

Master of Science FHNW in Virtual Design and Construction

# Einsatz eines relationalen Datenbanksystems in der digitalen Bauwerksmodellierung: Konzept für kleine und mittlere Unternehmen



Tufan, Öztürk

**Thesis-Begleiter:**

Prof. Lukas Schildknecht

lukas.schildknecht@fhnw.ch

**Thesis-Expert:**

Fredy Spring

**Praxispartner\*in:**

Virtuos 3D Modeling AG

## Eigenständigkeitserklärung

"Ich erkläre hiermit,

dass ich die vorliegende Master-Thesis mit dem Titel Einsatz eines relationalen Datenbanksystems in der digitalen Bauwerksmodellierung: Konzept für kleine und mittlere Unternehmen selbst und selbstständig verfasst habe,

dass ich sämtliche nicht von mir selbst stammenden Textstellen bzw. Bestandteile eines Werkes (Bilder, Grafiken, Codes, etc.) gemäss gängigen wissenschaftlichen Zitierregeln korrekt zitiert und die verwendeten Quellen gut sichtbar erwähnt habe;

dass ich in einem Verzeichnis alle verwendeten Hilfsmittel (KI-Assistenzsysteme wie Chatbots [z.B. ChatGPT], Übersetzungs- [z.B. DeepL] Paraphrasier- [z.B. Quillbot]) oder Programmierapplikationen [z.B. Github Copilot] deklariert und ihre Art der Verwendung offenlege und bei den entsprechenden Textstellen angegeben habe,

dass ich sämtliche immateriellen Rechte an von mir allfällig verwendeten Materialien wie Bilder oder Grafiken erworben habe oder dass diese Materialien von mir selbst erstellt wurden;

dass das Thema, die Arbeit oder Teile davon nicht bei einem Leistungsnachweis eines anderen Moduls verwendet wurden, sofern dies nicht ausdrücklich mit der Dozentin oder dem Dozenten im Voraus vereinbart wurde und in der Arbeit ausgewiesen wird;

dass ich mir bewusst bin, dass meine Arbeit auf Plagiate und auf Drittautorschaft menschlichen oder technischen Ursprungs (künstliche Intelligenz) überprüft werden kann;

dass ich mir bewusst bin, dass die Hochschule für Architektur, Bau und Geomatik einen Verstoß gegen diese Eigenständigkeitserklärung bzw. die ihr zugrundeliegenden Studierendenpflichten der Studien- und Prüfungsordnung der Hochschule für Architektur, Bau und Geomatik verfolgt und dass daraus disziplinarische (Verweis oder Ausschluss aus dem Studiengang) Folgen resultieren können.“

Vorname Nachname: Tufan Öztürk

Ort, Datum: Winterthur, 24.05.2024

Unterschrift:

## Vorwort

Ich möchte meine Dankbarkeit für die Unterstützung während der Erstellung dieser Master-Thesis zum Ausdruck bringen. Mein besonderer Dank gilt meinem Thesis Begleiter Prof. Lukas Schildknecht und meinem Thesis Experten Fredy Spring für ihre tatkräftige Unterstützung. Ein Dank an die Kollegen/Kolleginnen des Praxispartners Virtuos 3D Modeling AG für die treue Begleitung. Ein weiterer Dank geht an alle Fachgesprächspartner:innen, welche einen Einblick in Ihr Unternehmen gegeben haben. Auch meinem privaten Umfeld danke ich für die ständige Ermutigung. Diese Arbeit ist das Resultat der Forschung, Analyse und Reflexion, die von Interesse für die Themenbereiche der Datenbanken, der Softwareentwicklung und digitalen Bauwerksmodellierung angetrieben wurde. Ich hoffe, dass diese Arbeit nicht nur neue Erkenntnisse liefert, sondern auch dazu beiträgt, das Verständnis auf diesem Gebiet zu erweitern und weitere Forschungen anzuregen.

## Abstract

Die Modellierung digitaler Bauwerksmodelle (DBM) erfordert derzeit den Einsatz proprietärer Auto-  
rensoftware. Dies stellt eine Einstiegshürde dar und erschwert die Beteiligung fachkundiger Personen  
ohne Modellierungskennnisse an der Bearbeitung des DBMs. In dieser Master-Thesis wurde unter-  
sucht, wie kleine und mittlere Unternehmen (KMU) eine plattformunabhängige Datenbanktechnologie  
als BIM-Werkzeug nutzen können, welche die relevanten Konzepte des offenen Standards Industry  
Foundation Classes (IFC) in der Datenstruktur berücksichtigt. Ein wesentlicher Bestandteil dieser Un-  
tersuchung war die Sicherstellung der Datenkonsistenz und die Synchronisation zwischen der Daten-  
bank und der Autorensoftware.

Basierend auf den Erkenntnissen und Anforderungen der KMUs aus den Expertengesprächen erfolgte  
die methodische Auswahl der geeigneten Datenbanktechnologie und die konzeptionelle Beschreibung  
der Systemarchitektur. Schliesslich wurde die Machbarkeit durch einen Proof of Concept nachgewie-  
sen und die Nutzung reflektiert. Die Systemarchitektur bestand aus einer SQLite-Datenbank, einer  
IFC-Datei und dem Autodesk Revit. Zwei Python-Skripte gewährleisteten die Kommunikation sowie  
die Sicherstellung der Datenkonsistenz und Synchronisation zwischen den Systemen. Identifizierte  
Faktoren, die die Datenkonsistenz beeinflussen, führten zur Entwicklung technischer Mechanismen  
wie Integritätsbedingungen im Datenbankdesign, Datenhoheitsregelung sowie Steuerung des Daten-  
flusses. Die Beschreibung digitaler Bauteile mit Eigenschaften und Klassifizierung von Modellen sowie  
Mengenanzüge sind in der Praxis häufige Anwendungsfälle bei der Datenverwaltung von DBM, wel-  
che durch eine Mapping-Methode in eine geeignete Datenbankstruktur übersetzt wurden.

Die Analyse dieser Master-Thesis zeigt, dass weitere Studien erforderlich sind, um den Modellierungs-  
prozess digitaler Bauwerksmodelle in der Praxis umfassender zu untersuchen. Ziel ist es die beobach-  
teten Trends zu bestätigen und anhand einer Benutzeroberfläche die Praxistauglichkeit des Proof of  
Concepts zu überprüfen.

Schlagwörter: *KMU, Digitale Bauwerksmodellierung, IFC, relationales Datenbanksystem, Bauteilinfo-  
mationen, Synchronisation, Datenkonsistenz*

# Inhaltsverzeichnis

Vorwort .....	III
Abstract.....	IV
Abbildungsverzeichnis .....	VIII
Tabellenverzeichnis .....	IX
Hilfsmittelverzeichnis.....	X
Glossar .....	XI
Abkürzungsverzeichnis .....	XIII
1 Einleitung.....	1
1.1 Fragestellungen .....	2
1.2 Eingrenzung der Arbeit .....	3
1.3 Aufbau der Arbeit.....	3
2 Methodik .....	4
2.1 Methodik der Grundlagenstudie .....	4
2.2 Methodik der Literaturrecherche .....	5
2.3 Methodik der Datenerhebung in der Praxis .....	6
2.3.1 Zusammenstellung des Samplings .....	6
2.3.2 Gesprächsführung .....	6
2.3.3 Auswertung.....	6
2.4 Methodik der Entscheidungsfindung von Datenbanktechnologien.....	7
2.5 Methodik der semantischen Beschreibung.....	7
2.5.1 Anforderungsanalyse .....	7
2.5.2 Anwendungsfallmodellierung .....	7
2.6 Methodik der konzeptuellen Beschreibung.....	8
2.6.1 Notation .....	8
2.6.2 Modellierung .....	8
2.7 Methodik der logischen Umsetzung .....	9
2.7.1 Erstellung des Proof of Concepts.....	9
2.7.2 Rückmeldungen aus der Praxis .....	9
3 Stand der Forschung .....	10
4 Praktischer Einblick in KMUs .....	11
4.1 Das Sampling .....	11
4.2 Modellierungsstufen von DBM .....	12
4.3 Rollentrennung im Modellierungsprozess .....	13
4.4 Prozess zur semantischen Informierung des DBMs .....	14
4.5 Semantische Beschreibung von DBM .....	15
4.6 Datenbanken im Unternehmen .....	16
4.7 Generalisiertes Anforderungsprofil der KMUs .....	17
5 Datenbanktechnologien im Vergleich .....	18

5.1	In der Praxis übliche Datenbanktechnologien .....	18
5.2	Bewertungsfaktoren .....	19
5.3	Resultat aus der Entscheidungsmatrix .....	20
5.4	Entscheid der Datenbanktechnologie.....	21
6	Fachliche Beschreibung des Kontextes .....	22
6.1	Rollen im Modellierungsprozess .....	23
6.1.1	Fachmodellierende Person .....	23
6.1.2	Fachprojektleitende Person .....	23
6.2	Gesamtprozess.....	24
6.3	Kontextabgrenzung.....	25
6.4	Anwendungsfälle .....	26
7	Konzeptuelle Beschreibung .....	28
7.1	Systemarchitektur .....	28
7.2	Systemprozesse .....	28
7.3	Datenkonsistenzsystem .....	31
7.3.1	Kontrolle über den Datenfluss.....	32
7.3.2	Regelung der Datenhoheit .....	33
7.3.3	Protokollierung der Operationen .....	34
7.4	Systembaustein IFC-SPF.....	35
7.4.1	Mapping der IFC-Konzepte .....	35
7.4.2	IFC-Konzepte im IFC-Datenmodell .....	36
7.5	Systembaustein Datenbank .....	38
7.5.1	Mapping Methode .....	38
7.5.2	Primärschlüssel .....	40
7.5.3	Relationen .....	40
7.5.4	Datenbankschemas .....	42
7.5.5	Aktivität .....	47
7.6	Systembaustein Middleware .....	48
7.6.1	Datenbank aufsetzen .....	48
7.6.2	Prozess der Datenverarbeitung .....	48
7.6.3	Aktivität .....	48
7.7	Systembaustein Autorensoftware.....	50
7.7.1	Informationsanforderung.....	50
7.7.2	Export und Import .....	50
7.7.3	Aktivitäten .....	51
7.8	Sequenzierung.....	53
8	Entwicklung des Proof of Concepts.....	54
8.1	Verwendete Technologien .....	54
8.2	Systemdesign .....	55

8.3	Aufbau der relevanten Skripts .....	56
8.4	Projektordner im Dateisystem .....	57
8.5	Einrichtung Autodesk Revit .....	58
8.5.1	Benutzerdefinierte Parameter .....	58
8.5.2	3D Geometrien der Bauteile.....	59
8.5.3	Definierung des IFC-Exports.....	60
8.6	Export von IFC-SPF.....	60
8.7	Initialisierung SQLite-Datenbank.....	61
8.7.1	Mapping von Datentypen .....	61
8.7.2	Erstellung des Datenbankmodells.....	62
8.8	Datenimport in Revit .....	63
8.9	Mögliche GUI-Implementierung.....	65
8.10	Limitierungen und Weiterentwicklung .....	66
9	Rückmeldungen aus der Praxis .....	67
10	Diskussion.....	68
11	Fazit und Ausblick .....	69
12	Literaturverzeichnis .....	71
13	Anhang.....	73
13.1	Anhang A: Literaturrecherche – Begriffstabellen.....	73
13.2	Anhang B: Datenerhebung – Interview-Leitfaden .....	73
13.3	Anhang C: Datenerhebung – Transkription der Fachgespräche .....	73
13.4	Anhang D: Datenerhebung – Anforderungen aus den Fachgesprächen.....	73
13.5	Anhang E: Entscheidungsmatrix RDBMS.....	73
13.6	Anhang F: Proof of Concept – Quellcode .....	73

## Abbildungsverzeichnis

Abbildung 1: Ausgangslage des Praxispartners bei der Fachmodellierungsaufgabe .....	1
Abbildung 2: SOLL-Zustand und Einsatz der Master-Thesis .....	1
Abbildung 3: Ein grober Umriss der Systemarchitektur .....	2
Abbildung 4: Aufbau der Arbeit .....	3
Abbildung 5: angewendete Methodik (in Orange) in der jeweiligen Abfolge von Tätigkeiten .....	4
Abbildung 6: Themenfelder und deren Schnittmenge .....	4
Abbildung 7: Abfolge der Literaturrecherche und angewendeter Methoden in Orange .....	5
Abbildung 8: Methodik zur Beschreibung der Anwendungsfälle .....	7
Abbildung 9: UML Diagrammarten, Quelle: (Krcmar, 2015, p. 37) .....	8
Abbildung 10: Horizontale und vertikale Prototypen nach Balzer, Quelle: (Broy & Kuhrmann, 2021) ...	9
Abbildung 11: Unternehmensgrösse der befragten Unternehmen .....	11
Abbildung 12: Branchensegment der befragten Unternehmen .....	11
Abbildung 13: Funktionen der befragten Personen .....	11
Abbildung 14: Ortschaften der befragten Unternehmen .....	11
Abbildung 15: Auswertungstabelle «Modellierungsstufen von DBM» (MAXQDA 2022) .....	12
Abbildung 16: Auswertungstabelle «Rollentrennung im Modellierungsprozess» (MAXQDA 2022) .....	13
Abbildung 17: Auswertungstabelle «Herausforderung für fachmodellierende Personen» (MAXQDA 2022) .....	13
Abbildung 18: Modellierungsprozess bei den KMUs .....	14
Abbildung 19: relevante IFC-Konzepte aus fachlicher Sicht (MAXQDA 2022) .....	15
Abbildung 20: Auswertungstabelle Datenbanknutzung im Unternehmen (MAXQDA 2022) .....	16
Abbildung 21: Auswertungstabelle genannter Datenbanken im Unternehmen (MAXQDA 2022) .....	16
Abbildung 22: Rangliste der meistgenutzten RDBMS, Quelle: (solid IT, 2023), Datenstand Oktober 2023 .....	18
Abbildung 23: Resultat Entscheidungsmatrix RDBMS (blau: serverbasiert, rot: filebasiert, grün: cloudbasiert) .....	20
Abbildung 24: Beschreibung des übergeordneten Kontextes .....	22
Abbildung 25: Fachmodellierende Person .....	23
Abbildung 26: Fachprojektleitende Person .....	23
Abbildung 27: Beschreibung des IST-Prozesses .....	24
Abbildung 28: Beschreibung des SOLL-Prozesses im Zusammenhang mit der gewünschten Systemarchitektur .....	24
Abbildung 29: fachliche Kontextabgrenzung in Anlehnung an (Starke & Hruschka, 2011) .....	25
Abbildung 30: Resultierende Systemarchitektur .....	28
Abbildung 31: Anwendungsfalldiagramm der Systemprozesse .....	30
Abbildung 32: Adressierung möglicher Inkonsistenzen .....	31
Abbildung 33: Konzept zur Regelung des Datenflusses .....	32
Abbildung 34: Konzepte zur Regelung der Datenhoheiten .....	33
Abbildung 35: Protokollierung der Operationen in einer Logdatei .....	34
Abbildung 36: Mapping IFC-Konzept mit den fachlichen Beschreibungen der DBM aus den Fachgesprächen .....	35
Abbildung 37: Relevante Klassen aus den IFC-Konzepten als Pseudodarstellung .....	36
Abbildung 38: IFC-Datenschema mit den relevanten IFC-Klassen am Beispiel von «IfcValve» und «IfcValveType» als Objekte .....	37
Abbildung 39: Tabellenstrukturen von verschiedenen Mapping-Konzepten, Quelle: (Zhu & Shide, 2021) .....	39
Abbildung 40: Relation zwischen IFC-Objekt und PropertySet .....	41
Abbildung 41: Datenbankschema «Typisierung» .....	42
Abbildung 42: Datenbankschema «Eigenschaften» und «Mengen» .....	43
Abbildung 43: Datenbankschema «Gruppierung» .....	44
Abbildung 44: Datenbankschema «Klassifizierung» .....	45
Abbildung 45: Datenbankschema «Materialisierung» .....	46

Abbildung 46: Aktivitätsdiagramm: Datenbank initialisieren.....	47
Abbildung 47: Temporäre Datei im Prozess der Datenverarbeitung in der Datenbank.....	48
Abbildung 48: Aktivitätsdiagramm «Daten erfassen in DB».....	49
Abbildung 49: Temporäre Datei für die Autorensoftware.....	50
Abbildung 50: Aktivitätsdiagramm «Import Daten in Autorensoftware».....	51
Abbildung 51: Aktivitätsdiagramm «IFC-SPF Export».....	52
Abbildung 52: Sequenzdiagramm des gesamten Systems.....	53
Abbildung 53: Paketdiagramm des Middlewares.....	55
Abbildung 54: Paketdiagramm der Autorensoftware.....	55
Abbildung 55: technischer Aufbau des Skripts «Middleware».....	56
Abbildung 56: Revit Werkzeuge auf der GUI-Oberfläche von Revit 2022.....	56
Abbildung 57: technischer Aufbau des Skripts «pyRevit».....	57
Abbildung 58: Projektordner im Dateisystem des Betriebssystems.....	57
Abbildung 59: Ausschnitt aus dem IFC Shared Parameters-Revit.txt.....	59
Abbildung 60: 3D-Geometrien mit den dazugehörigen IFC-Parametern (links: Pumpe, rechts: Ventil und Ansicht aus Revit).....	59
Abbildung 61: Ausschnitt aus dem DefaultUserDefinedParameterSets.txt.....	60
Abbildung 62: Resultat von leeren Parametern aus Revit (Ansicht aus dem DB Browser).....	61
Abbildung 63: Mapping von Datentypen (in Anlehnung an (Wei et al., 2009) und (Lee et al., 2014))..	61
Abbildung 64: Physikalisches Datenbankmodell. Veranschaulichung der Relationen.....	62
Abbildung 65: Tabellen im DB Browser.....	63
Abbildung 66: Dialogfenster beim IFC-Export von pyRevit.....	64
Abbildung 67: Importvorgang in Revit (Dialogfenster für Print-Befehle aus dem Python-Skript über pyRevit).....	64
Abbildung 68: Ansicht der Pumpe mit den Eigenschaftensätzen, den Eigenschaften und den Eigenschaftswerten.....	65
Abbildung 69: GUI-Implementierung.....	65
Abbildung 70: Weitere Schritte bis zur Anwendung in der Praxis.....	66

## Tabellenverzeichnis

Tabelle 1: verwendete Literaturdatenbanken.....	5
Tabelle 2: Modellierungsstufen von DBM.....	12
Tabelle 3: Bewertungsfaktoren für die Entscheidungsmatrix von RDBMS.....	19
Tabelle 4: Kategorien der Datenbanktechnologien.....	20
Tabelle 5: Bedingungen zur Bestimmung des Primärschlüssels.....	40
Tabelle 6: Verwendete Technologien.....	54
Tabelle 7: Informationsanforderung als Parametertabelle.....	58
Tabelle 8: Mapping-Tabelle zur Informationsanforderung.....	59
Tabelle 9: Mapping der Attribute aus dem IFC zu Parameter in Revit.....	64

## Hilfsmittelverzeichnis

DeepL Übersetzer	Zum Übersetzen von Literatur und Quellen, die nicht in der Sprache der vorliegenden Master-Thesis verfasst sind.
MAXQDA 2022	Software für qualitative, quantitative und Mixed-Methods-Datenanalyse der aufgezeichneten Fachgesprächen und zur Identifizierung von Themen und Mustern durch Codierungen.
Töggli	Zur automatischen Transkription der Fachgespräche auf Schweizerdeutsch. Töggli nutzt eine künstliche Intelligenz, um gesprochenes Wort in Text umzuwandeln.
ChatGPT 3.5	Als Lektorat zur Überprüfung im Bericht von Rechtschreibung, Grammatik und Schreibstil sowie als Assistent für die Korrekturen im Quellcode.

## Glossar

### **Alphanummerische Informationen**

Daten, die aus einer Kombination von Buchstaben und Zahlen bestehen.

### **BIM-Werkzeug**

Eine Softwareanwendung, die für die Erstellung, Verwaltung und Analyse digitaler Gebäudemodelle im Rahmen BIM verwendet wird.

### **CPython**

Referenzimplementierung der Programmiersprache Python. Es ist in der Programmiersprache C geschrieben.

### **Datenbank**

Ein systematischer Ansatz zur Speicherung und Verwaltung von Daten. Zur Vereinfachung wird diese Bezeichnung für relationale Datenbanken verwendet.

### **Datenbankschema**

Die Definition der Tabellen und ihrer Attribute wird im relationalen Datenbankschema gespeichert. Dieses Schema umfasst auch die Festlegung der Identifikationsschlüssel sowie Regeln zur Sicherstellung der Datenintegrität.

### **Datenbankmodell**

Auf dem relationalen Modell basierend, was bedeutet, dass alle Daten und deren Beziehungen in Form von Tabellen dargestellt werden.

### **Datenmodell**

Beschreibung der logischen Struktur der Datenbank, einschliesslich der Art der Daten, ihrer Beziehungen zueinander und der Einschränkungen, die auf die Daten angewendet werden.

### **Digitales Bauwerksmodell**

Die digitale Repräsentation eines Bauwerks. Fachmodelle sind Domäne spezifische digitale Bauwerksmodelle. Zur Vereinfachung wird mit dieser Bezeichnung auch das Fachmodell genannt.

### **Entität**

Ein eindeutig identifizierbares Objekt in einer Datenbank oder in einem Datenmodell, das relevante Informationen speichert. Die Bezeichnung wird als synonym zur Bezeichnung «Klasse» verwendet.

### **EXPRESS**

Eine Modellierungssprache mit deren Hilfe objektorientierte Datenmodelle definiert werden können.

### **Geometrische Informationen**

Daten, die die Form, Lage, Grösse und Beziehung von geometrischen Objekten wie Punkten, Linien, Flächen und Volumenkörpern beschreiben.

### **IFC**

Ein offener Standard im Bauwesen zur digitalen Beschreibung von Gebäudemodellen.

### **IronPython**

Eine Implementierung der Programmiersprache Python, die auf der .NET-Framework basiert.

**KMU**

Marktwirtschaftliche Unternehmen, die weniger als 250 Beschäftigte zählen. Unterteilt werden die Unternehmen in mikro-, kleine und mittlere Unternehmen. Die Abstufung erfolgt über die Mitarbeiterzahl.

**Open Source**

Eine Software, deren Quelltext öffentlich ist und von Dritten, je nach Lizenzierung, eingesehen, geändert und genutzt werden kann.

**Regular Expression**

Eine Sequenz von Zeichen, die ein Suchmuster definieren.

## Abkürzungsverzeichnis

ACID	Atomicity, consistency, isolation und durability ( <i>Atomarität, Konsistenz, Isolation und Dauerhaftigkeit</i> )
API	Application Programming Interface ( <i>Anwendungsschnittstelle</i> )
BIM	Building Information Modeling ( <i>Bauwerksdatenmodellierung</i> )
CSV	Comma-separated values
DB	Datenbank
DBM	Digitaler Bauwerksmodell
ER	Entity-Relationship
ERM	Entity-Relationship-Modell
FK	Foreign Key ( <i>Fremdschlüssel</i> )
GIS	Geografische Informationssysteme
GUI	Graphical user interface ( <i>grafische Benutzeroberfläche</i> )
IA	Informationsanforderung
IFC	Industry Foundation Classes
IFC-SPF	IFC Step Physical File ( <i>IFC Austauschformat</i> )
IT	Informationstechnik
KMU	Kleine und Mittlere Unternehmen
MS	Microsoft
ODBC	Open Database Connectivity ( <i>offene Datenbank-Verbindungsfähigkeit</i> )
PK	Primary Key ( <i>Primärschlüssel</i> )
RAD	Rapid Application Development ( <i>schnelle Anwendungsentwicklung</i> )
RDB	Relationale Datenbank
RDBM	Relationales Datenbankmanagement
RDBMS	Relationales Datenbankmanagementsystem
Regex	Regular expression ( <i>Regulärer Ausdruck</i> )
SPF	STEP Physical File ( <i>STEP-Datei</i> )
SPF-ID	Identifizierungsnummer innerhalb des IFC-SPFs
STEP	Standard for the Exchange of Product model data
SQL	Structured Query Language ( <i>Strukturierte Abfrage-Sprache</i> )
UML	Unified Modelling Language ( <i>vereinheitlichte Modellierungssprache</i> )

# 1 Einleitung

Die Modellierung digitaler Bauwerksmodelle erfordert derzeit den Einsatz proprietärer Autorensoftware. Dies stellt eine Einstiegshürde dar und erschwert die Beteiligung fachkundiger Personen ohne Modellierungskenntnisse an der Bearbeitung des DBMs. Die Fachmodelle der einzelnen Projektbeteiligten bleiben in ihren eigenen proprietären Dateiformaten. Ein Ansatz zur gemeinsamen nativen Modellierung von Fachmodellen direkt im interoperablen Dateiformat IFC-SPF existiert aktuell lediglich in Form eines Whitepapers (Postle & Moul, 2022).

Die Arbeitsweise des Praxispartners «Virtuos 3D Modeling AG» im Modellierungsprozess unterscheidet zwischen zwei Aufgaben. Das Fachmodell wird zuerst in 3D modelliert und danach mit alphanumerischen Informationen in der Autorensoftware angereichert. Die Qualität und Funktionalität der Fachmodelle hängen, gemäss Praxispartner von der Expertise der jeweiligen fachmodellierenden Person ab (siehe Abbildung 1).

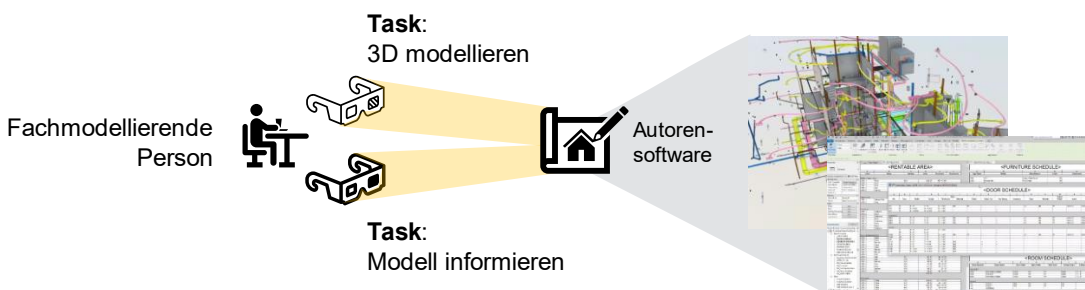


Abbildung 1: Ausgangslage des Praxispartners bei der Fachmodellierungsaufgabe

Der Praxispartner argumentiert, dass die Fachmodellierungsaufgabe mit einer höheren Konzentration verbunden ist, wenn beide Aufgaben iterativ oder im gleichen Zug von einer fachmodellierenden Person erfolgt. Dabei erhöht sich die Komplexität und das Fehlerpotential der Arbeit, womit ein ungewollter Leistungsdruck auf das Personal ausgeübt wird.

Deshalb wird in dieser Master-Thesis ein technischer Lösungsansatz konzeptuell erarbeitet, welcher erlaubt, die Trennung der beiden Aufgaben im Modellierungsprozess in einem KMU umzusetzen. Die Verwendung geeigneter Datenbanktechnologien mit synchronen Zusammenarbeitsmechanismen zur Gewährleistung der Datenkonsistenz zwischen Autorensoftware und Datenbank wird als mögliche Lösung betrachtet (siehe Abbildung 2). Ziel ist es, Fachleuten ohne Modellierungskenntnisse den Zugang zum DBM anhand eines BIM-Werkzeugs zu erleichtern und gleichzeitig die Lizenzkosten für die Autorensoftware zu senken. Zudem wird ermöglicht Erkenntnisse über alle Projekte hinweg aus den Informationen des DBMs zu gewinnen. Denn die Daten liegen strukturiert in einer Datenbank vor.

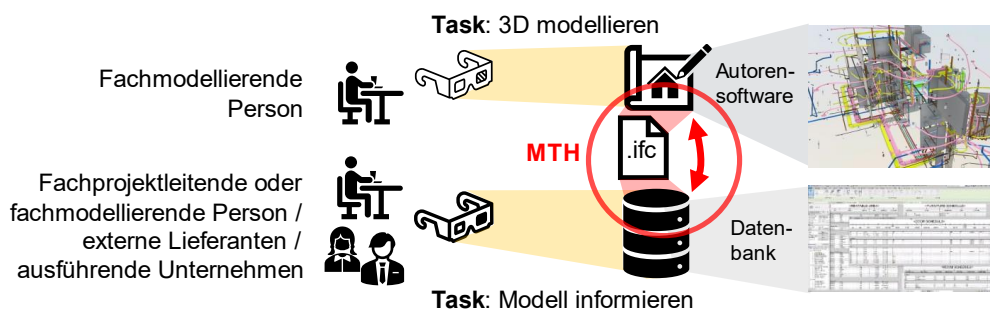


Abbildung 2: SOLL-Zustand und Einsatz der Master-Thesis

Diese Herausforderung betrifft wahrscheinlich nicht nur den Praxispartner als Mikrounternehmen, welches etwa 90% der Architektur- und Ingenieurbüros in der Schweiz ausmachen, sondern auch andere Planungsunternehmen (Bundesamt für Statistik, 2023). Die bisherigen Interviewstudien aus Deutschland zeigen, dass Unternehmen in der Branche bei der Digitalisierung mit technischen, finanziellen und personellen Herausforderungen konfrontiert sind. Insbesondere wird von einigen Beteiligten die zunehmende Verantwortung der Wissenschaft im Hinblick auf die Entwicklung einer Software, welche die Probleme der Datenqualität und des Datenaustauschs löst, betont (PricewaterhouseCoopers GmbH, 2019; Digital Construction Management, 2023).

Die Master-Thesis bietet anhand dieser Konzeptsdokumentation die erforderlichen Grundlagen und ein mögliches Vorgehen zur Entwicklung der in Abbildung 3 dargestellten Systemarchitektur.

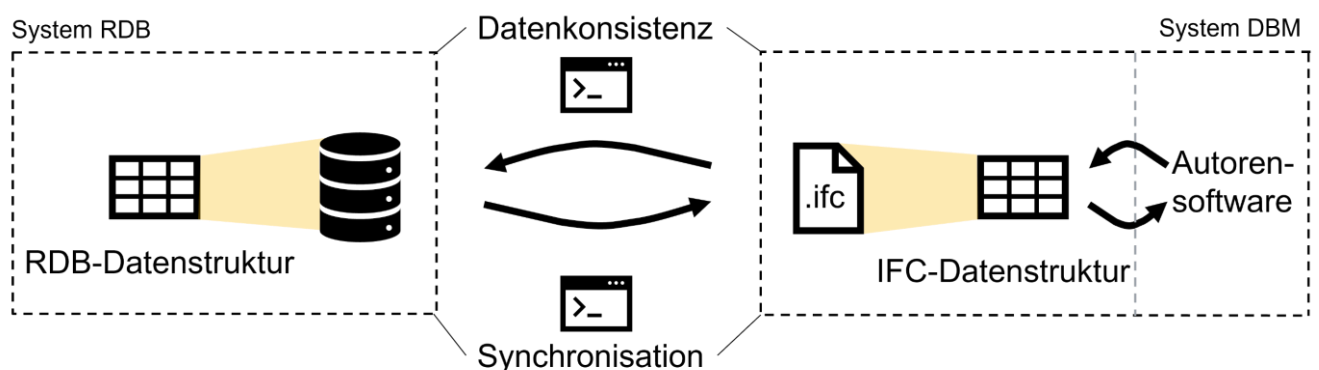


Abbildung 3: Ein grober Umriss der Systemarchitektur

## 1.1 Fragestellungen

Die zentrale Fragestellung lautet für die vorliegende Arbeit folgendermassen:

*Wie können KMUs geeignete Datenbanktechnologien nutzen eine Datenstruktur entwickeln, welche die zentralen IFC-Konzepte der DBM in Datenbanksystemen integriert und die Synchronisation sowie die Datenkonsistenz zwischen den beiden Systemen sicherstellt?*

Aus der zentralen Fragestellung leiten sich in chronologischer Abfolge folgende Teilfragestellungen ab, die sich in die Themen der Datenerhebung und der konzeptuellen Arbeit unterteilen:

- Wie werden heute bei den KMUs die DBM informiert und welche Datenbanktechnologien werden im Unternehmen eingesetzt?
- Welche Anforderungen werden von den KMUs an die jeweiligen Anwendungsfälle gestellt?
- Welche IFC-Konzepte haben heute in der praktischen Anwendung bei den KMUs eine Relevanz?
- Mit welcher Datenstruktur können die Daten aus dem IFC-Datenmodell in die RDB auf einfache Weise und unter Berücksichtigung der Datenkonsistenz implementiert werden?
- Welche Faktoren beeinflussen die Datenkonsistenz und wie kann eine Dateninkonsistenz zwischen einem DBM und dem RDB verhindert werden?
- Welche sind die kritischen Aspekte der Synchronisation der beiden Systeme und welche Herausforderungen ergeben sich daraus?

## 1.2 Eingrenzung der Arbeit

Das IFC-Datenmodell steht im Mittelpunkt dieser Arbeit, da es eine Schnittmenge zu sämtlichen Unternehmenskontexten abdeckt. Proprietäre Formate von Autorensoftware und deren direkte Verknüpfungen zu Datenbanken werden nicht behandelt. Aufgrund der Komplexität und Vielfalt des IFC-Datenmodells werden nur die relevanten IFC-Konzepte berücksichtigt, die aus den Fachgesprächen resultieren, um den Umfang der Arbeit einzugrenzen. Die Implementierung beschränkt sich auf die Erstellung eines Proof of Concepts, um ein funktionierendes Modell zur Überprüfung des Konzepts zu entwickeln. Es ist daher nicht zu vergleichen mit einem Prototyp. Erfassungsmasken für Nutzende und deren Authentifizierung bei der Nutzung der Datenbank werden nicht im Detail behandelt, es wird lediglich ein Ausblick gegeben. Die übergeordneten Geschäftsmodelle und -prozesse der KMUs, die mit dem Modellierungsprozess von Fachmodellen verbunden sind, bleiben ebenfalls ausserhalb dieser Arbeit. Für die Fachgespräche werden Interviewpartner:innen aus der Schweiz ausgewählt. Diese arbeiten alle im Hochbaubereich und haben den Sitz in der Schweiz.

## 1.3 Aufbau der Arbeit

Die Abbildung 4 stellt die Abfolge von Aktivitäten in der Master-Thesis dar und strukturiert diese in drei Phasen. Die Pfeile definieren die fortlaufende Richtung der Aktivitäten. Die Phase 1 umfasst den theoretischen Teil und die Phase 2 sowie Phase 3 den praktischen Teil der Arbeit. Die Abfolge der Aktivitäten bestimmt den inhaltlichen Aufbau der Arbeit. In Phase 1 erfolgt zunächst eine Grundlagenstudie und einer Recherche zum aktuellen Stand der Forschung. Anschliessend wird eine Datenerhebung in der Praxis durchgeführt sowie verschiedene Datenbanktechnologien miteinander verglichen. In Phase 2 werden die fachlichen und konzeptuellen Beschreibungen erarbeitet. Dabei wird die Arbeit anhand der logischen Umsetzung technisch überprüft. Abschliessend wird die gesamte Arbeit reflektiert.

