzugefügten Schlüssel-Wert-Paare aus den hinterlegten funktionalen Anweisungen am Produktmodell zu berechnen.

Die ersten Ideen zu diesen konzeptionellen Festlegungen bezüglich der Bedienung von Anforderungen des Lösungsansatzes dieser Arbeit sind dabei im von der Deutschen Forschungsgemeinschaft (DFG) geförderten Projekt „Ein modellgetriebener Ansatz zur Integration von Produktmodellen in domänenübergreifende Analyseprozesse“ (Hartmann & Both, 2015) entstanden. Im Rahmen des Entwicklungsprojektes „ifcModelCheck“ (Ebertshäuser & von Both, 2013) wurden folgend zusammengefasste, darauf aufbauende technische Grundbausteine weiterentwickelt und vorgestellt. Ihre Einbindung in die Gesamtarchitektur bzw. technische Realisierung als (komponentenbasierte) Basis der Werkzeugentwicklung werden im Umsetzungskapitel (siehe Kap.5) weiter ausgeführt.

- a) Grundbaustein zur dynamischen Kompilierung – Quelltext-Schablonen, die zur Laufzeit ausgeprägt, kompiliert und ausgeführt wird
- b) Grundbaustein dynamischer Objekt- Speicher – Datenbehälterklasse, deren Instanz dynamisch um Schlüssel-Wert-Paare erweitert wird. Die beinhalteten Werte (Objekte) können mittels des Schlüssels dann in einfacher Punktyntax⁵ (z.B. Datenbehältername.Schlüssel) abgerufen werden

Im Zusammenspiel der beiden Grundbausteine können funktionale Anweisungen anstelle von statischen Werten (Objekten) hinterlegt werden, die beim Aufruf des entsprechenden Schlüssels dann dynamisch berechnete Werte (Ob-

⁵ im Gegensatz zum Aufruf eines Wertes in einem „einfachen“ Dictionary (Standard Datentyp eines Objekt-Speichers der Programmiersprache) mittels Index (e.g. Datenbehälter[Schlüssel])

jekte) zurückgeben. Neben dieser Möglichkeit, statische Anweisungen für bestimmte dynamische Einzelwertermittlungen zu hinterlegen, ermöglichen darüber hinaus parametrisiert hinterlegte Anweisungen (Schablone mit Platzhalter), wie bei einer Methode eine (dynamische) Wiederverwendung. Die entsprechenden Parameter (Platzhalter in der Anweisungsschablone) werden hierfür ebenfalls als Einträge im Datenbehälter in Form von Schlüssel-Wert-Paaren hinterlegt und für wiederkehrende Berechnungen jeweils mit entsprechenden Argument-Werten belegt und dann von der parametrisierten Anweisung während der Ausführung abgerufen.

Des Weiteren ermöglicht eine Formalisierung des Datenbehälters als XML Schema (XSD) „intrinsische Marken“ (Schlüssel-Anweisung-Paare) XML-basiert zu Persistieren und somit für bestimmte Anwendungskontexte (z.B. bezüglich spezifischer Modellformate) entsprechende Kollektionen funktionaler Logik vorzuhalten. Hierauf aufbauend wird im Ansatz der die funktionale Logik betreffende Funktionsumfang der Softwareanwendung bedarfsorientiert erweiterbar.

2) Architektur zur Trennung der Belange: Präsentationsschicht (Benutzeroberfläche), technische Geschäftslogikschicht, Datenzugangsschicht

Im Lösungsansatz werden diese auf Seiten des technischen Ansatzes entwickelten bzw. gewählten Grundkonzepte synthetisiert und in einer übergeordneten Architektur zusammengestellt. Dabei wird, ausgehend von dem generellen Paradigma der objektorientierten Programmierung (OOP), das ursprünglich aus der Webentwicklung stammende (und in verschiedenen Ausprägungen weit verbreitete) Entwurfsmuster Model-View-Controller (MVC) zur Trennung der Implementierung der Präsentationsschicht von der technischen Geschäftslogikschicht des Werkzeuges bzw. dessen Datenzugangsschicht umgesetzt. Neben der oben beschriebenen generellen Trennung der Prüf- und Analyselogik von der Implementierung des Werkzeuges, ermöglicht dieses Architekturkonzept die flexible Anpassung an verschiedene Kontexte. Beispielsweise kann die in dieser Arbeit verfolgte Implementierung als eigenständige Anwendung (Desktop-Application) entlang der „Sollbruchstellen“ der Schichtenarchitektur entkoppelt und als Modul wieder zusammengesetzt

werden (vgl. Kap. 5 sowie Evaluationen des Ansatzes im Forschungsprojekt ISIS in Kap. 7.2).

3) Schnittstellen zu Nutzerdaten/-systemen: Keylists Import, CMIS

Ergänzend zu der “inneren” Schichtenarchitektur des Werkzeugansatzes ermöglichen weitere standardbasierte und lose gekoppelte Schnittstellen zum einen die bedarfsorientierte Einbindung des Werkzeuges in vorhandene informationstechnische Strukturen auf Seiten der designierten Anwender*innen, zum anderen das Einbringen weiterer nutzer- und anwendungsbezogener Informationen. Durch Bedienung des *Content Management Interoperable Standards* (CMIS) als Schnittstelle zu Content Management Systemen (CMS) können so, neben der Verarbeitung von lokalen Modelldaten, zu prüfende Modelle bzw. Prüfberichte direkt in bestehenden Datenhaltungssysteme abgerufen bzw. abgelegt werden. Insbesondere kann derart das bei der Datenhaltung wichtige konsistente Versionieren verschiedener Planungsstände im Prozessverlauf durch Einbindung bestehender Lösungen unterstützt werden, ohne eigene - und damit einhergehende Aufwände der Konsistenzwahrung - Lösungen entwickeln und anwenden zu müssen. Die in der anwendungsbezogenen Architektur seitens des Werkzeugansatzes angebotenen Datenaustauschschnittstellen werden im Rahmen des Lösungsansatzes konsequent auch auf Ebene der von den Anwender*innen gestaltbaren Prüf- und Analyselogik genutzt.

Für die auf dieser Seite extern verwalteten, bereits bei den Anwender*innen bestehenden Informationsressourcen und Richtlinien zum Gegenstand ‚Produktmodell‘, wie beispielsweise Klassifizierungen oder Dokumentenrichtlinien der Öffentlichen Hand (vgl. Koggelmann u. a., 2012) sind dazu im Lösungsansatz entsprechend datenbasierte Einbindungskonzepte in die Prüf- und Analyselogik vorgesehen. So wird es beispielsweise ermöglicht, eine TABELLendatei mit nutzerspezifischen Codelisten im Zuge einer die Kodierungen in der Modellinstanz überprüfenden Regel auszuwerten. Wichtig ist dabei, dass diese Inhalte nicht direkt in der Regellogik festgeschrieben sind, sondern dort

nur über eine Referenz auf einen generischen Endpunkt (z.B. Name der Tabledatei) abgerufen werden können. Damit kann bei geänderten Kodierungsvorgaben ohne Anpassung der die Inhalte aufrufenden Regeln die entsprechend referenzierte Informationsquelle aktualisiert werden.

Zur weiteren Dynamisierung der an die Gesamtarchitektur angebotenen Informationsquellen (z.B. Benutzer- bzw. Projektinformationen, Produktmodelle usw.), greift der Lösungsansatz, über die dokumentenbasierte Schnittstelle CMIS hinausgehend, mehrfach auf das Konzept des „Repository Pattern“ (vgl. Fowler, 2003; Abbildung 19) zurück.

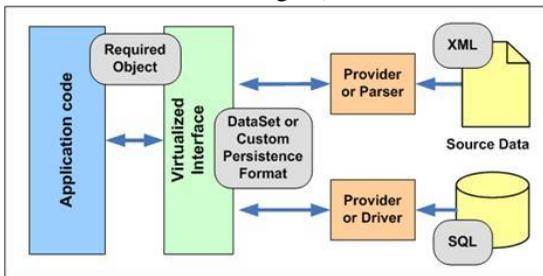


Abbildung 19 Schema des Repository Entwurfsmusters ©SpringPeople⁶

Das an verschiedenen Stellen zur Bedienung unterschiedlicher Anforderungen des Lösungsansatzes einheitlich eingebundene Entwurfsmuster wird beispielsweise auch im Kontext der von den Anwender*innen zum Personalisieren der Bearbeitungsvorgänge hinterlegbaren Nutzerinformationen (Name, Institution, Projekt usw.) angewendet.

Wie schon bei dem Konzept zu Regelsatz bzw. Regel formalisiert dabei ein eigenständiges XSD Schema das in der Business Logik des Werkzeugansatzes vorgesehene Informationsobjekt zu diesen Nutzerdaten. Mit einem Mapping zu auslesbaren Informationsinhalten der von den Anwender*innen angewendeten Benutzerverwaltungssysteme (z.B. Betriebssystemnutzer) unterstützt

⁶ <https://vdocuments.site/springpeople-building-web-sites-with-aspnet-mvc-framework.html>. Zugriff am 23.01.207, (Folie 6)

diese Schnittstellenkonzeption dann feingranulare Anpassung des Werkzeugansatzes an bereits bei den Anwender*innen vorhandenen technischen Infrastrukturen und Umgebungen.

Insbesondere jedoch für die zu bedienenden Produktmodelle stellt das Aufgreifen dieses Entwurfsmusters eine entscheidende Erweiterungssystematik im Lösungsansatz dieser Arbeit bereit. Durch die dadurch realisierbare Entkopplung des Modellzugriffes von den während der Laufzeit verwendeten Objekten, können – wie in Abbildung 19 schematisch dargestellt – verschiedene Datenquellen angebunden werden. Auf Seiten der Business Logik wird hierfür ein einheitlicher Zugriff auf die Modellinformationen für die in der Regellogik festgeschriebenen funktionalen Anweisung unter dem Schlüsselbegriff „Modell“ (vgl. virtuelles Interface in Abbildung 19) im oben beschriebenen Datenbehälter bereitstellt.

Die gemäß des Repository Pattern als virtuelle Schnittstelle (feststehender Begriff unter der die interne Repräsentation des Produktmodells aufgerufen und verarbeitet werden kann) ergänzte generelle Entkopplung der Analyse- und Prüfllogik (mittels der Regelkonzeption) von den sie verarbeitenden Mechanismen, ermöglicht darüber hinaus prinzipiell auch verschiedene Objektgraphen für die Laufzeitobjekte einzubinden. D.h. ein Parsen von beispielsweise XML-basierten originären Modellstrukturen in einen auf die XML Syntax aufbauenden Objektgraphen (bestehend aus den XML Hierarchie und den Elementen, vgl. Domain Object Model – DOM Ansätze; vgl. Kap. 2.3.2) könnte auch durch eine die fachliche Semantik und Struktur des Produktmodelles selbst aufgreifende Objektgraphen (auf Basis einer entsprechend generierten Klassenbibliothek, vgl. XML data binding in Kap. 2.3.2) ersetzt werden. Dies würde auf Ebene der Regellogik das Anpassen eines Ausdrucks beispielsweise zum Abrufen einer Elementeigenschaft (Attribut) erfordern:

Anstelle eines rein funktionalen Ausdrucks:

```
Modell.Element(„ELEMENTNAME“).Attribute(„EIGENSCHAFTS-  
NAME“)
```

Ein Ausdruck mit „Punktsyntax“ (gemäß Produktmodellsemantik):

Modell.Element(„ELEMENTNAME“).EIGENSCHAFTSNAME

Mit der Auswahl des zu prüfenden Modells können dann entsprechend hinterlegte Modelleinlesemechanismen abgerufen werden, um die persistente Modellinformation in Form von den benötigten Laufzeitobjekten in den Datenbehälter zu laden. Der in dieser Arbeit verfolgte, auf die (dokumentenorientierte) Schemasprache der XML-Notation XSD ausgelegte Ansatz lässt sich somit an dieser Stelle zur Bedienung weiterer Basisnotationen bzw. Protokollarten weiterentwickeln. Beispielsweise sind in letzter Zeit für einzelne Produktmodelle sogenannte Modellserver entwickelt worden (vgl. BIMserver.org für das Bauwerksmodell IFC bzw. 3DCityDB⁷ für das Stadtmodell CityGML). Generell verfolgen sie dabei das elementbasierte Bereitstellen der Modellinformation mittels weitverbreiteter Netzwerkprotokolle (z.B. auf Basis des W3C Standard *Web Services Description Language* - WSDL). Dabei stellen diese Protokolle eine netzwerkbasierte Kommunikationsschicht bereit, die die eigentliche (serverseitige) Persistierungstechnologie (z.B. webbasierte Datenbank) von der Anwendungsebene trennen. Im Gegensatz zu dokumentenorientierten (vollständigen) Modellen, wie beispielsweise das ifcXML Schema beschreibt, können darüber elementorientierte Teilmodellmengen (z.B. als XML-Fragmente) ausgetauscht werden. Durch die Entkopplung des Datenendpunktes von der aufrufenden Ebene der Regellogik mittels des Repository Pattern ist in der Gesamtarchitektur bereits eine Möglichkeit angelegt, Modellteile über solche Netzwerkprotokolle zu beziehen (und ggf. in Mengen zusammenzuführen), bevor sie der regelbasierten Modellprüfung zugeführt werden.

Der Lösungsansatz sieht zur Transparentmachung bspw. von auf Basis dieser Art von Protokollen bezogenen Teilmodellen, ein Beschreiben des Ladens von Modellen mit entsprechenden XML-basierten Regeln vor. Dabei handelt es sich im Gegensatz zu den im vorherigen Kapitel vorgestellten Prüf- und Analyseregeln, um eine vereinfachte (dynamisch) erweiterbare Grundstruktur, ähnlich zu dem oben beschriebenen Persistierungsformat des Datenbehälters. Anstelle einer möglichen Befüllung mit den für die Netzwerkprotokolle benötigten Anweisungen bzw. den Ablöseprozeduren der eigentlichen Modellteile

⁷ <http://www.3dcitydb.net/3dcitydb/3dcitydbhomepage/>

aus den Rückgaben von webbasierten Services (Modellserver), wurden diese Regeln zunächst nur zur Beschreibung von simultan geladenen dokumentenbasierten (Teil-)Produktmodellen untersucht. Darüber hinaus würde der Ansatz prinzipiell ebenfalls eine direkte Anbindung der Persistierungstechnologie ermöglichen, z.B. einer als Datenstrom (Stream) bedarfsgerecht bereitgestellten Modellinformation mittels entsprechender Implementierung der Prüf- und Analyselogik zugänglich zu machen. Diese Möglichkeit wurde jedoch im Rahmen dieser Arbeit (vgl. Umsetzungskapitel) nicht technisch umgesetzt.

3.2.3 Konzeptionelle Festlegungen bezüglich des Kontextes

Schnittstellen und Qualitätssicherung für den Datenroundtrip

Im Rahmen von etablierten, offen beschriebenen Produktmodellen wird auf verschiedenen Skalenebenen die Strukturierung von Information gemäß generalisierbarer Informationsbedarfe im Bereich des Bauwesens vorangetrieben (vgl. 2.2.3). Diese Standardisierung von kooperativ verwendeten Datenstrukturen stellt eine wichtige Voraussetzung zum beschriebenen Ansatz dar. Erst durch vereinheitlichte Modellsyntax, in deren Rahmen die Information strukturiert und aufbereitet wird, kann diese auf einheitliche Weise Analyse- und Prüffragestellungen zugänglich gemacht werden. Der Lösungsansatz dieser Arbeit benutzt dazu die Modellsprache zur Ausprägung funktionaler Ausdrücke, die dann in Regeln maschinenverwertbar formalisiert werden.

Maßgeblich bestimmt das Funktionieren dieses, auf Basis des Schemas ansetzenden Prinzips der vorliegenden Arbeit dabei freilich eine – sowohl formal valide wie auch kohärent bezüglich der Semantik angewendete – Modellsprache auf Seiten der entsprechenden Anwendung der Sprache. So bedingt eine *korrekte* Anwendung der mit dem Schema vorgegebenen Modellsyntax, sich ausdrückend in der Wohlgeformtheit damit gebildeter Modellinstanzen, grundsätzlich ob die beschriebenen Inhalte (vollständig) interpretiert werden können. Zur Gewährleistung dieser geforderten Validität der Form der gebildeten

Modelldaten werden begleitend zu den Datenformaten standardisierte Schemaprüfungen auf Ebene der Normung bereitgestellt. Die mit entsprechenden Validierungstools bestimmbare Fehlerfreiheit der Form der Modelldaten, bildet somit den ersten Schritt, um den Gehalt der Modellinhalte richtig fassen zu können. Dennoch würden Prüfregeleln aufgrund zwar formal korrekte jedoch in logisch inkohärenter Weise abgebildeter Information fehlerhafte Ergebnisse liefern. Es bedarf folglich über die Feststellung der Wohlgeformtheit hinausgehend das Überprüfen der sinngemäßen Anwendung der Modellsemantik. Diese als Modellqualität bezeichnete Güte des Modellinhaltes bildet somit die zweite fundamentale Voraussetzung vor einer sinnvollen inhaltlich-logischen Analyse und Prüfung der Modellinstanz. Im Gegensatz zur möglichen automatisierten Überprüfung von sich verschneidenden Geometrien (z.B. durch sog. clash detection), sind Probleme in der semantischen Qualität jedoch oftmals nicht mehr augenscheinlich feststellbar. Sie haben jedoch auf ein modellbasiertes Arbeiten entscheidende Auswirkungen. Sollen beispielsweise im Rahmen der Massenermittlung für Ausschreibungsunterlagen von einem in der Entwurfsphase erstellten Gebäudemodell Mengen abgeleitet werden, würde das syntaktisch gemäß der Modellsprache zulässige Modellieren von Treppenstufen mittels Bodenplatten-Elementen ggf. zu schwerwiegenden Folgefehlern führen (Eastman u. a., 2009).

Durch eine Sicherung der Modellqualität mit entsprechenden Qualitätsregeln, wird dieser wichtigen Anforderung des Lösungsansatzes Rechnung getragen. Insbesondere ermöglicht es der regelbasierte Ansatz dabei, den bestimmten Teilen des Modells, die von den inhaltlich-logischen Fragestellungen betroffenen sind, problemspezifisch inhaltliche Konsistenzprüfungen vorzuschalten. Mit diesen qualitätssichernden Regeln wird somit eine gesicherte Modellkonsistenz gewährleistet, auf der dann Analyse- und Prüfregeleln aufbauen, die zu den eigentlichen Fragestellungen *nachvollziehbare* Ergebnisse liefern. Insbesondere dem Hemmnis hin zu einer modellbasierten Arbeitsweise, wie Studien dazu zeigen (vgl. Abbildung 20), aufgrund mangelnder Zuverlässigkeit der Modellqualität, kann auf transparente Weise durch die somit gegebene bedarfsgerechte Sicherung relevanter Modellinhalte entgegengewirkt werden.

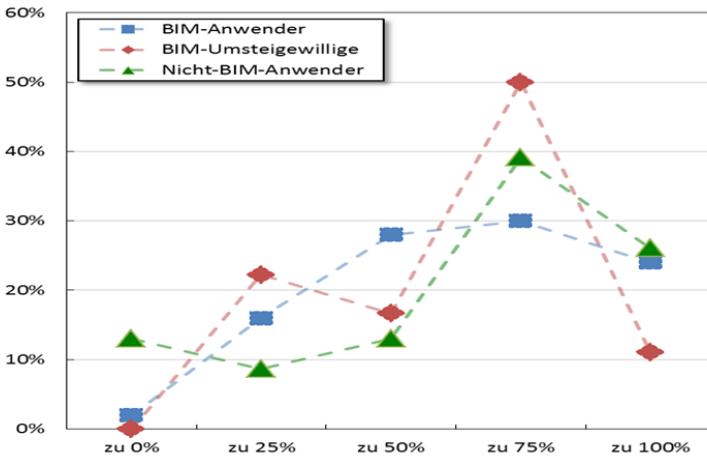


Abbildung 20 Zustimmungverteilung 'Die formale und inhaltliche Qualität von digitalen Gebäudemodellen ist für eine sichere Vertragsgestaltung nicht ausreichend normiert.' (von Both, 2011, S. 159)

Insgesamt bedingt daher eine sinngemäße Anwendung des Ansatzes zur Entscheidungsunterstützung auf Seiten des zu unterstützenden modellbasierten Arbeitens einer strategischen Einbindung des Themas der Qualitätssicherung der Modelldaten. Damit einer noch immer mangelnden Qualität der Schnittstellen, wie entsprechende Befragungen der Akteure der Praxis aufzeigen (vgl. Abbildung 21), noch bevor sie in den eigentlichen Planungsprozessen zu Problemen führen, wirksam für die avisierten Planungskontexte ausgeräumt bzw. entsprechend bedacht und spezifiziert werden können. Explizite Vorgaben an die Modellinhalte, wie beispielsweise Dokumentationsrichtlinien, als gewichtiger Teil dieser Strategien zur Modellqualitätssicherung ermöglichen dabei erst eine Qualitätsprüfung, indem von diesen entsprechend maschinen-verwertbare Regeln abgeleitet werden können. Da die Vorgaben im Planungskontext oftmals an verschiedenen Phasen festgemacht werden – Entwurfs, Werks-, Ausführungsplanung usw. – beschreiben beispielsweise im Bereich des BIM diesbezüglich eine geforderte Modellqualität zu einem bestimmten Datenübergabepunkt (sog. *Qualitygates*) im Prozess.

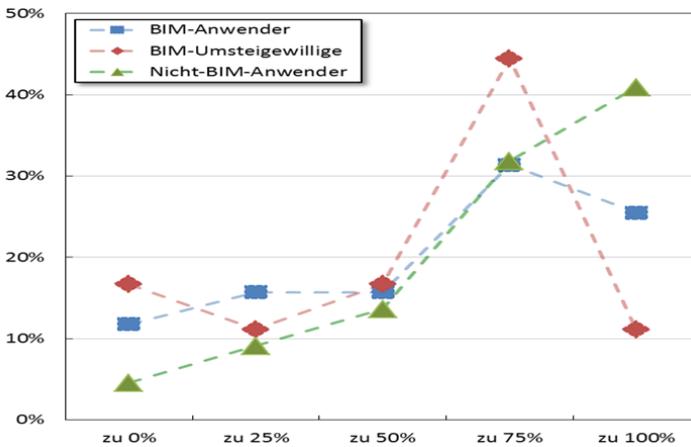


Abbildung 21 Zustimmungsverteilung 'Die verfügbaren Austauschformate für digitale Gebäudemodelle sind nicht durchgängig nutzbar' (von Both u. a., 2013, S. 150).

Diese Bestrebungen des prozess- bzw. phasenbezogenen Bestimmens entsprechender Reifegrade der Modelle kann dabei als „horizontale Sicht“ auf die Modellqualität analog zum (Plan-)Zustand des Bauwerks entlang seines Lebenszyklus gesehen werden. Möglichkeiten, diese „temporalen“ Aspekte der Modellqualität einordnen zu können, werden mit Zuordnungen zu – sowohl generellen als auch konkreten/spezifischen – Projekten im Rahmen des Lösungsansatzes dieser Arbeit vorgesehen.

Daneben kann für die Vielzahl der beteiligten Disziplinen in der Planung und ihre spezifischen Informationsaustauschanforderungen untereinander im Verlauf des Prozesses eine jeweils „vertikale Sicht“ auf die Bestimmung der Modellqualität zugeordnet werden, die es zu berücksichtigen gilt. Um diese Informationsanforderungen bereits auf strategischer Seite in der Modellqualitätssicherung adäquat adressieren zu können, wird beispielsweise im Kontext des Produktmodell IFC der MVD Standard zur formalen IFC-Spezifizierung aufgegriffen. Dieser stellt ein Prinzip zur Fassung bestimmter Partialmodelle des lebenszyklisch ausgerichteten Gesamtmodells IFC bereit und

ist im übergeordneten Rahmen eines (semi-formalen) Informationslieferungshandbuchs (IDM, vgl. Kap. 2.1.4) als ISO Norm standardisiert. So können thematisch dedizierte Qualitätsvorgaben zum Beispiel bezüglich des Teilmodells zur Standsicherheit, unabhängig von denen des technischen Ausbaus bestimmt werden. Zur Einordnung dieser Art von Aspekten bei der Anwendung des Lösungsansatzes dieser Arbeit wurde eine Möglichkeit entsprechende (generische) Kontextregeln zuzuordnen vorgesehen, die regelbasierte Operationen schon beim initialen Zugriff auf die Modellinhalte ermöglichen kann.

Diese wichtigen Aspekte – Möglichkeiten qualitätssichernde Vorgaben an die richtige Stelle im Prozess bzw. an die richtigen Adressaten zu richten – weisen generell auf eine weitere Voraussetzung, den Rahmen der Produktmodellierung betreffend, zur sinngemäßen Anwendung des oben beschriebenen Lösungsansatzes hin. Es braucht ein **Schema für ein integriertes Modell**, das die benötigten Fachaspekte abbilden kann, die zur Entscheidungsfindung betrachtet werden müssen.

Verschiedene Ansätze zur Vernetzung/Integration bzw. Anbindung unterschiedlicher Produktmodelle im Sinne eines Integrierten Produktmodells (IPM) sind erkennbar:

- Auf Ebene der Standardisierung hierzu das Bestreben die Produktmodelle unterschiedlicher Skalenebenen besser zu verknüpfen, z.B. IFC als detailliertester LOD in CityGML zu etablieren
- Thematische Konsistenz durch die Hinzunahme von generellen Klassifizierungen über die Produktmodellensprachen hinweg zu gewährleisten
- Verschiedene (fachliche) Teilmodelle, wie bspw. fertige Bauteilmodelle eines Herstellers, mittels Katalogreferenzen an Produktmodelle anzubinden (vgl. buildingSMART *DataDictionary*⁸)

⁸ <http://bsdd.buildingsmart.org/>

- Mittels Containerformaten wie (proprietäres) *BimObjects* bzw. *BIM Collaboration Format* (BCF) Produktmodellldaten und begleitende Informationen austauschen
- In Form einer übergeordneten Ontologie, bspw. durch *Ontology Web Language* (OWL) realisiert, die in verschiedenen Produktmodellen verteilten Informationen in einen Zusammenhang stellen

Ohne einen der Ansätze auszuschließen und dennoch eine gangbare (Zwischen-) Lösung zum Problem (noch) unvollständiger Produktmodelle bezüglich benötigter Fachaspekte aufgreifend, skizziert folgende Abbildung, wie in dieser Arbeit eine regelbasierte Erweiterung des Produktmodelles hin zu einem Integrierten Produktmodell ansetzt. Dabei umspannt der regelbasierte Lösungsansatz die bestehenden Produktmodelle und stellt zur Integration/ Anbindung neben qualitätssichernden Regeln eine weitere Ebene von Transformationsregeln den eigentlichen Prüf- und Analyseregeln voran. Dadurch kann die avisierte Entscheidungsunterstützung auf Basis der benötigten – und auf diese Art vervollständigten - IPM stattfinden. Noch unberücksichtigte Fachaspekte in den Produktmodellen lassen sich auf transparente, replizierbare Art einbeziehen.

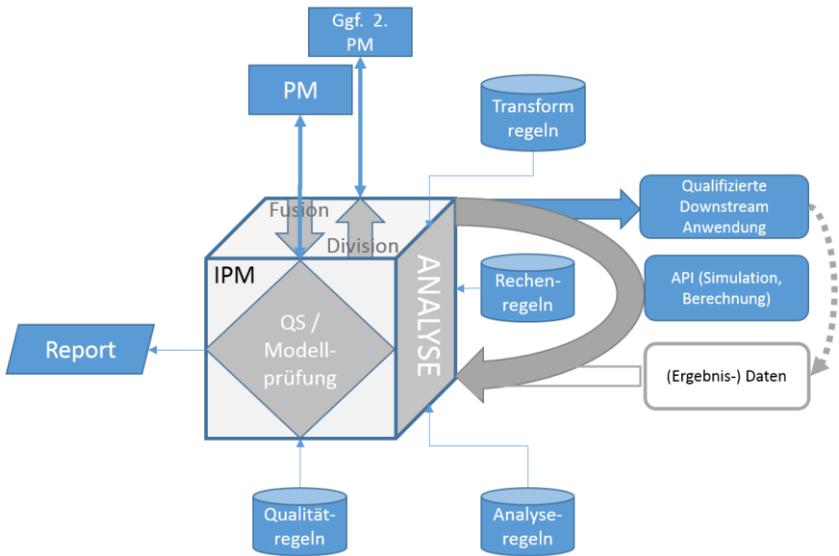


Abbildung 22 Schematische Übersicht zur Gesamtkonzeption

4 Anwendungsszenario

Eine vollständige Umsetzung des in Kapitel 3 beschriebenen Lösungsansatzes in eine spezifische Lösung für ein konkretes Planungsproblem bedarf einer abgestuften Vorgehensweise unter Einbeziehung der designierten Anwenderseite. Im Rahmen dieser Arbeit wurde aufbauend auf einem generischen Ansatz für Prüfungen bzw. Analysen von integrierten Produktmodellen (vgl. Kap. 3.2.2) auf technischer Seite Software-Funktionalitäten umgesetzt, die unterschiedliche Prüfkontexte durch ein flexibles und konfigurierbares Konzept zur Regelprüfung und Regelerstellung ermöglichen (Kap. 5). Darauf aufbauende softwareseitig unterstützte konzeptuelle Methoden und Hilfsmittel zum Verfassen von Analyse- und Prüflogik (vgl. Kap. 3.2.1) ermöglichen dann Anwender*innen die Anpassung dieser Softwareanwendung auf den konkreten Anwendungskontext.

Prinzipiell können so für verschiedene Anwendungskontexte spezifisch ausgeprägte Softwareanwendungen gemäß dem Lösungsansatz umgesetzt werden (vgl. Kap. 7.2). Im Rahmen von Forschungsprojekten konnten beispielsweise Lösungen für den Kontext von Modellqualitätssicherung gemäß BBR Dokumentationsrichtlinie bzw. (fachlichen) Vorgaben zur inhaltlichen Kohärenzprüfungen von Planungsmodellen (Ebertshäuser & Both, 2013 sowie Kap. 7.2.2) umgesetzt und evaluiert werden. Den dabei vorgenommenen Softwareanpassungen auf die behandelten Anwendungskontexte unterlagen jeweils spezifische Anforderungen. Diese bezogen sich inhaltlich auf einen „fertigen“ Planungsgegenstand, beispielsweise am Ende der Genehmigungsplanung. Ein Fokus lag dabei auf dem Prüfen der Vollständigkeit von a priori vorgegebenen, alphanumerisch im Modell hinterlegten Informationen. Bisher konnte dadurch ein wichtiger Teilbereich der im Lösungsansatz verfolgten *gesamtheitlichen* Zielstellung adressiert werden. So ermöglicht erst die Sicherstellung sowohl formal wie inhaltlich überprüfbarer Datenqualität des Planungsgegenstandes eine fundierte Basis, um darauf aufbauend planerische Entscheidungssituationen einer integralen Planung durch Bereitstellen von Planungshilfsmitteln unterstützen zu können. Die vollständig und schemakonform im Modell abgebildete Information stellt dazu den ersten Schritt dar, auf dem dann mittels

geeigneter Modellanalysen weiterführende Zusammenhänge als Entscheidungsgrundlagen ermittelt werden.

Eine evaluierbare Umsetzung des umfänglichen generischen Lösungsansatzes (Kap. 6.2.2.2) erfordert das Spezifizieren eines Anwendungskontextes der Entscheidungsunterstützung und den daraus resultierenden spezifischen Anforderungen. In diesem – möglichst praxisbezogenen – Anwendungskontext können dann auch die situativ in der planerischen Entscheidung auftretenden Fragestellungen spezifiziert werden. So können die Kriterien festgelegt werden, anhand derer die Auswahl der sinnvollsten und weiter zu verfolgenden Entwurfsvariante erfolgen kann. Die konkretisierten Bedarfe der Anwenderseite ermöglichen dann in dieser Arbeit die Konzeption und Umsetzung benötigter Softwareanpassungen und Regelkonfigurationen mit realem Bezug. An ihnen kann dann ein ganzheitlicher Mehrwert des Lösungsansatzes aufgezeigt werden.

Im szenariobasierten Rahmen eines spezifischen Planungsprozesses im Architekturbüro wird in diesem Kapitel mit einer spezifischen Planungssituation die konkrete Grundlage eines solchen Anwendungsfalls des Lösungsansatzes beschrieben, in der mittels praxisnaher Modellprüfungen und -analysen eine planerische Entscheidung unterstützt werden soll. Als Basis der in Kapitel 6 entwickelten Umsetzungsschritte zur Softwareanpassung an diesen szenariobasierten Anwendungskontext werden abschließend die darin beschriebenen Vorbereitungen und durchzuführenden Prüfungsschritte des modellbasierten Planungsgegenstandes in einem konkreten Anforderungskatalog zusammengefasst (vgl. Kap. 4.5). Aus dem Anwendungsszenario fließen dabei in dieses *Lastenheft*:

- methodische Anforderungen – für wen, wann, welche Fragestellungen (generell) relevant sind und wer diese auf welche Weise behandelt
- prozessbezogen formulierte, inhaltliche Informationsbedarfe bezüglich des als virtuelles Produktmodell repräsentierten Planungsgegenstandes

In Bezug auf die zu prüfenden Anforderungen sind generell zwei Ebenen zu unterscheiden:

- Primär geht es im Planungsprozess um (bau)fachliche Anforderungen, welche die planerische Qualität des Bauwerks bzw. Planungsgegenstandes spezifizieren.
- Ergänzend dazu sind aber auch Anforderungen an die Art der Modellierung des virtuellen Planungsgegenstandes zu vereinbaren, die in einem kooperativen Projekt als Mittel der Qualitätssicherung notwendig sind. Hier geht es beispielsweise um die Vollständigkeit des Modells, die Modellstruktur usw. Bei einer solchen modellbasierten Arbeitsweise sind arbeitstechnische Zielstellungen im Rahmen dieses Objektplanungsprozesses⁹ zu vereinbaren, die eine definierte Datenqualität für die Übergaben von Informationen im Prozess regeln und diese absprachelos gestalten. Als Grundlage der Entscheidungssituation spielen sie eine den fachlichen Entscheidungskriterien untergeordnete Rolle, indem sie die Modellqualität bestimmen, auf der dann die Abbildung der für die Kriterien bedeutsamen Modellinhalte aufgebaut werden kann. Im Themenfeld des BIM werden mit AIA sowie BAP vertragliche Rahmenwerke¹⁰ für die Modellqualität bei den Übergaben im Prozess geschaffen (vgl. Kapitel 2.1.4). Die Datenqualität spielt für die in dieser Arbeit verfolgte Entscheidungsunterstützung der Planenden eine dienende Rolle.

Beide Anforderungsebenen können mit dem im Rahmen dieser Arbeit entwickelten Konzept analysiert werden. Der Fokus des Evaluierungs-szenarios und damit auch der weiterführenden technischen Umsetzung wird auf die Ebene der primären fachlichen Qualitätszielsetzungen bezüglich des Planungsobjektes gelegt.

⁹ beispielsweise eine Entwurfsplanung, mit der ein Architekturbüro beauftragt wurde

¹⁰ siehe hierzu Richtlinienarbeiten, wie beispielsweise VDI 2552 Blatt 10

(<https://www.vdi.de/richtlinien/details/vdi-2552-blatt-10-building-information-modeling-auftraggeber-informations-anforderungen-aia-und-bim-abwicklungsplaene-bap>)

4.1 Herleitung objektbezogener Anforderungen der Objektplanung

Damit eine konkrete Entscheidungssituation nachvollziehbar im szenariobasierten Planungsprozess verortet werden kann, müssen zunächst die Randbedingungen, die darauf wirken, geklärt und exemplifiziert werden. In Abbildung 23 wird der durch eine stetige Konkretisierung des Planungsgegenstands geprägte Prozess abstrahiert dargestellt. Der Fokus des Anwendungsszenarios wird wie beschrieben auf die Qualitätszielsetzungen bezüglich des Planungsobjektes gelegt. Hierzu impliziert eine Integrale Planung zum Lösen komplexer Fragestellungen das schematisch in Abbildung 23 dargestellte iterativ planungsbegleitend wiederholte Muster zur Kontrolle des Planungsgegenstandes mittels fachlicher bzw. formal-struktureller Methoden: (1) idealerweise sind schon vor Projektbeginn geplant wann Kontrollen des Planungsgegenstands zu welchen Aspekten stattfinden sollen (vgl. ausgegrauten Kasten *Strategieplan*); (2) jeweils einer Freigabe zur nächsten Konkretisierungsstufe des Planungsgegenstandes vorgelagert, findet der die Planungsentscheidungen unterstützende Kontrollschritt (vgl. mittlerer Kasten *Supportprozess*) im laufenden Prozess statt; (3) im Nachgang des Objektplanungsprozesses lassen sich alle „kontrollierten“ Stände des schrittweise konkretisierten Planungsgegenstands nachvollziehen (vgl. grüner Kasten *Retrospektive*).

So bilden iterative Wiederholungen der Überprüfung der Planungslösung für den gesamtheitlich im Szenario verfolgten integralen Planungsansatz die wichtige Grundlage zum Erreichen der übergeordneten *strategischen* Zielstellungen des Planungsgegenstands. Mit konkreten modellbasierten Zielvorgaben (z.B. ein bestimmter Modellinhalt soll gemäß einer spezifizierten Bandbreite ausgeprägt werden) kann dabei ein genereller Lösungsraum für das Planungsergebnis bestimmt und im Planungsverlauf dann iterativ absprachelos überprüft werden. Das Erreichen eines hochwertigen energetischen Gebäudestandards, wie beispielsweise auch im Rahmen des Nachhaltigen Bauens vorausgesetzt, exemplifiziert hierzu diese konkretisierte Zielstellung für die Entwurfsaufgabe (s.u. Abschnitt *Rahmenbedingung Energieeffizienzkriterium*). Schon zur Entwurfsbeauftragung können so formale Qualitätsanforderungen an den zu erstellenden Entwurf zusammen mit den generell die Qualität

der Modellinhalte bei Übergaben bestimmenden Festlegungen (vgl. AIA usw.) eine nachvollziehbare (vertragliche) Sicherung der bedeutsamen Zielstellungen bilden.



Abbildung 23 Iterativer Prozessschritt Optimierung des Planungsgegenstandes

Gerade bei der Lösung komplexer Fragestellungen im Bereich der nachhaltigen und energieeffizienten Planung ermöglicht eine integrale Planung durch strukturiertes Vorgehen das Erreichen hochwertiger Gebäudeentwürfe, sowie deren Umsetzung. Wichtig dabei ist frühzeitiges Abstimmen und genaues Spezifizieren der (strategischen) Zielstellungen von zu erreichenden Qualitäten für das zu planende Bauwerk. Wie es beispielsweise als Passivhausstandard möglichst konkret schon während der initialen szenariobasierten Projektphase festgelegt wurde. Durchgängiges planungsbegleitendes fachliches Überprüfen und Evaluieren der entwickelten Planungslösungen anhand der gesetzten Zielstellungen unterstützt dann die Entscheider*innen bei der Zielerreichung in den betreffenden Planungsentscheidungen und fördert insgesamt eine optimierte Planungsqualität. Die Umsetzung einer dabei implizierten integralen Planung bedarf über das klare a priori Definieren der (strategischen) Zielstellung zum Planungsgegenstand hinausgehend auch des Ableitens von (messbaren) objektbezogenen Anforderungen der Ziele je nach Konkretisierungsgrad: Zum Beispiel eine charakterisierende Kenngröße, zu deren Bestimmung ein

fachlich bestimmtes Verfahren existiert, wie der Wärmedurchgangskoeffizient (U-Wert) eines Bauteils bzw. -gruppe. Idealerweise beziehen sich diese Zielgrößen und ihre berechnungstechnischen Basiswerte dann bereits in einer entsprechenden Spezifizierung auf eine (neutrale) einheitlich und durchgängig im Prozess verwendete, formale Abbildung der Planungsinformationen. Mittels eines im Rahmen der BIM Methode beschriebenen offenen virtuellen Modellstandards lassen sich Informationsstrukturen für die Abbildung des Planungsgegenstandes vereinbaren. So kann dies beispielsweise im Rahmen einer Entwurfsplanung, zwischen den Bauherr*innen und den Planer*innen vor der Auftragsvergabe geschehen. Diese als AIA in der neueren Normung definierten informationstechnischen Grundlagen bilden auch die Basis der in diesem Kapitel beschriebenen Prüfungs- und Analysevorgänge zur Unterstützung einer konkreten Entscheidungssituation (vgl. Kap. 4.2). Die konkreten, energiebezogenen (logischen) Kriterien an den Planungsgegenstand nehmen dabei Bezug zu dessen, bereits im Rahmen von nachgelagert exemplifizierten BIM-Qualitätssicherungsschritten (vgl. Kap. 4.4.1) überprüfter, formaler Abbildung in einem offenen BIM-Modellformat. Hierzu im Projektszenario angesetzte BIM-Anforderungen bezüglich der generellen Abbildung (Struktur/Topologie und Semantik) des Planungsgegenstands im virtuellen Bauwerksmodell werden in Form von BIM-Kriterien anhand eines Bauteils exemplifiziert (vgl. Kap. 4.3). Im szenariobasierten Gesamtprojektrahmen gültig, unterliegen sie dabei jeder einzelnen (fachlichen) Entscheidungssituation.

Zum einen wird eine einheitliche Informationsbasis durchgängig im Planungsprozess prinzipiell benötigt, um bei zunehmender Komplexität Architekt*innen, denen u.a. die inhaltliche Kontrollpflicht sämtlicher Planungsinformationen obliegt, Mittel an die Hand zu geben in der modellbasierten Arbeitsweise ihren Pflichten nachzukommen. Dies sind beispielsweise die Entwicklung von (logischen) Prüfplänen bzw. -routinen. Zum anderen können erst auf einer einheitlichen (formal bestimmten) durchgängigen Datenbasis Hilfsmittel aufbauen, wie das in dieser Arbeit verfolgte regelbasierte Planungstool zur planungsbegleitenden, absprachelosen (maschinenlesbaren, semi-automatisierten) Entscheidungsunterstützung.

Das komplexe Zusammenspiel einer Vielzahl zu beachtender (disziplinübergreifender) Aspekte stellt die Planung und Errichtung nachhaltiger, energieeffizienter Gebäude nach wie vor enorme Herausforderungen. Anhand der schrittweisen Etablierung zukunftsweisender Lösungsansätze wie beispielsweise konstruktiv-orientierter (Passivhaus) oder materialbezogener (Holzrahmenbauweise) Bauweisen kann beobachtet werden, dass ein Hochskalieren der davon unlängst im (überschaubaren) kleinen Rahmen (z.B. dem Einfamilienhaus) solide abgeleiteter Lösungen, mit einer Multiplikation von (kostentechnischen, technischen, regulatorischen usw.) Problemen¹¹ einhergeht. Neben den technischen Herausforderungen (wie beispielsweise einer systemgerechten Auslegung und Einstellung eines optimierten Zusammenspiels verschiedener Anlagenteile), können dies auch fundamentale Herausforderungen sein wie ein erforderliches Aufweiten regulatorischer Rahmenbedingungen (beispielsweise über ein zur Umsetzung einer innovativen Konstruktion benötigtes materialtechnisches Nachweisverfahren).

Auch wenn für die Evaluierung des hier entwickelten Lösungsansatzes auch ein Bezug zu weiterführenden (Simulations-) Ansätzen von planerischen Entscheidungshilfen möglich wäre, soll jedoch ein praxisrelevantes Szenario gewählt werden, das eine entsprechende Breitenwirkung bzw. Marktdurchdringung aufweist. Daher soll die Unterstützung der Planenden bei der Beherrschung der durch die regulatorische Seite vorgegebenen (energiebezogenen) Anforderungen als Anwendungsfeld zur Demonstrierung des in dieser Arbeit entwickelten planerischen Hilfsmittels herangezogen werden. Auf die Anwendbarkeit in fachspezifischen Experten-Werkzeugen in Form eines Ausblickes zum Lösungsansatz (Kap. 8.1) eingegangen, während im Weiteren das

¹¹ beispielsweise bezüglich holzbasierendem Hochhausbau in Deutschland: „Die Bauvorschriften hinken der Technologie und Wissenschaft weit hinterher“ (Zitat Prof. Winter, TU München; Quelle: <https://www.faz.net/aktuell/wirtschaft/wohnen/bauen/holz-hochhaeuser-gebaeude-aus-holz-so-stabil-wie-stahlbeton-15148513-p3.html>, Zugriff von 21.11.2017)

normbasierte und damit praxisnahe Anwendungsfeld der für die meisten Planungsprojekte relevanten Energieeinsparverordnung (ENEV¹²) fokussiert wird.

Ein geeigneter thematischer Ansatzpunkt zur Spezifizierung eines Rahmens für den szenariobasierten Anwendungsfall des Lösungsansatzes dieser Arbeit stellt somit die zunehmend stärker regulierte Energieeffizienz in den Bauvorschriften dar. Methodisch hat hier der Gesetzgeber mit der Einführung der Energieeffizienz-Standards für Bauwerke der Erkenntnis Rechnung getragen, dass ein allzu starres (isoliertes) Regulieren einzelner Aspekte, wie Anforderungen an die Bauwerkshülle oder die Anlagentechnik bzw. deren Energieträger, nicht zu den durch die Regulierung beabsichtigten energetischen Gebäudequalitäten führt. Gemäß der vormaligen einzelnen Referenzwertverfahren in den Vorgaben der nun überholten Verordnungen ließen sich trotz mangelhafter Hüllqualitäten durch (berechnungstechnisch ausgleichenden) Einsatz hochwertiger Anlagentechnik normkonforme Bauwerksnachweise rein rechnerisch erbringen, ohne tatsächlich bezüglich der Energieperformanz effizient zu sein (vgl. Tuschinski, 2015). Durch abgestimmte Vorgaben zu sowohl bilanziertem Energiebedarf als auch energetischer Qualität der Hülle sowie dem normierten Bereitstellen entsprechender (dynamischer) Berechnungsmethoden wurde diese Praxis in den nun gültigen Vorschriften weitestgehend beschränkt.

Auf der Seite der von diesen Regulationen betroffenen Objektplanung können wiederum die Methoden zur Nachweisführung sowie die darin angesetzten Kriterien der Effizienzhaus-Standards frühzeitig genutzt werden. Beispielsweise können sie zur Operationalisierung von Zielstellungen verwendet werden, wie einem zu erreichenden Passivhausstandard. Mit Normwerten sowie deren durch normbasierte Berechnungsmethoden unterlegte (fachliche) Zusammenhänge untereinander lassen sich dabei die bereits vor Beginn der Planung festgelegten Anforderungen (quantitativ und methodisch) beschreiben. Dies ermöglicht es im Sinne einer integralen Planung klare (energetische) Zielstellungen an das Bauwerk zu explizieren. Indem dabei Bezüge auf Vokabeln

¹² vgl. ENEV § 1 *Zweck und Anwendungsbereich* (https://www.gesetze-im-internet.de/enev_2007/_1.html; Zugriff vom 12.08.2017)

einer Modellsprache zur designierten Beschreibung des Planungsgegenstands genommen werden, können eindeutige objektbezogene Anforderungen den Planenden an die Hand gegeben werden. Darüber hinaus können damit für die sich in der Objektplanung im Rahmen von BIM etablierenden Fazilitäten zur Aushandlung abspracheloser Datenübergaben (vgl. IDM/MVD im Rahmen von AIA; vgl. Kap. 2.1.4) zwischen kooperierenden Partnern, strategische (energetische) Anforderungen an den Planungsgegenstand vereinbart werden (vgl. BIM-Kriterien in Kapitel 4.3), die über die Anforderungen zur Bestimmung der Datenqualität hinausgehen.

Das Erreichen eines hochwertigen energetischen Gebäudestandards exemplifiziert somit für den im folgenden Kapitel 4.2 konkretisierten Anwendungsfall einer Entscheidungssituation im Projektszenario eine spezifizierte (strategische) Zielstellung für die Entwurfsaufgabe zur Bestimmung der Bauwerksqualität. Darin wird mit folgender Herleitung und Spezifizierung eines konkreten Energieeffizienz-Kriteriums eine beispielhafte Determinante für die Entscheidungssituation weiter ausformuliert.

Vorgaben zum Vorhandensein bestimmter (energierelevanter) Modellinhalte, wie sie in Kapitel 4.3 exemplifiziert werden, bilden als Träger zugrundeliegender Fachinformation die Grundlage zur darüberhinausgehenden modellbasierten Festschreibung von Zielen zur Energieeffizienz, sowie insgesamt zur energetischen Qualität der Bauwerkshülle. Zu diesen Zielstellungen existieren auf baurechtlich regulatorischer Seite bereits entsprechende Vorgaben zu Energieverbrauch und Wärmeschutz (vgl. ENEV *Referenzgebäude*). Der Gesetzgeber regelt in seiner Verordnung ein rechtliches Nachweisverfahren, das u.a. festlegt, wann die Erstellung eines Energieausweises verpflichtend ist. Im Rahmen dieses geregelten energetischen Nachweises werden zur allgemeinen Steigerung der Energieeffizienz in der gebauten Umwelt schrittweise die Höchstwerte für einen bilanzierten Jahresprimärenergieverbrauch, sowie für den Transmissionswärmeverlust der Gebäudeumfassungsflächen abgesenkt. Da diesen Mindestanforderungen konsistente normbasierte Berechnungsvorschriften (vgl. z.B. DIN V 18599) unterlegt sind, bilden sie einen geeigneten Ausgangspunkt für die im Projektszenario festgelegten Energieeffizienzkriterien. Eine Motivation hierfür könnte auch in einer avisierten Beantragung

staatlicher Förderung liegen. So bedient sich die deutsche Kreditanstalt für Wiederaufbau (KfW) bei ihrem Gebäude-Förderprogramm ebenfalls dieser Vorgaben zur Bestimmung der Förderung je nach Unterschreitung der vorgegebenen Mindestwerte. Der im KfW-Programm entwickelte, sich am Jahresprimärenergieverbrauch orientierende KfW Effizienzhausstandard wurde in der letzten großen ENEC Novelle (vgl. ENEC 2014) aufgenommen. Mit dem rechtlichen Festschreiben als Effizienzhausstandard (vgl. *Anlagenteil* der ENEC 2014) wurde eine seitens der Förderung bereits praktizierte Verbesserung bisher unscharfer Bestimmungen eingeführt. Durch eine stärkere Einbeziehung des Transmissionswärmeverlustes der Gebäudeumfassungsflächen bei den Vorgaben wurde dabei die oben skizzierte Möglichkeit eingeschränkt, wonach allein durch den Einsatz hocheffizienter Anlagentechnik die primärenergetischen Vorgaben ohne Verbesserungen der energetischen Qualität der Bauwerkshülle erfüllt werden konnten. Eine problembezogen fachliche Auseinandersetzung hierzu findet in planungstechnisch optimierten Lösungen dieses Verhältnisses zwischen anlagentechnischer und rumzuschließender Seite statt, die beispielsweise im Passivhausansatz verfolgt werden. Für einen entsprechenden Passivhausstandard werden von Seiten der KfW-Bauförderung folgende Anforderungen zur Klassifizierung des Bauwerks gestellt. Sie dienen als Grundlage der szenariobasierten Festlegungen einer energetischen Zielstellung für die Entwurfsaufgabe.

„Für ein Passivhaus ist der Nachweis gemäß den Bilanzierungsvorschriften für KfW-Effizienzhäuser zu führen. Das heißt, dass der Jahres-Primärenergiebedarf (QP) und der auf die wärmeübertragende Umfassungsfläche des Gebäudes bezogene Transmissionswärmeverlust ($H'T$) des Neubauobjekts auf Grundlage der Bilanzierungsvorschriften für ein KfW-Effizienzhaus 40, 40 Plus oder 55 zu ermitteln sind. Alternativ kann für ein Passivhaus auch das KfW-Effizienzhaus 55 nach Referenzwerten nachgewiesen werden.“ (KfW, 2020)

„Für ein KfW-Effizienzhaus 40 gelten diese Bedingungen (Quelle KfW¹³):

¹³ <https://www.kfw-effizienzhaus-bauen.de/kfw-40-haus.html>; Zugriff 03.05.2019

- Sein Jahresprimärenergiebedarf (...) darf bei höchstens 40 % des Bedarfs eines entsprechenden Referenzgebäudes (...) liegen.
- Der Höchstwert für den Transmissionswärmeverlust (...) liegt bei 55 % von dem eines Referenzhauses.“

(Hartung, 2019)

Im Folgenden wird beispielhaft für den im Szenario zu erstellenden Gebäudeentwurf eine Zielmarke für den Transmissionswärmeverlust der Gebäudehüllbauteile im Rahmen dieser fachlichen Festlegungen zum Passivhausstandard weiter konkretisiert. Deren Bestimmung erfolgt basierend auf den Wärmedurchgangskoeffizienten der Bauteile des Gebäudes. Entsprechende vorgegebene Sollwerte zu dieser umgangssprachlich „U-Wert“ genannten Kenngröße sind in der Tabelle 6 aufgeführt. Zu diesen werden bezüglich der im Kapitel 4.3 veranschaulichten modellbasierten Abbildungen von Wandbauteilen alle weiteren energierelevanten Informationen ausformuliert. Über die BIM-Kriterien hinausgehend exemplifizieren sie dabei im Projektszenario zusätzliche bei der Abgabe des Entwurfes obligatorisch im Planungsmodell abzubildende Entitäten. Hierfür sind sie konkret am Beispiel der Außenwandbauteile spezifiziert und dienen als Basis der Beschreibung der U-Werte, sowie deren Berechnungsgrößen. In Anlehnung an den alternativen *Nachweis eines KfW-Effizienzhauses 55 nach Referenzwerten* (vgl. KfW, 2020) für die Klassifizierung eines KfW-Passivhausstandards geltende vereinfachte tabellarische Vergleichsverfahren sind als Grundlage einer Vorgabe für die Entwurfsaufgabe in der Tabelle 6 (angepasste) geforderte Sollwerte der Bauwerkshüllbauteile gelistet. Diese Sollwerte stellen dabei die szenariobasierte Vertragsgrundlage zu energetischen Entwurfskriterien als Auszug (strategischen Zielstellung) des initialen „Design Briefs“ (Ebertshäuser u. a., 2019) zur Bestimmung der thermischen (Hüll-) Qualität des Planungsobjekts dar.

Tabelle 6 Soll-Referenzwerte

	Bauteile bzw. Ausführung	Anforderung			
1	Dachflächen, oberste Geschossdecke, Dachgauben	U	≤	0,14	W/(m² K)

2	Fenster und sonstige transparente Bauteile	U_w	\leq	0,90	$W/(m^2 K)$
3	Außenwände, Geschossdecken nach unten gegen Außenluft	U	\leq	0,20	$W/(m^2 K)$
4	Sonstige opake Bauteile (Kellerdecken, Wände und Decken zu unbeheizten Räumen, Wand- und Bodenflächen gegen Erdreich, usw.)	U	\leq	0,25	$W/(m^2 K)$
5	Türen (Keller- und Außentüren)	D_U	\leq	1,2	$W/(m^2 K)$
6	Vermeidung von Wärmebrücken	ΔU_{WB}	\leq	0,035	$W/(m^2 K)$
7	Luftdichtheit der Gebäudehülle	n50	\leq	1,5	h-1

4.2 Anwendungsfall des Lösungsansatzes zur Unterstützung einer Entwurfsentscheidung

Wichtig für den in diesem Kapitel beschriebenen Anwendungsfall des Lösungsansatzes zur Entscheidungsunterstützung in einer spezifischen planerischen Entwurfsituation (vgl. Abbildung 24) sind die im vorherigen Kapitel hergeleiteten energetischen (strategischen) Zielstellung (vgl. Tabelle 6) als (fachliches) Entscheidungskriterium. Einerseits bilden die vorab im Projekt-szenario festgelegten Modellinhalte eine einheitliche durchgängige BIM-Datenbasis für die Planungsinformationen und andererseits für diese energetischen Zielstellungen an den Planungsgegenstand eine Beschreibungsgrundlage (vgl. Kap. 4.3). Am Wandbauteil werden zudem allgemeine (methodische) Anforderungen zur Modellprüfung und -analyse in der Entwurfsphase des szenariobasierten Planungsprozesses mit entsprechenden Prüf- und Analysefragestellungen dargestellt (Kap. 4.4.1).

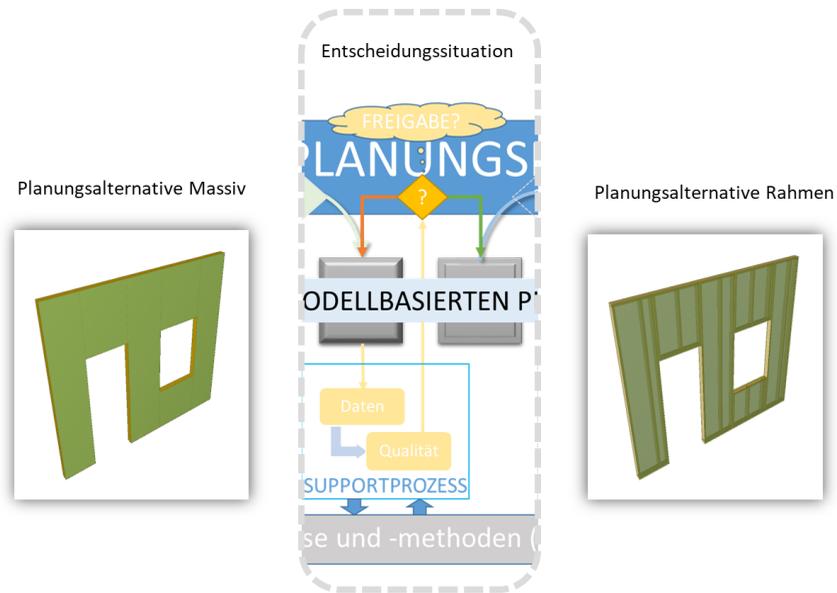


Abbildung 24 Schematische Darstellung der planerischen Entscheidungssituation

Die dabei umrissene genaue Bestimmung der Repräsentation des Planungsgegenstandes sowie entsprechende Kohärenzprüfungen bedarf es dabei um das Entscheidungskriterium „Bauteil soll einen bestimmten U-Wert einhalten“ (planungsbegleitend) als logische an die Planungslösung gerichtete Fragestellung zu operationalisieren. Diesbezüglich spezifizierte Abbildungskonzepte einer offenen Modellsprache bilden daher auch die modell(ierungs)seitige Grundlage für eine Anpassung des Lösungsansatzes an den Anwendungskontext. So werden darin Anforderungen bestimmt, wie beispielsweise zur (semantischen) Abbildung des Wandbauteiles sowohl bezüglich seiner Topologie – Wandobjekt dekomponiert in Wandschichten- bzw. Pfosten/Riegelkomponentenobjekte – als auch bezüglich der auszuprägenden (allgemeinen) Merkmale – spezifische Klassifizierung als “elementierte Wand“.

Aufbauend auf diesen BIM-Abbildungsanforderungen an den Planungsgegenstand wurden im Rahmen dieser Arbeit im Weiteren Modellprüfpläne bzw. Analyseroutinen zu den Entscheidungskriterien entwickelt, die für die Rolle

planender Architekt*innen Hilfestellungen in einer konkreten Entscheidungssituation bieten. Zwei alternative Planungsvarianten bilden dabei den inhaltlichen Ausgangspunkt dieser planerischen Entwurfsentscheidung. Gemäß ihrer konkreten modellbasierten Abbildung werden zunächst die logischen Schritte zur Prüfung/Analyse der angesetzten energetischen Kriterien zusammengestellt. Als natürlichsprachliche Anleitung der Modelhandhabung bilden sie die Grundlage zum Durchführen entsprechender Prüf-/Analyseroutinen, deren Ergebnis dann die Entscheidungsgrundlagen für die Entscheider*innen bildet. Bei einer zielgerichteten Auswahl der weiterzubearbeitenden Entwurfsvariante dienen diese der Unterstützung.

a) Entwurfsalternative mit Porenbeton Massivbauweise

Diese Alternative sieht die Umsetzung des exemplarischen Außenwandbauteils mit einer tragenden Mauerwerksschicht aus Porenbeton vor. Diese ist innenseitig mit einer Gipsputzschicht verkleidet. Auf der Außenseite ist auf das Mauerwerk eine Wärmedämmverbundschicht angebracht, auf die ein Kalkputz aufgezogen ist.