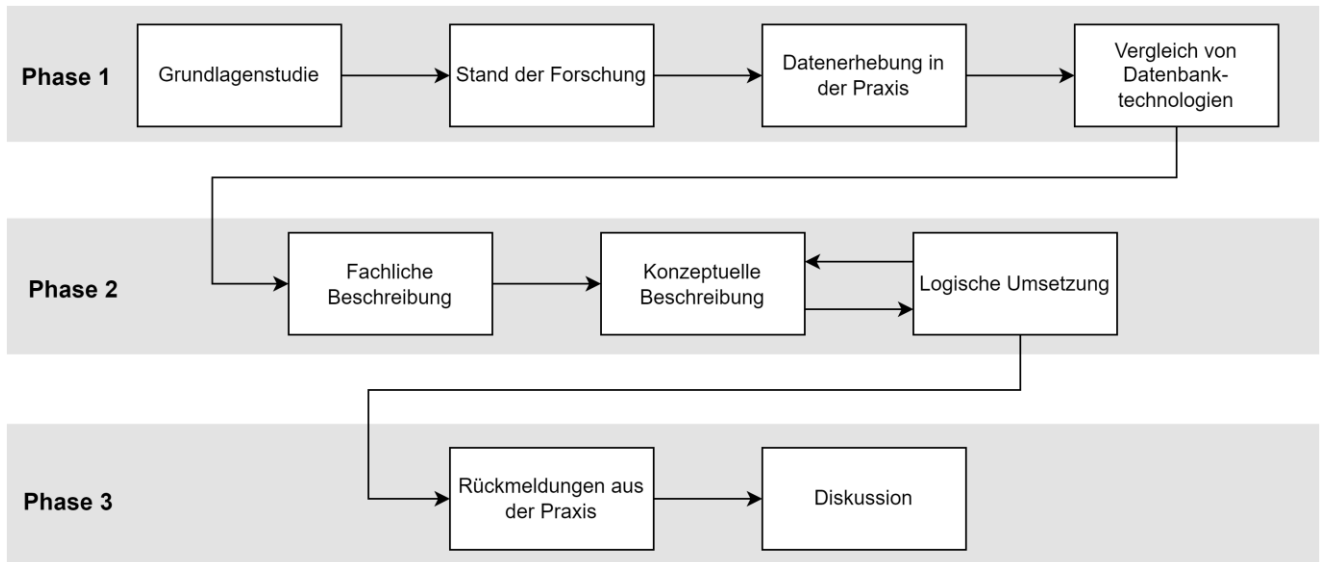


Abbildung 4: Aufbau der Arbeit

## 2 Methodik

Die Arbeit beinhaltet die Anwendung von unterschiedlichen Methoden. Diese Methoden erstrecken sich über die Abfolge von Aktivitäten im gesamten Arbeitsprozess, welche in der Abbildung 5 als orange Felder ersichtlich sind. Die Methoden sind in den weiteren Unterkapiteln im Detail beschrieben.

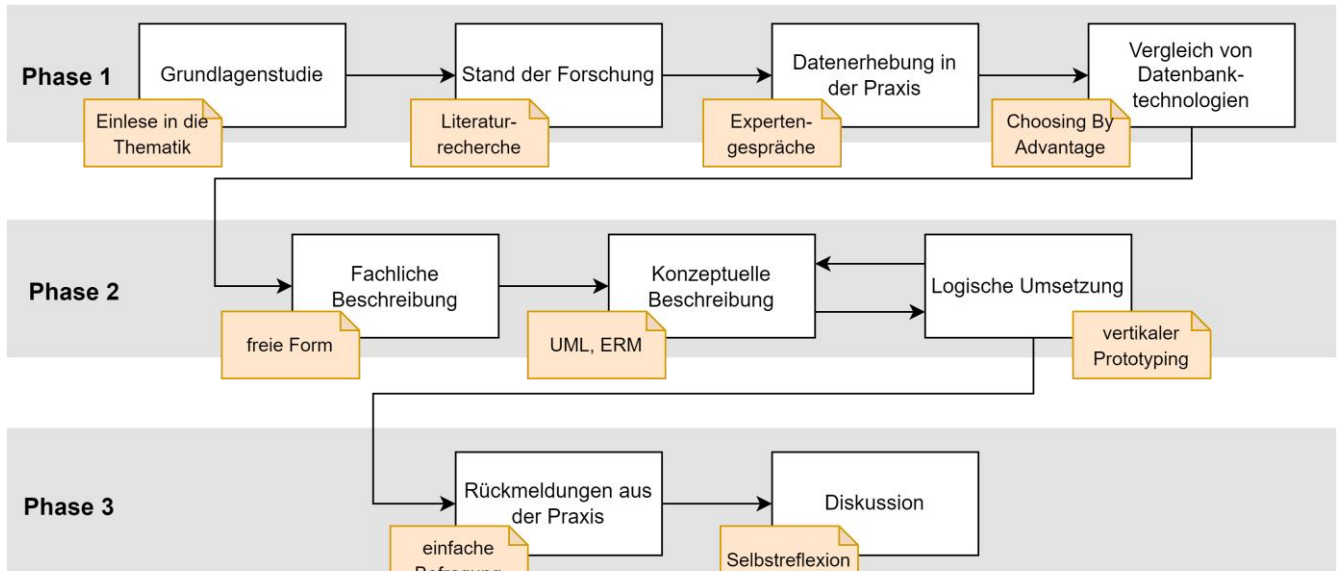


Abbildung 5: angewendete Methodik (in Orange) in der jeweiligen Abfolge von Tätigkeiten

### 2.1 Methodik der Grundlagenstudie

Die Arbeit grenzt an die Themenfelder der Datenbanken, der BIM und der Softwareentwicklung. Das Forschungsinteresse liegt in der Schnittmenge zwischen den drei Themenfeldern, was in der Abbildung 6 dargestellt ist.

Zu Beginn wird eine Fachliteraturrecherche in den aufgeführten Themenfeldern durchgeführt, um damit ein solides Fundament des Fachwissens zu erhalten. Für die Recherche von Fachliteraturen wird die Datenbank von Springer (<https://link.springer.com>) verwendet.

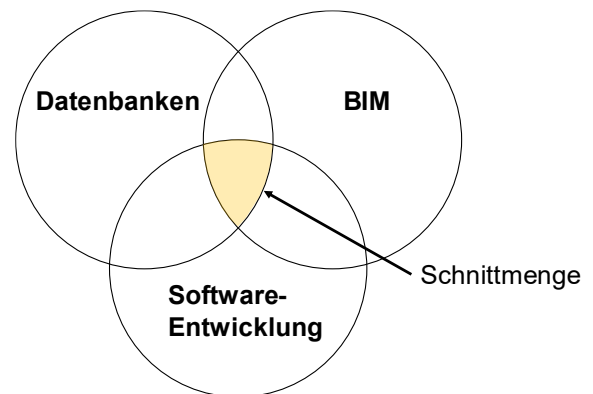


Abbildung 6: Themenfelder und deren Schnittmenge

Relevante Fachliteraturen:

- Krcmar, Helmut. Informationsmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. <https://doi.org/10.1007/978-3-662-45863-1>.
- Borrmann, André, Markus König, Christian Koch, und Jakob Beetz, Hrsg. Building Information Modeling: Technologische Grundlagen und industrielle Praxis. VDI-Buch. Wiesbaden: Springer Fachmedien Wiesbaden, 2021. <https://doi.org/10.1007/978-3-658-33361-4>.
- Broy, Manfred, und Marco Kuhrmann. Einführung in die Softwaretechnik. Xpert.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021. <https://doi.org/10.1007/978-3-662-50263-1>.
- Schicker, Edwin. Datenbanken und SQL. Informatik & Praxis. Wiesbaden: Springer Fachmedien Wiesbaden, 2017. <https://doi.org/10.1007/978-3-658-16129-3>.

## 2.2 Methodik der Literaturrecherche

Die Literaturrecherche erfolgt durch die Anwendung einer systematischen Abfolge von Methoden, die in Abbildung 7 dargestellt wird. Zuerst wird eine allgemeine Suche anhand der Schnellsuche «Quick and Dirty» durchgeführt, um einen Überblick über wissenschaftliche Arbeiten in dieser Thematik zu erhalten und erste relevante Literaturen zu identifizieren. Anschliessend wird der «Block Building Approach» angewendet, der die Thematik in verschiedene Themenblöcke unterteilt und mithilfe von Boole'schen Operatoren die Schnittmenge der Themenblöcke ermittelt, um die Relevanz für die Forschungsfrage zu präzisieren. Mithilfe des «Citation Pearl Growing» werden zusätzliche Literaturquellen durch die Untersuchung von Zitationen in den zuvor gefundenen Arbeiten identifiziert und so die relevanten Literaturen für die Arbeit zusammengestellt. Die Resultate aus der einfachen Suche und die verwendeten Begriffstabellen für die Literaturrecherche ist im Anhang A dokumentiert.

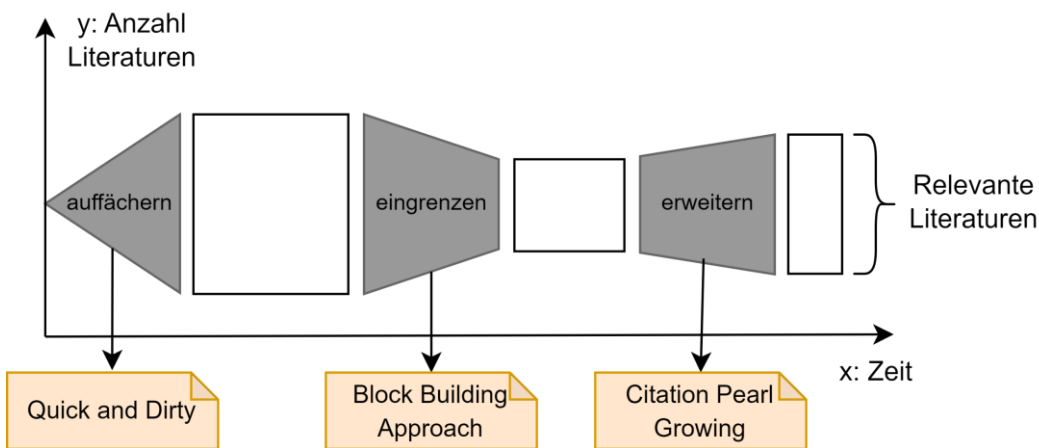


Abbildung 7: Abfolge der Literaturrecherche und angewandeter Methoden in Orange

Die Literaturrecherche stützt sich auf bekannte Literaturdatenbanken. Folgende Literaturdatenbanken werden bei der Literaturrecherche eingesetzt:

Table 1: verwendete Literaturdatenbanken

Literaturdatenbank	Erläuterung des Auswahl	Angewendete Methodik
Swisscovery	Die Plattform bietet Zugang auf die Bestände von Hochschulbibliotheken und weiteren Informationsdiensten in der Schweiz. Es ermöglicht nebst einer einfachen Suche auch eine erweiterte Suche.	«Quick and Dirty»
ResearchGate	Die Plattform bietet eine einfache Suche und erlaubt einen Einblick in die Forschungsergebnisse von Wissenschaftlern.	«Quick and Dirty»
Web of Science und Scopus	Diese Plattformen sind kostenpflichtige Datenbanken, welche viele verschiedene und fortgeschrittene Suchfunktionen erlauben.	«Block Building Approach» und «Citation Pearl Growing»

## 2.3 Methodik der Datenerhebung in der Praxis

Die Datenerhebung verfolgt das Ziel, die Grundlagen für die Anforderungsdefinition der Systemarchitektur zu schaffen und eine Basis für die konzeptuelle Ausarbeitung zu erhalten, wobei der Fokus auf KMUs liegt. Zur Ermittlung des Informationsbedarfs wird das subjektive Verfahren anhand einer offenen Befragung angewendet (Krcmar, 2015, p. 124). Um eine Struktur bei den Fachgesprächen zu erhalten wird ein Leitfaden verwendet, die sich an die Durchführung von Expertengesprächen an (Bogner et al., 2014, p. 27) orientiert und die Anforderungen zum Leitfaden von (Helfferich, 2011, p. 180) berücksichtigt. Zunächst wird ein Sampling zusammengestellt, welches im folgendem Kapitel näher erläutert wird. Nach dem Sampling werden Gesprächspartner:innen gesucht und angefragt. Die Gespräche werden durchgeführt, transkribiert und anschliessend mit Codierungen ausgewertet sowie relevante Informationen identifiziert.

### 2.3.1 Zusammenstellung des Samplings

Das Sampling wird nach eigenen Kriterien für die vorliegenden Forschungsfragen zusammengestellt und im bekannten Umkreis angefragt. Die Auswahl der gewünschten Unternehmen erfolgt basierend auf den folgenden Kriterien:

- Deutschsprachige Kommunikation
- Zugehörigkeit zur Kategorie der KMU
- Tätigkeiten im Sektor der Architektur- und Ingenieurdienstleistungen aufweisen
- Fachwissen im Bereich BIM aufweisen

Bei den ausgewählten Unternehmen werden Gesprächspersonen gesucht, die sowohl über Fachwissen im Bereich der Digitalisierung verfügen als auch in der Geschäftsleitung positioniert sind und von einem Nutzenden der Autorensoftware begleitet werden. Dieser Ansatz ermöglicht es, sowohl die Perspektive der Nutzenden aus Sicht der Datenbank als auch die Perspektive der Nutzenden der Autorensoftware zu erfassen.

### 2.3.2 Gesprächsführung

Die Datenerhebung wird mithilfe eines Leitfadens durchgeführt, der die Fragen und ihre Reihenfolge für die Expertengespräche festlegt. Dabei werden Hauptfragen pro Kapitel definiert. Es wird darauf geachtet, dass die Gespräche flexibel bleiben, um ein offenes Gespräch zu ermöglichen. Innerhalb der Hauptfragen orientieren sich die Gespräche an den Antworten der Befragten, um die Deutung der gesprochenen Inhalte erfahren zu können. Der Leitfaden ist im Anhang B aufgeführt und gliedert sich in vier Kapitel: Einleitung, Modellierungsprozess, Datenbanken und Anforderungen.

### 2.3.3 Auswertung

Die Analyse der Fachgespräche beginnt mit der Transkription der aufgenommenen Gespräche. Diese werden dann mithilfe der Software «MAXQDA 2022» anhand des vorher festgelegten Leitfadens kategorisiert, um die Hauptcodes (Basiscodierung) zu erhalten und die Gesprächsthemen innerhalb der Transkripte zu gliedern. Anschliessend werden die Gespräche in feinere Subcodes (Feincodierung) differenziert. Die Codes werden zur Auswertung genutzt, um Erkenntnisse zu gewinnen. Auf eine Gewichtung der codierten Segmente wird verzichtet.

## 2.4 Methodik der Entscheidungsfindung von Datenbanktechnologien

Bei der Entscheidungsfindung der geeigneten Datenbanktechnologie wird die Methode «Choosing by Advantages» nach Jim Suhr angewendet (Suhr, 1999). Die Methode basiert auf der Wichtigkeit von Vorteilen der verschiedenen Alternativen. Ein Punktesystem von 0 bis 100 beurteilt die Wichtigkeit des Vorteils gegenüber Alternativen für jeden Bewertungsfaktor. Eine höhere Zahl im Punktesystem weist auf eine bessere Bewertung hin.

## 2.5 Methodik der semantischen Beschreibung

In der semantischen Beschreibung werden die Anforderungen an die zu entwickelnde Systemarchitektur definiert. Diese Methodik zeichnet sich durch eine formale Herangehensweise aus, bei der schriftliche Beschreibungen als auch Darstellungen mit Diagrammen verwendet werden. Die Methodik ist primär fachlich orientiert. Das Ziel ist es die Systemanforderungen klar und verständlich für Anwender dokumentieren zu können. Für die Notation der semantischen Beschreibung wird die textuelle freie Form verwendet. Anhand von einfachen Skizzen wird die Beschreibung visuell gestützt (Rosenthaler & Schildknecht, 2010).

### 2.5.1 Anforderungsanalyse

Laut Broy und Kuhrmann muss die Grundstruktur der Anforderungen an ein Softwaresystem von Anfang an klar definiert sein. Daher folgt die Beschreibung der Anforderungsanalyse der Methodik von (Broy & Kuhrmann, 2021, p. 202) welche die Grundstrukturen Vision, Funktion, Nutzung und Details zur Beschreibung der Anforderungen verwendet.

### 2.5.2 Anwendungsfallmodellierung

Der Vorgang zur Beschreibung der Anwendungsfälle erfolgt schrittweise gemäss Abbildung 8. Im ersten Schritt wird das Anforderungsprofil der KMUs zusammengetragen. Im zweiten Schritt werden die Anwendungsfälle mit dem Praxispartner auf Grundlage des Anforderungsprofils erarbeitet. Zuletzt werden die Anwendungsfälle in einem Systemprozess beschrieben. Für die einzelnen Anwendungsfälle wird die Beschreibungsform nach dem «Use Case Specification Template» nach Cockburn (Broy & Kuhrmann, 2021) in vereinfachter Form verwendet. Zuletzt werden die Interaktion zwischen den Akteuren/Akteurinnen und dem System anhand des Anwendungsfalldiagramms der UML-Modellierungssprache aus Abbildung 9 beschrieben.

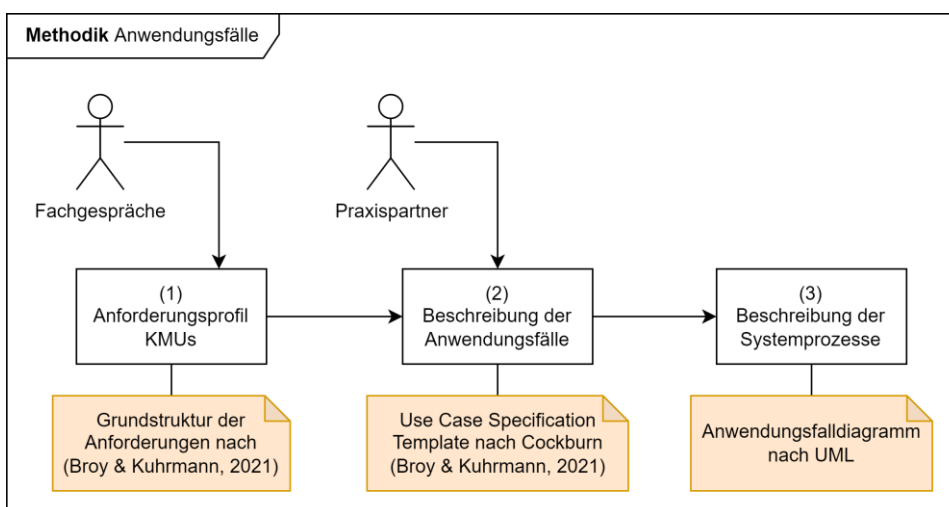


Abbildung 8: Methodik zur Beschreibung der Anwendungsfälle

## 2.6 Methodik der konzeptuellen Beschreibung

In der konzeptuellen Beschreibung wird die semantische Beschreibung konkretisiert und mit formaler Sprache beschrieben. Dabei werden Diagramme aus der objektorientierten Modellierungssprache UML verwendet. Damit bildet die konzeptuelle Beschreibung die Schnittstelle zwischen den Entwicklern und der Fachwelt (Rosenthaler & Schildknecht, 2010).

### 2.6.1 Notation

Das UML beinhaltet unterschiedliche Diagramme zur Beschreibung des vorliegenden Modells. Die Diagramme werden in Struktur- und Verhaltensdiagramme kategorisiert. Die Abbildung 9 zeigt die möglichen Diagrammartentypen im UML auf. Die Orange markierten Diagramme werden in dieser Arbeit verwendet.

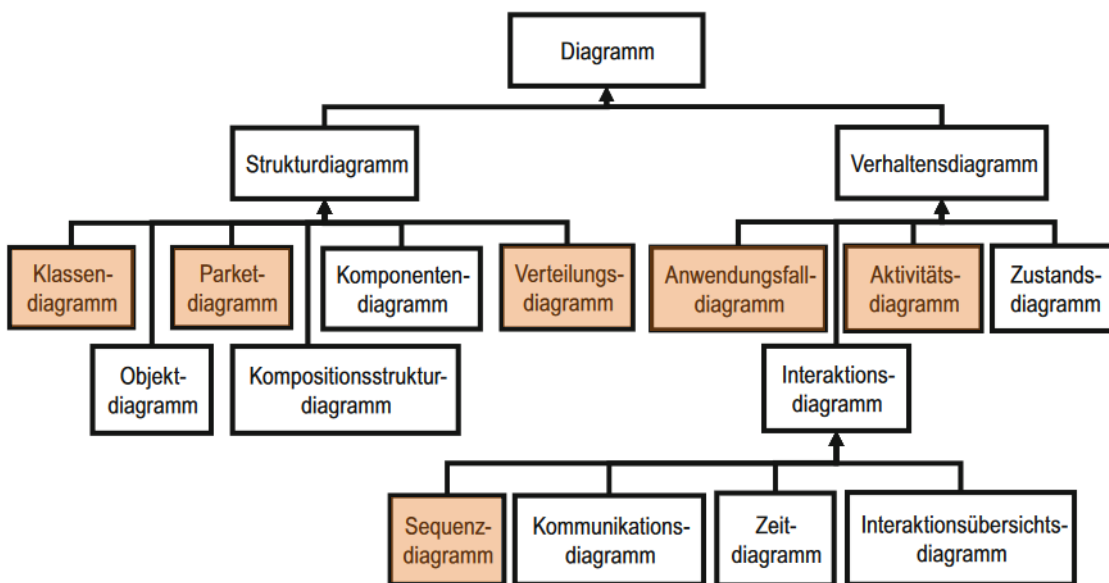


Abbildung 9: UML Diagrammartentypen, Quelle: (Krcmar, 2015, p. 37)

Nebst dem UML werden auch die grafische Darstellungsform des ERM mit der Chen-Notation für die Beschreibung der Datenbankschematas verwendet.

### 2.6.2 Modellierung

Die Prozessmodellierung nutzt textuelle Beschreibungen sowie Skizzen zur visuellen Unterstützung. Die Anwendungsfälle werden gemäss dem Anwendungsfalldiagramm modelliert. Die Modellierung der Systemarchitektur erfolgt mit Hilfe der Paket- und Verteilungsdiagramme. Die Datenmodellierung umfasst die Verwendung von Klassendiagrammen für das IFC-Datenmodell und ER-Diagrammen aus dem ERM für das Datenbankschema. Abläufe werden durch Aktivitäts- und Sequenzdiagramme modelliert.

## 2.7 Methodik der logischen Umsetzung

In der logischen Umsetzung wird die konzeptuelle Beschreibung anhand eines Proof of Concepts technisch erprobt und damit die Umsetzbarkeit beurteilt. Im Anschluss konkretisieren die Erkenntnisse die konzeptuelle Beschreibung. Dabei werden Diagramme aus dem UML und die Sprachelemente aus der Programmierung verwendet.

### 2.7.1 Erstellung des Proof of Concepts

Die logische Umsetzung kann gemäss Broy und Kuhrmann einerseits als horizontales Prototyping oder als vertikales Prototyping betrachtet werden. Die beiden Formen des Prototypings sind in der Abbildung 10 dargestellt, wobei das vertikale Prototyping rot umrahmt ist. Die logische Umsetzung erfordert eine systematische Herangehensweise. Der horizontale Prototyp realisiert einen ausgewählten Ausschnitt eines Systems. Der vertikale Prototyp realisiert einen Teil des Systems vollständig über alle Schichten der Architektur hinweg (Broy & Kuhrmann, 2021). Für die logische Umsetzung des Proof of Concepts wird die Methode des vertikalen Prototypings angewendet, um die Umsetzbarkeit der konzeptuellen Überlegungen zu demonstrieren.

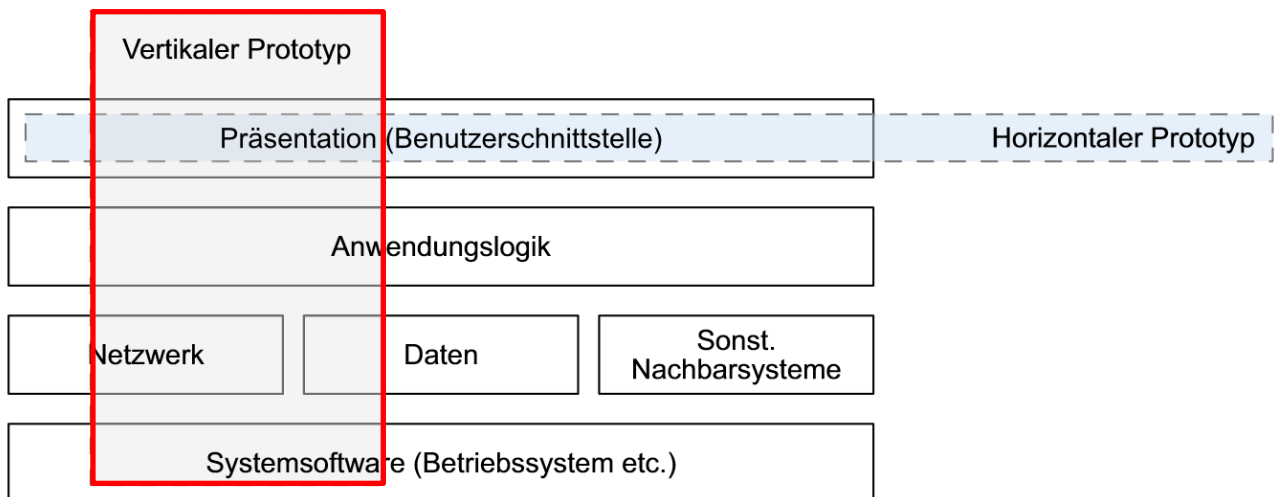


Abbildung 10: Horizontale und vertikale Prototypen nach Balzer, Quelle: (Broy & Kuhrmann, 2021)

### 2.7.2 Rückmeldungen aus der Praxis

Den Erhalt von Rückmeldungen zur logischen Umsetzung erfolgt mit einer Vorstellung des Proof of Concepts und mit einer anschliessenden offenen Befragung beim Praxispartner. Es wird ein Raum für konstruktive Kritik geschaffen. Zum Schluss wird dies als Erfahrungsbericht verschriftlicht.

### 3 Stand der Forschung

Zurzeit hängt der Austausch von Informationen über die Transferdatei IFC-SPF erheblich von der interoperablen IFC-Datenmodell interpretierenden Softwares ab. Dadurch ist die Datenmanipulation nur mit Hilfe von solchen Softwares möglich. In GIS ist es seit längerem bereits möglich unterschiedliche Fachdaten in verteilten Dateien oder Datenbanken zu verwalten (SIA 2051, 2017, p. 32). Die Autoren Buttgerit und Emara schlagen in Ihrer Arbeit datenbankbasierte DBMs als Alternative vor (Buttgerit & Emara, 2023, p. 33). Zudem wird auch erwähnt, dass eine IFC Server Applikation als Multi-User Datenbanksystem mit einer Datenbankstruktur basierend auf dem IFC-Datenmodell eine Lösung darstellt, um effizient Gebäudedaten auszutauschen und managen zu können (Lee et al., 2014). Seit dem Jahre 2000 wird versucht das gesamte IFC-Datenmodell in verschiedenen Datenbanktechnologien abzubilden (Buttgerit & Emara, 2023; Lee et al., 2014).

Zhu und Shide haben in ihrer Studie die Übersetzung des IFC-Datenmodells auf ein relationales Datenbanksystem untersucht und dabei die semantische Diskrepanz zwischen den beiden Sprachen SQL und EXPRESS adressiert. Die Diskrepanz liegt darin, dass das IFC-Datenmodell auf dem objektorientierten Konzept basiert und das RDB dies nicht tut. Es sind keine standardisierten Übersetzungsmethoden von STEP-basierten Modellen zu relationalen Datenbankmodellen vorhanden (You & Yang, 2004). Deshalb wurden verschiedene Übersetzungsmethoden von unterschiedlichen Studien erprobt und untersucht (You & Yang, 2004; Wei et al., 2009; Zhu & Shide, 2021). Aktuell ist das Dateiformat von SQLite (.sqlite) zur Abbildung des IFC-Datenmodells in relationale Datenbank SQLite in der Liste von IFC-Formaten von BuildingSMART International als Experimental kategorisiert und wird offiziell nicht unterstützt (BuildingSMART International, 2022). Im Forum «OSArch» wurde ein quelloffener Prototyp für die zeitgleiche Bearbeitung des DBMs vom Nutzer Dion Moullet präsentiert, indem ein IFC-kompatibles Datenbankmodell auf Basis von SQLite oder MySQL verwendet wird (Dion Moullet, 2023).

Aus der Praxis sind einige Implementierungen vorhanden. Das Unternehmen Nemetschek Group bietet für die Branche die kommerzielle Software «dRofus» an, welche als ein datenzentrierter Ansatz für BIM präsentiert wird («dRofus Software,» 2023). Im Jahr 2021 wurde von ERNE Holzbau AG ein eigener Lösungsansatz im Forum «OSArch» vorgestellt, welcher es erlaubt, die alphanummerischen Informationen der DBM in einer externen Datenbank abzuspeichern und die Daten über eine Weboberfläche zu manipulieren (Frédéric Beaupère (ERNE Holzbau AG), 2021). Es geht auch einfacher. Im Beispiel von Autodesk Revit hat DiRoots LDA die Applikation «SheetLink» auf den Markt gebracht, welches erlaubt alphanummerische Informationen aus dem Revit in eine Exceldatei zu exportieren, diese zu verändern und wieder in Revit zu importieren. Dies geschieht über einen manuellen Prozess anhand einer GUI (DiRoots, n.d.). Die Implementierungen verdeutlichen, dass in der Praxis ein Bedarf an spezifischer Software zur Datenverwaltung der DBM und deren Interaktion mit der Autorensoftware besteht.

## 4 Praktischer Einblick in KMUs

Das Hauptziel der Datenerhebung in der Praxis ist es einen Einblick in den Modellierungsprozess von DBM, in der Art und Weise der Beschreibung der Modellen und den Einsatz von Datenbanken im Unternehmen zu erhalten. Am Ende resultiert ein generalisiertes Anforderungsprofil von KMUs, welches in der konzeptuellen Beschreibung als Grundlage dienen soll.

Die Datenerhebung beginnt mit der Untersuchung der Modellierungsstufe der DBM, gefolgt von einer Analyse der beteiligten Akteure/Akteurinnen und ihrer Rollen. Im Anschluss wird der Modellierungsprozess, die fachliche Beschreibung der Modelle untersucht. Zuletzt wird untersucht, welche Datenbanktechnologien im Unternehmen zum Einsatz kommen.

### 4.1 Das Sampling

Es wurden insgesamt elf Fachgespräche mit Unternehmern durchgeführt. Vorwiegend handelt es sich um kleine Unternehmen aus Zürich, wie in Abbildung 11 und Abbildung 14 dargestellt. Die Unternehmen gehören vermehrt, wie in Abbildung 12 visualisiert, den Segmenten der Architektur und Gebäudetechnik an. Die Gespräche wurden vorwiegend mit den Geschäftsinhaber:innen und leitenden Angestellten geführt. Die Gesprächspartner:innen repräsentierten in erster Linie interne fachkundige Personen, welche zum Teil auch von sachbearbeitenden Personen begleitet wurden.

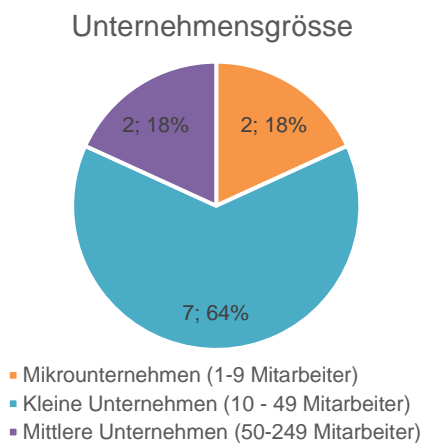


Abbildung 11: Unternehmensgrösse der befragten Unternehmen

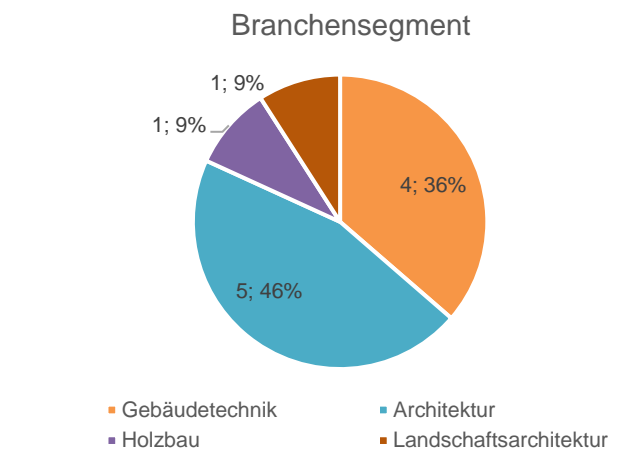


Abbildung 12: Branchensegment der befragten Unternehmen

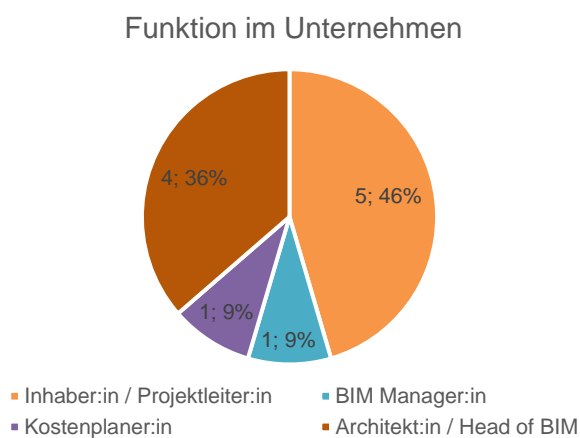


Abbildung 13: Funktionen der befragten Personen



Abbildung 14: Ortschaften der befragten Unternehmen

### 4.2 Modellierungsstufen von DBM

Es wird untersucht, in welchen Modellierungsstufen (siehe Tabelle 2) sich Unternehmen zum Zeitpunkt der Datenerhebung orientieren. Das Hauptziel besteht darin, die Anwendungskennnisse von Unternehmen im Bereich der BIM-Methodik zu bewerten, um daraus Erkenntnisse zu gewinnen, inwiefern die semantische Beschreibung von DBM eine Rolle spielt. Hierzu wird eine Differenzierung der Modellierungsstufen definiert und die erhobenen Daten entsprechend kategorisiert. In der Analyse sind die folgenden Stufen festgelegt:

Tabelle 2: Modellierungsstufen von DBM

Stufen	Beschreibung
Stufe 0: «Kein Modell»	Ein Modell ist nicht vorhanden.
Stufe 1: «3D»	Der Fokus liegt nur bei der geometrischen Modellierung.
Stufe 2: «3D und wenig Semantik»	Informationen werden nur bei Bestellung gepflegt sowie Mengenauszüge für die weiterführende Arbeit generiert.
Stufe 3: «3D und volle Semantik»	Informationen werden aus eigenem Interesse verwaltet und ausgewertet. Bestrebte Entwicklungen z.B. in Bereichen Kosten, Termine und Klassifikationen sind vorhanden.

Die Stufen lehnen sich nicht an den BIM-Reifegradstufen der Bauindustrie vom britischen BIM Task Group (Borrmann et al., 2021). Die Fachgespräche können mehrere Stufen aufweisen, wenn die Zuweisung an eine Stufe nicht konkret bestimmt werden kann. Die beobachtende Analyse zeigt folgendes Bild:

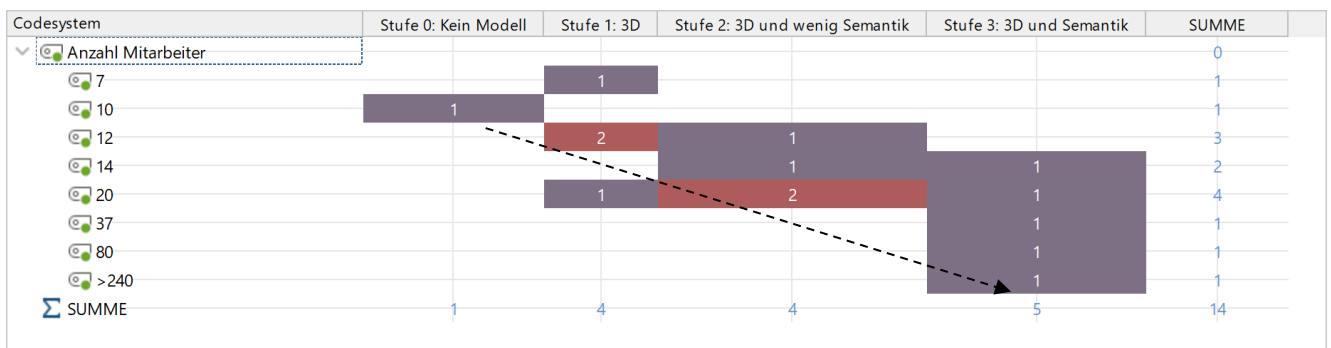


Abbildung 15: Auswertungstabelle «Modellierungsstufen von DBM» (MAXQDA 2022)

Die begrenzte Stichprobengröße deutet darauf hin, dass Mikrounternehmen tendenziell in der Modellierungsstufe 1 verharren, während kleine und mittlere Unternehmen Bestrebungen zeigen, die Modellierungsstufe 3 zu erreichen. Des Weiteren scheint die Tragbarkeit der Modellierungsstufe 3 ab einer Mitarbeiteranzahl von etwa 14 gegeben zu sein.

### 4.3 Rollentrennung im Modellierungsprozess

Nachfolgend wird ebenfalls überprüft, ob die hypothetische Annahme, die besagt, dass Unternehmen zwei unterschiedliche Rollen im Modellierungsprozess identifizieren, bestätigt werden kann. Hierbei werden zwei Rollen angenommen: die fachmodellierende Person, die für die geometrische und semantische Modellierung verantwortlich ist, sowie die fachprojektleitende Person, welche sich um die Leitung und Organisation des Projektes kümmert.

Folgende Kategorien sind für die Untersuchung definiert:

- Keine Rollentrennung
- Je nach Projektgröße und Komplexität
- Rollen klar getrennt

Die beobachtende Analyse zeigt folgendes Bild:

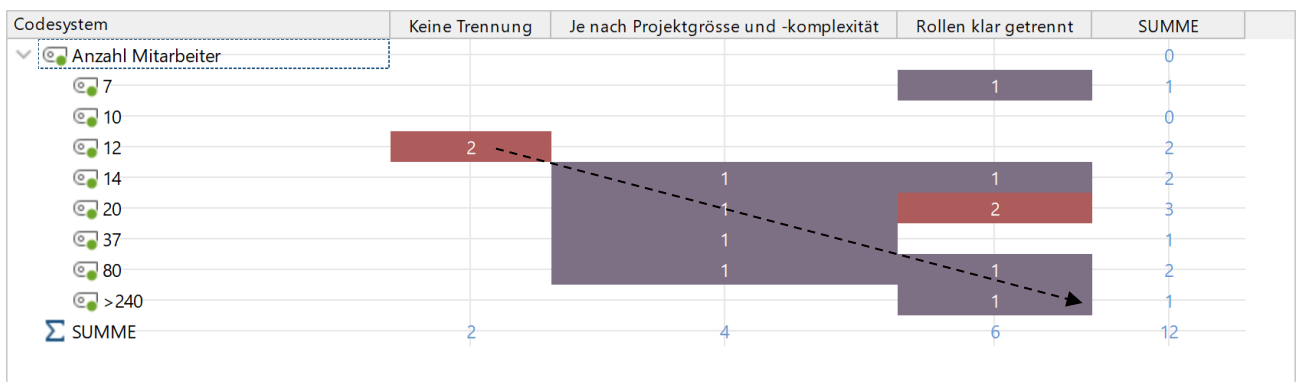


Abbildung 16: Auswertungstabelle «Rollentrennung im Modellierungsprozess» (MAXQDA 2022)

Es zeigt sich groberweise, dass kleine- und mittlere Unternehmen tendenziell eine deutliche Trennung der Rollen aufweisen. Im Fall von einem Fachgespräch weicht das beobachtete Verhalten von diesem Muster ab. Ab einer Mitarbeiterzahl von etwa 14 neigt die Organisation dazu, sich in Richtung einer klareren Trennung der Rollen zu entwickeln.

Im Rahmen der Untersuchung wird zusätzlich die Frage gestellt, ob die Rollentrennung für die fachmodellierende Person als herausfordernd empfunden wird und ob die Modellierungsstufe einen Einfluss darauf hat. Es lässt aus der Abbildung 17 jedoch kein klar erkennbares Muster feststellen, da die Meinungen divers sind und die Anzahl an Fachgesprächen zu gering dafür ist.

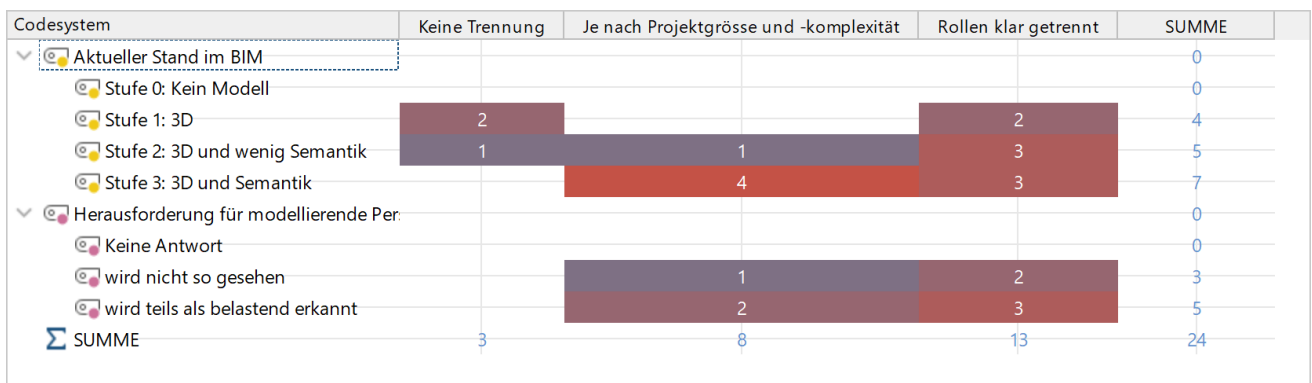


Abbildung 17: Auswertungstabelle «Herausforderung für fachmodellierende Personen» (MAXQDA 2022)

#### 4.4 Prozess zur semantischen Informierung des DBMs

Bei der Untersuchung des Modellierungsprozesses wird erwartet, dass ein Muster erkannt wird, um die aktuelle Situation bei den KMUs beschreiben zu können. Dabei wird beobachtet, wie der Prozess der semantischen Beschreibung der DBM innerhalb des Unternehmens abläuft.

Als Resultat zeigt sich, dass die semantische Beschreibung in mehreren Schritten erfolgt. Im ersten Schritt werden die Informationen hauptsächlich in der Autorensoftware eingetragen. Im zweiten Schritt werden die Informationen aus dem Autorensoftware exportiert in Dateiformaten wie PDF, CSV oder IFC-SPF und damit als Eingabe in weitere Prozesse verwendet. Einige Unternehmen gehen jedoch einen Schritt weiter und nutzen externe Tools wie Solibri, BIMCollab, BIG/Kaulquappe, DBDBIM usw., um die exportierte IFC-SPF aus dem Autorensoftware durch die Zugabe von weiteren Informationen weiterzuverarbeiten. Die Informationen dienen dazu, Auszüge zu generieren oder diese in weiteren Prozessschritten zu verarbeiten. Ein Beispiel hierfür ist die Klassifizierung nach eBKP-H oder BKP zur Kostenermittlung im Hochbau, welches von CRB, einer schweizerischen Zentralstelle für Baurationalisierung, herausgegeben wird. Dabei werden Herausforderungen genannt, wie beispielsweise, dass die verarbeitete IFC-SPF nicht in das DBM in der Autorensoftware zurückübertragen wird. Der grobe Umriss des Prozesses wird in der Abbildung 18 aufgezeigt.

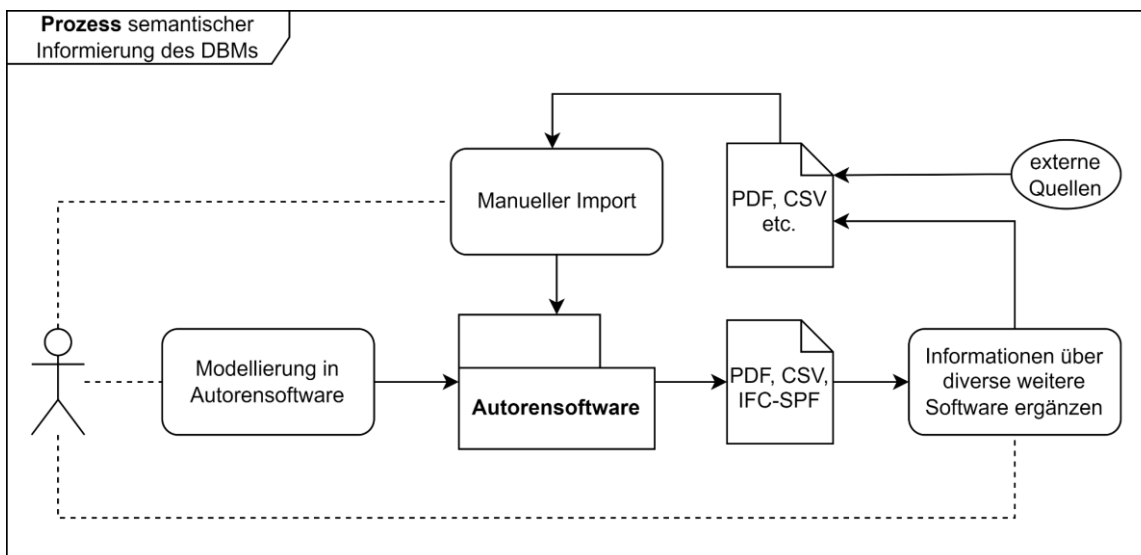


Abbildung 18: Modellierungsprozess bei den KMUs

Der Informationsaustausch in Bauprojekten erfolgt zwischen den Unternehmen durch die Verwendung von Excel-Dateien mit Import- und Exportprozessen. Alternativ werden PDFs oder andere Formate genutzt, die dann manuell in die Autorensoftware übertragen werden.

Daten, welche ins Autorensoftware zurückfließen sind mehrheitlich ein geringer Anteil. Die Daten fließen mit manuellem Import in Autorensoftware zurück. Generell wird in den Fachgesprächen festgestellt, dass eine bidirektionale Synchronisation zwischen der Autorensoftware und den verschiedenen anderen Tools nicht vorliegt. Es gibt unterschiedliche Ansichten in Unternehmen hinsichtlich der zentralen Bedeutung des Modells in der Autorensoftware. Einige betrachten es als zentral, während andere diese Perspektive nicht teilen.

### 4.5 Semantische Beschreibung von DBM

Im Rahmen der Untersuchung wird analysiert, welche IFC-Konzepte bei der semantischen Beschreibung der DBM im Unternehmen zu tragen kommen. Die Gespräche orientieren sich an der fachlichen Auseinandersetzung. Diese Entscheidung beruht darauf, dass die technische Kompetenz der Gesprächspartner:innen sowie das Wissen über IFC-Konzepte als unzureichend oder teilweise nicht vorhanden beurteilt wird. Die technische Umsetzung der IFC-Konzepte in der betrieblichen Praxis können in weiterführende Untersuchungen angeknüpft werden. Die Antworten zur semantischen Beschreibung werden im Anschluss auf die IFC-Konzepte übersetzt.

	Gespräch 1	Gespräch 2	Gespräch 3	Gespräch 4	Gespräch 5	Gespräch 6	Gespräch 7	Gespräch 9	Gespräch 10	Gespräch 11	Total
☑ Semantik im DBM (fachlich)											
☑ Materialisierung						8.3%	10.0%	22.2%			5.1%
☑ Projekt/Raumstrukturierung	8.3%						10.0%	11.1%			3.8%
☑ Aussehen				14.3%		8.3%	10.0%			16.7%	6.4%
☑ Mengen und Stückzahlen	8.3%	33.3%	33.3%	14.3%	33.3%	25.0%	10.0%	22.2%	50.0%	33.3%	23.1%
☑ Terminplanung	8.3%	33.3%				8.3%				8.3%	5.1%
☑ Klassifizierung	33.3%			28.6%	16.7%	16.7%	10.0%	11.1%		8.3%	15.4%
☑ Kosten	8.3%	33.3%	33.3%	14.3%	33.3%		10.0%		25.0%	8.3%	11.5%
☑ Eigenschaften	16.7%		33.3%	28.6%	16.7%	33.3%	30.0%	11.1%	25.0%	25.0%	23.1%
☑ Typisierung	16.7%						10.0%	22.2%			6.4%
Σ SUMME	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
# N = Dokumente/Sprecher	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	1 (10.0%)	10 (100.0%)

Abbildung 19: relevante IFC-Konzepte aus fachlicher Sicht (MAXQDA 2022)

Gemäss Abbildung 19 wird gezeigt, dass über alle Unternehmensgrössen verstärkt Modelle für «Mengen» und «Stückzahlen» erstellt werden, wobei verschiedene «Eigenschaften» genutzt und «klassifiziert» werden. Einzelne Unternehmen setzen zudem auf die Beschreibung von «Materialisierung» sowie «Aussehen» ihrer Modelle und setzen den Fokus auf die Typisierung von Bauteilen.

Die Nutzung von «Kosten» ist bereits bei mittleren Unternehmen etabliert, während andere noch in der Entwicklungsphase dieses Aspekts sind. Bisher spielt das Thema «Termine» in den Modellierungsprozessen keine herausragende Rolle. Mittlere Unternehmen befinden sich aktuell in der Entwicklungsphase hinsichtlich Kosten- und Terminplanung über Modelle.

Es ergibt sich eine Reihenfolge, in der die Beschreibung der Eigenschaften, Mengen und Stückzahlen am häufigsten verwendet werden, gefolgt von der Klassifizierung der Modelle. Zuletzt werden die Kosten, die Definition von Typen, das Aussehen des Modells und schliesslich die Terminplanung berücksichtigt.

### 4.6 Datenbanken im Unternehmen

Es wird die Nutzung von Datenbanktechnologien im Unternehmen untersucht, wobei der Fokus darauf liegt, festzustellen, ob die Unternehmen über Kompetenzen im Umgang mit Datenbanken verfügen und bereits entsprechende Technologien im Einsatz haben. Die Erkenntnisse aus den Gesprächen legen nahe, dass mehr als 60% der Unternehmen gegenwärtig keine Datenbanken nutzen, was möglicherweise auf mangelndes Know-how oder Ressourcen zurückzuführen ist (siehe Abbildung 20). Interessanterweise zeigt sich bei Unternehmen mit einer Mitarbeiteranzahl zwischen 20 und 37 eine verstärkte Tendenz zur Integration von Datenbanken. Des Weiteren lässt sich feststellen, dass mittlere Unternehmen bereits aktiv Datenbanken einsetzen. Eine Besonderheit stellt das Unternehmen mit 10 Mitarbeitern dar, da dieses Unternehmen als Einziger eine präzise Softwarelösung für den Umgang mit Daten benennen konnte.

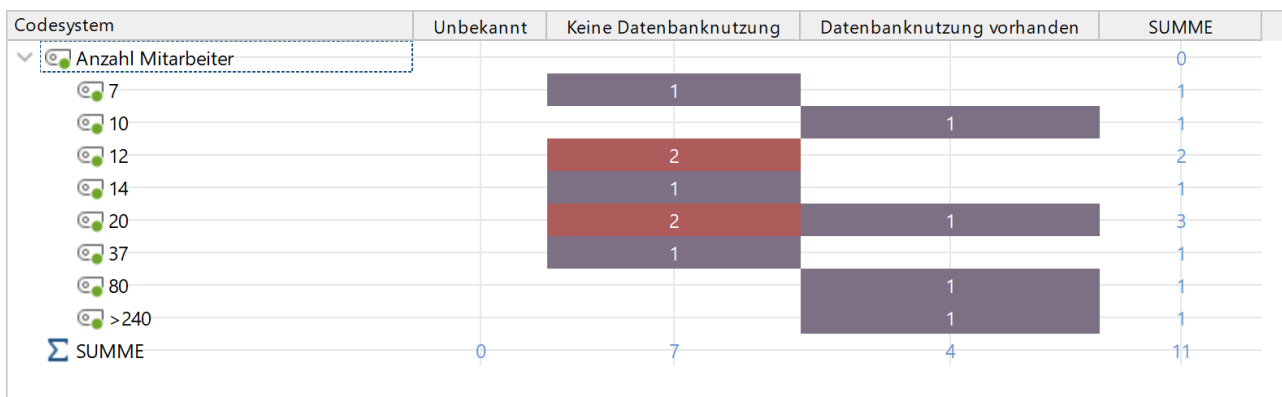


Abbildung 20: Auswertungstabelle Datenbanknutzung im Unternehmen (MAXQDA 2022)

Des Weiteren wird von den Gesprächspartner:innen in den Gesprächen vermehrt die Nutzung von Excel als Datenbank erwähnt. Da Excel keine Datenbanktechnologie ist, wird es in der Abbildung 21 nicht aufgeführt. Diese Präferenz kann darauf zurückzuführen sein, dass die Datenkompetenz in den Unternehmen begrenzt ist. Dies legt nahe, dass Excel als eine zugängliche Plattform zur Strukturierung von Daten wahrgenommen und genutzt wird. Deshalb wird Excel im Kapitel 5 beim Vergleich der Datenbanktechnologien als Tabellenkalkulationsprogramm mitberücksichtigt. Ziel ist es zu untersuchen, ob Excel nicht besser für KMUs geeignet ist als Datenbanksysteme. Folgende Systeme wurden als Datenbanken erwähnt:

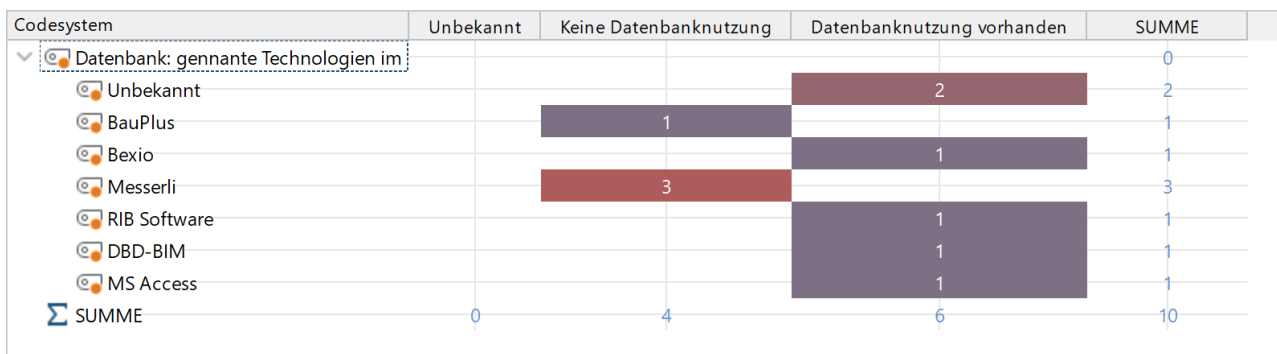


Abbildung 21: Auswertungstabelle genannter Datenbanken im Unternehmen (MAXQDA 2022)

#### 4.7 Generalisiertes Anforderungsprofil der KMUs

Der erfolgreiche Abschluss einer Softwareentwicklung ist entscheidend davon geprägt, ob das vollendete System den Vorstellungen aller beteiligten Akteure/Akteurinnen gerecht wird (Broy & Kuhrmann, 2021, p. 200). Das Anforderungsprofil aus der Anforderungsanalyse dient als Leitfaden, um Erwartungen von Nutzenden resp. von den relevanten Stakeholdern zu definieren und sicherzustellen, dass die Ziele erreicht werden können. Dabei wird die gewünschte Lösung den Gesprächspartner:innen vorgestellt und deren Anforderungen und Vorstellungen mit offenen Fragen erhoben. Das Anforderungsprofil teilt sich in Bereiche der Vision, Funktion, Nutzung sowie Details auf und werden weiter unten erläutert. Die Markierungen mit «(a..)» wird dazu verwendet, um die Anforderungen im weiteren Projektverlauf adressieren zu können. Im Anhang E sind die genannten Anforderungen im Detail als Quelle platziert.

Vision	Die Bewirtschaftung der Informationen im DBMs ist durch das System zu erleichtern. Für geschäftsinterne Nutzung ist eine Nachvollziehbarkeit von Änderungen nicht von grosser Bedeutung. Hingegen für externe Zugriffe ist es von entscheidender Bedeutung die Autorenschaft klar zu deklarieren und die Zugriffe zu regulieren <b>(a1)</b> . Die Kosten sollen dabei im kostengünstigen Rahmen bleiben <b>(a2)</b> . Die reibungslose Funktion im Hintergrund ist eine Anforderung, jedoch soll die Lösung nicht in einer Blackbox enden, sondern eine Opensource-Option ist wünschenswert <b>(a12)</b> . Die Archivierung für die kommenden zehn Jahre muss berücksichtigt werden <b>(a13)</b> . Die Lösung soll systemunabhängig sein <b>(a14)</b> .
Funktion	Die Anwendung soll die Möglichkeit bieten, Typen in der Datenbank zu erstellen und durch Listen zu verwalten <b>(a4)</b> . Des Weiteren ist der Export von PDF- und Exceltabellen für Ausschreibungen zu ermöglichen <b>(a5)</b> . Zur Sicherung der Datenqualität ist eine regelbasierte Kontrolle in Kombination mit einer modellbasierten Darstellung durchführen zu können <b>(a6)</b> . Eine integrierte History-Funktion soll dabei die Nachverfolgung von Datenänderungen garantieren <b>(a7)</b> . Wichtig ist dabei, dass diese Funktionalitäten im Hintergrund nahtlos arbeiten. Die Ausrichtung auf eine bidirektionale Synchronisation zwischen der Datenbank und dem DBM variiert je nach Unternehmensansicht. Dennoch besteht der Wunsch, dass Informationen im Autorensoftware direkt sichtbar sind und nicht über separate Tools eingesehen werden müssen <b>(a8)</b> .
Nutzung	Eine benutzerfreundliche Oberfläche mit einer visuellen Komponente hilft die einzutragenden Daten in einer 3D-Ansicht nachvollziehbar zu machen <b>(a9)</b> . Intuitives Arbeiten mit niedrigen Hürden und eine einfach verständliche Nutzung werden vorausgesetzt, um eine schnelle und unkomplizierte Erfassung von Daten zu ermöglichen <b>(a10)</b> . Die Zugänglichkeit über einen Webbrowser ist ebenfalls ein zentrales Anforderungskriterium <b>(a11)</b> . Die Anwendung des Systems soll nicht nur im Modellierungsprozess begrenzt werden. Es ist wünschenswert, dass die Daten auch zur Auswertung verwendet werden können <b>(a3)</b> .
Details	Es sind keine expliziten Details zu Schnittstellen oder technischen Implementierungen vorhanden.

Um den Umfang dieser Arbeit im Rahmen der Beantwortung der Fragestellungen zu halten, werden nur ausgewählte Teile (a2, a3, a4, a7, a8, a12, a13, a14) des umfangreichen Anforderungskatalogs berücksichtigt. Abgegrenzt wird zu GUI, Modellansichten, Zugriffsmanagement und dem Export sowie Anknüpfungen zu weiteren Datenbanken.

## 5 Datenbanktechnologien im Vergleich

Die Untersuchung des Einsatzes von Datenbanksystemen im Modellierungsprozess von DBM erfordert die Auswahl einer geeigneten Datenbanktechnologie. Diese Entscheidung beeinflusst sowohl die Konzeption als auch die Implementierung des Proof of Concepts. Zuerst werden die gängigen Datenbanktechnologien in der Praxis untersucht. Anschliessend werden diese Technologien mithilfe der CBA-Methode miteinander verglichen und das Resultat daraus dient letztendlich als Grundlage für die Entscheidung einer Datenbanktechnologie.

### 5.1 In der Praxis übliche Datenbanktechnologien

Derzeit stellt «solid IT» in der Praxis eine Rangliste der nachfolgenden relationalen Datenbankmanagementsysteme vor. Die entsprechende Methode wird gemäss den Erläuterungen von «solid IT» wie folgt dargelegt (solid IT, 2023):

- Anzahl der Erwähnungen des Systems auf Websites
- Allgemeine Interessen im System
- Häufigkeit der technischen Diskussionen über das System
- Anzahl der Stellenangebote, in denen das System erwähnt wird
- Anzahl der Profile in beruflichen Netzwerken, in denen das System erwähnt wird
- Relevanz in sozialen Netzwerken

Die Top 10 der bewerteten RDBMS dienen als Grundlage für die Entscheidungsmatrix (siehe Abbildung 22). Diese Auswahl basiert auf der Annahme, dass die Top 10 der Datenbanktechnologien die grösste Anwenderbasis, Entwicklerunterstützung und Dokumentation aufweisen, was eine reibungslose Implementierung begünstigen soll. In diesem Kontext wird auch Microsoft Excel als Tabellenkalkulationsprogramm in den Vergleich einbezogen. Dies geschieht aufgrund seiner als unverzichtbar erachteten Rolle in Architektur- und Ingenieurbüros, wie sich durch Fachgespräche mit Experten/Expertinnen herausgestellt hat.

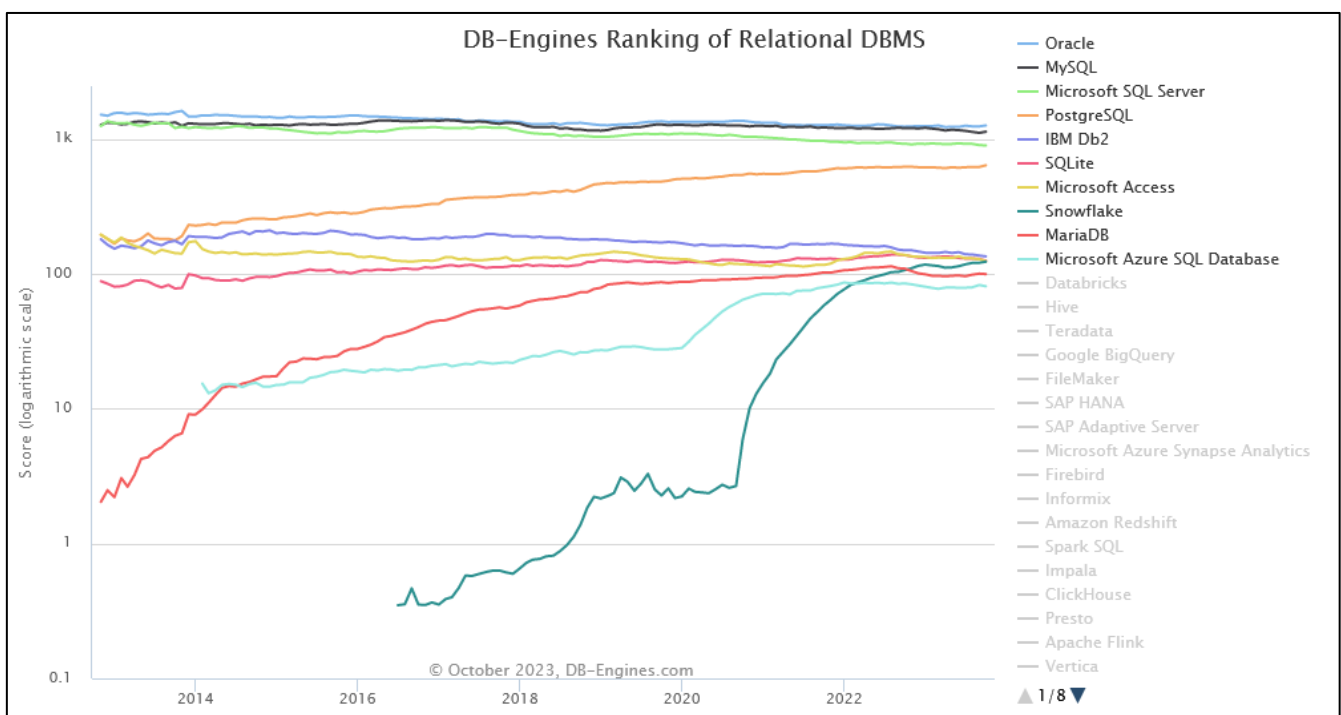


Abbildung 22: Rangliste der meistgenutzten RDBMS, Quelle: (solid IT, 2023), Datenstand Oktober 2023

## 5.2 Bewertungsfaktoren

Die acht Bewertungsfaktoren aus der Tabelle 3 werden sowohl aus den Anforderungen in Kapitel 4.7 als auch aus den allgemeinen RDBMS-Faktoren von «Solid IT» abgeleitet. Es ist dabei zu beachten, dass die Bewertungen der Datenbanktechnologien nicht nur auf objektiven Fakten beruhen, sondern auch subjektive Wahrnehmungen enthalten können. Die Bewertungen basieren hauptsächlich auf den Beobachtungen aus Fachgesprächen, dem zeitlichen Rahmen und der Umsetzbarkeit.

Tabelle 3: Bewertungsfaktoren für die Entscheidungsmatrix von RDBMS

Faktor	Beschreibung	Kriterium
Systemabhängigkeit	Systemunabhängigkeit bedeutet, dass die Datenbanktechnologie nicht von Hardware, Software oder Betriebssystem abhängig ist. Lizenzierung und Interoperabilität zeigen die Unabhängigkeit der Software.	Je unabhängiger, desto besser
Open Source	Die Lizenz der Datenbanktechnologie verdeutlicht den Grad der Offenheit und Transparenz der zugrunde liegenden Software.	Je offener der Quellcode, desto besser
Kosten	Die Kosten beziehen sich auf die finanziellen Aufwendungen, um die Datenbanktechnologie anwenden zu können. Die Preise aus offiziellen Quellen zeigen die verbundenen finanziellen Aufwendungen auf.	Je kostengünstiger, desto besser
Dokumentation und Bekanntheit	Die Dokumentation und Bekanntheit beurteilt die Verfügbarkeit und Verbreitung von Informationen über die Datenbanktechnologie innerhalb der Community. Es wird angenommen, dass die Vielfalt der Dokumentationen die Bekanntheit der Technologie beeinflusst, die durch Trendanalysen in Internet-Suchportalen aufgezeigt wird.	Je höher die Bekanntheit, desto besser
Zugriffskonzept	Das Zugriffskonzept betrachtet die Mechanismen, die den Zugang zur Datenbank regeln und kontrollieren. Die offiziellen Quellen zeigen auf, wie die Zugriffskonzepte der zugrunde liegenden Technologie aussehen.	Je mehr Funktionen zur Regelung der Zugriffe auf die Datenbank, desto besser
Einfachheit	Die Einfachheit bezieht sich darauf, wie einfach und intuitiv es ist, die Datenbank zu erstellen, zu verwalten, abzufragen und zu aktualisieren.	Je einfacher und intuitiver, desto besser.
Passend zum Anwendungsfall	Bezieht sich darauf, inwieweit die Datenbanktechnologie dem Anwendungsfall dieser Arbeit entspricht und geeignet ist.	Je passender der Anwendungsfall, desto besser
Umsetzbarkeit in der Master-Thesis	Bezieht sich darauf, inwieweit innerhalb der begrenzten Zeit, Ressourcen und Rahmenbedingungen der Arbeit realisierbar ist, einschliesslich der Verfügbarkeit von Daten, Methoden und Expertise.	Je eher die Technologie in dieser Arbeit eingesetzt werden kann, desto besser

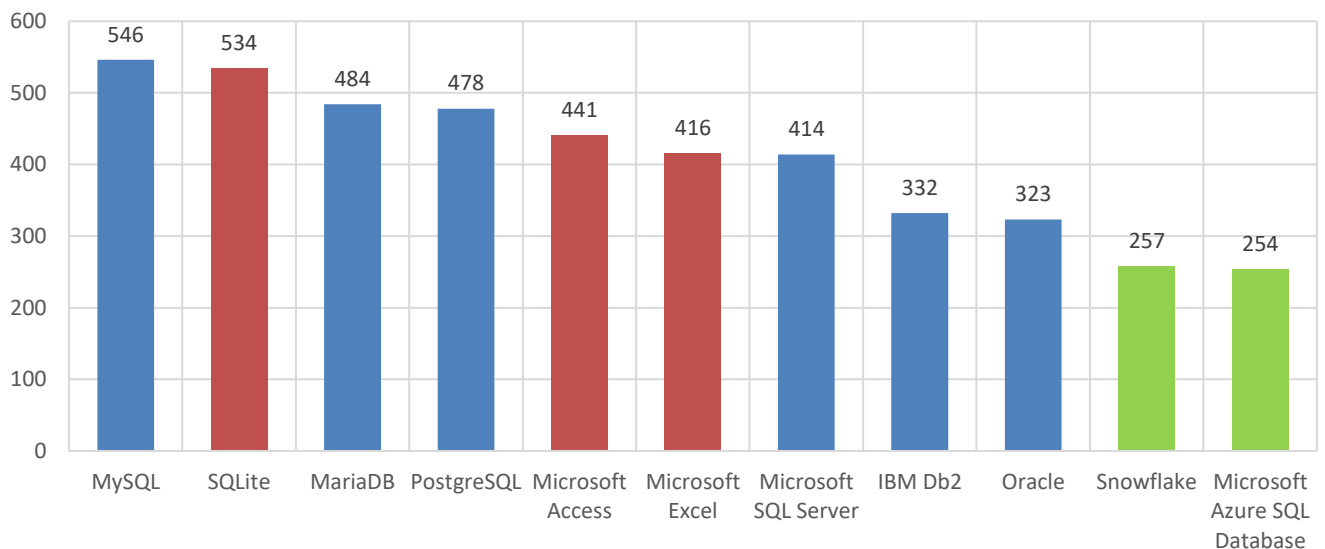
Die geeigneten Datenbanktechnologien werden in drei Kategorien unterteilt: dateibasierte-, serverbasierte und cloudbasierte Datenbanken, welche folgend erläutert sind:

*Table 4: Kategorien der Datenbanktechnologien*

Kategorie	Beschreibung
Dateibasierte Datenbanken	Dateibasierte Datenbanken speichern die Datenbankmodelle und die darin liegenden Daten in einer einzigen Datei ab.
Serverbasierte Datenbanken	Serverbasierte Datenbanken sind Datenbanken, die zentral auf einem Server gehostet werden und von Clientanwendungen über das Netzwerk zugänglich sind.
Cloudbasierte Datenbanken	Cloudbasierte Datenbanken bieten serverbasierte Datenbanken als «Database as a Service» an. Es erleichtert den Endnutzenden die Konfiguration, Einrichtung und Nutzung der Datenbank.

### 5.3 Resultat aus der Entscheidungsmatrix

Anhand der Entscheidungsmatrix (Anhang E) wird das Resultat in der Abbildung 23 dargestellt. Die y-Achse definiert die erreichte Punktzahl. Die maximal mögliche Punktzahl ist 800. Die Datenbanktechnologie MySQL zählt derzeit zu den führenden Datenbanktechnologien im Bereich der RDBMS. Angesichts der Kriterien aus Kapitel 5.2 und der verfügbaren Online-Dokumentationen ist es nicht überraschend, dass MySQL den Spitzenplatz einnimmt. Im Gegensatz dazu belegt SQLite als dateibasierte Lösung den zweiten Platz. Das weitbekannte Tabellenkalkulationsprogramm Microsoft Excel besetzt den mittleren Platz. Die letzten Plätze besetzen die cloudbasierten Datenbanktechnologien.



*Abbildung 23: Resultat Entscheidungsmatrix RDBMS (blau: serverbasiert, rot: filebasiert, grün: cloudbasiert)*

## 5.4 Entscheid der Datenbanktechnologie

Aus dem Resultat in der Abbildung 23 werden die beiden bestbewerteten Datenbanktechnologien für die Entscheidung berücksichtigt. Zur Auswahl stehen zwei Technologien aus unterschiedlichen Kategorien, die in der Bewertung eng beieinander liegen: MySQL als serverbasierte Datenbank und SQLite als dateibasierte Datenbank. Eine serverbasierte Datenbank würde beim Proof of Concept die Einrichtung des Servers, der Datenbank und des Datenbanksystems erfordern, was zusätzliche Komplexität mit sich bringt. Daher fällt die Wahl in dieser Arbeit auf SQLite. Die Adaption des Konzepts von SQLite auf MySQL sollte jedoch möglich sein, wie das Beispiel von Dion Moulton verdeutlicht, das beide Lösungen berücksichtigt (Dion Moulton, 2023).

SQLite ist eine Bibliothek in C-Programmiersprache, welche eine SQL-Datenbank implementiert (SQLite, 2024a). Die kostenfreie und gemeinfreie Datenbanktechnologie SQLite bietet einen einfachen Einstieg ohne Notwendigkeit einer Vorkonfiguration, wie etwa der Einrichtung von Servern, Verbindungen und Zugriffsverwaltung. Dank der Interoperabilität ermöglicht die Technologie eine Interaktion mit allen gängigen Datenbankmanagementsystemen (Runebook.dev, 2022).

Die Datenbanktechnologie SQLite findet diverse Verwendungen. Für den Anwendungsfall dieser Arbeit eignet sich SQLite als Applikationsdateiformat (SQLite, 2024b). Als Applikationsdateiformat wird die SQLite-Datenbank für Desktopanwendungen direkt auf der Festplatte des Rechners eingesetzt. Folgende Limitationen werden dabei beachtet:

- Desktopanwendung direkt auf dem Rechner
- Kein Zugriffsberechtigungskonzept
- Einzelbenutzersystem

Die Datenbank kann aber auch für Mehrbenutzersysteme auf einem Server deponiert werden. Es ist zu beachten, dass SQLite Lese- und Schreibsperrungen verwendet, um den Datenbankzugriff zu steuern. Wenn mehrere Rechner über das Netzwerk auf ein Dateisystem zugreifen und die SQLite-Datenbank bearbeiten, kann es zu Problemen mit dem Sperrmechanismus kommen. Je nach Verwendung kann diese Limitation als vernachlässigbar betrachtet werden (SQLite, 2023). Auf eine Serveranwendung der SQLite-Datenbank wird aufgrund der Komplexität in dieser Master-Thesis nicht eingegangen.