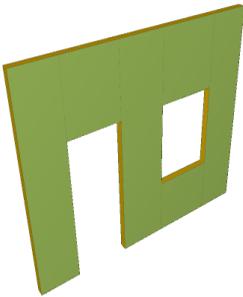


Abbildung 25 Modellabbildung massives Wandelement¹⁴

¹⁴ <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/figures/examples/ifcwallelementcase-solid.png>

b) Entwurfsalternative mit Holzrahmenbauweise

Bei dieser Alternative des exemplarischen Außenwandbauteils ist die tragende Schicht als Holzrahmen, bestehend aus Pfosten und Riegel, sowie einer Dämmung in deren Zwischenräumen, ausgebildet. Als statisch aussteifende Schicht zu dem Holzrahmen fungieren die als Innenverkleidung angebrachten Gipsfaserplatten. Von der Außen-seite sind hinterlüftete Eternitplatten mit entsprechenden Halterungen am Rahmen angebracht.

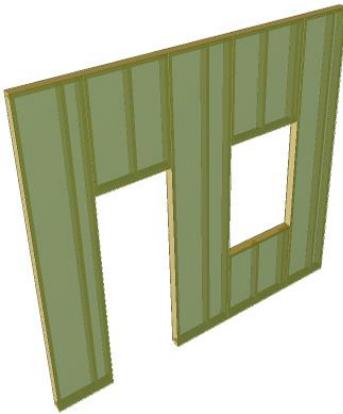


Abbildung 26 Modellabbildung Wandelement in Holzrahmenbauweise¹⁵

Sind die fachlichen Prüfinhalte spezifiziert, so müssen diese in Bezug gesetzt werden zu der Art und Weise, wie der Planungsgenstand digital modelliert wird. Hierzu müssen die am Anfang des Kapitels erläuterten beiden Prüfebene(n) (Inhaltliche Fachlogik und formale Modellstruktur) zusammengefasst werden. In unserem Anwendungsfall geschieht dies wie beschrieben mittels des offenen Austauschstandards IFC (vgl. Kap. 2.2.3). Dieser Schritt wird in

¹⁵ <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/figures/examples/ifcwallelementedcase-trans.png>

Kapitel 4.3. genauer beschrieben. Die bei der Modellierung des Planungsgegenstandes der beiden Modellvarianten für das Wandbauteil verwendeten IFC-Elemente sind allgemein in Tabelle 7 sowie in Tabelle 8 aufgeführt.

Spezifizierung des der Entscheidung zugrundeliegenden Prüfungsprozesses

Zur Spezifizierung der für spezifische fachliche Kontexte notwendigen Informationen hat sich ein Verfahren etabliert (vgl. IDM in Kap. 2.1.4), dass als Vorstufe konkreter Austauschinformationen zunächst die Austauschprozesse und die dabei involvierten Akteure analysiert und dokumentiert. Diese Methode soll auch in dieser Arbeit angewandt werden, so dass zunächst eine Analyse der Prozesse, welche der Entscheidungssituation zugrundeliegen, erfolgt. Dabei werden für die im Szenario benötigte Prüflogik durch eine Analyse des konkreten Prüfprozesses entsprechende Rollen (Treffen der Entscheidung, Erstellung der Entscheidungsgrundlagen usw.) sowie Austauschprozesse zwischen diesen identifiziert. Auf Basis der im Weiteren hierzu prozessbezogen anhand der beinhaltenen methodischen Aspekte konkretisierten Entscheidungssituation, werden die zugrundeliegenden modellbasierten Daten zwar benannt, jedoch zunächst nicht (in ihrer konkreten Ausprägung) weiter spezifiziert. Diese in den Prozessen benannten Daten werden zunächst in Kapitel 4.3 den entsprechenden IFC-Elementen zugeordnet. Daraufhin werden sie im Kapitel 4.5 bezüglich ihrer methodischen Behandlung in Anforderungen zusammengefasst. Dabei dienen diese im Rahmen der Anforderungen konkretisierten Informationsbedarfe als exemplifizierte Inhalte des Lastenhefts zu den Prüf- und Analyseplänen. Indem sie dann als Vorlage zur Anpassung des Lösungsansatzes im Rahmen eines korrespondierenden Pflichtenhefts mit spezifischen Wertausprägungen in konkretisierte Anforderungen umgesetzt werden (vgl., Kap. 6), bilden sie die konkrete (situationsbedingte) Lösung für die regelbasiert erstellten Entscheidungsgrundlagen (vgl. Kap. 6.2.2.2).

Die Entwurfsvarianten veranschaulichen Detaillierungen zum Tragsystem für die ersten, in der frühen Entwurfsbearbeitung aus dem Raumprogramm entworfenen Ansätze des Gebäudeentwurfs. Durch eine Bewertung werden die Varianten gemäß der für die Entwurfsphase geltenden Modellierungsvorgaben (vgl. Kap. 4.3) sowie der Einhaltung der energetischen Zielstellung überprüft.

Dadurch soll eine Entscheidungsgrundlage zur Auswahl einer der Varianten für das weitere Ausarbeiten des finalen Entwurfes aufbereitet werden. Der Entscheidungssituation liegen dabei zum einen der eigentliche Entscheidungsprozess der koordinierenden Ebene (Entwurfsarchitekt*in), sowie zum anderen ein Aufbereitungsprozess der Entscheidungsgrundlage auf der operativen Fachebene (Modellierung und Energieplanung) zugrunde (siehe Abbildung 27).

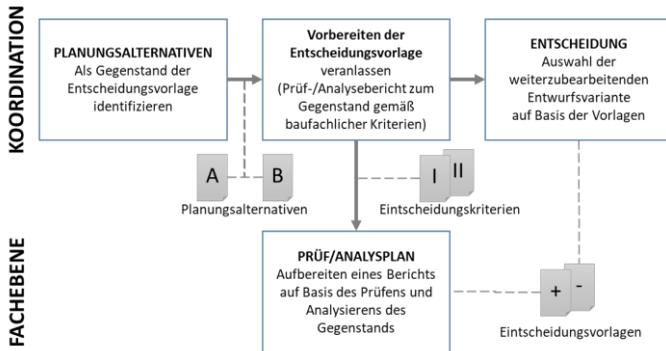


Abbildung 27 Prozess des Anwendungsfalls

Die folgenden konkretisierten Schritte beschreiben hierzu szenariobasiert den spezifischen Ablauf.

Koordinierende Ebene:

- Die zwei entworfenen Planungsalternativen, die als IFC-basierte Planungsmodelle vorliegen, bilden den Ausgangspunkt der Entscheidungssituation für die koordinierende Ebene.
- Die operative Fachebene (Modellierung und Energieplanung) wird mit der Erstellung einer (begründeten) Entscheidungsvorlage beauftragt und soll die beiden alternativen Planungsgegenstände anhand der Vorgaben zur Entwurfsphase (Modellierungsrichtlinie aus Kap. 4.3 sowie energetischer Zielstellung gemäß Tabelle 6) prüfen bzw. analysieren.

Ausführende Fachebene:

- Die Entscheidungsunterstützung wird durch die operative Fachebene geleistet, indem die Entscheidungsgrundlage mit der Ausführung des Prüfauftrages zu den beiden Planungsalternativen gebildet wird.
- Als Ergebnis wird die aufbereitete Entscheidungsvorlage (tabellarische Gegenüberstellung der energetisch bewerteten Bauteile) geliefert

Koordinierende Ebene (Entscheider*innen):

- Abwägen der Entscheidungsvorlage und Auswahl der weiterverfolgten Planungsalternative mit Bezug zu den berücksichtigten Entscheidungsgrundlagen dokumentieren (Entscheidungsbeurteilung).

Wiederverwendbarkeit spezifizierter Prüflösgen

Diese dargestellten, einzelnen Schritte eines Prüfprozesses implizieren eine benötigte (sehr aufwändige) Vorarbeit zur Anpassung des Lösungsansatzes auf die Entscheidungssituation. Daher stellt der Schritt der operativen Fachebene zur Bildung der Entscheidungsgrundlage einen methodischen Ansatzpunkt bezüglich einer möglichen Wiederverwendung dieser Anpassungsarbeit dar. Aus Sicht des Planungsprozesses bildet der *Prüfauftrag* darin zwar nur einen „einzigartigen“ Vorgang, der zur situationsbedingten Entscheidungsunterstützung zu erfüllen ist. Indem das Vorgehen zum Prüfauftrag dabei jedoch weiter aufgeteilt wird, eröffnet sich die Möglichkeit, die Teile der (logischen Vor-) Arbeit in wiederkehrenden Situationen des Gleichen bzw. ähnlich gelagerten Planungsprozesses wiederverwenden zu können (vgl. Kap. 4.4). Ziel ist es, die Regeln so zu strukturieren, dass die entstehenden Regelkomponenten möglichst generalisierbar sind. Daher wird der *Prüfauftrag* für eine szenariobasierte Anbindung eines konkreten Anwendungsfalles in die zwei generellen Vorgehensschritte gegliedert:

- *Bearbeitungsplan* erstellen (nötige Vorarbeit, die durch Wiederverwenden/Anpassen zuvor entwickelter Pläne entfallen kann, falls es sich bereits um eine wiederholte Anwendung des Prüfauftrages handelt)
- *Bearbeitungsplan* durchführen (bildet den eigentlichen Mehrwert für den Planungsprozess, indem als Ergebnis des Prüfauftrages eine Entscheidungsgrundlage aufbereitet wird)

In Abbildung 28 werden als Kontext des Prüfauftrags zwei Prozessebenen dargestellt, eine Ebene des Planungsprozesses (blauer Kasten) sowie eine darunterliegende Ebene der dienenden (wiederkehrenden) Prozessschritte (gelber Kasten). Dabei werden jeweils zu den einzelnen, aufeinander aufbauenden Aspekten des Prüfauftrages – Prüfen, Berechnen, Analysieren und Auswerten – ein entsprechender Datenaustausch zwischen den beiden Prozessebenen zugeordnet. In den Einzelschritten auf Seiten der Bearbeitungsplanentwicklung (Prüf-, Berechnungs-, Analyse- und Auswertungsplan) ergeben sich dann Anforderungen, die als Grundlage zur Anpassung des Lösungsansatzes auf diese spezifische Entscheidungssituation in Kapitel 4.5 zusammengefasst werden. Dabei werden durch Bestimmen entsprechender Modellelemente bzw. der Ausprägung ihrer Werte die beispielhaften Inhalte der schematisch als Informationsaustauschpaare (vgl. A-A', B-B' usw.) in Abbildung 28 dargestellten absprachelosen Übergeben konkretisiert. Ein sinngemäßes Zusammenstellen der Inhalte dieser Informationsübergeben bedarf zunächst die im Folgenden weiter prozessbezogen konkretisierten Datenübergabepunkte, wobei die spezifischen Randbedingungen des absprachelosen Informationsaustausches herausgestellt werden.

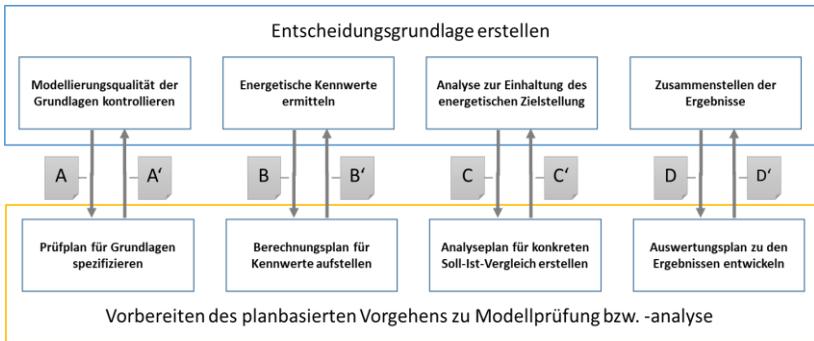


Abbildung 28 Detailprozess zur Erstellung der Entscheidungsgrundlage

Die folgenden Erstellungsschritte eines Prüf- bzw. Analyseplanes werden von dem szenariobasierten Prüfauftrag für die beiden konkreten Planungsalternativen abgeleitet. Jeweils zwischen einer den Prüfauftrag bearbeitenden Rolle und einer die entsprechenden Bearbeitungsschritte „planenden“ Rolle werden hierbei beispielhaft sowohl die ausgetauschten Informationen (vgl. A & A', B & B' usw.) sowie jeweils das Ziel und die Einzelergebnisse zu allen Datenübergabepunkten spezifiziert. Ein dadurch ermöglichtes planbasiertes Vorgehen bei der Erstellung der Entscheidungsgrundlage führt zu einem nachvollziehbar (und replizierbar) aufbereiteten Prüfergebnis zu den alternativen Planungslösungen. Die daraus bezüglich des Planungsgegenstandes herausgestellten Sachverhalte können daher begründet in die Entscheidungsvorlage einfließen.

- A | Modellierungsqualität der Planungsgegenstände kontrollieren
 - Ziel: Vollständigkeit energierelevanter Informationen sicherstellen
 - Ergebnis:
 - inhaltliche validierte Modellqualität
 - Fehleranzeige zur (iterativen) Nachbesserung

- A' | Prüfplan für die IFC-Modellvarianten spezifizieren
 - Dateimetadaten-, Schemaüberprüfung sowie objektbasierte Prüfungen aus dem Prüfplan des Entwurfsergebnisses für die konkreten Planungsalternativen zusammenstellen
- B | Energetische Kennwerte ermitteln
 - Ziel: Berechnungsschritte des energetischen Kennwertes ausführen
 - Ergebnis: konkrete Bauteilkennwerte Planungsalternativen
- B' | Berechnungsplan für die Kennwerte
 - Basisgrößen zu Bauteilschichten/-komponenten unter Einbezug der Abmaße (Geometriebeschreibung) sowie der hinterlegten Materialinformationen ermitteln
 - Normbasiertes Kumulieren der Basisgrößen zu tatsächlichem Bauteilkennwert
- C | Analyse zur Einhaltung des energetischen Zielwertes
 - Ziel: ermittelte Bauteilkennwerte mit der Zielvorgabe vergleichen
 - Ergebnis: Anzeige der Zielerfüllung (ggf. Abweichungsgrad)
- C' | Analyseplan des SOLL-IST Vergleichs
 - Analyseplan des Entwurfsergebnisses auf die konkreten Planungsalternativen anpassen
- D | Zusammenstellen der Ergebnisse der Planungsalternativen

- Ziel: Übersichtliche Aufbereitung der Ergebnisse des Prüfungsauftrages für Entscheider*innenebene, auch um ggf. mittels Detailänderungen (z.B. Materialanpassung) und entsprechender teilweiser (Re-)Iteration der Prüfung die Entscheidungsvorlage schärfen zu können.
 - Ergebnis: Unter dem Entscheidungskriterium (Einhalten des energetischen Zielwertes) für die jeweilige Planungsalternative sind die berechneten Zwischenwerte für Bauteilschicht bzw. -komponenten und die eingerechneten Abmaß- bzw. Materialinformationen abgestuft aufgeführt.
- D' | Auswertungsplan zur Gegenüberstellung der Planungsalternativen
- Es gilt, die Ergebnisse der in der logischen Abfolge ihrer Durchführung aufeinander aufbauenden Prüf-, Berechnungs- und Analysepläne in umgekehrter Reihenfolge zu organisieren. So dass unter dem Entscheidungskriterium der Einhaltung des energetischen Zielwertes (C') die ausgewählten (Zwischen-)Ergebnisse des Berechnungsplans (B') mit den entsprechend einbezogenen Basisgrößen zu Abmaßen bzw. Material (A') abgestuft gelistet werden.

4.3 Weiterführende informationstechnische und methodische Aspekte zum Anwendungsszenario

Dem Anwendungsszenario liegen gewisse (organisatorische) Rahmenbedingungen zugrunde. So wird davon ausgegangen, dass in den Phasen der Projektentwicklung eine Zusammensetzung des Projektteams sowie die operativen Zielsetzungen festgelegt wurden. Bei integral planenden Kooperationspartnern spielen dabei diese operativen Aspekte, also die Art und Weise der Zusammenarbeit bzw. die gemeinsame Arbeit(sinhalte) selbst betreffenden Zielstellungen, eine wichtige Rolle. Indem sie beispielsweise mittels einer iterativen

Überprüfung der Planungsqualität im Verlauf des Projektes für eine zielgerichtete Steuerung des Planungsprojektes genutzt werden (vgl. auch Abbildung 23), können sie generell dazu beitragen, hochwertige Planungsergebnisse zu erreichen. Exemplarisch wird hierzu im Projektszenario die operative Zielstellung des Anwendens der BIM-Methodik in der Planung verfolgt. Beim Beauftragen des Planungsteams werden daher rahmengebend für eine Umsetzung dieser Zielsetzung und angelehnt an Auftragsgeber Informationsanforderungen (AIA) mit beispielhaften „BIM-Kriterien“ generelle modellbasierte Anforderungen an die Übergaben der Planungsmodelle zwischen den Partnern im Projektverlauf definiert. Die wesentlichen Inhalte sind:

- ein Festlegen des digitalen Austauschformates – Abbilden des Planungsgegenstands im herstellerneutralen Modellstandard IFC
- sowie (darauf basierend) weitere formale Vorgaben – von Akteur A sind zum bestimmten Zeitpunkt festgelegte Modellinhalte an Akteur B zu liefern (vgl. auch Qualitygate in Kap. 2.1.4).

Im Szenario orientieren sich die diesbezüglich festgelegten Datenübergabepunkte, beispielsweise zur rechtlichen Fundierung der Verträge, an den normierten (Planungs-) Leistungsphasen der HOAI. Dabei werden Vorgaben an Modellinhalte, die als BIM-Zusatzleistungen für ein IFC-basiertes Ergebnis der Entwurfsplanung¹⁶ festgeschrieben werden auszugsweise definiert (vgl. Tabelle 7). Wie sich anhand dieser Vorgaben eine Überprüfung der zu den festgelegten Zeitpunkten vertraglich zugesicherten Qualitäten der Abbildung des Planungsgegenstandes in den gelieferten Modelldaten methodisch realisieren lässt, wird darauf aufbauend in einem Vorgang veranschaulicht, der auch dem Anwendungsfall des Lösungsansatzes in einer planerischen Entwurfsentscheidung (Kap. 4.2) als Blaupause vorgeschaltet werden kann: **Prüfen der Vollständigkeit von Modellinhalten der Energieplanung vorbereiten** (siehe Kap. 4.4.1). Damit inhaltliche Anforderungen, beispielsweise die Frage

¹⁶ HOAI 2013 - Grundleistung Leistungsphase 3 gemäß Anlage 10:

„b) Bereitstellen der Arbeitsergebnisse als Grundlage für die anderen an der Planung fachlich Beteiligten sowie Koordination und Integration von deren Leistungen“
(http://www.hoi.de/online/HOAI_2013/HOAI_2013.php#Anlage10)

betreffend, ob benötigte Materialinformationen im Modell enthalten sind, effektiv überprüft werden können, sind ergänzend *methodische* Anforderungen dargelegt, die sich auf Reihenfolge und Abstimmung der Inhalte beziehen. Zusammengestellt beschreiben die Anforderungen sowohl Gliederung als auch Gesichtspunkte eines zu erstellenden abgestuften Arbeitshilfsmittels „Prüfplan“ (Lastenheft, vgl. Kap. 4.5). In Kapitel 6 wird dieser Prüfplan bezüglich konkreter Strukturen und Entitäten des Modellformates dann weiterführend spezifiziert (Pflichtenheft). Damit wird die inhaltliche Grundlage geschaffen, auf der die Umsetzung (logisches Operationalisieren) der Anforderungen in Prüf- und Analyselogik für das IFC-basierte Planungsmodell ansetzt.

Auch ein weiterer Vorgang unterliegt – ebenfalls generalisiert als Prüfplanmuster – dem Aufbereiten der konkreten Entscheidungsgrundlagen zum Anwendungsfall: **Analysen zur Zieleinhaltung der Energieplanung vorbereiten** (vgl. Kap. 4.4.2). Hier werden für exemplarische Fragestellungen an den Planungsgegenstand entsprechende logische Schritte als Grundlage des Operationalisierens bezüglich des Modellformates aufgestellt. Diese adressieren dabei ein methodisches Vorgehen, wie sich die Zielvorgaben (Soll-Werte; vgl. Tabelle 6) mit den tatsächlich im Planungsmodell berechneten Werten vergleichen lassen.

Vorgaben zum Building Information Modeling – BIM-Kriterien und darauf aufbauender Auszug von (Basis-)Werten für die Energieplanung

Damit auf Basis des Ergebnisses der für das Szenario gewählten Entwurfsplanungsphase noch vor der Genehmigungsplanung von Seiten der Energieplanung ein vorläufiger Energiepass zum Entwurf erstellt werden kann, müssen alle energierelevanten Basisinformationen im Planungsmodell bereits zur Entwurfsabgabe vorhanden sein. Entsprechende Vorgaben sowie unterlegte generelle BIM-Kriterien werden dazu am Beispiel des Hüllbauteils einer Außenwand ausformuliert. Die zum Entwurfsmodell festgelegten BIM-Kriterien schreiben für das Abbildungskonzept des Bauteils vor, das spezialisierte Schichtbauteil (*IfcWallElementedCase*) zu verwenden. An einer dadurch implizierten IFC Strukturierung der Bauteilinformationen verdeutlichen folgende Zusammenstellungen die gemachten BIM-Vorgaben zur Abbildung des

Wandbauteils. Vorgegeben wird dabei im Entwurfsabgabemodell einerseits ein IFC Bauteiltypobjekt für allgemeine, z.B. alle Außenwände betreffende, Informationen (vgl. Tabelle 8) sowie andererseits ein Bauteilobjekt für konkrete Vorkommnisse bzw. Instanzen einer Wand (vgl. Tabelle 7). Auszugsweise sind dabei für die in der Entscheidungssituation in Kapitel 4.2 variierten Bauweisen einer konkreten Außenwand entsprechende beispielhafte IFC Elemente angegeben. Zur besseren Übersicht sind die für die Verbindung der Elemente benötigten Relationsklassen ausgeblendet.

Tabelle 7 Beispiel IFC elementierte Wandinstanz

IFC Klasse	IFC Attribut	Beispielwerte	Erläuterung
Konkrete Beispielinstantz einer Aussenwand mit variierten Beispielwerten (VARIA1 , VARIA2 , FALSCH)			
IfcElementedWallCase			Klassifizierung der konkreten Wandinstanz
	Name	Aussenwand Nord	Name der konkreten Wandinstanz
	Representation	geometrisch repräsentierte Umfassungsfläche	(Übergeordnete) Hüllgeometriebeschreibung des Gesamtbauteils. Eine konkrete Geometriebeschreibung ergibt sich aus den Einzelgeometrien der Bauteilkomponenten.
	IsDecomposedBy		Verweis auf die einzelnen Bauteilkomponenten
	IsTypedBy		Verweis auf Typbeschreibungs-Informationsobjekte
	HasOpenings	Fenster, Türen etc.	Verweis auf Öffnungen des Gesamtbauteils
	IsDefinedBy	spez. Eigenschaften	Verweis auf Eigenschafts-Informationsobjekte
	HasAssociations	IfcMaterialLayerSet	Aggregierte Materialbeschreibung anstelle der detaillierter dekomponierten Bauteilmaterialien (alphanumerische Masse)
Beispielinstantzen der Bauteilkomponentenzur Konkretisierung der Aussenwand			
IfcElementAssembly			Klassifizierung eines zusammengestellten Rahmens
	Name	Rahmenelement	Name des zusammengesetzten Rahmenbauteils
	IsDecomposedBy	Pfosten Dämmschicht Riegel	Verweise zu den einzelnen Komponenten des zusammengesetzten Rahmenbauteils
	IsTypedBy		Verweis auf Typbeschreibungs-Informationsobjekte
	IsDefinedBy	spez. Eigenschaften	Verweis auf Eigenschafts-Informationsobjekte
IfcBuildingElementPart			Klassifizierung einer Schicht eines zusammengest. B
	Name	Aussenpanelschicht Innenpanelschicht Aussenputzschicht Porenbetonmauer Innenputzschicht Dämmschicht	Name der Aussenbekleidungsschicht (Varia2) Name der Innenbekleidungsschicht (Varia2) Name der konkreten Aussenbekleidungsschicht Name der Rohbauwandschicht (Varia1) Name der konkreten Innenbekleidungsschicht Name der konkreten Schicht der Dämmung
	Representation	geom. repräsentierte Kuvertur der Schicht	Ggf. Parametrische beschriebene Vorkommnisse der Einzelgeometrien in der Typbeschreibung
	IsTypedBy		Verweis auf Typbeschreibungs-Informationsobjekte
	IsDefinedBy	spez. Eigenschaften	Verweis auf Eigenschafts-Informationsobjekte
	HasCoverings	Aussenputzschicht Innenputzschicht	Verweis zu Bekleidungsschicht-Informationsobjekten Verweis zu Bekleidungsschicht-Informationsobjekten
IfcMember			Klassifizierung einer Komponente des Rahmens
	Name	Pfosten Riegel	Name für Komponenten eines Rahmenbauteils Name für Komponenten eines Rahmenbauteils
	IsTypedBy		Verweis auf Typbeschreibungs-Informationsobjekte
	Representation	geom. repräsentierte Kuvertur	Ggf. Parametrische beschriebene Vorkommnisse der Einzelgeometrien in der Typbeschreibung
	IsDefinedBy	spez. Eigenschaften	Verweis auf Eigenschafts-Informationsobjekte

Tabelle 8 Beispiel Typobjekte der IFC Wandinstanz

IFC Klasse	IFC Attribut	Beispielwerte	Erläuterung
Konkretes Informationsobjekt zum Gesamtbaueteiltyp (Aussenwände)			
IfcWallType			Generelle Typbeschreibung Wandbauteils
	Name	Standardaufbau Aussenwand	
	PredefinedType		weitere Spezifizierung der Klasse
		ELEMENTEDWALL	Auszeichnung einer Pfosten-Riegel-Bauweise
		SOLIDWALL	Auszeichnung einer Massivbauweise
	IsDecomposedBy	Aussenpanele	Verweis auf Typbeschreibungen der einzelnen Komponenten des Wandaufbaus (Varia1)
		Gedämmter Rahmen	
		Innenpanele	
		Kalkputzschicht	Verweis auf Typbeschreibungen der einzelnen Komponenten des Wandaufbaus (Varia2)
		Wärmedämmung	
		Mauerwerk Porenbeto	
		Gipsputzschicht	
	HasPropertySets	allg. Eigenschaften	Referenzierte gemeinsame Eigenschaften
	HasAssociations	IfcMaterialLayerSet	Aggregierte Materialbeschreibung anstelle der detaillierter dekomponierten Bauteilmaterialien
Weitere Informationsobjekte zu Bauteilkomponententypen			
IfcElementAssemblyType			Generelle Typbeschreibung zusammeng. Rahmen
	Name	Gedämmter Holzrahmen	
	IsDecomposedBy	Kantholz 12x12	Verweis auf Typbeschreibungen der einzelnen Komponenten des Holzrahmens (Varia1)
		Wärmedämmung	
		Kantholz 3x12	
	HasPropertySets	allg. Eigenschaften	Verweis auf gem. Eigenschafts-Informationsobjek
IfcBuildingElementPartType			Generelle Typbeschreibung Bauteilschicht
	Name	Gipsfaserpanel Innen	
		Eternitpanel Aussen	
	PredefinedType	PRECASTPANEL	weitere Spezifizierung als vorgefertigtes Panel
		Gipsputzschicht	
	Name	Mauerwerk Porenbeton	
		Kalkputzschicht	
	PredefinedType	USERDEFINED	eigene Spezifizierung der Klasse
	Name	Wärmedämmung	
	PredefinedType	INSULATION	weitere Spezifizierung der Klasse als DÄMMUNG
	RepresentationMa	Einzelgeometrie	Geometriebeschreibung Einzelteil
	HasPropertySets	allg. Eigenschaften	Verweis auf gem. Eigenschafts-Informationsobjek
	HasAssociations	MATERIAL Gipsfaser	Verweis auf Material-Informationsobjekt
		MATERIAL Eternit	Verweis auf Material-Informationsobjekt
		MATERIAL Porenbeton	Verweis auf Material-Informationsobjekt
		MATERIAL Gips	Verweis auf Material-Informationsobjekt
		MATERIAL Kalk	Verweis auf Material-Informationsobjekt
IfcMemberType			Generelle Typbeschreibung e.g. Rahmenkomponer
	Name	Kantholz	
	RepresentationMa	Einzelgeometrie	Geometriebeschreibung Einzelteil, e.g. Holzprofil
	HasPropertySets	allg. Eigenschaften	Verweis auf gem. Eigenschafts-Informationsobjek
	HasAssociations	MATERIAL Fichtenholz	Verweis auf Material-Informationsobjekt

An diesen beispielhaft vorgegebenen Informationsstrukturen für eine Außenwand können sowohl die energierelevanten Basiswerte als auch die designierten Ergebniswerte einer energetischen Betrachtung des Bauteils weiter bestimmt werden. Wichtig hierfür sind die Flächenmaße der zusammengesetzten Bauteilkomponenten, die Schichtdicken und die zugeordneten Materialien.

Geometrische Beschreibung des Bauteils und zugeordnete Materialinformationen

Während beim Erarbeiten des Entwurfes, beispielsweise eine überschlägig angesetzte Information zum Schichtaufbau sowie designierten Materialien mittels einem aggregierten Informationsobjekt *IfcMaterialLayerSet* an der Typbeschreibung¹⁷ des Gesamtbauteils zur Plausibilisierung der energetischen Qualität ausreichen kann, erfordert der entsprechende Nachweis eine genauere Beschreibung. Insbesondere sind bei inhomogenen Bauteilaufbauten, wie im Falle einer Rahmenbauweise, im Gegensatz zu homogenen Massivbauschichten, die Bestimmung der unterschiedlich aufgebauten Teilflächen bei der genaueren Berechnung des Wärmeübergangs sowie der Wärmebrücken relevant. Exemplarisch geben die festgelegten Informationsstrukturierungen daher in der Variante Holzrahmenbauweise ein detailliertes IFC Abbildungskonzept für die Pfosten und Riegel vor. Um trotz dieser Detailtiefe die Geometriebeschreibung möglichst schlank zu halten, ermöglicht IFC hierzu eine parametrische Repräsentation zusammen mit einem hinterlegbaren Transformationsmechanismus. Dabei wird in der Typbeschreibung der Wand eine (vielfach referenzierbare) Geometrieinstanz mit Profilangaben verknüpft, z.B. der Querschnitt eines Kantholzes. In einem konkreten Vorkommnis dieses Wandtyps sind entsprechende Transformationsparameter dann mit der Elementwand verknüpft, die zusammen mit den Profilangaben erst beim Aufbereiten der Geometrie (z.B. für die Bildschirmdarstellung) zur entsprechenden geometrischen Ausprägung transformiert werden. Eine weitere exemplarische Festlegung für das Entwurfsmodell legt darüber hinaus bezüglich der Materialinformationen fest,

¹⁷ BuildingSMART Definition – „**Bauteiltyp**: Bezeichnung zur Zusammenfassung gleichartiger Bauteile zu einem Bauteiltyp (auch Konstruktionstyp genannt). Alternativ zum Namen des "Typobjekts", insbesondere wenn die Software keine Typen unterstützt“.

dass diese nur über eine jeweilige Typbeschreibung einer Komponente verknüpft werden soll. Da gemäß dem IFC Schema generell auch (redundant) an jede Komponenteninstanz ein Materialobjekt angehängt werden darf, wirkt diese vorgegebene Einschränkung einer redundanten Auszeichnung der gewählten Materialien (vgl. *HasAssociations* Attribut in Tabelle 8) entgegen. Zudem fördert die geforderte einheitliche Anbindung des Materialobjektes nur über das Typobjekt die Konsistenzwahrung bei nachgelagertem Verwerten der Materialinformationen des Entwurfsmodells (z.B. Mengenermittlungen, energetische Optimierungsvarianten).

Eigenschaften von Bauteil/Bauteilkomponente mit den energierelevanten (Ergebnis-)Werten

Zur Bestimmung der energetischen Qualität eines Bauteils, bzw. dessen Komponenten werden benötigte (Detail-) Informationen gemäß des IFC-Eigenschaftskonzeptes an die entsprechend ausgezeichneten Objekte für Material und Bauteil(komponenten) angehängt (vgl. *HasPropertySets* bzw. *IsDefinedBy* Attribute). Sowohl (kumulierte) energiebezogene Ergebniswerte, wie beispielsweise der Wärmedurchgangskoeffizient (U-Wert) eines spezifischen Gesamtbauteils, als auch (generelle) energetische Kennwerte, wie beispielsweise die Wärmekapazität für ein spezifisches Material, werden über entsprechend angehängte Eigenschaftswerte beschrieben. Zur konsistenten Abbildung der Eigenschaftsobjekte in der Modellinstanz sind exemplarische Vorgaben festgelegt, ähnlich der Festlegung bezüglich des Anhängens von Materialinformationen an die jeweiligen Typbeschreibungen. Das IFC-Schema ermöglicht auch hier generell ein Verknüpfen der Eigenschaftsobjekte, sowohl auf Ebene der Instanzen von Bauteil(komponenten) (vgl. *IsDefinedBy* Attribut Tabelle 7) als auch der Typbeschreibungen (vgl. *HasPropertySets* Attribut Tabelle 8). Dies wird gemäß szenariobasierter Vorgabe vereinheitlicht, indem:

- alle spezifischen energiebezogenen (Zwischen-) Ergebniswerte (z.B. mittels der konkreten Bauteilprägung aggregierte U-Werte) den Instanzen angehängt werden sollen

- alle allgemeingültigen Werte (z.B. normbasiert vorgegebene Bemessungswerte, wie die Wärmeübergangswiderstände R_{si} und R_{se} luftberührender Bauteilschichten nach DIN EN ISO 6946- Tabelle 1) über die Typbeschreibungen erfolgen sollen

Folgende Listung zeigt im IFC Schema definierte (energiebezogene) Eigenschaftssätze (*IfcPropertySet*), sowie auszugsweise Eigenschaften daraus. Exemplarisch ergänzte bzw. zur Umsetzung der Vorgaben von speziellen Typeneigenschaften neu anzulegende (nicht im IFC Schema enthaltene) Eigenschaftssätze bzw. Eigenschaften sind dabei entsprechend gekennzeichnet.

Materialobjekt

Pset_MaterialCommon – SpecificHeatCapacity usw.

Pset_MaterialEnergy – SpecificHeatTemperatureDerivative usw.

Wandobjekt (Gesamtbauteil)

Pset_WallCommon – ThermalTransmittance, (neu) TotalThermalTransferResistance (Wärmedurchgangswiderstand R_T), (neu) TotalThermalTransferResistanceUpperLimit / TotalThermalTransferResistanceLowerLimit (anzusetzende obere und untere Grenzwerte bei der Ermittlung von R_T gemäß DIN EN ISO 6946)

Bauteilkomponente (*IfcMember*)

(neu) Pset_MemberCommon – (neu) ThermalTransmittance, (neu) ThermalTransferResistanceDesignValue (Bemessungswert Wärmedurchlasswiderstand R_i der Schichten)

Bauteilkomponente (*IfcBuildingElementPart*)

(neu) Pset_ElementPartCommon – (neu) ThermalTransmittance, (neu) ThermalTransferResistanceDesignValue (Bemessungswert Wärmedurchlasswiderstand R_i der Schichten)

Bauteilkomponententyp (IfcBuildingElementPartType)

(neu) Pset_ElementPartTypeCommon – (neu) ThermalTransferResistanceDesignValueRsi / ThermalTransferResistanceDesignValueRse (Bemessungswert Wärmeübergangswiderstände R_{si} und R_{se})

[Ausschluss] Bekleidungsschicht / Bedeckung (engl. *Covering*)

Pset_CoveringCommon – ThermalTransmittance

4.4 Wiederkehrende (vorbereitende) Arbeitsschritte zur formal-strukturellen Qualitätssicherung der Modellinformationen

Der in Kapitel 4.2 dargelegte Anwendungsfall des Lösungsansatzes, in dem in einer konkreten planerischen Entscheidungssituation die planenden Architekt*innen mit Entscheidungsgrundlagen unterstützt werden, erfordert erhebliche Aufwände bei der Vorbereitung und Vorarbeit zur eigentlichen Erstellung der situationsbedingten Hilfestellung:

- Bestimmen der Modellierungsvorgaben
- Entwicklung und Aufsetzen von Prüf- / Analyseplan
- Umsetzen des Plans (Aufbereitung Entscheidungsgrundlage)

Um diesen spezifischen Aufwand für die Nutzer zu reduzieren, wird in dieser Arbeit zudem ein Konzept vorgestellt, das den Aufbau einer generalisierten Regelkomponenten-Bibliothek umfasst und das verschiedene projekt- bzw. büro-bezogene generalisierbare Methoden und Herangehensweisen szenariobasiert darlegt. Dabei wird der Frage nachgegangen, wie dieser situationsbedingt hohe Aufwand im Rahmen eines iterativen, durchgängigen Überprüfens der Planungslösung im Planungsprozess mit einem durch

Wiederverwendung gesteigerten Mehrwert des Ansatzes gesenkt insgesamt werden kann (vgl. Abbildung 23).

Bezugnehmend zum szenariobasierten Planungsprozess sind hierzu für das kooperativ arbeitende Planungsteam dedizierte Modellierungsvorgaben bezüglich des Austausches des Planungsgegenstandes vertraglich vereinbart (vgl. Kap. 4.3). Neben dem operativen Umsetzen dieser Vorgaben in der eigenen Arbeit obliegt einer mit der Gesamtkoordination einer Planungs- bzw. Ausführungsphase betrauten Rolle, beispielsweise Akteure eines mit der Entwurfsplanung beauftragtes Architekturbüros, auch die Kontrolle und Freigabe der Arbeitsergebnisse der Planungspartner zu den festgelegten Zeitpunkten. Dies impliziert beim modellbasierten Arbeiten (a priori) das Einrichten entsprechender Modellprüfungspläne, da sich durch Inaugenscheinnahme allein ein Feststellen inhaltlicher Vollständigkeit am Planungsgegenstand kaum noch bewerkstelligen lässt (vgl. Kap. 2.3 sowie Kap. 3.2.3). Des Weiteren werden für den Planungsprozess inhaltliche (energetische) Zielanforderungen an das Entwurfsergebnis bestimmt (vgl. vorheriges Unterkapitel). Szenariobasiert wird neben einem Prüfplan zum Feststellen der Vollständigkeit des Ergebnisses der Entwurfsphase daher ein Analyseplan benötigt, mit dem die Einhaltung dieser energetischen Anforderungen (idealerweise schon während der Entwurfstätigkeit) festgestellt werden kann.

Bezüglich einer koordinierenden Rolle, sowie einer Rolle im Bereich der Modellierung, werden zur Entwicklung von Prüf- und Analyseplänen im Folgenden beispielhafte Workflows im Kontext der Vertragsgestaltung zur Entwurfsphase beschrieben. Dabei werden zu den exemplifizierten BIM-Kriterien sowie darüberhinausgehende Vorgaben bezüglich der Abbildungsgrundlage energetischer Zielanforderungen an den Planungsgegenstand (vgl. Tabelle 7, Tabelle 8) die benötigten konkreten (logischen) Schritte der Überprüfung bzw. Analyse als methodische Grundlage des Teils vom Lastenheft (vgl. Kap. 4.5) für entsprechende Modellprüfpläne dargestellt.

Hierbei gilt es zunächst, die benötigten Vorarbeiten bzw. Prüflogiken in methodisch sinnvolle, wiederverwertbare Teile zu strukturieren und damit eine effiziente (wiederholte) Anwendung in verschiedenen Planungssituationen/ -

ständen zu ermöglichen. Anstelle immer wieder einen neuen situationsspezifischen Prüfplan zu erstellen, wie es beispielsweise bei einer Vollständigkeitskontrolle bestimmter Informationen im Planungsgegenstand zu einem Datenübergabepunkt im Prozess gängig ist, wird nun für die bestimmte Prüfungsart generalisiert ein „Musterprüfplan“ vorgehalten. Dieser wird in der Situation dann nur noch mit den benötigten Parametern ausgeprägt, die zuvor mit den entsprechenden Modellierungsvorgaben abgeglichen worden sind.

In zwei exemplifizierenden Musterprüfplänen (vgl. Kap. 4.4.1 sowie Kap. 4.4.2) werden im Folgenden Vorgänge beschrieben, denen zuvor in Anlehnung an die in der neuesten Normung zum Thema der BIM Methode (vgl. VDI 2552, Kap. 2.1.4) entsprechende Modellierungsvorgaben für den szenariobasierten Planungsprozess im Kapitel 4.3 vorangestellt wurden. Vor dem Zusammenfassen aller Anforderungen bezüglich des abgestuften Prüf- und Analyseplans dieser Arbeit in Kap. 4.5 werden dabei noch diejenigen methodischen Arbeitsschritte aus den Musterprüfplänen exemplifiziert, die sich besonders oft – auch schon innerhalb eines einzelnen Prüfungsvorganges – wiederholen können. Beispielsweise ist dies ein Herauslesen bestimmter Merkmale zu einem Bauteil auf architektonischer Seite, bzw. die Bestimmung eines U-Wertes auf Basis von Eingangparameter auf Seite der Fachplanung. Hierzu werden entsprechende Bausteine zu den Musterprüfplänen im Rahmen von projektübergreifenden Bürostandards als methodische Lösungsansätze zum effizienten modellbasierten Arbeiten diskutiert (vgl. Kap. 4.4.3). Mit diesen zu den Mustermodellprüfplänen beschriebenen Vorgehensweisen sollen querschnittende Belange der integralen Planung eines energieeffizienten, nachhaltigen Bauwerks dabei als Ausgangspunkt entsprechender Lösungssammlungen (vgl. Regelbasis in Kap. 6.1) adressiert werden. Die in Kapitel 6 beschriebene Vorbereitung einer maschinenverwertbaren regelbasierten Umsetzung durch Entwickeln eines Pflichtenhefts zu diesen exemplifizierten Mustermodellprüfplänen ermöglicht den Ansatz eines Baukastensystems, das sukzessive fortgeschrieben bzw. wiederverwendet werden kann.

4.4.1 Prüfen der Vollständigkeit von Modellinhalten der Energieplanung vorbereiten

Ausgehend vom übergeordneten Anlass auf Ebene der Projektkoordination stellt Abbildung 29 einen Workflow zur Vorbereitung eines Modellprüfplanes dar. Als Fragestellungen an die Modellierung bilden diese Schritte die Grundlage des in Kapitel 4.5 zusammengeführten Lastenheftes für einen Prüfplan.

Damit sollen die festgelegten Modellierungsqualitäten zu den beabsichtigten Zeitpunkten im Planungsverlauf in einer für alle Planungsbeteiligten vorab bekannten und nachvollziehbaren Weise überprüft werden können. Für die einzelnen Modellierungsvorschriften kann ausgehend von den jeweiligen konkreten Fragestellungen an das Modell dann je Übergabepunkt eine Grundlage für ein vereinheitlichtes Vorgehen beim Modellprüfen entwickelt werden. Die Projektkoordination veranlasst hierzu im Szenario das Aufstellen eines Lastenheftes für den Prüfplan durch eine Rolle mit modellierungstechnischem Fachwissen. Als Ausgangspunkt zum Formulieren der einzelnen Schritte der Modellprüfungen werden dabei die entsprechenden Modellierungsvorgaben übergeben. In drei wesentlichen Bearbeitungsschritten werden darauf hin Fragestellungen an die darin festgeschriebenen Inhalte in das Lastenheft zu den Übergabepunkten zusammengestellt (vgl. untere Reihe in Abbildung 29).

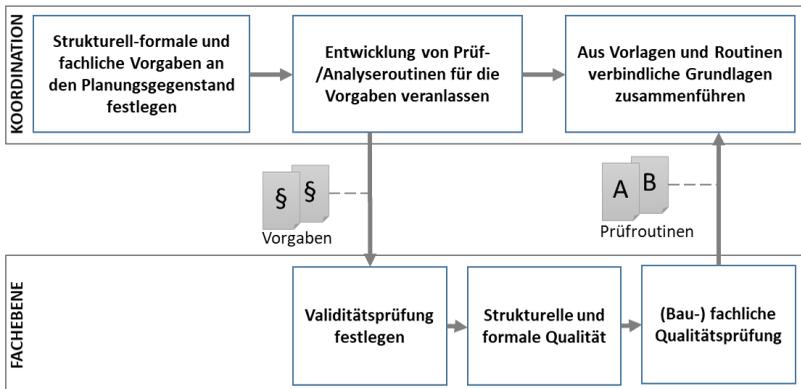


Abbildung 29 Vorbereiten Modellprüfungsplan

Die eine Reihenfolge vorgebenden Bearbeitungsschritte sind dabei: Validitätsprüfung festlegen, Vollständigkeit der Grundstruktur und Dokumentationsvorgabe sowie darauf aufbauend die fachlichen Ergebniswerte prüfen. Gegliedert nach diesen werden im Folgenden beispielhafte Modellierungsvorschriften für die Inhalte der Energieplanung eines Wandbauteils (vgl. Kap. 4.3) im Kontext einer dokumentenbasierten Übergabe des Ergebnisses der Entwurfsphase auszugsweise dargestellt:

- Rahmenbedingungen des Modellformats prüfen
 - Ist die übergebene Modelldatei korrekt bezeichnet, sowie ggf. durch weitere Metadaten wie Autorennamen ausgezeichnet und kann sie ausgelesen werden (z.B. korrekte Dateieindung, Basis-Codierung bzw. Lese-Berechtigungen gesetzt)?
 - Liegt das Modellinstanzdokument in der geforderten Version des vorgegebenen Modellformats vor (d.h. ist dies im Dokument korrekt angegeben z.B. durch Namensraumangabe im Kopfteil des Modells)?
 - Ist das XML-basierte Modellinstanzdokument valide, wohlgeformt und vollständig gegenüber den Schemavorschriften des Modellstandards (z.B. referenzielle Integrität der Modellinstanz, d.h. alle referenzierten Elemente sind im Modell vorhanden)?
- Kohärenz der Modellinhalte zu den rahmengebenden (Struktur-) Konzepten des Modellschemas
 - Sind die geforderten Projektmetadaten vorhanden (z.B. Autorennamen, Version, Planungsphase usw.)?
 - Ist die räumliche Gebäudemodelltopologie mit korrekt klassierten Objekten (d.h.: Projekt -> Grundstück -> Bauwerk -> Stockwerke -> Raum) modelliert?
 - Sind dabei alle Dokumentationsvorschriften (z.B. Bezeichner, Identifikatoren usw.) eingehalten?

- Sind alle Modellobjekte zur Abbildung der physischen Bauteile Instanzen der festgelegten Bauteilklassen vorhanden (z.B. keine Stellvertreter Objekte verwendet) und sind sie zudem in der richtigen Stelle der räumlichen Struktur verortet (z.B. Aufzug dem Gebäude zugeordnet bzw. Wand einem Stockwerk zugeordnet) und dabei richtig gekennzeichnet (z.B. korrekter Stockwerksnamenszusatz in den Bezeichnern)?
- Wurden zur einheitlichen Verknüpfung der festgelegten, gemeinsamen Eigenschaften (z.B. eine Angabe der Kostengruppe nach DIN 276 für eine korrekte, nachgelagerte Ableitung normbasierter Mengen) für alle Modellobjekte – Wände, Decken, Öffnungselemente, Bauteilkomponenten usw. – entsprechende Objekttypen angelegt?
- Auszug Kohärenz der Bauteilbeschreibung zu (fachlichen) Vorgaben
 - Sind in dem referenzierten Bauteiltypobjekt der Wandinstanz die vorgegebenen Spezialisierungsangaben als elementierte bzw. massive Bauteilart ausgezeichnet?
 - Wurden im Wand-Bauteiltyp die geforderten subordinierten Typen der Bauteilkomponenten (z.B. Pfosten-Riegel-Konstruktion bzw. Schichtaufbauten) konstruktionsgerecht modelliert und entsprechend integer referenziert (z.B. Schicht „gedämmter Holzrahmen“ -> Komponente „Pfosten“ + Komponente „Riegel“ + Komponente „Dämmung“)?
 - Sind im Gefüge der Typenobjekte (Bauteil-, Komponenten bzw. Schichttyp) die gemeinsamen Eigenschaften kohärent ausgezeichnet (z.B. DIN276 Kostenobergruppe 330 für das Gesamtbau teil passend zur Angabe der Kostengruppen 331 bzw. 336 für die Einzelteile des Bauteils)?

- Liegen alle Materialinformationen vollständig vor und wurde die Vorgabe eingehalten, dass diese über die Typobjekte auszuzeichnen sind?
- Sind gemäß dem durch die Bauteil-, Schichten-, bzw. Komponententypen abgebildeten Aufbau des Gesamtbauteils alle konkreten Instanzen der exemplifizierten „Außenwand in der Nordfassade“ vorhanden?
- Sind die für die energetischen Berechnungen benötigten Basiswerte in allen Materialinformationen angegeben (z.B. Bemessungswert der Wärmeleitfähigkeit usw.)?
- Sind alle für die energetischen Berechnungen relevanten Abmaße (als Geometriebeschreibung oder daraus abgeleitete Werte, wie beispielsweise Schichtdicke) an den Instanzen der Komponenten bzw. Schichten der Wand vorhanden?
- Existieren alle Basis- (z.B. Wärmedurchlasswiderstand der innersten bzw. äußersten Wandschicht usw.) sowie Ergebniswerte (U-Werte der einzelnen Schichten bzw. des Gesamtbauteils usw.) für die Berechnungen der Energieplanung?
- Sind für das elementierte Wandbauteil die bauphysikalischen Kenngrößen vollständig vorhanden, die als Grundlage der weiteren Berechnung aller Hüllbauteile (z.B. Fensterbauteile usw.) im Rahmen der Bewertung der energetischen Qualität der wärmeübertragenden Umfassungsfläche des Bauwerkes dienen?

4.4.2 Analyse zur Zieleinhaltung der Energieplanung vorbereiten

Aufbauend auf den in Kapitel 4.4.1 beschriebenen Fragestellungen zur Vollständigkeit der Modellinhalte und -werte wird im Folgenden die Vorbereitung eines fachlichen Zielwertevergleiches im Rahmen einer Modellanalyse in einem Workflow exemplarisch veranschaulicht. Im Beispielszenario veranlasst

die koordinierende Ebene, hierzu eine Grundlage für einen Analyseplan zu erstellen. Damit sollen die Schritte formuliert werden, um festzustellen, ob eine energetische Zielvorgabe am Ergebnis der Entwurfsphase eingehalten wurde (siehe Abbildung 30). Dazu werden für die Modellierungsseite aus der implizierten Fragestellung der Zielvorgabe an den Planungsgegenstand (*Werden die vorgegebenen Sollwerte eingehalten?*) Grundlagen zu einem entsprechenden Modellanalyseabschnitt des in Kapitel 4.5 zusammengestellten Lastenheftes für den Modellprüfplan zur Entwurfsphase entwickelt.

Für alle Planungsbeteiligten wird damit schon beim Schließen der Verträge nachvollziehbar gemacht, wie das Ergebnis der Entwurfsphase (gemäß dem energetischen Projektziel) bewertet werden wird. Des Weiteren können die einheitlichen Schritte der Modellanalyse planungsbegleitend als Grundlage von Zwischenprüfungen der Zielerreichung an Planungsalternativen (vgl. Kap. 4.2) verwendet werden.

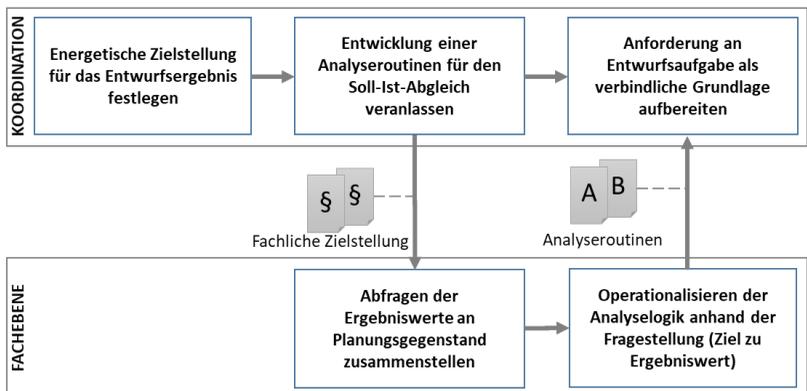


Abbildung 30 Vorbereiten Modellanalyseplan

In Tabelle 6 wird ein konkreter Zielwert des zu erreichenden bauphysikalischen Kennwertes für das exemplarische Wandbauteil als Basis der Modellanalyse angegeben (vgl. Zeile 3). Diesem unterliegt darüber hinaus eine fachliche Berechnungsvorschrift (DIN V 18599), die durch die Vorgaben zum

energetischen Nachweis nach ENEC referenziert wird. Die Berechnungsvorschrift gibt dabei vor, welche Basiswerte aus den konkreten Informationen eines Gebäudes zur Berechnung des avisierten Zielwertes verwendet werden müssen. Diese bei der Berechnung vorgegebenen Basis- und Ergebniswerte bilden für das Szenario den Umfang der Modellinhalte, die in folgenden zwei Arbeitsschritten von einer Rolle mit Expertise in Modellierung und Energieplanung zu entsprechenden Fragestellungen der Modellanalyse aufbereitet werden.

- Überprüfen, ob alle benötigten Werte für die Analyse vorhanden sind
 - Sind alle durch die Vorschrift angegebenen Basiswerte der Berechnung an den Instanzen der Komponenten bzw. Schichten des Gesamtbauteils vorhanden und haben sie eine gültige Wertausprägung bzw. Einheit?
 - Wärmedurchlasswiderstand der innersten bzw. äußersten Wandschicht
 - Materialinformationen
 - Angaben zu den benötigten Abmaßen (Flächen, Dicken usw.)
 - Sind Wärmedurchgangskoeffizienten (U-Werte) der einzelnen Schichten als Zwischenergebnisse zum kulminierten Gesamt-U-Wert des Bauteils vorhanden und haben die entsprechenden Eigenschaftsfelder eine gültige Wertausprägung bzw. Einheit?
 - Ist das Gesamtergebnis der U-Wert Berechnung, der Wärmedurchgangskoeffizient (U-Wert), am Gesamtbauteil vorhanden und hat das entsprechende Eigenschaftsfeld eine gültige Wertausprägung bzw. Einheit?
- Analyse: SOLL-IST Abgleich des Wärmedurchgangskoeffizienten (U-Wert)

- Ist die Wertausprägung des Wärmedurchgangskoeffizienten (U-Werte) vom Gesamtbauteil kleiner oder gleich dem vorgegebenen Zielwert gemäß Tabelle 6?

4.4.3 Potentielle Backoffice Routinen: Aspekte des Modellhandling verstetigt als Bürostandard

In Kapitel 4.4.1 sowie 4.4.2 wurden die Vorbereitung projektspezifischer Prüfpläne zur Kontrolle von Anforderung an die Modellierung exemplifiziert. In diesem Unterkapitel wird die Bildung von – möglicherweise bereits im Vorfeld des Beispielprojektes bei den Planungspartnern vorhandenen – unterstützenden Grundlagen szenariobasiert veranschaulicht. Damit soll in einer weiteren Stufe eine mögliche Reduktion der für den in Kapitel 4.2 beschriebenen Anwendungsfall benötigten Aufwände zum Vorbereiten der Entscheidungsunterstützung durch Mehrfachverwendung der situationsbedingt erstellten Vorlagen (z.B. mit dem Einfließenlassen in einen entsprechenden Bürostandard) weiter ausgeführt werden.

So führt ein modellbasiertes Arbeiten in verschiedenen Projekten zu wiederkehrenden Behandlungen von Modellierungsanforderungen. Mittels interner Arbeitsnormen können diese organisiert und ggf. mit weiteren Vorgaben, wie normbasierten Anforderungen der Abrechnung/Ausschreibung bzw. Mengenermittlung oder externen Dokumentenrichtlinien, harmonisiert werden. Für die operative Ebene werden sie idealerweise als Modellierungsstandard in den Planungsbüros gepflegt und fortgeschrieben. Ein hierzu begleitendes Organisieren von Prüflisten zur Kontrolle dieser Vorgaben am sie umsetzenden Planungsmodell ermöglicht darauf aufbauende bedarfsgerechte Zusammenstellungen von konkreten projektspezifischen Prüfplänen als Kontrollinstrument (z.B. Checklisten für nachweisliche Datenübergabe). Neben dem Fortschreiben der Modellierungsstandards aus den in Projekten gewonnenen Erfahrungen, können auch die in der Projektarbeit (ad hoc) angewandten Prüf- bzw. Kontrollroutinen nach Projektende als Checklisten in thematisch gegliederte Kataloge für die Wiederverwendung eingeordnet werden. Zum Beispiel könnten diese gemäß dem Verlauf nach Planungsarten wie Genehmigungs- und Werkplanung, sowie nach der Art der Projekte, wie öffentlicher Auftraggeber,

Wohn- bzw. Bürogebäudeplanung, als Musterprüfpläne strukturiert werden. Innerhalb dieser Kataloge können die Prüf- bzw. Kontrollfragestellungen an den Planungsgegenstand dann bezüglich wiederkehrender modellierungstechnischer Sachverhalte weiter harmonisiert werden. So könnten Teilstrategien zum Überprüfen von in Objektattributen, Eigenschaftsobjekten bzw. (kumulierten) Mengenobjekten abgebildeten Informationen in grundlegenden Musterprüflisten zusammengefasst werden. Darüber hinaus könnten entsprechende Muster für fachliche (wiederkehrende) Analysen, wie beispielsweise das Zusammenstellen normbasierter Mengen oder die Ableitung einer energetischen Kenngröße gemäß einer normierten Berechnungsvorschrift angelegt werden. Zur Verdeutlichung des Mehrwertes im Rahmen dieser Arbeit werden „Kleinstmusterprüfungen“ als binäre Bausteine zur Erstellung bedarfsgerechter Musterprüfpläne exemplifiziert. Analog zu einer (losen) Sammlung an Frageitern für das Zusammenstellen eigener Checklisten, bilden „binäre“ Anforderungen eine erste Kategorie des Lastenheftes (vgl. Kap. 4.5). Daraus im Pflichtenheft abgeleitete (modellbasierte) „binäre“ Prüfmuster (z.B. besitzt das Bauteilobjekt eine Eigenschaft?) können in ein maschinenverwertbares Baukastensystem zum weiteren Zusammenstellen der Regelinhalte überführt werden. Wichtig für bedarfsgerechte Entwicklungen dieses Regelmuster-Baukastens als Bürostandard ist der Abgleich bzw. eine iterative Rückkopplung mit den tatsächlichen Prozessen (Planung und Backoffice) innerhalb des Planungsbüros. Erst wenn dort (im Tagesgeschäft) die oftmals wiederkehrend verwendeten „Nuklei“ in angewandter Prüf- bzw. Analyselogik sondiert und zu Mustern aufbereitet sind, ist eine zielgerichtete Baukastenanwendung im Büroalltag möglich. Die Beispielauswahl wurde daher an möglichst allgemein wiederverwendbaren modellierungstechnischen Fragestellungen ausgerichtet.

4.5 Zusammenfassung modellbezogener und methodischer Anforderungen im Lastenheft

In den vorangegangenen Unterkapiteln sind vor dem Hintergrund eines Anlasses, wie der seitens Planender zu erbringenden Grundleistung einer Kollisionprüfung ihrer Planungsdaten, ein Vorgehen sowie Inhalte zur Erstellung von Prüf- bzw. Analyseplänen auf verschiedenen Ebenen eines Planungsprojekt-szenarios beschrieben worden. Dazu wurden jeweils methodische Bereiche ausgewiesen, in denen die Bearbeitungsschritte der entsprechenden Pläne beispielhaft dargestellt sind. Neben dem Bereich einer konkreten Entscheidungsunterstützung ist dies auf der einen Seite ein das szenariobasierte Planungsprojekt übergreifender Bereich, der jeweils bei einzelnen Planungspartnern (Architektur, Fachplanung) verortet ist (Bürostandard/Backoffice, wiederkehrendes Modellhandling). Auf der anderen steht der Bereich des Planungsprojektes selbst, mit den Planungsprozess (direkt) dienenden Prozessen.