## 6 Fachliche Beschreibung des Kontextes

Basierend auf der Anforderungsanalyse erfolgt die fachliche Beschreibung in der semantischen Ebene des Kontexts. Die Untersuchung konzentriert sich auf die KMUs im Bereich der Architektur- und Ingenieurbüros. Dabei erfolgt lediglich eine periphere Betrachtung angrenzender Unternehmen, wie beispielsweise Lieferanten oder ausführende Firmen. Dies liegt in der Tatsache, dass diese Unternehmen die Rolle von «externen Fachleuten» einnehmen, die Fachwissen und spezifische Daten in ein Projekt einbringen. Die Annahme legt vor, dass ausschliesslich Architektur- und Ingenieurbüros den Prozess der Modellierung des gemeinsamen DBMs, oder des disziplinspezifischen Fachmodellierens durchlaufen. Innerhalb der Architektur- und Ingenieurbüros werden im Modellierungsprozess beteiligten Personen zwischen «fachmodellierende Person» und «fachprojektleitende Personen» differenziert. Es wird nicht ausgeschlossen, dass eine Person beide Rollen einnehmen kann. Die Differenzierung erlaubt eine Strukturierung der Rollen und die dazugehörigen Aufgaben (siehe Abbildung 24).

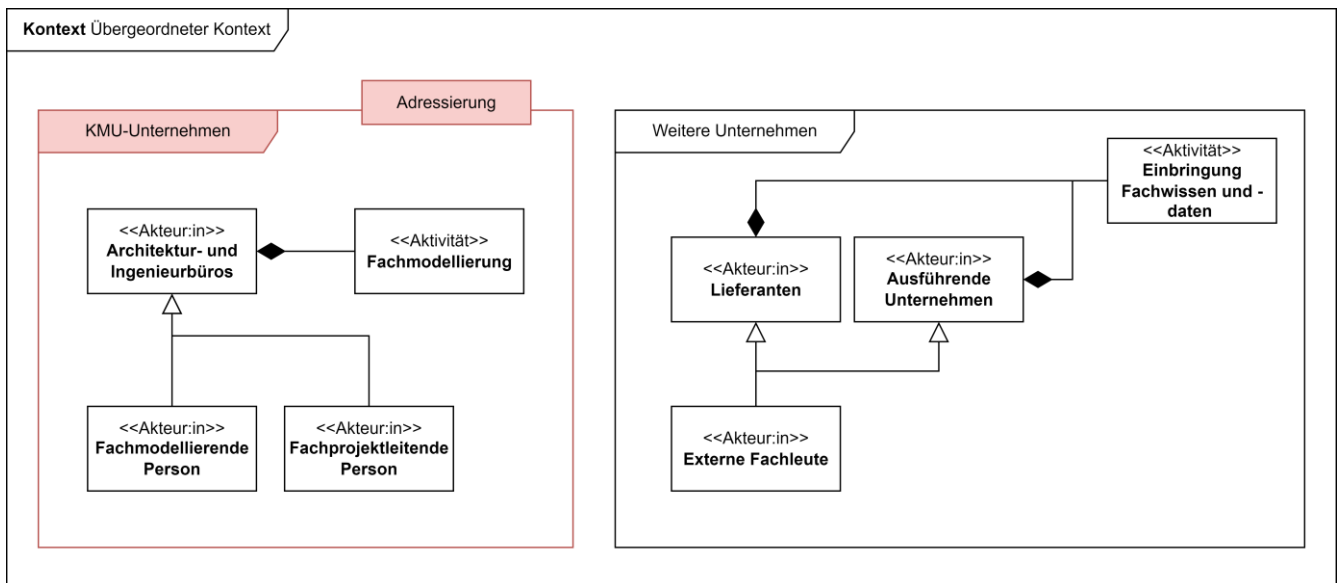


Abbildung 24: Beschreibung des übergeordneten Kontextes

## 6.1 Rollen im Modellierungsprozess

Eine Rolle in der Projektorganisation bezeichnet vorübergehend die Funktion einer Person. Sie umfasst Aufgaben, Verantwortlichkeiten und Befugnisse. Die Aufgaben einer Rolle werden durch die zugehörigen Prozesse definiert, für die sie verantwortlich ist (SIA 2051, 2017). Innerhalb des beschriebenen Kontextes sind die Rollen des fachmodellierenden und des fachprojektleitenden Person involviert.

### 6.1.1 Fachmodellierende Person

Die fachmodellierende Person (Abbildung 25) hat die Fachkompetenz im jeweiligen Fachgebiet z.B. Hochbauzeichner:in oder Gebäudetechnikplaner:in. Die Person wird vermehrt im Unternehmen als Sachbearbeiter:in oder als Modellierer:in eingesetzt und ist im Umgang mit den gängigen Autorensoftwares versiert. Die Autorensoftware ist für die Person ein alltägliches Werkzeug. Die Rolle wird auch als sekundäre:r Akteur:in im System bezeichnet.

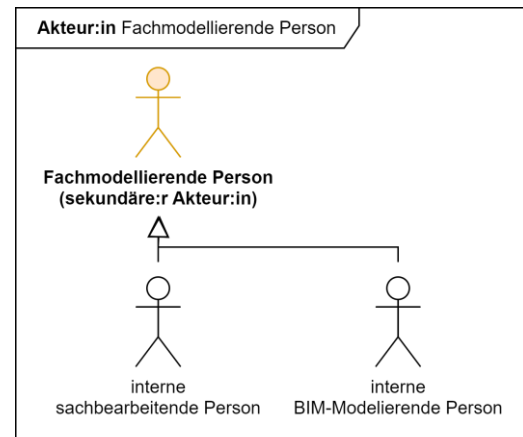


Abbildung 25: Fachmodellierende Person

### 6.1.2 Fachprojektleitende Person

Die fachprojektleitende Person (Abbildung 26) hat die Fachkompetenz im jeweiligen Gebiet. Die Person hat keine bis schwache Modellierungskenntnisse. Das Autorensoftware wird nicht bis selten als Werkzeug verwendet. Die Person wird im Unternehmen vermehrt als Projektleiter:in, Teamleiter:in oder als Expert:in eingesetzt. Die Rolle wird auch als primäre:r Akteur:in im System bezeichnet.

Die fachprojektleitende Person ist in zwei weiteren Rollen spezialisiert: als externe Fachleute und interne Fachleute. Externe Fachleute sind ausserhalb von Architektur- und Ingenieurbüro tätige Personen. Sie haben zurzeit keinen bis geringen Bezug zu DBM. Die externen Fachleute werden im Verlauf der Projekte kontaktiert und nach Ihren Produkten, Fachwissen oder deren Dienstleistungen angefragt. Interne Fachleute sind im Unternehmen tätige Personen, welche die Rollen von Fach- oder Gesamtprojektleitungen übernehmen. Sie sind verantwortlich für die Definierung von Konzepten, Entwürfen und den einzusetzenden Produkten in Fachmodellen.

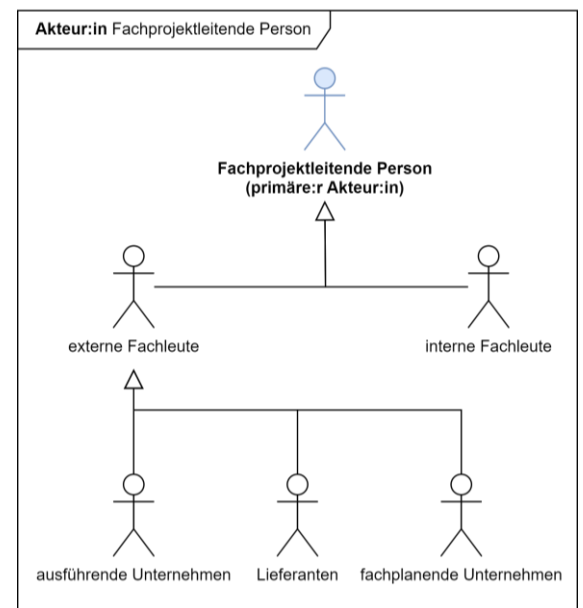


Abbildung 26: Fachprojektleitende Person

## 6.2 Gesamtprozess

Die Untersuchung setzt am Modellierungsprozess von digitalen Bauwerksmodellen an. Gemäss Abbildung 27 wird die aktuelle Abfolge von Aktivitäten zur Modellierung von disziplinspezifischen Fachmodellen in einem internen Unternehmen dargestellt. Dies leitet sich aus dem Kapitel 4.4 und dem Mit einbezug des Praxispartners ab. Zunächst wird die Autorensoftware für das Bauprojekt eingerichtet, und anschliessend wird mit der 3D-Modellierung begonnen. Danach erfolgt die Auswahl der einzusetzenden Produkte. Die Informationen zu den Produkten werden über PDF oder Excel ausgetauscht und schliesslich in das DBM eingepflegt. Der Informationsaustausch erfolgt zurzeit zwischen externen und internen fachprojektleitenden Personen und zwischen fachmodellierenden und fachprojektleitenden Personen.

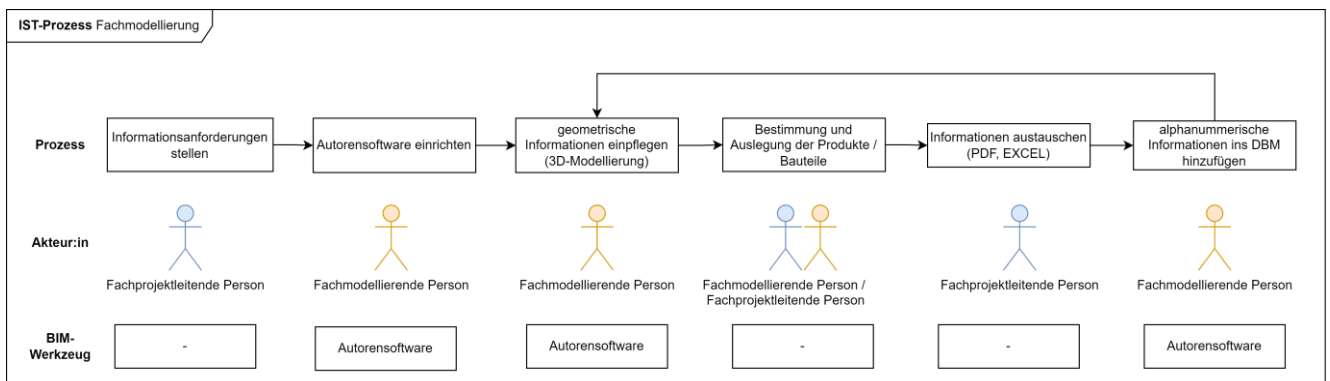


Abbildung 27: Beschreibung des IST-Prozesses

Das Ziel besteht darin, den Informationsaustausch über verschiedene Datenquellen zu minimieren und einen direkten Zugang zum DBM für die Datenpflege zu ermöglichen. Die Abbildung 28 veranschaulicht diese Beschreibung und ermöglicht eine Vereinfachung der Prozessschritte mithilfe der Datenbank. Die gestrichelten grauen Bereiche zeigen die Änderungen zwischen dem IST-Prozess und SOLL-Prozess.

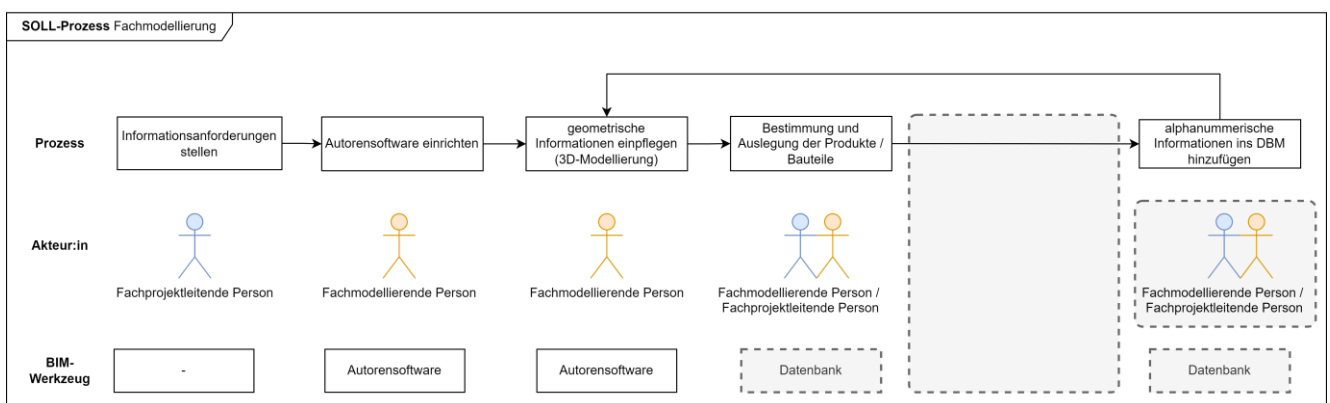


Abbildung 28: Beschreibung des SOLL-Prozesses im Zusammenhang mit der gewünschten Systemarchitektur

### 6.3 Kontextabgrenzung

Die fachliche Kontextabgrenzung, wie von den Autoren Stark & Hruschka beschrieben, hat den Zweck, das Umfeld des Systems sowie dessen Beziehungen statisch zu erfassen (Starke & Hruschka, 2011, p. 57). Dabei handelt es sich um eine Top-Level-Darstellung mit den für das Projekt relevanten Aspekten. Die Abbildung 29 veranschaulicht die Kontextabgrenzung in dieser Arbeit.

Die Hauptaufgabe des Systems besteht darin, dem/der Akteur:in eine Schnittstelle für den Zugriff auf die Datenbank bereitzustellen, Aktualisierungen in der Datenbank durchzuführen und eine Verbindung zur Autorensoftware herzustellen. Des Weiteren umfasst die Aufgabe die Änderungen in der Datenbank an die Autorensoftware zu übermitteln sowie die geometrischen Informationen von der Autorensoftware in der Datenbank zu aktualisieren. Im System wird einerseits die Datenbank und andererseits die Anwendung «Middleware», welche die Systemaktivitäten ausführt, beheimatet.

Bei einer erweiterten Betrachtung des Konzeptes könnten zusätzliche, benachbarte Systeme integriert werden, beispielsweise für die Erstellung von Berichten oder zur Übertragung von Daten an eine übergeordnete Datenbank. Dies bezieht sich auf die Anforderungen «a3» und «a5» aus dem Kapitel 4.7. In dieser Arbeit wird jedoch nur das zentrale System und die Autorensoftware als benachbarte Systeme betrachtet.

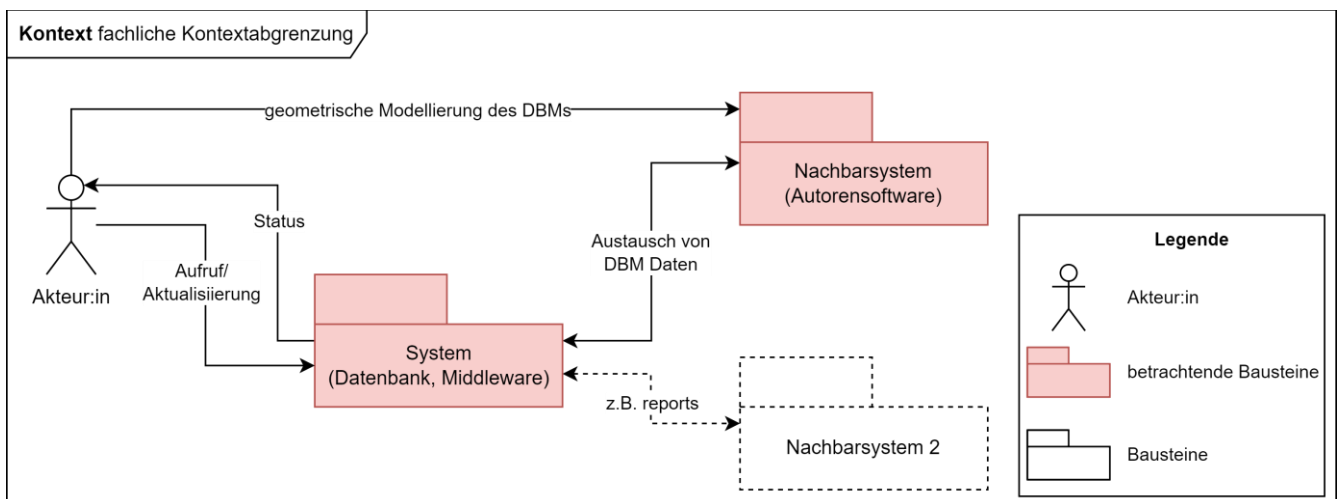


Abbildung 29: fachliche Kontextabgrenzung in Anlehnung an (Starke & Hruschka, 2011)

## 6.4 Anwendungsfälle

In Zusammenarbeit mit dem Praxispartner, einem Mikrounternehmen im Bereich Architektur und Ingenieurwesen, werden die Anwendungsfälle erarbeitet. Die Anwendungsfälle basieren auf den generalisierten Anforderungen aus dem Kapitel 4.7 und spezifizieren die Interaktion zwischen dem Nutzenden und dem System.

### Anwendungsfall 1: Informationsanforderungen stellen

- Ziel:** Erforderliche und relevante Informationen sind im DBM aufzufinden.
- Kurzbeschreibung:** Der Export von Informationen aus der Autorensoftware soll so konfiguriert werden, dass die benötigten und relevanten Informationen für die Nutzung der Datenbank enthalten sind. Daher ist es erforderlich, die Informationsanforderungen für den Nutzenden, der die Autorensoftware bedient und einrichtet, im Voraus zu beschreiben.
- Vorbedingung:** Das Wissen über die notwendigen Informationen ist vorhanden.
- Ablaufbeschreibung:**
- Beschreibung der Informationsanforderungen
  - Zustellung an die Nutzende Person der Autorensoftware
  - Einrichtung des Exports von IFC-SPF
- Ergebnis:** Die IFC-SPF entspricht den geforderten Informationsanforderungen

### Anwendungsfall 2: Initialisierung der Datenbank

- Ziel:** Aktuelle Daten sind in der Datenbank vorhanden.
- Kurzbeschreibung:** Die Datenbank ist beim Starten zu initialisieren. Dadurch werden die aktuellen Daten aus dem DBM in die Datenbank übertragen. Ist das System zu diesem Zeitpunkt leer, wird die Datenbank erstellt und die Daten übertragen.
- Vorbedingung:** Das DBM basiert im IFC-Datenmodell. Die Daten aus dem DBM sind vorhanden und aktuell.
- Ablaufbeschreibung:**
- Anwendung starten
  - DBM lesen
  - Datenbank erzeugen
  - Daten übertragen
  - Meldung übergeben
- Ergebnis:** Die aktuellen Daten aus dem DBM können über die Datenbank betrachtet und weiterverarbeitet werden. Das Ergebnis dieses Anwendungsfalls definiert zudem auch den Anwendungsfall «Daten einsehen».

**Anwendungsfall 3: Meldungen erhalten und einsehen**

- Ziel: Der Status im System sowie erfolgreiche und fehlgeschlagene Systemprozesse sind bekannt.
- Kurzbeschreibung: Nach jeder Ausführung eines Prozesses sind Meldungen mit einer Zusammenfassung dem Nutzenden zu übergeben.
- Vorbedingung: Ein Prozess wurde gestartet.
- Ablaufbeschreibung:
- Prozess angestossen
  - Prozess beendet
  - Mitteilung übergeben
  - Mitteilung anzeigen und lesen
- Ergebnis: Mitteilung erhalten.

**Anwendungsfall 4: Daten in der Datenbank erfassen und speichern**

- Ziel: Eingesehene Daten sind erfasst.
- Kurzbeschreibung: Die Daten werden in der Datenbank erfasst, verändert und gelöscht. Der Anwendungsfall beschränkt sich auf das Aktualisieren vorhandenen oder nicht vorhandenen Daten.
- Vorbedingung: Die Datenbank ist erfolgreich initialisiert und kann verwendet werden.
- Ablaufbeschreibung:
- Daten einsehen
  - Daten aktualisieren
  - Die Datenbank speichert die aktualisierten Daten in der Festplatte
  - Eine Systemanwendung wird gestartet, um die aktualisierten Daten im DBM zu schreiben
- Ergebnis: Daten sind aktualisiert in DBM.

**Anwendungsfall 5: Daten prüfen**

- Ziel: Gültige Daten oder nicht gültige Daten sind bekannt.
- Kurzbeschreibung: Die Prüfung der Daten kann semantisch und technisch erfolgen. Semantische Prüfung bezieht sich auf die Überprüfung durch die fachliche Kompetenz des Benutzers. Dabei sind spezifische Ansichten für die Prüfung zu erstellen. Die technische Prüfung erfolgt mithilfe der integrierten Prüffunktionen der Datenbanktechnologie.
- Vorbedingung: Die Datenbank ist einsatzbereit.
- Ablaufbeschreibung:
- Verbindung zur Datenbank
  - Technische Prüfung
  - Semantische Prüfung
  - Verarbeiten der Daten
- Ergebnis: Technisch und semantisch validierte Daten in der Datenbank.

In der Zusammenarbeit resultieren weitere Anwendungsfälle, welche sich nicht im Betrachtungsbereich befinden. Die betroffenen Anwendungsfälle adressieren die Zugriffskonzepte, den GUI zur Erstellung von Ansichten und den Export an weitere Nachbarssysteme:

- Anwendungsfall 6: Anmelden im System
- Anwendungsfall 7: Ansichten erstellen
- Anwendungsfall 8: Daten exportieren
- Anwendungsfall 9: Datenbank mit Externen teilen

## 7 Konzeptuelle Beschreibung

Die konzeptuelle Beschreibung ist eine systemunabhängige und formale Beschreibung des Vorhabens. Dies ermöglicht das Vorhaben flexibel mit verschiedenen Systemen und Bausteinen anzuwenden. Aufbauend auf der fachlichen Beschreibung wird die konzeptuelle Beschreibung des Systems erstellt. Sie legt den Grundstein für die weitere Planung, Entwicklung und Implementierung des Systems. Die Erfahrungen aus der Entwicklung des Proof of Concepts (Kapitel 8) fließen in die konzeptuelle Beschreibung ein.

### 7.1 Systemarchitektur

Die Systemarchitektur setzt sich aus verschiedenen Komponenten zusammen, darunter einer «Datenbank», einer «Middleware», dem offenen und interoperablen Dateiformat «IFC-SPF» sowie aus einer «Autorensoftware». Die Abbildung 30 zeigt die zentralen Bausteine, den Einsatzort und die Interaktionen zwischen ihnen auf. Die Bauteile «Middleware», «Datenbank» und der indirekte Baustein «Datenbank Interface» bilden gemeinsam die Anwendung. Der Baustein «Datenbank Interface» wird im groben Umfang behandelt und wird als indirekter Baustein deklariert. Die Bausteine «IFC-SPF» und «Autorensoftware» gelten dabei als das DBM. Beide Bausteine repräsentieren in deren Formaten das Bauwerk digital. Um die Komplexität zu verringern und die Machbarkeit in einem einfacheren Kontext zu überprüfen, wird auf die Anwendung der Systemarchitektur in einem verteilten System verzichtet. Die Systembausteine sind im selben Dateisystem als Einsatzort untergeordnet.

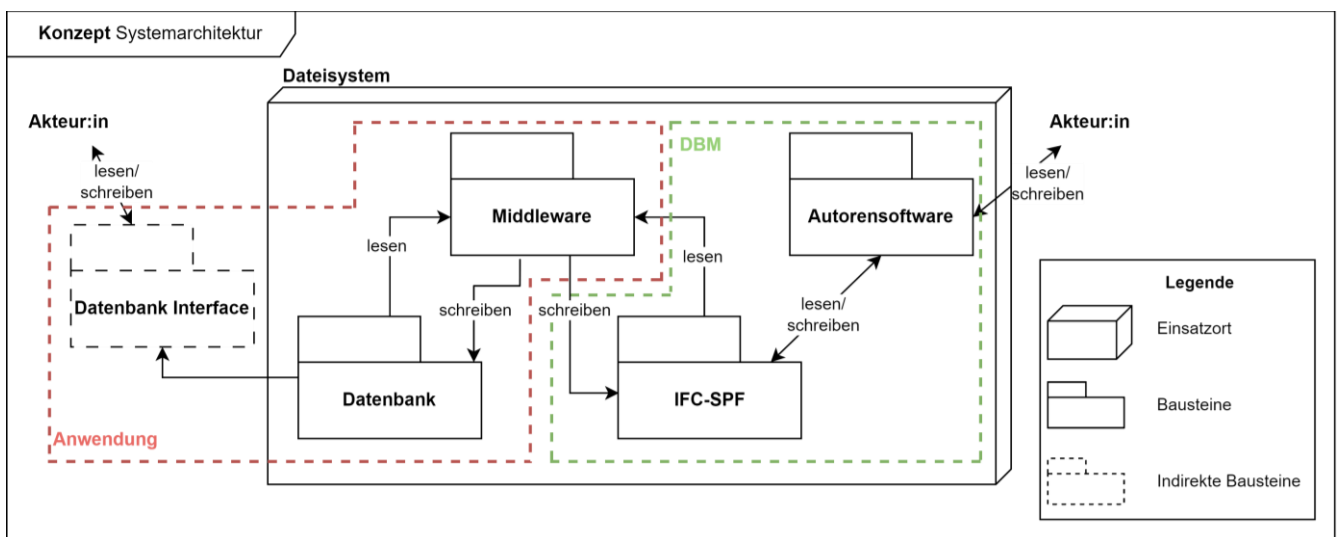


Abbildung 30: Resultierende Systemarchitektur

### 7.2 Systemprozesse

Die erforderlichen Systemprozesse und deren Zusammenhänge sind in Abbildung 31 anhand der beschriebenen Anwendungsfälle aus Kapitel 6.4 dargestellt. Zudem wird ein Bezug zu den Anforderungen aus dem Kapitel 4.7 hergestellt. Die Systemarchitektur aus Abbildung 30 definiert die Grundstruktur des Anwendungsfalldiagramms. Im Anwendungsfalldiagramm werden interaktive und systemrelevante Anwendungsfälle unterschieden. Die interaktiven Anwendungsfälle werden von den primären und sekundären Akteuren/Akteurinnen initiiert. Zwischen diesen Anwendungsfällen bestimmen die verketteten systemrelevanten Anwendungsfälle die Funktionalität des Systems. Die Kernfunktionen des Systems umfassen die Systemprozesse zum Schreiben und Validieren von Daten, die Übermittlung von Mitteilungen sowie die Erstellung des Datenmodells.

Der Anwendungsfall «Informationsanforderungen stellen» ermöglicht die Einrichtung der Autorensoftware sowie den Export von IFC-SPF. Der Anwendungsfall «Datenbank initialisieren» umfasst die Erzeugung der Datenbank, einschliesslich des Schreibens von Daten in die Datenbank. Wenn Daten in der Datenbank eingesehen oder erfasst werden, stösst dies den Systemprozess «Daten schreiben» an, der sich in das Schreiben von Daten in die Datenbank, in IFC-SPF und in die Autorensoftware aufteilt. Jeder Schreibprozess beinhaltet die technische Validierung von Daten und die Übermittlung von Meldungen an die Akteure/Akteurinnen. Die Validierung soll nicht nur technisch, sondern auch semantisch erfolgen, wobei Letztere von den Akteuren/Akteurinnen durchgeführt werden.

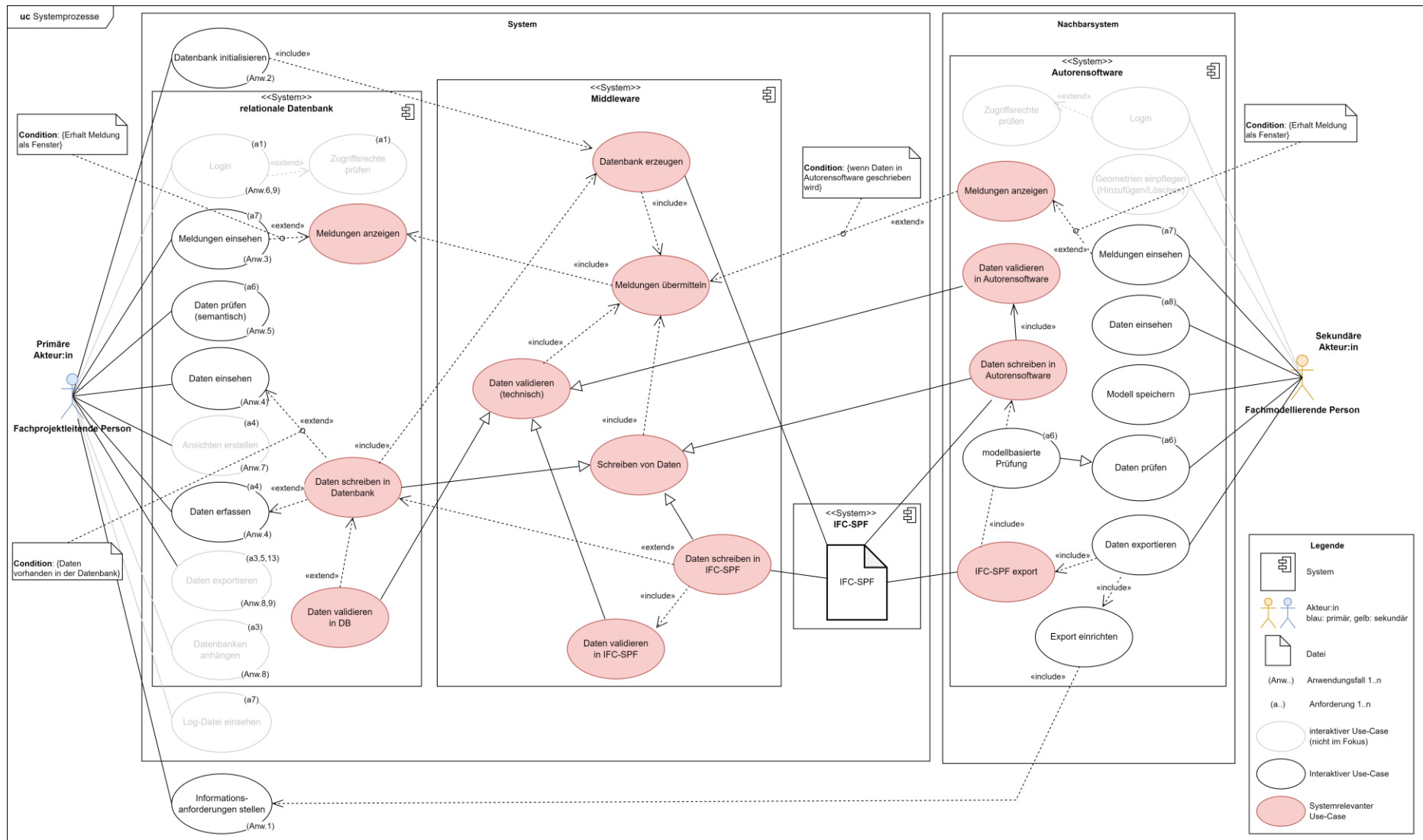


Abbildung 31: Anwendungsfalldiagramm der Systemprozesse

### 7.3 Datenkonsistenzsystem

Im Hinblick auf die Systemarchitektur und die darauf basierenden Systemprozesse ist ein Datenkonsistenzsystem zur Erhaltung der Datenkonsistenz zwischen den Systemen resp. den Bausteinen in der Systemarchitektur abzubilden. Dieses hat die Aufgabe Daten zwischen den Systemen inhaltlich auf dem gleichen Stand zu halten (Schumann-Giesler, 1997). Die Datenkonsistenz zwischen Systemen wird von verschiedenen Faktoren beeinflusst, wie in Abbildung 32 dargestellt. Es werden Bereiche in der Systemarchitektur gezeigt, in denen die Datenkonsistenz beeinträchtigt werden kann. Neben den technischen Aspekten sind auch menschliche Faktoren zu beachten, da Benutzende Fehler bei der Dateneingabe machen können. Zu den technischen Faktoren gehören die Wahl der Mapping-Methode des IFC-Datenmodells für die Datenbank, das Datenbankdesign sowie die Einschränkungen wie Primär- und Fremdschlüssel und das Mapping der Datentypen. Ebenso spielen Transaktionen von Datensätzen, der Zugriff auf Dateien, Datenbank und die Autorensoftware eine Rolle.

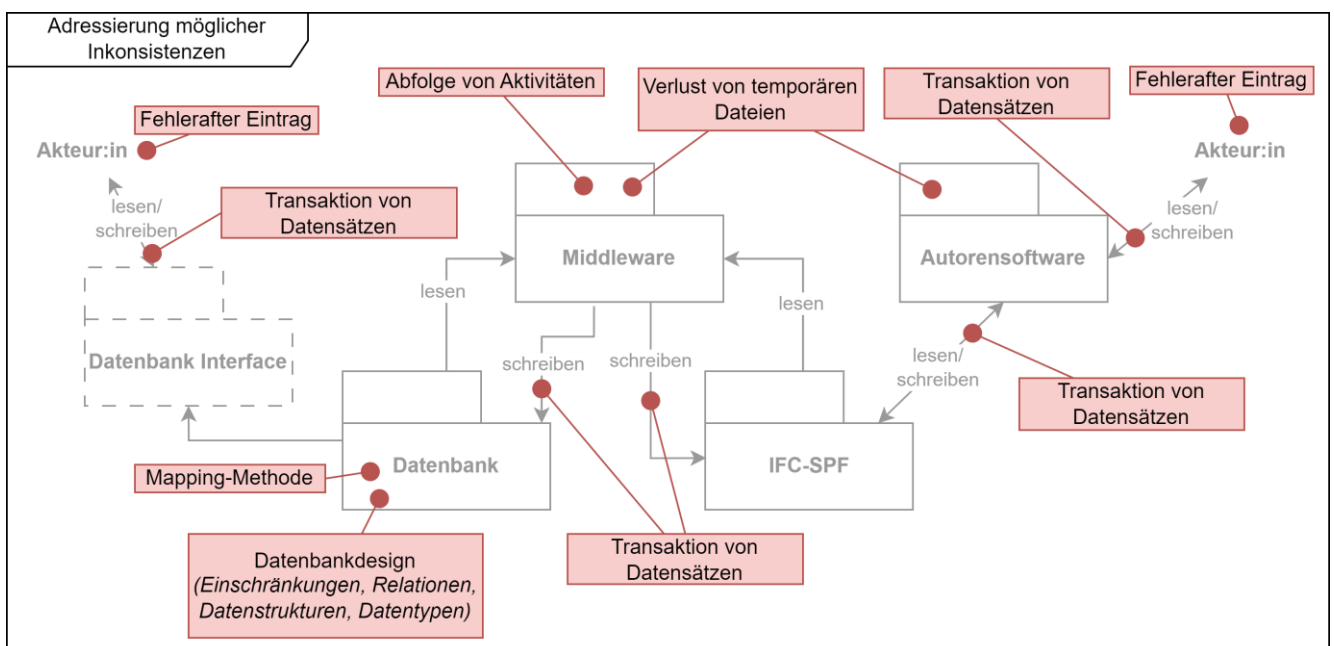


Abbildung 32: Adressierung möglicher Inkonsistenzen

Die datenkonsistenzsichernde Aufgabe gliedert sich in zwei Teilbereiche: die physische und logische Datenkonsistenz (Jablonski & Ruf, 1991). Die physische Datenkonsistenz wird von der Datenbanktechnologie und dem Rechnersystem gewährleistet, daher liegt die Verantwortung zur logischen Datenkonsistenz bei der Systemarchitektur.

Bei der logischen Datenkonsistenz werden systemseitige und anwendungsorientierte Integritätsbedingungen unterschieden. Die systemseitigen Integritätsbedingungen werden beispielsweise durch Primär- und Fremdschlüssel in der Datenbank festgelegt und sind im Datenmodell verankert. Die Transaktionen innerhalb der Datenbank werden dem zugrundeliegenden Datenbanksystem wie auch dem Autorensoftware überlassen, dass die ACID-Eigenschaften umfasst. Die anwendungsorientierte Integritätsbedingungen hingegen ergeben sich aus den spezifischen Anforderungen einer bestimmten Anwendung. Hauptsächlich ist bei einem Mehrbenutzerbetrieb sicherzustellen, dass konkurrierende Zugriffe voneinander isoliert ablaufen. Hierfür nennen (Jablonski & Ruf, 1991), dass die entsprechenden Verriegelungen (Schreib- und Lesesperren) erforderlich sind, um die Transaktionen in eine logisch serielle Ausführungsreihenfolge zu bringen. In den darauffolgenden Kapiteln werden die implementierbaren Mechanismen zur Bewältigung von Dateninkonsistenzen beschrieben.

### 7.3.1 Kontrolle über den Datenfluss

Die Steuerung des Datenflusses dient als Mechanismus, um die Flussrichtung der Daten zu bestimmen und damit die Sicherstellung der Datenkonsistenz zu unterstützen. Die Abbildung 33 beschreibt die gefassten Mechanismen konzeptuell zusammen.

Eine formale Vereinbarung bestimmt die Zusammenarbeit zwischen den beiden Akteuren/Akteurinnen. Es wird klargestellt, welche Partei welche Informationen in das DBM überträgt. Die alphanumerischen Informationen sind über die Datenbank an das DBM zu übertragen, während die geometrischen Informationen über die Autorensoftware übertragen werden.

Ein technischer Mechanismus ordnet die Aktivitäten entsprechend den Datenflussrichtungen. Dadurch wird sichergestellt, dass die Daten konsistent zwischen den Systemen sind. Die Ausrichtung der Daten erfolgt durch definierte Operationen in den jeweiligen Aktivitäten, die sich auf ein bestimmtes System beziehen. Die Aktivität «IFC-SPF Export» initiiert den Prozess aus der Perspektive der Autorensoftware heraus und die Aktivität «Datenbank initialisieren» leitet die Daten in Richtung der Datenbank weiter. Durch die Aktivität «Daten in DB erfassen» wird die Datenrichtung umgekehrt, wobei die Daten zur IFC-SPF fließen, gefolgt von der Aktivität «Daten in die Autorensoftware importieren», die die Daten zurückbezieht. Die Aktivitäten «Datenbank initialisieren» und «Daten in die Autorensoftware importieren» werden von komplementären Aktivitäten angestossen, wodurch zuerst Daten von der IFC-SPF bezogen und dann wieder zurückgegeben werden.

Eingesetzt werden Zwischenspeicher innerhalb der Aktivitäten bei kritischen Operationen wie z.B. beim Speichern von Datensätzen oder der systemübergreifenden Datenmanipulationen. Bei Fehlfunktionen können Aktivitäten dadurch rückgängig gemacht werden.

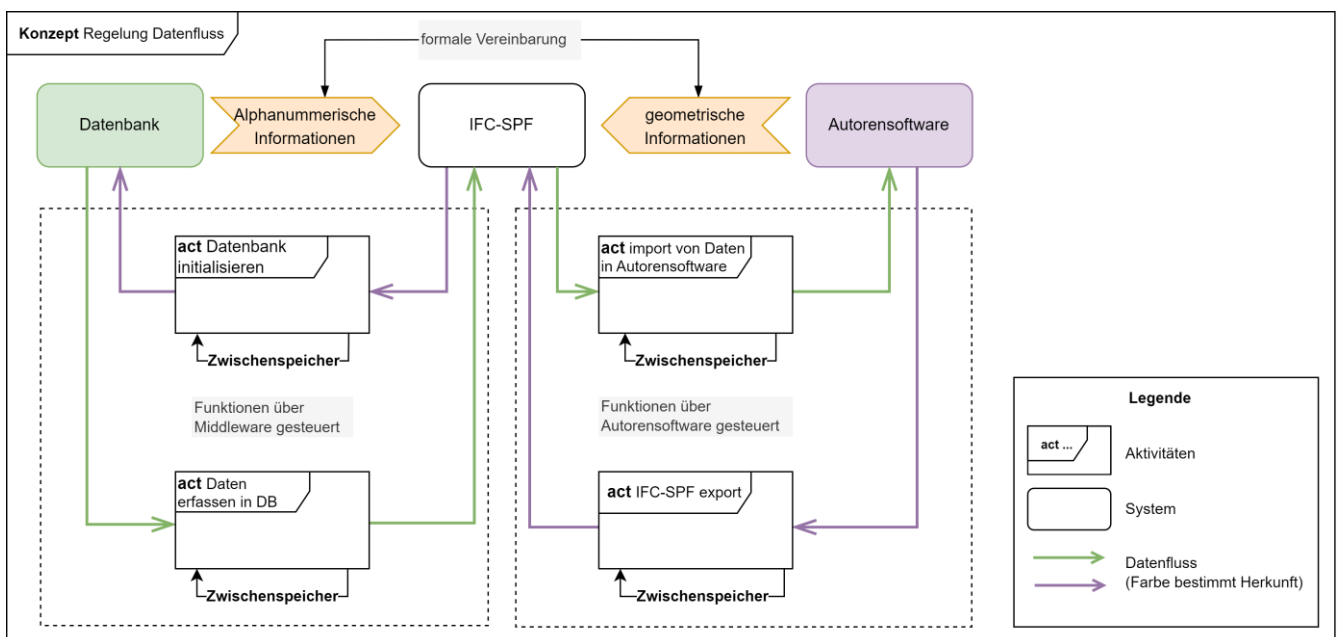


Abbildung 33: Konzept zur Regelung des Datenflusses

### 7.3.2 Regelung der Datenhoheit

Bei der Gewährleistung der Datenkonsistenz in der Systemarchitektur werden Regelungen für die Datenhoheit auf das gemeinsam genutzten Datenressource bestimmt, welche als Schreib- und Lesesperren fungieren. Das IFC-SPF dient dabei als gemeinsam genutzte Datenressource. Die Regelung erfordert die Definition, welche Parteien befugt sind, welche Arten von Datenmanipulationen durchzuführen sind und die Festlegung der Prozesse für diese Manipulationen. Die möglichen Datenmanipulationen werden anhand des SQL-Sprachelements «Data Manipulation Language (DML)» untersucht. Die Autorensoftware darf in diesem Fall nur Datensätze hinzufügen und löschen. Dies geschieht durch das Hinzufügen und Löschen von Geometriemodellen in der Autorensoftware. Die Datenbank wiederum darf nur bestehende Datensätze ändern. In Abbildung 34 werden zwei potenzielle Ansätze für die Regelung der Datenhoheit einer Datei dargestellt.

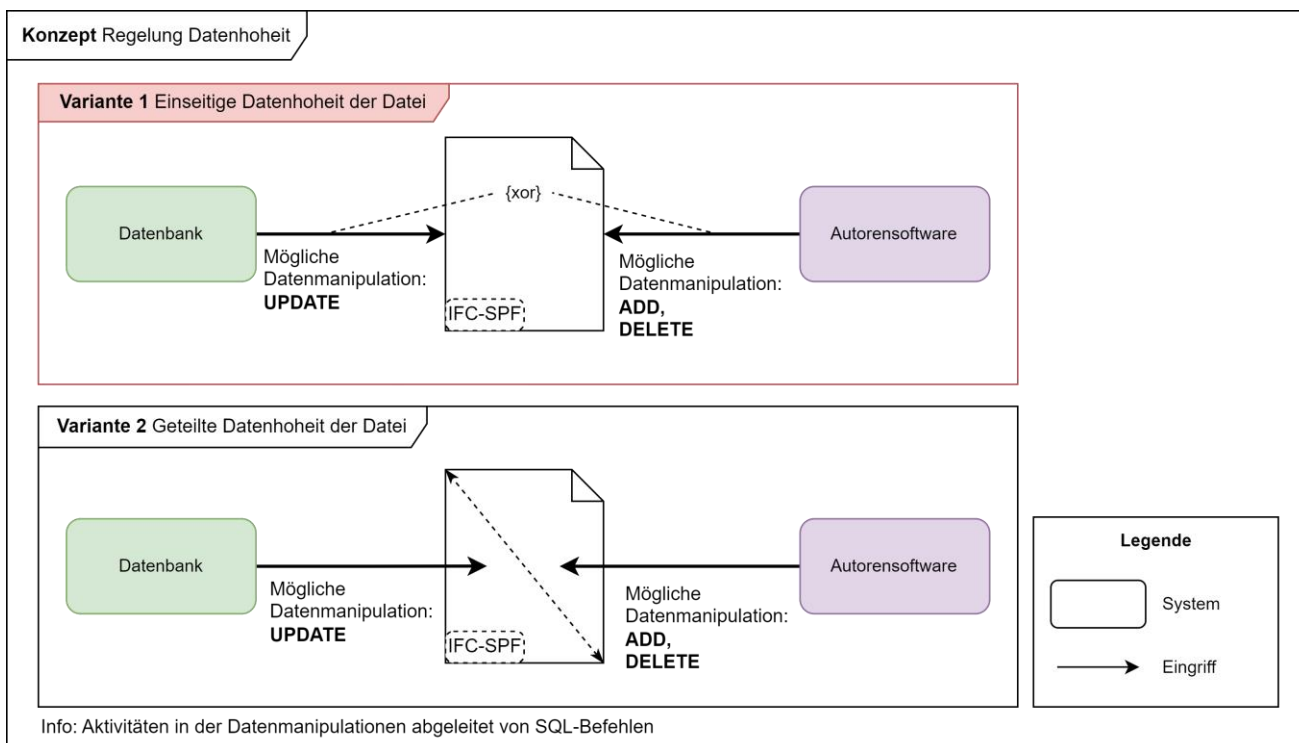


Abbildung 34: Konzepte zur Regelung der Datenhoheiten

Bei der Variante 1 hat eine Partei die alleinige Kontrolle über die Datei und kann diese nach Belieben bearbeiten. Dies bedeutet, dass zum Zeitpunkt der Datenmanipulation nur diese Partei die Autorität über die Datei ausübt und damit auch den Zugriff für die andere Partei sperrt.

Bei der Variante 2 teilen sich beide Parteien die Kontrolle über der Datei, wobei jede Partei unterschiedliche Berechtigungen und Autoritäten besitzt. Während eine Partei möglicherweise nur die Befugnis hat, Daten anzupassen, kann die andere Partei die Möglichkeit haben, Datensätze zu löschen oder hinzuzufügen. Dies schafft eine dynamische Umgebung, in der beide Parteien eine gewisse Kontrolle ausüben können, jedoch in unterschiedlichem Umfang und mit unterschiedlichen Befugnissen.

Die Variante 1 bietet gegenüber der Variante 2 eine einfachere Implementierung für die Gewährleistung der Datenkonsistenz, weshalb im Rahmen dieser Arbeit die Implementierung des Proof of Concepts die Variante 1 verwendet wird.

### 7.3.3 Protokollierung der Operationen

In Datenbanksystemen werden sämtliche Datenbankzugriffe und -änderungen protokolliert, um im Falle eines Fehlers die offenen Transaktionen zurückzusetzen und bereits abgeschlossene Transaktionen wiederherzustellen (Schicker, 2017). Die Protokollierung von Änderungen wird auch von (Schumann-Giesler, 1997) erwähnt und ist auch als Anforderung «a7» aus dem Kapitel 4.7 deklariert.

In den jeweiligen Aktivitäten werden die ausgeführten Operationen in Logdatei gesichert. Die Logdatei befindet sich im Dateisystem der Systemarchitektur. Mit der Logdatei können im Fehlerfall alle Änderungen rückgängig gemacht werden. Die Abbildung 35 beschreibt konzeptuell den Datenfluss zwischen den Aktivitäten und der Logdatei. Jede Aktivität erhält Zugriff auf die Logdatei. Der Zugriff wird anhand der Regelung der Datenhoheiten (Kapitel 7.3.2) und dem kontrollierten Ablauf der Aktivitäten (Kapitel 7.3.1) bestimmt.

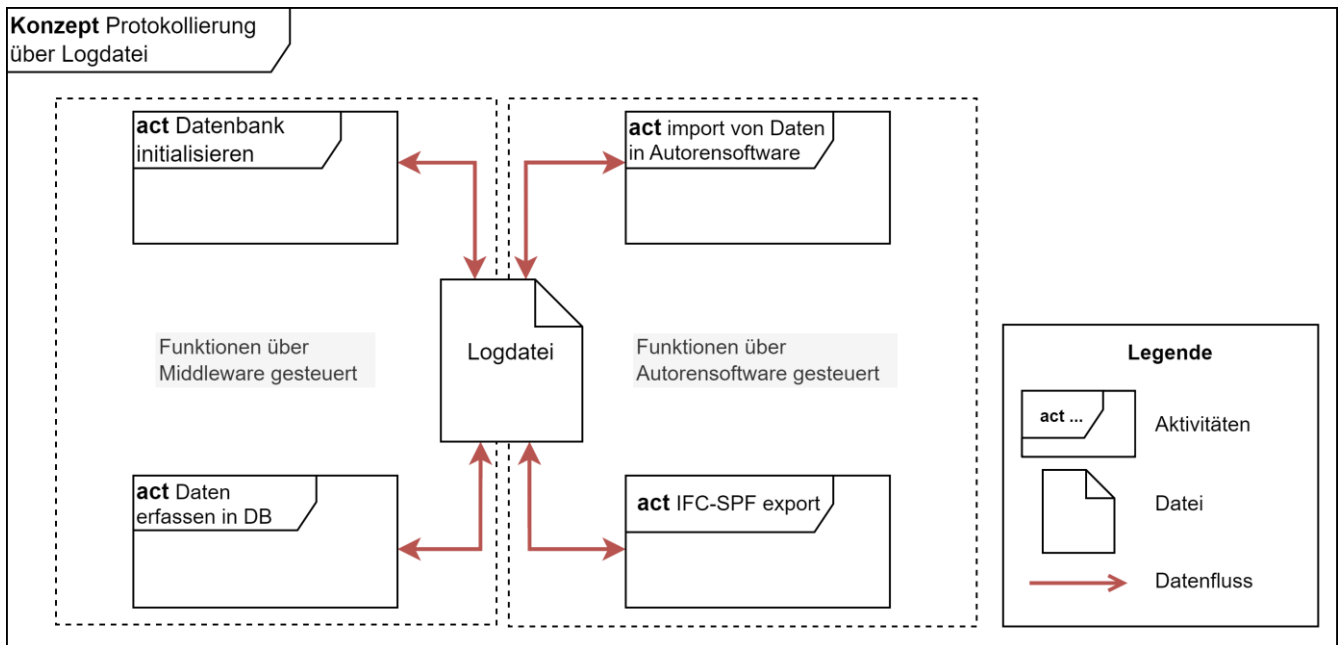


Abbildung 35: Protokollierung der Operationen in einer Logdatei

## 7.4 Systembaustein IFC-SPF

Das IFC-SPF fungiert als zentrales Element zur Speicherung sämtlicher Daten aus der Datenbank und aus der Autorensoftware im IFC-Datenmodell. Es bietet gemäss den Anforderungen der KMUs ein systemunabhängiges Speichermedium des DBMs, das als verlustfreier Informationsaustausch zwischen verschiedenen Domänen dient.

### 7.4.1 Mapping der IFC-Konzepte

Die vorgebrachten fachlichen Beschreibungen des DBMs der KMUs aus dem Kapitel 4.5 weisen eine Ähnlichkeit mit den entsprechenden IFC-Konzepten aus der Dokumentation von BuildingSMART International (BuildingSMART International, 2020). Trotz des Mangels an detaillierten Informationen über die technische Umsetzung der KMUs, wird ein mögliches Mapping zu den IFC-Konzepten hergeleitet. In der Abbildung 36 wird das Mapping aufgezeigt. Die Eigenschaften der Bauteile werden den «Property Sets for Objects» zugeordnet, während die Mengenangaben und Stückzahlen sowohl durch das Zählen von Datensätzen als auch durch die Verwendung des IFC-Konzeptes «Quantity Sets» erreicht werden können. Die Klassifizierung der Bauteile erfolgt über die «Classification Association», die Materialisierung über die «Material Association» oder über den «Element Covering». Letzteres wird nicht weiter behandelt, um die Anzahl der IFC-Konzepte überschaubar zu halten. Die Typisierung erfolgt über «Object Typing», wobei das Attribut «PredefinedType» bei den Bauteilen auf «USERDEFINED» gesetzt oder über «Object Predefined Type» definiert wird. Letzteres ist eine Enumeration mit verschiedenen vordefinierten Werten. Als Ausnahme wird das IFC-Konzept «Group Assignment» in der Fachmodellierung von gebäudetechnischen Einrichtungen beim Praxispartner als relevant eingestuft. Daher wird dieses Konzept für die Untersuchung ergänzt.

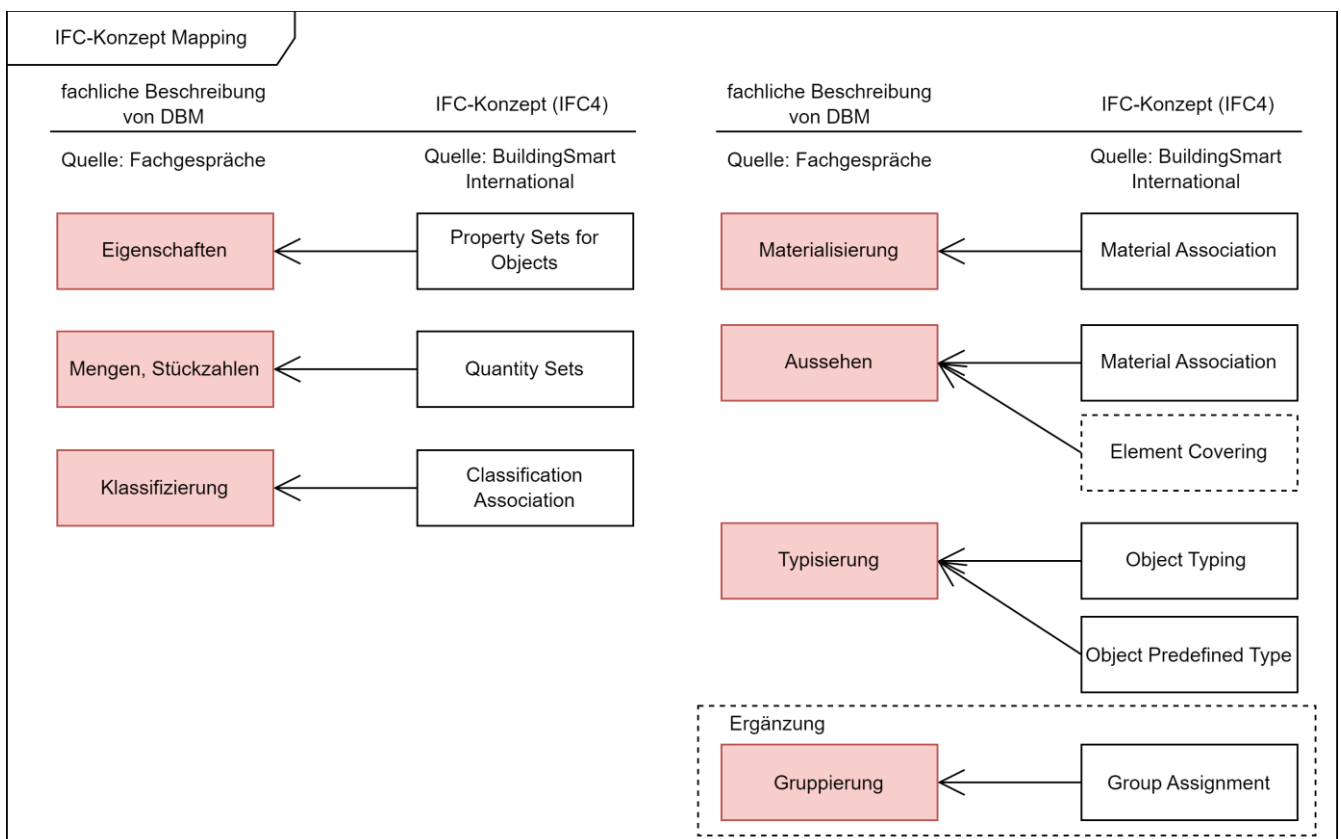


Abbildung 36: Mapping IFC-Konzept mit den fachlichen Beschreibungen der DBM aus den Fachgesprächen

Die Zusammenhänge der relevanten IFC-Konzepte sind in der Abbildung 37 veranschaulicht. Es ist zu beachten, dass es sich um eine Pseudodarstellung handelt. Es soll die Herkunft und Beziehungen der relevanten Klassen aus den IFC-Konzepten als Gesamtbild darstellen. Die Darstellung zielt darauf ab, überblickend die fachliche Sicht der berücksichtigten IFC-Konzepte zu veranschaulichen. Dabei sind die Klassen «Produkte» «Produkttypen» die zentralen Klassen, welche mit den restlichen Klassen semantisch beschrieben werden. Die Herkunft der Klassen sind einerseits die «Eigenschaftendefinition», die «Objektdefinition» und die «Ressourcen».

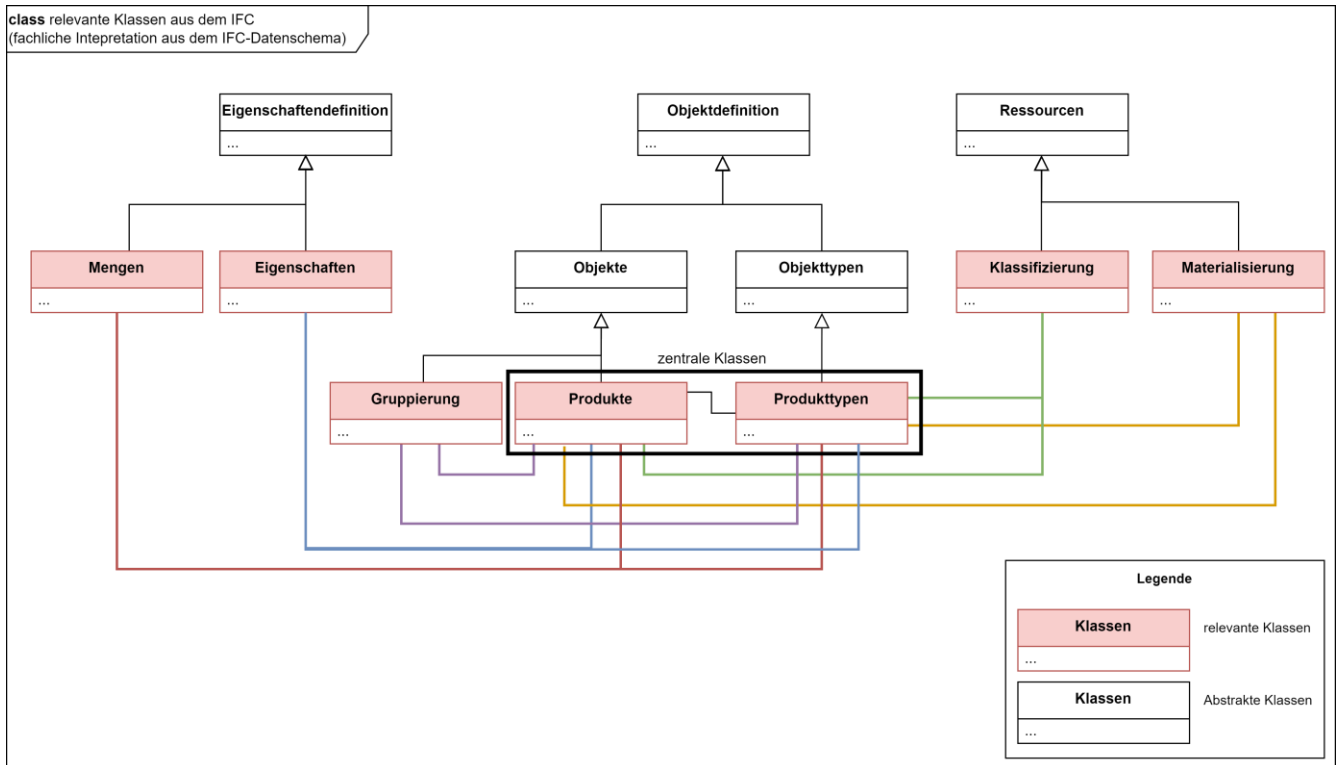


Abbildung 37: Relevante Klassen aus den IFC-Konzepten als Pseudodarstellung

### 7.4.2 IFC-Konzepte im IFC-Datenmodell

Die IFC-Konzepte nach dem IFC-Datenmodell werden in der Abbildung 38 als Gesamtbild anhand von einem Beispiel mit den IFC-Entitäten «IfcValve» und «IfcValveType» dargestellt. Dabei wird die Abbildung 37 als Orientierung verwendet. Im Gesamtbild lässt sich auch erkennen, wo sich die relevanten Daten in einem IFC-Datenmodell befinden. Die grünen Markierungen in der Abbildung verdeutlichen die relevanten Attribute für die Datenverwaltung in der Datenbank. Es wird ersichtlich, dass diese relevanten Attribute über verschiedene Entitäten verteilt sind. Für die Generierung von Informationen sind mehrere Verbindungen zwischen den Objekten «IfcValve» und «IfcValveType» zu erstellen. Dabei sind nicht nur End-Entitäten von Bedeutung, sondern auch Eltern-Entitäten, da relevante Daten auch dort zu finden sind, wie beim Beispiel der Entität «IfcGroup».

Das Gesamtbild bildet die konzeptuelle Grundlage für die Erstellung des Datenbankmodells, welche auf dem IFC-Datenmodell basieren soll. Die Kardinalitäten geben zudem Aufschluss darüber, wie die Beziehungen konstruiert werden sollen.

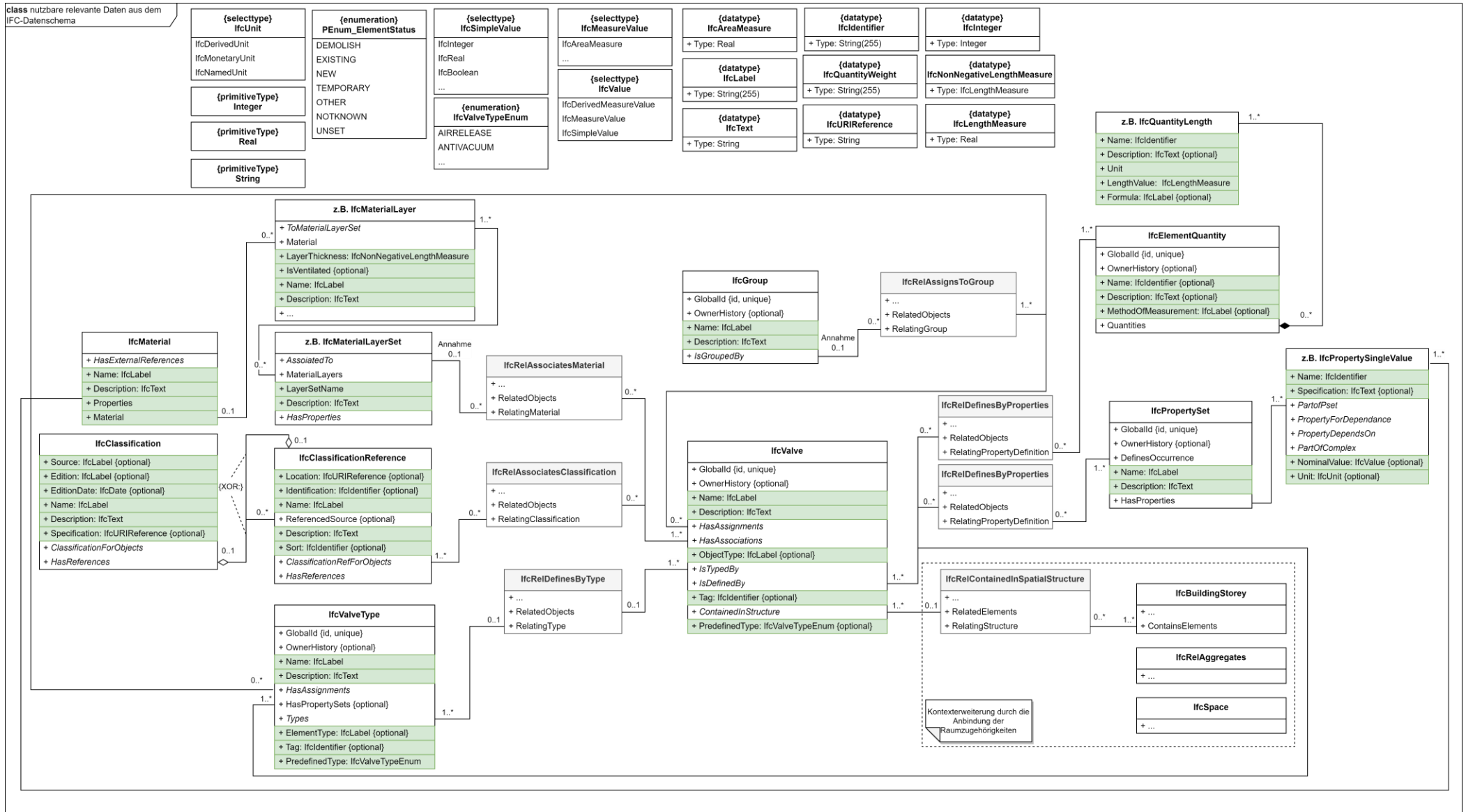


Abbildung 38: IFC-Datenschema mit den relevanten IFC-Klassen am Beispiel von «IfcValve» und «IfcValveType» als Objekte

## 7.5 Systembaustein Datenbank

Die Datenbank fungiert als Fundament zur Anreicherung des DBMs mit alphanumerischen Informationen, indem ausschliesslich Daten aus dem IFC-SPF übernommen werden. Auf dieser Grundlage ist es möglich, eine GUI zu implementieren, um die Datenbank zu nutzen. Die Datenbank dient dem primäre:n Akteur:in als Werkzeug. In diesem Abschnitt wird das Datenbankschema behandelt, beginnend mit den Mapping-Methoden, gefolgt von der Definition der Primärschlüssel und der Beziehungen zwischen den Tabellen. Abschliessend werden die Datenbankschematas auf der Grundlage der IFC-Konzepte beschrieben.

### 7.5.1 Mapping Methode

Bereits im Jahr 2004 beschäftigte sich die Studie mit der Implementierung von STEP basierten Produktmodellen in Datenbankmodellen zu beschreiben (You & Yang, 2004). In der Untersuchung von (Zhu & Shide, 2021) wird die Thematik wieder aufgenommen und deklariert, dass beim Mapping des IFC-Datenmodells in Datenbanken in zwei Kategorien unterteilt werden können: «Entitäten-zu-Tabellen» und «Attribute-zu-Tabellen». Weiter wurden in der Untersuchung vier Mapping Konzepte untereinander verglichen, welches in der Abbildung 39 dargestellt ist. Als Beispiel wird der Teilbereich «Prozesse» aus dem IFC-Datenmodell verwendet.

In der Methode P1 werden Daten von Entitäten mit Vererbungsbeziehungen in einer einzigen Tabelle zusammengeführt. Dadurch werden Abfragen beschleunigt, wobei NULL-Werte resp. leere Inhalte die Datenkonsistenz beeinträchtigen können. P2 erstellt Tabellen für Entitäten unabhängig von ihrer Abstraktheit und erlaubt das Hinzufügen von «NOT NULL-Constraints», was die Datenintegrität verbessert, jedoch die Komplexität erhöhen kann. P3 reduziert die Tabellenanzahl, indem abstrakte Entitäten nicht in separate Tabellen erzeugt werden. Zwar werden Datenabfragen erleichtert, aber aufgrund der mehrfachen Daten aus den abstrakten Entitäten in den Datensätzen Dateninkonsistenzen entstehen können. P4 unterscheidet sich grundlegend gegenüber den anderen Methoden. Es trennt Entitätsinstanzen und Attribute in Tabellen basierend auf ihren Basistypen. Dadurch wird auf der einen Seite das Datenmanagement erleichtert, auf der anderen Seite wird die Effizienz bei Datenabfragen beeinträchtigt (Zhu & Shide, 2021).

Die verschiedenen Mapping-Methoden richten sich an unterschiedliche Anforderungen, wobei die Herausforderung darin besteht, ein Gleichgewicht zwischen der Zugriffseffizienz und der Datenkonsistenz zu finden (Zhu & Shide, 2021). P1 ist die einfachste Implementierung, trägt jedoch ein höheres Risiko für Dateninkonsistenzen bei. P2 verbessert die Datenkonsistenz von P1, erschwert jedoch die Nachvollziehbarkeit bei der Entwicklung von GUIs und Anwendungen durch JOIN-SQL-Befehle. P3 erleichtert GUI-Entwicklung, schwächt jedoch die Datenintegrität durch mehrfache Abbildung von abstrakten Entitäten in Datenbanktabellen. P4 deckt die Herausforderungen von P3 ab, erhöht jedoch die JOIN-SQL-Befehlskomplexität, was die Implementierung von Anwendungen erschwert.

Der Mittelweg ist die Implementierung der Mapping Methode P3. Die Mapping Methode P3 beinhaltet teilweise die Vorteile von P2 und P4. Einerseits wird die Entwicklung von GUIs und dabei die Anwendung erleichtert, andererseits wird dadurch eine zufriedenstellende Datenkonsistenz erreicht. Die Schwächung der Datenintegrität der Methode P3 kann für den Anwendungsfall der Arbeit als vernachlässigbar betrachtet werden. Ein Beispiel der Implementierung der P3-Methode ist die Arbeit von Dion Moul, welche die Transformation von IFC-SPF zu SQLite / MySQL untersuchte (Dion Moul, 2023). Die Entscheidung für die Methode P3 und die Behauptung der geschwächten Datenintegrität ist in der praktischen Anwendung zu überprüfen.



Abbildung 39: Tabellenstrukturen von verschiedenen Mapping-Konzepten, Quelle: (Zhu & Shide, 2021)

## 7.5.2 Primärschlüssel

Ein Primärschlüssel in einer Datenbank wird verwendet, um einen Datensatz eindeutig zu identifizieren. Die Identifizierung von IFC-Objekten erfolgt üblicherweise über die Attribute «GlobalId» und «Name». Nicht jedes Objekt ist jedoch direkt dem Hauptobjekt «IfcRoot» untergeordnet, um die «GlobalId» automatisch zu erben. Ist das Attribut «GlobalId» vorhanden, wird dieses als Primärschlüssel verwendet. Hingegen wird das Attribut «Name» als eine Alternative für die Identifizierung betrachtet. Wobei die Eindeutigkeit entweder durch die Autorensoftware oder durch den Benutzer sichergestellt werden muss. Mehrere Instanzen von IFC-Objekten können denselben Namen besitzen. Das Attribut «Name» ist optional und muss nicht immer einen Wert enthalten. Alternativ kann im IFC-SPF die generierte SPF-ID verwendet werden. Diese ID ist jedoch nur lokal gültig und ändert sich jedes Mal, wenn das SPF neu exportiert wird. Die SPF-ID dient der lokalen Referenz innerhalb der Transferdatei zwischen den IFC-Objekten. Sie wird durch das Zeichen «#», gefolgt von einer positiven Zahl dargestellt ("ISO 10303-21," 2023, pp. 10303–21). Wenn die SPF-ID als Primärschlüssel verwendet wird, muss das Datenbankschema nach jedem Export des IFC-SPF aktualisiert werden.

Um nicht von den Nutzenden sowie der Autorensoftware die Qualität des Primärschlüssels anhand des Attributs «Name» bestimmen zu lassen, wird in diesem Fall die SPF-ID verwendet. Folgende Bedingungen treten bei der Bestimmung des Primärschlüssels auf:

Tabelle 5: Bedingungen zur Bestimmung des Primärschlüssels

Bedingung	Primärschlüssel	Einschränkung in Datenbank
Wenn IFC-Objekt ein GlobalId besitzt	GlobalId	Datentyp Text/String und leere Felder sind nicht erlaubt
Wenn IFC-Objekt kein GlobalId besitzt	SPF-ID	Datentyp Integer und leere Felder sind nicht erlaubt

## 7.5.3 Relationen

Die Relationen werden gemäss der Definition des IFC-Datenmodells der Version 4.0 im Beispiel vom IFC-Konzept «Eigenschaften für Objekte» folgendermassen beschrieben (BuildingSMART International, 2020):

*«The IfcRelDefinesByProperties is a 1-to-N relationship, as it allows for the assignment of one property set to a single or to many objects. Those objects then share the same property definition.»*

Diese Beschreibung von BuildingSMART International kann in zwei Varianten interpretiert werden. In der ersten Variante wird ein «IfcRelDefinesbyProperties» verwendet, um mehrere IFC-Objekte mit derselben «IfcPropertySet» zu verbinden. In der zweiten Variante werden für dieselbe Beschreibung der Relation mehrere «IfcRelDefinesByProperties» erzeugt. Die Varianten sind in der Abbildung 40 dargestellt.