Folgende Tabellen (Tabelle 9, Tabelle 10 und Tabelle 11) fassen die bei den prozessbezogenen Beschreibungen exemplarisch ausgewählten Inhalte der Modellprüfung bzw. -analyse als eine Art Lastenheft zusammen. Darin werden die bei der Erstellung eines Prüfplanes implizierten Fragenstellungen an den als virtuelles IFC-basiertes Gebäudemodell abgebildeten Planungsgegenstand aufeinander aufbauend zusammengestellt. Im ersten Abschnitt 0.0.0 der Tabelle sind zunächst allgemeine (modellbezogene) Anforderungen zur Modellprüfung aufgeführt. Darauf bauen in Abschnitt 1.0.0 dann Anforderungen auf, die im Szenariokontext in der Entwurfsphase stehen und sich über die Modellprüfung hinaus mit der energetischen Zielstellung des Entwurfsergebnisses befassen. Für die Umsetzung einer Anpassung des Lösungsansatzes der vorliegenden Arbeit auf einen spezifischen Anwendungskontext in Kapitel 6 werden in Abschnitt 2.0.0 die konkreten Anforderungen mit Bezug zu den entsprechenden generellen bzw. phasenbezogenen Anforderungen spezifiziert. Der unterlegte Aufbau dieser Anforderungen dient neben der Staffelung von generellen zu den konkreten Inhalten einer spezifischen Entscheidungssituation auch der Abgrenzung verschieden benötigter Kenntnisstände auf Seiten der

Anwender beim Umsetzen in maschinenverwertbare Prüf- und Analyselogik (vgl. Tabelle 5). Eine damit ermöglichte „Baukasten-artige“ Umsetzungsstrategie setzt für das Umsetzen der konkreten Anforderungen des Abschnitts 2.0.0 dann die geringsten Kenntnisse voraus.

Tabelle 9 Lastenheft Unternehmensspezifische Modellprüfung und -analyse

IDX	GEGENSTAND	ANFORDERUNG
0.0.0	(Fortlaufend) projektübergreifende Modellprüfung und -analyse	
0.1.0	Modelldatei	Metadaten der Modelldatei
0.2.0	Modell Daten-schema	Validität zum unterlegten Modellschema
0.3.0	IFC Typobjekt Konzept	Das zugeordnete Typobjekt eines IFC Objektes identifizieren
0.4.0	IFC Eigenschaftskonzept	Alle einem IFC Objekt zugeordneten Eigenschaften identifizieren

Tabelle 10 Lastenheft Projektspezifische Modellprüfung und -analyse

IDX	GEGENSTAND	ANFORDERUNG
1.0.0	Lastenheft projektweite Modellprüfung und -analyse	
1.1.0	Rahmenbedingungen der Modelldatei	
1.1.1	Modelldatei Binärformatierung	Stimmt der Dateityp mit dem designierten binären IFC XML Format überein (Dateiendung: .ifcxml)?
1.1.2	Modelldatei Namenskonvention	Besteht der Name der Modelldatei aus folgenden (mit einem Unterstrich getrennten) Teilen? <ul style="list-style-type: none"> - Projektname - Planungsphase - Planungsstand Bsp.: Musterprojekt_Entwurfsphase_Entwurfsvariante_Holzbau.ifcxml
1.1.3	Dateitypdeklaration, BOM (document type declaration, Byte Order Mark)	Beginnt das Dokument mit der korrekten Dateityp Deklaration <code><?xml version="1.0" encoding="UTF-8"?></code> ?

1.1.4	IFC Standard Versionsangabe	Ist die korrekte IFC XML Version ADD2 im Kopfteil des Dokuments als Hauptnamensraum (<code>xmlns</code>) angegeben?
1.1.5	IFCXML Dokument	Ist das XML-Instanzdokument wohlgeformt und valide gegenüber dem angegebenen IFC Anwendungsschema (Schemavalidierung) ?
1.2.0	(Auszug) Vorgaben Klassifizierung und Grundstruktur	
1.2.1	Projektklassifizierung: Name, Phase	Existiert ein Projektobjekt mit der korrekten Angabe des Projektnamens Musterprojekt, sowie der Angabe der aktuellen (Planungs-) Phase des Projekts?
1.2.2	Projektklassifizierung: Autor, Tool, Erstellungsdatum	Ist der Autor, das CAD-Werkzeug sowie das Erstellungs- bzw. Änderungsdatum der aktuellen Projektinstanz angegeben?
1.2.3	Projektklassifizierung: Basiseinheiten	Sind die vorgegebenen standardisierten (metrischen) Einheiten für den Projektkontext deklariert? Längeneinheit: Meter Flächeneinheit: Quadratmeter Volumeneinheit: Kubikmeter Masseinheit: Kilogramm Winkelmaßeinheit: Grad Thermodynamische Temperatureinheit: Grad Celsius Zeiteinheiten: Sekunde Energieeinheit: Joule
1.2.4	Projektklassifizierung: Koordinatensystem	Ist für die geometrische Beschreibung ein (Welt-) Koordinatensystem mit Ursprungspunkt, sowie der Angabe der (wahren) Nordrichtung angegeben?
1.2.5	Projektklassifizierung: einheitliche Sammlung der Bauteiltypen	Sind alle Typobjekte mit den gemeinsamen Eigenschaften der im Projektkontext vorkommenden Bauteile an der designierten Stelle unterhalb des Projektobjektes instanziiert?
1.2.6	Projektklassifizierung: Eigenschaften und assoziierte Quellen	Besitzt das Projekt die vorgegebenen Eigenschaften zum Design Brief, in denen bauteilbezogene Verweise zu den (im angehängten Tabellendokument) energetischen Zielwerten aufgeführt sind?

1.2.7	Projektklassifizierung: Ausgangspunkt räumliche Gliederung	Fußt die räumliche Gliederung an der designierten Stelle unterhalb des Projektobjektes?
1.2.8	Räumliche Hauptdekomposition	Ist die instanziierte räumliche Dekomposition des IFC-Modells konsistent (sowie richtig klassifiziert) aufgebaut? Vorgabe: Projekt -> Baugrund -> Bauwerk -> Stockwerk -> Raum
1.2.9	Grundordnung der Bauteile	Sind alle (physischen) Bauteil-Vorkommnisse korrekt in der räumlichen Gliederung (auf Stockwerkebene) eingeordnet und ist dies gemäß Dokumentationsvorgabe in ihren Namen richtig ausgezeichnet?
1.2.1 0	Klassifizierung der Bauteile	Entsprechen alle Bauteilobjekte den vorgegebenen Klassifizierungen (keine Stellvertreterobjekte)?
1.2.1 1	Bauteiltypen zu allen Bauteilen	Ist jedem vorkommenden Bauteil (Komponente, Schicht usw.) ein korrekter Bauteiltyp zugeordnet und darin als gemeinsame Eigenschaft mindestens die Klassierung nach DIN 276 korrekt angegeben?
1.3.0	Spezifische Modellierungsvorgaben	EXEMPLARISCHES WANDBAUTEIL
1.3.1	Bauteiltyp	Ist die Bauart im Objekt des Wand-Bauteiltyps dessen Spezialisierung (elementiert bzw. massiv) ausgezeichnet?
1.3.2	Dekomposition des Bauteiltyps	Ist der Aufbau der Wand auf Ebene der Bauteil-, Komponenten- bzw. Schichtentypen korrekt durch entsprechende Referenzierungen abgebildet?
1.3.3	Dekomposition Wandbauteilobjekt	Entspricht der subordinierte Aufbau aus Schicht- bzw. Komponentenobjekten des spezifischen Wand-Bauteilobjekts der Abbildung auf Ebene der Typobjekte?
1.3.4	Gemeinsame Eigenschaften	Sind in allen Typobjekten des dekomponierten Wand-Bauteiltyps die jeweiligen Eigenschaften (Klassierung nach DIN 276: Kostengruppe 330 bzw. deren Unterpunkte Wandbekleidung usw.) inhaltlich kohärent ausgezeichnet?

1.3.5	Assoziierte Materialinformation über Typobjekt	Sind alle Materialinformationen gemäß Vorgabe durch die entsprechenden Typobjekte angegeben (keine direkten Angaben in Bauteil-, Komponente bzw. Schichtinstanz)?
1.4.0	Auszug Energetische (Basis-) Werte	
1.4.1	Materialinformationen	Enthalten die ausgezeichneten Materialien die vorgegebenen (energierelevanten) Informationen?
1.4.2	Geometrische Abmaße	Sind alle für die energetische Berechnung benötigten Maßangaben aus dem modellierten Bauteil (bzw. dessen dekomponierten Aufbau) ablesbar?
1.4.3	Basiskenngrößen	Sind für alle Schichten bzw. Komponenten des Wandbauteils die Wärmedurchlasswiderstände R vorhanden, sowie an der innersten und äußersten (luftberührenden) Wandschicht R_i bzw. R_a bestimmt?
1.4.4	Koeffizienten	Sind für alle Schichten bzw. Komponenten die Wärmedurchgangskoeffizienten (U-Werte) sowie der für das Gesamtbauteil Kulminierte bestimmt?
1.5.0	Analysen zu energetischer Zielstellung	
1.5.1	SOLL-IST Abgleich	Entspricht (oder unterschreitet) der Wärmedurchgangskoeffizient des Wandbauteils den durch die energetische Zielstellung Vorgegebenen?

Tabelle 11 Lastenheft der situationsspezifischen Modellprüfung und -analyse

IDX	GEGENSTAND	ANFORDERUNG
2.0.0	Lastenheft Situations	spezifische Modellprüfung und -analysen
2.1.0	Modellierungsqualität Wandbauteil	
2.1.1	Grundlegende Modellierung	Prüfplan für beide Planungsalternativen gemäß der Abschnitte 1.0 – 1.2 zusammenstellen.
2.1.2	Modellierungsvorgaben zur Holzrahmenbauweise	Spezifischen Prüfplan für die Alternative Holzrahmenbauweise nach den Abschnitten 1.3 und 1.4 zusammenstellen.

2.1.3	Modellierungsvorgaben zur Massivbauweise	Spezifischen Prüfplan für die Alternative Massivbauweise nach den Abschnitten 1.3 und 1.4 zusammenstellen.
2.2.0	Ermittlung Energetischer Kennwerte	
2.2.1	Einzelbemessungswerte der Schichten und Komponenten	Bemessungswerte für die einzelnen Schichten gemäß Berechnungsvorschrift DIN EN ISO 6946 für die beiden Alternativen bestimmen.
2.2.2	Ermittlung des Wärmedurchgangswiderstands	Wärmedurchgangswiderstand der Wandkonstruktion gemäß Berechnungsvorschrift DIN EN ISO 6946 für die beiden Alternativen bestimmen.
2.2.3	Energetischen Kennwert für Zielabgleich aufbereiteten	Umrechnen des ermittelten Wärmedurchgangswiderstands der Wandkonstruktion gemäß Berechnungsvorschrift DIN EN ISO 6946 in den äquivalenten Wärmedurchgangskoeffizienten (U-Wert).
2.3.0	Analyse Zielerreichung	
2.3.1	SOLL-IST Vergleich Alternative Holzrahmenbauweise	Entspricht bzw. unterschreitet der berechnete U-Werte der Wandkonstruktion dem Geforderten für eine Außenwand gemäß SOLL-Werteliste (Tabelle 6, Zeile 3)?
2.3.2	SOLL-IST Vergleich Alternative Massivbauweise	Entspricht bzw. unterschreitet der berechnete U-Werte der Wandkonstruktion dem Geforderten für eine Außenwand gemäß SOLL-Werteliste (Tabelle 6, Zeile 3)?
2.4.0	Gegenüberstellung Planungsalternativen	
2.4.1	Gestaffelte Ergebnisse der vorherigen Abschnitte	Liegt ein Zielwertabgleich für die Wandkonstruktion als Folgeergebnis darunter gestuft durchgeführter Prüfschritte vor? Wenn NEIN: Welcher der gestuft unterlegten Prüfschritte verhindert das Gesamtergebnis (was ist dessen Einzelergebnisbericht, z.B. „Kein Materialobjekt von Komponente <i>IfcMember</i> „Pfofen“ verwiesen)?
2.4.2	Tabellarische Zusammenfassung	Wie lassen sich die beiden Gesamtergebnisse übersichtlich gegenüberstellen, zusammen mit den wesentlichen Einzelergebnissen der Vorprüfung bzw. -berechnungen?

5 Umsetzung der Ebene des dynamischen Anwendungsansatzes mit Werkzeugen für modellbasierte Analyseprozesse

Unabhängig von dem im vorherigen Kapitel konkretisierten Anwendungskontext bzw. dessen spezifischen Anforderungen, wird im ersten Teil der Umsetzung des Lösungsansatzes in diesem Kapitel zunächst die Realisierung benötigter Werkzeuge für die Ebene des Softwareanwendungsansatzes beschrieben. Ausgangspunkt dieser bezüglich der fachlichen Anwendungslogik (Produktmodellschema bzw. entsprechenden Analyselogik) neutral gehaltenen, technischen Umsetzung bildet ein Baukastenansatz einer modularen Gesamtarchitektur. Die darin entwickelten Komponenten (vgl. Kap. 5.1) setzen die verschiedenen dynamischen Anteile des Lösungsansatzes (bspw. regelbasiertes Prüfen und Analysieren, Editor zum Bearbeiten von Analyselogik, usw.) als einzelne, konfigurierbare Werkzeugmodule um. Durch die gleichförmige Einordnung in eine rahmengebende Schichtenarchitektur (vgl. Kap. 5.2) bei der Realisierung, können die Werkzeugmodule dann jeweils zu einer eigenständigen Softwareanwendung mit verschiedenem Werkzeugumfang zusammengestellt werden. Dabei werden die jeweiligen Oberflächen der Komponenten, die eine Nutzerinteraktion umsetzen, in einer Ebene der Benutzeroberfläche zusammengeführt, mit der sie einheitlich an die Geschäftslogik bzw. Datenebene der jeweils konfigurierten Anwendung angebunden werden können. Die wechselseitige Wirkungslogik der Komponenten wird aus technischer Sicht anhand der (Haupt-) Funktionen eines prototypisch umgesetzten Standardmodus einer Softwareanwendung für die Rolle von Endanwender*innen vorgefertigter Regeln (Modellprüfung und -analyse) verdeutlicht. Für den Rahmen einer Softwareanwendung zuschaltbaren Regelbearbeitungsmodus für die Rolle von Regel-Administrator*innen bzw. Regelentwickler*innen (Entwicklung und Verwaltung der Analyse- und

Prüflogik) wird anschließend die Einbindung weiterer Komponenten zur Erweiterung des Standardmodus der Software veranschaulicht. Dabei wird zum einen das technische Zusammenspiel der jeweils beinhalteten Werkzeuge (z.B. Modellprüfkonsole und Prüfberichtserstellung bzw. Regelbibliothek und Konfiguration der Regeln) erläutert. Zum anderen werden querschnittende Belange der umgesetzten Gesamtarchitektur verdeutlicht. Zum Beispiel die abgestufte Fehler- bzw. Ereignisbehandlung, die von der Ebene der einzelnen Werkzeugmodule in die übergeordnete, die Module zusammenführende Anwendungsarchitektur spannt und eine ganzheitliche Grundlage zur Benachrichtigung der Anwenderbene bereitet. Auf diesem Mechanismus bauen dann neben den laufzeitbezogenen Systembenachrichtigungen die Ereignismeldungen beim Ausführen von Prüf- und Analyselogik auf die Produktmodelle auf. Zum einen sind dies die in der Prüf- und Analyselogik veranlagten Ergebnisbenachrichtigungen, zum anderen ad hoc Fehlermeldungen an die Ersteller der Prüf- und Analyselogik bei deren Testdurchläufen zur Anzeige der Problemstellen in der erarbeiteten Logik.

5.1 Komponentenentwicklung als Grundlage modularer Softwarearchitektur

Modularer Aufbau Architektur

Für die skalierbare Entwicklung einer Desktop-Softwareanwendung wurde ein bestehender Ansatz eines komponenten-basierten Baukastensystems (vgl. BLM ToolKIT, Kap. 5.1.1) aufgegriffen und erweitert. Dabei ermöglichte das Unterlegen einer generellen Schichtenarchitektur die Eingliederung der Komponenten in drei grundlegende Bereiche – Benutzeroberfläche, Business Logic und Datenzugang. Aufgebaut wurde die Weiterentwicklung des Rahmenwerkes auf einer Ebene verschiedener (bereits generalisiert vorgehaltener) Basiskomponenten, die dann gemäß der unterlegten Gesamtkonzeption erweitert und ausgeprägt wurden. Im Weiteren wird speziell auf die Basiskomponenten hingewiesen, die von anderen Mitarbeitern am Fachgebiet des Autors entwi-

ckelten wurden bzw. bei deren Entwicklung der Autor mitgewirkt hat. Da sowohl die von anderen entwickelten wie auch die für diese Arbeit entwickelten Komponenten in einer rahmengebenden Softwarearchitektur der Desktop-Anwendung zusammengestellt sind, soll damit klar unterschieden werden können, welche der Komponenten im Rahmen dieser Arbeit entwickelt worden sind.

Thematisch werden in diesem Kapitel zunächst anhand ausgewählter Komponenten der prototypisch umgesetzten Software fundamentale Bausteine ihrer Architektur dargestellt. Korrespondierend zu ihrer im Rahmen der verfolgten Schichtenarchitektur (vgl. Kap. 5.2) verdeutlichten Wechselwirkung, gliedert folgende Struktur die wesentlichen Klassen bzw. realisierenden Komponenten, die die (Grund-)Funktionalitäten der Software nach ihren aufeinander aufbauenden Abhängigkeiten kapseln:

- A) Basiskomponenten
- B) Anwendungsebenen – Oberfläche, Geschäftslogik und Datenzugang
 - a) Data Access – Repository Muster zur Entkopplung der bedienten Datenstrukturen von der Werkzeugebene
 - i) Regel Repository und weitere XML-basierte Objektspeicher zur Datenbereitstellung für die Anwendung
 - ii) Persistierbarer Laufzeit Objektspeicher - Einbindung des BLM ToolKIT Bausteins DynamicObjects
 - iii) Einlesen und Zugang zu den Modelldaten
 - b) Business Logic
 - i) Laden funktionaler (modell-diagnostischer¹⁸) Kontexte

¹⁸ „model-diagnostischer Kontext“ enthält generalisierte (modellübergreifende) Funktionen und Methoden, deren modellspezifische Bestandteile durch Fallunterscheidungen aufgrund von eindeutiger Erkennungsmerkmale der behandelten Modellformate entsprechend zugeschaltet werden.

- ii) Umsetzung der regelbasierten Geschäftsprozesse (ToolKIT DynamicExecution)

c) User Interface

- i) Use Case und Ansichtsfenster zu den Arbeitsschritten
- ii) MVC Muster zur Anbindung der Benutzeroberfläche an die Business Logik

Aufgrund dieser beim Umsetzen der Komponenten für die Softwareanwendung unterlegten Struktur konnte auch des Baukastensystems des BLM ToolKIT weiterentwickelt werden. Prinzipiell lassen sich durch diese Rückkopplung mit der Erweiterung des ToolKITs, beispielsweise durch Entwicklungen neuer Basiskomponenten für Modelleditierung bzw. für weiterführende Anbindung von Persistenztechnologien (wie Datenbanken, webbasierte Datenspeicher usw.) auch die daraus abgeleitete Softwareanwendung in ihren einzelnen Schichten erweitern. Darüber hinaus kann die Gestalt der abgeleiteten Anwendungssoftware entlang der unterlegten Schichtenarchitektur auch generell angepasst werden. Neben dem prototypisch realisierten Dialogmodus für die Nutzerinteraktion lässt sich unter Wiederverwendung entwickelter Komponenten auch ein Stapelverarbeitungsmodus umsetzen. Dies erfordert, dass entsprechende Komponenten anstelle der Benutzeroberfläche eine Dienstschicht (bspw. eine API) realisieren. Damit werden Hintergrunddienstmodule ermöglicht, z.B. als Monitoringanwendung zur programmatischen, regelbasierten Inhaltsprüfung von Datenbanken. In der vorliegenden Arbeit wurde mit der umgesetzten Struktur die Grundlage dieser Art Erweiterungen geschaffen. Sie konnten dann bei weiterführenden Entwicklungen in weiteren, den Softwareansatz aufgreifenden Forschungsprojekten umgesetzt und demonstriert werden (vgl. Kap. 7.2).

5.1.1 Basiskomponenten des BLM ToolKIT

Am Institut des Autors wurden im Rahmen eines DFG Projektes Vorläuferkomponenten für eine aspektorientierte modellbasierte Datenhandhabung entwickelt (Hartmann & Both, 2015), auf denen Teile der Umsetzung der Software

basieren. Die ganze Entwicklung der vorliegenden Arbeit wurde aufgrund der Entscheidung, teilweise auf die im BLM ToolKIT verwalteten Basiskomponenten aufzubauen, ebenfalls in Microsofts Laufzeitumgebung .NET (v. 4.5) umgesetzt.

Eine erste, nicht selbst entwickelte Basiskomponente `DynamicExecution` ermöglicht für die streng typisierte Sprache C# das dynamische Kompilieren und Ausführen von Quelltext zur Laufzeit.

Insbesondere für eine schemaneutrale Modellhandhabung bildet diese Komponente zur beliebigen Einbringung von (neuem) Quelltext in das bereits laufende Programm eine flexible Behandlungsmöglichkeit verschiedener Modellstrukturen, ohne die Software jeweils neu Erstellen zu müssen. Das in Abbildung 31 gezeigte Interface wird beim Einbinden der Komponente implementiert. Damit kann dann in der Anwendung die Methode `CreateClassInstance` zum Übersetzen eines im Feld `CSharpCode` hinterlegten Quelltextes genutzt werden. Die Klassenmethoden einer übersetzten und kompilierten Instanz (`ClassInstance`) des eingebrachten Klassen-Quelltextes lassen sich dann über die Helfermethode `ExecuteMethod` aufrufen.



Abbildung 31 Interface der Basiskomponente `DynamicExecution`

Durch eine weitere nicht selbst entwickelte Basiskomponente `DomainObjects` wird im Zusammenspiel eine dynamische Laufzeitkollektion für Objekte (Objektpool) erstellt. Dabei wird Quelltext zur Modellbehandlung bedarfsorientiert übersetzt und ausgeführt (Lazy-Load-Prinzip) und für die laufenden Prozesse der Software bereitgestellt. Diese Komponente ermöglicht dabei als Ergänzung zur Datenebene die Datenfelder die den Quelltext als Wert beinhalten in Form einer XML-Datei zu persistieren. Diesbezüglich sind für diese Datenfelder entsprechende Wertetypen mit Parsing-Methoden sowie XML-Bindungen umgesetzt. Die in Abbildung 32 dargestellte Hauptklasse der

Komponente nutzt für das zu persistierende Klassen-Feld `PoolVariables` eine Implementierung der .NET XML-Schnittstelle `IXmlSerializable`, die deren Inhalte dann an entsprechende XML Elemente bidirektional anbinden und ein Lesen und Schreiben entsprechender XML-Instanzdokumente ermöglichen. Um die Basisfunktionalitäten eines .NET Kollektionstyps zu erben, leitet die Klasse des Wertetyps `SerializableDictionary` von der .NET Basisklasse `Dictionary` ab. Bei der Handhabung der entsprechenden deserialisierten `PoolVariables`-Objekte als Objektlexikon können zur Laufzeit damit dann z.B. die Methode `Add` oder die Schlüsselmengeneigenschaft `Keys` der Basisklasse genutzt werden.

Die Komponente baut funktional auf das in .NET 4 in die Laufzeitumgebung aufgenommene Prinzip dynamisch erweiterbarer Klassen (.NET Klasse `DomainObject`) auf. Ein damit umgesetzter Ansatz eines erweiterten Lexikonobjekts erlaubt es, zur Laufzeit dynamisch Einträge (Schlüssel – Wert Paare)

in das Feld `PoolVariables` hinzuzufügen. Hervorzuheben ist, dass ein initial dabei hinterlegter Wert zunächst Quelltext ist, der zum einen dann über den oben beschriebenen Wertetyp `SerializableDictionary` persistiert werden kann. Zum anderen wird beim ersten Abruf des `PoolVariables`-Eintrages über seinen Schlüssel dann (im Zusammenspiel mit der `DynamicExecution` Komponente) im Hintergrund der zunächst als Quelltext hinterlegte Wert des Eintrag-Tupels übersetzt und ausgeführt. Der aufrufenden Seite wird zuletzt der Rückgabewert einer Methode von der dynamisch erzeugten Klasseninstanz zurückgegeben.

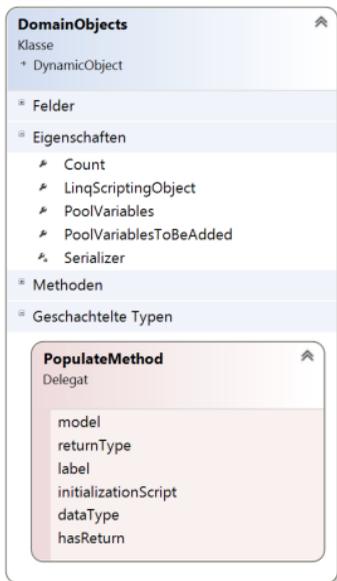


Abbildung 32 DomainObjects Komponente

So kann beispielsweise hinter einem Schlüssel „Wände“ initial der Quelltext ei-

ner Methode „AlleWände“ zum Filtern aller Wandobjekte aus einem Bauwerksinformationsmodell hinterlegt sein, wobei dieser Quelltext durch ebenfalls ermöglichte Abhängigkeiten der Einträge untereinander auf einem Schlüssel „Modell“ (mit hinterlegtem Quelltext zum Einlesen eines Bauwerksinformationsmodelles) aufbaut. Wenn der Schlüssel „Wände“ in dem Lexikonobjekt aufgerufen wird, werden zunächst alle Schlüssel von denen der hinterlegte Quelltext abhängt aufgerufen, d.h. Rückgabe des Schlüssels Modell ist das als interne Repräsentation (z.B. Objektgraf) geladene Bauwerksmodell. Hierauf liefert dann der im Hintergrund an die dynamisch erstellte Methode „AlleWände“ weitergeleitete Aufruf eine Kollektion an Wandbauteilen des Bauwerksinformationsmodelles zurück. Die durch selbstvergebene Schlüsselworte ermöglichte Abstraktion, die eine hinterlegte (komplexe) Logik vor dem Anwender verdecken, bildet eine wesentliche Grundlage der für die vorliegende Arbeit prototypisch umgesetzten Softwareanwendung.

5.1.2 Weitere (eigenständige) der Desktop-Softwareanwendung zugrundeliegende Komponenten

5.1.2.1 ErrorHandling Komponente – generalisierte Behandlung von Fehlern und Ausnahme

Bezugnehmend zu Helfert (2002) steht im Ausgangspunkt dieser Komponente ein wichtiger Aspekt zum Aufbau von aufeinander aufbauenden Messstufen, wobei diese bezüglich der Überprüfung eines Datenwertes nicht vollständig durchlaufen werden müssen, da ein in einer Stufe festgestellter Fehler das Prüfen der nächsten Stufe obsolet macht (vgl. Kap. 2.3.3). Hiervon ist das Leitmotiv der Komponente abgeleitet: sämtliche (nicht standardmässig behandelte) Ereignisse bzw. Fehler zur Laufzeit der Software – von denen der technischen Programmebene bis zu denen der nutzer*innenbezogenen Ebene der regelbasierten Prüf- und Analyselogik für Produktmodelle – in kaskadierten Behandlungsmethoden abzufangen.

Zielstellung: Mit der abgestuften Fehler- bzw. Ausnahmebehandlung, die von der Ebene der einzelnen Komponenten und Werkzeugmodule (Programmierung) in die übergeordnete, die Softwareteile zusammenführende Desktop-Anwendung (vgl. Businesslogikschicht) spannt, sollen in einer ganzheitlichen Grundlage zu sämtlichen Ereignissen der Laufzeit der Software entsprechende Benachrichtigungen für die Benutzeroberfläche (bzw. des Anwendungsprotokolls) aufbereitet werden. Neben den laufzeitbezogenen Systembenachrichtigungen bauen dann auch die Ereignismeldungen zur Ausführung von Prüf- und Analyselogik auf die Produktmodelle auf diesem Mechanismus auf. Zum einen sind dies die in der Prüf- und Analyselogik veranlagten Ergebnisbenachrichtigungen, zum anderen ad hoc Fehlermeldungen an die Ersteller der Prüf- und Analyselogik bei deren Testdurchläufen zur Anzeige der Problemstellen in der erarbeiteten Logik.

Umsetzung: Die Komponente ist unter Mitwirkung von Mitarbeitern des Fachgebiets entwickelt worden. Die Abbildung der beiden Hauptbereiche möglicher Ereignismeldungen – Systembedingte Ausnahmeereignisse und Regelbedingt ausgelöste Meldungen – werden in den zwei von der Basisklasse `ErrorItem` ableitenden Klassen `ApplicationError` sowie `ModelError`

gekapselt (siehe Abbildung 33). Insbesondere bei der Realisierung der `DynamicExecution` Komponente ist das Abfangen und strukturierte Behandeln von Ausnahmen wichtig, da in der Komponente die Kontrolle, Kompilierung und schließlich Ausführung von Programmiersprache in eigenen Prozessen abläuft. Die Kommunikation findet hierbei nur über sogenannte Callback Methoden statt, die jeweils zu bestimmten Punkten in den externen Abläufen Rückmeldungen an den Hauptprozess z.B. als

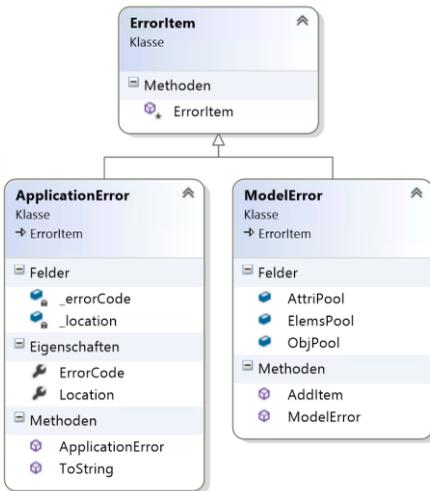


Abbildung 33 ErrorHandler Komponente

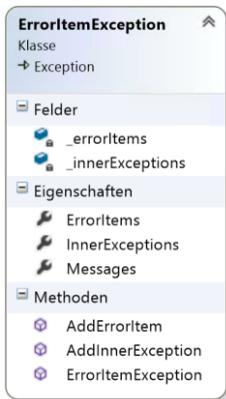


Abbildung 34 Klasse zum gestaffelten Sammeln von Ereignissen

Statusmeldung geben. Zur genauen Verortung von den durch die `ErrorHandling` Komponente verarbeiteten Ereignissen folgen alle Ausnahmebehandlungsmethoden (in .NET C# werden diese mit den Vokabeln `try` und `catch` definiert) einer gestaffelt veranlagten Konvention. Kann ein Ereignis nicht direkt an der Stelle, an der es entsteht, behandelt werden, wird in einer Instanz der `ApplicationError` Klasse eine entsprechende Meldung abgelegt. Diese Meldung steht dann, durch die in verschiedenen Kaskaden gestaffelt angelegten Methoden zur Ausnahmebehandlung der nächsten, das Ereignis abfangenden zur Verfügung. Ist an dieser Stelle auch keine Behandlung des Ereignisses möglich, werden sowohl die hereingereichte sowie eine neuerstellte Instanz mit einer Meldung von dieser Stelle an die übergeordnete

Ereignisbehandlungskaskade weitergereicht. Zur Umsetzung des an die `Exception` Konzeption von C# angelehnten und dieses in den Quelltext der Software erweiternde Prinzip der Ereignisbehandlung leitet die Klasse `ErrorItemException` stellt von dieser Basisklasse `Exception` ab (vgl. Abbildung 34). Mit dem Kollektionsfeld `ErrorItems` für die Sammlung der gestaffelt erhobenen Meldungen wird dann das `Exception` Fazität der `InnerExceptions` erweitert.

Können die Meldungen von keiner der übergeordneten Kaskaden zur Behandlung des Ereignisses genutzt werden, wird dieses immer weitergereicht, bis es schließlich von der äußersten Kaskade als systemimmanente Ausnahme oder als im Protokoll zu dokumentierende bzw. den Nutzer vorzulegende Ereignismeldung behandelt wird. Somit kann durch die Auswertung von einem das Ereignis begleitend instanziierten `ErrorItemException` Objekts – über die Grenzen der Prozesse der Software hinweg – stets eindeutig die Stelle des Ereignisursprungs bestimmt werden.

5.1.2.2 ProjektManagement Komponente

Grundlegende, den Anwender*innen angebotene Funktionen zur Modellhandhabung werden über die Klasse `ProjectManager` gesteuert bzw. laufzeitbezogen zusammengestellt. Im Hintergrund des Designs dieser Komponente steht eine Userstory, wonach jeweils in einem Projekt ein Vorgang zur Modellprüfung bzw. Analyse kapselt wird. Die Komponente realisiert dies, indem für die Verarbeitung verschiedener Produktmodellformate jeweils ein spezifischer Kontext aufgestellt wird. Da alle (Geschäfts-) Logik, die das Handhaben der unterschiedlichen Modellformate ermöglicht, in Quelltextform vorgehalten

wird, werden an dieser Stelle schemaneutrale, die Basisnotation XML betreffende Mustermethoden verwaltet. Die Handhabung der Methoden wird dabei von der Basiskomponente `DomainObjects` realisiert (vgl. 5.1.1), die wie in Abbildung 36 veranschaulicht in den Mustermethoden mit dem vorgehaltenen Quelltext instanziiert wird und dann die designierte Hilfsmethode zur Verwendung zentralen Objektspeicher `POOL` bereitstellt. Zum Beispiel ermöglicht eine Methode das Ausprägen einer Quelltext-Schablone zur Filterung eines Instanz-Verweis Objektpaares (auf Basis des Verweisprinzips von `id` – `href` XML-Attributpaaren) mit spezifischen Klassennamen eines bestimmten Modellschemas (vgl. Abbildung 36). Der Funktionsumfang für den Handhabungs-Kontext zum entsprechenden Modellschema wird durch Kompilieren und Bereitstellen daraus gewonnener modellspezifischer Filtermethoden erweitert. Neben diesen dynamischen kontextbestimmenden Eigenschaften werden im `ProjectManager` (siehe Abbildung 35) schemaspezifische Standardeinstellungen verwaltet. So ermöglichen beispielsweise Standard Enumerationen, wie `DefaultQueryLabel`, das Vorhalten bestimmter Werte zu einem Modellformat, die beim Aufbereiten

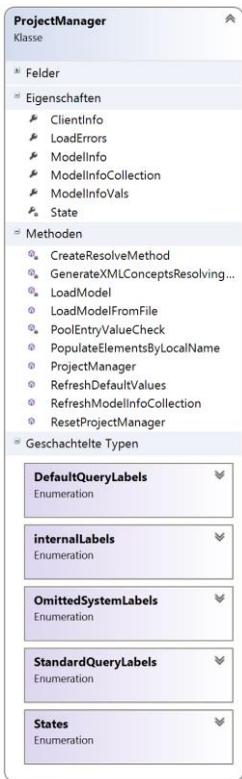


Abbildung 35 Projekt Management Komponente

eines Kontextes dann neben den modell-diagnostisch zusammengestellten Konfigurationen für Standardeinstellungen, wie dem Zusammenstellen einer Modellübersicht beim Modelleinlesen, genutzt werden können.

```

if (((DomainObjects)ModelInfo.POOL).GetDictValueBase("GetReferencedElement") == null)
{
    // lets add some model domain specific resolve function
    mkey =
        "GetReferencedElement(System.Xml.Linq.XElement refInstance)";
    resolveMethodDvb = new DictValueBase()
    {
        MethodName = mkey,
        CSharpDataTypeString = "System.Xml.Linq.XElement",
        SystemDataTypeString = typeof(XElement).FullName,
        IsIntrinsic = true,
        Description = "na",
        MethodBlockString =
            @"
                XAttribute refAttr = refInstance.Attribute("ref");
                XAttribute hrefAttr = refInstance.Attribute("href");

                // CASE 1: return the handed in 'REF' instance if does not have a href/ref attribute
                if(refAttr == null && hrefAttr == null)
                    return refInstance;

                // CASE 2: return the element with an corresponding id attribute
                return (
                    from XElement elem in ((XElement)POOL.Model).Descendants() as IEnumerable<XElement>
                    let idAttr = elem.Attribute("id")
                    where idAttr != null
                    && ((refAttr != null && refAttr.Value == idAttr.Value)
                    || (hrefAttr != null && hrefAttr.Value == idAttr.Value))
                    select elem
                ).First();"
    };
}

```

Abbildung 36 Modell-diagnostisch vorgehaltener Quelltext für Hilfsmethode

Wesentlich für die unten beschriebene Business Logic der Software werden neben Informationen zu den Anwender*innen in einem Projektkontext auch die zu prüfenden bzw. zu analysierenden Modelle verwaltet. Sie werden jeweils in einer Instanz der **ModelInfo Klasse** mit den spezifischen Modellinformationen gehalten. Eine beim Einlesevorgang bereits zuschaltbare Erstausswertung der Modellinhalte kann mit der **ModelInfoCollection** aus dem in der **ModelInfo** Instanz gebildeten dynamischen Modelllexikon (vgl. folgendes Unterkapitel sowie Objektpool in Kap. 5.1.1) gelesen werden. Dies geschieht nach dem Lazy-Load Prinzip, in dem allein der entsprechende Quelltext der Auswertung im Lexikon hinter einem Schlüsselbegriff hinterlegt. Dessen mit

der Methode im `ProjectManager` verknüpfter Aufruf startet dann erst den (ggf. kostspieligen) Vorgang.

5.1.2.3 Komponente zur Verwaltung der Modelldaten

Prinzipiell können zwei Strategien zur Handhabung eines XML-basierten Dokuments mit dem in der `ModelInfo` Klasse (vgl. Abbildung 37) verfolgten Ansatz realisiert werden:

- ein direkter Zugang zu den in einer Datenbank persistierten Daten
- ein Datei-basierter Zugang zu einem im Verzeichnissystem abgelegten XML-Instanzdokument.

Aus Performancegründen erscheint bei der ersteren Strategie insbesondere im Rahmen der Handhabung großer XML-basierter Modelldokumente neben dem reinen Auslesen des gesamten Dokuments (äquivalent zum Aufruf eines Dokuments im Dateisystem) auch ein direktes Ausführen von Abfragen auf die Datenbank interessant. Zu Umsetzungen dieses vielversprechenden Ansatzes¹⁹ mit nativen XML-Datenbanken und dem offenen Abfragestandard XQUERY, der im Rahmen der vorliegenden Arbeit nicht weiterverfolgt werden konnte, sei auf weiterführende Arbeiten des Autors im Rahmen von Forschungsprojekten im Ausblick verwiesen. In der im Rahmen dieser Arbeit entwickelten Komponente wurde das Prinzip des Einlesens sämtlicher Modelldaten in Form eines Domain Object Models (DOM) umgesetzt, und zwar auf Basis der von Seiten des .NET Framework bereitgestellten LINQ Komponente `LINQ2XSD` (vgl. Kap. 2.3.2). Vorbereitend zur Prüfung bzw. Analyse der Modellinhalte werden dabei alle XML Elemente des Modelldokuments in eine interne Repräsentation mit einer an der XML-Notation ausgerichteten Struktur übersetzt. Im Gegensatz zu einem direkten Binden der Elemente der internen Repräsentation an die Klassen der XML-Anwendungsschemas, das ein direktes Aufrufen der Elemente mit den im Schema vergeben Namen ermöglichen würde, besteht der

¹⁹ In der Open Source PostGre Datenbank stehen intrinsische Funktionalitäten zu trigonometrischen Operationen zum Modellhandling von Geodaten zur Verfügung. Seitens offener nativer XML-Datenbanken sind diese internen Berechnungsmöglichkeiten nicht verfügbar.

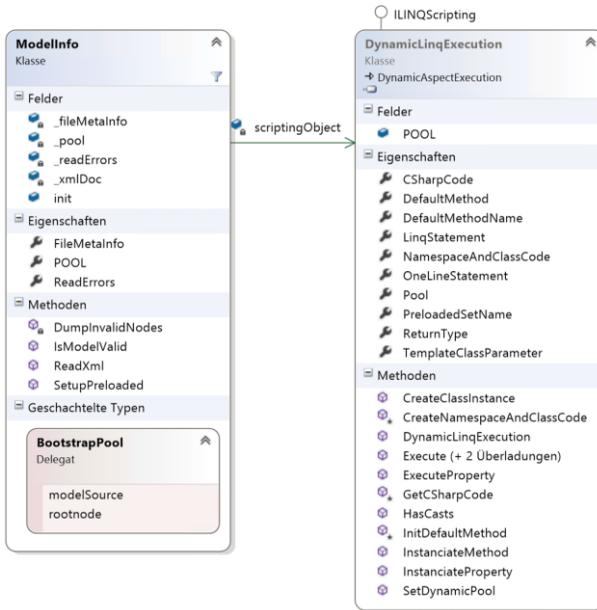


Abbildung 37 ModelInfo Klasse und implementiertes Interface zur Basiskomponente DynamicExecution

Modellschemas mit Klassenbibliotheksansätzen²⁰ verfolgen, durch erhebliche Geschwindigkeitsvorteile ab. Beim Einlesen des XML-Dokuments werden mittels einer weiteren Komponente des .NET Frameworks die Schemakonformität des Produktmodells geprüft und gegebenenfalls entsprechende Unstimmigkeiten und Fehler gemeldet (vgl. Kap. 5.1.2.1). Der eingelesene Objektgraf bildet dann den Ureintrag des auf Basis der *DomainObjects* Komponente (vgl. Kap. 5.1.1) umgesetzten (Modell-) Objektlexikons POOL, das den zentralen (persistierbaren) Objektspeicher aller Geschäftsobjekte zur Laufzeit der Softwareanwendung realisiert (vgl. Kap. 3.2.2). Über die POOL-Eigenschaft der *ModelInfo* Klasse kann mit dem standardmäßigen Schlüssel *Model* di-

von LINQ generierte Objektgraf aus XElement Objekten, die erst über eine Abfrage ihres Namenattributs den über das Produktmodell-schemadefinierten Elementnamen preisgeben. Diese für spätere Abfragen dadurch kostspieligere Abstraktion setzt sich insgesamt im Vergleich zu bestehenden Lösungen, die ein direktes Binden der

²⁰ Vgl. XML Data Binding in Kapitel 2.3.2 sowie e.g. IFC Engine DLL (<http://rdf.bg/product-list/ifc-engine>), IFC Framework (<http://www.apstex.com/>) oder IfcOpenShell (<http://www.ifcopenshell.org/>)

rekt auf den Eintrag mit den (statisch) eingelesenen Modelldaten im Objektlexikon zugegriffen werden. Damit auch „dynamische“ Einträge im POOL registriert werden können, deren zunächst als Quelltext hinterlegte Werte dann beim Aufruf des entsprechenden Schlüssels kompiliert und ausgeführt den eigentlichen Wert zurückliefern, ist ein wechselseitiger Initialisierungsvorgang der miteinander verflochtenen Basiskomponenten `DomainObjects` und `DynamicExecution` zum Bilden des Objektlexikons erforderlich. In Abbildung 37 ist für den ersten Schritt hierfür die Einbindung der Basiskomponente `DynamicLinqExecution`, welche die `DomainObjects` Klasse als Wertetyp ihres Feldes `POOL` anbindet, in die `ModelInfo` Klasse dargestellt. Dabei wird im Feld `scriptingObject` das `ILinqScripting` Interface implementiert und von der darin dann hinterlegten Instanz der `DynamicLinqExecution` Klasse dann im zweiten Initialisierungsschritt das `POOL`-Feld durch die `POOL`-Eigenschaft der `ModelInfo` zugänglich gemacht.

Direkt nach Initialisierung und Einlesen eines Produktmodells können somit auch dynamische Einträge im Objektlexikon registriert werden, deren Quelltext auf den `Model` Schlüssel Bezug nimmt, um damit beispielsweise den Anwender*innen einen Überblick zu Metadaten und Inhalten des geladenen Modells aufzubereiten. Das im Feld `POOL` gemäß des `DynamicExecution` Mechanismus (vgl. auch Kap. 5.1.2.4) aus dem implementierten Interface verknüpfte Objektlexikon `DomainObjects` bildet den Kern der Klasse `ModelInfo`, die darüber hinaus Metadaten zum Modell, wie Speicherort, Dokumentname usw. in entsprechenden Feldern vorhält. Bei jedem durch die Anwender*innen ausgelösten Modelleinlesevorgang bildet die verwaltende `ProjectManager` Komponente jeweils eine neue `ModelInfo` Klasseninstanz, die dann neben dem als Objektgraphen repräsentierten Modelldokument mit der `POOL`-Variable ein modellspezifisches zentrales Objektlexikon bereitstellt, in das alle Vorgänge zum Modell einfließen. So werden darin im weiteren Prüfungs- bzw. Analyseverlauf durch das Ausführen von Regeln auf das eingelesene Modelldokument dann jeweils die zur Laufzeit kompilierten Regeleinheiten abgelegt. Dabei wird den Regeln durch Einbinden der `POOL`-Variable in ihrem Quellcode der Zugang zum aktuell geladenen Modell sowie die Möglichkeit zum Hinzufügen ihrer eigenen spezifischen Einträge (z.B. zu be-

stimmten Mengen usw.) zur Verfügung gestellt. Die `ModelInfo` Klasseninstanz bildet somit den zentralen (laufzeitbezogenen) Datenspeicher eines Prüfung- und Analyselaufes, der wie ein Staffelpfad von Vorgang zu Vorgang weitergereicht und dabei angereichert wird, bis er zuletzt bei der Auswertungskomponente zur Ergebnisberichterstattung Verwendung findet. Da sämtliche Quelltextanteile darin gemäß dem zugrundeliegenden Prinzip der `DomainObjects` Komponente in eine XML-Datei persistiert werden können, bildet sie die (replizierbare) „einzige Wahrheitsquelle“ (eng. *Single Source of Truth*) sämtlicher zur Laufzeit durch die Anwender*innen behandelten Informationen (Modelle sowie Regeln). Hierdurch sind, sowohl zu Archivierungszwecken wie auch zum Vorhalten bestimmter Kontexte/Randbedingungen im Rahmen der Anwendung der Software entsprechende Grundlagen gegeben.

5.1.2.4 RuleManager und RuleHandling Komponenten

Zur Handhabung der in eigenständigen XML-Regeldateien gehaltenen Prüf- und Analyselogik für die Produktmodelle werden alle wesentlichen Vorgänge durch die `RuleManager` Klasse gesteuert. Neben der Anbindung des oben beschriebenen zentralen Objektspeichers `POOL` (vgl. Kap.5.1.2.3) kapselt sie die Methode für den zentralen (iterativ ausführbaren) Mechanismus zur Abarbeitung der Regeln. Dabei verwendet sie verschiedene `Engine` Klassen der `RuleHandling` sowie `Reporting` Komponenten, um die einzelnen darüber ausführbaren Vorgänge in dieser Gesamtprozedur zusammenzuführen. Beispielsweise zeigt Abbildung 38 die `RuleEngine` Klasse der `RuleHandling` Komponente, die mittels einer erweiterten Ableitungsklasse `DynamicRuleExecution` die Basiskomponente `DynamicExecution` zur laufzeitbezogenen Kompilierung und Ausführung der als Quelltext aus den XML-Dateien ausgelesenen Regellogik implementiert. Dabei setzt sie die Handhabung des mit dem Laden des Produktmodell initialisierten und durch den Objektgraphen der Modellinformation initial befüllten (vgl. Kap. 170) Objektlexikons auf Grundlage der von den Basiskomponenten `DynamicExecution`, sowie Basiskomponente `DomainObjects` (vgl. Kap. 5.1.1) bereitgestellten Methoden fort.

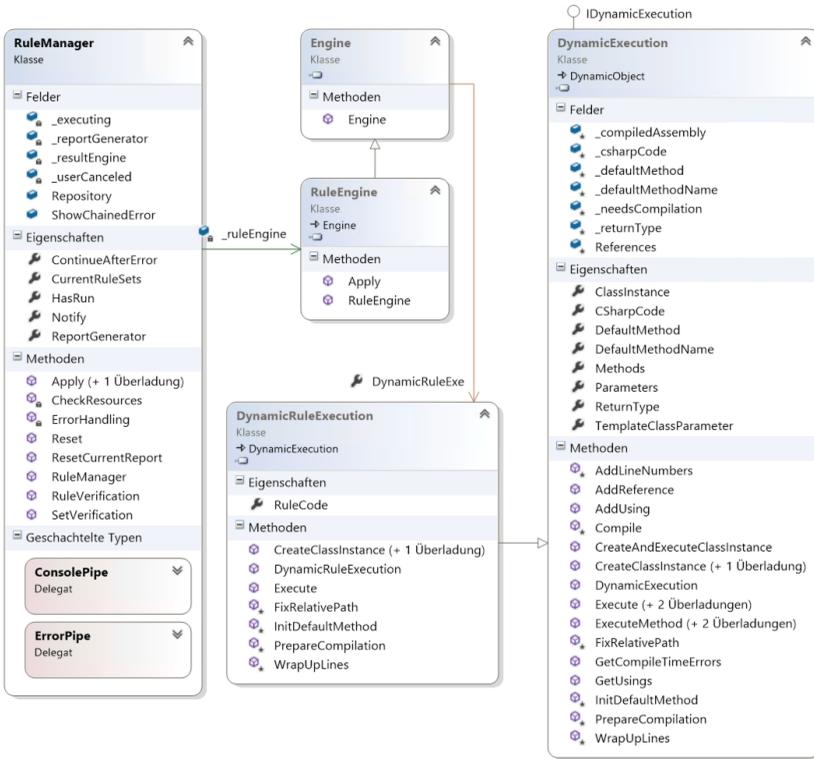


Abbildung 38 RuleManager Komponente sowie durch die RuleHandling Komponente implementierte Basiskomponente DynamicExecution

5.1.2.5 ReportManager

Bei der Umsetzung dieser Komponente wurde der Autor von Mitarbeitern des Fachgebiets unterstützt. Als letztes Glied in der Ausführungskette des Durchlaufs einer regelbasierten Modellprüfung bzw. Analyse werden mit der Berichtskomponente die bereits zum Ablauf der Regel durch den RuleManager ausgelöst, in der Regel hinterlegten Ereignisberichte zusammengestellt. Die Komponente wird dafür schon beim Instanzieren der Klasse RuleManager für einen Regelausführungsvorgang aufgerufen und legt im Speicher ein leeres Tabellendokument an. Hierfür nutzt sie die EXCEL Komponente des .NET

Frameworks, welches dann auch für die Möglichkeit der Erstellung eines entsprechenden Dokuments zum abschließenden Export eines Ergebnisberichts genutzt werden kann. Darüber hinaus ist dadurch auch das Speichern als Komma-separiertes Tabellenformat (CSV) möglich.

5.2 Integration der Komponenten in die Architektur der Desktop- Anwendungssoftware

Die im Rahmen des BLM ToolKIT entwickelten Komponenten (vgl. Kap. 5.1 sowie Abbildung 39; grün – weiterentwickelt und blau – neuentwickelt) bilden die Basis, auf der die prototypische Softwareanwendung im Rahmen der vorliegenden Arbeit als Desktop-Softwareanwendung implementiert wurde.

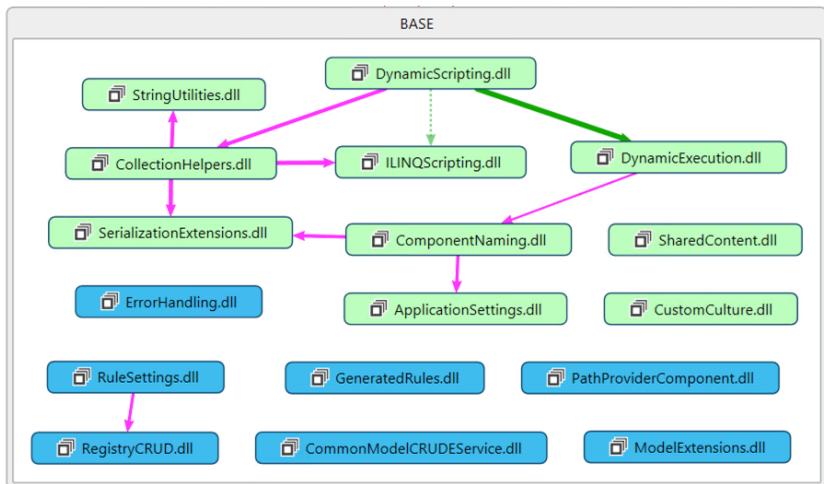


Abbildung 39 Verwendete bzw. (weiter-)entwickelte BLM ToolKIT Komponenten

Darüber hinaus entwickelte Komponenten gliedern sich zur besseren Skalierbarkeit in eine Schichtenarchitektur ein. In Abbildung 40 sind diese Ebenen der Softwareanwendung in einem Komponentendiagramm mit entsprechenden

Abhängigkeiten dargestellt. Neben den Basiskomponenten des ToolKIT ordnen sich die Komponenten in drei wesentliche Schichten ein:

- Benutzeroberfläche (UI –User Interface) Ebene
- Business Logic (BL) Ebene
- Daten (DA – Data Access) Ebene

Konzeptionell folgt diese Schichtenarchitektur einerseits zur losen Kopplung der Benutzeroberfläche mittels Data Binding Methoden dem Model View Controller (MVC²¹) Entwurfsmuster in der UI-Schicht (vgl. Kap. 5.2.1). Andererseits greifen zur Entkopplung der Persistenztechnologien die für die Schichtenarchitektur entwickelten Komponenten das Repository Entwurfsmuster und die damit bezüglich der Datenhaltung angesetzten Data Binding Prinzipien in der Schicht des Datenzugangs (vgl. Kap. 2.3.2) auf. Die Gesamtsteuerung sowie das Zusammenschalten der Einzelkomponenten zu Endnutzerfunktionalitäten im Rahmen von Werkzeugmodulen der Softwareanwendung werden von den Komponenten der Business Logic Schicht (vgl. Kap. 5.2.2) realisiert. Sie verbindet somit die Komponenten der beiden erstgenannten Schichten und bildet den Kern der Gesamtarchitektur. Im letzten Unterkapitel 5.2.2.4 wird dann auf die Schnittstellen der Datenebene zu den persistenten Daten eingegangen. Dabei werden die im Rahmen der vorliegenden Arbeit in einem Dokumentenbasierte Ansatz persistierten Nutzerdaten, ihre Formate sowie Ablagestrukturen verdeutlicht. Die Oberflächen der Softwareanwendung selbst sowie die integrierten Werkzeuge und deren Handhabung werden dann aus Sicht der Nutzer*innen mit den entsprechenden Nutzerinteraktionen in Kapitel 6.2.2.2 im Rahmen der Evaluation vorgestellt.

²¹ Entwurfsmuster zur Trennung der Oberflächenelemente von den Geschäftsdaten bzw. -vorgängen (ursprünglich Gang of Four); daneben lehnt sich die vorliegende Arbeit u.a. an konzeptionelle Weiterentwicklungen im Rahmen des Model View View-Model (MVVM) Entwurfsmusters an.

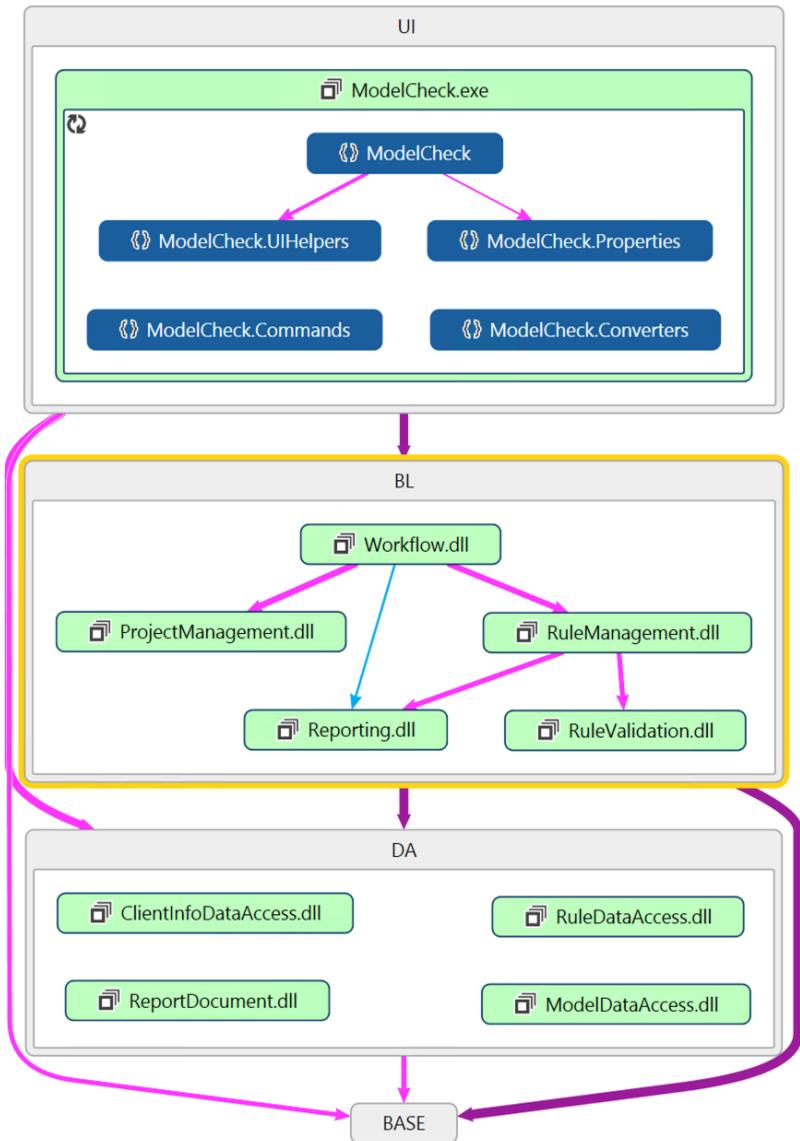


Abbildung 40 Architektur- und Abhängigkeitsdiagramm der Desktop-Anwendung

5.2.1 Benutzeroberfläche (UI) Ebene – Hauptfenster und enthaltene Werkzeugansichten (Views)

Technologisch wurde in der User Interface (UI) Schicht eine Windows Präsentation Framework (WPF) Applikation auf Basis des Microsoft .NET Frameworks umgesetzt. Bei dieser Applikationsart liegt der Einstiegspunkt in den Kontrollfluß der Anwendung standardmäßig in der beinhalteten APP.CS Klasse. Einfachheitshalber wurde anstelle einer eigenen Dienstschicht (engl. service layer), aus welcher der Kontrollfluß unabhängig zu der UI Schicht heraus programmatisch gestartet werden kann, ein Bereitstellungspunkt (engl. stub) innerhalb der APP Klasse (siehe Abbildung 41) implementiert. Obschon damit die Trennung von Belangen, wie es ein strikt eingehaltenes MVC Entwurfsmuster vorsieht, nicht vollständig eingehalten wird, kann somit vor den Initialisierungsmethoden der Benutzeroberfläche der Kontrollfluss an die Business Logic Schicht (vgl. Kap. 5.2.2) übergeben werden. Mit der Initialisierung der Business Objekte werden alle Funktionalitäten der Software, sowie die Controller zur Steuerung der Ansichten in der Oberfläche in die Laufzeit geladen.

Die Ansichten (engl. View) der Benutzeroberfläche sind in einer Hauptansicht (MainWindow.xaml) sowie untergeordnete Werkzeugansichten (bspw. Daten öffnen – ModelSelection.xaml, Regelauswahl – RuleSelection.xaml usw.) gegliedert. Ein Registerkarten-Steuerungselement stellt in der Hauptansicht den Einschub für die als einzelne Reiter darin geladenen Werkzeuge dar. Die Inhalte des aktuellen Reiters werden dann durch die Steuerungselemente der jeweiligen Werkzeugansicht dargestellt.

Für Steuerelemente, die in verschiedenen Werkzeugansichten wiederkehrend dargestellt werden, sind die entsprechenden Ansichten in einem übergeordneten Ressourcen-Lexikon vorgehalten. Durch entsprechende Bindungen in der Werkzeugansicht werden sie anforderungsorientiert wiederverwendet. Die Ansicht eines im Modell-Laden Werkzeug mit generellen Daten zum geladenen Modelldokument aufbereiteten Informationskastens kann somit bspw. in anderen Ansichten, wie der Ergebnisausgabe, wiederverwendet werden.

Über zentrale Steuerelementansichten des Hauptfensters werden den Anwender*innen neben den in einer Abfolge stehenden (zuschaltbaren) Werkzeugansichten auch weitere ständig bedienbare Steuerelemente, wie das Menu zur Konfiguration der Software bzw. Ein-/Ausschalten von Zusatzmodulen präsentiert.

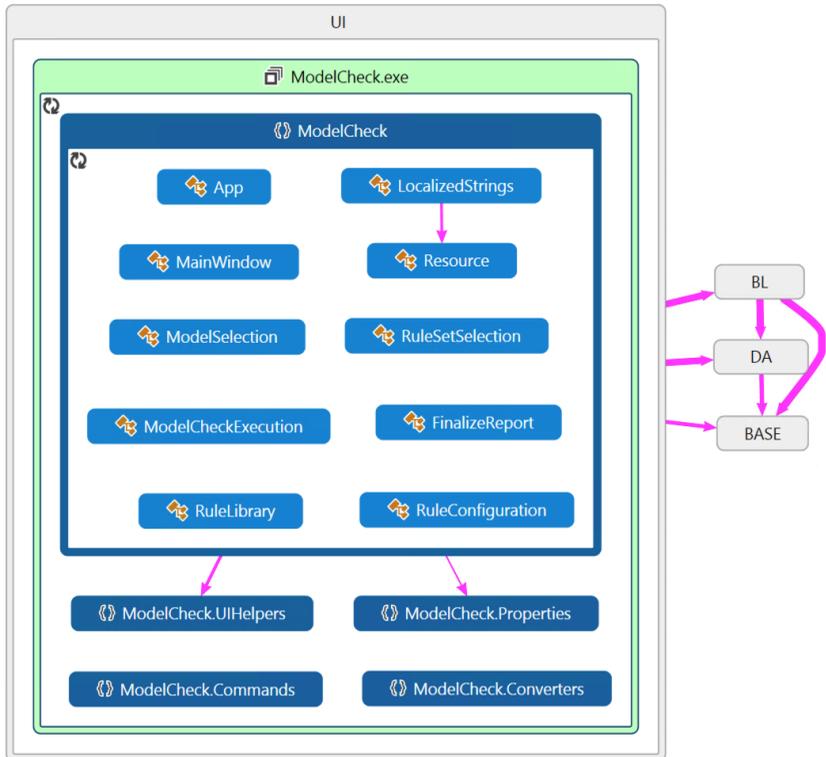


Abbildung 41 Klassen der wesentlichen Ansichten der Benutzeroberfläche

5.2.2 Business Logic (BL) Ebene – Steuerung der Nutzerinteraktion und Datenhandhabung

5.2.2.1 Workflow – RootController, Controller und Viewmodels

Den zentralen Kern des Kontrollflusses der Softwareanwendung bildet die Klasse `RootController` der `Workflow` Komponente. Beim Starten werden im Rahmen der Instanziierung eines einzigen Objektes dieser Klasse alle weiteren benötigten Objekte in einer Initialisierungsprozedur nacheinander erstellt. Im Wesentlichen sind dies verschiedene Manager Module, mit welchen die Geschäftsprozesse gesteuert werden, sowie Steuerungsklassen (engl. Controller) der Nutzerinteraktion und Datentransferobjekte zur Übertragung spezifischer Geschäftsdaten an die Oberfläche. Die beiden Letztgenannten entsprechen dabei im MVC Entwurfsmuster den in der Business Logic Schicht vorgesehenen Teilen – Model bzw. View Model und Controller – die jeweils einer Ansicht – View (vgl. Abbildung 42) – der UI Schicht zugeordnet sind. In verschiedenen Ansichten wiederverwendete Funktionalitäten werden in einer Basisklasse `Interaction` aller Controller gebündelt. Äquivalent werden gemeinsame genutzte Daten, wie bspw. eine Liste der aktuell ausgewählten Regeln, in einer allen (View) Modellen gemeinsamen Basisklasse `CommonUI` gehalten. Diese beiden Basisklassen werden jeweils nur einmalig instanziiert und dann durch den `RootController` durch alle Vorgänge hinweg in allen ableitenden Objekten synchronisiert, so dass sie in allen Ansichten jederzeit den aktuellen Stand besitzen. Die Erweiterbarkeit der Softwareanwendung mit weiteren Modulen setzt an der `Interaction` Klasse an. Von dieser Basisklasse leitet, wie in der Abbildung 42 dargestellt der `RootController` selbst ab. In dieser Klassenhierarchie erbt die vom `RootController` ableitende `AdminController` Klasse alle Steuerungsfunktionalitäten und stellt zudem Module zur Regelverwaltung und -bearbeitung durch die Anwender*innen zur Verfügung. Über eine bei der Installation der Software gesetzte Systemvariable wird beim Starten die Instanziierung der entsprechenden Kernklasse – `Root-` bzw. `AdminController` – gesteuert. Zur weiteren Ergänzung von Modulen (mit Benutzeroberflächen) würde gemäß dem umgesetzten Muster eine weitere Ableitung in dieser Kernklassenhierarchie z.B. `MyExtendedRoot-`

Controller zur zentralen Erweiterung des generellen Kontrollflusses benötigt. In dieser könnten dann auch weitere eigene MVC-Tupel, d.h. eine jeweils zuvor angelegte Ansicht in der WPF Applikation (UI) sowie zugeordnet ein Controller und ein View Modell initialisiert werden.

5 Umsetzung der Ebene des dynamischen Anwendungsansatzes mit Werkzeugen für modellbasierte Analyseprozesse

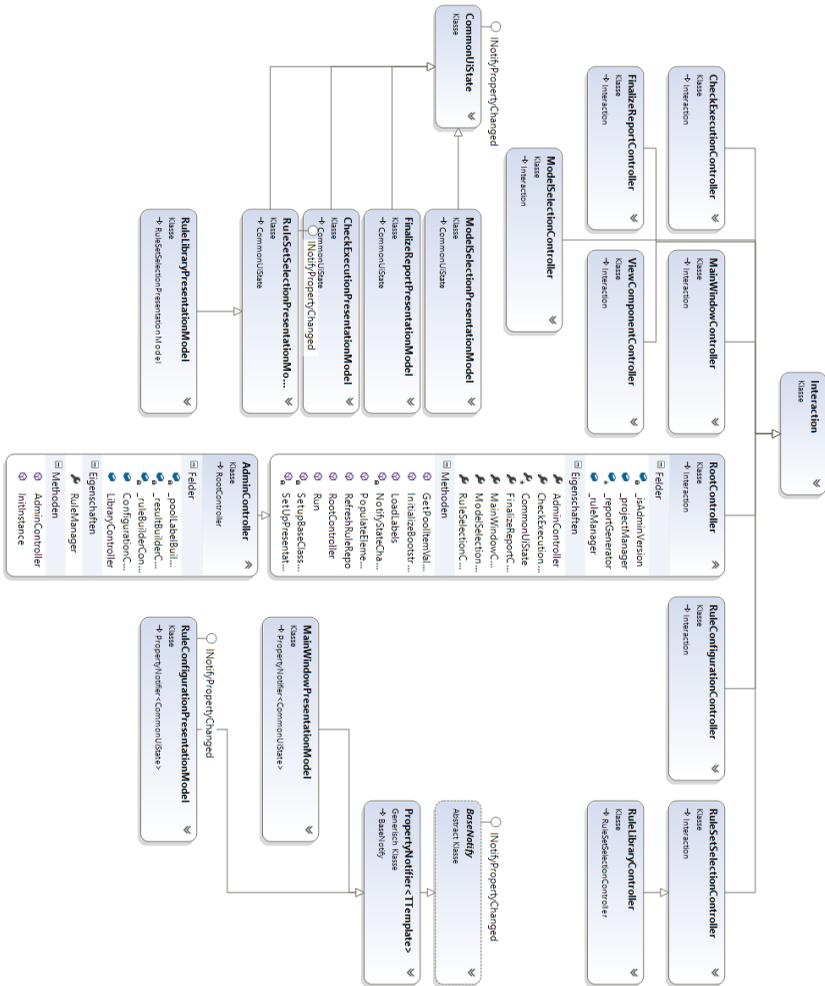


Abbildung 42 Workflow Komponente - Realisierung der Business Logik

In Abbildung 43 wird diese prinzipielle Erweiterungsmöglichkeit durch ein Befolgen des MVC-Entwurfsmusters anhand des Berichtsmoduls verdeutlicht.

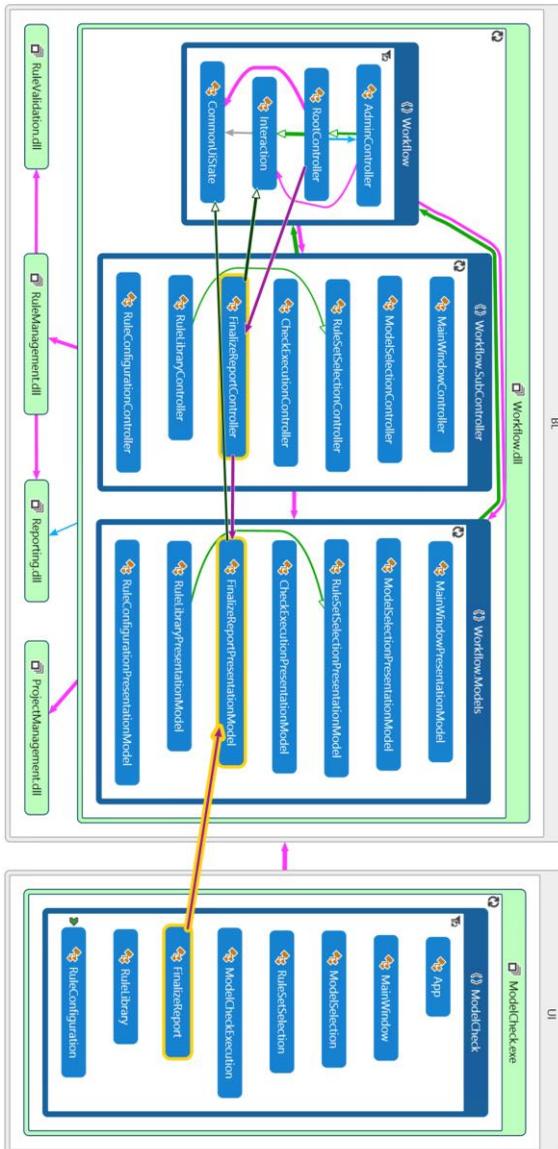


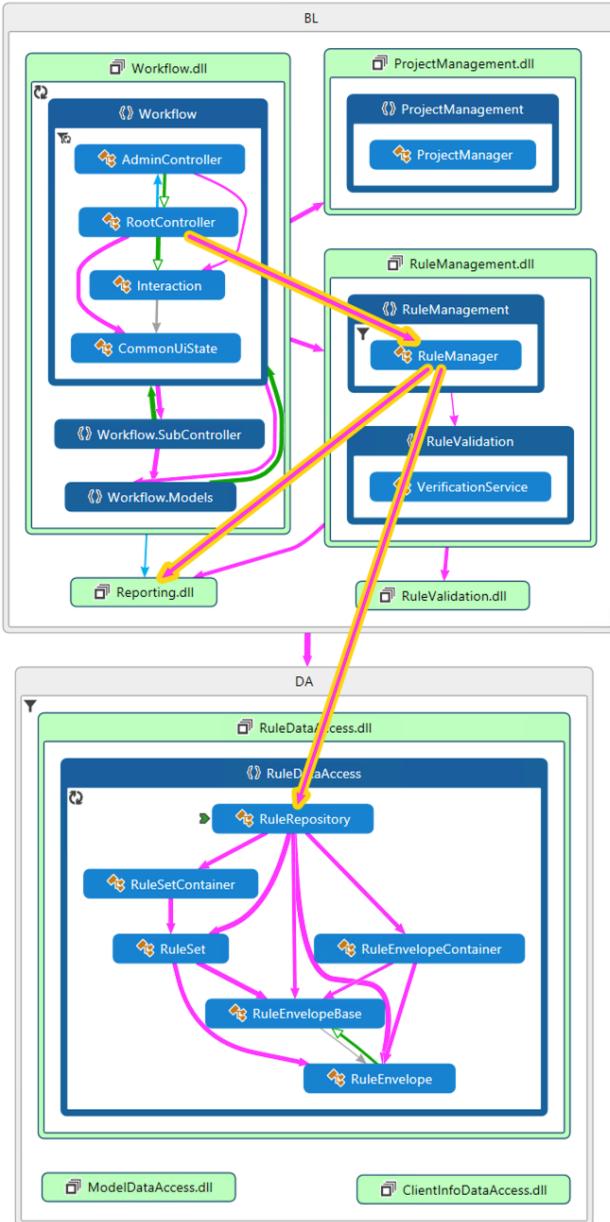
Abbildung 43 Realisiertes MVC-Muster am Beispiel vom Berichtsmodul

Im `RootController` wird eine Instanz der `ReportController` Klasse erstellt, um damit die darin gekapselten Funktionalitäten zur Berichtsverfassung zum einen an die Nutzerebene durch eine der Klasse zugeordnete Instanz einer `ReportPresentation` Klasse zu binden. Die Instanz dieses View-Modelles wird hierfür über einen generellen Bindungsmechanismus in einem Ressourcenlexikon der `App` Klasse in der UI Schicht registriert. Darüber wird sie dann später im Initialisierungsvorgang beim Instanzieren der entsprechenden Berichtsansicht an die vorgesehenen Oberflächenelemente angebunden (vgl. gelb hervorgehobenen Pfeil in Abbildung 43).

Zum anderen können durch die Basisklasse `Interaction` die Einzelzustände des Moduls, beispielsweise ob ein gerade bearbeiteter Bericht vollständig ist, sodass er exportiert werden kann, in der generellen Steuerungsmethode für den Kontrollfluss berücksichtigt werden. Insbesondere wird durch diese zentrale Steuerung immer ein eindeutiger, aktueller Zustand in der Laufzeit der Anwendung gewährleistet. Zudem werden von diesem durch das Synchronisieren der entsprechenden Basisklassen-Objekte für alle abhängigen Module ein zentraler Benachrichtigungsmechanismus bereitgestellt.

5.2.2.2 Integration der Modul Manager

In der Business Logic Schicht sind neben dem oben beschriebenen Umsetzen des MVC Musters zur Entkopplung der Benutzeroberfläche von den Geschäftsprozessen der Softwareanwendung im Kernmodul `Workflow` auch die Hauptarbeitsvorgänge eingebunden, um damit in der entkoppelten Datenzugangsebene die Handhabung der Geschäftsdaten steuern zu können. Gemäß einem intendierten Zweck, bspw. um die laufzeitbezogenen Kontextvariablen einer konkreten Anwendung der Software zu verwalten oder um ein generelles Hantieren von Regeldaten zu ermöglichen, wird hierfür die entsprechende Steuerungslogik in sogenannten Manager Klassen gekapselt. Zum Steuern der Geschäftsdaten werden diese im `RootController` instanziiert und der Gesamtsteuerung sowie den entsprechenden Nutzerinteraktionen in den Controller Klassen zur Verfügung gestellt. Sowohl für den Kernprozess der Softwareanwendung – Prüfen und Analysieren von Produktmodellen – als auch für deren Erweiterung um einen Regeldadministrations- bzw. Bearbeitungsmodus



zur dynamischen Skalierung des regelbasierten Prüf- und Analysefunktionalitäten stellen die Klassen **ProjectManager** (vgl. Kap. 5.1.2.2) sowie **RuleManager** (vgl. Kap. 5.1.2.4) benötigte Grundprozeduren bereit. Abbildung 44 stellt die durch die Manager Komponenten an die BL Schicht angebundene Datenzugangsebene beispielhaft an den Verbindungen der **RuleManager** Klasse dar.

Daten (DA – Data Access) Ebene – persistierbare Nutzerdaten für dynamische Erweiterungen

Im Gegensatz zu dem an das Domain Object Model (DOM) angelehnte Prinzip beim

Abbildung 44 Anbindung der Datenzugangsebene durch Manager Komponenten

Einlesen persistierter Produktmodelldaten mittels der .NET Komponente LINQ2XSD (vgl. 5.1.2.3), wurden die weiteren Datenzugänge der Softwareanwendung nach dem Data Binding Prinzip realisiert. Indem dabei das Repository Entwurfsmuster verfolgt wurde, in dessen Kern eine Entkopplung verschiedener Datenzugänge mittels Schnittstellen steht, konnte eine flexible Lösung umgesetzt werden, die für weitere Datenhaltungstechnologien (Datenbanken, Webservices usw.) erweitert werden kann. Als Syntaxbasis aller im Kontext der Softwareanwendung stehenden strukturierten Daten, wurde aufgrund ihrer ubiquitären Verbreitung und der daraus resultierenden Vielzahl an Basistechnologien für ihre Verarbeitung und Handhabung die XML Notation sowie die korrespondierende Schemasprache XSD gewählt. Verschiedene Technologien stehen zur Generierung von Klassenbibliotheken aus einem in XSD verfassten Schemadokument zur Verfügung. Im Rahmen der vorliegenden Arbeit wurde auf den freiverfügbaren Klassengenerator XSD2CODE (vgl. Kap. 2.3.2) zurückgegriffen, um für die einzeln spezifizierten XML Datenformate jeweils korrespondierende C# Klassen mit eingebetteten Bindungsmethoden zum Lesen und Schreiben entsprechender XML-Instanzdokumente zu erstellen.

5.2.2.3 Repositorien – mittels Data Binding entkoppelte Datenzugänge

Zur Verdeutlichung des angewendeten Repository Entwurfsmusters auf Basis des oben einleitend beschriebenen Data Binding Prinzips wird im Folgenden die technische Umsetzung anhand der wichtigsten Geschäftsdaten der Softwareanwendung – den Regeldaten – beispielhaft dargestellt. Der in Kapitel 5.2.2.2 erläuterten, durch die RuleManager Komponente realisierten Steuerungsebene der Regeldatenhandhabung wird mit der RuleRepository Klasse eine alle datenzugangsbetreffenden Vorgänge kapselnde Repository zur Verfügung gestellt. Im Wesentlichen sind das neben sogenannten CRUD Operationen zu den im Repository gehandelten Datenformaten jeweils Objektspeicher, in denen die objektifiziert, mittels des Data Binding eingelesenen Daten aus den physischen XML-Instanzdokumenten vorgehalten werden. Im Falle der RuleRepository Komponente werden in Abbildung 45 die beiden jeweils an ein XML-Dokument gebundenen Klassen RuleSet und

RuleEnvelope sowie die entsprechenden Objektspeicher (Container) Klassen dargestellt. Als verbindendes Element, da ein RuleEnvelope Objekt mehreren Regelsatz (RuleSet) Objekten zugeordnet werden kann, fungiert die in beiden XML Formaten gebundene Klasse RuleEnvelopeBase (vgl. Abbildung 46) mit ihrem einzigen Attribut zur Beschreibung eines eindeutigen Identifikators (GUID).

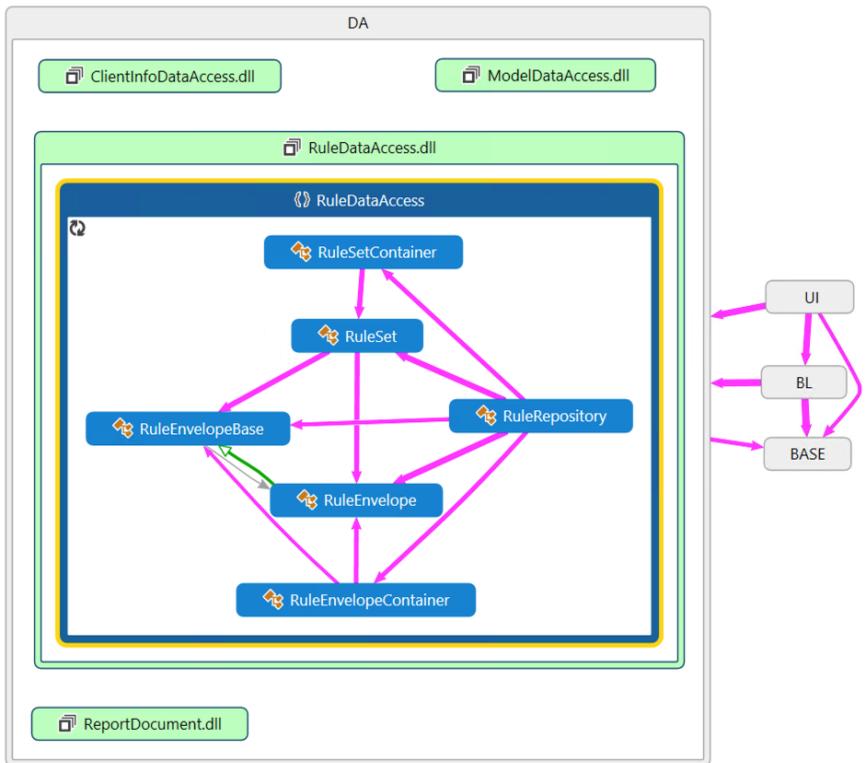


Abbildung 45 Wesentliche Bestandteile der RuleRepository Komponente

Während bei der ClientInfoDataAccess Komponente das Bearbeiten der Nutzerdaten schon im Standardmodus der Softwareanwendung ermöglicht wird, können die CRUD-Operationen zum Bearbeiten der Regeldaten nur mittels des Zuschaltens des Regeladministrations- und Bearbeitungsmodus frei-

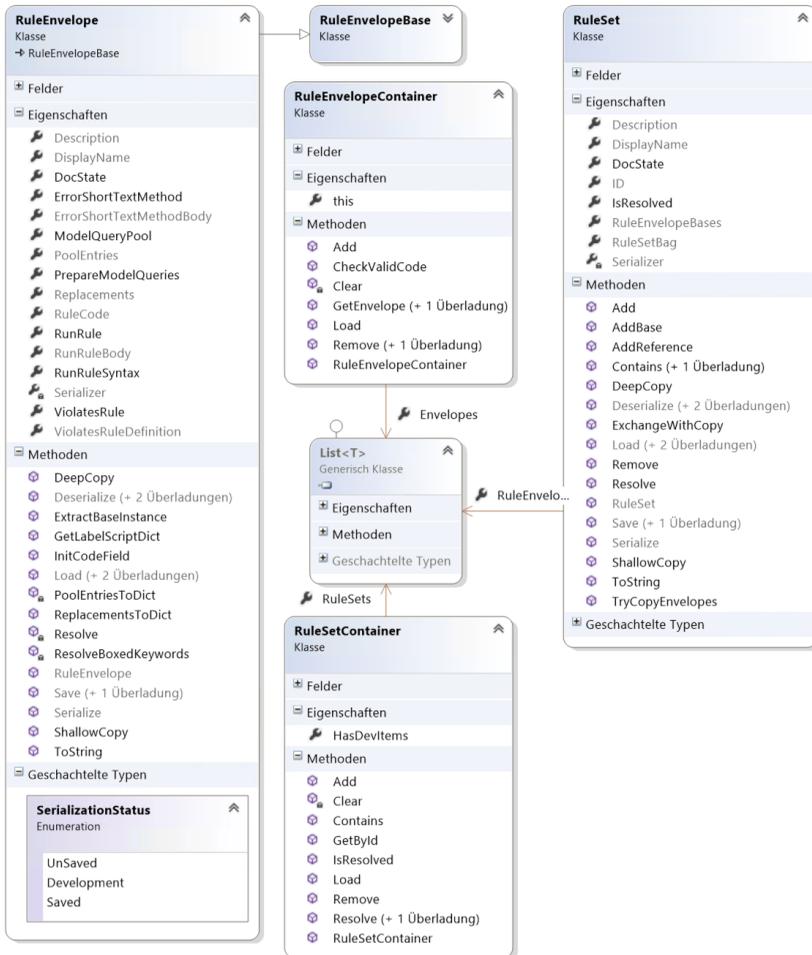


Abbildung 46 An XML-Repräsentation gebundene (grau) und zur Laufzeit erweiterte Eigenschaften und Methoden der RuleRepository Komponenten Klassen

geschaltet werden. Abbildung 46 zeigt die über die Klassen selbst bereitgestellten Bearbeitungsmethoden. Alle die Verwaltung von Regel(-satz)daten betreffende Methoden – validieren, hinzufügen, entfernen usw. – werden direkt durch die Objektspeicherklassen ermöglicht. Bei den an das XML Persistierungsformat gebundenen Klassen `RuleEnvelope` und `RuleSet` werden bei deren Instanziierung erweiterte, nicht in der persistenten Form vorhandene Methoden und Eigenschaften initialisiert. Damit werden die Inhaltsteile betreffende Basismethoden bereitgestellt, die von den Gesamtbearbeitungsmethoden der `RuleRepository` Klasse dann verwendet werden. So können beispielsweise die in einer binären, zerlegten XML Form der `RuleEnvelope` Klasse persistierten Teile eines Skriptausdruckes zur Bearbeitung zusammengesetzt und nach der Bearbeitung wieder in zerlegter Form in die XML-gebundenen Felder abgelegt werden.

Im Diagramm zur Veranschaulichung der Methoden der im Bearbeitungsmodus der Softwareanwendung erweiterten `RuleRepository` Klasse in Abbildung 47 sind die beiden zur Bearbeitung hinzugefügten Regel bzw. Regelsatz Kollektionsfelder hervorgehoben dargestellt. Neben den beiden Eigenschaften `DevRuleSets` und `DevRuleEnvelopes`, über die auf die jeweiligen Inhalte der Kollektionen zugegriffen werden kann, ermöglichen die beiden Eigenschaften `CurrentEnvelope` und `CurrentSet` die Bearbeitung einzelner Objekte. Auf der linken Seite des Diagramms sind hierzu eine Auswahl an Methoden dargestellt, die es beispielsweise erlauben, neue Regel resp. Regelsatzobjekte zu erstellen und den hervorgehoben dargestellten Kollektionen zuzufügen. Auf der rechten Seite exponieren verschiedene Eigenschaften zum einen für ein gerade im `CurrentRuleEnvelope` bearbeitetes Objekt dessen Inhaltsteile – logische Regelkomponenten (vgl. Kap. 3.2.1) zur Bearbeitung. Zu den an das persistente XML-Dokument gebundenen Datenelementen, die die Regelkomponente als `NamedValue` Typ repräsentiert, bilden diese Eigenschaften eine Fassade zum veränderbaren Wertparameter, der als Text vorgehalten wird. Mit dem Auslösen der `SaveCurrent` Methode (auf der linken Seite in Abbildung 47) werden dann die entsprechenden XML-gebundenen Datenelemente erstellt, bzw. deren Wert aktualisiert und das persistente Dokument entsprechend angepasst. Als Hilfestellung für die Anwender*innen ex-

poniert die Klasse zum anderen weitere Eigenschaften. Über diese können verschiedene bereits in den Regeldaten vorhandenen Informationen, wie beispielsweise die Menge aller Merkmal-Komponenten Werte, die in den vorhandenen Regeln vergebenen wurden, abgerufen werden, um damit die Eingaben der Anwender*innen zu unterstützen.

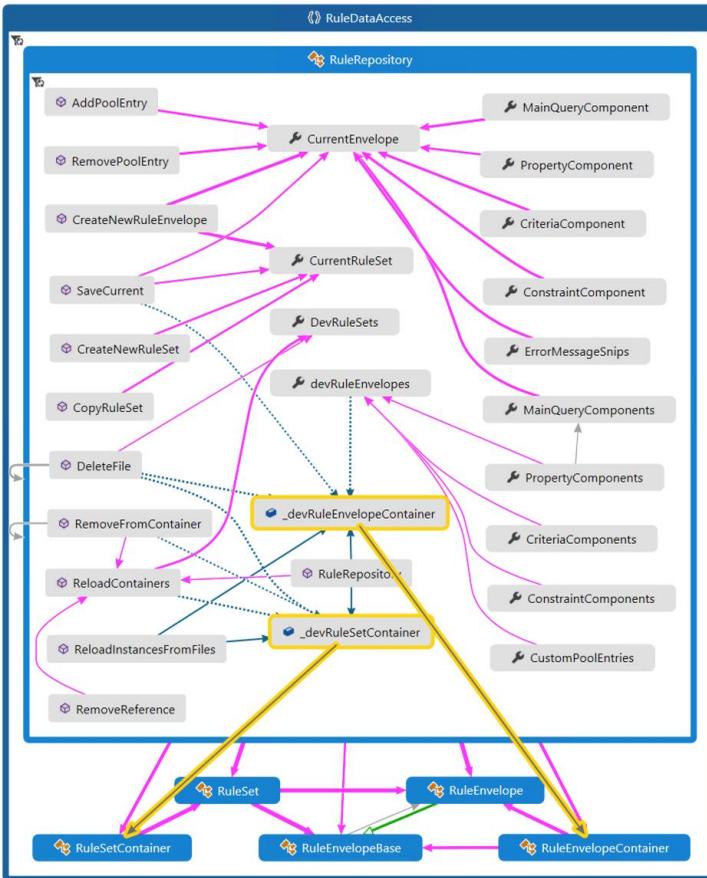


Abbildung 47 Zusätzliche Felder, Eigenschaften bzw. Methoden der um Bearbeitungsmöglichkeiten von Regeldaten erweiterten RuleRepository Klasse

5.2.2.4 Formate der Persistenzebene und Nutzerdaten

Die wichtigsten nutzer*innenspezifischen Daten der Softwareanwendung sind die Regeldaten, da mit ihnen der Funktionsumfang erweitert werden kann, und somit eine Anpassung auf einen spezifischen Anwendungskontext vollzogen wird. Die weiteren XML-basierten Datenstrukturen für die nutzerspezifischen Informationen, wie beispielsweise die in der `ClientInfoRepository` verwendeten XML-Dokumente spielen eine untergeordnete Rolle. Da es u.a. bereits entsprechende Standards zur Abbildung von Personen und/oder Organisationen sowie Projektstrukturen gibt, stellen sie nur einen Platzhalter der prototypischen Implementierung dar, der bei einer Professionalisierung der Software sicherlich durch entsprechende Standard-Implementierungen zu ersetzen wäre.

Dem Beispiel des vorherigen Kapitels 5.2.2.3 folgend werden daher anhand der Regeldatei `RuleEnvelope` exemplifizierte Inhalte zu den Nutzerdaten verdeutlicht und dann ihre Einordnung in die von der Softwareanwendung angelegte Strukturierung der nutzer*innenbezogenen Dateien auf dem System der Nutzer*innen dargestellt.

Die Spezifikation des Regelaufbaus im XSD Schema folgt der Konzeption der Zerlegung der Regel in Komponenten. Allerdings wurde für das Persistieren anstelle der konkreten vier konzeptionell vorgesehenen Regelkomponenten sowie einer Auswertungskomponente ein generisches (skalierbares) Beschreibungskonzept mit einem Lexikon aus Elementen vom Typ `NamedValue` umgesetzt. Damit lassen sich neben diesen bereits im Lösungsansatz der Arbeit definierten Regelteilen bedarfsorientiert eigene Semantik im Sinne von hinter eigenen Schlüsselworten stehenden beliebigen Termen bzw. Ausdrücken realisieren. Ein weiteres Beschreibungskonzept ermöglicht über diesen statischen Text, der über selbstgewählte Synonyme in die Regelentwicklung eingebunden werden kann, das Persistieren von Ausdrücken einer Skriptsprache bzw. Quelltextausdrücken hinter selbstvergebenen Schlüsselbegriffen (vgl. mittlere SCRIPT Ebene in Abbildung 48). Dieses dynamische Prinzip schließt dabei programmatisch über die `RuleEngine` Komponente (vgl. Kap. 5.2.2.2) an den Ansatz eines zentralen Objektlexikon an.

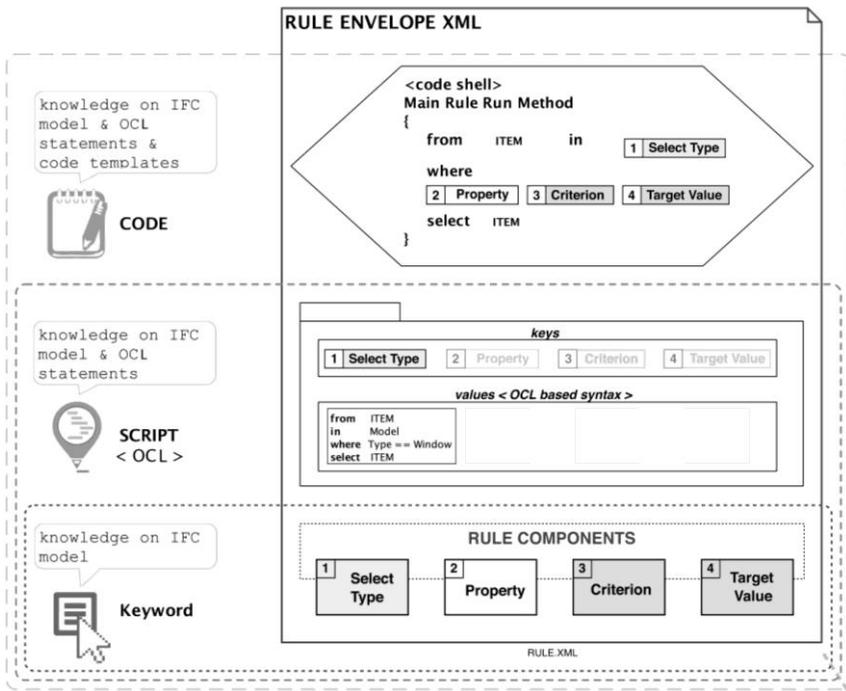


Abbildung 48 Schematisches Regeldateikonzept (Ebertshäuser & von Both, 2013)

In sogenannten Entry Elementen werden dabei im XML-Dokument entsprechende Schlüssel-Wert-Paare vorgehalten. Der im Wert befindliche Quelltext/-ausdruck wird dann beim Initialisieren als ausführbarer Code zusammengestellt, kompiliert und mit dem Schlüsselwort als Index in das Objektlexikon POOL eingetragen. Damit realisierbare dynamische Ausdrücke persistieren die im Lösungsansatz vorgesehene Ebene der Skriptsprache im Rahmen des Regel-XML-Dokumentes. Der rahmende Quelltext einer Regel wird in einem Element – `RuleBody` – als Lückentext abgelegt (vgl. oberste Ebene `CODE` in Abbildung 48). Erst durch das Aufrufen einer den Gesamtquelltext zusammenstellenden Methode beim Initialisieren der Regel wird der lauffähige

hige Quelltext aus diesem Lückentext gebildet. Dabei werden neben den standardmäßigen Schlüssel-Wert-Paaren der Regelkomponenten ggf. noch weitere statische bzw. dynamische Paare zum Füllen der Lücken einbezogen. Eine im standardmäßigen RuleBody Lückentext aufgerufene `ViolatesRule` Methode mit einem Booleschen Rückgabewert wird ebenfalls durch ein gleichnamiges Element im XML-Dokument als Lückentext persistiert. Neben diesen Beschreibungsmöglichkeiten sind im XML Schema noch Metadaten zur Regel wie ein Name, ein eindeutiger Identifikator, Beschreibungsfeld sowie eine Urheberrechtsangabe spezifiziert. Abbildung 49 zeigt an welcher Stelle in der Dateistruktur der Nutzerdaten die Regeldateien, die diesem Schema folgen abgelegt werden müssen, damit sie beim Initialisieren der Softwareanwendung berücksichtigt werden.

Neben den „produktiven“ Regeldateien (vgl., `RuleEnvelope.XML` im Hauptordner `RULES` in Abbildung 49) welche den Umfang der Regelbasis des Standardmodus der Software ausmachen,

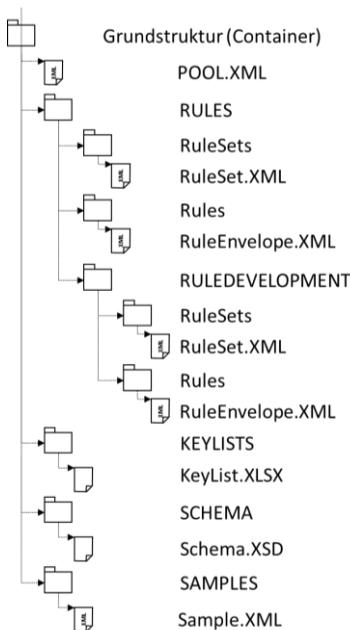


Abbildung 49 Strukturierung Nutzerdaten der Softwareanwendung

werden bei zugeschaltetem Administrations- und Bearbeitungsmodus auch die im Entwicklungsstatus befindlichen Regeln berücksichtigt, welche sich unter dem Hauptordner `RULEDEVELOPMENT` in der Dateistruktur befinden. Für das Anpassen des Kontexts der Softwareanwendung beispielsweise durch Ergänzen weiterer Hilfsfunktionen (vgl. Kap. 3.2.1), steht ein alle Grundfunktionalitäten beinhaltendes `POOL.XML` zur Verfügung, welches damit das Kernelement eines `REGELKONTEXTES`, d.h. eines mit Regel(sätze)n befüllten Containers (vgl. Abbildung 49 sowie Tabelle 12) als Regelbasis der Anwender*innen bildet. Diese wichtige Konfigurationsdatei beinhaltet dabei den persistierten durch

Schlüsselworte adressierbaren Quelltext, der als Vorlage beim Initiieren eines Modellladevorganges im zentralen Geschäftslogikspeicher für das Anlegen der initialen Einträge (z.B. Hilfsfunktionen sowie ihre Argumente) in das Objektlexikon POOL (vgl. Kap. 5.1.2.3 sowie Kap. 5.1.1) verwendet wird.

Generell sind im Rahmen der Softwareanwendung folgende Einheiten zur Strukturierung der Prüf- und Analyselogik als Mittel zur Organisation der Regelbasis vorgesehen:

Tabelle 12 Strukturierungseinheiten der Anwenderdaten

Strukturierungselement	Inhalt	Funktionen	Datentyp
CONTAINER	Ordnerstruktur	(ergänzender) Import/ (umfänglicher) Export	ZIP Datei
POOL	(intrinsische) Marken	Schlüssel/Wert Paare (entweder statische oder funktionale Wertausprägung)	XML Datei
REGELSATZ	Bezeichner, Beschreibung und Referenzen zu Regeln	Gruppierte Regeln werden seriell ausgeführt	XML Datei
REGEL	(spezifische) Marken und Regelkomponenten	Schlüssel/Wert Paare (entweder statische oder funktionale Wertausprägung) sowie feststehende Schlüsselbegriffe eines einheitlichen funktionalen Ausdrucks	XML Datei

Folgende Zusammenstellung listet zusammenfassend die in diesem Kapitel vorgestellten Nutzer*innendaten(strukturen) auf. Zu deren einzelnen Einheiten werden dabei jeweils der Inhalt und die Funktion aufgeführt.

- Strukturierungselement **CONTAINER** (ZIP Datei)

Inhalt (Ordnerstruktur der Nutzer*innendaten):

- Rules
 - (produktiv) RuleEnvelope
 - (produktiv) RuleSet
 - RuleDevelopment
 - RuleEnvelope
 - RuleSet
- (aktive) POOL.xml
- (optionale, anwender*innenspezifische) ClientInfo & Authors
- KeyLists
- Samples
- Schema
- (optional) LogFile.txt

Funktion: Die im Container vorgehaltenen Anwender*innendaten (Regelbasis und Kontextdaten) können (ergänzend) in das aktuelle Arbeitsverzeichnis zur Einrichtung der aktuellen Arbeitsumgebung importiert werden. Durch einen möglichen Export des aktuellen Arbeitsverzeichnisses in verschiedene Container, können – umfangliche oder teilweise – Sammlungen an Anwender*innendaten (Regelbasis und Kontextdaten) bedarfsgerecht zusammengestellt und zum Einrichten verschiedener Anwendungsumgebungen vorgehalten werden. Zu Archivierungszwecken einer durchgeführten Modellprüfung, kann zudem die aktuelle Arbeitsumgebung in einen entsprechenden Container exportiert und damit die vorgenommene Prüfung (replizierbar) abgelegt werden.

- Strukturierungselement **POOL** (XML Datei)

Inhalt: Schlüssel – Wert Tupel als Einträge eines Lexikons

- Wertetyp 1: statischer Wert (String, Integer usw.)
- Wertetyp 2: Funktionaler Ausdruck (Linq, C#)
- Wertetyp 3: Funktionaler Methodenausdruck (C#)

Funktion: In der Datei wird das Kernelement eines (funktionalen) Regelkontexts – standardmäßige (dynamische) Einträge für den zentralen (Laufzeit-) Objektspeicher POOL – persistiert; die als Quelltext persistierten Werte werden beim Initialisieren des POOL-Objektlexikons zur Laufzeit mit ihren Schlüsseln registriert und ergeben dann beim Aufruf der Schlüssel – kompiliert und ausgeführt – den eigentlichen (Laufzeit-bezogenen) Wert des Lexikoneintrags.

- Strukturierungselement **REGELSATZ** (XML Datei)

Inhalt: Referenzen (GUID) zu den beinhalteten Regeln, Name und Beschreibung

Funktion: Durch Referenzieren von Regeln, können diese in verschiedenen Prüf- und Analysekonstellationen jeweils in einem Regelsatz gruppiert vorgehalten werden.

- Strukturierungselement **REGEL** (XML Datei)

Inhalt:

- Key/Value (spezifische) Marken
 - Wertetyp 1: statischer Wert (String, Integer usw.)
 - Wertetyp 2: Funktionaler Ausdruck (Linq, C#)
- Regelkomponenten (standardmäßige feststehende Schlüssel der Wertetypen 1 und 2) welche in den standardmäßigen Lückentexten der RuleBody bzw. ViolatesRule Elemente eingebunden sind

- RuleBody Element – Lückentext der den rahmengebenden, zur Ausführung der Regel kompiliert und den ausgeführten Quelltext definiert.
- ViolatesRule Element – Lückentext der standardmäßig die Regelkomponenten in einem (Frage-)Ausdruck zusammengestellter Quelltext einer Methode, die im RuleBody Lückentext standardmäßig aufgerufen wird.

Funktion: Eine Regeldatei kapselt und persistiert eine Prüf- und Analyselogik zu einem definierten Sachverhalt, die auf ein bestimmtes Produktmodell ausgeführt werden kann. Zudem beinhaltet die Regel entsprechende natürlichsprachlich verfasste Meldungen zu Ereignissen, die möglicherweise zum Sachverhalt im Modell auftreten können.

Neben den bisher beschriebenen Nutzerdaten können im Rahmen der Regeln beliebige Referenzdaten eingelesen werden. Beispielsweise können in den im Ordner `KeyLists` (vgl. Abbildung 49) befindlichen Tabellendokumente zum Abgleich mit gewissen Sachverhalten im Modell entsprechende Referenzwerte stehen. Darüber hinaus können durch die im Ordner `Schema` abgelegten Produktmodellschemen Regeln mit Bezug zur Schemabeschreibung, die beispielsweise eine Hierarchie einer Basisklasse auslesen, realisiert werden. Für den Verifizierungsvorgang einer Regel, d.h. wenn die Anwender*innen eine selbstentwickelte Regel in die „produktive“ Regelbasis überführen, stehen im Ordner `Samples` Referenzmodelle bereit, mit denen die Regel verifiziert werden kann.

6 Umsetzung der Ebene der Methoden und Hilfsmittel für die Analyse- bzw. Prüflösgik

Im Folgenden werden gemäß den Anforderungen an die modellbasierten Planungsgegenstände des in Kapitel 4 konkretisierten Anwendungsszenarios die Konzeption und Umsetzung benötigter Prüf- und Analyselogik sowie deren bedarfsgerechte Strukturierung entlang des szenariobasierten Planungsprozesses beschrieben. Dabei sind mit den in der Softwareanwendung durch das Verwaltungswerkzeug vorgegebenen Strukturierungsobjekte der Regeln eine geeignet bezeichnete Grundstruktur (vgl. Kap. 5.2.2.4) der Regelbasis anzusetzen. Darin bedarfsgerecht für jede Anforderung eines umzusetzenden Prüfplanes (vgl. Kap. 4.5) erstellte, sowie gemäß den aufeinander aufbauenden Fragestellungen sortierte, geeignet bezeichnete Regelsätze bilden sodann das Grundgerüst der eigenen Regelbasis. Schließlich werden zur optimierten Wiederverwendung abgestimmte Bezeichner für die einzelnen Regeleinheiten gemäß der logischen Abfolge – vom Sicherstellen der vorhandenen Grundlagen zur eigentlichen Fragestellung an das Produktmodell – entwickelt und in die Regelbasisstruktur eingeordnet.

Alle Bedarfe zur Durchführung des Prüfplans sind bereits durch den Abgleich der organisatorischen Anforderungen mit der inhaltlichen Struktur und der in den Inhaltsbezeichnungen entwickelten strukturellen Basis als Regelbibliothek erfüllt. Darauf aufbauend können dann konkrete Konzepte und Methoden der jeweils in den Regeleinheiten gekapselten Anforderungen an Prüf-/Analyselogik entwickelt werden. Dabei bildet ein zu bedienender Workflow – entwickeln und durchführen eines geeigneten Prüf- bzw. Analyseplans – den (thematischen) Ausgangspunkt für die konkreten Anforderungen der Entscheidungsunterstützung planender Architekt*innen. Prinzipiell folgt der Anwendungsfall der Zielstellung einer Optimierung des Planungsgegenstands mit der Aufbereitung energetischer Aspekte zur entworfenen Planungslösgung. Durch das Aufgreifen von bestimmten fachlichen Sachverhalten der Energieplanung können die exemplifizierten Entscheidungshilfen dabei gleichfalls

auch als Grundlage zur Betrachtung von Auswirkungen des Entwurfes auf Seite der Ökobilanz dienen. Für den Anwendungsfall wird zunächst eine an den organisatorischen Bedarfen eines Architekturbüros orientierte übergeordnete Regelverwaltung aus verschiedenen Regelbasen konzipiert und umgesetzt. Diese Regelverwaltung ist auch in Bezug auf die konzeptionellen Festlegungen im Kontext des Anwendungsfalls (vgl. Kap. 3.2.3) gegliedert, und zwar nach den in dessen Rahmen enthaltenen, szenariobasiert aufgegriffenen Vorgängen, wie z.B. einer der Entscheidungsunterstützung vorgeschalteten Qualitätsprüfung des virtuellen Gebäudemodells. Die aufeinander aufbauende Struktur der Regelverwaltung folgt dabei dem bereits durch das Lastenheft (vgl. Kap. 4.5) vorgegebene (sowie das ggf. die logisch festgelegte Abfolge der Regelausführung betreffende) Durchführungskonzept und enthält dazu jeweils geeignete (sinngemäß bezeichnete sowie inhaltlich spezifizierte) Regelkonzepte. Vor der Umsetzung der Regelkonzepte in technologiebasierte Algorithmen werden wiederkehrende Prüf- und Analysevorgänge (z.B. bezüglich des Produktmodellschemas ein Auslesen eines Eigenschaftsobjekts, das von einem Modellelement referenziert wird) sondiert und in wiederverwendbare Hilfsfunktionen (vgl. Kap. 3.2.1) gekapselt. Als Kollektionen in einer Regelbasis zusammengestellt können somit verschiedene Regelkontexte (vgl. Kap. 5.2.2.4) gebildet werden. Übergeordnet spezialisieren sie selektiv z.B. für einen bestimmten virtuellen Modellstandard bzw. erweitern sie generell den regelbasierten Funktionsrahmen der Anwendung. Auf den Umsetzungskonzeptionen mit entsprechend dem Anwendungsfall strukturierten Regelkonzeptionen sowie diesbezüglich realisierten jeweils eine konkrete Regelbasis kapselnde Regelkontexte aufbauend, wird dann die Entwicklung der konzipierten Regeln sowie ihre Zusammenstellung in Regelsätze inhaltlich beschrieben. Abschließend werden für die formalen Realisierungen von Regelkontexten, Regelsätzen und Regeln die entsprechenden inhaltlichen Konzepte erläutert. Als logische Grundlage der beispielhaft im folgenden Kapitel 6.2.2.2 verdeutlichten Werkzeuge der Software zur Erstellung maschinenverarbeitbaren XML-Dokumente (vgl. Tabelle 12) werden in diesem Kapitel die Anforderungen des Lastenhefts (vgl. Kap. 4.5) in ein korrespondierendes Pflichtenheft überführt. Dabei werden jeweils die (logischen) Regelinhalte entwickelt, die dann anhand der entsprechenden Bearbeitungswerkzeugen der Softwareanwendung in ausführbare Regeln formalisiert werden können. Anhand einer

zudem entwickelten, aufeinander aufbauende (logischen) Strukturierung der Regelinhaltsteile zur baukastenförmigen Wiederverwendung, können die Regeldateien durch Administrationswerkzeuge der Software mit Regelsatzdateien gruppiert und in selbstverwaltet strukturierte Regelbasen eingeordnet werden.

6.1 Umsetzungsstrategie zur gestuften Anpassung der Softwareanwendung an den Anwendungsfall

Mit der Entwicklung von Regeln zur Prüfung bzw. Analyse der Planungsmodelle („Geschäftslogik“) wird die im Kapitel 5 beschriebene Softwareanwendung auf einen bestimmten Anwendungskontext angepasst. Das wesentliche Prinzip dabei ist eine Kapselung der technischen sowie fachlichen Prüf- und Analyselogik in entsprechend für einen Anwendungsfall aufbereitete und bedarfsorientiert bezeichnete Regeleinheiten. Damit kann eine die Planung koordinierende Entscheider*innenebene prinzipiell ohne inhaltliche Kenntnis der zugrundeliegenden Mechanismen, durch selbständig rechnergestützt erstellte Entscheidungsgrundlagen unterstützt werden. Diese Endanwender*innenebene zum Ausführen bedarfsgerecht vorbereiteter Regeln auf die Planungsmodelle unterscheidet die Softwareanwendung dabei von den mit vorausgesetzter Fachkenntnis zu bedienenden Expertenwerkzeugen. Anstelle in einer Entscheidungssituation weitere Fachexpertise einbeziehen und entsprechende Entscheidungsgrundlagen ad hoc organisieren zu müssen, werden diesbezügliche Anforderungen a priori von den Endanwender*innen formuliert (vgl. Kap. 4.5). Die weiteren Rollen mit entsprechender Fachexpertise setzen diese Anforderungen dann mit den Bearbeitungswerkzeugen der Softwareanwendung zur Regelentwicklung in Regeln um (vgl. Kap. 7.1), die gemäß den formulierten Bedarfen bezeichnet sind. Im zuschaltbaren Bearbeitungsmodus wurde ebenfalls versucht, durch einen gemäß dem Regelaufbau (vgl. Kap. 3.2.1) gestuften Ansatz der Bearbeitungsfunktionen das Anpassen der Regeln nichtversierten Anwender*innen zu ermöglichen. Die dabei erweiterte Soft-

ware verdeckt zwar durch das Prinzip zur eigenen Vergabe domänenspezifischer Semantik in den Regeln die Komplexität der darunterliegenden Skript- bzw. Programmiersprache, jedoch tendiert sie mit Merkmalen einer Entwicklungsumgebung mehr zum Expertenwerkzeug, welches mindestens profunde Kenntnisse zum Regel- sowie Produktmodellenaufbau voraussetzt (siehe auch Tabelle 5, vgl. Abbildung 48).

Damit das volle Potential der losen Kopplung dieser beiden Anwendungsebenen der Softwareanwendung – Endanwendung und Regelentwicklung – mittels der gestuften Regelkonzeption ausgeschöpft werden kann, bedarf es einer strategischen Organisation der Regelentwicklungen. Dabei gliedert sich idealtypischerweise die Verteilung des Regelentwicklungsaufwandes in den gestuften Anwendungskontexten vom eigentlichen Anwendungsfall in einer spezifischen Entscheidungssituation hin zum sie beinhaltenden Planungsprozess und schließlich weiter zum ausführenden Planungsbüro selbst. Die darauf abgestimmte Zielstellung (vgl. Kap. 4.4) sieht mit einem aufeinander aufbauenden Ordnungsprinzip – in dem jeweils der untergeordnete Kontext die im Übergeordneten bereits (generalisiert) vorhandenen Regeln allein durch geringfügige bedarfsgerechte Anpassung verwenden kann – für jede Stufe eine eigene Regelbasis vor. Eine dritte Anwendungsebene zur Verwaltung der Regeln mit übergeordneten Strukturierungseinheiten ermöglicht in dem der Software zuschaltbaren Bearbeitungs- und Administrationsmodus dazu den (einen ganzheitlichen arbeitsmethodischen Einsatz der Software organisierenden) Anwender*innen, eine geeignete Umsetzungsstrategie zu verfolgen. Dabei geht es um die optimale Aufteilung und Strukturierung der einzelnen Regeleinheiten zur effizienten Wiederverwendung bereits entwickelter Regellogik im gesamtheitlichen (über die drei exemplifizierten Stufen gegliederten) Entwicklungsprozess. Hierfür umspannt die Umsetzungsstrategie die folgenden drei Strukturierungsstufen, auf denen die Regeln (aufeinander aufbauend – „vom Generellen zum Spezifischen“) entwickelt werden:

- Regelbasis Bürostandard – projektübergreifender Grundstock mit Regelschablonen und -methoden

- Regelbasis Planungsprozess – ein projektspezifisch organisierter Regelstock mit (erweiterten) Regelschablonen bzw. deren bedarfsgerechte ausgeprägter Regeln
- Regelbasis Planungsentscheidung – situationsspezifischer Regelstock mit den für die Endanwendung (final anzupassenden) Regeln.

Indem in der letztgenannten Stufe neben der Auswahl geeigneter Regeln aus dem Grundstock bzw. der Projektregelbasis idealerweise allein das Ausprägen der in den vorherigen Stufen selbstvergebenen Regelsemantik verbleibt, ermöglicht die Umsetzungsstrategie im Regelentwicklungsprozess den Aufwand bzw. die benötigte Expertise, bei diesem finalen Anpassen auf den Anwendungsfall zu minimieren. Darüber hinaus sieht die Umsetzungsstrategie auf den einzelnen Stufen konventionsgetriebene Regelentwicklungszyklen vor, wobei die Regeln jeweils pro Stufe in einer Entwicklungsbibliothek konzipiert und umgesetzt werden und dann verifiziert in einer Produktivbibliothek für die Weiterentwicklung auf der nächsten Stufe bzw. der Endanwendung als Regelkontext (vgl. Kap. 5.2.2.4) bereitgestellt werden. Ein dadurch einheitlich im gestuften Entwicklungsprozess ermöglichtes Freigabeverfahren für den produktiven Einsatz der Regeln bildet dabei auch eine Grundlage zum strukturierten Versionieren und Archivieren des insgesamten Regelbestandes.

Zur Einordnung eines sinnmäßigen Umgangs mit Werkzeugen der Softwareanwendung entsprechend der generell erläuterten Umsetzungsstrategie werden im Weiteren anhand der drei in Abbildung 50 dargestellten Stufen arbeitsmethodische Aspekte einer gestuften Organisation der Regelentwicklung verdeutlicht. Bezugnehmend zum (szenariobasiert) gestuften Anwendungskontext werden dabei die Funktionsbereiche der Softwareanwendung – Grundlagenentwicklung/Verwalten, Erstellen/Anpassen sowie Konfigurieren/(End-) Anwendung – methodisch im Rahmen der entsprechenden Stufe verdeutlicht.

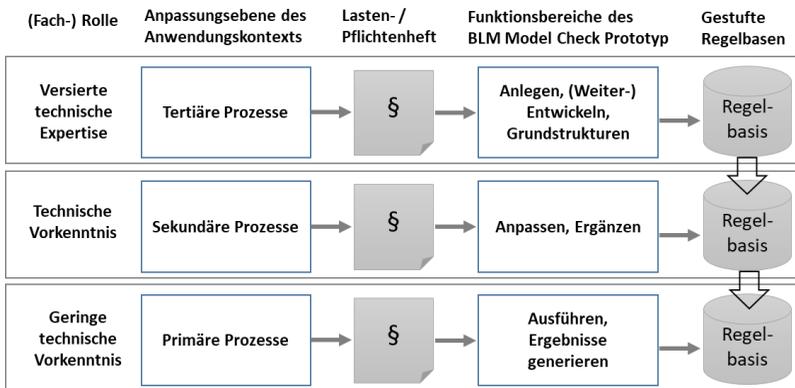


Abbildung 50 Skizze der Umsetzungsstrategie einer gestuften Regelentwicklung

Im Kapitel 4 werden exemplifizierende Prüf- bzw. Analysepläne (szenario-basiert) beschrieben. Sie bilden die thematische Grundlage für die Entwicklungsstufen und die beispielhaften Inhalte entsprechender Regelbasen – Grundstock, Projektregeln, Situationsregeln. Diese neutral formulierten Anforderungen im Rahmen der Zusammenstellung eines Lastenhefts (vgl. natürlichsprachliche Prüf- und Analysefragestellung mit Bezug zu Entitäten des Produktmodells in Kap. 4.5) wurden dabei in eine rahmengebende Struktur eingeordnet, die durchgängig die einzelnen Entwicklungsstufen verbindet, und gemäß ihrem Zweck im Prüf- bzw. Analyseplan als designierte Regeleinheit bezeichnet. Somit ist der entwicklungsstrategische Zusammenhang zwischen den in der einen Stufe beispielsweise als Grundregel zu entwickelnder Regelinhalt mit der in der nächsten Stufe dann anzupassenden „Anwendungsregel“ gegeben. Diese die Stufen übergreifenden Zusammenhänge werden im folgenden Kapitel 6.2 durch entsprechende aufeinander aufbauende Methoden und Konzepte einzelner Modellprüfungen und -analysen beim Ausformulieren der (logischen) Regelinhalt im Pflichtenheft berücksichtigt. Diese Struktur wird ausgehend von der grundlegenden Stufe der Entwicklung und Fortschreibung eines Regelgrundstocks im Weiteren als Grundlage der ganzheitlichen Umsetzungsstrategie bis zur Stufe der Regelvorbereitung zur Endanwendung spezifiziert. Für jede Stufe werden dabei die von der Softwareanwendung zur Verwaltung der Regelbasis vorgesehenen Strukturierungsobjekte (vgl. Kap. 5.2.2.4) gemäß

dem Lastenheft anhand entsprechender Container, Regelsätzen und Regeln exemplifiziert.

6.1.1 Entwicklungsstufe des Regel-Grundstocks

Grundlegend für die projektübergreifende Entwicklung der Regelbasis innerhalb eines (Planungs-) Unternehmens setzt auf dieser Stufe die (Weiter-) Entwicklung geeigneter Namenskonventionen der Regelbezeichnungen sowie deren sinngemäße Kategorisierung/Gliederung an. Eine die Gesamtentwicklung organisierende Rolle benötigt zunächst allein die Kenntnis der Prozesse im Unternehmen, um diese grundlegende Ordnerstruktur (Grundstock) für Regeln anzusetzen. Abbildung 51 zeigt die konkrete Ordnerstruktur eines exemplifizierten Containers „Bürostandard (IFC)“ für diese Stufe. Darin setzt beispielhaft die Entwicklung einer Hilfsfunktion (vgl. Kap. 3.2.2 und Kap. 5.2.2.4) für das Produktmodell IFC zur (wiederkehrenden) Behandlung von Eigenschaften an, die an ein Bauteilobjekt verknüpft abgebildet werden. Mit dem im Folgenden damit (methodisch) dargestellten beispielhaften Entwicklungszyklus können prinzipiell grundlegende Regelschablonen zur Modellprüfung und -analyse eines Unternehmens entwickelt und ein Grundstock an Regeln als Regelbasis (Container) skaliert bzw. verwaltet werden.

Vor einer Bereitstellung im Regelbibliothekswerkzeug der Software von dieser Art Schablonen für den produktiven Einsatz (d.h. ihrer Verwendung in Planungsprojekten nachdem sie gemäß den projektspezifischen Anforderungen problemspezifisch ausgeprägt wurden) findet ihre iterative Entwicklung und Erprobung im Rahmen des Werkzeugs der Regelentwicklungsbibliothek statt. Zwei mögliche Pfade führen hiervon in den produktiven Einsatz. Zum einen werden die als Vorlage zur projektspezifischen Ausprägung verwendeten Regelschablonen mittels eines automatisierten Verifikationsprozesses in das Regelbibliothekswerkzeug überführt. Weisen die beinhalteten Modellprüfungs- bzw. -analysevorgänge einen hohen Grad an Wiederverwendbarkeit auf, kann zum anderen der funktionale Inhalt der Schablone auch als Hilfsfunktion in den zentralen persistierbaren Geschäftslogikspeicher (vgl. POOL.XML in Kap. 5.2.2.4) überführt werden und über die dadurch direkt erweiterte Geschäftslogik der Softwareanwendung aufgerufen werden. Bei der Letzteren

verbleibt allein der Aufruf dieser Hilfsfunktion in der Schablone, die dann wie die Erstere für eine darauf aufbauende bedarfsorientierte Anpassung (Ausprägen der Werte gemäß einem intendierten konkreten Prüfzweck) in das produktive Regelbibliothekswerkzeug als Vorlage zum Aufruf der Hilfsfunktion überführt wird. Da sich die in diesem Kapitel beschriebene Umsetzung der Regelbasis auf das im Anwendungsszenario verwendete Produktmodell IFC bezieht, stehen auch die exemplifizierten Regelentwicklungen im Rahmen dieses Modellformats. Eine grundlegende Gliederung zur Bedienung weiterer Modellstandards besteht daher prinzipiell darin, den umgesetzten Aufbau der IFC-basierten Regelbasis äquivalent für diese zu entwickeln und in eigenständigen Containern, bspw. einer Regelbasis „CityGML-Bürostandard“ zu verwalten. Für standardübergreifende Regelansätze z.B. um Gebäudemodelle im Rahmen von Stadtmodellen zu analysieren können darüber hinaus mehrskalige, übergeordnete Regelbasen auf diesen Standardspezifischen aufbauen.