Die technische Implementierung kann je nach Autorensoftware variieren. Wenn «IfcRelDefinesByProperties» als Relationstabelle im Datenbankmodell verwendet wird, werden in der ersten Variante mehrere Tabellen mit «IfcPropertySet» als Many-to-Many-Relation erstellt, wobei mehrere Spalten als Fremdschlüsselbeziehungen dienen. Dabei muss dieses Konzept NULL-Werte in den Spalten zulassen. Dies führt aber zu weniger JOIN-SQL-Befehlen bei der Entwicklung einer GUI. Bei der Umsetzung der zweiten Variante entstehen keine NULL-Werte in den Relationen zwischen den Tabellen, was zu einem Datenbankmodell mit besserer Datenqualität führt. Dabei ist ein anderes

Benennungskonzept der Tabellen anzuwenden, da z.B. unter SQLite Tabellen mit derselben Bezeichnung nicht erlaubt sind. Für dieses Vorhaben wurde die zweite Variante gewählt, um eine bessere Datenqualität zu gewährleisten.

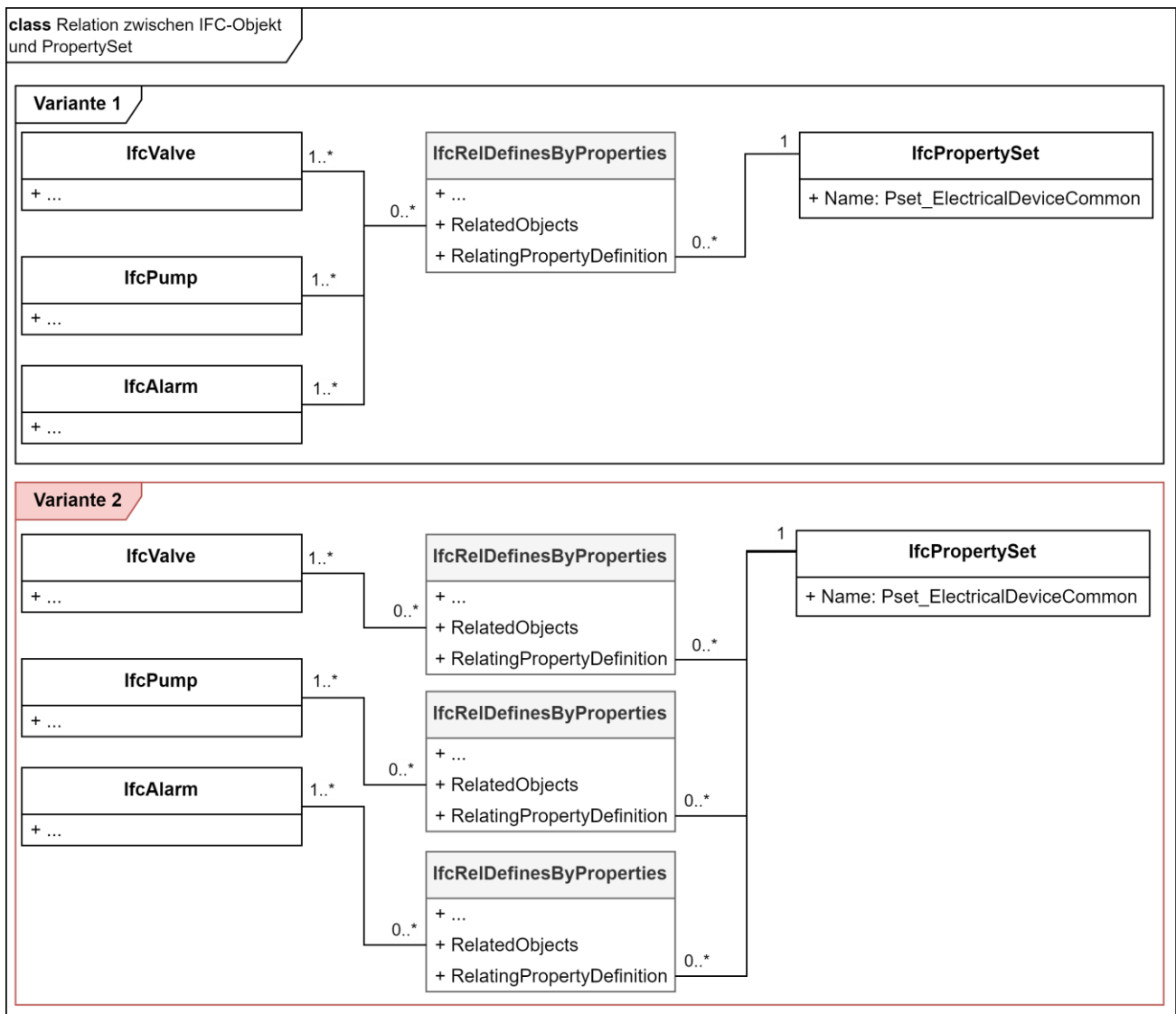


Abbildung 40: Relation zwischen IFC-Objekt und PropertySet

### 7.5.4 Datenbankschemas

Basierend auf der Mapping Methode P3 werden die Datenbankschematas für die jeweiligen IFC-Konzepte erstellt. Dabei wird die Modellierungsmethode Entity-Relation-Diagramm, welches auf der Chen Notation basiert, angewendet. Folgende Abbildungen stellen die IFC-Konzepte im Datenbankschema dar:

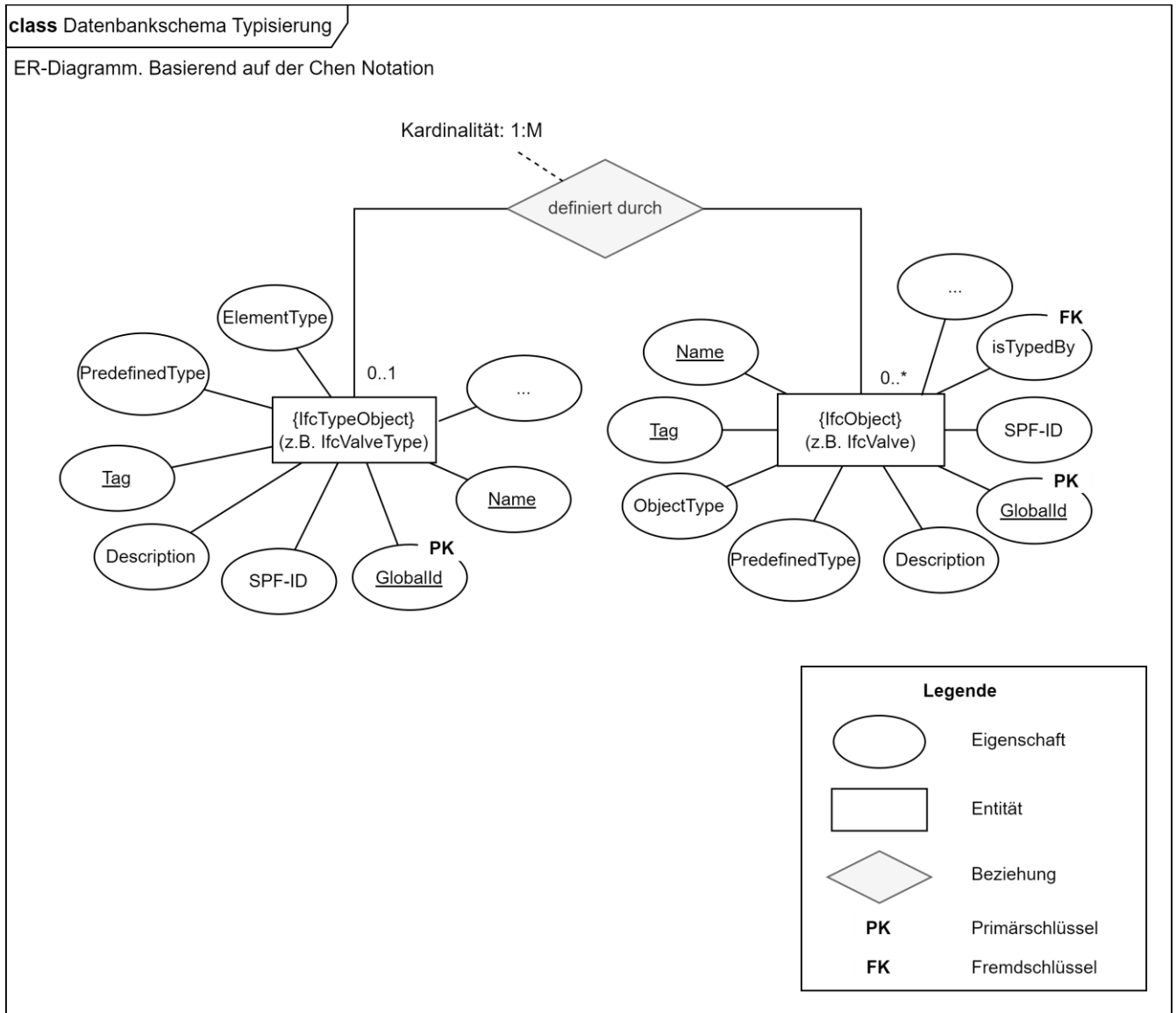


Abbildung 41: Datenbankschema «Typisierung»

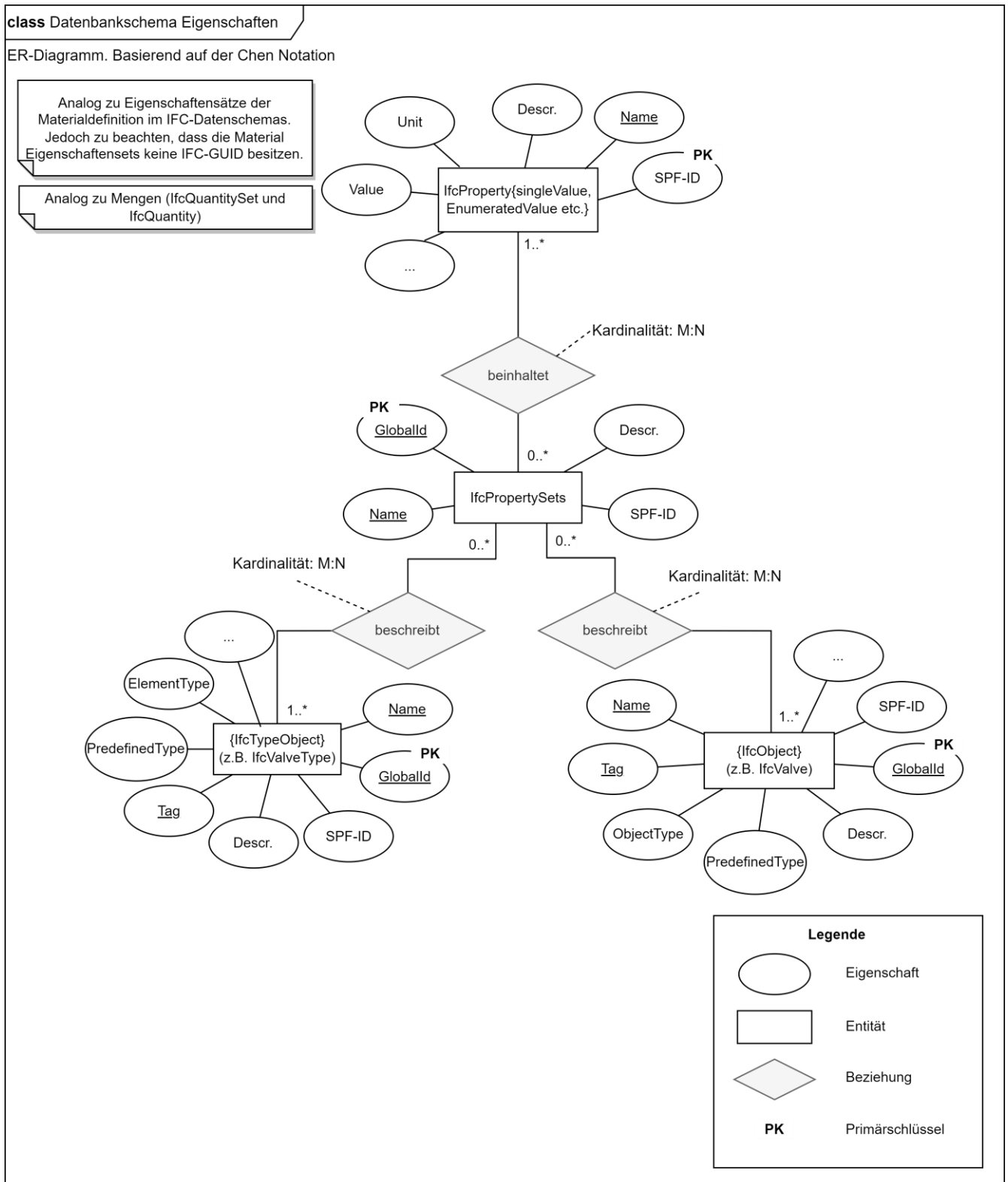


Abbildung 42: Datenbankschema «Eigenschaften» und «Mengen»

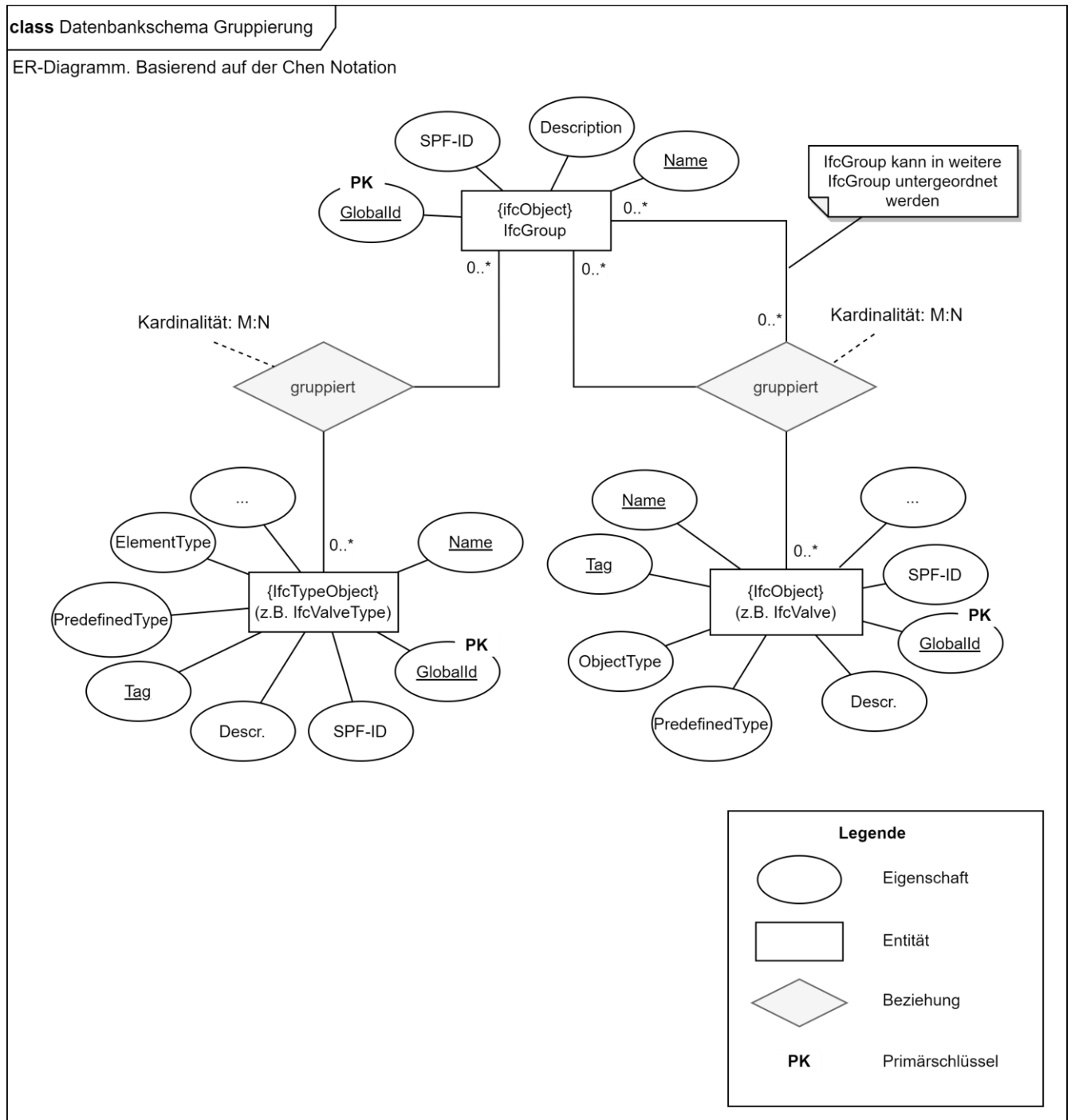


Abbildung 43: Datenbankschema «Gruppierung»

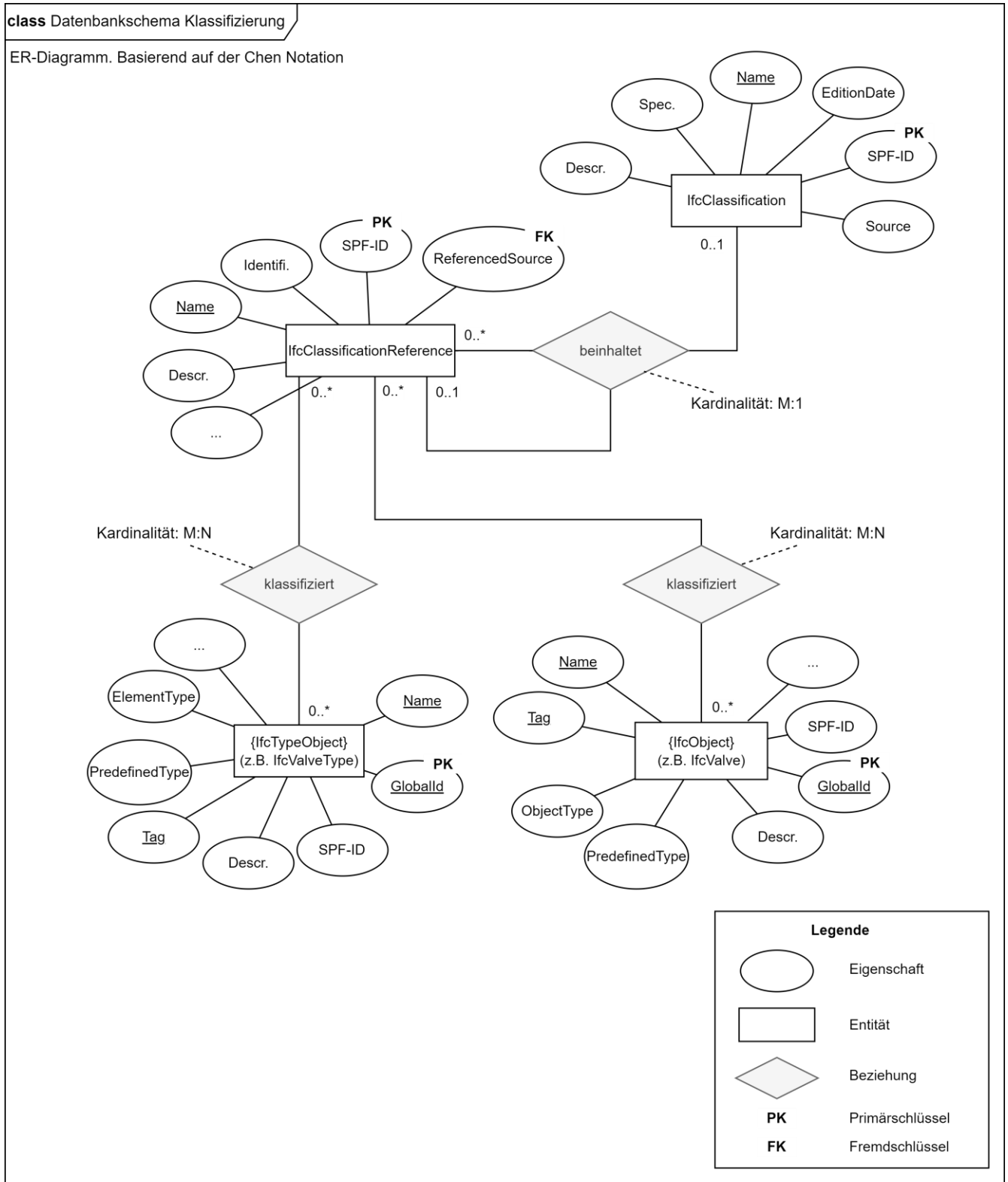


Abbildung 44: Datenbankschema «Klassifizierung»



### 7.5.5 Aktivität

Die Aktivität, die mit dem Systembaustein «Datenbank» verbunden ist, umfasst gemäss Abbildung 33 die Initialisierung der Datenbank. Dabei wird das Datenbankmodell gemäss dem IFC-SPF erstellt. Anschliessend werden die IFC-Objekte und ihre Attributdaten in die Datenbank übertragen.

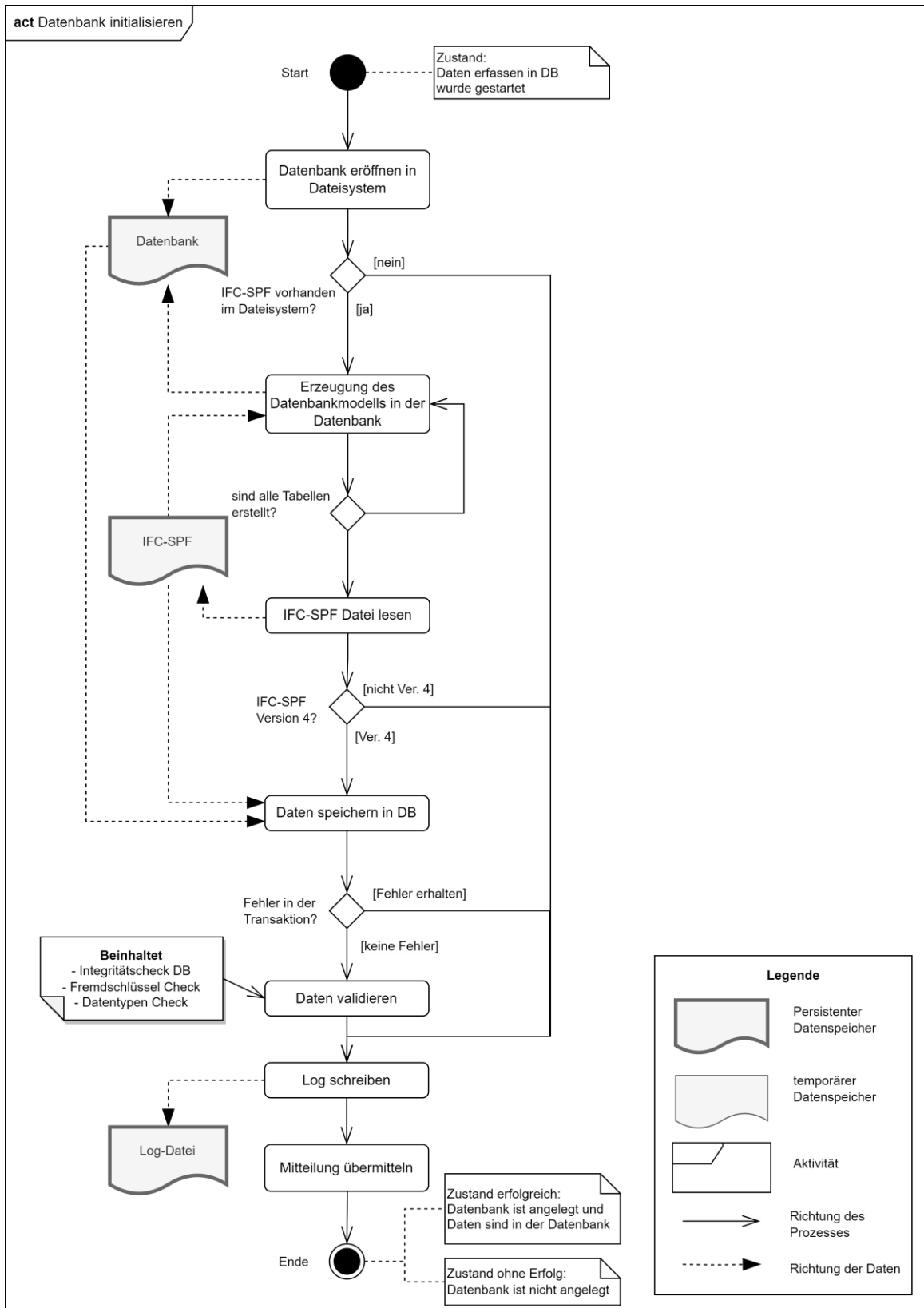


Abbildung 46: Aktivitätsdiagramm: Datenbank initialisieren

## 7.6 Systembaustein Middleware

Die Middleware fungiert als Verbindungsglied zwischen der Datenbank und dem DBM. Ihre Hauptaufgabe besteht darin, die Datenbank zu erstellen, Daten einzufügen, Änderungen an den Daten in der Datenbank zu verfolgen und diese Änderungen dann in das IFC-SPF zu schreiben.

### 7.6.1 Datenbank aufsetzen

Die Grundlage für die Initialisierung der Datenbank bildet das IFC-SPF. Beim Export des IFC-SPF aus der Autorensoftware wird das gesamte DBM in einer einzigen Datei erzeugt, nicht nur Teile davon. Wie bereits im Kapitel 7.5.2 diskutiert, werden beim erneuten Export des IFC-SPF die SPF-IDs neu generiert, und diese können zwischen den Versionen nicht mehr als eindeutige Identifikatoren verwendet werden. Um den SPF-ID als Primärschlüssel verwenden zu können, wird bei einem erneuten Export des DBMs die Datenbank gemäss der in Abbildung 46 dargestellten Aktivität erneut initialisiert. Dies hat zur Bedeutung, dass das Datenbankmodell neu aufgebaut wird und die Daten aus dem DBM in die Datenbank geschrieben werden.

### 7.6.2 Prozess der Datenverarbeitung

Nach der Initialisierung der Datenbank kann der Prozess der Datenerfassung in der Datenbank beginnen. Dabei wird ein Datenbank-Interface verwendet, um eine aktive Verbindung zur Datenbank zu starten und so dem primäre:n Akteur:in die Datenerfassung zu ermöglichen. Dabei muss das Konzept zur Regelung der Datenhoheit aus Kapitel 7.3.2 berücksichtigt und der Datenfluss gemäss Abbildung 33 eingehalten werden. Eine pragmatische Lösung besteht darin, eine temporäre Datei im Dateisystem zu erstellen. Diese Datei dient der Autorensoftware als Signal für eine aktive Verbindung zur Datenbank und verhindert die Erzeugung eines neuen DBMs. Sobald keine Nutzer:innen mehr eine aktive Verbindung zur Datenbank haben, wird die temporäre Datei entfernt. Dies signalisiert, dass die Datenhoheit über das IFC-SPF der Autorensoftware obliegt. Dieses Konzept wird in der Abbildung 47 veranschaulicht. Die temporäre Datei wird dabei verwendet, um Änderungen in der Datenbank beobachten zu können und die Änderungen entsprechend in das IFC-SPF zu schreiben.

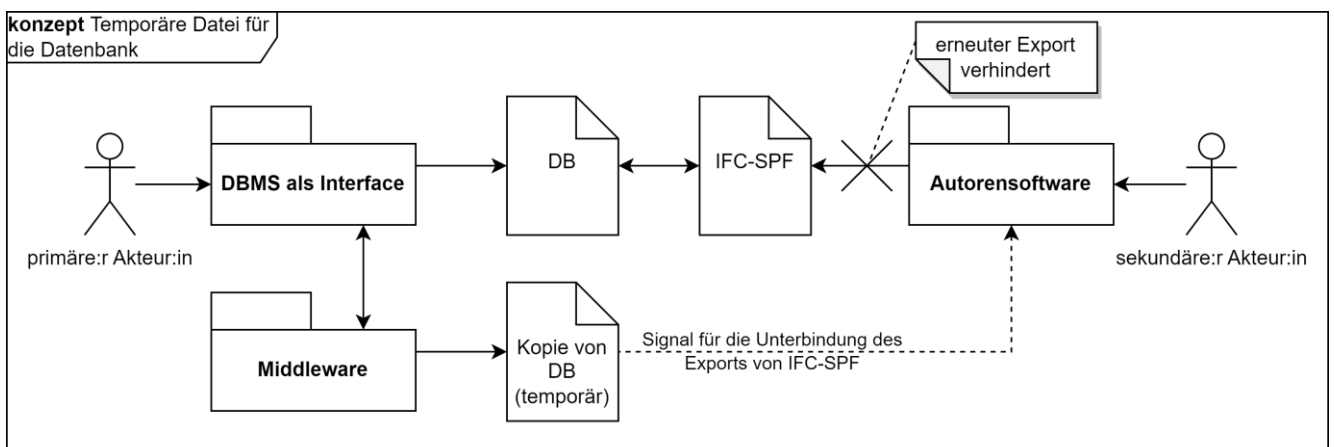


Abbildung 47: Temporäre Datei im Prozess der Datenverarbeitung in der Datenbank

### 7.6.3 Aktivität

Die Aktivität, die mit dem Systembaustein «Middleware» verbunden ist, umfasst gemäss Abbildung 33 die Erfassung von Daten in der Datenbank. Dabei wird das Datenbankmodell gemäss dem vorliegenden IFC-SPF erstellt. Anschliessend werden die IFC-Objekte und ihre Attributdaten in die Datenbank übertragen.

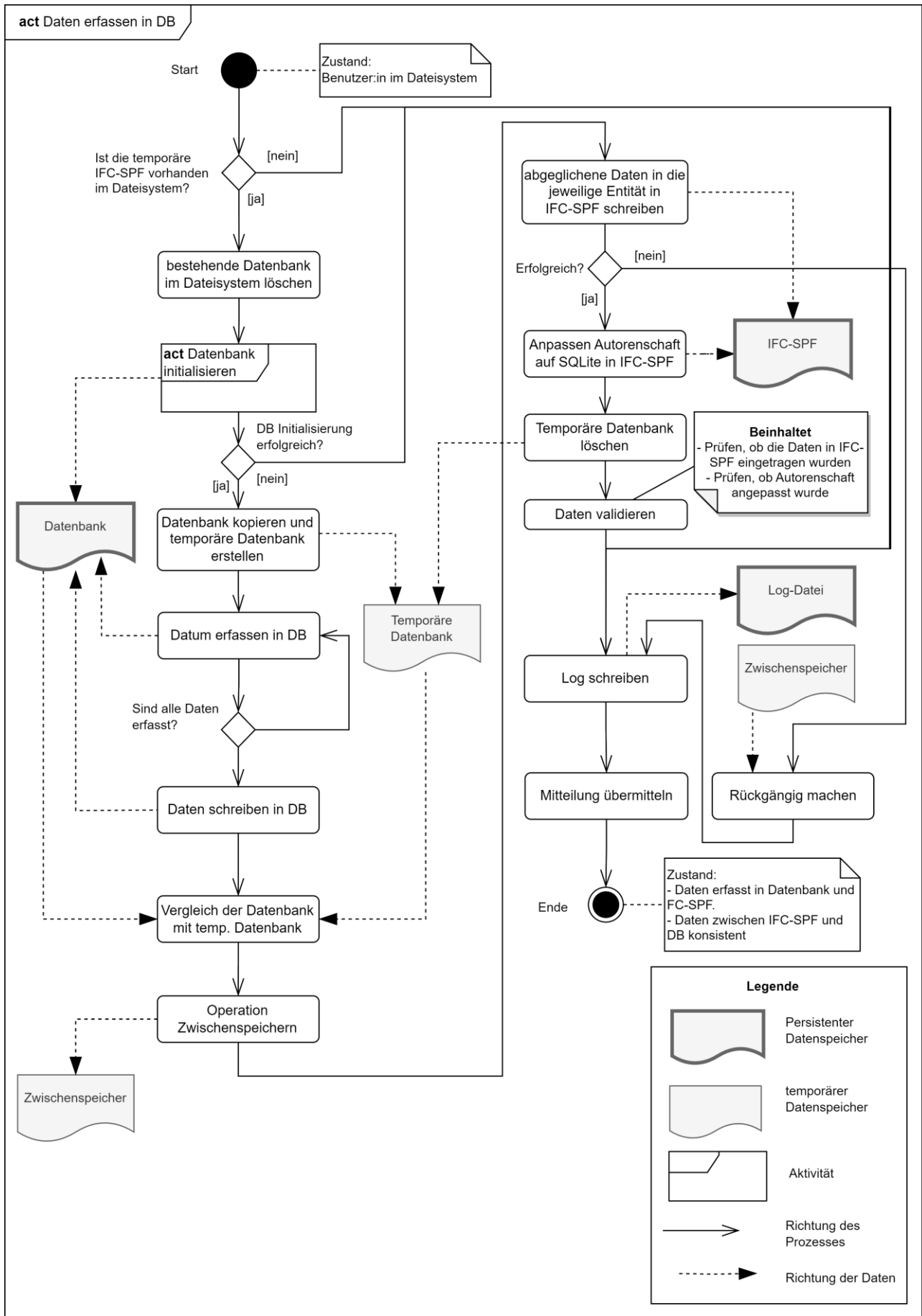


Abbildung 48: Aktivitätsdiagramm «Daten erfassen in DB»

## 7.7 Systembaustein Autorenssoftware

Das Nachbarsystem Autorenssoftware beinhaltet die Anwendersicht zur Erzeugung von geometrischen Informationen des DBMs. Es ist das Hauptwerkzeug für den/die sekundäre:n Akteur:in. In diesem Abschnitt werden die Handhabung der Informationsanforderungen in der Autorenssoftware, der Export aus der Autorenssoftware und der Import von Daten behandelt.

### 7.7.1 Informationsanforderung

Die Informationsanforderung wird für die Konfiguration der Autorenssoftware benötigt. Es liegt in der Verantwortung des primären Akteurs oder der primären Akteurin zu deklarieren, welche Parameter aus der Autorenssoftware exportiert werden sollen. Diese Parameter sind notwendig, um die erforderlichen alphanumerischen Informationen für das DBM über die Datenbank erfassen zu können. Die Einrichtung der Autorenssoftware muss vom sekundären Akteur/Akteurin vorgenommen werden. Da die Einrichtung je nach Software unterschiedlich ist und den Nutzenden bekannt sein sollte, wird hier nicht im Detail darauf eingegangen. Beinhalten sollte die Einrichtung das Erstellen von Parametern und den korrekten Export des IFC-SPFs anhand der Informationsanforderung.

### 7.7.2 Export und Import

Beim Export von IFC-SPF wird die exportierte Version dupliziert und als temporäre IFC-SPF im Dateisystem gespeichert. Die Funktion der temporären IFC-SPF liegt darin, einerseits die Initialisierung der Datenbank zu ermöglichen und andererseits den Import der Daten aus dem IFC-SPF zu erlauben. Das Importieren der Daten aus dem IFC-SPF in die Autorenssoftware erfolgt durch den Vergleich der temporären IFC-SPF, welches beim letzten Export generiert wurde und dem aktuellen IFC-SPF, das über die Datenbank aktualisiert wurde. Dabei werden die geänderten Datensätze zwischen den beiden Dateien identifiziert und in die Autorenssoftware übertragen. Liegt die Kopie der IFC-SPF nicht vor, kann kein Import erfolgen.

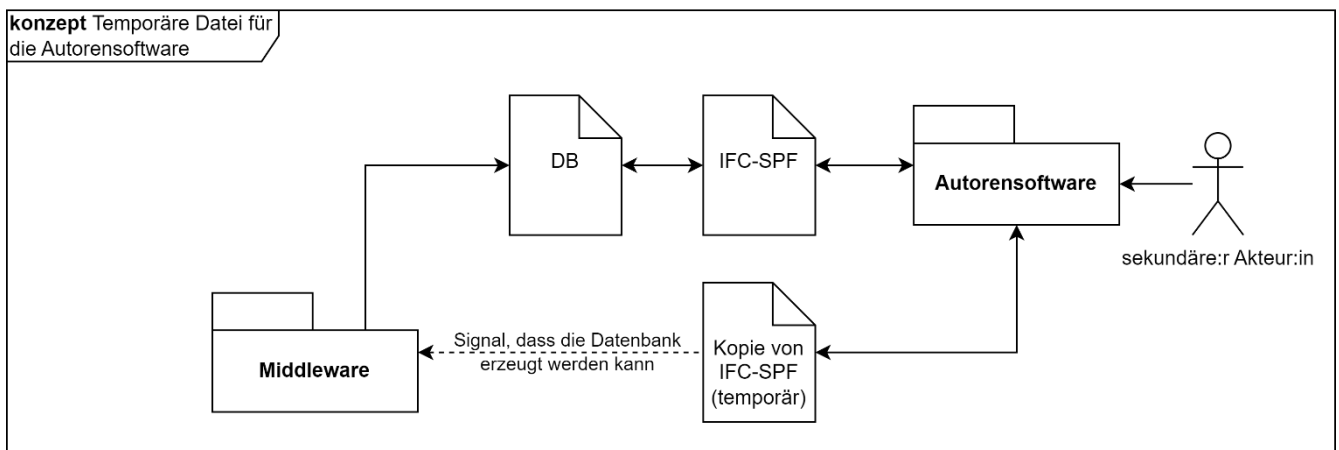


Abbildung 49: Temporäre Datei für die Autorenssoftware

### 7.7.3 Aktivitäten

Die Aktivitäten, die mit dem Systembaustein «Autorensoftware» verbunden sind, umfassen gemäss Abbildung 33 einerseits den Import von Daten aus dem IFC-SPF ins Autorensoftware und andererseits den Export des IFC-SPFs aus der Autorensoftware.