Um den Entwicklungszyklus auf dieser Stufe zu veranschaulichen werden im folgenden Kapitel für die in Abbildung 51 exemplifizierten Regeleinheiten (logische) Inhalte einer Hilfsfunktion, sowie einer entsprechenden Regelschablone zum Iterieren aller IFC Eigenschaftsobjekte die einem IFC (Bau-)Element zugeordnet sind, entwickelt. Die als Container persistierte Regelbasis „Bürostandard (IFC)“ hat dazu folgenden Aufbau und selbstvergebene Namen bzw. Schlüsselbegriffe:

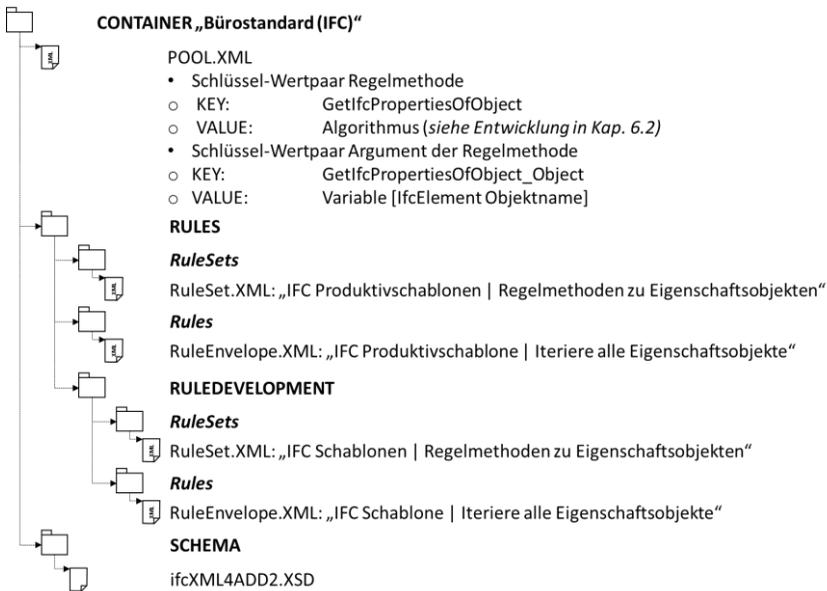


Abbildung 51 Regel Container "Bürostandard (IFC)"

6.1.2 Entwicklungsstufe der Projektregeln

Für die im Rahmen eines Planungsprojektes benötigten Modellprüfungs- bzw. Analyseregeln wird jeweils eine Projektregelbasis in Form eines projektspezifischen Containers aufgebaut. Nach Beendigung des Projektes, können die darin neu hinzugekommen bzw. weiterentwickelten Regeln wieder in den (im vorherigen Kapitel als Regelbasis *Bürostandard* exemplifizierten) Grundstock zurückgeführt, bzw. mit dessen Regeln harmonisiert werden. Abbildung 52 zeigt die exemplifizierte Struktur der Regelbasis *Planungsprojekt*, die dann als baukastenartige Vorlage für die in Kapitel 6.1.3 dargestellte Struktur der darauf aufbauenden Regelbasis zu einer spezifischen Entscheidungssituation im Projektzenario dient.

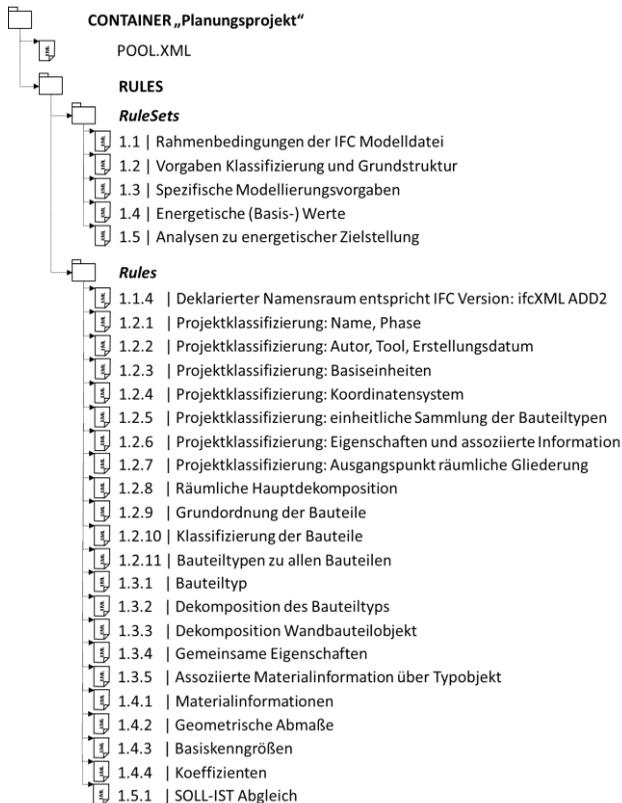


Abbildung 52 Container "Planungsprojekt": Gliederung und Regelbezeichnungen

Gemäß den spezifizierten Anforderungen (vgl. Tabelle 11) werden dabei Regel/-sätze dargestellt, die u.a. beispielsweise wie die „Gemeinsamen Eigenschaften“ auf Vorlageregeln in der Regelbasis Bürostandard aufbaut und diese auf konkrete Eigenschaften von Wandbauteilen anpasst. Als konfigurierbare Vorlage, idealerweise allein durch Anpassen einzelner Werte der beinhalteten Regeln, können diese Regelsätze für spezifische planerische Entscheidungssituationen im Planungsprozess ausgeprägt werden.

6.1.3 Entwicklungsstufe der Situationsregeln

Die für den Rahmen des szenariobasierten Planungsprojektes (vgl. Kap. 4) im vorherigen Abschnitt spezifizierte Struktur, sowie den darin eingeordneten Regeleinheiten, stellen die Grundbausteine, auf denen eine die konkrete Planungssituation (vgl. Kap. 4.2) unterstützende Regelbasis aufbaut. Die Anforderungen dieser planerischen Entscheidungssituation in der Entwurfsplanung bilden dabei den Rahmen einer finalen Anpassung bisher generalisierter Regeln der Regelbasis Planungsprojekt. Die designierte Endanwendung dabei als Vorlage genutzter Regeln ist eine Modellprüfung und -analyse des ausgearbeiteten Planungsgegenstands am Ende der Entwurfsphase. Sie müssen für den spezifischen Fall einer Endanwendung angepasst werden, bei der sie der Erstellung von Entscheidungsgrundlagen zu zwei gegenübergestellten Planungsalternativen gemäß energetischer Zielstellungen dienen. Beim Spezifizieren konkreter Anforderungen des thematisch unterlegten Erstellens von Kontroll- und Analyseplänen wurden entsprechende Zusammenhänge (Planungsphase – Planungssituation) bereits im Lastenheft herausgearbeitet (vgl. Tabelle 11). Die im Folgenden spezifizierte Struktur, sowie der darin angelegten Regelbezeichnungen (vgl.

Abbildung 53) nimmt darauf Bezug und spezifiziert damit den Umfang der im Kapitel 6.2 umgesetzten Entwicklung von Prüf- und Analyselogik für die beiden Planungsalternativen des Anwendungsfalls. Strategisch stellt der hierfür exemplifizierend angesetzte Container „Planungssituation“ dabei bezüglich einer idealtypischen Arbeitsmethodik einen im Verlauf der Planung (zu bestimmten Zeitpunkten) situationsspezifisch für die Planungslösung aufbereiteten (bzw. ausgeprägten) Stand der Regelbasis Planungsprojekt dar. Wird von Beginn der Planungsphase, konsequent für alle relevanten Entscheidungssituationen, ein entsprechender Stand der Regelbasis fortgeschrieben, so ermöglichen diese einzelnen (reproduzierbaren) Zwischenstände eine retrospektive Nachvollziehbarkeit der getroffenen Entscheidungen, wenn hierzu parallel im (versionierten) virtuellen Modell die entsprechenden Stände des stetig konkretisierten Planungsgegenstands planungsbegleitend vorgehalten werden.

6.1 Umsetzungsstrategie zur gestuften Anpassung der Softwareanwendung an den Anwendungsfall



Abbildung 53 Container "Planungssituation" (Struktur des Pflichtenhefts)

6.2 Konzipieren und Umsetzen des exemplifizierten Pflichtenhefts für Prüf- und Analysepläne

Im vorherigen Kapitel wurde mit einer Umsetzungsstrategie die Ebene der Struktur und Bezeichner für die Regelentwicklungen zu den aus dem Anwendungsszenario in Kapitel 4 abgeleiteten Anforderungen spezifiziert. Im Folgenden werden gemäß dem „inneren“ Konzept der Regeln (vgl. 3.2.1) die Inhalte konzipiert und umgesetzt. Die Anforderungen beschreiben hierzu die Fragestellungen einer konkreten Modellprüfung bzw. -analyse (Lastenheft). Zur Entwicklung eines auf die Strukturen des Modellstandards IFC anwendbaren Modellprüfungsvorgehens (d.h. Prüfplan als natürlichsprachliche Grundlage der in den Regeln zu formalisierenden maschinen-verwertbaren Logik) müssen zunächst die den Fragestellungen zugrundeliegende IFC Entitäten sondiert werden. Des Weiteren muss geklärt werden, wie die sondierten IFC Objekte zur Beantwortung der Prüffragestellung ausgeprägt sein können (Spezifikation des Pflichtenhefts zum Prüfplan als logische Grundlage einer Regeleinheit). Hierauf aufbauend kann mit dem „in eine logische Abfolge bringen“ der eigentliche Regelinhalt konzipiert werden. Beim Notieren der Prüf- und Analyselogik werden bezüglich der querschnittenden Umsetzungsstrategie, durch Vergabe entsprechender Bezeichner auf den einzelnen Ebenen des Regelaufbaus (Code/Abfragesprache, Regelkomponente bzw. selbstvergebene Schlüsselbegriffe, vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**) die Zusammenhänge der einzelnen Regeleinheiten untereinander ausformuliert. Die somit in diesem Kapitel entwickelten Regelkonzepte können dann mit den der Softwareanwendung inbegriffenen Werkzeugen der Regelentwicklung (vgl. Kap. 5.1.2.4) in ihre formale Gestalt überführt und als XML-Datei persistiert werden (vgl. Kap. 5.2.2.4 sowie Kap. 7.1). Der Zwischenschritt – Spezifizieren eines Pflichtenhefts sowie Notieren der Regelkonzeption – ist dabei im Rahmen dieser Arbeit gewählt worden, um die auf Seiten der Nutzer*innen benötigte „logische Vorarbeit“ der Regelentwicklung beim Arbeiten mit der Softwareanwendung zu veranschaulichen. Bei einem produktiven Softwareeinsatz könnte dies bei entsprechend erfahrenen Anwender*innen entfal-

len, da beispielsweise die konzeptionelle Abfrage-logik bezüglich der Modellstrukturen direkt mit der Abfragesprache in einer Regel formalisiert werden kann. Zur besseren Verständlichkeit der im Folgenden dargelegten Entwicklung der Regelkonzepte sind diese einheitlich jeweils mit einem tabellarischen Pflichtenheftteil dargestellt. Zudem werden bei ausgewählten bedeutenden Anforderungen mit einem in BPMN notierten Logikteil (der den Abfragealgorithmus repräsentiert) die spezifische Abfolge in der Prüflogik exemplifiziert.

Mit der Beschreibung dieser logischen Prüf- bzw. Analyseabfolge werden dann neben den dazu spezifizierten (Schlüssel-) Begriffen und deren Zusammenhang mit dem Konzept eines Regelaufbaus eine entsprechende Auswertungslogik dargestellt. Da bezüglich des objektorientierten Produktmodells IFC viele der im Folgenden spezifizierten Anforderungen auf wiederkehrender Abfrage-logik beruhen, wird jeweils beim ersten Vorkommen eines entsprechenden Falles (z.B. Prüfen, ob der Attributwert ein bestimmter Modellelementtyp ist) zunächst eine benötigte Schablone als Regelvorlage entwickelt. Diese wird dann mit den konkreten Bedingungen der Anforderung ausgeprägt. Somit werden bei den nachfolgenden Anforderungen, die auf dieses Grundkonzept der Prüfung bzw. Analyse aufbauen, an dieser Stelle nur noch auf die Beschreibung des entsprechenden ursprünglichen Rumpfs bzw. Schablone verwiesen und die darin definierten Schlüsselbegriffe zur Abbildung der Anforderung (wieder-) verwendet. Diese (Vor-) Entwicklung zur Umsetzung einer konkreten Anforderung beschreibt dabei die bei der Evaluierung der Software dann mit entsprechenden Werkzeugen der Regelentwicklung innerhalb einer Entwicklungsbibliothek benötigten Bearbeitungsschritte.

Standardregelvorlage – Zu dem am häufigsten wiederkehrenden Prüfprinzip – hat ein Modellelement ein bestimmtes Attribut, und wenn ja welchen Wert besitzt dieses – wird gemäß dem Regelaufbau eine Standardregelvorlage für die Entwicklung vorgehalten. Hierbei können die vier Schlüsselbegriffe wie folgt ausgeprägt werden, um mittels der hinterlegten Auswertungslogik ein entsprechendes Ergebnis zu erzielen:

Regelkomponente *Hauptelement*

Regelkomponente *Merkmal*

Attribut- oder Subelementname spezifizieren

Regelkomponente *Prüfkriterium*

Regelkomponente Soll-Wertausprägung

Querschneidende Hilfsfunktion für XSD Schemainstanzen – Durch den Bezug der spezifizierten Anforderungen auf die XML-basierte Variante der IFC sind darin auch spezifische XML Abbildungskonzepte berücksichtigt, wie beispielsweise die Konstrukte des XSD Schema Instance (XSI) Namensraumes. Als querschneidender Belang bilden sie eine Grundlage für (eigenständig kapselbare) *Hilfsfunktionen*. Exemplifiziert wird dies an ausgewiesenen Fallunterscheidungen, welche sich auf das, von Seiten der Syntaxbasis XSD in dem Anwendungsschema IFCXML übernommene, Grundkonzept von „leeren“ Verweisobjekten beziehen. Gemäß diesem eine flexible Strukturierung des XML-Instanzdokuments ermöglichenden Konzepts besitzen die Verweisobjekte jeweils ein Attribut *href*, dessen Wert mit dem Attribut *id* eines korrespondierenden Instanzobjektes übereinstimmt. Somit können, ob der streng-hierarchischen XML-Struktur relationale Beziehungsgeflechte ausgedrückt werden. Durch die Standardisierung dieses Prinzips²² für den Rahmen von XML-Prozessoren kann eine referenzielle Integrität – für jedes *href* existiert im XML-Instanzdokument ein entsprechendes *id* – bereits bei einer Schemaprüfung gewährleistet werden. Mit der Entwicklung einer Hilfsfunktion werden im Weiteren alle diesbezüglichen Fallunterscheidungen der Anforderungen behandelt. Diese Hilfsfunktion steht dabei beispielhaft für alle modellübergreifenden Methoden. Sie kann auch bei CityGML sowie allen anderen Modellformaten, die dieses Konstrukt in ihren XML Anwendungsschemata einbinden angewendet werden. Um insbesondere die vorhandenen Namensräume im Wurzelement eines XML-Dokuments beim initialen Einlesen zu interpretieren, sind mehrere dieser Hilfsmethoden in den standardmäßigen Rumpf der zentralen *POOLXML* Konfigurationsdatei eingebunden, die als

²² Vergleiche XLink Standard der W3C (<https://www.w3.org/TR/xlink/>)

Kern eines Regelkontexts den generellen Funktionsumfang der Regelbasis bestimmt (vgl. 5.1.2.3).

6.2.1 Regelbasis „Bürostandard“

Schemaprüfung und Metadaten eines XML-Dokuments

Die Überprüfungsfunktionalitäten bezüglich des dokumentenbasierten (IFC-) Modells zu den beiden folgenden ersten Einträgen im Lastenheft (vgl. 4.5) sind bereits als Teil der Businesslogik der Softwareanwendung realisiert. Standardmäßig im Hintergrund beim Einlesevorgang eines IFC-Modells ausgeführt, werden die Nutzer*innen über ihre Ereignismeldungen der Schemaprüfung mit entsprechenden Benachrichtigungsdialogen informiert (vgl. Kapitel 5). Zudem werden die Metadaten zum Modelldokument in einer Übersichtsansicht zum Modell präsentiert (Benutzeroberflächen in Kap. 7.1).

0.1.0 Metadaten der Modelldatei

0.2.0 Validität zum unterlegten Modellschema

Bestimmung aller Eigenschaftsobjekte eines IFC-Objekts

Im Lastenheft sind bereits zu den beiden zugrundeliegenden IFC-Abbildungsparadigma von Eigenschaftsobjekten jeweilige Abschnitte untergliedert. So können Eigenschaftsobjekte gemäß dem flexiblen IFC-Schema entweder direkt an das Objektvorkommnis (beispielsweise einer Wandbauteilinstanz) angebunden werden, oder sie werden über einen (möglicherweise auch mehreren Vorkommnissen) zugeordneten Typobjekt (beispielsweise einem Bauteiltyp einer massiven Außenwand) verknüpft. Als Grundlage der vollständigen Identifizierung aller Eigenschaftsobjekte eines Objektvorkommnisses wird daher im Lastenhefteintrag 0.3.0 zunächst die Ermittlung eines diesem ggf. zugeordneten Typobjekte gefordert. Im Pflichtenheft (siehe Tabelle 13) werden die dazu benötigten Teilfragestellung anhand der entsprechenden IFC Entitäten weiter spezifiziert und in logischer Abfolge bezüglich des Prüfvorgehens aufbereitet (Pflichtenhefteinträge 0.3.1 bis 0.3.3, vgl. Tabelle 13). Das eigentliche,

im Lastenhefteintrag 0.4.0 geforderte Ermitteln aller einem IFC Objekt – sowohl Objektvorkommnis, wie auch Typobjekt – zugeordneten Eigenschaftsobjekte spezifizieren dann (aufbauend auf den vorherigen) die Pflichtenhefteinträge 0.4.1-0.4.3.

Tabelle 13 Pflichtenheft Inhalte einer designierten Hilfsfunktion

IDX	Gegenstand	ANFORDERUNG
0.3.0	IFC Typobjekt Konzept	Das zugeordnete Typobjekt eines IFC Objektes identifizieren
0.3.1	IFC Objekt (alle Spezialisierungen der Basisklasse IfcObject)	A) Ist das Inverse Attribute ²³ IsTypedBy mit einer objektifizierten Assoziationsklasse IfcRelDefinesByType ausgeprägt und ist diese: A1) als Instanzobjekt verschachtelt, d.h. besitzt sie ein Attribut GlobalId A2) ein Verweisobjekt, das ein Instanzobjekt referenziert, d.h. besitzt sie ein Attribut href ?
		B) Ist das Attribut id des Objekts ausgeprägt ?
0.3.2	Objektifizierte Assoziationsklasse IfcRelDefinesByType	Existiert eine objektifizierte Assoziationsklasse IfcRelDefinesByType , die A) im Objekt verschachtelt ist (Fall 0.3.1 A1), B) als eigenständiges Instanzobjekt ein Attribut GlobalId ausgeprägt hat sowie B1) ein Attribut id besitzt, auf welches verwiesen wird (Fall 0.3.1 A2), B2) in ihrem Assoziationsende RelatedObjects ein Verweisobjekt besitzt, das mit einem entsprechenden Attribut href auf das Attribut id des Objekts verweist (Fall 0.3.1 B) ? Ist das Assoziationsende RelatedType

²³ Ab der Version 4 des ifcXML Schema ermöglicht das sogenannte Inverse Attribut eine Verschlinkung der XML-Instanzdokumente durch eine komprimiertere Repräsentation. Dadurch kann auf eine eigenständige Assoziationsklasse zugunsten einer direkten Verschachtelung innerhalb des Attributs verzichtet werden, wobei ein ansonsten benötigtes Assoziationsende entfällt.

		<p>C) ein Verweisobjekt auf ein entsprechendes Instanzobjekt mittels eines ausgeprägten Attributs href?</p> <p>D) verschachteltes²⁴ Instanzobjekt, d.h. mittels Attribut xsi:type spezifiziert und durch sein Attribut GlobalId instanziiert?</p>
0.3.3	<p>IFC Typobjekt (alle Spezialisierungen der Basisklasse IfcObjectType)</p>	<p>Existiert ein IFC Typobjekt,</p> <p>A) der mittels Attribut GlobalId instanziiert wurde und ein Attribut id besitzt, auf das verwiesen wird (Fall 0.3.2 C)?</p> <p>B) der in der Assoziationsklasse verschachtelt ist (Fall 0.3.2 D)?</p>
		<p>Leitet das Typobjekt von der Basisklasse IfcObjectType ab sowie (nur wenn Typenname mittels Suffix "Type" von dem Objektnamen gebildet wird!) entspricht die Spezialisierung der des verweisenden IFC Objektes, z.B. IfcWall – IfcWallType)?</p>
0.4.0	IFC Eigenschaftskonzept	Alle einem IFC Objekt zugeordneten Eigenschaften identifizieren
0.4.1	<p>Sammlung aller dem IFC Objekt direkt (bzw. über sein Typobjekt) zugeordneter Eigenschaftssatzdefinitionen (gilt für alle Spezialisierungen der Basisklasse IfcPropertySetDefinitionSelect z.B.</p>	<p>A) Ist das <i>Inverse Attribute IsDefinedBy</i> des IFC Objektes mit einer Kollektion objektifizierter Assoziationsklassen IfcRelDefinesByProperties ausgeprägt und ist (mindestens ein) Mitglied dieser:</p> <p>A1) als Instanzobjekt verschachtelt, d.h. besitzt es ein Attribut GlobalId?</p> <p>A2) ein Verweisobjekt, das ein Instanzobjekt referenziert, d.h. besitzt es ein Attribut href?</p> <p>B) Existiert eine eigenständige objektifizierte Assoziationsklasse IfcRelDefinesByProperties, die in ihrem Assoziationsende RelatedObjects ein Verweisobjekt mit einem Attribut href besitzt, das auf das Attribut id des IFC Objektes verweist und dessen Assoziationsende</p>

²⁴ Das IFCXML Schema erlaubt bei vielen Abbildungskonzepten neben der Ausprägungsmöglichkeit eines Verweises zwischen zwei (eigenständigen) Objekten mittels des Attributpaars *id* und *href* auch ein direktes Verschachteln des verwiesenen Objekts.

	IfcPropertySet, IfcQuantitySet)	<p>RelatingPropertyDefinition mindestens ein Eintrag hat, der</p> <p>B1) als Instanzobjekt verschachtelt, d.h. ein Attribut GlobalId besitzt?</p> <p>B2) als Verweisobjekt ein Instanzobjekt referenziert, d.h. ein Attribut href besitzt?</p> <p>Wenn das Objekt ein Typobjekt verweist (Fall 0.3.0)</p> <p>C) Ist die IfcPropertySetDefinition-Kollektion HasPropertySets des Typobjektes ausgeprägt und besitzt mindestens einen Eintrag?</p>
0.4.2	IFC Eigenschaftsätze filtern	Alle mit 0.4.1 sondierten Objekte, welche die Klasse IfcPropertySet implementieren.
0.4.3	Zugeordnete Eigenschaften (alle Spezialisierungen der Basisklasse IfcProperty)	<p>Objekte deren Attribut HasProperties ausgeprägt und (mindestens) ein Eintrag hat, der als Objekt</p> <p>A) eine Spezialisierungsklasse von IfcSimpleProperty implementiert</p> <p>B) die Eigenschaftskollektion IfcComplexProperty implementiert und (iterativ) 0.4.3 erfüllt.</p>

Konzeption und Entwicklung einer dreigeteilten Hilfsfunktion

Als übergreifende Umsetzung der Pflichtenhefteinträge 0.3.0 und 0.4.0 wird im Folgenden ein konkreter Modellprüfungsvorgang als parametrisierter Teil des Prüfplans entwickelt, der auf jedes in den weiteren Anforderungen erforderliche Prüfen einer Eigenschaft (sowie das in IFC artverwandte Beschreibungskonzept von Mengen – **IfcQuantitySet**) angewendet werden kann. Neben der oben angeführten Hilfsfunktion werden hierfür die in anderen übergeordneten Fragestellungen wiederverwendbaren Prüfschritte, wie beispielsweise das Identifizieren von Typobjekten eines Objektvorkommnisses, in drei weiteren Teilhilfsfunktionen gekapselt. Hierbei definierte, und mit einer Logik zur Behandlung des IFC-Modells belegte, Hilfsfunktionsnamen sowie dazugehörige Argumentnamen bilden die Variablen in einer Schablone, in der abschließend alle Teilhilfsfunktionen in die konsistente Abfolge des gesamtheitlichen Modellprüfungsvorganges zusammengeführt werden. Dieser Vorgang wird dabei ergänzt mit entsprechenden Fallunterscheidungen als Basis der Einzelauswer-

tungen der inbegriffenen Teilhilfsfunktionen beim Durchführen auf eine Modellinstanz. Als einzelne aufeinander aufbauende Abschnitte wird im Weiteren die inhaltliche Grundlage entwickelt, die zur Formalisierung entsprechender Hilfsfunktionen und einer diese zusammenführende, formalisierte Beispielregel (Schablone) in der Softwareanwendung benötigt wird (vgl. Kap. 2517).

Teilhilfsfunktion (0.3.1 - 0.3.3) – Bereits im Lastenheft sind als Grundlage der Entwicklung eines Prüfplans zur Kontrolle aller einem Modellobjekt zugeordneten Eigenschaften, zwei wesentliche Anforderungsbereiche notiert. Unabhängig von den direkt an das Objektvorkommnis verknüpften Eigenschaften, werden zunächst die ggf. über ein dem Vorkommnis zugeordnetes Typobjekt verknüpfte (gemeinsame) Eigenschaften berücksichtigt. Das Behandeln der Zuordnung zu einem Typobjekt als generelles Konzept der IFC, wird zur besseren Wiederverwendung innerhalb der für eine Hilfsfunktion exemplarisch verfolgten Entwicklungsstrategie dabei in einer eigenständigen Teilhilfsfunktion gekapselt. Damit lässt sich die entwickelte Logik zum Filtern der einem Objektvorkommnis zugeordneten Typobjekte (vgl. Abbildung 54) über den exemplifizierend auskonkretisierten Fall in der Hilfsfunktion hinaus bezüglich der Eigenschaften generell wiederverwenden.

Hilfsfunktion GetTypeObject

Argument GetTypeObject_obj

Entwickelte Logik

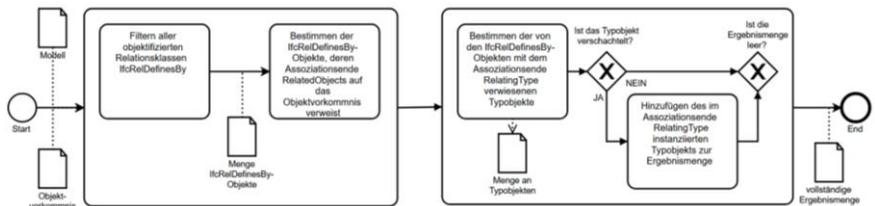


Abbildung 54 Logische Abfolge zum Filtern von Typobjekten

Teilhilfsfunktion (0.4.1- 0.4.2) – Unter Einbeziehung der oben beschriebenen Teilmethode zum Auslesen der einem Objektvorkommnis zugeordneten Typobjekte, kann nun die logische Abfolge zur Bestimmung aller Eigenschaftsobjekte zu dem Vorkommnis entwickelt werden. Durch die von IFC vorgegebene Abbildungskonzeption, dass jedes von der Basisklasse IfcProperty abgeleitete Eigenschaftsobjekt Mitglied eines Eigenschaftssatzobjektes sein muss, bildet das Sammeln aller Sätze hierbei den ersten Schritt. Darin werden die objektifizierte Assoziationsklassen *IfcRelDefinesByProperties*, die ein Objektvorkommnis direkt oder über ein zugeordnetes Typobjekt, verweisen ausgewertet (vgl. Abbildung 55).

Hilfsfunktion `GetPropertySetsOfObject`

Argument `GetPropertySetsOfObject_obj`

Entwickelte Logik

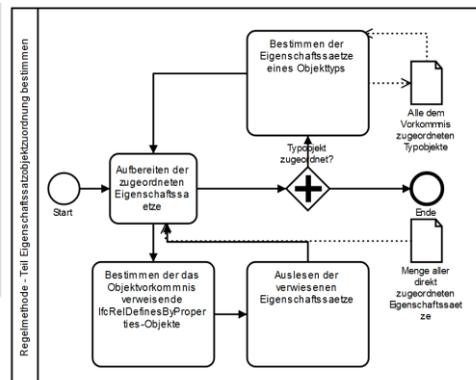


Abbildung 55 Logische Abfolge zum Filtern von Eigenschaftssätzen

Teilhilfsfunktion (0.4.3) – Im letzten Schritt werden die als Teilergebnis ermittelte Menge aller Eigenschaftssätze eines Objektvorkommnisses bezüglich der beinhalteten Eigenschaften gefiltert. Das zweite Argument Eigenschaftsname bildet das Kriterium nach dem gefiltert wird. Zunächst wird in der Teil-

methode die Menge aller Eigenschaftsobjekte des im ersten Argument bestimmten Eigenschaftssatzes gebildet. Hierzu werden die im Attribut HasProperties des Eigenschaftssatzes zugeordneten Objekte ausgewertet. Dieser Vorgang ist dabei in eine Hilfsmethode gekapselt, damit er unabhängig von dieser Teilmethode bei der Auswertung in der Gesamtmethode (siehe unten) wieder verwendet werden kann. Eine mehrfach mögliche Verschachtelung von Eigenschaftskollektionen im Konzept einer komplexen Eigenschaft (IfcComplexProperty) wird dabei berücksichtigt (vgl. Abbildung 56). In einer Schleife wird dabei jeweils die Hilfsmethode für jede komplexe Eigenschaft aufgerufen, so dass die zurückgegebene Menge sämtliche Eigenschaften vollständig enthält. Die gebildete Menge wird nun nach dem gesuchten Eigenschaftsnamen gefiltert. Als Rückgabewert liefert die Teilmethode eine Menge der gefundenen Eigenschaften (bzw. wenn nicht vorhanden eine leere Menge).

Hilfsfunktion TryGetPropertyValueByName

1. Argument TryGetPropertyValueByName_propertySet
2. Argument TryGetPropertyValueByName_propertyName

Entwickelte Logik

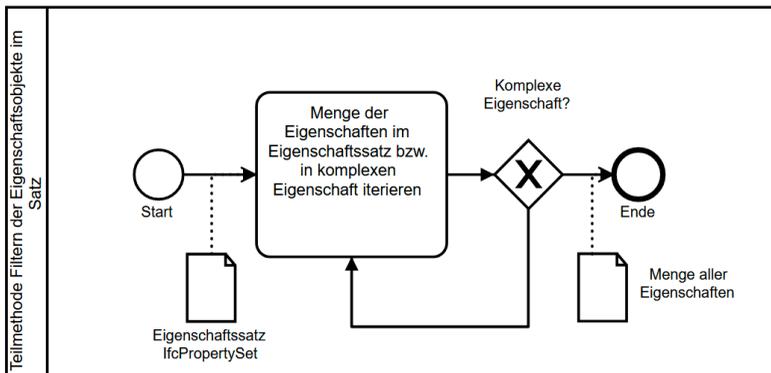


Abbildung 56 Filtern der Eigenschaften eines Satzes

Zusammenführen der Teilhilfsfunktionen in eine Schablone – Die Logik in den drei Hilfsfunktionen bildet die Grundlage der exemplifizierten „übergeordneten“ Hilfsfunktionen zum Überprüfen von Eigenschaftssätzen bzw. Eigenschaften gemäß dem von IFC vorgegebenen eindeutigen Namensattribut bei Eigenschafts(satz)objekten, sowie korrespondierend abgebildeten Mengen(satz)objekten. Folgende Logische Abfolge (siehe Abbildung 57) zeigt die verschalteten Teilhilfsfunktionen und die möglichen Ergebnisse. Diese Logik bildet den designierten Inhalt einer Regelschablone, mit welcher durch Ausprägen der Variablen (1) Objektvorkommnis oder Typobjekt, (optional 2) Eigenschaftssatzname, (optional 3) Eigenschaftsname sowie ggf. (optional 4) Wertausprägung das IFC Eigenschaftskonzept (sowie in leicht abgeänderter Form das IFC Mengenkonzept) überprüft werden kann.

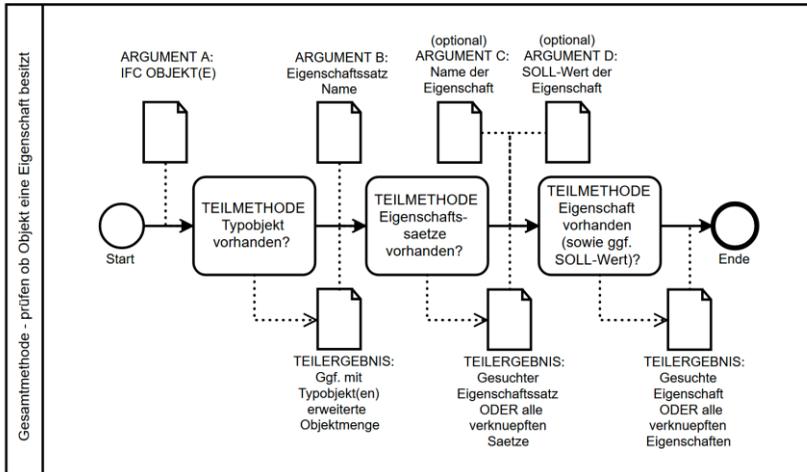


Abbildung 57 Logik einer Schablone für zusammengeführten Teilmethodenaufwurf

Diese vier Variablen bilden demnach die Argumente der Hilfsmethode zum Untersuchen der IFC Eigenschaften und stellen dabei die Anknüpfungspunkte der von Seiten der Anwender*innen auszuprägenden Teilen beim Designieren eigener Regel von dieser Schablone dar. Durch das für alle Regeln einheitlich eingeführte Bezeichnerkonzept des Regelaufbaus können dabei sowohl das erste Argument mit der Regelkomponente *Hauptelement*, als auch das Vierte

mit der Regelkomponente *Soll-Wertausprägung* direkt verknüpft werden (vgl. Kap. 3.2.1). Eine nutzerspezifische Bezeichnung der beiden anderen Argumente wird durch die Regelkomponente *Merkmal* verknüpft. Mit Bezug zu der Struktur des behandelten Modellformates ifcXML sind dazu folgende (selbst erklärenden) Schlüsselbegriffe festgelegt:

- Attribut Name des assoziierten Eigenschaftssatzes
- Attribut Name der assoziierten Eigenschaft

Erweiterte Ausgaben der Hilfsfunktion – Wird keines der beiden Argumente beim Anwenden der Hilfsfunktion ausgeprägt, so werden alle dem angegebenen Hauptelement zugeordneten Eigenschaftssätze in einer Objektmenge als Ergebnis zurückgegeben. Dieselbe Rückgabe erfolgt, wenn zwar ein Eigenschaftssatzname als Argument ausgeprägt wird, dieser jedoch nicht beim Ausführen der Hilfsfunktion gefunden werden kann. Nach diesem Prinzip wird äquivalent verfahren, wenn, neben einem Argument *Eigenschaftssatzname* (für das ein entsprechend gesuchter Satz auch tatsächlich mit dem Hauptelement verknüpft ist) in einem zweiten Argument ein Eigenschaftsname bestimmt wird. Kann dabei dann der gesuchte Eigenschaftsname nicht in der gefilterten Eigenschafts-Menge des Eigenschaftssatzes gefunden werden, so wird die ganze Menge zu Auswertungszwecken zurückgegeben. Andernfalls wird die durch Angabe ihres Namens gesuchte Eigenschaft zurückgegeben. Durch diese Rückgabe eines gesuchten Eigenschaftsobjekts, bildet die Teilhilfsfunktion die Grundlage, um eine (Soll-)Wertprüfung darauf aufzubauen. Da diese Werteüberprüfung in der Grundstruktur der Regel verankert ist (vgl. Soll-Ausprägung-Komponente in Kap. 3.2.1 sowie Unterpunkt i), wird beim Ausprägen eines geforderten Soll-Wertes zu einer gefundenen Eigenschaft auf die standardmäßigen Rückgaben zurückgegriffen. Entspricht der Wert nicht dem geforderten, wird dabei der tatsächliche Wert zurückgegeben.

6.2.2 Regelbasis „Planungsprojekt“

Anmerkung zu Lastenhefteinträgen 1.1.1 bis 1.1.3 sowie 1.1.5 – Die Punkte 1.1.1 und 1.1.2 beziehen sich auf eine Datei als Binärformatierung des XML-

Inстанздokuments und somit direkt auf Informationen, die von der Business Logic der Softwareanwendung im Falle des Einlesens einer Modelldatei bereits behandelt werden. Prinzipiell kann eine Regeleinheit zur Prüfung des Dateinamens konzipiert werden. Da jedoch zum einen im Auswahlfenster zur Selektion einer Modelldatei der Dateityp „IFCXML“ festgeschrieben ist, sowie zum anderen in einem Detailfenster die Metainformationen zur Modelldatei einsehbar sind, wird diese Anforderung nicht umgesetzt. Die Schemaprüfung der XML-Dokumentinstanz (1.1.5), sowie das Prüfen der Byte Order Mark (BOM, 1.1.3) werden ebenfalls standardmäßig in der Businesslogik der Softwareanwendung beim Einlesevorgang des Modells durchgeführt (und entsprechende Ereignisse in der Benutzeroberfläche als eigenständiges Dialogfenster angezeigt). Daher werden diese prinzipiell in einer Regelkonzeption abbildbaren Prüfungen auch nicht umgesetzt.

6.2.2.1 Metadaten

Das Prüfen der korrekten Versionsangabe des IFC Schemas im XML-Instanzdokument wird am Wurzelement vollzogen. Darin sind mit dem Namensraum die Version des im XML-Instanzdokument persistierten Modells bestimmt.

IDX	Gegenstand	ANFORDERUNG
1.1.4	IFC Standard Versionsangabe	Ist die korrekte IFC XML Version ADD2 im Kopfteil des Dokuments als Hauptnamensraum (<code>xmlns</code>) angegeben?
1.1.4.1	Wurzelement des Instanzdokuments	A) Existiert das Wurzelement <code>ifcXML</code> ?
		B) Hat das Attribut <code>xmlns</code> sowie (bei Vergabe eines entsprechenden Subnamensraums IFC) das Attribut <code>xmlns:ifc</code> den Wert <code>http://www.buildingsmart-tech.org/ifcXML/IFC4/Add2?</code>

6.2.2.2 Vorgaben Klassifizierung und Grundstruktur

In diesem Abschnitt des Pflichtenhefts wird die grundlegende strukturelle und formale Qualität des Modells untersucht. In Tabelle 14 sind gemäß dem Lastenheft bezugnehmend zu den IFC Abbildungsvorgaben die einzelnen Anforderungen konkretisiert.

Tabelle 14 Pflichtenheft Klassifizierung und Grundstruktur

IDX	Gegenstand	ANFORDERUNG
1.2.1	Projektklassifizierung: Name, Phase	Existiert ein Projektobjekt mit der korrekten Angabe des Projektnamens Musterprojekt sowie der Angabe der aktuellen (Planungs-) Phase des Projekts?
1.2.1.1	IfcProject Objekt	A) Existiert (nur) ein IfcProject Objekt das ein ausgeprägtes Attribut GlobalId besitzt? B) Wert des Attributs Name ist „Musterprojekt“? C) Wert des Attributs Phase ist „Entwurfsphase“?
1.2.2	Projektklassifizierung: Autor, Tool, Erstellungsdatum	Ist der Autor, das CAD-Werkzeug sowie das Erstellungs- bzw. Änderungsdatum der aktuellen Projektinstanz angegeben?
1.2.2.1	IfcProject Objekt	Kollektion OwnerHistory mit mindestens zwei Einträgen (IfcOwnerHistory -Objekte) ausgeprägt?
1.2.2.2	IfcOwnerHistory-Objekt	Attribut OwningUser mit einem IfcPersonAndOrganization -Objekt ausgeprägt, in dem A) das im Attribut ThePerson ein IfcPerson -Objekt besitzt, welches im Attribut FamilyName den Wert „Musterautor“ trägt B) das im Attribut TheOrganization ein IfcOrganization -Objekt besitzt, welches im Attribut Name der Wert „Musterbüro“ trägt
1.2.2.3		Attribut OwningApplication mit einem IfcApplication -Objekt ausgeprägt, in dem das im Attribut ApplicationFullName den Wert „Muster CAD-System“ trägt
1.2.2.4		Attribut CreationDate mit einem IfcTimeStamp -Objekt ausgeprägt, das den Wert „1526922926 ²⁵ “ hat.
1.2.2.5		Attribut LastModifiedDate mit einem IfcTimeStamp -Objekt ausgeprägt, das den Wert „1526922926“ hat.

²⁵ Gemäß IEEE Std 1003.1. entspricht dies Sekunden die seit 1 Jan 1970, 00:00:00 UTC am 2018-05-21T17:15:26Z verstrichen sind

1.2.3	Projektklassifizierung: Basiseinheiten	Sind die vorgegebenen standardisierten (metrischen) Einheiten für den Projektkontext deklariert?
1.2.3.1	IfcProject Objekt	Attribut UnitsInContext mit einem IfcUnitAssignment -Objekt ausgeprägt?
1.2.3.2	IfcUnitAssignment-Objekt	Kollektion Units mit folgenden von der Klasse IfcSIUnit abgeleiteten Objekten ausgeprägt, die jeweils ein Attribut UnitType mit einem IfcUnitEnum -Enumerationswert sowie ein Attribut Name mit einem IfcSIUnitName -Enumerationswert? Längeneinheit: LENGTHUNIT, METRE Flächeneinheit: AREAUNIT, SQUARE_METRE Volumeneinheit: VOLUMEUNIT, CUBIC_METRE Masseinheit: MASSUNIT, GRAM Winkelmaßeinheit: PLANEANGLEUNIT, RADIAN Thermodynamische Temperatureinheit: THERMODYNAMICTEMPERATUREUNIT, DEGREE_CELSIUS Zeiteinheiten: TIMEUNIT, SECOND Energieeinheit: ENERGYUNIT, JOULE
1.2.4	Projektklassifizierung: Koordinatensystem	Ist für die geometrische Beschreibung ein (Welt-) Koordinatensystem mit Ursprungspunkt sowie der Angabe der (wahren) Nordrichtung angegeben?
1.2.4.1	IfcProject Objekt	Attribut RepresentationContexts mit IfcGeometricRepresentationContext -Objekt ausgeprägt?
1.2.4.2	IfcGeometricRepresentation-Context-Objekt	Ist das Attribut WorldCoordinateSystem mit einem IfcAxis2Placement3D -Objekt sowie das Attribut TrueNorth mit einem IfcDirection -Objekt ausgeprägt?
1.2.4.3	IfcAxis2Placement3D-Objekt	Attribut Location mit IfcCartesianPoint -Objekt ausgeprägt, dessen Attribut Coordinates eine Kollektion von drei IfcLengthMeasure -Primitivdatentypen jeweils mit dem Wert 0 enthält?

1.2.4.4	IfcDirection-Objekt	Attribut DirectionRatios mit drei IfcReal -Primitivdatentyp (Werte: 6.12303176911189E-17, 1,) ausgeprägt?
1.2.5	Projektklassifizierung: einheitliche Sammlung der Bauteiltypen	Sind alle Typobjekte mit den gemeinsamen Eigenschaften der im Projektkontext vorkommenden Bauteile an der designierten Stelle unterhalb des Projektobjektes instanziiert?
1.2.5.1	IfcProject Objekt (deklarierte Typobjekte)	Ist das Attribut Declares mit einem IfcRelDeclares -Objekt ausgeprägt, dessen Attribut RelatedDefinitions eine Kollektion an den, von der Basisklasse IfcTypeObject abgeleiteten, Typobjekten enthält, z.B. IfcWallType ? Besitzen alle darin enthaltenen Typobjekte ein GlobalId Attribut?
1.2.5.2	ifcXML Objekt	Entspricht die in 1.2.5.1 zusammengestellte Sammlung an Typobjekten allen im Instanzdokument vorkommenden Typobjekten, die entsprechenden Verweisobjekte nicht eingerechnet?
1.2.6	Projektklassifizierung: Eigenschaften und assoziierte Quellen	Besitzt das Projekt die vorgegebenen Eigenschaften zum Design Brief, in denen bauteilbezogene Verweise zu den (im angehängten Tabellendokument) energetischen Zielwerten aufgeführt sind?
1.2.6.1	IfcProject Objekt	A) Ist das Attribut IsDefinedBy mit einem IfcRelDefinesByProperties -Objekt ausgeprägt, dessen Assoziationsende RelatingPropertyDefinition ein IfcPropertySet -Objekt enthält? B) Ist das Attribut HasAssociations mit einem IfcRelAssociatesDocument -Objekt ausgeprägt?
1.2.6.2	Eigenschaftssatzobjekt IfcPropertySet	Existiert das Attribut id und enthält das Attribut Name den Wert Pset_DesignBrief sowie die Kollektion HasProperties ein IfcComplexProperty -Objekt?
1.2.6.3	Eigenschaftskollektionsobjekt IfcComplexProperty	Hat das Attribut Name den Wert Pset_EnergeticDesignTarget , das Attribut UsageName den IfcIdentifizier -Primitivdatentyp Wert „ProjectAssociatedDocument_DesignBriefAnnex_EnergeticDesignTarget“,

		das Attribut Description den Wert „Erreichen eines Effizienzhausstandard 40 nach ENEC 2016“ sowie im Attribut HasProperties ein IfcPropertySingleValue-Objekt?
1.2.6.4	Eigenschaftsobjekt IfcPropertySingleValue	Hat das Attribut Name den Wert „P_Referenz_U-Wert_Aussenbauteile“ sowie das Attribut NominalValue den IfcText -Primitivdatentyp Wert „Spalte 4, Zeile 5“ (vgl. Tabelle 6)?
1.2.6.5	IfcRelAssociatesDocument-Objekt	Enthält das Assoziationsende RelatedObjects ein IfcPropertySet -Verweisobjekt mit einem Attribut href , dessen Wert auf das id Attribut des Instanzobjektes im Fall 1.2.6.1 A verweist? Enthält das Assoziationsende RelatingDocument ein IfcDocumentInformation -Objekt?
1.2.6.6	IfcDocumentInformation-Objekt	Hat das Attribut Name den Wert „ProjectAssociatedDocument_DesignBriefAnnex_EnergeticDesignTarget“ sowie das Attribut Location den IfcURIReference -Datentyp ²⁶ Wert „ProjectAssociatedDocument_DesignBriefAnnex_EnergeticDesignTarget.xlsx“ (Tabelle 6)?
1.2.7	Projektklassifizierung: Ausgangspunkt räumliche Gliederung	Fußt die räumliche Gliederung an der designierten Stelle unterhalb des Projektobjektes?
1.2.7.1	IfcProject Objekt	Ist das Attribut IsDecomposedBy mit IfcRelAggregates -Objekt ausgeprägt, dessen Assoziationsende RelatedObjects (mindestens) ein von der räumlichen abstrakten Basisklasse IfcSpatialStructureElement abgeleitetes Objekt besitzt?
1.2.8	Räumliche Hauptdekomposition	Ist die instanziierte räumliche Dekomposition des IFC-Modells konsistent (sowie richtig Klassifiziert) aufgebaut? Vorgabe: Projekt -> Baugrund -> Bauwerk -> Stockwerk -> Raum

²⁶ Uniform Resource Identifier (URI) – nach der Definition in RFC 3986 der Network Working Group NWG Standard

1.2.8.1	Ebene des Baugrundstücks	Instanziiert das einzige im designierten Ausgangspunkt der räumlichen Dekomposition (Fall 1.2.7.1) existierende Objekt die Klasse IfcSite ?
1.2.8.2	IfcSite Objekt	Ist das Attribut IsDecomposedBy mit IfcRelAggregates -Objekt ausgeprägt, dessen Assoziationsende RelatedObjects ein IfcBuilding -Objekt enthält?
1.2.8.3	IfcBuilding Objekt	Ist das Attribut IsDecomposedBy mit IfcRelAggregates -Objekt ausgeprägt, dessen Assoziationsende RelatedObjects (mindestens) ein IfcBuildingStorey -Objekt enthält?
1.2.8.4	IfcBuildingStorey Objekt	A) Ist das Attribut IsDecomposedBy mit IfcRelAggregates -Objekt ausgeprägt, B) dessen Assoziationsende RelatedObjects (mindestens) ein IfcSpace -Objekt enthält?
1.2.8.5	ifcXML Objekt	Entsprechen die in 1.2.8.1-1.2.8.4 identifizierten räumlichen (von der Basisklasse IfcSpatialElement abgeleiteten) Objekte insgesamt den Vorkommnissen im Instanzdokument (entsprechende Verweisobjekte nicht eingerechnet)?
1.2.9	Grundordnung der Bauteile	Sind alle (physischen) Bauteil-Vorkommnisse korrekt in der räumlichen Gliederung (auf Stockwerkebene) eingeordnet und ist dies gemäß Dokumentationsvorgabe in ihren Namen richtig ausgezeichnet?
1.2.9.1	ifcXML Objekt	Welche Bauteilobjekte (ausgenommen: IfcBuildingElementProxy , IfcMember sowie IfcCovering), die von der Basisklasse IfcBuildingElement ableiten und durch ein ausgeprägtes Attribut GlobalId gekennzeichnet sind, existieren im XML-Instanzdokument?
1.2.9.2	Bauwerksebene (Sonderfall stockwerksübergreifender Bauteile z.B. Treppen)	Befinden sich von den im Fall 1.2.9.1 ermittelten Objekten alle Instanzen der Bauteilklassen IfcStair bzw. IfcCurtainWall im Assoziationsende RelatedElements eines IfcRelContainedInSpatialStructure -Objekts, das im Attribut ContainsElements eines Bauwerkobjekts (Fall 1.2.8.3) residiert?

1.2.9.3	Dokumentationsvorgabe zu 1.2.9.2	Entspricht das Namenspräfix im Attribut Name exakt dem Gebäudekürzel „G.1_“?
1.2.9.4	Stockwerks- bzw. Raumbene	Befinden sich, außer den in Fall 1.2.9.2 behandelten, alle weiteren im Fall 1.2.9.1 ermittelten Objekte jeweils im Assoziationsende RelatedElements eines IfcRelContainedInSpatialStructure -Objekt, das A) im Attribut ContainsElements eines Stockwerkobjekts (Fall 1.2.8.4) residiert? B) im Attribut ContainsElements eines Raumobjekts (Fall 1.2.8.4 B) residiert?
1.2.9.5	Dokumentationsvorgabe zu Objekten im Fall 1.2.9.3 A	Ist das Namenspräfix im Attribut Name zusammengesetzt aus: 1) einem Kürzel „G.1_“ 2) einem Ebenen-Index* für das Stockwerkobjekt? Beispiel: „G.1_E.1“ * entspricht dem Präfix des Stockwerknamens
1.2.9.6	Dokumentationsvorgabe zu Objekten im Fall 1.2.9.3 B	Ist das Namenspräfix im Attribut Name zusammengesetzt aus: 1) einem Kürzel „G.1_“ 2) einem Ebenen-Index* für das dem Raumobjekt gemäß Fall 1.2.8.4 A übergeordnete Stockwerkobjekt 3) einem Raum-Index (R.1, R.2 usw.)? Beispiel: „G.1_E.1_R.1“ * entspricht dem Präfix des Stockwerk- bzw. Raumnamens
1.2.10	Klassifizierung der Bauteile	Entsprechen alle Bauteilobjekte den vorgegebenen Klassifizierungen (keine Stellvertreterobjekte)?
1.2.10.1	Objektmenge im Fall 1.2.9.1	Leitet keines der Objekte von der Klasse IfcBuildingElementProxy ab?
1.2.10.2	ifcXML Objekt	Leitet keines der Objekte von der Klasse IfcProxy ab?
1.2.11	Bauteiltypen zu allen Bauteilen	Ist jedem vorkommenden Bauteil (Komponente, Schicht usw.) ein korrekter Bauteiltyp zugeordnet

		und darin als gemeinsame Eigenschaft mindestens die Klassierung nach DIN 276 korrekt angegeben?
1.2.11.1	ifcXML Objekt	Welche Bauteilobjekte (ausgenommen: IfcBuildingElementProxy), die von der Basisklasse IfcElement ableiten und durch ein ausgeprägtes Attribut GlobalId gekennzeichnet sind, existieren im XML-Instanzdokument?
1.2.11.2	Objektmenge im Fall 1.2.11.1	Existiert im <i>Inverse Attribute</i> IsTypedBy ein IfcRelDefinesByType -Objekt dessen Assoziationsende RelatingType A) als typisiertes Verweisobjekt mit dem Attribut xsi:type ausgeprägt und dessen Attribut href auf ein Attribut id eines entsprechenden Typobjekts verweist? B - Warnung nicht wohlgeformt verschachtelte Instanz) als Typobjekt mit dem Attribut xsi:type ausgeprägt und als Objektinstanzmerkmal das Attribut GlobalId trägt?
1.2.11.3	Verwiesenes Typobjekt	Besitzt das Typobjekt im Attribut HasPropertySets ein IfcPropertySet -Objekt?
1.2.11.4	Eigenschaftsobjekte des Eigenschaftssatzobjekt	Besitzt das IfcPropertySet -Objekt im Attribut HasProperties ein IfcSinglePropertyValue -Objekt mit dem Namensattributwert „P_Klassierung_DIN276“ sowie einer gültigen Wertausprägung des Attribut NominalValue ?
1.2.11.5	Klassifizierungsreferenz IfcClassificationReference	Existiert ein IfcRelAssociatesClassification -Objekt, das in seinem Assoziationsende RelatedObjects auf das Typobjekt verweist und im Assoziationsende RelatingClassification ein IfcClassificationReference -Objekt besitzt?

6.2.2.3 Spezifische Modellierungsvorgaben

Tabelle 15 Pflichtenheft Spezifische Modellierungsvorgaben

IDX	Gegenstand	ANFORDERUNG
1.3.1	Bauteiltyp	Ist die Ausführungsart im Objekt des Wand-Bauteiltyps als dessen Spezialisierung (elementiert bzw. massiv) ausgezeichnet?
1.3.1.1	Wandbauteilobjekt (Korrektes PredefinedType)	Ist das Wandobjekt eine Instanz der Klasse IfcWallElementedCase und hat ein IfcWallType Wandtypobjekt assoziiert, das als Enumerationswert im Attribut PredefinedType einen der folgenden Werte besitzt: A) ELEMENTEDWALL B) SOLIDWALL
1.3.1.2	Wandbauteilobjekt (Korrektes PredefinedType)	(Kontercheck:) Verfügt das IfcWallElementedCase -Objekt selbst im Attribut PredefinedType über keine Wertausprägung?
1.3.2	Dekomposition des Bauteiltyps	Ist der Aufbau der Wand auf Ebene der Bauteil-, Komponenten- bzw. Schichtentypen korrekt durch entsprechende Referenzierungen abgebildet?
1.3.2.1	Aufbau der Typobjekte im Fall 1.3.1 A	Besitzt das IfcWallType -Typobjekt in der Kollektion IsDecomposedBy folgende Typobjekte: 1) IfcBuildingElementPartType -Objekt mit Präfix des Namensattributwerts „Außenpaneel“ sowie Enumerationswert PRECASTPANEL im Attribut PredefinedType 2) IfcElementAssemblyType -Objekt mit Präfix des Namensattributwerts „Gedämmter Rahmen“ sowie Enumerationswert BRACED_FRAME im Attribut PredefinedType und dessen Kollektion IsDecomposedBy folgende Typobjekte enthält: a) IfcMemberType -Objekt Präfix des Namensattributwerts „Riegel“ sowie Enumerationswert PLATE im Attribut PredefinedType b) IfcMemberType -Objekt Präfix des Namensattributwerts „Pfosten“ sowie Enumerationswert STUD im Attribut PredefinedType

		<p>c) IfcBuildingElementPartType-Objekt Präfix des Namensattributwerts „Wärmedämmung“ sowie Enumerationswert INSULATION im Attribut PredefinedType</p> <p>d) GGF. Verweisobjekt auf 1.3.3.1 3 als aussteifende Gipsfaserplatten</p> <p>3) IfcBuildingElementPartType-Objekt mit Präfix des Namensattributwerts „Innenpaneel“ sowie Enumerationswert PRECASTPANEL im Attribut PredefinedType</p>
1.3.2.2	Aufbau der Typobjekte im Fall 1.3.1 B	<p>Besitzt das IfcWallType-Typobjekt in der Kollektion IsDecomposedBy folgende IfcBuildingElementPartType-Typobjekte:</p> <p>1) Präfix des Namensattributwerts „Außenputzschicht“ sowie Enumerationswert USERDEFINED im Attribut PredefinedType</p> <p>2) Präfix des Namensattributwerts „Wärmedämmung“ sowie Enumerationswert INSULATION im Attribut PredefinedType</p> <p>3) Präfix des Namensattributwerts „Mauerwerk“ sowie Enumerationswert USERDEFINED im Attribut PredefinedType</p> <p>4) Präfix des Namensattributwerts „Innenputzschicht“ sowie Enumerationswert USERDEFINED im Attribut PredefinedType</p>
1.3.3	Dekomposition Wandbauteilobjekt	Entspricht der subordinierte Aufbau aus Schicht- bzw. Komponentobjekten des spezifischen Wandbauteilobjekts der Abbildung auf Ebene der Typobjekte?
1.3.3.1	Aufbau der Bauteilobjekte im Fall 1.3.1 A	<p>Besitzt das IfcWallElementedCase-Objekt in der Kollektion IsDecomposedBy folgende Objekte:</p> <p>1) IfcBuildingElementPart-Objekt mit Präfix des Namensattributwerts „Außenpaneel“</p> <p>2) IfcElementAssembly-Objekt mit Präfix des Namensattributwerts „Gedämmter Rahmen“, dessen Kollektion IsDecomposedBy (ggf. mehrfach) Objekte mit folgenden Merkmalen enthält:</p>

		<p>e) IfcMember-Objekt Präfix des Namensattributwerts „Riegel“</p> <p>f) IfcMemberType-Objekt Präfix des Namensattributwerts „Pfofen“</p> <p>g) IfcBuildingElementPart-Objekt Präfix des Namensattributwerts „Wärmedämmung“</p> <p>h) GGF. Verweisobjekt auf 1.3.3.1 3 als aussteifende Gipsfaserplatten</p> <p>3) IfcBuildingElementPart-Objekt mit Präfix des Namensattributwerts „Innenpaneel“</p>
1.3.3.2	Aufbau der Bauteilobjekte im Fall 1.3.1 B	<p>Besitzt das IfcWallElementedCase-Objekt in der Kollektion IsDecomposedBy folgende IfcBuildingElementPart-Objekte:</p> <p>1) Präfix des Namensattributwerts „Außenputzschicht“</p> <p>2) Präfix des Namensattributwerts „Wärmedämmung“</p> <p>3) Präfix des Namensattributwerts „Mauerwerk“</p> <p>4) Präfix des Namensattributwerts „Innenputzschicht“</p>
1.3.4	Gemeinsame Eigenschaften	Sind in allen Typobjekten des dekomponierten Wand-Bauteiltyps die jeweiligen Eigenschaften (Klassierung nach DIN 276: Kostengruppe 330 bzw. deren Unterpunkte Wandbekleidung usw.) inhaltlich kohärent ausgezeichnet?
1.3.4.1	Konkrete gemeinsame Eigenschaften	Entspricht der Wert einer gemäß dem Fall 1.2.11.2 und 1.2.11.3 den folgenden Typobjekten zugeordneten Eigenschaft jeweils den ausgewiesenen DIN 276 Schlüsseln?
1.3.4.2	IfcWallType -Typobjekt	Eigenschaftswert im <i>Fall 1.3.1 A sowie B</i> : 330 Außenwände/Vertikale Baukonstruktionen, außen
1.3.4.3	IfcBuildingElementPartType -Typobjekt	Eigenschaftswert im Fall 1.3.2.1 Ziffer 1 & 1.3.2.2 Ziffer 1 und 2: 335 Außenwandbekleidungen, außen
1.3.4.4	IfcBuildingElementPartType -Typobjekt	Eigenschaftswert im Fall 1.3.2.1 Ziffer 3 & 1.3.2.2 Ziffer 4: 336 Außenwandbekleidungen, innen

1.3.4.5	IfcBuildingElementPartType-Typobjekt	Eigenschaftswert im <i>Fall 1.3.2.2 Ziffer 3</i> : 331 Tragende Außenwände
1.3.4.6	IfcElementAssemblyType-Typobjekt	Eigenschaftswert im <i>Fall 1.3.2.1 Ziffer 2</i> : 337 Elementierte Außenwandkonstruktionen
1.3.5	Assoziierte Materialinformation über Typobjekt	Sind alle Materialinformationen gemäß Vorgabe durch die entsprechenden Typobjekte angegeben (keine direkten Angaben in Bauteil-, Komponente bzw. Schichtinstanz)?
1.3.5.1	Verweis Materialobjekt an Typobjekt im Fall 1.3.2	Hat die Kollektion im Attribut HasAssociations genau einen Eintrag, der von der Klasse IfcRelAssociatesMaterial ableitet?
1.3.5.2	IfcRelAssociatesMaterial-Objekt	Ist das Assoziationsende RelatingMaterial als Verweisobjekt mit einem Attribut xsi:type ausgeprägt, das den Wert IfcMaterial hat und verweist ein Attribut href auf ein Attribut id eines entsprechenden IfcMaterial -Objekts?