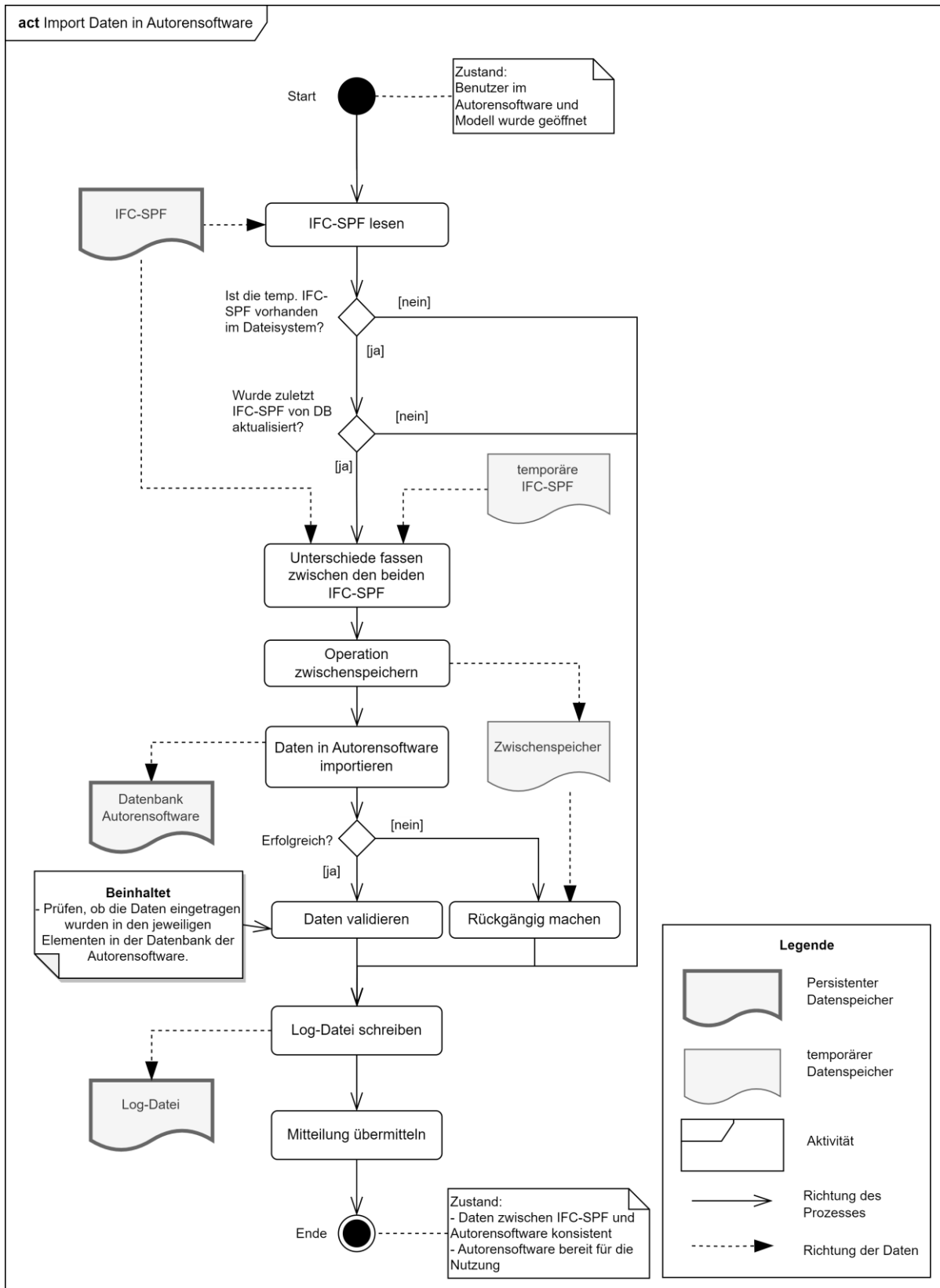


Abbildung 50: Aktivitätsdiagramm «Import Daten in Autorensoftware»

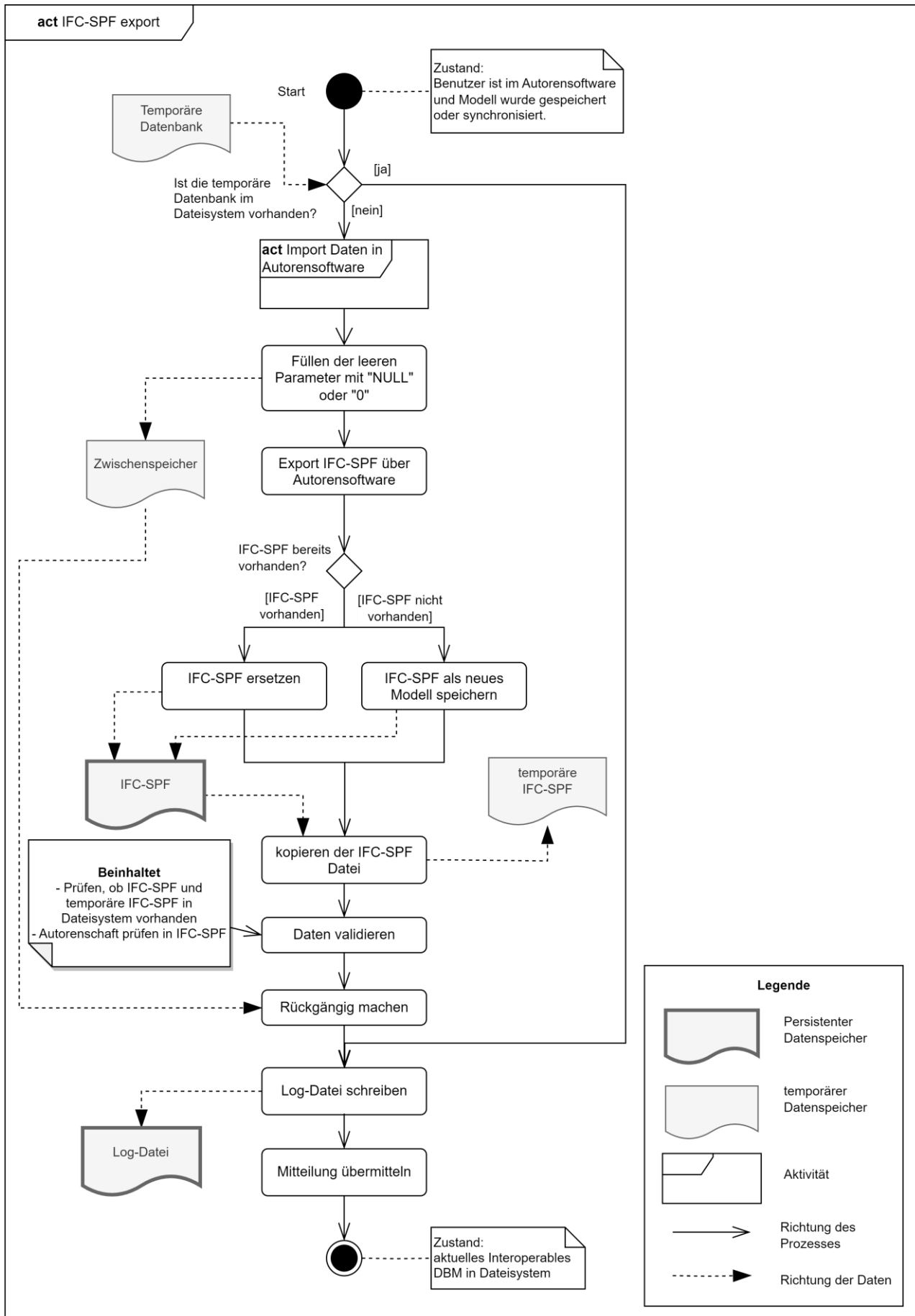


Abbildung 51: Aktivitätsdiagramm «IFC-SPF Export»

### 7.8 Sequenzierung

Die Sequenzierung stellt den zeitlichen Ablauf der genannten Aktivitäten aller Systembausteine dar.

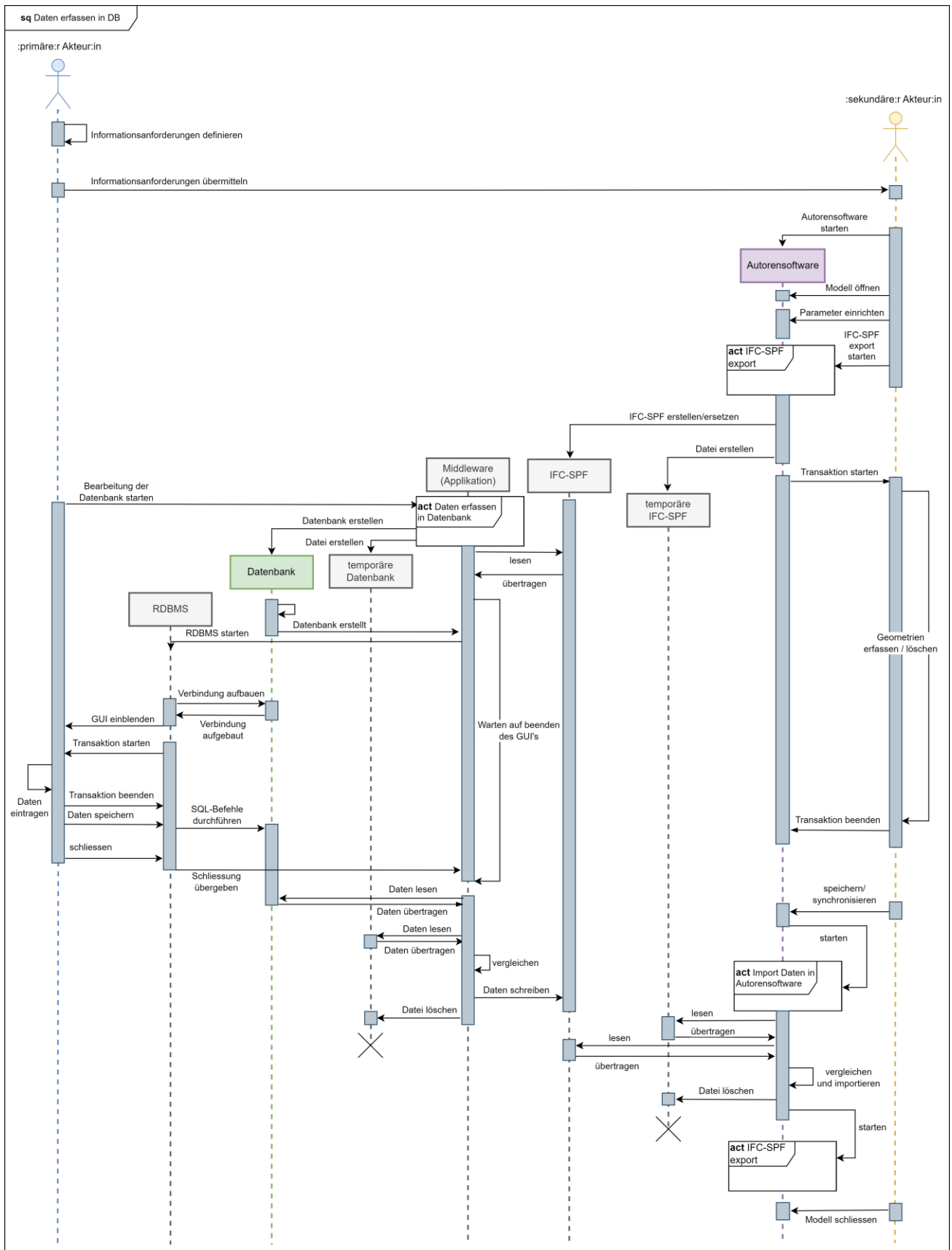


Abbildung 52: Sequenzdiagramm des gesamten Systems

## 8 Entwicklung des Proof of Concepts

Der Proof of Concept dient dazu, die Machbarkeit der konzeptuellen Arbeit zu zeigen, indem die Kernfunktionalitäten aus den Überlegungen implementiert werden. Die gewonnenen Erfahrungen werden dokumentiert, reflektiert und kritisch hinterfragt, um weitere Entwicklungen zu ermöglichen. Für den Nachweis der Machbarkeit werden insbesondere die Anwendungsfälle Nr. 1 bis 4 berücksichtigt, da sie eine zentrale Bedeutung hinsichtlich des Machbarkeitsnachweises haben. Der Quellcode des Proof of Concepts befindet sich im Anhang F und auf der Plattform Codeberg: <https://codeberg.org/tufanoztk/msc-db-ifc-revit>

### 8.1 Verwendete Technologien

Im Proof of Concept kommen diverse Softwares von unterschiedlichen Funktionen und technologischen Hintergründen zum Einsatz. Als Autorensoftware wird Revit von Autodesk Inc. verwendet, da es bereits vom Praxispartner eingesetzt wird und entsprechendes Wissen vorhanden ist. Dabei wird die stabile Version 2022 von Autodesk Revit benutzt. Autodesk Revit bietet eine Programmierschnittstelle (API) in .NET-Framework an, welche erlaubt auf die Funktionen innerhalb der Autorensoftware zuzugreifen. Die Anwendung pyRevit als RAD-Umgebung für das Autodesk Revit kann dazu verwendet werden, um auf die Programmierschnittstellen von Revit mit Python Skripten zuzugreifen. Nebst der Datenbanktechnologie SQLite wird die Software DB Browser verwendet, um ein Datenbank-Interface als Beispiel im Einsatz zu haben. Mit dem Programm «Database Difference Utility sqldiff.exe» von SQLite werden Unterscheidungen in den Datenbanken aufgezeigt. Mit IfcOpenShell kann die IFC-SPF programmtechnisch modifiziert werden. Python wird als Programmiersprache gewählt, da sie einerseits in PyRevit genutzt wird und eine umfangreich dokumentierte Sprache ist. Die Anwendung kann dadurch in einer einzigen Programmiersprache verfasst werden. Es wird die Python Version 3.8.5 eingesetzt, da zu diesem Zeitpunkt pyRevit Version 4.8.13 die CPython 3.8.5 Implementierung verwendet. Die beschriebenen Softwares sind in der Tabelle 6 zusammengestellt.

Tabelle 6: Verwendete Technologien

Software	Funktion	Version
Autodesk Revit	Autorensoftware	2022
SQLite	Datenbanktechnologie	3.35.5.
IfcOpenShell	IFC-Parser	0.7.0
DB Browser	Datenbank-Interface	3.12.2
pyRevit	Add-In für Autorensoftware	4.8.13
Autodesk Revit API	API	2022
sqldiff.exe	Programm	-
Python	Programmiersprache	3.8.5

## 8.2 Systemdesign

Das Systemdesign des Proof of Concepts wird gemäss der definierten Systemarchitektur aus Kapitel 7.1 entwickelt. Dabei werden die Technologien aus Tabelle 6 den jeweiligen Systembausteinen zugeordnet. Die Beschreibung des Systemdesigns wird in zwei Abbildungen aufgeteilt. Die Schnittstelle zwischen den beiden Abbildungen bildet das IFC-SPF. Die Abbildung 53 beschreibt die Funktion der Middleware in Verbindung mit der Datenbank und dem IFC-SPF. Der Python-Skript bildet den Dreh- und Angelpunkt für die Funktionen und Anweisungen des Teilsystems.

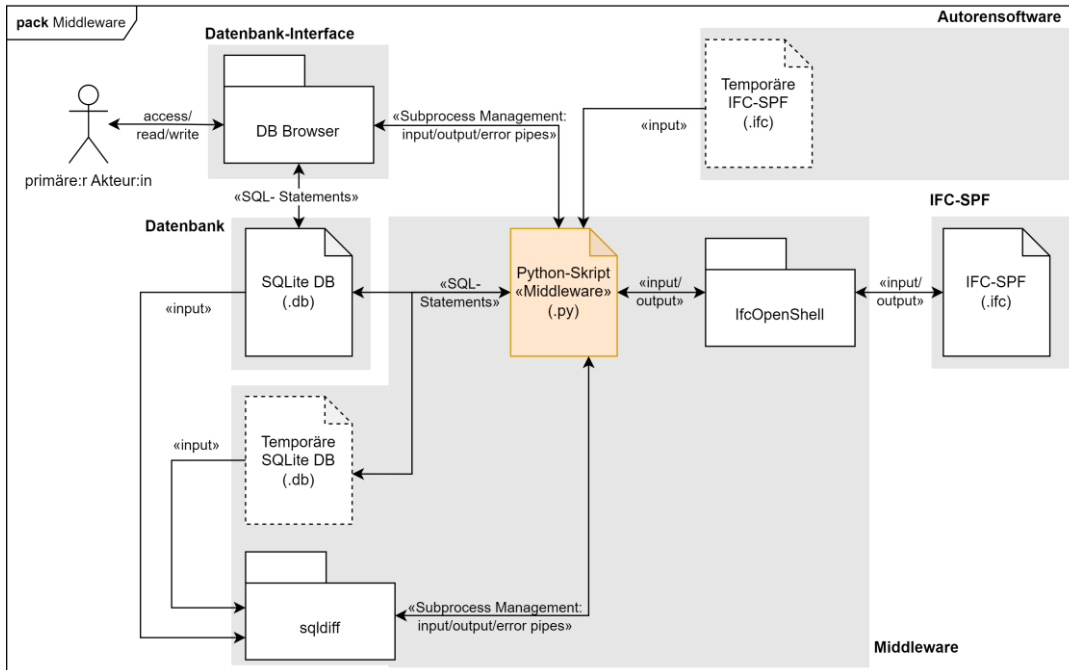


Abbildung 53: Paketdiagramm des Middlewares

Die Abbildung 54 beschreibt die Funktionen der Autorensoftware in Verbindung mit dem IFC-SPF. Auch hier bildet der Python-Skript den Dreh- und Angelpunkt für die Funktionen und Anweisungen des Teilsystems.

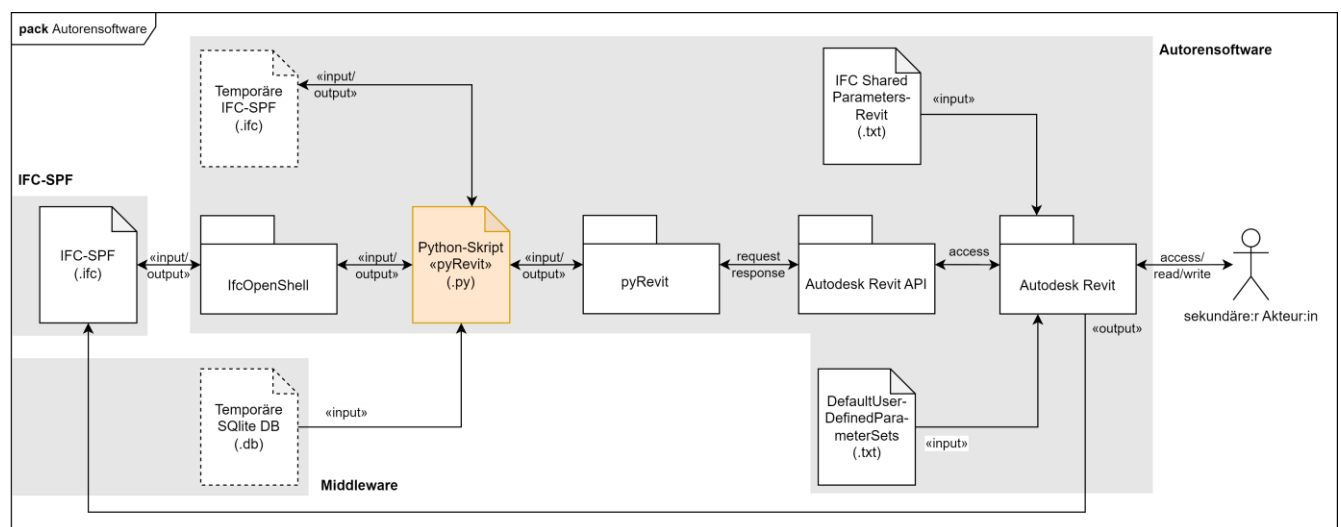


Abbildung 54: Paketdiagramm der Autorensoftware

### 8.3 Aufbau der relevanten Skripts

In einfacher Form mit Elementen aus dem UML werden die relevanten Komponenten im Detail aufgeschlüsselt und in einem Diagramm dargestellt. Die Abbildung 55 beschreibt den Aufbau des Skripts «Middleware». Initiiert wird das Skript mit dem Starten der Datenbank über den «db\_processor.py» aus dem Anhang F (Pfad: ./middleware/db\_processor.py). Es löst dabei das Paket «db\_init.py» aus, um die Datenbank zu erzeugen, das Datenmodell und die Tabellenrelationen anhand dem IFC-SPF zu erstellen und anschliessend die Daten aus dem IFC-SPF in die Datenbank zu übertragen. Im Nachgang wird der DB Browser über den «watchdog und subprocess» gestartet und die erzeugte Datenbank als temporäre SQLite-Datenbank im Dateisystem kopiert. Über den DB Browser kann die Datenbank bearbeitet werden. Nach dem Schliessen des DB Browsers folgt die Zwischenspeicherung der Änderungen in der Datenbank über den «sqldiff.exe». Mit einem Regex-Pattern werden die Output-Strings des Zwischenspeichers, welche als SQL-Statement vorliegen, in Attribute und Werte zerlegt. Schliesslich werden die Änderungen zusammen mit dem Überschreiben der Autorenschaft im IFC-SPF mithilfe von IfcOpenShell geschrieben.

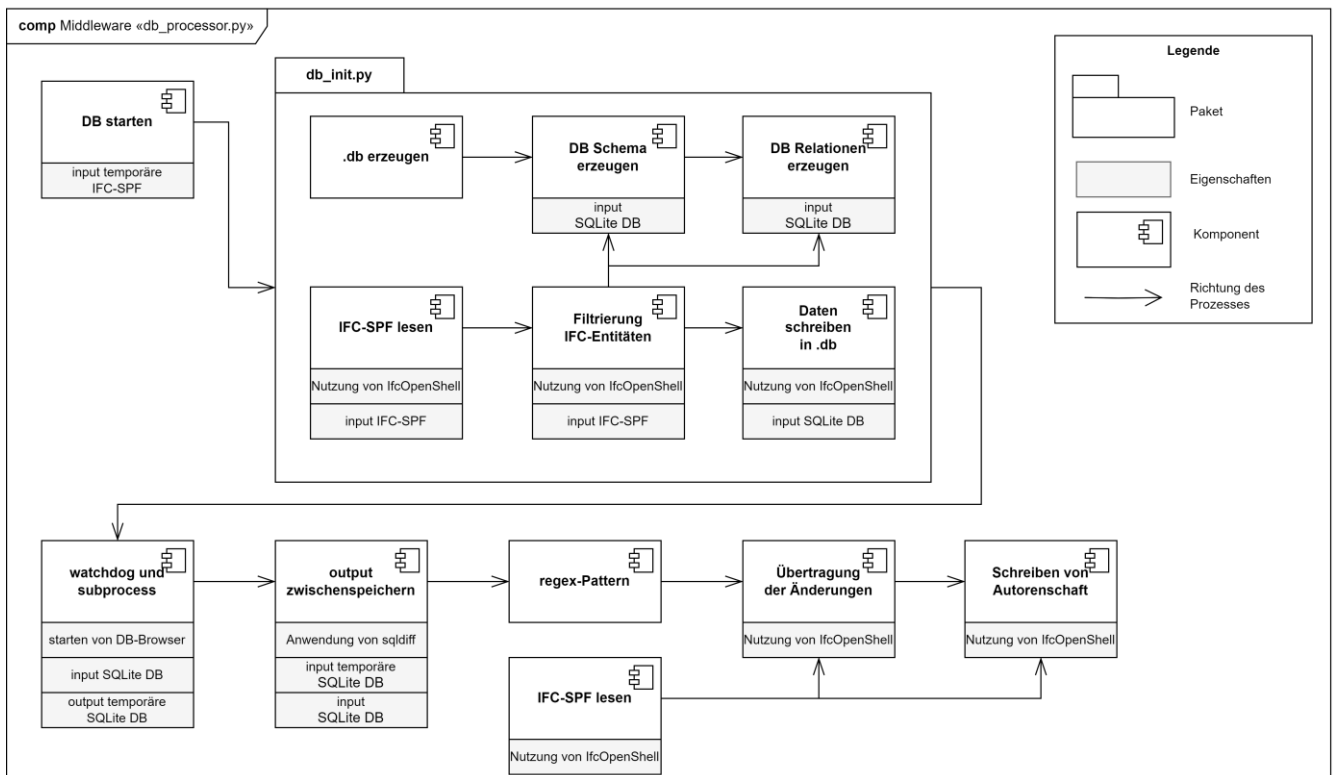


Abbildung 55: technischer Aufbau des Skripts «Middleware»

Die Abbildung 57 beschreibt den Aufbau des Skripts «pyRevit». Zurzeit wird das Skript manuell über einen Knopf in der GUI-Oberfläche initiiert, welcher über pyRevit konfiguriert ist (siehe Abbildung 56). Mit der Ergänzung des Event-Handlers aus dem Revit-API im Skript kann der Knopf, z.B. nach dem Speichern des Modells, automatisiert werden.

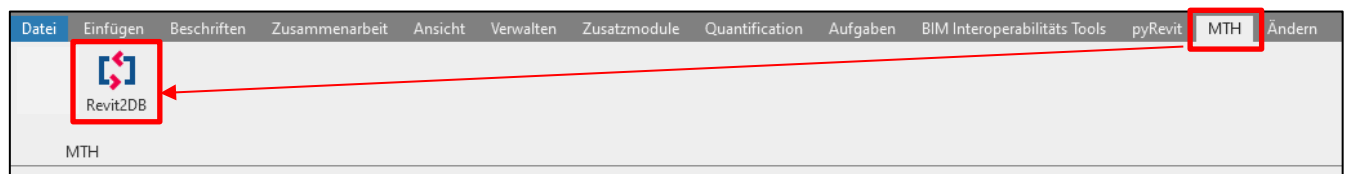


Abbildung 56: Revit Werkzeuge auf der GUI-Oberfläche von Revit 2022

Nach dem Starten des Skripts «script.py» aus dem Anhang F (Pfad: ./pyrevit/MTH.extension/MTH.tab/MTH.panel/Revit2DB.pushbutton/script.py) wird zunächst die Datenhoheit überprüft, indem im Dateisystem nach der temporären Datenbank gesucht wird. Falls diese vorhanden ist, wird der Prozess abgebrochen und ein Fenster (siehe Abbildung 66) wird angezeigt. Ist die temporäre Datenbank nicht vorhanden, erfolgt eine Überprüfung der Autorenschaft innerhalb des IFC-SPFs, um festzustellen, ob die Datei bereits in der Datenbank bearbeitet wurde. Mithilfe der Python-Programm-bibliothek «difflib» wird ein Vergleich zwischen dem IFC-SPF und dem temporären IFC-SPF durchgeführt, um festzustellen, ob Änderungen vorgenommen wurden. Erst nach dieser Überprüfung werden die Änderungen aus dem IFC-SPF in Revit übertragen. Abschliessend werden leere Parameter mit Werten gefüllt, das IFC-SPF exportiert und eine Kopie davon als temporäre Datei erstellt.

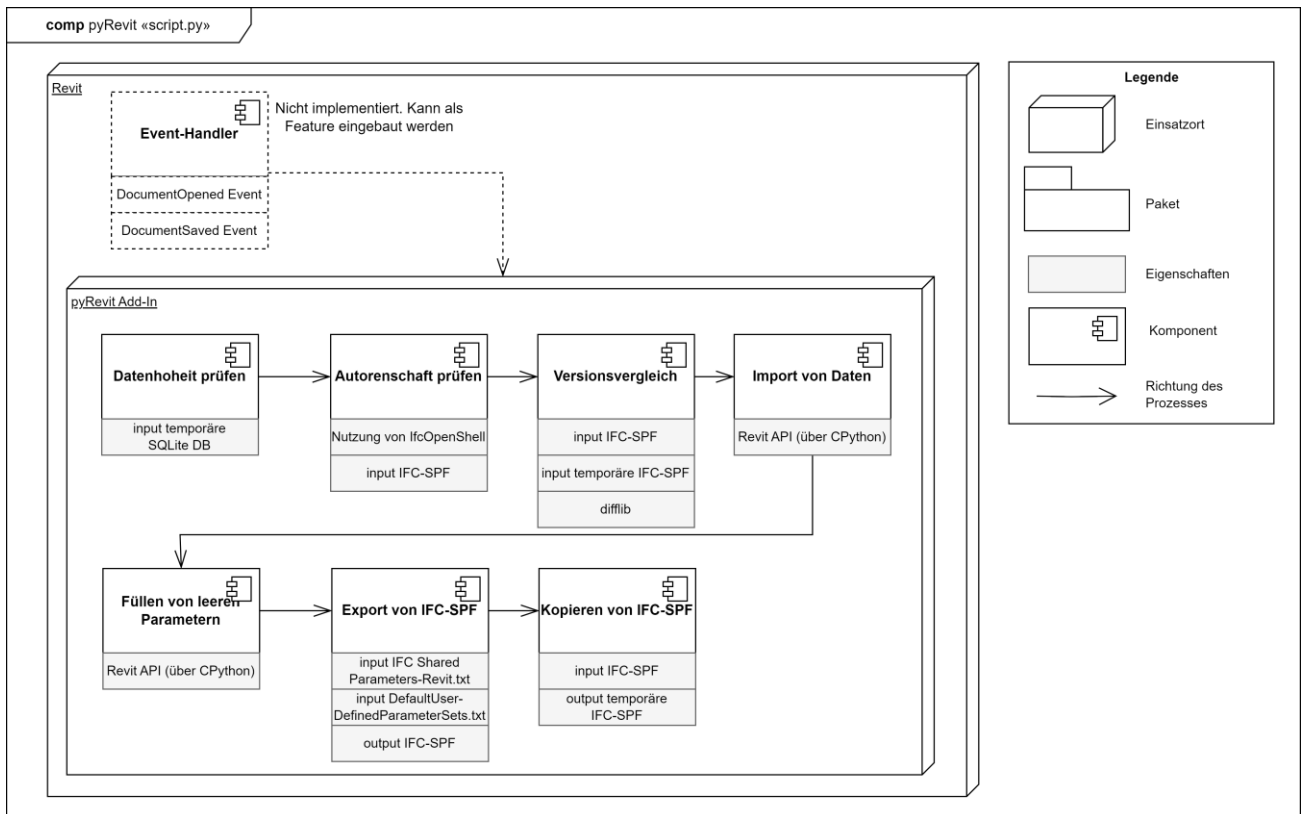


Abbildung 57: technischer Aufbau des Skripts «pyRevit»

### 8.4 Projektordner im Dateisystem

Als Projektordner wird ein Ordner im Dateisystem bezeichnet, bei dem zentrale Elemente wie die Revit-Projektdatei, die IFC-SPF und die SQLite-Datenbank abgelegt werden. Zusätzlich werden dort die relevanten Dateien für die Einrichtung der Revit-Datei abgelegt und temporäre Dateien werden über die Python-Skripts zur Steuerung des Datenflusses und der Datenhoheit verwaltet.

Name	Typ	Beschreibung
src	Dateiordner	Python-Skripte
.ifc_revit.ifc	IFC File	temporäre IFC-SPF
.temp.db	Data Base File	temporäre SQLite Datenbank
DefaultUserDefinedParameterSets.txt	Textdokument	Textdatei für den IFC-SPF export
IFC Shared Parameters-Revit.txt	Textdokument	Textdatei für die Definierung der Parameter
MTH.db	Data Base File	SQLite Datenbank
MTH.ifc	IFC File	IFC-SPF
MTH.rvt	Autodesk Revit-Projekt	Revit Projektdatei

Abbildung 58: Projektordner im Dateisystem des Betriebssystems

## 8.5 Einrichtung Autodesk Revit

Anhand einer Informationsanforderung kann die Einrichtung des Projektes im Revit erfolgen, um die erforderlichen Parameter für den Export im IFC-SPF bereitzustellen. Eine einfache Form mit den gewünschten Parametern ist in der Tabelle 7 aufgeführt. Die Beschreibungen lehnen sich an die IFC-Spezifikation. Im Rahmen des Proof of Concepts werden zwei Bauteile aus der Gebäudetechnik verwendet: ein Ventil und eine Pumpe. Ein Rahmenwerk oder eine Methodik zur ausführlichen Beschreibung von Informationsanforderungen wird nicht im Detail behandelt.

Tabelle 7: Informationsanforderung als Parametertabelle

Parameter	Beschreibung	Zugehörige Bauteile
Name	Name des Bauteils	Ventil, Pumpe
Testdruck	Der maximale Druck, dem das Ventil bei der Prüfung ausgesetzt war.	Ventil
Status	Status bzw. Phase des Bauteils <ul style="list-style-type: none"> <li>- «Neu» als neues Bauteil als Ergänzung</li> <li>- «Bestand» als Bauteil, das erhalten bleibt</li> <li>- «Abbruch» als Bauteil, das abgebrochen wird</li> <li>- «Temporär» als Bauteil und andere Bauelemente, die vorübergehend eingebaut werden.</li> </ul>	Ventil, Pumpe
Anschlussdimension	Die Grösse des Anschlusses des Bauteils.	Pumpe
Basistyp	Definiert allgemeine Typen von Pumpengrundplatten. Beinhaltet folgende Werte «FRAME», «BASE», «NONE» und «OTHER»	Pumpe

Das Revit wird entsprechend der Informationsanforderung konfiguriert. Hierbei wird die Textdatei «IFC Shared Parameters-Revit.txt» verwendet. Zusätzlich müssen für die Parametereinrichtung in Revit die IFC-Exporte korrekt übersetzt werden. Hierfür wird die Textdatei «DefaultUserDefinedParameterSets.txt» verwendet. Dieser Prozess der Einrichtung vom Revit kann über ein Skript automatisiert werden.

### 8.5.1 Benutzerdefinierte Parameter

Im folgendem Beispiel werden die Eigenschaftensätze von IFC genutzt. Dabei werden die gewünschten benutzerdefinierten Parameter resp. die gemeinsam genutzten Parameter für die Elemente aus der Domäne Gebäudetechnik, wie Pumpen und Ventile, den entsprechenden Revit-Familien zugeordnet. Es werden die Eigenschaftensätze «Pset\_PumpTypeCommon», «Pset\_PumpOccurrence» sowie «Pset\_ValveTypeCommon» verwendet. Innerhalb dieser IFC-Eigenschaftensätze werden die erforderlichen Parameter aus der Informationsanforderung entnommen. Die Übersetzung ist in der Tabelle 8 aufgeführt. Für einen korrekten Export von IFC-Entitäten werden zusätzlich die allgemeinen Parameter «IfcName», «IfcExportType», «IfcExportAs» und «IfcDescription» verwendet. Die benutzerdefinierten Parameter werden im «IFC Shared Parameters-Revit.txt» aufgeführt (siehe Abbildung 59).