6.2.2.4 Energetische (Basis-) Werte

Auf Basis der im letzten Abschnitt im Pflichtenheft aufgestellten Anforderungen zur formalen und strukturellen Qualität, werden in der Tabelle 16 und Tabelle 17 die energierelevanten Parameter bezüglich ihrer Vollständigkeit in den einzelnen Anforderungen zusammengestellt. Dabei werden alle für die darauf aufbauende Berechnung des die (bau-)fachlichen Zielstellung repräsentierenden U-Wertes des Wandbauteils überprüft. Darüber hinaus verwenden die zusammengestellten Anforderungen die aus dem Planungsmodell stammenden, zuvor auf Vollständigkeit geprüften Parameter im Rahmen normbasierter Verfahren zur Berechnung des Ist-Wertes für den Planungsgegenstand.

Tabelle 16 Pflichtenheft Energetische (Basis-) Werte

IDX	Gegenstand	ANFORDERUNG
1.4.1	Materialinformationen	Enthalten die ausgezeichneten Materialien die vorgegebenen (energierelevanten) Informationen?

1.4.1.1	Verwiesene Ifc-Material-Objekte im Fall 1.3.5.2	Besteht die Kollektion im Attribut HasProperties aus (mindestens) zwei IfcMaterialProperties -Objekten?
1.4.1.2	IfcMaterialProperties-Objekt	Hat das Attribut Name den Wert A) Pset_MaterialCommon ? B) Pset_MaterialEnergy ? Existieren im Attribut Properties jeweils IfcSinglePropertyValue -Objekte?
1.4.1.3	IfcSinglePropertyValue-Objekt	Hat das Attribut Name im Fall 1.4.2 A) den Wert SpecificHeatCapacity bzw. im Fall 1.4.2 B) den Wert SpecificHeatTemperatureDerivative

Einschränkungen bei exemplifizierter Prüfung geometrischer Beschreibung

Aus der Vielzahl möglicher IFC Geometriebeschreibungen werden dabei nur geprüft:

- a) der gängige (vereinfachte) B-REP – Boundary Representation Model – Verfahrenstyp mit der Klasse **IfcFacetedBrep** aus planaren Ebenen
- b) der innovative (parametrische) **MappedRepresentation** Verfahrenstyp mit der Klasse **IfcExtrudedAreaSolid**

Als logische Grundlage zur Identifizierung der Tiefe (Dicke) einer Bauteilkomponente bzw. Schicht wird dabei folgende Annahme zugrunde gelegt:

- a) Tiefe ist bei Schichten immer kleiner als die Höhe bzw. die Breite
- b) Tiefe ist beim Rahmen die Länge des Profils die bei Pfosten und Riegel übereinstimmt

Tabelle 17 Pflichtenheft Energetische (Basis-) Werte (Fortsetzung)

IDX	Gegenstand	ANFORDERUNG
1.4.2	Geometrische Abmaße	Sind alle für die energetische Berechnung benötigten Maßangaben aus dem modellierten Bauteil (bzw. dessen dekomponierten Aufbau) ablesbar?
1.4.2.1	IfcFacetedBrep -Geometrieobjekt	Ist der Wert des Attributs Outer ein IfcClosedShell -Objekt?
1.4.2.2	IfcClosedShell -Objekt	Befinden sich in der Kollektion CfsFaces mindestens 6 IfcFace -Objekte?
1.4.2.3	IfcFace -Objekt	A) Befinden sich in der Kollektion Bounds mindestens ein IfcFaceOuterBound -Objekt? B) Befindet sich (wenn das Flächen-Objekt Öffnungen enthält) in der Kollektion Bounds mindestens ein IfcFaceBound -Objekt?
1.4.2.4	IfcFaceOuterBound bzw. IfcFaceBound Objekte	Befinden sich Attribut Bound ein IfcPolyLoop -Objekt dessen Kollektion Polygon aus mindestens 3 IfcCartesianPoint -Objekten besteht? Im Fall 1.4.2.3 A) Welche drei Verbindungslängen zwischen jeweils zwei der beinhalteten IfcCartesianPoint -Objekten existieren? Berechnet nach: Zwei Punkte $P(p_1 p_2 p_3)$ und $Q(q_1 q_2 q_3)$ im dreidimensionalen Raum R^3 haben den Abstand $ \rightarrow PQ = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + (q_3-p_3)^2}$ (vgl. https://www.mathematik-oberstufe.de/vektoren/a/abstand-2p-in-r3.html) Plausible Zuordnung der drei ermittelten Längen: Kürzeste entspricht der Dicke/Tiefe
1.4.2.5	IfcClosedShell -Objekt Längenmaße	Kürzeste gemäß dem Fall 1.4.2.4 ermittelte Verbindungslänge entspricht der Dicke/Tiefe, die beiden anderen entsprechen der Länge bzw. Breite.
1.4.2.6	IfcClosedShell -Objekt Flächenmaß	Welche Fläche ergibt die Multiplikation der gemäß dem Fall 1.4.2.5 ermittelten Verbindungslängen für die Länge bzw. Breite?
1.4.2.7	IfcMappedItem -Geometrieobjekt	A) Verweist das im Attribut MappingSource angegebene Verweisobjekt mit seinem Attribut href

		<p>auf ein Attribut id eines IfcRepresentationMap-Objektes?</p> <p>B) Ist das Attribut MappingTarget als Instanzobjekt mit:</p> <p>B1) einem Attribut xsi:type, das den Wert IfcCartesianTransformationOperator3DnonUniform hat,</p> <p>B2) einem Attribut LocalOrigin, welches einen IfcCartesianPoint instanziiert,</p> <p>B3) einem Attribut Scale3 mit gültiger Wertausprägung, ausgeprägt?</p>
1.4.2.8	IfcRepresentationMap-Objekt	<p>Besitzt das Attribut MappingOrigin ein IfcAxis2Placement3D-Objekt, das im Attribut Location einen IfcCartesianPoint instanziiert?</p> <p>Ist das Attribut MappedRepresentation als Instanzobjekt mit einem Attribut xsi:type, das den Wert IfcShapeRepresentation hat, ausgeprägt?</p>
1.4.2.9	IfcExtrudedAreaSolid-Objekt	<p>A) Hat das Attribut Depth eine gültige Wertausprägung?</p> <p>B) Ist das Attribut SweptArea als Instanzobjekt mit einem Attribut xsi:type, das den Wert IfcRectangleProfileDef hat, und gültige Wertausprägungen in den Attributen XDim und YDim besitzt, ausgeprägt?</p> <p>C) Ist das Attribut Position ein IfcAxis2Placement3D-Objekt, das in seinem Attribut Location die Klasse IfcCartesianPoint sowie in den Attributen Axis und RefDirection jeweils eine Klasse IfcDirection instanziiert?</p> <p>D) Instanziiert das Attribut ExtrudedDirection eine IfcDirection Klasse?</p>
1.4.2.10	IfcMappedItem -Objekt Längenmaße	<p>A) Die Dicke eines Rahmens ist in den Profilmäßen entweder der im Fall 1.4.2.9 B bestimmte XDim oder YDim Attributwert, entscheidend hierfür ist der, im Vergleich zu einem weiteren IfcMappedItem-Objekt, das z.B. einen entsprechenden Pfos-</p>

		<p>ten zum abgebildeten Riegel der Rahmenkonstruktion darstellt, der dieselbe Wertausprägung aufweist.</p> <p>B) Die Breite eines Rahmenteils (Pfosten oder Riegel) ist jeweils das andere zum im Fall 1.4.2.10 A ermittelte Profilmaß.</p> <p>C) Die (parametrische) Länge ergibt sich aus dem mittels dem Transformationsfaktor als Wert des durch Fall 1.4.2.7 B3 bestimmten Attributs Scale3 und dem Wert des durch Fall 1.4.2.9 A bestimmten Attributs Depth</p>
1.4.2.11	IfcMappedItem-Objekt Flächenmaß	Die Fläche eines Pfostens bzw. Riegels ergibt sich aus der Multiplikation der im Fall 1.4.2.10 B bestimmten Breite und der im Fall 1.4.2.10 C bestimmten Länge.
1.4.2.12	IfcShapeRepresentation-Objekt	<p>Befindet sich in der Kollektion Items mindestens</p> <p>A) ein IfcFacetedBrep-Objekt?</p> <p>B) ein IfcMappedItem-Objekt?</p> <p>C) nur im Fall 1.4.2.8 ein IfcExtrudedAreaSolid-Objekt?</p>
1.4.2.13	Fläche IfcShapeRepresentation-Objekt	Was ergibt die Addition der Flächen gemäß dem Fall 1.4.2.6 bzw. 1.4.2.11 für jedes Objekt in der Kollektion Items ?
1.4.2.14	Geometrische Abmaße IfcBuildingElementPart -Objekt	Bestimmt durch das IfcShapeRepresentation -Objekt als Wertausprägung des Attributs Representation .
1.4.2.15	Geometrische Abmaße IfcElementAssembly -Objekt	Bestimmt durch die Einzelgeometrien der IfcMember - bzw. IfcBuildingElementPart -Objekte in der Kollektion IsDecomposedBy .
1.4.2.16	Geometrische Abmaße IfcMember -Objekt	Bestimmt durch das IfcShapeRepresentation -Objekt als Wertausprägung des Attributs Representation .
1.4.3	Basiskenngrößen	Sind für alle Schichten bzw. Komponenten des Wandbauteils die Wärmedurchlasswiderstände R vorhanden sowie an der innersten und äußersten

		(luftberührenden) Wandschicht Ri bzw. Ra bestimmt?
1.4.3.1	IfcWallElementedCase-Objekt	<p>Ist ein Eigenschaftssatz mit dem Namensattributwert Pset_WallCommon gemäß Fall 0.4 zugeordnet?</p> <p>A) Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert TotalThermalTransferResistance sowie einer gültigen Wertausprägung?</p> <p>B) Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert TotalThermalTransferResistanceUpperLimit sowie einer gültigen Wertausprägung sowie eine Eigenschaft mit dem Namensattributwert TotalThermalTransferResistanceLowerLimit sowie einer gültigen Wertausprägung?</p>
1.4.3.2	IfcBuildingElementPart-Objekt	<p>A) Ist ein Eigenschaftssatz mit dem Namensattributwert Pset_ElementPartCommon gemäß Fall 0.4 zugeordnet?</p> <p>Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert TotalThermalTransferResistance sowie einer gültigen Wertausprägung?</p> <p>B) Wenn das Attribut Name des Objekts in seinem Wert den Begriff „INNEN“ enthält:</p> <p>Ist ein Eigenschaftssatz mit dem Namensattributwert Pset_ElementPartTypeCommon gemäß Fall 0.4 (in Verbindung mit Fall 0.3) über ein referenziertes Typobjekt zugeordnet?</p> <p>Existiert in diesem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert ThermalTransferResistanceDesignValueRsi sowie einer gültigen Wertausprägung?</p> <p>C) Wenn das Attribut Name des Objekts in seinem Wert den Begriff „AUSSEN“ enthält:</p>

		<p>Ist ein Eigenschaftssatz mit dem Namensattributwert <code>Pset_ElementPartTypeCommon</code> gemäß Fall 0.4 (in Verbindung mit Fall 0.3) über ein referenziertes Typobjekt zugeordnet?</p> <p>Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert <code>ThermalTransferResistanceDesignValueRse</code> sowie einer gültigen Wertausprägung?</p>
1.4.3.3	IfcMember-Objekt	<p>A) Ist ein Eigenschaftssatz mit dem Namensattributwert <code>Pset_MemberCommon</code> gemäß Fall 0.4 zugeordnet?</p> <p>B) Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert <code>TotalThermalTransferResistance</code> sowie einer gültigen Wertausprägung?</p>
1.4.4	Koeffizienten	Sind für alle Schichten bzw. Komponenten die Wärmedurchgangskoeffizienten (U-Werte) sowie der für das Gesamtbauteil Kulminierte bestimmt?
1.4.4.1	IfcWallElementedCase-Objekt	<p>Ist ein Eigenschaftssatz mit dem Namensattributwert <code>Pset_WallCommon</code> gemäß Fall 0.4 zugeordnet?</p> <p>Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert <code>ThermalTransmittance</code> sowie einer gültigen Wertausprägung?</p>
1.4.4.2	IfcBuildingElementPart-Objekt	<p>Ist ein Eigenschaftssatz mit dem Namensattributwert <code>Pset_ElementPartCommon</code> gemäß Fall 0.4 zugeordnet?</p> <p>Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert <code>ThermalTransmittance</code> sowie einer gültigen Wertausprägung?</p>
1.4.4.3	IfcMember-Objekt	Ist ein Eigenschaftssatz mit dem Namensattributwert <code>Pset_MemberCommon</code> gemäß Fall 0.4 zugeordnet?

		Existiert in dem zugeordneten Eigenschaftssatz gemäß Fall 0.4 eine Eigenschaft mit dem Namensattributwert ThermalTransmittance sowie einer gültigen Wertausprägung?
--	--	--

6.2.2.5 Analysen zu energetischer Zielstellung

In diesem Abschnitt des Pflichtenhefts werden die Anforderungen zusammengestellt die im Rahmen des Analyseplans die eigentliche (bau-) fachliche Fragestellung an den Planungsgegenstand mit dem vorgegebenen Zielwert abgleicht. In Tabelle 18 wird die entsprechende Anforderung mit den Verweisen zu den Anforderungen auf dem der Abgleich aufbaut dargestellt.

Tabelle 18 Pflichtenheft Analysen zu energetischer Zielstellung

IDX	Gegenstand	ANFORDERUNG
1.5.1	SOLL-IST Abgleich	Entspricht (oder unterschreitet) der Wärmedurchgangskoeffizient des Wandbauteils den durch die energetische Zielstellung Vorgegebenen?
1.5.1	Vergleichsoperation	Fall 1.4.1.1 (U-Wert) <= Dem in Fall 1.2.6.5 spezifizierten Eintrag in der durch Fall 1.2.6.6 verwiesenen (externen) Zielwertstabelle.

6.2.3 Regelbasis „Planungssituation“

In den im Weiteren dargestellten Tabellen des Pflichtenhefts – Tabelle 19, Tabelle 20 und Tabelle 21 – werden die situationsspezifisch ausformulierten Anforderungen zusammengestellt.

0.2.0 Lastenheft Situationsspezifische Modellprüfung und -analysen

Die Anforderung der vorhergehenden Abschnitten des Pflichtenhefts bilden die Mustervorlage für die folgenden, auf die beiden Planungsalternativen der in Kapitel 4.2 beschriebenen planerischen Entscheidungssituation angepassten Anforderungen. Im Wesentlichen sind dabei die Anforderungen konkretisiert,

die zur Bestimmung des als Entscheidungskriterium angesetzten U-Werte dienen.

6.2.3.1 Modellierungsqualität Wandbauteil

Für die spezifischen IFC Elemente der in den beiden Planungsalternativen variierten Wandbauteile – Holzrahmen- sowie Massivbauweise – werden in diesem Abschnitt die formale und strukturelle Festlegung in den konkretisierten Anforderungen als spezifischer Prüfplan abgebildet. In Tabelle 19 werden dabei jeweils der Bezug zu den bereits allgemein zu den zu prüfenden Modellqualitäten aufgestellten Anforderungen dargestellt, auf denen sie aufbauen (vgl. Pflichtenheftabschnitte zu Bürostandard in Kap. 6.2.1 und Planungsprozess in Kap. 6.2.2).

Tabelle 19 Pflichtenheft Modellierungsqualität Wandbauteil

IDX	Gegenstand	ANFORDERUNG
2.1.1	Grundlegende Modellierung	Prüfplan für beide Planungsalternativen gemäß der Abschnitte 1.0 – 1.2 zusammenstellen.
2.1.1.1	Entspricht Pflichtenheft für Planungsprojekt (vgl. Kap. 6.2.2)	
2.1.2	Modellierungsvorgaben zur Holzrahmenbauweise	Spezifischen Prüfplan für die Alternative Holzrahmenbauweise nach den Abschnitten 1.3 und 1.4 zusammenstellen.
2.1.2.1	Alternative Holzrahmenbauweise nach Abschnitt 1.3.0 <ul style="list-style-type: none"> ▪ 1.3.1 ▪ 1.3.2 	Spezifischer Prüfplan mit folgenden Ausprägungen: <ul style="list-style-type: none"> Entspricht die Wandbauteil-Typ Spezialisierung ELEMENTEDWALL? Verweist der Bauteil-Typ IfcWallType auf (innenliegende bzw. aussenliegende) Bekleidungsschichttypen IfcBuildingElementPartType sowie den zusammengesetzten Holzrahmentyp IfcElementAssemblyType?

	<ul style="list-style-type: none"> ▪ 1.3.3 ▪ 1.3.4 ▪ 1.3.5 <p>1.4.0</p> <ul style="list-style-type: none"> ▪ 1.4.1 	<p>Verweist der Holzrahmentyp auf die Komponententypen IfcMemberType für Pfosten- und Riegelprofile sowie auf den innenliegenden Dämmschichttyp IfcBuildingElementPartType?</p> <p>Verweist das Bauteil IfcWallElementedCase auf (innenliegende bzw. aussenliegende) Bekleidungsschichten IfcBuildingElementPart sowie den zusammengesetzten Holzrahmen IfcElementAssembly?</p> <p>Verweist der Holzrahmen auf die Komponenten IfcMember für Pfosten- und Riegelprofile sowie auf die innenliegende Dämmschicht IfcBuildingElementPart?</p> <p>Verweisen die Typobjekte IfcWallType bzw. IfcBuildingElementPartType sowie IfcElementAssembly auf entsprechende gemeinsame Eigenschaftsobjekte Pset_IfcWallCommon bzw. Pset_ElementComponentCommon?</p> <p>Verweisen allein die Typobjekte IfcWallType, IfcBuildingElementPartType und IfcMember auf entsprechende Materialinformationsobjekte IfcMaterial?</p> <p>Spezifischen Prüfplan für den Teil des Abschnitts 1.4.0 zur Existenzprüfung der Inputwerte für die energetische Berechnung fortschreiben:</p> <p>Verweisen die Materialobjekte IfcMaterial “Fichtenholz”, „Mineralfaser“, “Eternit” sowie “Rigips” auf Eigenschaftsobjekte</p>
--	--	---

	<ul style="list-style-type: none"> ▪ 1.4.2 	<p><code>Pset_MaterialCommon</code>, die entsprechende normbasierte (ISO 10456) Bemessungswert der Wärmeleitfähigkeit des Stoffes enthalten?</p> <p>Maßangaben <code>FLÄCHE</code> und <code>DICKE</code> aus dem Wandbauteil <code>IfcWallElementedCase</code> ablesbar? Maßangaben <code>FLÄCHE</code> und <code>DICKE</code> jeweils aus den <code>IfcBuildingElementPart</code> innenliegender und aussenliegender Wandbekleidungsschicht sowie Innendämmung des Holzrahmens ablesbar?</p> <p>Sind alle Einzelmaßangaben zu <code>DICKE</code>, <code>BREITE</code> und <code>LÄNGE</code> der Pfosten und Riegel <code>IfcMemberType</code> für die (Konstruktions-) <code>FLÄCHE</code> des Holzrahmens (ggf. in Form von Mengenobjekten <code>Qto_MemberBaseQuantities</code> aus parametrischer Geometriebeschreibung aggregiert) vorhanden?</p> <p>Können hierfür die Maßangaben <code>DICKE</code> und <code>BREITE</code> aus den Profilingaben der Komponententypen <code>IfcMemberType</code> des Holzrahmentyps sowie die Maßangabe zur tatsächliche <code>LÄNGE</code> von den Vorkommnissen der Pfosten und Riegel <code>IfcMember</code> im Holzrahmen abgelesen werden?</p>
2.1.3	Modellierungsvorgaben zur Massivbauweise	Spezifischen Prüfplan für die Alternative Massivbauweise nach den Abschnitten 1.3 und 1.4 zusammenstellen.
2.1.3.1	<p>Alternative Massivbauweise nach Abschnitt 1.3.0</p> <ul style="list-style-type: none"> ▪ 1.3.1 ▪ 1.3.2 	<p>Spezifischer Prüfplan mit folgenden Ausprägungen:</p> <p>Entspricht die Wandbauteil-Typ Spezialisierung <code>SOLIDWALL</code>?</p> <p>Verweist der Bauteil-Typ <code>IfcWallType</code> auf seine <code>IfcBuildingElementPartType</code> Teile (in-</p>

	<ul style="list-style-type: none"> ▪ 1.3.3 ▪ 1.3.4 ▪ 1.3.5 <p>1.4.0</p> <ul style="list-style-type: none"> ▪ 1.4.1 ▪ 1.4.2 	<p>nenliegende bzw. außenliegende) Bekleidungs-schichttypen, Dämmschichttyp und Porenbeton Mauerwerkstyp?</p> <p>Verweist das Bauteil <code>IfcWallElementedCase</code> auf die <code>IfcBuildingElementPart</code> Einzelteile (innenliegende bzw. außenliegende) Bekleidungs-schichten, Dämmschicht sowie das Porenbeton Mauerwerksobjekt?</p> <p>Verweisen die Typobjekte <code>IfcWallType</code>, <code>IfcBuildingElementPartType</code> sowie <code>IfcCoveringType</code> auf die entsprechenden gemeinsamen Eigenschaftsobjekte <code>Pset_IfcWallCommon</code>, <code>Pset_ElementComponentCommon</code> sowie <code>Pset_CoveringCommon</code>?</p> <p>Verweisen allein die Typobjekte <code>IfcWallType</code>, <code>IfcBuildingElementPartType</code> und <code>IfcMember</code> auf entsprechende Materialinformationsobjekte <code>IfcMaterial</code>?</p> <p>Spezifischen Prüfplan für den Teil des Abschnitts 1.4.0 zur Existenzprüfung der Inputwerte für die energetische Berechnung fort-schreiben:</p> <p>Verweisen die Materialobjekte <code>IfcMaterial</code> "Kalk", "Porenbeton", „Mineralfaser“ sowie "Gips" auf Eigenschaftsobjekte <code>Pset_MaterialCommon</code>, die entsprechende normbasierte (ISO 10456) Bemessungswerte der Wärmeleitfähigkeit des Stoffes enthalten?</p> <p>Maßangaben FLÄCHE und DICKE aus dem Wandbauteil <code>IfcWallElementedCase</code> ablesbar?</p>
--	---	--

		Maßangaben FLÄCHE und DICKE aus IfcBuildingElementPart für Porenbeton Mauerwerk, Innen-/Außenputz- sowie Dämmschicht ablesbar?
--	--	--

6.2.3.2 Ermittlung Energetischer Kennwerte

Gemäß der normbasierten Berechnungsvorgabe werden in den konkretisierten Anforderungen in diesem Teil des Pflichtenhefts zu den beiden Planungsalternativen die Ermittlung der U-Werte abgebildet. Alle hierfür benötigten Zwischenschritte werden dabei ausgehend von den zuvor auf Vollständigkeit überprüften Basiswerten in der Tabelle 20 zusammengestellt.

Tabelle 20 Pflichtenheft Ermittlung Energetischer Kennwerte

IDX	Gegenstand	ANFORDERUNG
2.2.1	Einzelbemessungswerte der Schichten und Komponenten	Bemessungswerte für die einzelnen Schichten gemäß Berechnungsvorschrift DIN EN ISO 6946 für die beiden Alternativen bestimmen.
2.2.1.1	Einzelbemessungswerte der Schichten und Komponenten	Bemessungswerte für die einzelnen Schichten gemäß Berechnungsvorschrift DIN EN ISO 6946
	Alternative Holzrahmenbauweise	Bemessungswert Wärmeübergangswiderstand R_{se} ? IfcBuildingElementPartType – Pset_ElementPartTypeCommon – [ThermalTransferResistanceDesignValueRse]
	1 Homogene Schicht Eternit:	Bemessungswert Wärmedurchlasswiderstand R_1 ? IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]
		A Konstruktionsfläche Pfosten/Riegel:

	<p>2 Inhomogene Schicht gedämmter Holzrahmen</p>	<p>Bemessungswert Wärmedurchlasswiderstand R_{2a}? IfcMember – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue] B Gedämmte Zwischenraumfläche: Wärmedurchlasswiderstand R_{2b}? IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]</p>
	<p>3 Homogene Schicht Rigips:</p>	<p>Bemessungswert Wärmedurchlasswiderstand R_3? IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]</p>
	<p>Alternative Massivbauweise</p>	<p>Bemessungswert Wärmeübergangswiderstand R_{si} IfcBuildingElementPartType – Pset_ElementPartTypeCommon – [ThermalTransferResistanceDesignValueRsi]</p>
	<p>Alternative Massivbauweise</p>	<p>Bemessungswert Wärmeübergangswiderstand R_{se}? IfcBuildingElementPartType – Pset_ElementPartTypeCommon – [ThermalTransferResistanceDesignValueRse]</p>
	<p>1 Homogene Außenputzschicht</p>	<p>Bemessungswert Wärmedurchlasswiderstand R_1? IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]</p>
	<p>2 Homogene Dämmschicht</p>	<p>Bemessungswert Wärmedurchlasswiderstand R_2?</p>

	<p>3 Homogene Porenbetonschicht</p> <p>4 Homogene Innenputzschicht</p>	<p>IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]</p> <p>Bemessungswert Wärmedurchlasswiderstand R_3?</p> <p>IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]</p> <p>Bemessungswert Wärmedurchlasswiderstand R_4?</p> <p>IfcBuildingElementPart – Pset_ElementPartCommon – [ThermalTransferResistanceDesignValue]</p> <p>Bemessungswert Wärmeübergangswiderstand R_{si}</p> <p>IfcBuildingElementPartType – Pset_ElementPartTypeCommon – [ThermalTransferResistanceDesignValueRsi]</p>
2.2.2	Ermittlung des Wärmedurchgangswiderstands	Wärmedurchgangswiderstand der Wandkonstruktion gemäß Berechnungsvorschrift DIN EN ISO 6946 für die beiden Alternativen bestimmen.
2.2.2.1	Ermittlung des Wärmedurchgangswiderstands	Die Ergebnisse der folgenden Fragestellungen zum Wandbauteil <code>IfcWallElementedCase</code> werden in dessen Eigenschaftssatz „ <code>Pset_WallCommon</code> “ aufgeführt.
	Alternative Holzrahmenbauweise	Inhomogene Bauteilschichten
	DIN EN ISO 6946 6.2.3	Oberer Grenzwert Wärmedurchgangswiderstand R'_T ? TotalThermalTransferResistanceUpperLimit
	DIN EN ISO 6946 6.2.4	Unterer Grenzwert Wärmedurchgangswiderstand R''_T ?

	Alternative Massivbauweise	<p>TotalThermalTransferResistanceLowerLimit</p> <p>Wärmedurchgangswiderstand R_T? Formel: $R_T = R'_T + R''_T / 2$ TotalThermalTransferResistance</p> <p>Homogene Bauteilschichten</p> <p>Wärmedurchgangswiderstand R_T? Formel: $R_T = R_{se} + R_1 + R_2 + R_3 + R_{si}$ TotalThermalTransferResistance</p>
2.2.3	Energetischen Kennwert für Zielabgleich aufbereiten	Umrechnen des ermittelten Wärmedurchgangswiderstand der Wandkonstruktion gemäß Berechnungsvorschrift DIN EN ISO 6946 in den äquivalenten Wärmedurchgangskoeffizienten (U-Wert).
2.2.3.1	U-Wert der Wandkonstruktion	<p>Wärmedurchgangskoeffizient U_{wand}?</p> <p>Formel: $U_{wand} = 1/R_T$ IfcWallelementedCase – Pset_WallCommon – [ThermalTransmittance]</p>

6.2.3.3 Analyse Zielerreichung und Gegenüberstellung Planungsalternativen

In diesem letzten Abschnitt des Pflichtenhefts werden anhand „redaktioneller“ Zusammenstellungen, zum einen ein Abgleich der Planungsalternativen mit den geforderten Zielwerten (energetisches Entscheidungskriterium) sowie eine Gegenüberstellung der beiden gemäß dem Soll-Wert bewerteten Alternativen dargestellt.

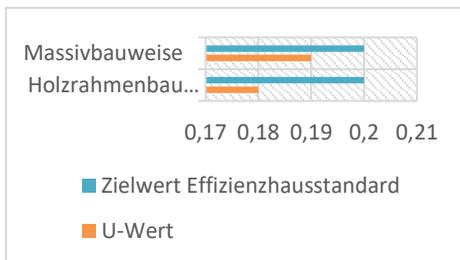
Tabelle 21 Pflichtenheft Analyse Zielerreichung

IDX	Gegenstand	ANFORDERUNG
2.3.1	SOLL-IST Vergleich Alternative Holzrahmenbauweise	Entspricht bzw. unterschreitet der berechnete U-Werte der Wandkonstruktion dem Geforderten für eine Außenwand gemäß SOLL-Werteliste (Tabelle 6, Zeile 3)?

2.3.1.1	Der in 2.2.3.1 ermittelte Wandbauteil U-Wert	Der ermittelte Wert der Planungsalternative wird mit dem vorgegebenen abgeglichen; IST: IfcWallElementedCase – Pset_WallCommon – [ThermalTransmittance] SOLL: Außenwände, Geschossdecken nach unten gegen Außenluft (vgl. Spalte 3 in Tabelle 6) $U \leq 0,20 \text{ W}/(\text{m}^2 \text{ K})$
2.3.2	SOLL-IST Vergleich Alternative Massivbauweise	Entspricht bzw. unterschreitet der berechnete U-Werte der Wandkonstruktion dem Geforderten für eine Außenwand gemäß SOLL-Werteliste (Tabelle 6, Zeile 3)?
2.3.2.1	Der in 2.2.3.1 ermittelte Wandbauteil U-Wert	Der ermittelte Wert der Planungsalternative wird mit dem vorgegebenen abgeglichen; IST: IfcWallElementedCase – Pset_WallCommon – [ThermalTransmittance] SOLL: Außenwände, Geschossdecken nach unten gegen Außenluft (vgl. Spalte 3 in Tabelle 6) $U \leq 0,20 \text{ W}/(\text{m}^2 \text{ K})$

Pflichtenheft Gegenüberstellung Planungsalternativen

Unter dem Punkt 2.4 werden, wenn alle vorherigen Prüfschritte erfolgreich



sind, im Lastenheft zur finalen Zusammenstellung der Entscheidungsgrundlagen eine Zusammenstellung der beiden erstellten Zielwertabgleiche zu

Abbildung 58 Beispielhafte Entscheidungsgrundlage

den Planungsalternativen (siehe Tabelle 21 Idx 2.3.1.1 und 2.3.1.2) gefordert. Abbildung 58 zeigt hierzu beispielhaft, wie die beiden U-Werte der Planungsalternativen zusammen mit dem Zielwert aus der Tabelle 6 grafisch aufbereitet werden können.

7 Beispielhafte Anwendung und Evaluierung der erstellten Softwarelösung

In diesem Kapitel werden alle Ansichten der Benutzeroberfläche mit der jeweiligen Hauptfunktionalität erläutert. Dazu werden die Nutzer*inneninteraktionen der einzelnen Werkzeuge der konfigurierbaren Softwareanwendungen jeweils beispielhaft von den in Kapitel 4 bestimmten Rollen des szenariobasierten Planungsprozesses (Entscheidung bzw. Regeladministration und -entwicklung) angewendet (Kap. 7.1). Die jeweiligen Vorgänge und Arbeitsschritte (vgl. als User Story die methodische Beschreibungen in Kap. 5) zur Administration einer Regelbasis (vgl. Kap. 6.1), Entwicklung von Regeln (vgl. Kap. 6.2) und schließlich ihre Anwendung zur Entscheidungsunterstützung (vgl. 4.2) werden dabei anhand von Werkzeugoberflächen in beiden Modi der Softwareanwendungen demonstriert.

Mit damit aufgezeigten, exemplifizierten Ständen in der Benutzeroberfläche zu Einzelarbeitsschritten, wie dem Prüfen eines Modells anhand ausgewählter Regeln, dem Anlegen eines Regelsatzes usw., können die szenariobasiert beschriebenen Bearbeitungsvorgänge der Anwender*innen nachvollzogen werden.

Der im Anwendungsszenario (Kap. 4) aufgegriffene und zur Evaluation des Gesamtansatzes der Arbeit diesbezüglich mit Umsetzung sowie Anpassung auf den Lösungsansatz exemplifizierte Bezug zur planerischen Praxis konnte im Rahmen der Arbeit nur einen kleinen Ausschnitt eines weitspannenden thematischen Kontextes integrale Planung aufzeigen. Zudem erforderte die technische Umsetzung eines Prototyps ein weiteres szenariobasiertes Abgrenzen des Kontexts und somit einen relativ engen Fokus auf dessen Anwendung, hierbei wären weitere Sichten für ein evaluiertes Ergebnis der Arbeit hilfreich. Daher werden andere Anwendungskontexte aus verschiedenen Forschungsprojekten vorgestellt, in deren Rahmen der Lösungsansatz bereits prototypisch angepasst und evaluiert werden konnte. Dabei wird auf jeweilige Besonderheiten beim

Ausprägen benötigter Prüf- und Analyselogik für deren Anwendungskontexte, sowie bezüglich spezifischer konzeptioneller Anpassungen auf Ebene des Softwareansatzes eingegangen. Jeweils verfolgte praxisnahe Zielstellungen in den Forschungsprojekten erlauben eine weitaus fundiertere, breitere Aufbereitung praxisrelevanter Inhalte als Grundlage der Evaluierung des Lösungsansatzes der Arbeit (Kap. 7.2).

Ein weiterer Anwendungskontext der Ökobilanzierung und der Bewertung von Bauwerken ist noch Gegenstand laufender Forschungsvorhaben. Darin werden Planungsgegenstände in rahmenden Prozessen der integralen Planung im Hinblick auf Fragestellungen der Ökobilanz untersucht, wobei eine Aufbereitung des Lösungsansatzes dieser Arbeit auf der datentechnischen Ebene angedacht ist. Ein Ausblick auf diese weiteren Anwendungsmöglichkeiten wird zum Abschluss der Arbeit dann im Kapitel 8.1 gegeben.

7.1 Bedienung der Benutzeroberflächen

Alle Ansichten der Softwareanwendung werden jeweils mit den darin bereitgestellten Schaltflächen und Funktionen erläutert. Dabei werden sowohl die möglichen Nutzer*inneninteraktionen der Software im Standard- als auch im erweiterten Administrations- und Bearbeitungsmodus dargestellt. Zunächst werden die gemeinsamen Funktionen nach dem Starten der Softwareanwendung betrachtet.

Übersicht und Menus der im Hauptfenster der Benutzeroberfläche

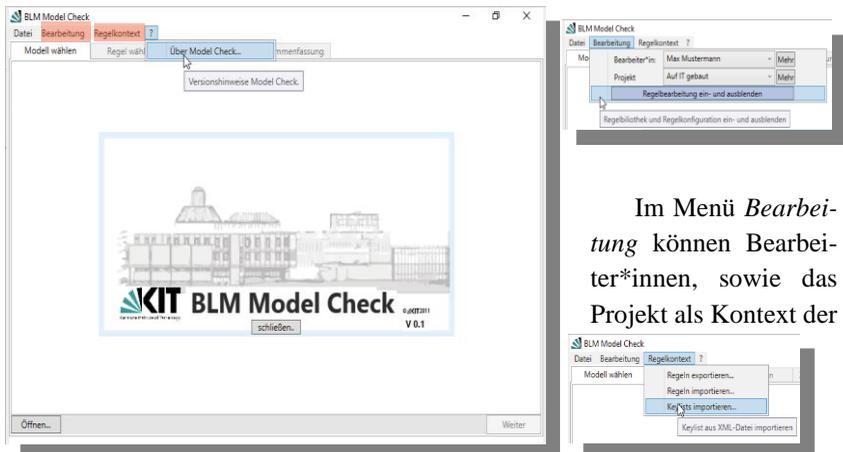


Abbildung 59 Menüs und Informationsdialog im Hauptfenster des BLM Model Check

Prüfung bzw. Analyse spezifiziert werden. Diese Informationen können dann nach dem Durchlauf einer Modellprüfung im entsprechenden Prüfbericht (s.u.) ausgegeben werden. Zudem kann bei dem erweiterten Modus der Softwareanwendung durch die Schaltfläche *Regelbearbeitung ein- und ausblenden* zwischen dem Standardmodus und dem zur Administration und Bearbeitung der Regeln gewechselt werden.

Im Menü *Regelkontext* können die für verschiedene Anwendungskontexte in Form von Containern (vgl. 5.2.2.4) gehaltenen Regelbasen (vgl. 6.1) für beabsichtigte Prüfungen/Analysen eines Modells bzw. (Weiter-)Entwicklungen und Bearbeitungen der Regeln in die Softwareanwendung im- und exportiert werden. Darüber hinaus können mit dem Menüpunkt *Keylists importieren* für die Prüfregeln ggf. benötigte Vorgaben bestimmt werden, beispielsweise eine Liste an vorgegebenen Bauelementnamen, mit einer entsprechenden Regel für die Elemente im BIM-Modell überprüft werden.

BLM ModelCheck Standard Modus – in vier geführten Schritten zur Modellprüfung bzw. -analyse

Die folgenden vier Abbildungen zeigen die in einem Registerkarten Ansichtselement (engl. tabview) des Hauptfensters organisierte Benutzeroberflächen einzelner in vorgegebener Reihenfolge stehender Werkzeuge der Software. Zum Freischalten des nächsten Reiters (engl. tab) wird dabei zunächst der Arbeitsschritt im Aktuellen fertiggestellt. Zu den einmal freigeschalteten Reitern kann jederzeit zurückgewechselt werden, um entsprechend den Stand des Arbeitsschritts einzusehen bzw. zu ändern. Falls von ermöglichten Änderungen im vorherigen Arbeitsschritt der Arbeitsstand des Aktuellen, beispielsweise ein durch die neugewählte Regel obsolet gemachter Prüfbericht zur vormals Gewählten, wird dies in einem Dialogfeld benachrichtigt und konsistent gehalten. Ein dadurch die Anwender*innen führender Programmablauf beginnt beim Laden eines Modells zur designierten regelbasierten Überprüfung bzw. Analyse über die *Öffnen* Schaltfläche bzw. das Menü *Datei* (vgl. Abbildung 59).

Modellübersichtswerkzeug (Reiter MODELL WÄHLEN) – Nach vollendetem Ladevorgang des Modells werden den Anwender*innen in aufklappbaren Informationsbereichen (vgl. Schaltflächen *Mehr* im oberen Werkzeugbereich in Abbildung 60) zu den Metadaten des Modelldokuments sowie einer konfigurierbaren Zusammenstellung von Modellinhalten, wie den Namen von IFC Projekt-, Gebäudeobjekt oder den Geschossen, eine Übersicht zum geladenen Produktmodell präsentiert. Im Hauptfeld des Werkzeuges wird beim Laden eines IFC Modells die RDF Viewer Komponente (QUELLE RDF) eingebunden, mit der den Anwender*innen eine interaktive Baum- sowie Geometriedarstellung präsentiert wird.

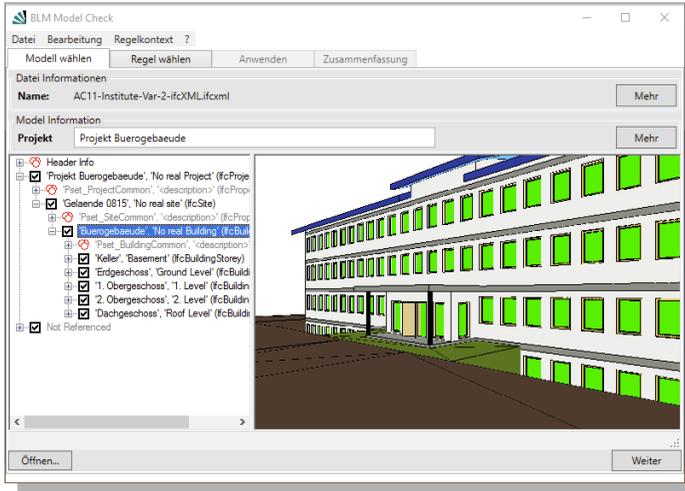


Abbildung 60 Modellübersichtswerkzeug - Informationen zum geladenen Modell

Regelauswahlwerkzeug (Reiter REGEL BIBLIOTHEK) – Dieses Werkzeug ermöglicht den Anwender*innen die Auswahl der Regeln, die sie zur Prüfung bzw. Analyse auf das zuvor geladene Modell anwenden möchten. Durch das oben beschriebene Importieren eines Regelkontextes werden die in der geladenen Regelbasis befindlichen Regelsätze und beinhaltenen Regeln in der Regelbibliothek auf der rechten Seite angezeigt. Im unteren Bereich dieses Bibliotheksfensters, sowie im gegenüberliegenden Auswahlfenster befinden sich jeweils ein Informationsbereich, der die Beschreibungstexte des aktuell selektierten Regelsatzes bzw. der (durch dessen Aufklappen darin befindlichen) selektierten Regel anzeigt. Die Anwender*innen können, nachdem in der *Bibliothek* ein zur Anwendung auszuwählender Regelsatz selektiert wurde, diesen sowohl durch Bedienen der *Auswahl*-Schaltfläche in der unteren Leiste des Werkzeugs als auch über „Drag and Drop“ dem Auswahlfenster hinzufügen.

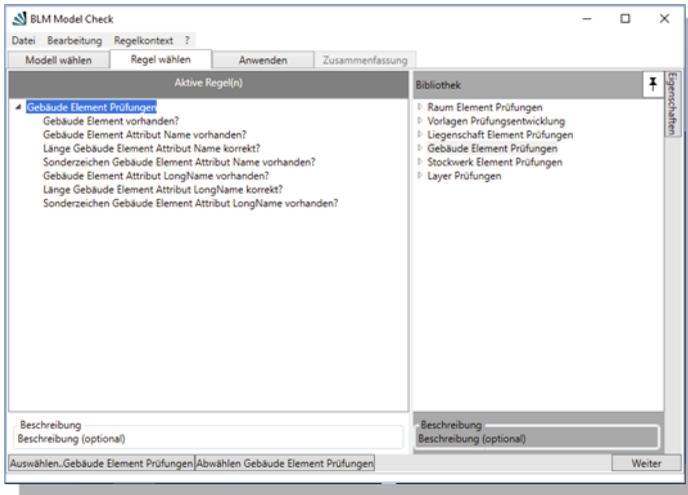


Abbildung 61 Regelauswahlwerkzeug

Regelausführungswerkzeug (Reiter ANWENDEN) – Nachdem sowohl ein Modell zur Prüfung bzw. Analyse geladen ist als auch die entsprechende(n) Regel(n) ausgewählt sind, wird die Anwendung der Regel auf das Modell in diesem Werkzeug ermöglicht. Als Zusammenfassung der aktuellen Auswahl befinden sich im oberen Bereich einerseits das bereits im Modellübersichtswerkzeug dargestellte Übersichtsfeld zu den Modellinformationen. Darüber werden in einer Listenansicht die ausgewählten Regelsätze angezeigt. Beim regelbasierten Prüfen eines Modells können sich durch nicht vorheriges überprüfen, ob das Element mit dem zu überprüfenden Sachverhalt überhaupt existiert, erwartbare „Folgefehler“ ergeben. Diese von der Programmlogik konventionsbedingt abgefangenen ggf. nicht durch die Regel mit einer aussagekräftigen Ereignismeldung bedachten Ereignisausgaben können mit der Auswahl Schaltfläche *Folgefehler nicht ausgeben* unterdrückt werden. Mit dem Bedienen der Schaltfläche *Ausführen* wird der Prüfvorgang ausgelöst. Im zentralen Bereich des Fensters werden in einem Konsolenfenster die einzelnen Ereignismeldungen, durch zusammenfassende Zustandsmeldungen (beispielsweise wieviel Ereignisse die aktuelle Regelprüfung insgesamt ergab, usw.) ergänzt, fortlaufend (chronologisch) den Anwender*innen präsentiert, so dass

diese über den aktuellen Stand des Ablaufs der regelbasierten Prüfung bzw. Analyse informiert werden.

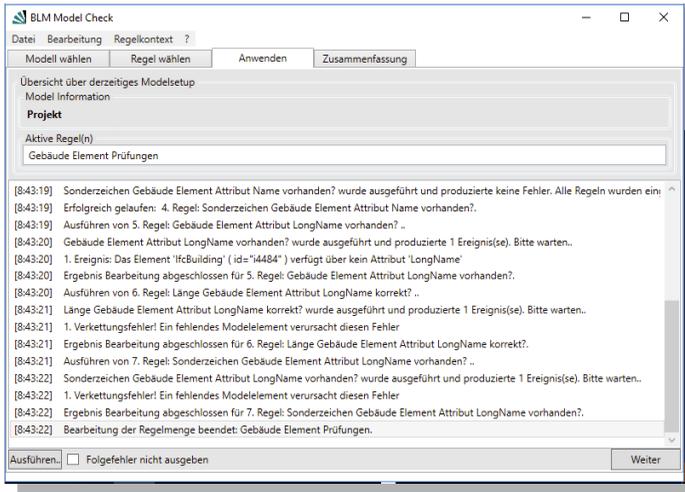


Abbildung 62 Regelausführungswerkzeug

Berichtswerkzeug (Reiter ZUSAMMENFASSEN) – Nach erfolgreich durchlaufenen Modellprüfung bzw. -analyse werden die Ereignisse den Anwender*innen in strukturierter Weise in einer verschachtelten Listendarstellung angezeigt (vgl. Abbildung 63). Dabei können zu jedem durchlaufenen Regelsatz die beinhaltenden Regeln wie in der *Bibliothek* durch Aufklappen dargestellt werden. Zu einzelnen Regeln können dann durch den gleichen Aufklappmechanismus die entsprechenden Ereignismeldungen eingesehen werden. Dieser Zusammenfassung Ereignisliste des Prüfungsvorganges steht in jeder Zeile eine Kommentierungsmöglichkeit zur Verfügung. Neben den generellen, über das Menü anzugebenden Informationen zu Prüf- bzw. Analyseprojekt sowie Bearbeiter*in und Organisation, werden diese Einzelkommentare den Ereignissen zugeordnet und im tabellarisch strukturierten Prüfbericht ausgegeben, der exportiert werden kann.

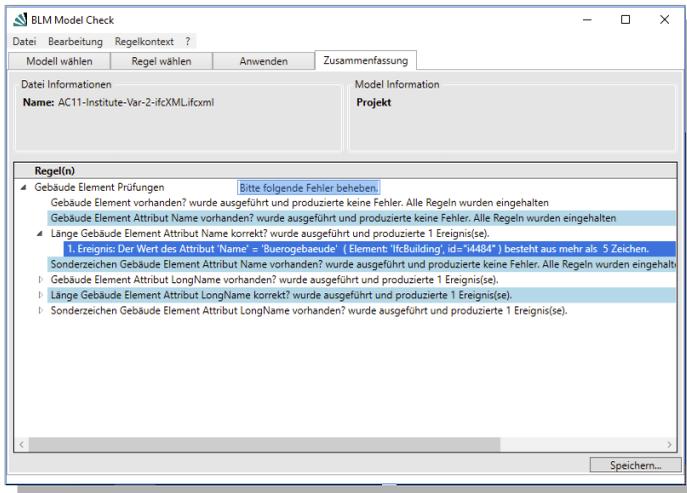


Abbildung 63 Berichtswerkzeug

Durch die bisher beschriebenen Schritte ermöglicht der Model Check Standardmodus den Endanwender*innen das eigene Auswählen von Regeln und dementsprechendes regelbasiertes Prüfen- bzw. Analysieren von IFC-basierenden Produktmodellen. Das eigenständige Erstellen z.B. von Entscheidungsgrundlagen auf Basis einer Analyse wird dabei ebenfalls durch die in den Regeln natürlichsprachlich vorgehaltenen Ereignismeldungen unterstützt. Diese können zusammenfassend für alle auftretenden Ereignisse eines Prüfungsvorganges in einen aussagekräftigen Ergebnisbericht überführt und (bereits ergänzend kommentiert) in ein Tabellendokument exportiert werden. Das entsprechende Bedienen der Software erfordert keine weiteren Kenntnisse, vorausgesetzt eine problemorientierte Regelbasis mit zweckgebunden bezeichneten Regeln liegt dabei vor. Damit konnte für die im Projektszenario angesetzte Rolle der Entscheider*innen gemäß dem Lösungssatz ein Hilfsmittel umgesetzt werden, das sie (auch) als Nicht-Expertenwerkzeug in ihren planerischen Aufgaben unterstützen kann. Indem im Vorhinein mit dem Umsetzen Prüf- und Analyseauftrag durch versierte Anwender*innen im erweiterten Softwaremodus (s.u.) entsprechende Regelbasis erstellt werden, können sich

die Architekt*innen bezugnehmend zu dem konkreten szenariobasierten Anwendungsfall (vgl. Kap. 4.2) damit selbständig ihre (retrospektiv dokumentierten) Entscheidungsgrundlagen schaffen.

BLM ModelCheck zuschaltbarer Modus Regeladministration und -bearbeitung – Aufbereiten eines Prüf- bzw. Analysekontextes

Im BLM Model Check sind standardmäßig dabei keine Regeln hinterlegt. Das bedeutet, dass beim initialen Starten nach der Neuinstallation (beider Modi) der Softwareanwendung allein im Modellübersichtswerkzeug das Laden sowie übersichtliche Informationen zu Produktmodellen ermöglicht werden. Durch fehlende Regeln können ohne das Laden eines Regelkontextes (Import Regelbasis, s.o.) im Standardmodus keine Modellprüfungen bzw. -analysen durchgeführt werden. Allein das Umschalten in den Modus der Regeladministration und -bearbeitung (s.o. Menu *Bearbeiten*) in der erweiterten Softwareanwendung ermöglicht den Anwender*innen neben diesem Einlesen externer Regelkontexte durch ein erweitertes bzw. ein hinzu geschaltetes Werkzeug auch eine neue Regel(basis) zu erstellen.

Administrationswerkzeug (erweiterter Reiter REGEL BIBLIOTHEK) – Das Umschalten in den Modus der Regeladministration und -bearbeitung erweitert das Regelauswahlwerkzeug des Standardmodus zu dem Administrationswerkzeug für das Organisieren des Regelbestandes. Hierzu wird auf der linken Seite der Oberfläche eine weitere Regelbibliothek *Regel-Entwicklung* ergänzt. Prinzipiell können nur Regeln in dieser Entwicklungsbibliothek verwaltet werden. Sowohl das Erstellen einer neuen Regel und/oder eines neuen Regelsatzes, sowie sämtliche Bearbeitungsfunktionalitäten werden ausschließlich mit dem im folgenden Abschnitt dargestellten Regelbearbeitungswerkzeug ermöglicht.

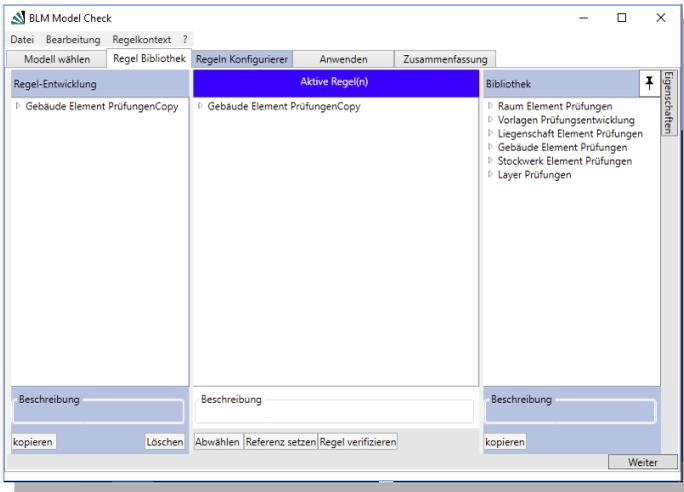


Abbildung 64 Administrationswerkzeug

In den mittleren Bereich *Aktive Regeln* des Werkzeugs können wie im Regelauswahlwerkzeug die für eine Modellprüfung ausgewählten, nun aktuell zu bearbeitenden Regelsätze aus den Produktiven der *Bibliothek* aus der rechten Seite per Drag and Drop zusammengestellt werden. Dasselbe gilt für die Regelbibliothek *Regel-Entwicklung* auf der linken Seite. Da die sich in der *Bibliothek* befindlichen produktiven Regelsätze bzw. Regeln (zur Konsistenzwahrung der Regelbasis) nicht mehr bearbeitet werden können, wird bei ihrem Auswählen zur Bearbeitung im mittleren Bereich *Aktive Regeln* direkt eine Kopie des Regelsatzes erstellt. Die Kopie wird dabei in der Entwicklungsbibliothek angelegt. Alle beinhalteten Regeln des Satzes referenzieren in diesem Falle noch die produktiven Regeln in der *Bibliothek*. Äquivalent wird eine Regelkopie automatisch angelegt, wenn eine der Regeln dann tatsächlich bearbeitet wird (vgl. nächsten Abschnitt). Alternativ zum Drag and Drop können zum Anlegen einer Kopie von Regelsätzen, nachdem diese in sowohl *Bibliothek* als auch *Regel-Entwicklung* selektiert wurden, die entsprechenden Schaltfläche *Kopieren* bedient werden.

Da jede Regel immer mindestens in einem Regelsatz referenziert sein muss, lassen sich mit folgenden Funktionen beliebige Kombinationen der Regeln in bedarfsorientierte Regelsätze zusammenstellen. Für alle Funktionen zum Administrieren der Regeln im Rahmen von Regelsätzen werden entsprechende Schaltflächen jeweils in den unteren Leisten der drei Bereiche des Werkzeugs angezeigt. Die Bearbeitungsfunktionen beziehen sich dabei immer auf einen im mittleren Bereich *Aktive Regeln* mittels Selektion (hervorgehoben angezeigte) aktuell zur Bearbeitung ausgewählten Regelsatz. Regeln in Regelsätzen der produktiven *Bibliothek* sowie der Regel-Entwicklung, die diesem zugeordnet werden sollen, können, nachdem sie selektiert wurden, durch Bedienen der Schaltflächen *Referenz Setzen* dem aktuell bearbeiteten Regelsatz hinzugefügt werden. Damit können ohne das Ändern der eigentlichen Regelinhalte, unter Wiederverwendung bereits vorhandener Regeln allein durch Reorganisation und Zusammenstellen der Regeln in (verschiedenen) bedarfsgerechten Regelsätzen eine Anpassung für spezifische Prüfkontexte erfolgen. Über das Setzen von Referenzen hinaus, kann vorbereitend zur Bearbeitung einzelner (bestehender) Regeln und beziehungsweise zum oben genannten Anlegen einer Regelsatzkopie ebenfalls durch die Selektion einer Regel in Sätzen der *Bibliothek* bzw. *Regel-Entwicklung* durch ein Bedienen der Schaltfläche *Kopieren* anstelle der Referenz eine Kopie der Regel angelegt werden.

Eine Überführung von Regeln respektive Regelsätzen von der *Regel-Entwicklung* in die produktive *Bibliothek* ermöglicht eine Verifikationsprozedur, wodurch sie dann auch den Anwender*innen im Regelauswahlwerkzeug zur Verfügung gestellt werden können. Dazu ist mit den in der Software angesetzten Stapelverarbeitungsmöglichkeiten ein Hintergrundvorgang implementiert (siehe auch Read, Eval, Print & Loop – REPL im nächsten Abschnitt sowie vgl. Kap. 7.2.3), bei dem die zu verifizierende Regel durch Ausführen auf ein in den Nutzer*innendaten (vgl. Kap. 5.2.2.4) hinterlegtes Referenzmodell überprüft wird. Der Vorgang kann dabei für eine im Bereich *Aktive Regeln* selektierte Regel der *Regel-Entwicklung* durch Bedienen der Schaltfläche *Regel Verifizieren* ausgelöst werden und muss die Überführung in die produktive *Bibliothek* erfolgreich durchlaufen. Eine Verhaltensvariante dieser Schaltfläche, wenn Anwender*innen anstelle einer Regel einen Regelsatz im Bereich *Aktive Regeln* zur Verifikation selektieren, regelt die Konvention, wonach ein

Regelsatz erst in die Bibliothek überführt werden kann, wenn sämtliche beinhalteten Regeln verifiziert wurden. Dieser wichtige (durch das Einbringen eigener auf die Regel(satz)inhalte abgestimmte Referenzmodelle) Qualitätssicherungsmechanismus vor dem Bereitstellen der Regeln für die Endanwendung ermöglicht es dabei, auch bei der Entwicklung von Regeln entsprechende Testungs- und Freigabeschritte in der eigenen Arbeitsmethodik mit der Software anzusetzen.

Regelbearbeitungswerkzeug (Reiter REGEL KONFIGURIEREN) – Das Anlegen der Kopie einer Regel im oben beschriebenen Administrationswerkzeug löst ein automatisiertes Umschalten in dieses Werkzeug für deren Bearbeitung aus, wobei bereits Regelinhalte eingelesen sind und direkt bearbeitet werden können. Über das Karteiregistersteuerelement lässt sich andernfalls mit der Bedienung der Reiter-Schaltfläche *Regel konfigurieren* in das Regelbearbeitungswerkzeug (siehe Abbildung 65) wechseln. Auf der linken Seite befindet sich der Bereich *Aktive Regeln* mit den im (oben beschriebenen) Administrationswerkzeug auszuwählenden Regelsätzen.

Im bisher in den anderen Werkzeugen zur Anzeige dienenden Beschreibungsfeld im unteren Teil des linken Bereiches können für jeweils selektierte Regel

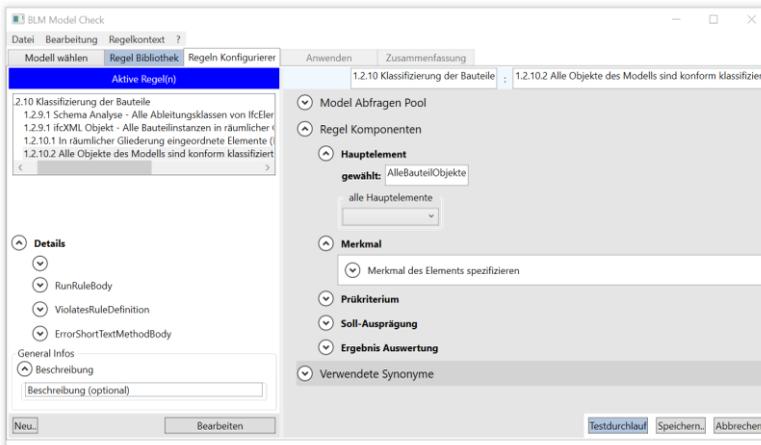


Abbildung 65 Regelbearbeitungswerkzeug

(bzw. -sätze), ohne ein weiteres Laden oder Einlesen der Regel in das Werkzeug, der Beschreibungstext bearbeitet und mit der Schaltfläche *Speichern* in der unteren Leiste der rechten Seite in der Regel(satz)datei persistiert werden.

Zur Bearbeitung der übrigen Regel(satz)inhalte können bereits bestehende Regeln im Bereich *Aktive Regeln* durch Selektion ausgewählt und mit der Schaltfläche *Bearbeiten* in die Bearbeitungsfelder des Werkzeugs zur Anpassung ihrer Inhalte eingelesen werden. Wenn dabei kein Regelsatz selektiert wurde, ermöglicht es die Schaltfläche *Neu* es eine neue Regel anzulegen, wobei direkt auch ein neuer Regelsatz, der sie referenziert, erstellt wird, da keine verwaisten Regeln existieren dürfen. Durch das Selektieren eines designierten Regelsatzes wird hingegen diesem durch Bedienen der Schaltfläche *Neu* eine neue Regel hinzugefügt.

Wie bereits zum Beschreibungsfeld von Regel(sätzen) erläutert, können durch Betätigen der Schaltfläche *Speichern* alle mit folgenden Bearbeitungsmöglichkeiten durchgeführten Änderungen der Regelinhalte im Werkzeug jeweils in die Regeldatei geschrieben werden. So können beispielsweise bei einer aktiv zur Bearbeitung ins Werkzeug geladenen Regel auf der rechten Seite in der Kopfleiste der Bearbeitungsfelder jeweils der angezeigte Bezeichner, sowie der entsprechende Regelsatzbezeichner angepasst werden. Der darunterliegende Bereich der inhaltlichen Bearbeitungsfelder gliedert sich in drei aufklappbare Abschnitte:

- *Modell Abfragen Pool* (Verwalten der POOL Einträge einer Regel) – oberster Abschnitt (vgl. Abbildung 66) mit der Schaltfläche *Neu* zum Anlegen neuer Einträge in der mittleren Spalte; als Kopiervorlagen stehen in der linken sowie rechten Spalte jeweils alle Einträge anderer vorhandenen Regeln bzw. alle systeminternen Einträge (bspw. Hilfsfunktionen vgl. Kap. 3.2.2 und Kap. 5.2.2.4) zur Verfügung
- *Regel Komponenten* – die vier Eingabefelder der Komponenten des standardmäßigen Regelrumpfes (Quelltext als Lückentext mit Platzhaltern für die über die Komponenten eingegebenen Werte, vgl. Kap.

3.2.1); die Komponentenwerte der anderen vorhandenen Regeln werden dabei als Hilfestellung jeweils in einem Dropdown-Steuerelement zur Erleichterung der Eingabe angeboten.

- *Verwendete Synonyme* (Tabelle aller Schlüsselbegriffe der Regel) – im unteren Abschnitt werden zusammenfassend alle statisch vergebenen (natürlichsprachlichen) Synonyme gelistet, die im Lückentext des Hauptquelltextes einer Regel verknüpft sind und deren hinterlegte Werte dann vor der Ausführung entsprechende Leerstellen ausfüllen. Im tabellarischen Steuerelement können sie dabei direkt bearbeitet/angelegt werden.

Bezugnehmend zu den entsprechend ihres Inhalts dargelegten persistierten Felder einer Regel (Kapitel 5.2.2.4), korrespondieren diese drei Abschnitte mit einer dreistufigen Eingabemaske für die eigenständigen Regeldateien. Im Gegensatz zum einfachen Aufbau einer Regelsatzdatei, die als Gruppierungselement jeweils eine Liste mit eindeutigen Identifikatoren (GUID) der referenzierten Regeln beinhaltet, wurde dieser dreistufiger Aufbau der Regeldatei zur Unterstützung verschieden versierter Nutzer*innengruppen umgesetzt (vgl. Tabelle 5).

Aufgrund des gestuften Aufbaus der Regeldatei stehen der Regelkonfiguration drei Ebenen der Anpassung, bzw. (Neu-)Verfassung der Regelinhalte zur Verfügung. So besteht die einfachste – einer reinen (Re-)Konfiguration dienenden – Ebene aus Schlüsselbegriffen. Den Schlüsseln können im Objektlexikon als Eintrag (Schlüssel – Wert Paar) sowohl dynamisch²⁷ (d.h. erst bei ihrem Abruf zur Laufzeit berechnete, vgl. POOL Einträge im Kap. 5.1.2.4) als auch statisch²⁸ (vgl. standardmäßige Regelkomponenten bzw. Synonyme) hinterlegte Werte zugeordnet werden. Diese Schlüsselbegriffe können gemäß den Anforderungen (und auch bspw. gemäß dem in Büro/Fachdomäne gebräuchlichen

²⁷ wie beispielsweise einem Schlüssel *Wände* mit dem ein Skriptausdruck bzw. Quelltext zum Bilden einer Elementmenge aller Wandbauteile im Modell verdeckt wird

²⁸ wie beispielsweise ein Schlüssel *Wand*, der einen Modell-, Skript- bzw. Quelltext-spezifischen Ausdruck verdeckt IfcWall

Vokabular) des spezifischen Prüfungskontexts jeweils selbst vergeben werden. Damit lassen sich Regelvorlagen (Schablonen die bspw. erst im Rahmen der Projektarbeit situativ mit konkreten Werten ausgeprägt werden) allein durch diese Konfigurationsebene für technisch nicht versierte Anwender*innen mit entsprechend eigener (fachlichen) Begrifflichkeit aufbereiten, die ohne technische Kenntnisse der hinter den Schlüsseln verdeckten Komplexität bedarfsgerecht angepasst werden können.

Ein Ändern bzw. Verfassen der hinter den Schlüsselbegriffen stehenden berechnungstechnischen Algorithmen wird dann auf einer zweiten Ebene in einem an den Abfragesprachenstandard Object Constraint Language (OCL) angelehnten Vokabular ermöglicht. Durch Kenntnis der .NET Moduls LINQ, sowie den entsprechenden Ausdrücken, können Anwender*innen nach der Erstellung eines neuen Regel POOL Eintrags (s.o.; vgl. *Spezifische Pool Einträge* Abbildung 66) mit einem Standardausdruck zum Bilden einer Modellelementemenge erstellten Eintrag in zwei alternativen Ansichten zur Bearbeitung selektieren. Standardmäßig wird der Ausdruck beim Aufklappen der Inhalte eines Eintrags in strukturierter Form, d.h. jeder Teil des Ausdrucks wird jeweils beginnend mit einer Hauptvokabel – FROM, IN, WHERE, SELECT – in eigenem Texteingabefeld in untereinanderliegenden Zeilen aufgelistet.

Durch Umschalten mittels der Reiter-Schaltfläche *CompleteTerm* oberhalb des dargestellten Ausdruckes eines Eintrages kann der Code als Fließtext bearbeitet werden. Die Eingabehilfe ermöglicht mit dem Dekomponieren des Ausdrucks entlang seiner Hauptvokabeln bereits eine rudimentäre Syntaxprüfung.

In dem die Einträge als Schlüssel-Wert-Paar direkt im zentralen Objektspeicher POOL beim Ausführen einer Regel geladen werden, besteht die Möglichkeit, im Abschnitt *Modell Abfragen Pool* auch für Argumente der im Quelltext wiederkehrende Modellabfragen behandelnden Hilfsfunktionen zugehörige Schlüsselbegriffe mit entsprechenden Werten zu hinterlegen. So kann die Hilfsfunktion beispielsweise zum Auslesen eines Elementeigenschaftssatzes „GetPropertySetsOfObject“ (IfcPropertySet; vgl. Kap. 6.2.1) durch Anlegen eines Eintrags mit ihrem Argumentbezeichner „GetPropertySetsOfObject_obj“ als Schlüsselbegriff und einem spezifischen Bauelement belegten Wert im *Modell Abfragen Pool* vorbereitet werden. Die Eigenschaftsmenge,

die dem Element zugeordnet ist, wird dann entsprechend mit dem Aufruf der Hilfsmethode zurückgeliefert.

Auch Ergebnis(teil-)mengen zur (detaillierten) Weiterverarbeitung in folgenden Regeln können in den Eingabefeldern dieses Abschnitts definiert werden. Prinzipiell können alle Ausdrücke der LINQ Komponente als (dynamische) Werte der Einträge verwendet werden. Durch die einfach zu erlernende Abfragesprache können somit auf dieser Ebene bereits entsprechend geschulte Anwender*innen umfängliche Prüfungen und Analysen in entsprechende Regeln verfassen.

Der mittlere Abschnitt der linken Seite des Werkzeugs beinhaltet die Eingabefelder der Regelkomponenten. Diese vier Komponenten zur Bestimmung von jeweiligen Teilen eines logischen Ausdruckes einer Regel (vgl. Kap. 3.2.1), sowie die Komponente *Ergebnis Auswertung*, ermöglichen es, wie eine (einfach zu erlernende) Fassade die dahinter liegende Skriptausdrücke und den Quelltext zu verdecken. Beim Erstellen von Regelrumpfen können diese Eingabefelder dazu verwendet werden, nicht-versierten Endanwender*innen in gewissem Umfang das Anpassen der Regel zu ermöglichen. So bezieht sich das die Komponente *Hauptelement* in Abbildung 65 beispielsweise auf eine bestimmte (wie oben beschrieben zuvor in einem Pool Eintrag festgelegte) Elementemenge aller Bauteile des Modells. Hier kann nun ohne weitere Kenntnis eine andere Menge aller Fenster und Türen aus einer anderen Regel (vgl. Abbildung 66 rechte Spalte *Alle Spezifischen Einträge*) eingetragen werden, an denen dann mit der angepassten Regel der unter der Komponente *Merkmal* spezifizierte Sachverhalt überprüft werden kann.

Diese durch Komponenten vereinfachte Darstellung grundlegender Aspekte einer Regel zur Erleichterung ihrer Anpassung und Wiederverwendung bilden dabei einen (inhaltlichen) Teil der im untersten Abschnitt *Verwendete Synonyme* der linken Seite des Werkzeugs bezüglich aller in der Regel verwendeten, als Eingabefelder zusammengefassten Schlüsselworte. Damit werden die dahinterstehenden (statischen) Ausdrücke, die beim Zusammensetzen des finalen Regelquelltextes in die unten beschriebenen Lückentexte die designierten Leerstellen füllen, übersichtlich präsentiert und sind kompakt für Bearbeitung bzw. Erstellen/Entfernen zugänglich.

Auf der detailliertesten (eine höchste Stufe an Vorkenntnissen erfordernde, vgl. Tabelle 5) Ebene des Quelltexts sind dann die für die Berechnung der einzelnen in der Abfragesprache verfassten Ausdrücke sorgenden Codierungen verfasst. Hier finden grundlegende Anpassungen der Art der Regel durch erfahrene (Regel-)Entwickler statt. Da diese die gesamte Regel rahmende Programmiersprachenschablonen dann jeweils erst zur Laufzeit der Softwareanwendung kompiliert und ausgeführt werden, kann der Funktionsumfang allein durch das Gestalten der eigenständigen Regeln umfänglich erweitert werden. Hierfür stehen den Programmierer*innen zur Bearbeitung des Regelquelltexts im linken Bereich des Werkzeuges aufklappbare Eingabefelder zur Verfügung. Entsprechend der zwei jeweils Lückentext beinhaltenden Hautteile eines Regelquelltexts, der in den beiden Feldern RuleBody und ViolatesRule der Regeldatei persistiert wird (vgl. Kap, 5.2.2.4), kann dieser in den gleichlautenden Eingabefeldern bearbeitet werden. In einem weiteren aufklappbaren Textfeld kann, nachdem ein geänderter Lückentext in einer der zwei Hauptteile des Regelquelltextes geändert und mit der Schaltfläche *Speichern* persistiert wurde, (gemäß des „Print“ im Read-Eval-Print Loop – REPL Prinzip) der gesamte Quelltext (d.h. die Lückentexte befüllt mit den hinter den dynamischen sowie statischen Schlüsselworten stehenden Werten) eingesehen werden. Zudem be-

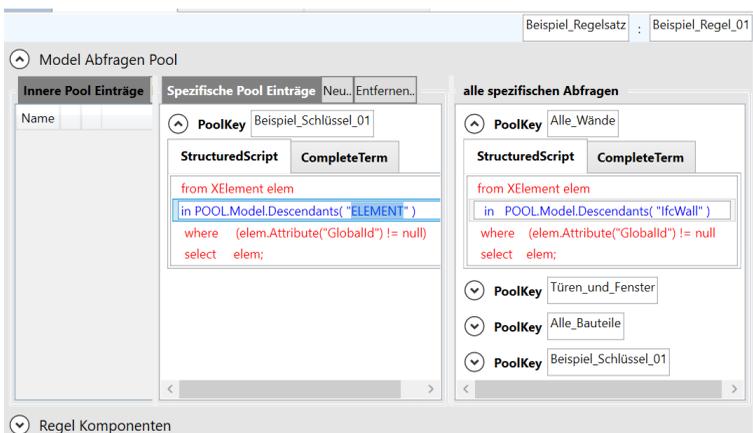


Abbildung 66 Regelbearbeitungswerkzeug - Anlegen spezifischer Einträge im Modell Abfragen Pool einer Regel

finden sich dort Eingabefelder für den Lückentext der nach dem gleichen Prinzip funktionierenden Ereignisbenachrichtigung. Die Entwickler*innen können in dem ersten ausklappbaren Abschnitt den durch hinterlegte Mechanismen zusammengeführten finalen Quelltext einsehen. Mit der Schaltfläche Testdurchlauf können sie neben dieser visuellen Kontrolle auch gemäß eines REPL Entwicklungsprinzips den Quelltext ausführen und (falls kein Modell geladen ist, anhand des standardmäßig hinterlegten Referenzmodells in den Nutzer*innendaten) testen. Wenn bereits ein (Referenz-) Modell geladen wurde, kann die bearbeitete Regel darauf (wie auch schon im Standardmodus der Software beschrieben, s.o.) ausgeführt werden, indem manuell in das Regelausführungswerkzeug umgeschaltet und dort die entsprechende Schaltfläche betätigt wird. In beiden Fällen werden die Anwender*innen zunächst mit Dialogfenstern auf eventuelle Codierungsprobleme hingewiesen und können die Ausführung stoppen. Eine ausführliche Fehlerausgabe findet im Falle von Problemen mit dem Quelltest in der Konsole des Regelausführungswerkzeugs statt. Da hierbei die Kompilierungs- bzw. Ausführungsmeldungen direkt vom (Just In Time) JIT Compiler des .NET Frameworks bzw. entsprechende Windows Systembenachrichtigungen an die Anwender*innen weitergereicht werden, ist die diesbezügliche Anwender*innenbenachrichtigung der Software vergleichbar zu Kompilierungsereignisausgaben moderner Entwicklungsumgebungen (z.B. Microsofts Visual Studio). Für ein grundsätzliches (Neu-)Entwickeln von Regelrümpfen sowie ein umfängliches Anpassen bestehender Regeln auf der Quelltext-Ebene stehen mit den Bearbeitungs- und Ausführungswerkzeugen der Software den Anwender*innen geeignete Mittel der Regelentwicklungsarbeit zur Verfügung.

Weitere Hilfestellung zur Entwicklung – Modellexplorerwerkzeug - Für die Regelanpassungen wird der geführte Ablauf zur Überprüfung eines Modells (s.o.) mit weiteren Funktionen ergänzt, und wird zur iterativen Regelentwicklung verwendet. Im ersten Schritt wird die Visualisierung durch einen Modellmengen Explorer ergänzt.

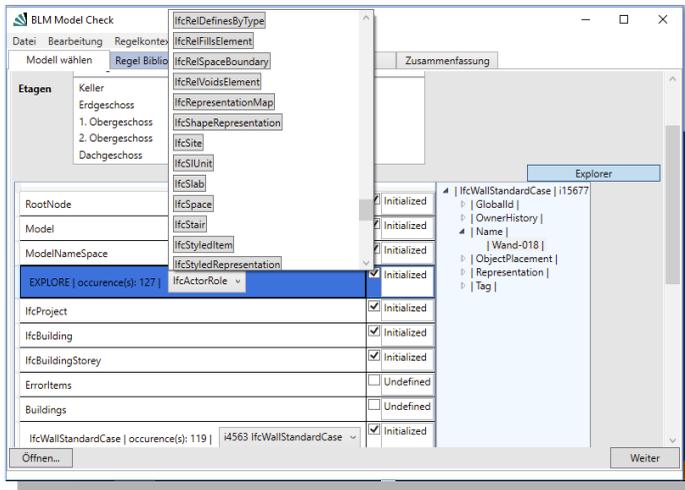


Abbildung 67 Modellexplorerwerkzeug

Nach dem Laden eines Modells können darin beim Abruf des Eintrags „EXPLORER“ sämtlicher Elementnamen der Modellinstanz gelistet werden. Mit der Auswahl eines Elementnamens wird eine Abfrage zur Bildung einer Elementmenge generiert und ein abrufbarer Eintrag dem Explorer hinzugefügt. Mit dem Abruf dieses Eintrags wird die Elementmenge gebildet und alle im Modell vorkommenden Elementinstanzen in einer Auswahlliste aufbereitet. Darüber kann ein bestimmtes Element aufgerufen werden, dessen Inhalt dann in einer Detailansicht dargestellt wird. Neben der Möglichkeit mit dem Explorer das Modell inhaltlich zu untersuchen, stehen die generierten Elementmengen Abfragen für deren spätere Einbindung in die Regelentwicklung zur Verfügung (s. o. Regelbearbeitungswerkzeug).

7.2 Anwendungsbeispiele aus weiteren Projekten

Der in dieser Arbeit vorgestellte Lösungsansatz ermöglicht eine flexible Anpassungsmöglichkeit auf verschiedene Kontexte durch den inbegriffenen regelbasierten Erweiterungsmechanismus. Damit konnte der Grundansatz nach initialer Entwicklung für den Rahmen der vorliegenden Arbeit, schon im Laufe deren Verfassens, bereits in Kontexten von verschiedenen Projekten evaluiert werden. In drei Projekten werden im Weiteren vor dem Hintergrund darin untersuchter Handlungsfelder verschiedene Aspekte des Lösungsansatzes bezüglich der entsprechend implizierten Anwendungskontexte beleuchtet. Die dabei betrachteten Aspekte beziehen sich sowohl auf die softwareseitige Umsetzung jeweiliger Projektprototypen als auch auf die Entwicklung und Umsetzung von Prüf- bzw. Analyselogik auf Seiten der Anwender*innen zur bedarfsgerechten Anpassung der Prototypen auf die Projektkontexte.

7.2.1 BBR Model Check

Eines der ersten Pilotprojekte zum Themenfeld BIM setzte das BBR beim Neubau des Berliner Humboldt Forums an (vgl. Abbildung 68). Schon im Vorfeld der Planungen dieses Bundesbauprojektes wurde die bestehende Baufachricht-



Abbildung 68 Modellansicht des Berliner Humboldt Forum (Quelle: M. Reif, BBR)

linie BFR GBestand der öffentlichen Hand (Koggelmann u. a., 2012) durch Inbezugsetzen zum IFC Modellstandard als *BBR BIM-Qualitätsstandards* aufbereitet. Auf Seiten der Werkzeuge waren bereits Tools zur geometrischen

Prüfung von Verschneidungen in BIM-Modellen im BBR im Einsatz, um die von den Planenden erstellten modellbasierten Planungsgegenstände auf Kollisionen hin prüfen zu können. Bezüglich der alphanumerischen Inhalte, die durch die Richtlinie für die Abgabemodelle festgelegt wurden, fehlte es jedoch noch an Hilfsmitteln zur Kohärenzprüfung. Darüber hinaus wurde bestimmt, dass für zukünftige Revisionen bzw. Erweiterungen entsprechende Kohärenzprüfungen im Hilfsmittel durch die Mitarbeiter*innen des BBR anpassbar sein sollten. Da eine Marktanalyse unter Berücksichtigung der spezifischen Bedarfe ergab, dass keine Lösung verfügbar war, wurde die Entwicklung des Hilfsmittels ausgeschrieben.

Im Rahmen dieses Entwicklungsauftrages konnte auf Basis des Lösungsansatzes der vorliegenden Arbeit die Softwareanwendung BBR Model Check (vgl. Abbildung 69) zur Unterstützung der Mitarbeiter*innen des BBR ausgeliefert werden (vgl. von Both & Ebertshäuser, 2018).

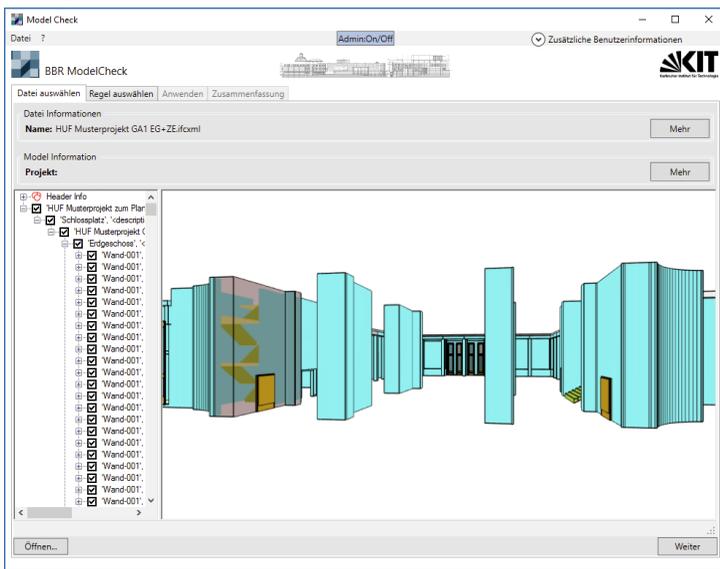


Abbildung 69 BBR ModelCheck (überarbeitete, erweiterte Version 0.5 – 2016)

In der Praxis-Anwendung im Bauprojekt Berliner Humboldt Forum ermöglichte das Tool während der frühen Entwurfsplanung eine Überprüfung von

Planungsmodellen anhand der für die spezifischen Prüfbedarfe entwickelten und mit der Software ausgelieferten Regeln. Die regelbasierte Prüfung von BIM-Modellen (IFC) beinhaltete dabei u.a.:

- Bauteiltypen
- Attribuierung (z.B. Raumcodes, Merkmalslisten)
- Modelstruktur / Topologie
- Materialisierung / Bemusterung

Aus den in der Dokumentationsrichtlinie festgelegten Namensnomenklatur ergaben sich dabei beispielsweise besondere Prüfbedarfe. Dies erforderte neben dem Auslesen von extern vorgegebenen, in eigenständigen Tabellen bereitgestellten Bezeichnungsvorgaben auch einen dynamisch im Modell aus verschiedenen Elementen zusammengesetzten Sollwert, gegenüber dem ein bestimmter Bauelementname – wie der einer Tür – auf Kohärenz hin geprüft werden sollte. Ein zu überprüfender Türbezeichner setzt sich dabei z.B. aus verschiedenen lagebestimmten Präfixen von Liegenschaft-, Gebäude- und Stockwerkskürzel zusammen, die jeweils vor der eigentlichen Kohärenzprüfung aus den entsprechenden kontextuellen Modellelementen zusammengestellt werden mussten. Die ausgelieferte Regelbasis folgt konzeptionell der in der vorliegenden Arbeit vorgestellten Umsetzungsstrategie (vgl. Kap. 6.1). Durch das Bereitstellen von Regelschablonen als Grundbausteinen, auf die dann die eigentlichen Regeln aufbauen, wird dadurch eine flexible Grundlage zur (eigenen) Weiterentwicklung durch die Endanwender*innen ermöglicht. Neben dem Ausführen von vorhandenen Regeln auf ein bestimmtes Planungsmodell, war das eigene Konfigurieren, bzw. die benötigte Kenntnis der den Regeln unterlegten Skriptsprache hierfür, Bestandteil einer eintägigen Schulung für die Mitarbeiter*innen des BBR. Bereits rollenspezifisch gemäß der Vorkenntnis sowie den designierten Aufgaben (Endanwendung bzw. Regelentwicklung) – in der Schulung adressiert – konnten die Mitarbeiter*innen allein unterstützt durch wenige Telefonberatungen die Softwareanwendung in Gebrauch nehmen. Dabei waren sie auch in die Lage versetzt, vorhandene Regeln selbstständig an ihre ggf. geänderten Bedarfe anzupassen. Nach der Rückkopplung mit den ersten Erfahrungen aus der Anwendung der Software, konnte

in einer Programm- sowie Regelrevision sowohl auf der Seite von Ergonomie der Benutzeroberfläche bzw. Softwarefunktionalität wie auch auf der Seite von den Grundregeln und der darauf aufgebauten Regeln Schwachstellen behoben und Verbesserungen erzielt werden.

7.2.2 Verbundprojekt EnEff-BIM

Im vom Wirtschaftsministerium geförderten Verbundprojekt wurden Methoden und Konzepte untersucht, wie die Planung energieeffizienter Gebäude mit IT-gestützten Hilfsmittel unterstützt werden kann. Ein wichtiger Schwerpunkt lag dabei auf der Verwendung von offenen Standards, sowohl auf Seiten der Planung mit dem etablierten BIM Austauschformat IFC als auch auf fachlicher Simulationsseite mit dynamischen Modellierungen in der offenen Modelica-Sprache²⁹ sowie eine Kopplung von Simulationsmodellen mit dem Functional Mockup Interface (FMI). Durch die verfolgten Standardisierungsansätze konnten mit den Projektergebnissen auf den Ebenen der Planungspraxis, sowie der wissenschaftlichen (Zusammen-)Arbeit hilfreiche Beiträge geleistet werden. Durch die Einbindung des Verbundprojekts in das internationale Rahmenprojekt Annex 60 der International Energy Agency (IAE) konnte dabei das Ziel unterstützt werden, den Modelica Standard in der Baubranche weiter zu etablieren. Hierzu wurde von den Projektpartnern mit Simulationsexpertise am Entwickeln einer gemeinsamen Basisbibliothek³⁰ für bereits zur Gebäudesimulation an verschiedenen Hochschulen angesetzte Modelica Komponenten Bibliotheken mitgewirkt (vgl. Wetter & Treeck, 2017). Für die Seite der Planungspraxis zeigt Abbildung 70 schematisch die im Projekt zur Unterstützung der Planer*innen angesetzte Werkzeugkettenkonzeption mit den durch den Autor beigetragenen Werkzeugentwicklungen (vgl. blauer Kasten). Hierdurch sollten Hemmnisse in der Praxis zum Einsatz der IT-gestützten Hilfsmittel ge-

²⁹ Das wesentliche Merkmal der objekt-orientierten Simulationssprache besteht in einem jedem Element inhärenten Teil eines numerischen Ausdrucks, der zum Simulieren dann gemäß der instanziierten Topologie (Objektgraf) zusammengesetzt wird.

³⁰ Webbasierte Dokumentation Version 1.0 der Basisbibliothek: <http://www.iea-annex60.org/releases/modelica/1.0.0/help/Annex60.html>

senkt werden, deren Ursachen u.a. in der komplexen Bedienung von Fachsoftware sowie in der aufwändigen Neuerstellung bzw. Nachmodellierung von Planungsmodellen gemäß den Anforderungen der Simulation liegen (vgl. von Both u. a., 2017). Dazu wurde eine Anbindung der im offenen BIM Standard IFC ausgetauschten Planungsinformationen an die Modelica-basierten Simulationsmodelle verfolgt. Indem dabei ein in der offenen EnergyPlus Plattform verwendetes Format SimModel als Zwischenformat verwendet wurde, konnte schon durch nur eine Konvertierung eine erste Simulation angestoßen werden.

Der nächste Schritt in der Werkzeugkette besteht darin, dieses Zwischenformat durch ein Werkzeug zur Parametrisierung aus entsprechenden Komponentenbibliotheken bezogenen Modelica-Simulationsmodellen zu verwenden und diese parametrisierten Modelle dann im Kernel einer Simulationsengine zu simulieren (vgl. Abbildung 70).

Durch die Vereinheitlichung des Arbeitsgangs in einem generalisierten werkzeuggestützten Referenzprozess sollten die (bisher an den verschiedenen Hochschulen mit spezifischen Parametrisierungsvorgängen entwickelten) Simulationsprozesse vereinfacht und damit einer praktischen Anwendung nähergebracht werden (vgl. von Both u. a., 2017).

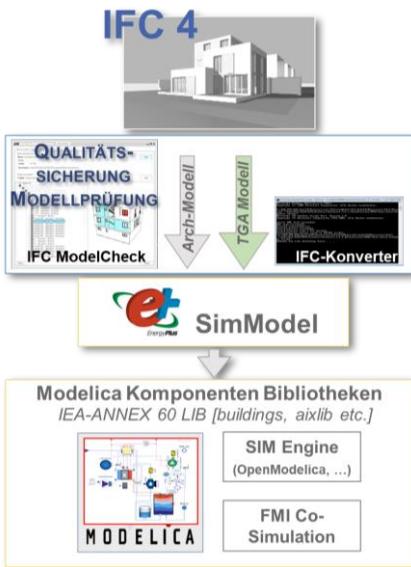


Abbildung 70 Schematischer EnEff-BIM Ansatz mit KIT Tools (ergänzte Grafik nach v. Both)

Der Lösungsansatz der vorliegenden Arbeit in einem Softwareprototyp umgesetzt werden. Die Grundlage für die Anpassung an den Anwendungskontext wurden

Für den Anwendungskontext einer Vollständigkeits- und Kohärenzprüfung von den in die im Verbundprojekt entwickelte Werkzeugkette eingehenden BIM-basierten Planungsmodelle (siehe Abbildung 71) konnte der Lösung

im Rahmen einer Modelsicht (MVD) für die Energiesimulation zum IFC Schema von den Projektpartnern entwickelt. Darin sind die Anforderungen bezüglich der Vollständigkeitsprüfung von energiebezogenen Informationen im Modell definiert. Die Anforderungen umfassen dabei einerseits eine generelle Existenzprüfung zu obligatorischen bzw. optionalen Modellelementen und andererseits die Kohärenzprüfung zu geforderten (spezifisch auszuprägenden) Inhalten bzw. Wertausprägungen in den Datenfeldern der (überprüftermaßen) existierenden Elemente. Durch ein vorhandenes Brennerobjekt kann beispielsweise, erst mit der zusätzlichen Angabe des primären Energieträgers (z.B. Gas), eine obligatorische Wärmeversorgungsanlageninformation im Modell vollständig beschrieben werden (vgl. Ebertshäuser & Both, 2016b).

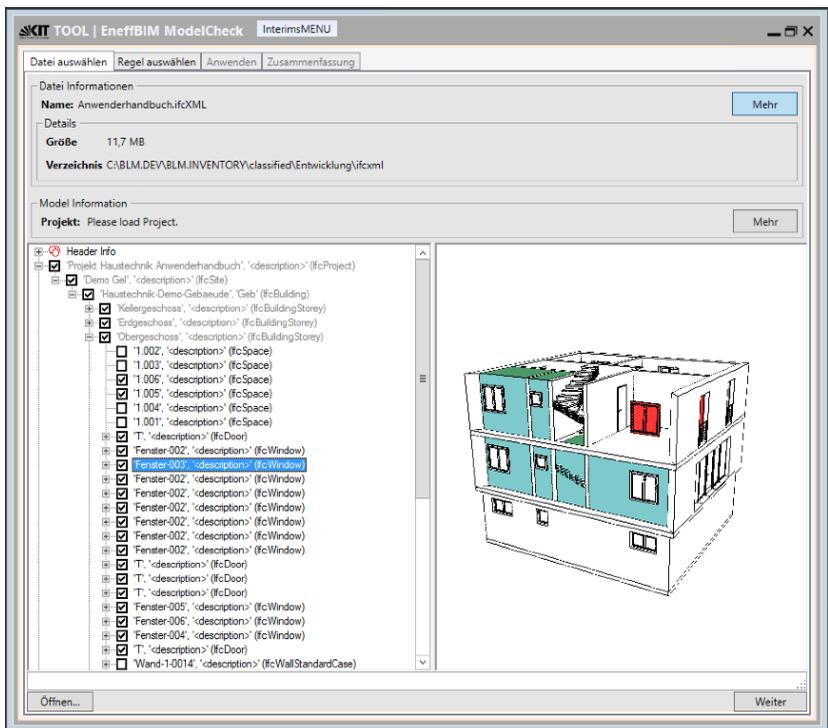


Abbildung 71 Softwareprototyp EneffBIM ModelCheck

Über diese (statisch) ermittelbare Vollständigkeit von Modellinhalten hinaus benötigen bestimmte Anlagensimulationen einen vorherigen Konsistenzcheck von in Verbindung stehendem Gefüge aus mehreren Modellelementen, damit sie bedeutungsvolle Ergebnisse erbringen können. Beispielsweise setzt sich ein Leitungssystem aus verschiedenen Rohrleitungsteilen (z.B. Leitungsrohr, Verbindungsmuffe usw.) sowie Anlagenteilen (z.B. Pumpe usw.) zusammen. Wenn alle Teile vorhanden sind, bedeutet dies noch nicht, dass das Leitungssystem geschlossen ist und bestimmte obligatorische Elemente mit einschließt. Eine Konsistenz als geschlossener Kreislauf bedarf eines schrittweisen Prüfens der einzelnen verbundenen Systemelemente – ausgehend von einem ersten Element dann der Reihe nach bis schließlich die letzte Verbindungsbeschreibung wieder auf das Ausgangselement verweist. Mit dieser Art von Anforderungen werden die über die statischen Anforderungen zur Vollständigkeitsprüfung hinausgehenden Aspekte für eine Anpassung des Lösungsansatzes auf den Anwendungskontext im Projekt definiert. Durch die Entwicklung von einer entsprechend in Regeldateien formalisiert zusammengestellten Prüflogik zu beiden Arten von Anforderungen konnte dies im einfach zu bedienenden Softwareprototyp (vgl. hierzu *Standardmodus* in Kap. 7.1) in einer bedarfsorientierten Regelbasis umgesetzt werden. In Abbildung 72 werden die Regeln zu den in folgender Liste exemplifizierend zusammengestellten Anforderungen (vgl. Ebertshäuser, 2017) im Rahmen der Regelbibliothek des Regelauswahlwerkzeuges in der Bedienoberfläche des Prototyps dargestellt.

Obligatorische Regeln:

- Gebäudekontextinformationen, z.B. Einheiten, passende Hierarchie der Elemente, geografische Lage des Bauwerks
- Den Bauelementen sind Materialien (Schichten) zugeordnet (Bemusterung)
- Den Raumelementen sind Raumbegrenzungselemente zugeordnet
- Alle (TGA) IFC Komponenten müssen zu einem Anlagensystem (IfcSystem oder einem davon abgeleiteten Element) verknüpft sein

- Luftführende Systeme sind mit einer thermischen Zone verknüpft
- Mechanische luftführende Leitungssysteme beinhalten mindestens eine Lüfterkomponente

Optionale Regeln:

- Mechanische luftführende Leitungssysteme sind entweder als Frischluft- oder als geschlossene Kreislaufsysteme ausgezeichnet
- Geregelte Wassersysteme beinhalten mindestens eine Pumpenkomponente

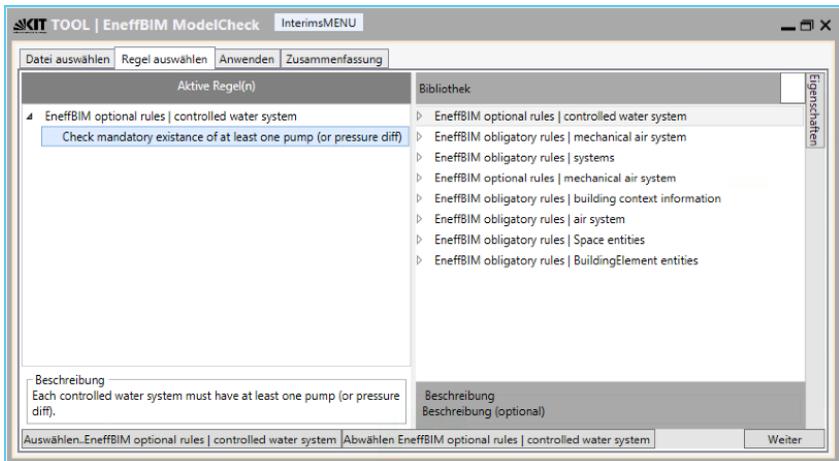


Abbildung 72 EneffBIM ModelCheck - Exemplifizierte TGA Regelbasis

Da im Projektkontext kein Anpassen der mit dem Prototyp ausgelieferten Regeln vorgesehen war, wurde die Regelbasis nicht als eigenständige Regeldateien bereitgestellt, sondern in den Rahmen der Software eingebunden. Bei einer möglichen Weiterentwicklung des Prototyps können diese Programmressourcen aufgrund ihrer Formatierung jedoch ohne Änderung wieder eigenständig bereitgestellt von der Software abgerufen werden. Die in diesem Projekt entwickelten Tools werden gemäß ihrer Förderbestimmungen aus

der EnTool Initiative des Bundeswirtschaftsministeriums zur Förderung des Transfers von IT-gestützten Forschungsergebnissen in die Praxis der Öffentlichkeit frei zur Verfügung³¹ gestellt (vgl. Ebertshäuser, 2017).

7.2.3 Integriertes Semantisches Informationsmodell für den Kontext der energieeffizienten Stadt

Zur fristgerechten Erreichung der aktuellen klima- und energiepolitischen Zielsetzungen der Bundesrepublik Deutschland kommt den Städten und Gemeinden in den kommenden Jahren eine entscheidende Bedeutung zu. Die Konzeption, Evaluierung und Umsetzung nachhaltiger Konzepte in der kommunalen Planung erfordert den Einsatz integrativer Instrumente, mit denen sich frühzeitig alle relevanten Aspekte und Akteure bzw. Betroffene innerhalb des komplexen, dynamischen Systems „Stadt + Energie“ einbeziehen lassen. Ein lösungsneutrales Bestimmen von geeigneten Handlungsfeldern wird dabei oftmals bereits in den planungseinleitenden Phasen durch die mangelhafte Verfügbarkeit einer geeigneten Informationsbasis als adäquate Analyse- und Entscheidungsgrundlage erschwert. Falsche oder fehlende Informationen zur kommunalen Ausgangslage führen daher, unter einem stetig zunehmenden Handlungsdruck, nicht selten zu nicht zweckmäßigen Zielsetzungen und fehlgeleiteten Maßnahmen (Brüggemann & Both, 2014).

Im Rahmen des vom Bundeswirtschaftsministerium geförderten Forschungsvorhabens „ISIS – Entwicklung eines integrierten semantischen Informationsmodells als Planungshilfsmittel“ entwickelte das BLM ein mehrdimensionales und -skaliges Informationsmodell für den Anwendungskontext der energieeffizienten Stadtentwicklung. Gemäß den Anforderungen einer Querschnittsanalyse des kommunalen Informationsmanagements sind im Konzeptmodell alle kontextuell relevanten Aspekte eingeflossen. Wobei das Ziel verfolgt wurde, allen beteiligten Akteuren und Disziplinen auf der strategischen, taktischen und operativen Planungsebene eine einheitliche, konvergente und transparente

³¹ Download der vom Autor im Projektrahmen entwickelten Softwareprototypen unter: download.building-lifecycle-management.de

Informationsbasis zur Verfügung zu stellen. Durch die Realisierung einer fortlaufend modellbasierten Prozesskette in der kommunalen Planung soll auch die technische Interoperabilität verbessert werden (ebd.).

Medienbrüche aufgrund fehlender Schnittstellen führen in der planerischen Praxis vielfach zu Fehlinterpretationen und Informationsverlusten, was sowohl die Effizienz als auch die Effektivität der Verfahren maßgeblich beeinträchtigt. Weiterhin würden bei einer kontinuierlichen Anwendung obligatorische Redundanzen bei der Datenhaltung und -erfassung zugunsten einer gesteigerten Zeit- und Kosteneffizienz langfristig vermieden. Der Ansatz einer ganzheitlichen Darstellung des Stadtsystems erlaubt ferner die konsistente Generierung domänenspezifischer Fachsichten auf den Gegenstand, ohne dabei systemische Abhängigkeiten und kausale Wechselwirkungen auszublenden (vgl. Abbildung 73). Diese holistische Sicht beinhaltet ebenso die integrale Abbildung

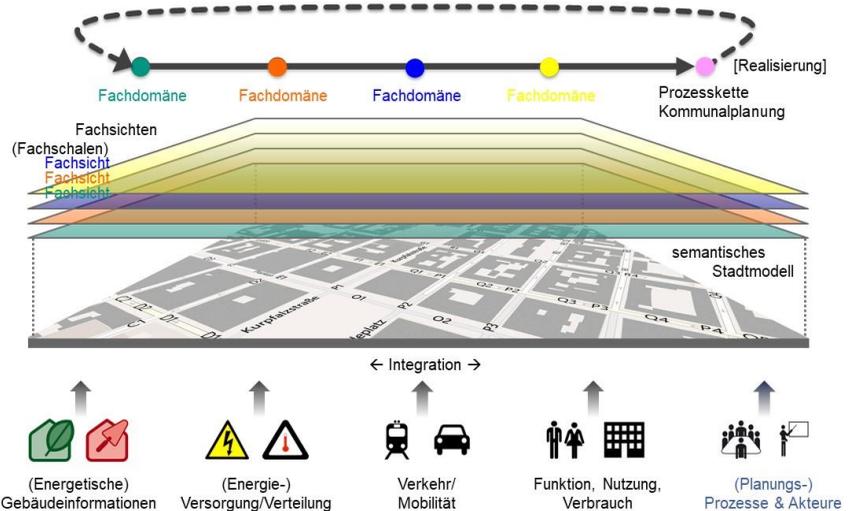
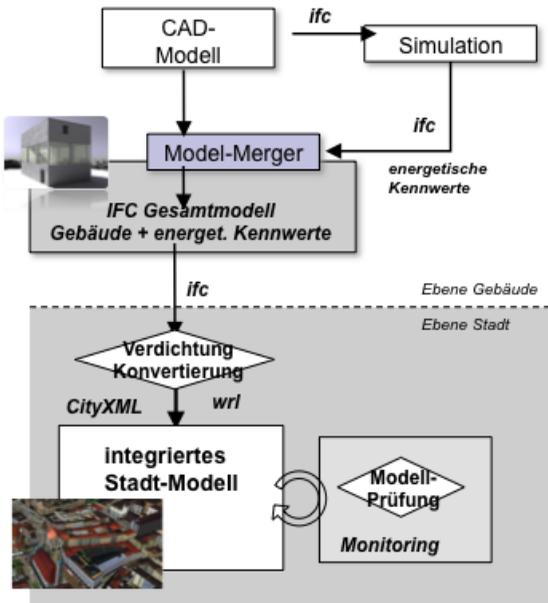


Abbildung 73 Schematisches Inhaltskonzept des integrierten Stadtmodells (Quelle: Brüggemann & Both, 2014)

und kontextuelle Verortung aller relevanten semantischen Informationen zu den kommunalen Energieflüssen (Erzeugung – Verteilung – Verbrauch) sowie

energetischer Fachdaten aus Gebäudemodellen (BIM) auf der Zielebene urbaner Strukturen (ebd.). Ein Schwerpunkt des Projektes lag bei der Umsetzung auf einer Demonstration des Mehrwerts des entwickelten integrierten Konzeptmodells mit IT-gestützten Hilfsmitteln (vgl. Abbildung 74), wobei die modellierungstechnischen Konzeptionen zunächst in etablierte offene Produktmodellstandards der verschiedenen Skalenebenen – Quartier und Gebäude – übertragen wurden. So konnte bei der (Weiter-)Entwicklung der Werkzeuge auf bestehende Komponenten zur Handhabung der etablierten Standards (z.B. Visualisierungskomponente) zurückgegriffen werden. Im Gegensatz zur generellen Erweiterung der Standards (vgl. Model View Definition – MVD bzw. Application Domain Extension – ADE) bei den im Ausblick dargestellten Forschungsarbeiten (vgl. Kap. 8.1), wurde bei der Übertragung auf die inhärenten standardbasierten Erweiterungsmechanismen von sowohl Stadtmodellstandard CityGML (core:genericAttribute Element) wie auch Gebäudestandard IFC (IfcPropertySet Element) zurückgegriffen.



Damit sämtliche sowohl prozess- bzw. aktorsbezogenen Informationen in das integrierte konzeptionelle Stadtmodell mit einbezogen werden konnten, wurden aus einer umfangreichen Analyse von formal vorgegebenen Bauleitplanungsprozessen generalisierte Planungspiloten als ein Grundbaustein des Prozessmodells abgeleitet. Darin wurden sämtlichen von den entsprechenden

Vorgaben im Baugesetzbuch abgeleiteten Prozessschritte mit den zudem möglichen Rückkopplungsschleifen formalisiert abgebildet. Darauf aufbauend wurde ein idealtypischer Planungspilot „Quartierssanierung“ für ein Beispielszenario *Konvoisanierung*³² entwickelt, der sich auf Leitfäden (vgl. Hausladen, u. a., 2011) sowie reale Pilotprojekte im Rahmen der energieeffizienten Stadtentwicklung stützt (vgl. T. Brüggemann u. a., 2015). Eine mittels strukturierten prozessbezogenen Vorgehen ermöglichte Transparenz – sowohl bezüglich der internen akteursbezogenen und externen bevölkerungsbezogenen Kommunikation als auch bezüglich einer retrospektiven Dokumentation der einzelnen Planungs- und Entscheidungsschritte – sollte auf technischer Seite durch einen konzipierten Prozessnavigator verdeutlicht werden. Dies wurde als „Mockup“ einer funktionalen Prozessschale für den Prozesspiloten in der prototypischen ISIS Plattform zur Stadtmodellhandhabung umgesetzt. Daran konnten konkrete planungsunterstützende (skalenübergreifende) Modelloperationen demonstriert werden, für die jeweils einzelne prototypische Werkzeuge entwickelt wurden. In Abbildung 74 sind hierzu die bereits zur Antragsstellung konzeptionell avisierten modellbasierten Grundprozeduren dargestellt, die szenariobasiert in einem konkretisierten Beispielvorgang im Erstellungsprozess eines Energienutzungsplans (vgl. Prozessnavigator in Abbildung 75) eingebunden wurden. So sieht der mehrskalige Ansatz zunächst auf Seiten der fachlichen energetischen Gebäudebewertung eine mittels Simulation bzw. Berechnung angereicherte modellbasierte Planungslösung vor (vgl. Model-Merger). Dieser Teilvorgang unterliegt im idealtypischen Vorgang der Anreicherung eines BIM-Modells mit energiebezogenen Werten, die im Rahmen eines Energieausweises nach ENEV stehen (vgl. Ebertshäuser & Both, 2016a). Hierzu wurden gemäß dem zur Erhebung und Vorhalten der Ausweisdaten für die Erfüllung ihrer Kontrollpflicht der Energieausweisdaten vom Deutschen Institut für Baunormung veröffentlichten Datenschema *Kontrollsystem.xsd* die entsprechenden energetischen Gebäudedaten in Form von erweiterten Eigenschaften an das IFC Modell hinzugefügt. Beispielsweise kamen zu dem bereits

³² Das Szenario folgte einem realen Projekt der Stadt Karlsruhe, bei dem im Stadtteil Waldstadt durch Förderung von Energieausweisen in einem Quartier mit Reihenhaustypologie die Eigentümer zur gemeinschaftlichen Sanierung animiert werden sollten.

am Bauteil hinterlegbaren U-Wert, Werte zur Beschreibung des primären bzw. sekundären Energiebedarfs usw. als Eigenschaften an das Bauwerk hinzu.

Von der Ebene des Bauwerks wurde eine weitere Modellhandlungsprozedur eingearbeitet, mittel der die BIM-basierten Gebäudedaten in eine Bauwerksrepräsentation im Stadtmodell (vgl. *Konvertierung* in Abbildung 74) transformiert werden können. Im Rahmen eines Unterauftrages wurde für die Umsetzung dieses Transformationsschritts das KIT Institut für angewandte Informatik (IAI) mit der Erweiterung ihres bereits sowohl IFC wie auch GML unterstützenden Werkzeuges FZK Viewer beauftragt. Dabei werden die energetischen Parameter an das CityGML Bauwerk mit übertragen, sodass die Gebäuderepräsentation des integrierten konzeptionellen Stadtmodells mit der Abbildung im davon abgeleiteten CityGML-Anwendungsmodell übereinstimmt. Dieses und die weiteren auf Basis des Lösungsansatzes der vorliegenden Arbeit prototypisch implementierten Werkzeuge können aus der rahmengebenden ISIS Plattform heraus aufgerufen werden. Beispielsweise kann das in Abbildung 76 dargestellte Projektmodul durch entsprechende Auswahl im Prozessmodul für den Beispielvorgang zum Kartieren eines Projektgebietes anhand der Datenintegration von Energieausweisdaten der darin befindlichen Gebäude bedarfsorientiert ausgeprägt und gestartet werden (vgl. Abbildung 75). Dazu wird im *Prozessnavigator* den Anwender*innen eine exemplifizierte

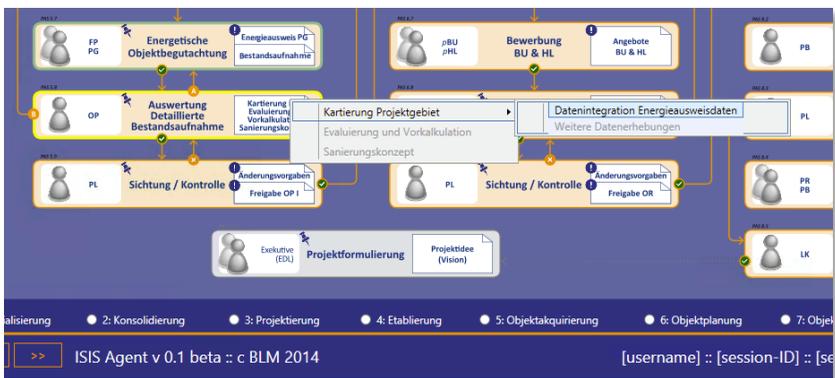


Abbildung 75 Stadtmodellplattform ISISAgent – exemplifiziertes Modul des Prozessnavigators

Auswahlmöglichkeit im Prozessschritt *Auswertung detaillierter Bestandsaufnahme* des präsentierten Planungspiloten zum Erstellen des Energieleitplans angezeigt. Durch die demonstrierte Kopplung zum Prozess gelangen die Anwender*innen zum bereits gemäß dem avisierten Prozessschritt erforderlichen modellbasierten Funktionalitäten bestückten Projektmodul. Die Auswahl eines Projektes ermöglicht durch eine dem Modul unterlegten, gemäß dem Lösungsansatz der vorliegenden Arbeit gestalteten Komponente eine regelbasierte Filterung des Projektgebietsausschnitts im Stadtmodell (vgl. Abbildung 76). Für jedes Gebäude wird in dem Werkzeug ein geführter Ablauf durch verschiedene Stationen einer Werkzeugkette vorgehalten – *Bestandsanalyse*, *Gebäude überführen* sowie *Datenintegration*. Mit einem finalen Schritt *Prozess abschließen* kann der Bearbeitungsvorgang für die aktuell bearbeiteten Gebäudedaten dann abgeschlossen werden, indem der Vorgang in eine Liste abgearbeiteter Gebäudeintegrationen abgelegt wird (vgl. ebd.). Schon beim Aufruf eines bestimmten Projekts im Projektmodul, wurde das Modul mit eigenständigen Auswahlregeln zum das Filtern aller einem Projekt zugehörigen Teile des Stadtmodells sowie der darauf basierenden Aufbereitung einzelner (geführter) Arbeitsvorgänge bestückt. In gleicher Weise wird jeweils beim Bedienen der Schaltfläche

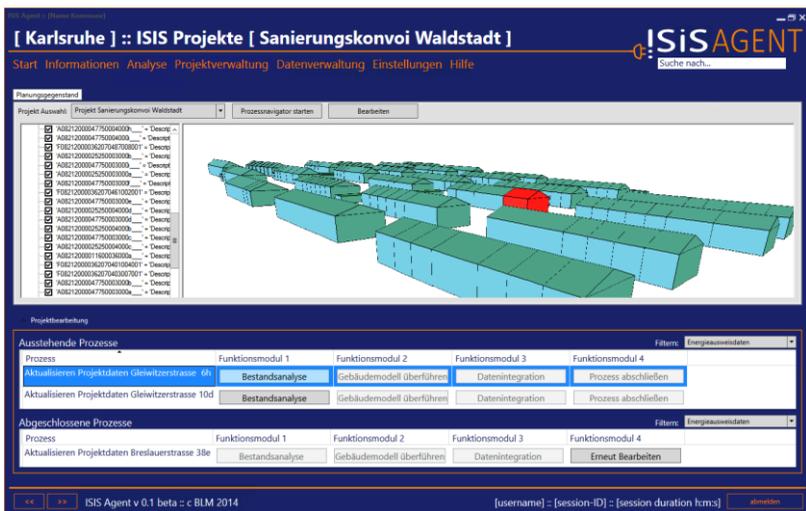


Abbildung 76 ISISAgent – Projektmodul mit eingelesenem Stadtquartiersmodell

eines Schrittes im dargestellten Arbeitsvorgang für jedes Gebäude ein dadurch aufgerufenes Funktionsmodul mit entsprechenden Regeln für den konkreten Arbeitsschritt bestückt.

Die thematische Ausgangsbasis für diesen beschriebenen, implementierten Prozessschritt *Auswertung detaillierter Bestandsaufnahme* im szenariobasierten Rahmen des Quartierssanierungsprojekts wird durch eine Förderung von Energieausweisen an die teilnehmenden Hauseigentümer gebildet. Im Kontext des Szenarios repräsentieren die oben beschriebenen IFC-bezogenen Beispieldaten dabei die zur Ausstellung der Ausweise erforderlichen energierelevanten Bauwerksdaten, die ein Energieberater als neuen Datensatz jeweils zum Gebäude erstellt. Diese gegenüber den „alten“ Gebäudedaten im Stadtmodell (vgl. Abbildung 77) aktualisierten energetischen Gebäudeinformationen können mittels eines Vorgangs zu jedem Gebäude in den Quartiersausschnitt des

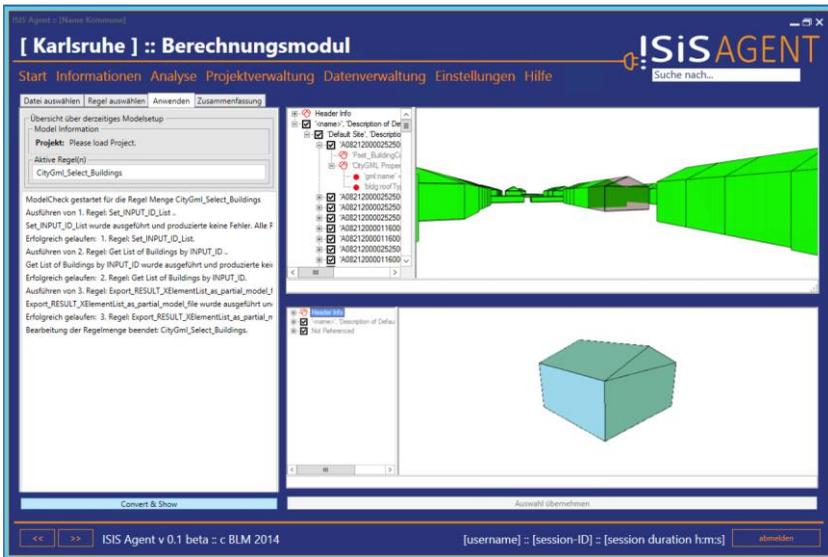


Abbildung 77 ISiSAGENT - Funktionsmodul für mehrskalige Analysen Projektgebietes und damit zur Aktualisierung in das Stadtmodell übernommen bzw. zu integriert werden.

Indem die Sacharbeiter*innen in einem ersten Schritt zu diesen Übernahmen der aktualisierten Gebäudedaten die Funktionsmodul-Schaltfläche *Bestandsanalyse* zu einem Gebäude des Projektgebiets bedienen, wird zur Ansicht des Berechnungsmoduls geleitet (Abbildung 77). Das Modul wird dabei bezüglich der spezifischen Gebäudedaten mit vorkonfigurierten Einstellungen geladen, z.B. eine Regel zum Filtern des Gebäudes im Stadtmodell anhand seiner Adressdaten. Wie beim Dissertationsprototyp der vorliegenden Arbeit sind im aufgerufenen Modul in einem Registerkarten-Oberflächenelement verschiedenen Werkzeuge zum Ausführen von Regeln auf ein Produktmodell angeordnet. Die Werkzeuge sind dabei bereit bis zum Reiter *Anwenden* durch die kontextuellen Informationen ausgeprägt: das entsprechende Gebäudemodell ist geladen und Regeln zum Überprüfen dieses Bestandsgebäudes sind ausgewählt. Ohne weitere Einstellungen vornehmen zu müssen, können die Anwender*innen daher die kontextbezogen geladene Regel auf das geladene Gebäudemodell ausführen und das (optional einsehbare) Ergebnis der Kontrolle der Bestandsdaten mit der Schaltfläche *Auswahl übernehmen* bestätigen. Dadurch wird dieser Arbeitsschritt abgeschlossen, das Funktionsmodul geschlossen und der Nutzer wieder in das Projektmodul zurückführt (vgl. Ebertshäuser & Both, 2016).

Der zweite Schritt *Gebäude überführen* startet in einem eigenen Prozess automatisch das FZK Viewer Werkzeug und übergibt dabei das vorliegende³³ IFC Modell. Nach dem im externen Tool die Transformation des IFC Gebäudemodells zu einem äquivalenten CityGML Gebäude getätigt und als Datei abgelegt wurde, schließen die Anwender*innen das Programm und werden zurück ins Projektmodul geführt. Zum aktuell bearbeiteten Gebäude ist dort dann der dritte Bearbeitungsschritt mit der Schaltfläche *Datenintegration* freigeschaltet. Äquivalent zur *Bestandsanalyse* wird damit zu einem entsprechend vorkonfigurierten Funktionsmodul zum Integrieren der transformierten Daten in das Stadtmodell weitergeleitet. Dabei wird eine Gegenüberstellung von den im ersten Funktionsmodul sondierten Bestandsdaten und den aktualisierten Informa-

³³ gemäß dem Szenario durch Energieberater in der Plattform eingestellter, mit energierelevanten Daten angereicherter aktueller Datensatz

tionen zur abschließenden Kontrolle des alten und neuen Datenstandes angezeigt. Bei Bedarf können hier weitere Überprüfungen mit selbst gewählten bzw. ad hoc in der Regelkonfiguration angepassten Regeln aus der Bibliothek angestoßen werden (vgl. ebd.).

Dabei besteht die Möglichkeit des mehrskaligen Zugriffes auf die Modelle (original IFC- bzw. transformiertes CityGML-basiertes Gebäudemodell sowie gesamtes Stadtmodell) im Rahmen komplexer Analyseregeln. Beispielsweise einen vorgegebenen Sollwert, wie einen für das Projektgebiet im (externen) Stadtmodell festgelegten mindestens zu erreichenden U-Wert der Bauwerks-hülle (vgl. den im gleichen Modell abgelegten Sollwert in Kap. 4.1, 4.5 sowie 6.2.3), aus einem Modell zur Prüfung des die Detailinformationen tragenden IFC-basierten Abgabemodells, bevor dessen spezifischen energierelevanten Ergebnisdaten (z.B. der Endenergieverbrauch) dann in seine CityGML-basierte Repräsentation im Stadtmodell übertragen werden. Neben der kontextbezogenen Einbindung als jeweils vorkonfigurierte, ausführbereite Module in einem geführten Arbeitslauf, steht den Anwender*innen in einem eigenständigen Analysemodul der ISISAgent Plattform auch dem Ansatz der vorliegenden Arbeit folgende Werkzeuge zur eigenen Erstellung bzw. Anpassung und Ausführung von Regeln auf beliebige (mehrschalige) Modelle zur Verfügung. Damit können die Anwender*innen auch die kontextbezogen in den Modulen eingebundenen Regeln (weiter-)entwickeln bzw. anpassen und jeweils als in einem Container gekapselte Regelbasis für sowohl das Projektmodul als auch die Funktionsmodule zur *Bedarfsanalyse* bzw. *Datenintegration* aufbereiten.

8 Schlussbetrachtung

Das in der vorliegenden Arbeit verfolgte Ziel spannt durch den disziplinübergreifenden Charakter über ein weites thematisches Feld, von der Planungsmethodik auf den Skalenebenen Stadt und Bauwerk zur objektorientierten Programmierung und Datenmodellierung im Bereich der Informatik. Vor dem Hintergrund einer Digitalisierung, die der Planungswelt rund um die gebaute Umwelt ein sich schnell wandelndes digitales Instrumentarium an die Hand legt, gilt es die Lösungen der Arbeit kritisch zu reflektieren. Erkannte Verbesserungspotentiale des Erreichten werden dabei aufgezeigt. Daher sind die vom Autor angesetzten Konzepte, Methoden und schließlich umgesetzten Lösungen in diesem Kapitel nochmals bezugnehmend auf Problem- und Zielstellung zusammenzufassen. Dabei ist der ggf. aus isolierter Sicht der einzelnen Disziplinen nicht direkt ablesbare Mehrwert der Ergebnisse der Arbeit aus übergeordneter Perspektive auszuleuchten. Durch die Ergebnisse der Arbeit ermöglichte weitere bereits nachgegangene sowie angedachte Lösungswege werden schließlich aufgezeigt und ein Ausblick auf zukünftige Weiterentwicklungsmöglichkeiten hierzu gegeben.

8.1 Zusammenfassung

Die vorliegende Arbeit ist im Kontext der Entwicklung und Bereitstellung integraler Planungshilfsmittel für die Planer- und Entscheider*innenebene verortet. Sie beschäftigt sich mit der Fragestellung, wie Ergebnisse, die aus Simulation und Berechnung modellbasiert in den Planungsprozess zurückgeführt wurden, fach- und aspektübergreifenden Analyse- und Auswertungsprozessen zugänglich gemacht werden können. Dabei erfolgt im Rahmen der Dissertation die Konzeption und Umsetzung einer IT-gestützten planerischen Entscheidungsunterstützung für ein modellbasiertes Arbeiten im Planungsprozess. Vor dem Hintergrund eines durch die Digitalisierung stark im Wandel begriffenen Planungsbereiches bezüglich der gebauten Umwelt liegt der methodische Ansatzpunkt im Vorgehensparadigma einer *Integralen Planung*. Wegen einer

schwer erfassbaren und generalisierbaren, derzeit in der planerischen Praxis heterogen verlaufenden Erprobung eines konkreten Umgangs mit den neuen digitalen Methoden, wie beispielsweise dem Bestimmen geeigneter Datenstrukturen für einen absprachelosen Informationsaustausch im Rahmen der BIM Methode, knüpft die Arbeit damit methodisch an prinzipiellen Vorgehensweisen zum Optimieren der Planungslösung an, die impliziert werden durch die rahmende Planungsmethodik. Im Ausgangspunkt des gewählten Lösungsansatzes stehen die methodischen Hilfsmittel einer integralen Planung. Diese stehen dabei im Kontext eines zentralen anforderungsorientierten Regelmechanismus (von Both & Ebertshäuser, 2018) zur iterativen planungsbegleitenden Überprüfung und Optimierung der Planungslösung anhand der verfolgten (strategischen) Zielstellungen im Verlauf des Prozesses. Dieses methodische Kernprinzip einer (vgl. Abbildung 15) integralen Planung bildet auch den generalisierbaren Anwendungsrahmen des Lösungsansatzes zur regelbasierten Entscheidungsunterstützung. Dabei baut der Lösungsansatz auf eine durchgängige ganzheitliche digitale Informationsbasis als einem Grundbaustein einer integralen Planung mit Bezügen zu offenen in der Planung etablierten Produktmodellstandards auf. Da den meisten etablierten (Planungs- bzw. Datenaustausch-) Modellen dabei mit der XML eine einheitliche Syntaxbasis unterliegt, können dann mit zusammengestellten Einzelmodellen bedarfsgerecht (mehrschalige) Systemmodelle als Grundlage ganzheitlicher Analysen und Prüfungen des Planungsgegenstands aufbereitet werden. Auf dieser Basis sieht der Lösungsansatz Methoden und Konzepte vor, die es den Planenden ermöglichen, ihre (situativen) Fragestellungen an ihre Planungslösungen oder -alternativen zu stellen und damit prinzipiell die Einhaltung ihrer fachlichen wie auch formalen Zielstellungen transparent und nachvollziehbar zu prüfen bzw. insgesamt das Planungsergebnis zu optimieren. Hierfür beschreibt der Lösungsansatz das technologieneutrale Prinzip einer Softwareanwendung zur regelbasierten Analyse und Prüfung von Produktmodellen (vgl. Kap. 3.1), um in Planungssituationen fachliche Entscheidungsgrundlagen aufbereiten und damit die Planenden unterstützen zu können. Damit die Planungshilfestellung ohne technisches Wissen direkt zur Entscheidungsunterstützung angewendet kann, sieht der Lösungsansatz eine rollenbasierte Arbeitsteilung vor: eine Rolle mit technischer Expertise bereitet die Erstellung situationsspezifischer Entscheidungsgrundlagen in Form von eigenständigen Regeldateien vor. In

der Entscheidungssituation können diese Regeln dann von einer Rolle ohne technische Expertise auf die als Produktmodell repräsentierten Planungsgegenstände ausgeführt werden.

Die Prüf- bzw. Analyselogik der Produktmodelle soll von den designierten Anwender*innen aus dem Bereich der Planung selbst erstellt werden können, da sie das nötige fachliche Wissen zu den relevanten Aspekten der Entscheidungssituation besitzen. Damit auch ohne profunde technische Expertise bei der Vorbereitung der Modellprüfung bzw. -analyse mitgewirkt werden kann, sieht der Lösungsansatz bei der Formalisierung der Prüf- bzw. Analyselogik in die Regeldateien verschiedene Stufen an erforderlichen (technischen) Vorkenntnissen vor. Eine obligatorische Voraussetzung für die Bearbeitung der Regelinhalte stellt dabei eine technisch-fachliche Kenntnis der Strukturen und Inhalte von den behandelten Produktmodellen dar. Zwei intermediäre Stufen an erforderlicher technischer Fachkenntnis der Regelbearbeitung sind gemäß dem Lösungsansatz dann einer vollumfänglichen Programmierkenntnis vorangestellt. Durch die beiden aufeinander aufbauenden Stufen wird die darunterliegende Komplexität der Programmiersprache verdeckt und somit ein Anpassen von den Regeln auch programmiertechnisch nicht versierten Anwender*innen ermöglicht. Mit der Möglichkeit, selbstvergebene Schlüsselwörter als Platzhalter in Lückentexten einer zur Abfrage der Modellinhalte verwendeten Skriptsprache anzuwenden, können in der ersten Stufe allein durch Kenntnis der domänenspezifisch bzw. im Bürokontext verwendeten Semantik Regelinhalte angepasst werden. Erst in der zweiten Stufe kommen dann durch die erforderliche Kenntnis von Skriptausdrücken einer Modellabfragesprache (vgl. OCL) ein Merkmal eines Expertenwerkzeuges im Rahmen Regelbearbeitung hinzu. Am gestuften Aufbau der Regelinhalte setzen im Umsetzungsteil der Arbeit rollenbasierte Bearbeitungsstrategien zur Entwicklung von aufeinander aufbauenden Regelinhalten an, die das Ziel verfolgen, durch eine möglichst hohe Wiederverwendbarkeit entwickelter Regeln einen benötigten Vorbereitungsaufwand zur Anwendung des implizierten regelbasierten Planungswerkzeugs möglichst gering zu halten.

Zur Umsetzung des Lösungsansatzes in eine Softwareanwendung zur Unterstützung von planerischen Entscheidungssituationen sind darin enthaltene Methoden und Konzepte weiterführend ausformuliert. Zunächst werden dazu umsetzungsrelevante Anforderungen, wie die Kenntnisstufen oder der entsprechend gestufte Aufbau der Regelinhalte, für das regelbasierte Grundprinzip des Ansatzes in Kapitel 3.2.1 konkretisiert. Im Lösungsansatz sind neben diesen Konzeptbestandteilen auch bereits (technische) Aspekte enthalten, wie das Aufgreifen einer Skriptsprache als intermediäre Vorkennsstufe zur Regelentwicklung. In Kapitel 3.2.2 werden diese gewichtigen technologieorientierten Gesichtspunkte in Bezug auf die konkretisierten Anforderungen des Lösungsansatzes auf der Ebene der Prüf- und Analyselogik ausgeführt. Zudem werden sowohl technische Anforderungen an das *Produktmodell* als Gegenstand der designierten Softwareanwendung wie auch dessen (kooperative) Handhabung im Sinne diesbezüglicher Softwarefunktionalitäten dargestellt. Aufgrund des Ansatzpunktes der Dissertation in einem bezüglich einer modellbasierten Arbeitsweise sich im Wandel befindlichen Kontextes werden in Kapitel 3.2.3 konzeptionelle Festlegungen an die (idealisierte) Arbeitsumgebung des Planungshilfsmittels zusammengestellt. Für eine optimale Einsatzmöglichkeit des Softwareanwendungsansatzes werden darin idealtypische Bedingungen des designierten Arbeitsumfelds mit Bezug zu Prozessen der gängigen Praxis eingeordnet. Zum Beispiel ist eine wesentliche Prämisse zum Schöpfen des Mehrwertes aus dem Lösungsansatz bezüglich kooperativ genutzter Prüf- und Analyselogik das Verwenden offener herstellerneutraler Austauschformate (vgl. *Open BIM*) für die absprachelosen Datenübergaben im Planungsprozess.

Damit im Rahmen der Dissertation eine prototypische Software als Entscheidungshilfsmittel umgesetzt werden kann, bedarf es zusätzlich zum Lösungsansatz konkretisierter Anforderungen und technischen Festlegungen wie auch einer thematischen Abgrenzung eines Anwendungsfalles (vgl. bspw. User Story in der Softwareentwicklung). Bedingt durch den modellübergreifenden technischen Ansatz zur Informationsverarbeitung sowie den anwender*innen-orientierten, regelbasierten Erweiterungsansatz der Prüf- und Analysefunktionalitäten im Softwareansatz ermöglicht erst ein abgegrenztes Anwendungsszenario eine konkrete Umsetzung. Anhand eines Anwendungsfalles einer planerischen

Entscheidungssituation im Rahmen eines Projektszenarios werden in Kapitel 4 daher alle thematischen Rahmenbedingungen und Anforderungen für die Implementierung des Dissertationsprototyps konkretisiert. Insbesondere werden dabei aufeinander aufbauende Anforderungen als Basis der Entwicklung von Prüf- und Analyselogik für zwei als Produktmodell repräsentierte Planungsalternativen hergeleitet. Die Anforderungen an den Planungsgegenstand betreffen dabei zum einen die Modellqualität (vgl. Kap. 4.3), zu der sie formale bzw. strukturelle Aspekte bestimmen, wie die Art der Dateninhalte sowie ihre Zusammensetzungen. Zum anderen betreffen hierauf aufbauende Anforderungen eine fachliche Bestimmung der (planerischen) Bauwerksqualität (vgl. Kap. 4.1) mit beispielsweise energetischen Kennwerten zu den Bauteilen. Letztere tragen dabei die für das Aufbereiten von Entscheidungsgrundlagen benötigten Kriterien respektive der entsprechenden modellbasierten Basiswerte. Im szenariobasierten Rahmen des Anwendungsfalls werden diese Kriterien dann anhand entsprechend zu entwickelnder Modellprüfung bzw. -analyse an den Planungsalternativen ausgewertet (vgl. Kap. 4.2). Die Vorbereitungsschritte für die Erstellung und Ausführung von Prüf- bzw. Analyseplänen (vgl. Kap. 4.4) bilden dabei methodisch die Basis der Entscheidungsgrundlagen zur Unterstützung der Planenden bei der Auswahl einer der beiden Alternativen. Alle vor dem thematischen Hintergrund dieses dargestellten Anwendungsfalls stehenden modellbasierten Anforderungen werden abschließend in einem Lastenheft zusammengefasst (vgl. 4.5). Alle entwicklungstechnischen Aspekte sind damit hinreichend spezifiziert, so dass dann die fachlich neutral umzusetzende Softwareanwendung mittels der Entwicklung entsprechender Regeln an diesen thematisch konkretisierten Anwendungskontext angepasst werden kann.

Die Umsetzung des regelbasierten Entscheidungshilfsmittels wird in zwei Ebenen dargestellt. Einerseits die Entwicklung der prototypischen Desktop-Anwendung zur regelbasierten Prüfung und Analyse von Planungsmodellen. Und andererseits die (logische) Entwicklung von Regelinhalten zur Anpassung dieses Prototyps an die konkrete Entscheidungssituation. In Kapitel 5 werden auf der technischen Seite der Implementierung der Softwareanwendung die wesentlichen Komponenten der Software beschrieben, ausgehend von den der Architektur zugrundeliegenden Entwurfsmustern. Eingegangen wird dabei auch

auf das Zusammenwirken der z.T. auf Basiskomponenten, die bereits am Institut in einer Komponentenbibliothek BLM ToolKIT vorhandenen waren (vgl. Kap. 5.1.1), aufbauenden Softwareteile im Rahmen einzelner Werkzeuge zur Realisierung einzelner Softwarefunktionen. Im Kern der Funktionalität der Softwareanwendung steht das Verarbeiten der in eigenständigen Regeldateien gekapselten Analyse- und Prüflogik für die Produktmodelle. Vier zu einem geführten standardmäßigen Arbeitsablauf aneinander gekoppelte Werkzeuge der Softwareanwendung ermöglichen es den Anwender*innen Produktmodelle einzulesen, Regeln auszuwählen und diese auf die Modelle auszuführen. Ein Werkzeug zum Zusammenstellen und Ablegen von Ergebnisberichten vervollständigt dann diesen als Standardmodus der Software umfassten Teil der Funktionalität. Dieser bisher auf die Regelanwendung beschränkte Funktionsumfang wird durch weitere hinzuschaltbare Werkzeuge im Rahmen eines erweiterten Softwaremodus mit Funktionen zur Verwaltung und Anpassung der Regeldateien ergänzt. Die im Regelbearbeitungsmodus zusammengefassten (erweiterten) Werkzeuge sind dabei eine *Regelverwaltung* zur Administration der Regeln sowie eine *Regelkonfiguration* zu ihrer inhaltlichen Bearbeitung. Mit diesen zusätzlichen Werkzeugen kann der (fachliche) Funktionsumfang der Softwareanwendung von den Anwender*innen durch das Erstellen einer eigenen Regelbasis im Sinne einer Erweiterung der Geschäftslogik an die Gegebenheiten ihrer konkreten Planungssituationen angepasst werden. Die neutral implementierte Softwareanwendung können die Anwender*innen somit selbst zu anforderungsorientierten (Entscheidungs-) Hilfsmitteln für ihre konkreten Planungsprozesse umgestalten.

Alle Aspekte, welche die Entwicklung der Regelinhalte für den im Projektszenario konkretisierten Anwendungsfall betreffen, werden auf der Ebene der Anpassung der Softwareanwendung auf den Anwendungskontext in Kapitel 6 dargestellt. Dabei werden zunächst beziehend auf (idealerweise) verteilte Anwendung der Software im Planungsbüro bzw. -prozess Strategien zur Kapselung von aufeinander aufbauender Prüf- und Analyselogik in unterschiedlichen Regelbasen erörtert. Für das szenariobasierte Lastenheft werden dann durch Spezifizieren konkreter modellbasierter Prüf- und Analyselogik die inhaltliche Grundlage zum Formalisieren von Regeldateien in Form eines Pflichtenhefts dargestellt. Indem darin alle fachlichen Fragestellungen als

Prüf- bzw. Analyselogik konzipiert und mit Bezug zu den entsprechenden Modellentitäten spezifiziert sind, können anhand dieser inhaltlichen Grundlage mit den Werkzeugen der Softwareanwendung entsprechende Regeln entwickelt werden, die eine Entscheidungsunterstützung in der szenariobasierten Planungssituation ermöglichen. Diese Darstellung der entwickelten Regellogik vervollständigt die Umsetzung des „auf die Situation angepassten“ Prototyps im Rahmen der Dissertation. Als Handlungsanleitung zur Strukturierung und Entwicklung erforderlicher Regeln ermöglicht sie die prinzipiell auf Seite der Anwender*innen verortete Anpassung der Softwareanwendung auf den konkreten Anwendungsfall im Projektszenario. Zur Veranschaulichung der darin spezifizierten Logik von (aufeinander aufbauenden) Formalisierungen in konkrete Regeldateien sowie deren Einordnung in den darin angedeuteten Strukturierungen in Regelbasen sind entsprechende Muster im Dissertationsprototyp hinterlegt.

Eine ausführliche Betrachtung der einzelnen umgesetzten Werkzeuge aus Sicht der Anwender*innen bildet den ersten Teil des Kapitels 7 zur beispielhaften Anwendung und Evaluierung der im Ergebnis der Dissertation resultierenden Softwareanwendung. Dabei werden neben den realisierten Benutzer*innenführungen, die Interaktionsmöglichkeiten und Funktionalitäten der Bedienoberflächen dargestellt und erläutert. Bezugnehmend auf die technische Entwicklung sowie die entwickelte Regellogik in der Umsetzung des Prototyps kann anhand der dargestellten Bedienoberflächen somit auch das Erstellen bzw. Anpassen der Software an den szenariobasierten Anwendungskontext (vgl. Kap. 4) nachvollzogen werden. Ergänzend zu dem im Rahmen der Dissertation vor dem Hintergrund des Projektszenarios implementierten Prototyp werden in Kapitel 7.2 weitere Umsetzungen des erarbeiteten Lösungsansatzes, sowie die Anpassung auf entsprechende Anwendungskontexte aus verschiedenen (Forschungs-) Projekten dargestellt. Damit kann der durch die szenariobasierte Abgrenzung nur teilweise ermöglichte Praxisbezug der prototypischen Softwareanwendung weiter verdeutlicht werden. So kann beispielsweise die Möglichkeit zum selbständigen Anpassen von Prüflölogik zu Dokumentationsvorgaben an dem zur Prüfung der BIM Richtlinien der Öffentlichen Hand an das BBR ausgelieferte Prüfwerkzeug BBR ModelCheck veranschaulicht wer-

den. Des Weiteren kann beispielsweise eine weiterführende technische Umsetzung anhand einer kontextsensitiven Einbindung des Softwareansatzes in einer übergeordneten Plattform gezeigt werden. Dabei sind in einer rahmengebenden Prozessplattform verschiedene, den jeweiligen Prozessschritten bedarfsorientiert zugeordnete Regeln hinterlegt. Die in diesem Forschungsprojekt u.a. als Hintergrundmodul implementierte Softwarelösung zur Prüfung und Analyse von Stadtmodellen (vgl. Kap. 7.2.3) in Verbindung zu einzelnen Gebäudemodellen wird dabei dann jeweils zu einem Bearbeitungsschritt mit den im Kontext befindlichen Regeln (automatisch) konfiguriert.

8.2 Reflexion und Ausblick

Kritische Reflexion

Die vorliegende Arbeit widmet sich der Thematik IT-gestützter Entscheidungsunterstützung im Kontext einer integralen Planung der gebauten Umwelt. Freilich klammert sie dabei – da bekanntermaßen im Rahmen einer Dissertation nur ein Ausschnitt des angeschnittenen Themenkomplexes behandelt werden kann – andere Themen aus. Auf diese dabei für die Arbeit nicht unbedeutende, ggf. sogar wichtige jedoch nicht mitaufgenommene Themen und ihre Bedeutung im Rahmen der Arbeit soll daher im Folgenden (kurz) hingewiesen werden. Damit wird einer zusammenfassenden Bewertung des in der vorliegenden Arbeit erreichten (methodischen sowie technischen) Ergebnisses eine Zusammenstellung der in der Arbeit nur teil- bzw. auszugsweise umgesetzter bzw. unberücksichtigter Gesichtspunkte aus Sicht des Autors vorangestellt. Beleuchtet werden sollen insbesondere aus heutiger Sicht – nach 6 Jahren der Bearbeitung des Themenfeldes – dabei erkannte verbesserungswürdige Aspekte in der Arbeit – ausgehend von den Thesen und Annahmen, auf denen der Ansatz beruht, bis hin zu den technisch ergriffenen Mitteln der Umsetzung als Softwareprototyp einer regelbasierten Entscheidungshilfe.

Damit die im Weiteren dargestellte kritische Reflexion richtig eingeordnet werden kann, erscheint es wichtig zu berücksichtigen, dass der Ansatzpunkt der Arbeit vor dem Hintergrund einer voranschreitenden Digitalisierung der

Planungs-, Ausführungs- und Betriebspraxis mit einem sich rapide im Wandel begriffenen Instrumentarium der Akteure stattfindet. Bezugnehmend zu dem in der Arbeit verfolgten (möglichst) auf Standards aufbauenden Lösungsansätze lagen beispielsweise Lösungen wie die modellspezifische Abfragesprache für virtuelle Bauwerksmodelle mit der BIM Query Language (vgl. Mazarac & Beetz, 2013) zu Beginn der Arbeit noch nicht vor und sind erst zu einem Zeitpunkt veröffentlicht worden, als konzeptionelle Weichen für diese Dissertation bereits festgelegt waren und die daher nicht mehr einbezogen werden konnten. Dies gilt auch für zahlreiche IT-gestützte Hilfsmittel, die für den spezifischen Rahmen einzelner Produktmodelle entwickelt wurden, bezüglich der inhaltlichen Prüfung von Produktmodellen beispielsweise standardorientierte Softwarelösungen mit Kohärenzprüfungen auf Basis des Spezifizierungsstandards MVD. Diese nutzen die inhärente Regelsprache im MVD Standard, um damit definierte Sachverhalte in IFC-basierten Modellen überprüfbar zu machen, wie u.a. (vgl. Zhang u. a., 2014) auch in den neuesten Entwicklungen um den FZK Viewer (vgl. Geiger & Hagenmeyer, 2018).

Die in der vorliegenden Arbeit vor dem Hintergrund gängiger Standards und Methoden erstellte Softwareanwendung wurde dabei auch unter der Zielstellung verwirklicht, komplexe modellbasierte Aufgabenlösungen (technisch) nicht versierten Anwender*innen zugänglich zu machen. Durch das bedarfsorientierte Bereitstellen entsprechender Regeldateien lässt sich dies auch mit dem im Standardmodus der Software implementierten *geführten* Arbeitsgang bewerkstelligen. Ein durch den gestuften Ansatz der in eigenständigen Regeldateien persistierten Prüf- und Analyselogik auf Seiten der Regelentwicklung umgesetzter Administrations- und Bearbeitungsmodus, erfordert jedoch zumindest modellierungstechnische Vorkenntnisse zu objekt-orientierten Produktmodellen, in denen der jeweilige (Planungs-) Gegenstand abgebildet ist. Auch für diesen Teil der Softwareanwendung wurde mit der Möglichkeit, durch selbstvergebene Schlüsselbegriffe eine domänen- bzw. unternehmensspezifische Regelsemantik in die Entwicklung eigener Regelbasen einzubinden, eine Grundlage geschaffen, auf der den technisch nicht versierten Anwender*innen das Anpassen der Inhalte ermöglicht werden kann. Durch die erforderliche Modellkenntnis sowie ein benötigtes klares Verständnis der De-

komposition einer Regellogik in die vier standardmäßigen Komponenten tendiert die Bedienung der Softwareanwendung auf Seite der Regelentwicklung allerdings mehr zu einem Expertenwerkzeug (vgl. auch Kap. 6.1 sowie Kap. 7.1). Darüber hinaus wird mit dem Aufgreifen der Verben von der an den OCL Modellabfragestandard angelehnten .NET LINQ Komponente zum Verfassen von Skriptausrücken, die mittels Constraints (Randbedingungen) ein eindeutiges Bestimmen von Modellabfragen ermöglichen, eine einfach zu erlernende „Zwischenstufe“ vorausgesetzter Informatikkenntnisse zur Regelentwicklung dabei verfolgt. Da anstelle von standardisierten OCL-Ausdrücken hierbei der LINQ „Dialekt“ zur Umsetzung der dynamischen Schlüssel-Wert-Paare in einer Regel (vgl. Kap. 5.2.2.4) verwendet wurde, erfordern die ursprünglich im Ansatz (herstellerneutral vorgesehenen) standardbasierten Vorkenntnisse im umgesetzten Ergebnis der Arbeit Kenntnisse zu den in der .NET Programmiersprache C# integrierten LINQ Komponente. Als eine bessere Lösung zum im Lösungsansatz angestrebten Aufbau dieser Zwischenstufe allein auf dem OCL Standard erscheint das Einbinden von den auf informatischer Seite verfolgten Ansätzen zur sprachenübergreifenden Transformation der nativen OCL Ausdrücke in die Programmiersprache (vgl. z.B. Vajk & Mezei, 2010). Hierauf aufbauend wäre dann auch beispielsweise eine stringendere Trennung der Belange bezüglich weiterer zur Optimierung der Performance einer Regelausführung auf ein Produktmodell in die LINQ Ausdrücke eingeflossener Programmierlogik möglich gewesen. So sind in der umgesetzten Lösung in LINQ Ausdrücken mit aufwendiger, arbeitsintensiver³⁴ Rechenleistung selektiv einzelne Methodenaufrufe zur Parallelisierung „eingestreut“, womit durch die (technologieabhängige) .NET Komponente Parallel LINQ (PLINQ) die Rechnerarbeit dann auf die (ggf. vorhandenen) verschiedenen Prozessoren des die Software ausführenden Desktop-Computers aufgeteilt wird.

Auch auf der Ebene der Bearbeitung des Quelltexts einer Regel in Form der Programmiersprache C# wäre aus heutiger Sicht ein konsequenter formalisierterer Ansatz einer einheitlicheren Vorlage, der die beliebigen Beschreibungsmöglichkeiten zugunsten eines strenger reglementierteren Regelrumpfes eingeschränkt hätte, sinnvoller gewesen. In der Arbeit wurde mit einer einfachen

³⁴ wie z.B. eine durch alle Modellentitäten iterierende Schleife

Umsetzung der Regelkomponenten ein „konventionsgetriebener“ Ansatz einer Codeschablone umgesetzt, der zwar einen ersten funktionalen Ansatzpunkt zum Formalisieren darstellt, jedoch durchaus im Sinne einer formalisierten Regelsprache/Notation verbessert werden kann. Durch eine auf einem formalen Rahmen basierend formulierte Regellogik könnte dann die maschinenverarbeitbare Form beispielsweise bezüglich einer Syntaxprüfbarkeit bzw. einer Wiederverwendbarkeit optimiert werden. Bezugnehmend zur Generalisierbarkeit der Regelinhalte wurde ebenfalls in Konvention zum Regelaufbau eine einheitliche Grundlage des Modelldatenzugangs mit dem Grundansatz zum alleinigen Bedienen von XML-basierten Modellformaten eingeschränkt. Damit konnte mittels einer auf der gemeinsamen XML-Syntaxbasis aufbauenden Parserkomponente eine einheitliche Eingabemaske zum Bestimmen der Modellinhalte für verschiedene, mehrskalige Produktmodelle in der Regellogik den Anwender*innen zur Verfügung gestellt werden. Dies wurde im Ergebnis der Arbeit mit der LINQ-to-XML Programmierschnittstelle der .NET LINQ Komponente zur Bindung der XML-Dokumente an eine dem DOM Standard angelehnte interne Repräsentation als Objektgraf umgesetzt. Neben den bereits von den OCL Standardverben abweichenden Ausdrücken der LINQ Komponente bedeutet diese umgesetzte Eingabeschnittstelle zum Formalisieren der Regelinhalte jedoch ein weiteres Abweichen von dem ursprünglich auf eine vollständige standardbasierte Regelbeschreibung abzielenden Lösungsansatz. Den Regelentwickler*innen werden damit auch Kenntnisse zu den von den im W3C DOM Standard festgelegten Begriffen abweichenden (technologiegebundenen) Ausdrücken der Programmierschnittstelle abverlangt. Diese beiden mit technologiegebundenen Lösungen umgesetzten Beschreibungsfazilitäten der Regelinhalte realisieren die ursprünglich avisierte „allein auf Standard basierte“ Zielstellung nur teilweise. Eine zielorientiertere Realisierung stellt hierzu eine in der Gesamtarchitektur zwar veranlagte jedoch nicht implementierte Erweiterung mit entsprechenden Komponenten zur Bedienung des Standards OCL (bzw. wie im Ausblick bezüglich XML nativen Datenbanken dargestellten XQUERY; vgl. Kap. 8.2) sowie des Standards DOM dar. Im Rahmen der Arbeit wurde dies aufgrund einer erheblich vereinfachten Implementierung des regelbasierten Ansatzes mittels der integrierten .NET Komponenten und darüber hinaus einer gegenüber alternativen Lösungen vielfach

besseren Performance bei der Verarbeitung der Regeldateien jedoch nicht weiter umgesetzt.

Bei der Softwareentwicklung im Rahmen der Arbeit wurde mit dem Umsetzen einer generellen Schichtenarchitektur ein typischerweise Desktop-Anwendungen unterlegtes Entwicklungsprinzip verfolgt. Durch ein Unterlegen des Softwaredesigns sowie der Entwicklung mit weiteren Entwurfsmustern wurde dabei durchgehend versucht, im Sinne einer objektorientierten Programmierung das Prinzip der Trennung der Belange einzuhalten. Neben den gelungen umgesetzten Ansätzen, mittels eines Repository Entwurfsmusters eine lose Kopplung der Datenzugangsebene an die persistenten Daten zur Wahrung der Skalierbarkeit zu bewerkstelligen, erscheint die Umsetzung der Bedienoberfläche mit dem .NET Windows Presentation Framework (WPF) sowie dabei ange-setzte Entwurfsmuster zur Trennung der Oberfläche von den Geschäftsprozessen- und -daten noch erhebliches Verbesserungspotential (vgl. Kap. 5.2.1) zu beinhalten. Eine strikte Einhaltung der Trennung zwischen der UI und der BL Schicht ist zwar schon aufgrund der inhärent im .NET WPF Ansatz verwobenen Programmlogik erschwert, wäre jedoch mittels einer Umsetzung eines Inversion of Control (IoC) Ansatzes bzw. des Einziehens eines zusätzlichen Service Layers einer besseren Lösung zuführbar als im Ergebnis der Arbeit. In der umgesetzten Form besteht entgegen dem Prinzip einer Entkopplung der UI Schicht durch die App Klasse eine Abhängigkeit auf die BL Schicht (vgl. Kap. 5.2.1). Prinzipiell bildet die Bedienoberfläche die Interaktionsmöglichkeiten der zugrundeliegenden Funktionskonzeptionen ab. Die erreichte Softwarelösung könnte jedoch insgesamt durch mehr anwendungsbezogene Oberflächenkonzepte im Sinne einer ergonomischer gestalteten Bedienbarkeit verbessert werden. Aufgrund erheblicher Aufwände bei der Oberflächenentwicklung wurden hierzu zwar einzelne Features wie ein Drag'n'Drop umgesetzt, für ein durchgängiges effizientes Arbeiten mit der Software wären jedoch auch bei den hauptsächlich durch Textfelder geprägten Eingabemasken dieses Arbeiten mit dem Mauszeiger (bzw. mit Gestensteuerung beim Touchpad) vorteilhaft. Auch die Informationsdarstellung könnte durch weniger Text(eingabe)felder geprägt und mit geringerer Einbindung der WPF-Standardoberflächenelemente, wie dem Registerkarten-Steuerelement usw., ansprechender gestaltet und übersichtlicher aufbereitet werden. Hier wurde im Ergebnis der Arbeit auf

eine *funktionale* (prototypische) Lösung geachtet und ein die Nutzungserfahrung steigerndes, eigenständiges Oberflächenkonzept aufgrund der sehr aufwändigen Programmierung und Testung nicht weiterverfolgt. Gerade für technisch nicht versierte Anwender*innen wären jedoch beispielsweise ein modernes Dashboard zur Übersicht der Prüfungsergebnisse hilfreicher als die umgesetzte Konsole für die Ausgabe der Ergebnisse.

Insgesamt konnte mit dem in der Arbeit dargestellten Ausschnitt der planerischen Praxis nur ein kleiner Anwendungsbereich für die Lösung ausgeleuchtet werden. Zur Unterstützung einer einzelnen Entwurfsentscheidung im Rahmen des Planungsprozesses ist durch das Entwickeln benötigter Regeln ein sehr hoher Aufwand erforderlich. Erst durch die beispielhaft dargestellten Arbeitsroutinen und einem iterativen mehrfachen Einsatz der Softwareanwendung im Rahmen des Planungsprozesses bzw. im Büroalltag erscheint der einhergehende Aufwand bezüglich des Nutzens verhältnismäßig zu sein. Von einer durchgängig für den Anwendungsfall in Kapitel 4 spezifizierten Regellogik, die in einzelnen Regeleinheiten aufeinander aufbaut, konnten wegen des hohen Aufbereitungsaufwandes nur beispielhaft Regeldateien in den designierten Regelbasen zur Anpassung des Prototyps auf den Anwendungsfall zur Entscheidungsunterstützung umgesetzt werden. Eine umfängliche Evaluation der Lösung konnte damit nur in Teilen und unter Zuhilfenahme bereits im Kontext anderer vom Autor erarbeiteter prototypischer Umsetzungen des Lösungsansatzes der Arbeit im Rahmen weiterer Projekte dargestellt werden. Die Arbeit wurde auf der Nahtstelle einer integralen Planung und der Informatik ange-setzt. Daher vermag erst durch einen disziplinübergreifenden Standpunkt betrachtet, die – sowohl auf Ebene des Hilfsmittels als auch bei dessen Anwendung – auf eine Integration informatischer Themen in den Bereich der Planungsmethodik hinwirkende Arbeit einen wesentlichen Mehrwert darstellen zu können.

Ausblick

In laufenden Forschungsprojekten finden derzeit weitere Implementierungen des in der Arbeit dargestellten Lösungsansatzes, respektive entsprechender regelbasierter Anpassungen auf ihre (thematischen) Anwendungskontexte statt. Dabei wird neben dem dokumentenbasierten Modelldatenzugang auf der datentechnischen Seite auch eine Anbindung nativer XML-Datenbanken mit inhärenten Programmierschnittstellen auf Basis des Abfrageskriptsprachstandards XQUERY untersucht. Die Kapselung der Regellogik in eigenständigen Dateien ermöglicht es dabei, entsprechende Prüf- und Analyselogik auch für besonders umfangreiche Modelle bzw. Gruppen an Modellen vorzuhalten, deren vollständiges Einlesen in den Arbeitsspeicher des Computers als Objektgraphen nicht mehr effizient handhabbar wäre. Diesbezüglich werden die Prüfung und Analyse mehrskaliger Modelle, die bereits im Lösungsansatz der vorliegenden Arbeit veranlagt sind, in dem im Kontext der Stadtplanung verteteten Verbundprojekt EnEffStadt ModelSIM weiter untersucht. Stadtmodell-daten werden dabei bezüglich der Vollständigkeit zur Verwendung bei der Parametrisierung auf Seiten der energetischen Quartierssimulation mit der objektorientierten Simulationssprache Modelica überprüft. Dabei wird auf die Nutzung von typologischen Daten ein modellierungstechnischer Fokus gelegt zur Vervollständigung der oftmals lückenhaften kommunalen Datengrundlage (vgl. Ebertshäuser & von Both, 2019). Auf Basis der in das Stadtmodell zurückgeführten Ergebnisse der Quartierssimulation setzen dann weitere regelbasierte Analysen zur Unterstützung der Stadtplaner*innen bei ihren Aufgaben in der Stadtentwicklung an (vgl. Ebertshäuser u. a., 2018).

Bezugnehmend zu regelbasierten Modellprüfungen und -analysen im Lösungsansatz der Arbeit liegt in einem weiteren Forschungsprojekt ein Schwerpunkt auf der Behandlung typologischer Daten bzw. Referenzwerte zum Forschungsgegenstand im Themenfeld des Nachhaltigen Bauens. Allerdings werden hierbei anstelle von Leerstellen bzw. „Gaps“ in kommunalen Bestandsanalysen damit unscharfe Datengrundlagen in Planungslösungen der frühen Planungsphasen als Inputparameter für IT-gestützte Planungshilfen ergänzt. Um Simulations- und Berechnungswerkzeuge mit Modellinformationen in diesen frühen Planungsstadien anwenden zu können, werden Systematiken

untersucht, wie sich zu Beginn der Bauwerksplanung naturgemäß noch nicht konkretisierte Bauwerksinformationen durch typische Referenz- und Benchmarkwerte vervollständigen lassen (Rexroth u. a., 2018). Mit einer großen thematischen Schnittmenge zum in der vorliegenden Arbeit behandelten fachlichen Thema der Energieeffizienz stellt die Ökobilanz (LCA) dabei einen weiteren Anwendungskontext (Ebertshäuser u. a., 2019) dar, vor dessen Hintergrund im Verbundprojekt *BIM basierte integrale Planung* (BIM2LCA4IP) der Lösungsansatz der Dissertation weiterführend untersucht und verfeinert werden kann. Dabei werden neben der Kohärenzprüfung von Daten, die in die Werkzeuge der Ökobilanz seitens der modellbasierten Planung übergeben werden, auch Analysen zurückgeführte LCA-Ergebniswerte zur Anzeige der Umweltwirkung untersucht. Durch die im Verbundprojekt verfolgte Zielstellung, Simulations- und Ökobilanzwerkzeugen möglichst frühzeitig in den Planungsprozess einzubinden, sollen Möglichkeiten untersucht werden, wie sich die Qualität der Planungslösung durch das Anzeigen möglicher Umweltauswirkungen von Entwurfsentscheidungen verbessern lässt. Für ökologischen Fußabdruck eines Bauwerks bezüglich seiner Nachhaltigkeit sind besonders die Planungsentscheidungen in den (sehr) frühen Planungsphasen bedeutsam. Neben spezifischen LCA-relevanten Informationen im Rahmen von Nachhaltigkeitsbewertungen werden die im Projekt entwickelten Prototypen für regelbasierte, fachliche Analysen daher auch bezüglich typischen bzw. benchmarkbasierten Referenzwerten zur Entscheidungsunterstützung ausgeprägt. Dieser derzeit auf großes externes Interesse treffende Ansatz soll zukünftig auf den urbanen Maßstab gespiegelt werden, wodurch sich die Komplexität der Prüfungen sodann zum einen bezüglich der Größe der zu prüfenden Modelle in Richtung „Multi-Modell-Analysen“ entwickeln wird. Zum anderen kann der Mehrwert des in dieser Arbeit vorgestellten Lösungsansatzes integrierter regelbasierter Modellanalysen durch die gestiegene fachliche Komplexität im Sinne interdisziplinärer stadtplanerischer Fragestellungen stärker zum Tragen kommen.

Literaturverzeichnis

Achammer, C. M., & Salzburger, A. (2014). *Integrale Planung und BIM*.

Präsentation. Abgerufen von http://www.bim-information.com/wp-content/uploads/2014/12/Achamer_BIM_Kongress_2014_Achammer_Salzburger.pdf

Akehurst, D. H., Howells, W. G., Scheidgen, M., & Mcdonald-Maier, K. D.

(2008). C# 3.0 makes OCL redundant! *Electronic Communications of the EASST, Volume 9*(Ocl4All-Modelling Systems with OCL).

<https://doi.org/http://dx.doi.org/10.14279/tuj.eceasst.9.103>

Beetz, J., Borrmann, A., & Weise, M. (2015). Prozessgestützte Definition von

Modellinhalten. In A. Borrmann, M. König, C. Koch, & J. Beetz (Hrsg.), *Building Information Modeling* (S. 129–147). Wiesbaden:

Springer Fachmedien Wiesbaden. Abgerufen von

http://link.springer.com/10.1007/978-3-658-05606-3_7

Biehal, F. (1994). *Lean Service: Dienstleistungsmanagement der Zukunft für*

Unternehmen und Non-Profit-Organisationen (2., Aufl.). Bern;

Wien: Paul Haupt, Bern.

Biljecki, F., Kumar, K., & Nagel, C. (2018). CityGML Application Domain

Extension (ADE): overview of developments. *Open Geospatial*

Data, Software and Standards, 3(1). <https://doi.org/10.1186/s40965-018-0055-6>

BMUB, B. für U., Naturschutz, Bau und Reaktorsicherheit (Hrsg.). (2016).

Leitfaden Nachhaltiges Bauen : Zukunftsfähiges Planen, Bauen und Betreiben von Gebäuden (2. Aufl.). Berlin.

BMWi. (2015). *Bundesbericht Energieforschung 2015 -*

Forschungsförderung für die Energiewende (S. 1–44). Berlin:

Bundesministerium für Wirtschaft und Energie (BMWi).

Bramann, H., & May, I. (2015). *Stufenplan Digitales Planen und Bauen -*

Einführung moderner, IT-gestützter Prozesse und Technologien bei Planung, Bau und Betrieb von Bauwerken. Berlin:

Bundesministerium für Verkehr und digitale Infrastruktur (BMVI).

Abgerufen von

https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/stufenplan-digitales-bauen.pdf?__blob=publicationFile

Bringel, H., Caetano, A., & Tribolet, J. (2004). BUSINESS PROCESS

MODELING TOWARDS DATA QUALITY ASSURANCE - An

Organizational Engineering Approach: In I. Seruca, J. Filipe, S.

Hammoudi, & J. Cordeiro (Hrsg.), *Proceedings of the Sixth*

International Conference on Enterprise Information Systems -

(*Volume 3*) (S. 565–568). Porto: SciTePress - Science and Technology Publications.

<https://doi.org/10.5220/0002649305650568>

Brüggemann, T., & Both, P. von. (2014). Semantisches Informationsmodell als kollektive Datenbasis für eine nachhaltige Stadt- und Energieleitplanung. In M. K. Koch & R. McKenna (Hrsg.), *Wettbewerb „Energieeffiziente Stadt“*. Bd. 3: *Methoden und Modelle* (Bd. 3, S. 41–50). Münster Berlin: LIT.

Brüggemann, T., Ebertshäuser, S., Pinhal, G., & Both, P. von. (2015). *Entwicklung eines integrierten semantischen Informationsmodells als Planungshilfsmittel* (Schlussbericht No. Berichtszeitraum: 01.01.2013-31.03.2015) (S. 218). Karlsruhe: Karlsruher Institut für Technologie (KIT). Abgerufen von <https://doi.org/10.2314/GBV:863440991>

Brüggemann, T., Ebertshäuser, S., & von Both, P. (2020). 3D-Stadtmodellierung: CityGML. In A. Borrmann, M. König, C. Koch, & J. Beetz (Hrsg.), *Building Information Modeling: technologische*

Grundlagen und industrielle Praxis (2.). Wiesbaden: Springer Vieweg.

Claus, V., & Schwill, A. (Hrsg.). (1997). *Schülerduden, Informatik: [ein Sachlexikon für die Schule]* (3., neu bearb. Aufl.). Mannheim: Dudenverl.

Dörner, D. (2003). *Die Logik des Mißlingens. Strategisches Denken in komplexen Situationen.* (13, Erweiterte Neuausgabe Aufl.). Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag.

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, 18(8), 1011–1033. <https://doi.org/10.1016/j.autcon.2009.07.002>

Ebertshäuser, S. (2017). IFC Model Checking. In M. Wetter & C. van Treeck (Hrsg.), *IEA EBC Annex 60: New Generation Computing Tools for Building and Community Energy Systems – Final Report* (S. 182–189). Abgerufen von <http://www.iea-annex60.org/downloads/iea-ebc-annex60-final-report.pdf>

Ebertshäuser, S., Both, P. V., Wirtz, M., Brüggemann, T., Ochse, S., Lauster, M., ... Wallner, S. (2018). Unterstützung kommunalplanerischer Prozesse mit CityGLM-basierter Anbindung Modelica-getriebener

Quartierssimulationen. *Karlsruhe*.

<https://doi.org/10.5445/ir/1000085752>

Ebertshäuser, S., & Both, P. von. (2016). Entwicklung eines integrierten semantischen Informationsmodells als Planungshilfsmittels (ISIS).

Forschungsjahrbuch Energie 2015 – Forschungsberichte im Überblick. Abgerufen von

<https://www.adelphi.de/en/publication/forschungsjahrbuch-energie-2015>

Ebertshäuser, S., & Both, P. von. (2016). Model-based Planning for a Sustainable Urban Development. In *Real Corp 2016 -Smart me Up! Proceedings* (Bd. Real Corp Conference Proceedings). Wien.

Abgerufen von http://conference.corp.at/archive/CORP2016_30.pdf

Ebertshäuser, S., & Both, P. von. (2016). Prüfung und Konvertierung von HVAC ifcXML zu SimModel. In J. Grunewald (Hrsg.), *Proceedings CESBP Central European Symposium on Building Physics /*

BauSIM 2016 (S. 621–628). Dresden: Fraunhofer IRB Verlag.

Abgerufen von

https://www.baufachinformation.de/CESBP_BauSIM_Dresden_2016/CESBP_BausSIM/Proceedings.pdf

Ebertshäuser, S., Graf, K., Both, P. V., Rexroth, K., Bari, R. D., & Horn, R. (2019). Sustainable building information modeling in the context of model-based integral planning. *IOP Conference Series: Earth and Environmental Science*, 323, 12113. <https://doi.org/10.1088/1755-1315/323/1/012113>

Ebertshäuser, S., & von Both, P. (2013). ifcModelCheck. In *ECAADe 2013: Computation and performance : Proceedings of the 31st international conference on education and research in computer aided architectural design in Europe : 18-20 September 2013 Delft* (S. 525–534). Delft: Netherlands faculty of architecture, Delft University of Technology. Abgerufen von http://cumincad.architexturez.net/system/files/pdf/ecaade2013_213.content.pdf

Ebertshäuser, S., & von Both, P. (2019). Standardized Representation Of Typological Data As Common Input For Urban Performance

- Simulation. In V. Corrado & A. Gasparella (Hrsg.), *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*. Rome.
- English, L. P. (1999). *Improving data warehouse and business information quality: methods for reducing costs and increasing profits*. New York: Wiley.
- Fallon, K. K., & Palmer, M. E. (2007). *General Buildings Information Handover Guide* (Text No. 7417) (S. 99). Gaithersburg: NIST.
- Abgerufen von
http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=861038
- Flake, S., & Müller, W. (2001). Spezifikation von Echtzeiteigenschaften mit OCL für eine MFERT-basierte Fallstudie. Gehalten auf der 1st Workshop Ablaufmodellierung in ingenieurwissenschaftlichen Anwendungen, Berlin. Abgerufen von
https://www.researchgate.net/publication/268290903_Spezifikation_von_Echtzeiteigenschaften_mit_OCL_fur_eine_MFERT-basierte_Fallstudie
- Fowler, M. (2003). *Patterns of enterprise application architecture*. Boston: Addison-Wesley.
- Gausemeier, J., Frank, T., & Humpert, A. (1995). Systemkonzept zum Aufbau spezifischer Konstruktionssysteme. In U. Dietrich, B.

Kehrer, & G. Vatterrott (Hrsg.), *CA-Integration in Theorie und Praxis* (S. 11–32). Berlin, Heidelberg: Springer Berlin Heidelberg.
Abgerufen von http://www.springerlink.com/index/10.1007/978-3-642-79836-8_2

Geiger, A., & Hagenmeyer, V. (2018). Prüfung von openBIM Gebäudemodellen für die thermische Gebäudesimulation. In *BauSIM2018 - 7. Deutsch-Österreichische IBPSA-Konferenz : Tagungsband*. Hrsg.: P. von Both (S. 578–585). KIT, Karlsruhe.

Ghobad, L. (2018). Daylighting and Energy Simulation Workflow in performance-based Building Simulation Tools. In *Proceedings* (S. 382–389). Chicago: ASHRAE and IBPSA-USA. Abgerufen von <https://www.ashrae.org/File%20Library/Conferences/Specialty%20>

Conferences/2018%20Building%20Performance%20Analysis%20C
onference%20and%20SimBuild/Papers/C053.pdf

Händle, F., & Jensen, S. (1974). *Systemtheorie und Systemtechnik; sechzehn Aufsätze*. München: Nymphenburger Verlagshandlung.

Hartmann, U., & Both, P. von. (2015). A model-driven approach to the integration of product models into cross-domain analyses. *Journal of Information Technology in Construction (ITcon)*, 20(17), 253–274.

Hartung, F. (2019, Mai 22). Die KfW-Effizienzhäuser 40 und 40 plus [Bauberatung]. Abgerufen von <https://www.hausbauberater.de/energieeffizient-bauen/kfw-effizienzhaus-40>

Hausladen, G., Wagner, T., Schmid, T., Bonnet, C., Hamacher, T., Tzscheutschler, P., & Burhenne, R. (2011). *Leitfaden Energienutzungsplan*. (Bayerisches Staatsministerium für Umwelt und Gesundheit, Bayerisches Staatsministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie, & Oberste Baubehörde im Bayerischen Staatsministerium des Innern, Hrsg.). München. Abgerufen von www.energieatlas.bayern.de

Helfert, M. (2002). *Planung und Messung der Datenqualität in Data-Warehouse-Systemen*. Bamberg: Difo-Druck GmbH. Abgerufen von

<http://docplayer.org/4536595-Planung-und-messung-der-datenqualitaet-in-data-warehouse-systemen.html>

Herrmann, E., & Westphal, T. (Hrsg.). (2017). BIM in Zahlen - Status quo. *detail Research, Band 2*(Management-Band 2), 6–13.

Hildebrand, K., Gebauer, M., & Hinrichs, H. (Hrsg.). (2015). *Daten- und Informationsqualität: Auf dem Weg zur Information Excellence* (3., Aufl.). Wiesbaden: Springer Vieweg.

ISO (2005) Geographic information - Spatial schema (EN ISO 19107:2003), International Organization for Standardization, European Committee for Standardization

ISO (2006) Geographic information - rules for application schema (EN ISO 19109:2005), International Organization for Standardization, European Committee for Standardization

ISO (2007) Geographic information - Spatial referencing by coordinates (EN ISO 19111:2007), International Organization for Standardization, European Committee for Standardization

ISO (2008) Geographic information – Geography Markup Language (ISO 19136:2007 GML), International Organization for Standardization,

European Committee for Standardization.

<https://www.iso.org/standard/32554.html>.

ISO (2016) Building information models – Information delivery manual – P

art 1: Methodology and format (ISO 29481-1:2016 IDM),

International Organization for Standardization, European Committee

for Standardization. <https://www.iso.org/standard/60553.html>.

ISO (2018) Industry Foundation Classes (IFC) for data sharing in the

construction and facility management industries — Part 1: Data

schema (ISO ISO 16739-1:2018 IFC), International Organization for

Standardization, European Committee for Standardization.

<https://www.iso.org/standard/70303.html>.

Janke, R. (1978). *Architekturmodelle*. Stuttgart: Hatje.

KfW. (2020, Januar 1). Anlage zum Merkblatt - Energieeffizient Bauen:

Technische Mindestanforderungen. Kreditanstalt für Wiederaufbau.

Abgerufen von [https://www.kfw.de/PDF/Download-](https://www.kfw.de/PDF/Download-Center/F%C3%B6rderprogramme-(Inlandsf%C3%B6rderung)/PDF-Dokumente/6000003465_M_153_EEB_TMA_2018_04.pdf)

[Center/F%C3%B6rderprogramme-\(Inlandsf%C3%B6rderung\)/PDF-](https://www.kfw.de/PDF/Download-Center/F%C3%B6rderprogramme-(Inlandsf%C3%B6rderung)/PDF-Dokumente/6000003465_M_153_EEB_TMA_2018_04.pdf)

[Dokumente/6000003465_M_153_EEB_TMA_2018_04.pdf](https://www.kfw.de/PDF/Download-Center/F%C3%B6rderprogramme-(Inlandsf%C3%B6rderung)/PDF-Dokumente/6000003465_M_153_EEB_TMA_2018_04.pdf)

Koggelmann, Bayat, Mayerhöfer, Reif, Hieke, Botezatu, ... Lueb. (2012).

BFR GBestand. (Bundesministerium für Verkehr, Bau und

Stadtentwicklung, Hrsg.). Bundesministerium für Verkehr, Bau und

Stadtentwicklung - Referat Z 25, Druckvorstufe/Hausdruckerei.

Abgerufen von <https://www.fib->

[bund.de/Inhalt/Richtlinien/BFRGBestand/bfr_gbestand.pdf](https://www.fib-bund.de/Inhalt/Richtlinien/BFRGBestand/bfr_gbestand.pdf)

Kovacic, I., Achammer, C., Müller, C., Seibel, H., Wiegand, D., Srcekovic,

M., & Glöggler, J. (2012). *Integrated Planning GUIDELINE for*

public policy, investors and planners. TU Wien. Abgerufen von

<http://rgdoi.net/10.13140/2.1.4887.6329>

Kuhlman, D. (2020). generateDS -- Generate Data Structures from XML

Schema. [Webseite] davekuhlman.org. Abgerufen von

<https://www.davekuhlman.org/generateDS.html>

Lützkendorf, T., König, H., Kohler, N., & Kreißig, J. (2009).

Lebenszyklusanalyse in der Gebäudeplanung. Walter de Gruyter.

May, I. (2014). *BIM-Strategie Deutschland (Skizze) Digitalisierung der*

Wertschöpfungskette Bau. Folienpräsentation gehalten auf der

Arbeitsgruppe Moderne IT-gestützte Planungsmethoden (BIM) In

der Reformkommission Großprojekte, Berlin. Abgerufen von

<https://www.buildingsmart.de/kos/WNetz?art=File.download&id=1>

918&name=May_Digitalisierung_Wertsch%C3%B6pfungskette_Ba
u_004.pdf

Mazairac, W., & Beetz, J. (2013). BIMQL – An open query language for building information models. *Advanced Engineering Informatics*, 27(4), 444–456. <https://doi.org/10.1016/j.aei.2013.06.001>

Motzko, C., Löhr, M., Kochendörfer, B., & Pabst, L. (2012). *Entwicklung der Planungsprozesse 1992 - 2012 - gutachten_entwicklung_planungsprozesse_1992-2012_tu-darmstadt_tu-berlin* (Gutachten) (S. 152). Berlin: AHO – Ausschuss der Verbände und Kammern der Ingenieure und Architekten für die Honorarordnung e.V. Abgerufen von http://www.aho.de/pdf/gutachten_entwicklung_planungsprozesse_1992-2012_tu-darmstadt_tu-berlin.pdf

Pazold, M., Antretter, F., & Radon, J. (2014). Anbindung von detaillierter Anlagentechnik an hygrothermische Gebäudesimulation. In *Human-centred building(s) - Proceedings of BauSIM 2014: 5th German-Austrian Conference of the International Building Performance Simulation Association* (S. 263–270). Aachen. Abgerufen von

http://www.ibpsa.org/proceedings/bausimPapers/2014/p1148_final.pdf

Plotnikov, B., Schubert, G., & Petzold, F. (2016). Tangible Grasshopper: A method to combine physical models with generative, parametric tools.

Remmen, P., Cao, J., Ebertshäuser, S., Frisch, J., Lauster, M., Maile, T., ... Treeck, C. van. (2015). An Open Toolchain for Generating Modelica Code from Building Information Models. In *14th International Conference of the International Building Performance Simulation Association* (S. 383–391). Hyderabad.
<https://doi.org/10.3384/ecp15118383>

Rexroth, K., & Both, P. von. (2014). Planungsprozessmodelle: Synthese und Übertragbarkeit für den Anwendungskontext energieeffiziente Stadt. In M. K. Koch & V. McKenna (Hrsg.), *Methoden und Modelle* (S. 51–60). Münster Berlin: LIT.

Rexroth, K., Both, P. von, Gantner, J., & Schmid, C. (2018). Ein Ansatz für die typologiebasierte Vervollständigung von Gebäudeinformationen zur Unterstützung der frühen Planungsphasen am Beispiel LCA. In

- Karlsruhe* (S. 76–84). Karlsruhe: kitopen.
<https://doi.org/10.5445/ir/1000085748>
- Rexroth, K., & von Both, P. (2016). Integrale Planung für die energieeffiziente Stadt. In H.-J. Wagner & P. von Both (Hrsg.), *Gute Beispiele der Umsetzungsphase* (S. 137–148). Berlin Münster: LIT.
- Rittel, H. (1972). On the Planning Crisis: Systems Analysis of the ‘First and Second Generations’. *Bedriftsokonomien*, Nr. 8, 390–396.
- Rumpe, B. (2011). *Modellierung mit UML: Sprache, Konzepte und Methodik* (2. Aufl.). Berlin: Springer.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Wien: Springer.
- Suter, P., Gfeller, H. R., Kohler, N., & VanGilst, J. (Hrsg.). (1986). *Haustechnik in der Integralen Planung*. Bern: EDMZ.
- Treeck, C. van, Antretter, F., Heusler, I., Maas, A., Nicolai, A., Plessner, S., ... Wagner, A. (2014). EnOB-geförderte internationale Forschungsaktivitäten und Software-Tools. In *Energieinnovationen in Neubau und Sanierung - Neues aus der Forschung für mehr Energieeffizienz, Raumkomfort, Wirtschaftlichkeit und Nachhaltigkeit* (S. 17–23). Essen: Projektträger Jülich (PtJ).
- Abgerufen von

http://www.enob.info/fileadmin/media/Publikationen/EnOB/Tagungsdokumentationen/Tagungsbeitraege_EnOB-Symposium_2014x.pdf

Tuschinski, M. (2015). EnEV 2014 verschärft den Neubau-Standard ab 2016.

Betroffene Bauvorhaben - Ausnahmen und geänderte, energetische Anforderungen. *Der Bausachverständige, Jg.11, Nr.5*(Baurecht), 48–53.

Vajk, T., & Mezei, G. (2010). Incremental OCL to C# code generation (S.

277–280). IEEE. <https://doi.org/10.1109/ICCCYB.2010.5491277>

Van der Vlist, E. (2002). *XML schema: [the W3C's object-oriented descriptions for XML]* (1. ed). Beijing: O'Reilly.

von Both, P. (2006). *Ein systemisches Projektmodell für eine kooperative Planung komplexer Unikate*. Karlsruhe: Univ.-Verl. Karlsruhe.

von Both, P. (2011). Aktuelle Ansätze zur Unterstützung transdisziplinärer Zusammenarbeit im Bauwesen. In G. Banse (Hrsg.), *Wissenschaft*

im Kontext: Inter- und Transdisziplinarität in Theorie und Praxis (S. 227–242). Berlin: trafo-Wiss.-Verl.

von Both, P., & Ebertshäuser, S. (2018). BIM-basierte Hilfsmittel für eine integrale Planung. In C. M. Achammer & I. Kovacic (Hrsg.), *Construction Goes Digital*. Wien: Klein Publishing GmbH.

von Both, P., Koch, V., & Kindsvater, A. (2013). *BIM - Potentiale, Hemmnisse und Handlungsplan: Analyse der Potentiale und Hemmnisse bei der Umsetzung der integrierten Planungsmethodik Building Information Modeling - BIM - in der deutschen Baubranche und Ableitung eines Handlungsplanes zur Verbesserung der Wettbewerbssituation*. Stuttgart: Fraunhofer-IRB-Verl.

von Both, P., Treeck, C. van, Ebertshäuser, S., Arfaoui, S., Stratbücker, S., Herkel, S., ... Weise, M. (2017). Planung, Auslegung und Betriebsoptimierung von energieeffizienten Neu- und Bestandsbauten durch Modellierung und Simulation auf Basis von Bauwerkinformationsmodellen. In K. Ehrhardt (Hrsg.), *1. Kongress ENERGIEWENDEBAUEN 2017 - Forschung zu effizienten Technologien und intelligenten Konzepten für die Strom- und Wärmewende* (S. 268–279). Jülich: Forschungszentrum Jülich.

Abgerufen von

https://projektinfos.energiewendebauen.de/fileadmin/user_upload/Publicationen/Veranstaltungsdokumentationen/Tagungsband_1._Kongress_EWB_2017.pdf

Vossen, G. (2008). *Datenmodelle, Datenbanksprachen und*

Datenbankmanagementsysteme, 5. Auflage. Abgerufen von

https://www.researchgate.net/publication/220688745_Datenmodelle_Datenbanksprachen_und_Datenbankmanagementsysteme_5_Auflage

Wang, R. Y., & Strong, D. M. (1996). Beyond Accuracy: What Data Quality

Means to Data Consumers. *Journal of Management Information Systems, 12*(4), 5–33.

<https://doi.org/10.1080/07421222.1996.11518099>

West, M. (1999). *Developing High Quality Data Models*. EPISTLE.

Abgerufen von

<https://sites.google.com/site/drmatthewwest/publications/princ03.pdf>

Wetter, M., Treeck, C., Helsen, L., Maccarini, A., Saelens, D., Robinson, D.,

& Schweiger, G. (2019). IBPSA Project 1: BIM/GIS and Modelica framework for building and community energy system design and

operation – ongoing developments, lessons learned and challenges.

IOP Conference Series: Earth and Environmental Science, 323,

12114. <https://doi.org/10.1088/1755-1315/323/1/012114>

Wetter, M., & Treeck, C. van. (2017). *IEA EBC Annex 60: New Generation*

Computing Tools for Building and Community Energy Systems.

Abgerufen von <http://www.iea-annex60.org/pubs.html>

Wildenauer, A. A. (2016). Spielregeln für ein erfolgreiches integrales Projekt

mittels BIM. *Integrale Planung, Nachhaltiges Bauen –*

Energieeffizientes Handeln(Edition 2016), 46–53.

Zhang, C., Beetz, J., & Weise, M. (2014). Model view checking: automated

validation for IFC building models. In A. Mahdavi, B. Martens, &

R. Scherer (Hrsg.), *eWork and eBusiness in Architecture,*

Engineering and Construction (S. 123–128). CRC Press. Abgerufen

von <https://www.taylorfrancis.com/books/9781315736952>