Tabelle 8: Mapping-Tabelle zur Informationsanforderung

Bauteil	Parameter (aus IA)	Attribut	IfcPropertySet	IfcProperty
Ventil, Pumpe	Name	Name	-	-
Ventil	Testdruck	-	Pset_ValveTypeCommon	TestPressure
Ventil	Status	-	Pset_ValveTypeCommon	Status
Pumpe	Status	-	Pset_PumpTypeCommon	Status
Pumpe	Anschlussdimension	-	Pset_PumpTypeCommon	ConnectionSize
Pumpe	Basistyp	-	Pset_PumpOccurrence	BaseType

```
# This is a Revit shared parameter file.
# Do not edit manually.
*META VERSION MINVERSION
META 2 1
*GROUP ID NAME
GROUP 1 IFC Parameters
GROUP 516 Pset_PumpOccurrence
GROUP 518 Pset_PumpTypeCommon
#
*PARAM GUID NAME DATATYPE DATACATEGORY GROUP VISIBLE DESCRIPTION USERMODIFIABLE HIDEWHENNOVALUE IFC-Export
PARAM 5e9ea49b-cd01-4f34-acd9-4d5dbf8bcaf5 IfcName TEXT 1 1 1 0
PARAM d9ec89db-5993-4392-bc5e-70a3d4e16eca IfcExportType TEXT 1 1 1 0
PARAM e33ed9e8-258d-4fb7-a3fe-9e6bde86ab5d IfcExportAs TEXT 1 1 1 0
#
*PARAM GUID NAME DATATYPE DATACATEGORY GROUP VISIBLE DESCRIPTION USERMODIFIABLE HIDEWHENNOVALUE Pumpe
PARAM e6afba00-d1fe-11e1-8000-00215ad4efdf Pset_PumpTypeCommon.ConnectionSize LENGTH 518 1 1 IfcPositiveLengthMeasure 1 0
PARAM c67ffb00-d1fe-11e1-8000-00215ad4efdf Pset_PumpTypeCommon.Status TEXT 518 1 1 PEnum_ElementStatus 1 0
PARAM 8587e600-d1fe-11e1-8000-00215ad4efdf Pset_PumpOccurrence.BaseType TEXT 516 1 1 PEnum_PumpBaseType 1 0
#
GROUP 717 Pset_ValveTypeCommon Vent
*PARAM GUID NAME DATATYPE DATACATEGORY GROUP VISIBLE DESCRIPTION USERMODIFIABLE HIDEWHENNOVALUE
PARAM 103808cf-5fe6-4d58-9e73-17c7493147b5 Pset_ValveTypeCommon.Status TEXT 717 1 1 PEnum_ElementStatus 1
PARAM 7875f200-d214-11e1-8000-00215ad4efdf Pset_ValveTypeCommon.TestPressure NUMBER 717 1 1 IfcPressureMeasure 1
```

Abbildung 59: Ausschnitt aus dem IFC Shared Parameters-Revit.txt

### 8.5.2 3D Geometrien der Bauteile

Für die Modellierung der Geometrien werden einfache Revit-Familien verwendet, welche standardmässig von Autodesk zur Verfügung gestellt werden. Zusätzlich werden eigene Parameter aus der Tabelle 8 in die jeweiligen Revit-Familien ergänzt. Der Parameter «IfcGUID» wird automatisch erzeugt, wenn im IFC-Export «StoreIFCGUID» inkludiert wird.

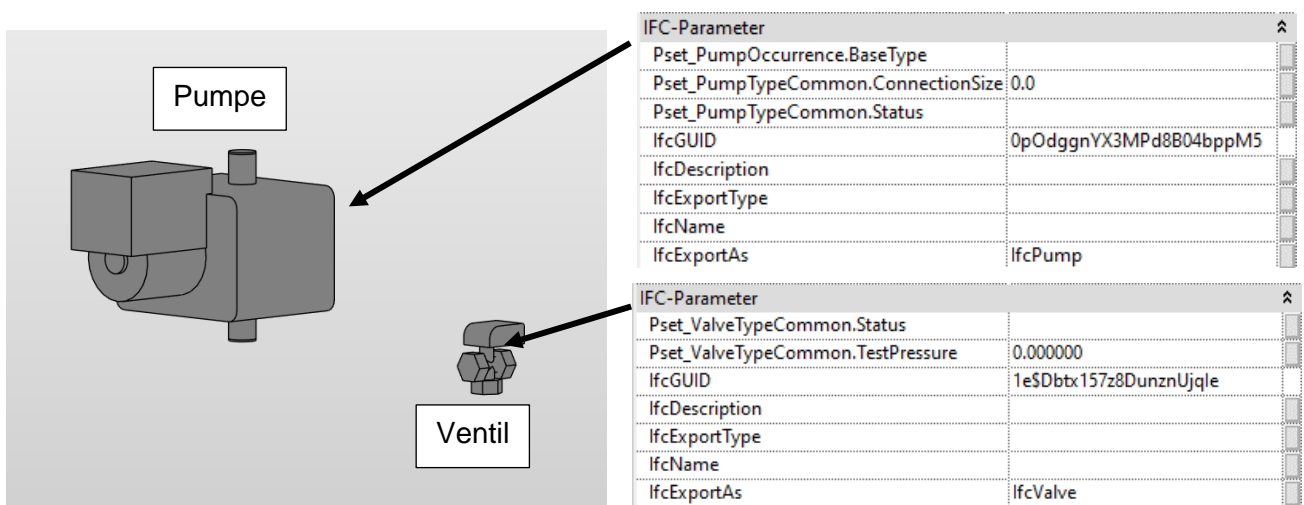


Abbildung 60: 3D-Geometrien mit den dazugehörigen IFC-Parametern (links: Pumpe, rechts: Ventil und Ansicht aus Revit)

### 8.5.3 Definierung des IFC-Exports

Die eingerichteten Parameter in Revit sind auch im IFC-Export zu definieren. Hierbei werden die Eigenschaftensätze als Export in die entsprechenden IFC-Entitäten wie «IfcPump» und «IfcValve» definiert. Die Zeilen werden am Ende der Textdatei ergänzt (siehe Abbildung 61).

```
# User Defined PropertySet Definition File
#
# Format:
#   PropertySet:  <Pset Name>      I[instance]/T[type]      <element list separated by ','>
#   <Property Name 1>  <Data type>  <[opt] Revit parameter name, if different from IFC>
#   <Property Name 2>  <Data type>  <[opt] Revit parameter name, if different from IFC>
#   ...
#
# Data types supported: Area, Boolean, ClassificationReference, ColorTemperature, Count, Currency,
# ElectricalCurrent, ElectricalEfficacy, ElectricalVoltage, Force, Frequency, Identifier,
# Illuminance, Integer, Label, Length, Logical, LuminousFlux, LuminousIntensity,
# NormalisedRatio, PlaneAngle, PositiveLength, PositivePlaneAngle, PositiveRatio, Power,
# Pressure, Ratio, Real, Text, ThermalTransmittance, ThermodynamicTemperature, Volume,
# VolumetricFlowRate
#
# Since 24.1.0, type properties are not exported in case they are not specified in the user-defined property sets file.
# To export type properties for the IFCElement, for example,
# a user should include "IFCElementType" in the proper .txt file.
#
PropertySet:  Pset_PumpTypeCommon      I      IfcPump
Pset_PumpTypeCommon.Status      Text
Pset_PumpTypeCommon.ConnectionSize      PositiveLength
PropertySet:  Pset_PumpOccurrence      I      IfcPump
Pset_PumpOccurrence.BaseType      Text
PropertySet:  Pset_ValveTypeCommon      I      IfcValve
Pset_ValveTypeCommon.Status      Text
Pset_ValveTypeCommon.TestPressure      Integer
```

Abbildung 61: Ausschnitt aus dem DefaultUserDefinedParameterSets.txt

Im Python-Skript «pyRevit» (Abbildung 54 und Abbildung 57) wird der IFC-Export folgendermassen definiert:

```
(python3 und Autodesk.Revit)
def ifc_export_func(project_folder_path, ifc_file_name):
    ifc_export = IFCExportOptions()
    ifc_export.FileVersion = IFCVersion.IFC4DTV
    ifc_export.ExportBaseQuantities = False
    ifc_export.AddOption("StoreIFCGUID", "True")
    ifc_export.AddOption("UseFamilyAndTypeNameForReference", "False")
    ifc_export.AddOption("ExportUserDefinedPsets", "True")
    ifc_export.AddOption("ExportInternalRevitPropertySets", "False")
    ifc_export.AddOption("ExportIFCCommonPropertySets", "False")
    ifc_export.AddOption("ExportUserDefinedPsetsFileName", user_defined_parameter)
```

### 8.6 Export von IFC-SPF

Beim Export von IFC-SPF sind im Revit die leeren Parameter mit Werten entsprechend den Datentypen zu füllen, um im IFC-SPF Export von Revit berücksichtigt zu werden. Die Parameter vom Datentyp «String» werden mit dem Wert «NULL» erfasst, während die Parameter der Datentypen «Integer» oder «Double» mit dem Wert «0» erfasst werden. Die Erfassung findet über den Python-Skript «pyRevit» statt. Die Abbildung 62 zeigt ein Beispiel für die «IfcPropertySingleValues», welche nach dem Ausführen der Anwendung, in der Datenbank erscheinen. Die Werte sind dann entsprechend vom Nutzenden nochmals auf die gültigen Werte zu ändern.

spf_id	Name	Description	NominalValue	Unit
1	2426	Pset_PumpTypeCommon.Status	{ "type": "IfcText", "value": "NULL" }	NULL
2	2428	Pset_PumpOccurrence.BaseType	{ "type": "IfcText", "value": "NULL" }	NULL
3	3587	Pset_ValveTypeCommon.Status	{ "type": "IfcText", "value": "NULL" }	NULL
4	3588	Pset_ValveTypeCommon.TestPressure	{ "type": "IfcInteger", "value": 0 }	NULL

Abbildung 62: Resultat von leeren Parametern aus Revit (Ansicht aus dem DB Browser)

### 8.7 Initialisierung SQLite-Datenbank

Zusätzlich zum Revit wird auch die SQLite-Datenbank eingerichtet. Die Konfiguration erfolgt über das Python-Skript «Middleware» (Abbildung 53 und Abbildung 55). Das Datenbankschema «Eigenschaften» (Abbildung 42) wird spezifisch für das System im Proof of Concept aufgebaut.

#### 8.7.1 Mapping von Datentypen

Die Datentypen im IFC-Datenmodell sind gemäss der EXPRESS-Sprache definiert. Es ist jedoch nicht möglich, die Datentypen der EXPRESS-Sprache direkt und ohne Anpassungen in SQL-Sprache zu überführen, da beide Sprachen unterschiedliche Datendefinitionen aufweisen. Die vorhandenen Datentypen aus der EXPRESS-Sprache, die im IFC-Datenmodell verwendet werden, können in drei Hauptkategorien unterteilt werden (BuildingSMART International, n.d.; Zhu & Shide, 2021):

- Simple Data Types
- Aggregation Data Types
- Declared Data Types

Die «Simple Data Types» umfassen primitive Datentypen wie String, Integer und Boolean. Bei den «Aggregation Data Types» handelt es sich um die Verschachtelung mehrerer primitiver Datentypen zu einem zusammengefassten Datentyp. Unter den «Declared Data Types» fallen Enumeration und Select Datentypen, die Sammlungen von genannten Datentypen enthalten können, wie beispielsweise andere Entitäten oder Listen von String-Werten. Der Unterschied zur Enumeration besteht darin, dass bei SELECT nur ein Wert verwendet werden kann (BuildingSMART International, n.d.).

Simple Data Types Mapping		
IFC (EXPRESS)	SQLite	Implementierung
REAL	REAL	REAL
INTEGER	INTEGER	INT
NUMBER	REAL	REAL
LOGICAL (true, false, unknown)	INTEGER	INT (1, 0, 2)
BOOLEAN (true, false)	INTEGER	INT (1, 0)
STRING	TEXT	TEXT
BINARY	BLOB	BLOB
-	NULL	NULL

Aggregation Data Types Mapping		
IFC (EXPRESS)	SQLite	Implementierung
SET	JSON / JSONB	TABLE / JSON
BAG	JSON / JSONB	TABLE / JSON
LIST	JSON / JSONB	TABLE / JSON
ARRAY	JSON / JSONB	TABLE / JSON

Declared Data Types Mapping		
IFC (EXPRESS)	SQLite	Implementierung
ENUMERATION	TEXT	TEXT (+Einschränkungen)
SELECT	JSON / JSONB	TABLE / JSON

Abbildung 63: Mapping von Datentypen (in Anlehnung an (Wei et al., 2009) und (Lee et al., 2014))

Das SQLite basiert auf der SQL-Sprache und bedient sich von vorhandenen Sprachelementen. Jede Datenbanktechnologie implementiert die Sprachelemente der SQL-Sprache unterschiedlich. SQLite

bietet zentrale Datentypen, welche als Speicherklassen definiert werden. Benutzerdefinierte Datentypen können im SQLite nicht erzeugt werden (SQLite, 2022).

Das Mapping in Abbildung 63 präsentiert die entsprechenden Datentypen der IFC-Spezifikation sowie SQLite und zeigt die abschliessende Implementierung. Die fett markierten Varianten innerhalb der Implementierung werden effektiv umgesetzt. Bei den «Simple Data Types» können die meisten primitiven Datentypen aus dem IFC übernommen werden, während andere durch eine Logik basierend auf dem Datentyp INTEGER abgebildet werden können.

Die «Aggregation Data Types», sowie der Datentyp SELECT aus den «Declared Data Types», können entweder durch separate Tabellen gemäss der Mapping-Methode P4 aus Kapitel 7.5.1 beschrieben werden oder einfacher durch die Verwendung eines JSON-Datenformats. Die Anwendung wird mit einem JSON-Datenformat implementiert, um das Datenbankmodell übersichtlicher zu gestalten, wodurch keine weiteren Tabellen und Relationen implementiert werden müssen.

Für die «Declared Data Types» werden die Enumerationen über den Datentyp TEXT dargestellt. Da alle Enumerationen im IFC-Datenmodell als Datentyp STRING ausgedrückt werden können, können sie bei der Speicherung in der Datenbank im Datentyp TEXT übersetzt werden (Zhu & Shide, 2021). Die Werte aus den Enumerationen im IFC werden als Einschränkungen in Form von «CHECK-Constraints» implementiert. Eine Alternative zu «CHECK-Constraints» ist die Trigger-Funktion von SQLite, die jedoch nicht genutzt wird, da die Implementierung von «CHECK-Constraints» einfacher ist.

### 8.7.2 Erstellung des Datenbankmodells

Das Datenmodell in Abbildung 64 basiert auf dem IFC-Konzept «Eigenschaften» und zeigt im Beispiel der «Pumpe» auf, welche Tabellen, Tabellenspalten, Datentypen, Primärschlüssel sowie Beziehungen zwischen den Tabellen vorhanden sind. Bei der Erstellung der Tabellen in der Datenbank wird die Namensgebung der Tabellen und Tabellenspalten (Attribute der IFC-Objekten) aus dem IFC übernommen. Ausser die Beziehungstabellen haben eine benutzerdefinierte Namensgebung mit einem Präfix «\_\_» und der Kombination der beteiligten IFC-Entitäten. Die SQLite erlaubt nicht Tabellen mit denselben Namen zu erzeugen. Weshalb die Beziehungstabellen nicht nach «IfcRelDefinesByProperties» benannt werden.

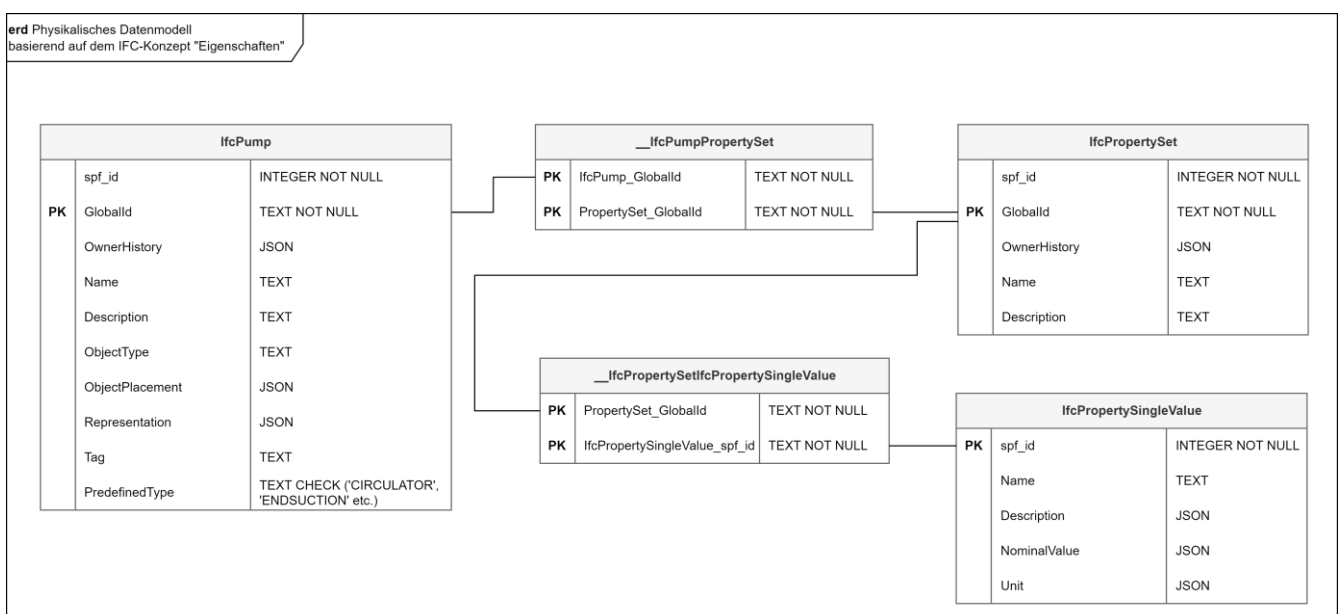


Abbildung 64: Physikalisches Datenbankmodell. Veranschaulichung der Relationen

Dieses Beispiel wird für die Erstellung der Tabelle «IfcPump» folgendermassen als SQL-Befehl aufgebaut:

```
(python3 und sqlite3)
CREATE TABLE "IfcPump" (
  "spf_id" INTEGER NOT NULL,
  "GlobalId" TEXT,
  "OwnerHistory" JSON,
  "Name" TEXT,
  "Description" TEXT,
  "ObjectType" TEXT,
  "ObjectPlacement" JSON,
  "Representation" JSON,
  "Tag" TEXT,
  "PredefinedType" TEXT CHECK(
  "PredefinedType" IN (
  'CIRCULATOR', 'ENDSUCTION', 'NOTDEFINED',
  'SPLITCASE', 'SUBMERSIBLEPUMP', 'SUMPPUMP',
  'USERDEFINED', 'VERTICALINLINE',
  'VERTICALTURBINE')),
  PRIMARY KEY("GlobalId")
);
```

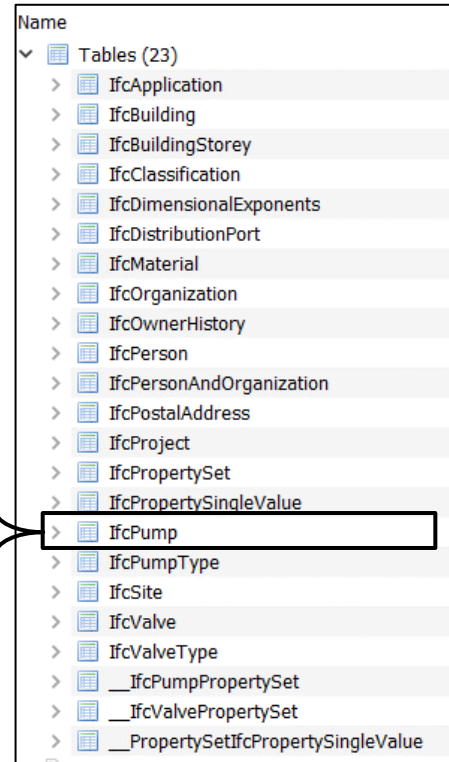


Abbildung 65: Tabellen im DB Browser

Zur Beschreibung der Beziehungen im Datenbankmodell wird die Variante 1 aus dem Kapitel 7.5.3 verwendet, da auch Revit 2022 diese Variante bevorzugt. Implementiert wird dies wie folgt für die Beziehung zwischen einem IFC-Objekt und einem IfcPropertySet:

```
(python3 und sqlite3)
def create_pset_tables(entity_type):
  statement = f"""CREATE TABLE IF NOT EXISTS __{entity_type}PropertySet (
  {entity_type}_GlobalId TEXT NOT NULL,
  PropertySet_GlobalId TEXT NOT NULL,
  PRIMARY KEY({entity_type}_GlobalId, PropertySet_GlobalId),
  FOREIGN KEY(PropertySet_GlobalId) REFERENCES IfcPropertySet(GlobalId),
  FOREIGN KEY({entity_type}_GlobalId) REFERENCES {entity_type}(GlobalId)
  );"""
  db_conn.execute(statement)
```

## 8.8 Datenimport in Revit

Der Datenimport von IFC-SPF in Revit erfolgt durch den Vergleich zwischen der originalen IFC-SPF und einer temporären Kopie. Zuerst wird über das Attribut «IfcApplication» festgestellt, wer zuletzt die IFC-SPF revidiert hat. Anschliessend werden die beiden Dateien mithilfe der Python-Programm-bibliothek «difflib» verglichen. Die festgestellten Unterschiede werden in einer Liste gespeichert und mithilfe der Revit-API den entsprechenden Elementen in Revit zugeordnet. Die Zuordnung zu den Elementen erfolgt über den Revit-Parameter «IFC-GUID» für IFC-Entitäten mit einem GlobalId. Wenn hingegen IFC-Entitäten ohne GlobalId z.B. das «IfcPropertySingleValue» in der Datenbank verändert wird, wird

dies anhand der Beziehungen zu den IFC-Objekten im IFC mit IfcOpenShell zurückverfolgt, um festzustellen, zu welcher IFC-Entität diese gehören und dann entsprechend die Daten in Revit zu übertragen. Das Mapping der Attribute aus dem IFC zu den Parametern in Revit wird anhand der Tabelle 9 durchgeführt.

Tabelle 9: Mapping der Attribute aus dem IFC zu Parameter in Revit

Attribute aus dem IFC	Parameter in Revit
GlobalId	IFC-GUID
PredefinedType	IfcExportType
Name	IfcName
Description	IfcDescription

Mit Dialogfenstern (Abbildung 66 und Abbildung 67) wird im Revit dem Nutzenden verdeutlicht, wer die Datenhoheit im IFC-SPF besitzt. Zudem werden mit Print-Befehlen die Prozessschritte aus dem Python-Skript im Revit nachvollziehbar gestaltet.



Abbildung 66: Dialogfenster beim IFC-Export von pyRevit

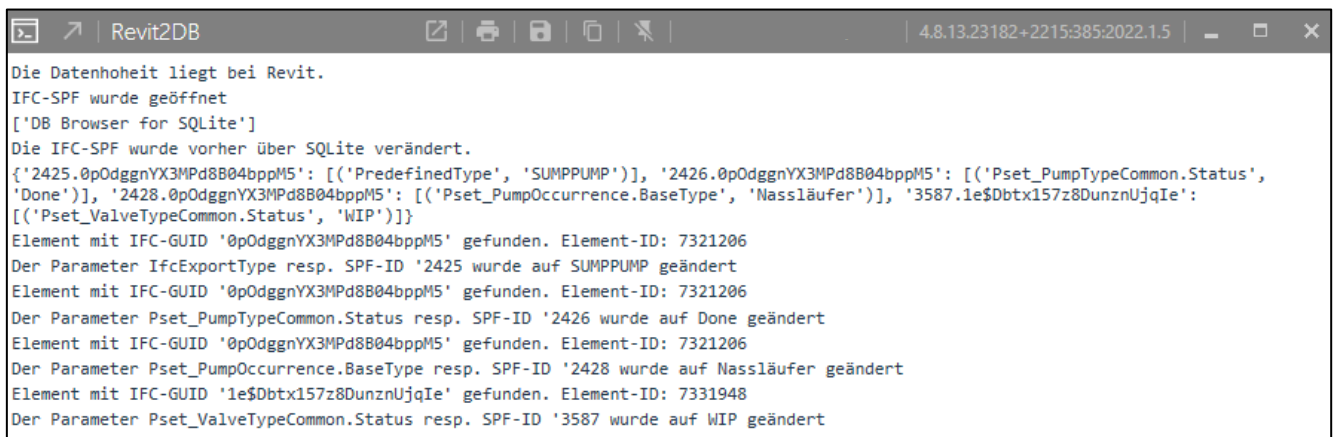


Abbildung 67: Importvorgang in Revit (Dialogfenster für Print-Befehle aus dem Python-Skript über pyRevit)

### 8.9 Mögliche GUI-Implementierung

Einige der Anforderungen der KMUs aus dem Kapitel 4.7 konzentrieren sich auf den Bereich der GUI-Entwicklung. Um diesen Anforderungen gerecht zu werden, widmet sich dieses Kapitel einer groben Übersicht. Dabei wird aufgezeigt wie ein SELECT-Statement aussehen könnte bei der Erstellung von Ansichten in der Datenbank und wie beispielsweise eine GUI entwickelt werden könnte. Im DB Browser kann mithilfe eines SQL-Sprachelements SELECT eine Ansicht erstellt werden. Im nachfolgenden Quellcode wird die Ansicht für das Datenbankmodell (Abbildung 64) erzeugt. Das SELECT-Statement greift auf die Spalten in den Tabellen «IfcPump», «IfcPropertySet» und «IfcPropertySingleValue» zu.

```
(sqlite3)
SELECT IfcPump.GlobalId as "GlobalId der Pumpe", IfcPump.Name as "Name der Pumpe", IfcProperty-
Set.Name as "Eigenschaftenset der Pumpe", IfcPropertySingleValue.Name as "Eigenschaft", IfcProperty-
SingleValue.NominalValue as "Wert"
FROM IfcPump
JOIN __IfcPumpPropertySet ON IfcPump.GlobalId = __IfcPumpPropertySet.Ifcpump_GlobalId
JOIN IfcPropertySet ON __IfcPumpPropertySet.PropertySet_GlobalId = IfcPropertySet.GlobalId
JOIN __PropertySetIfcPropertySingleValue ON IfcPropertySet.GlobalId = __PropertySetIfcPropertySin-
gleValue.PropertySet_GlobalId
JOIN IfcPropertySingleValue ON IfcPropertySingleValue.spf_id = __PropertySetIfcPropertySingleVa-
lue.IfcpumpPropertySingleValue_spf_id;
```

Das SELECT-Sprachelement erzeugt eine Tabelle mit den relevanten Spalten, welche im Anschluss in einer Web GUI angezeigt und modifiziert werden kann. Das Ergebnis der Ansicht im DB Browser ist in der nachstehenden Abbildung dargestellt.

GlobalId der Pumpe	Name der Pumpe	Eigenschaftenset der Pumpe	Eigenschaft	Wert
0pOdggnyX3MPd8B04bppM5	Heizungspumpe	Pset_PumpTypeCommon	Pset_PumpTypeCommon.Status	{"type": "IfcText", "value": "Done"}
0pOdggnyX3MPd8B04bppM5	Heizungspumpe	Pset_PumpOccurrence	Pset_PumpOccurrence.BaseType	{"type": "IfcText", "value": "Nassläufer"}

Abbildung 68: Ansicht der Pumpe mit den Eigenschaftensätzen, den Eigenschaften und den Eigenschaftswerten

Eine GUI kann über die Python-basierte, sogenannte Microweb framework, «Flask» erstellt werden (GeeksforGeeks, 2023). Flask wird für Webapplikation verwendet. Dadurch wird es möglich eine einfache Webbasierte GUI zu implementieren und den DB Browser damit zu ersetzen (siehe Abbildung 69). Eine einfachere Alternative ist die Anbindung der SQLite-Datenbank an das MS Access von Microsoft über einen ODBC-Treiber, wobei MS Access als GUI zur Erstellung von Formularen und Ansichten dient. Es ist anzumerken, dass es sich dann dabei um eine Windows-Umgebung handelt.

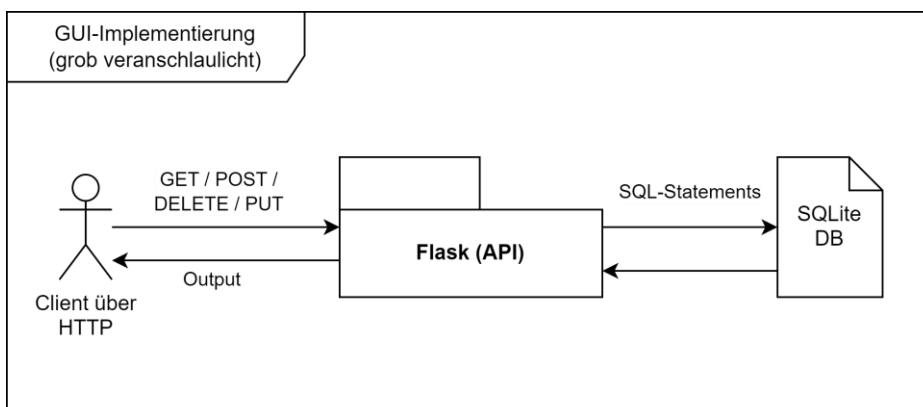


Abbildung 69: GUI-Implementierung

## 8.10 Limitierungen und Weiterentwicklung

Das Proof of Concept weist derzeit einige Limitationen auf, die angegangen werden sollten. Eine dieser Limitationen der Anwendung ist es, dass es ein Einzelbenutzersystem ist. Die Mehrbenutzerfähigkeit hat für den Nachweis der Machbarkeit eine sekundäre Bedeutung. Eine mögliche Lösung könnte die Zählung der Datenbankzugriffe und das Löschen der temporären SQLite-Datenbank nach dem Schliessen aller Zugriffe sein. Dabei wird SQLite als eine serverbasierte Datenbank eingesetzt. Weitere Mechanismen zur Sicherung der Datenkonsistenz, wie Zwischenspeicherung und Logdateien, müssen in die Python-Skripte integriert werden. Die Implementierung beschränkt sich auf das IFC-Konzept «Eigenschaften» und berücksichtigt nur die IFC-Version 4.0. In der Datenbank kann derzeit nur ein Wert in einem Tupel geändert werden, aufgrund der Beschränkungen des vorhandenen Regex-Patterns. Dies gilt auch Änderungen im Datensätzen mit Datentypen von Integer und Double. Die Synchronisation erfolgt derzeit manuell, aber durch die Implementierung eines Event-Hooks über pyRevit kann dieser Prozess automatisiert und im Hintergrund beim Speichern des Revit-Projekts ausgelöst werden. Zuletzt ist eine korrekte Umkodierung von UTF-8 auf ISO-8859-1 während des Schreibprozesses aus der Datenbank ins IFC-SPF zu implementieren (BuildingSMART International, 2023).

Weitere Schritte sind erforderlich, bis das System seine Anwendung in der Praxis finden kann. In der Abbildung 70 sind für den Praxispartner fünf Schritte bis zur Vollendung der Anwendung aufgeführt. Dies soll den aktuellen Stand der Implementierung verdeutlichen. Der nächste Schritt besteht darin, die Kernfunktionalitäten zu vervollständigen. Darauf folgt die Entwicklung einer GUI und die Ergänzung von Zusatzfunktionen. Abschliessend sind die Migration der Anwendung in die Cloud und die damit verbundenen Zugriffsregelungen zu konzipieren und zu erarbeiten.



Abbildung 70: Weitere Schritte bis zur Anwendung in der Praxis

## 9 Rückmeldungen aus der Praxis

Das Ergebnis des Proof of Concepts wurde dem Praxispartner präsentiert, wobei die Anwendung und ihr Konzept ausführlich vorgestellt wurden. Ziel war es festzustellen, ob die Implementierung den Anforderungen des Praxispartners entspricht und in der Praxis nützlich ist. In diesem Kapitel werden die Rückmeldungen der Mitarbeiter des Praxispartners aus verschiedenen Positionen zusammengefasst.

Das Feedback ist positiv zu beurteilen, da der implementierte Lösungsansatz den Anforderungen des Praxispartners entspricht. Mehrwerte sind zu beobachten, wie beispielsweise das Einbinden von externen Fachleuten im Modellierungsprozess und eine engere Zusammenarbeit mit weniger Informationsaustausch, ohne die Notwendigkeit der Lizenzierung der Autorensoftware. Hierbei handelt es sich lediglich um ein Proof of Concept, weshalb die Bedürfnisse bezüglich der grafischen Benutzeroberfläche nicht demonstriert werden konnte. Die Möglichkeiten wurden jedoch erläutert.

Während der Diskussion wurden verschiedene Fragen hinsichtlich der Systemeigenschaft aufgeworfen. Dabei ging es um die Möglichkeit einer gemeinsamen Nutzung der Datenbank sowie um die Option, diese in der Cloud zu speichern, beispielsweise über Plattformen wie Microsoft OneDrive oder Autodesk Docs. Zudem wurde die Frage aufgeworfen, warum Excel nicht für den Import und Export verwendet wurde. Excel ist allgemein bekannt und kann von jeder Fachperson ohne zusätzliche Erklärungen genutzt werden. Es wurde angemerkt, dass es in diesem Beispiel vorteilhafter wäre, Excel zu verwenden. Thematisiert wurde auch, ob die Systemarchitektur bei Grossprojekten an seine Grenzen stösst. Auf die Frage nach der gemeinsamen Nutzung der Datenbank bzw. der Mehrbenutzerfähigkeit wurde ergänzt, dass dieser Ansatz nicht als kritisch angesehen wird. Zudem wird die gleichzeitige Bearbeitung der Datenbank als unwahrscheinlich eingeschätzt.

Es wurde festgestellt, dass die vorgeschlagene Lösung möglicherweise zu komplex für kleine Unternehmen sein könnte. Es wird vermutet, dass sich kleine Unternehmer möglicherweise den Aufwand für die Implementierung der Systemarchitektur nicht vorstellen können. Allerdings wird darauf hingewiesen, dass ein Softwarehersteller möglicherweise die Implementierung für die Branche übernehmen und diese als Produkt speziell für kleine und mittlere Unternehmen anbieten könnte.

Trotz der kritischen Hinterfragung beabsichtigt der Praxispartner, die Ressourcen zu nutzen, um die vollständige Implementierung im Unternehmen voranzutreiben und den Modellierungsprozess einzubeziehen. Zunächst wird die Nutzung intern im Unternehmen erfolgen, bevor sie im zweiten Schritt mit externen Projektbeteiligten geteilt wird.

## 10 Diskussion

Das Proof of Concept demonstrierte die Möglichkeit, eine Datenbank im Modellierungsprozess von digitalen Bauwerksmodellen einzusetzen. Die Rückmeldungen aus der Praxis haben ergeben, dass dieser Ansatz grundsätzlich in die richtige Richtung geht, jedoch sind einige Fragen zur praktischen Anwendbarkeit offen. Um praktikabel zu werden, muss das Proof of Concept weitere Entwicklungsprozesse durchlaufen. Die Umsetzung erfordert ein kompetentes Personal mit IT-Kenntnissen bei den KMUs. Dies steht jedoch im Widerspruch zu den Wünschen aus den Fachgesprächen, die nach einfachen Lösungen verlangen. Allerdings bietet es den Unternehmen aus der Softwarebranche an, auf Grundlage dieser Arbeit eine Anwendung für das Kundensegment anzubieten.

Die Fachgespräche, die im Rahmen der Studie geführt wurden, lieferten wertvolle Einblicke zur Unternehmensorganisation und den Modellierungsprozess im Unternehmen. Anhand der gestellten Anforderungen dienten sie als Wegweiser für die Entwicklung des Konzepts. Aufgrund der begrenzten Anzahl an Fachgesprächen konnte die Studie nicht das gesamte Spektrum der KMUs abdecken. Daher wird empfohlen, weitere quantitative und qualitative Studien durchzuführen, um ein umfassenderes Verständnis über den aktuellen Trend in der digitalen Bauwerksmodellierung zu erlangen.

Die Entscheidung, SQLite als Datenbanktechnologie zu verwenden, ermöglichte eine einfache Implementierung mit der Programmiersprache Python und bot Interoperabilität mit verschiedenen Systemen. Laut den Rückmeldungen aus der Praxis hätte auch Excel genutzt werden können. Es ist jedoch wichtig zu beachten, dass Excel nicht dieselben Datenkonsistenzmerkmale wie eine Datenbank bietet und Daten nur semi-strukturiert vorliegen. Excel ist ein Tabellenkalkulationsprogramm und keine Datenbank. In diesem Anwendungsfall wird mit strukturierten Daten gearbeitet, die auch in einer strukturierten Umgebung gespeichert werden sollen. Die Benutzerfreundlichkeit sollte separat betrachtet und dabei eine geeignete Benutzeroberfläche entwickelt werden.

Die Mapping-Methode P3 stellte aus Sicht der Datenbankmodellierung einen Mittelweg zwischen den möglichen Alternativen dar. Die Herausforderung dieser Methode wurde jedoch erst bei der theoretischen Erprobung von GUIs deutlich. Es erforderte vermehrt das SQL-Sprachelement «JOIN» und bildet wie gesehen im Kapitel 8.9 komplexere Befehle, um die Abfragen ausführen zu können. Alternativ hätte auch eine Kombination aus P1 und P3 erfolgen können, bei der jede IFC-Entität eine eigene Tabelle bildet und die Relationen zu den IFC-Entitäten als Spalten hinzugefügt werden. Dies wäre jedoch auf Kosten der Datenkonsistenz erfolgt. Eine umfassende Studie unter Einbezug der GUI-Implementierung wäre interessant, um zu untersuchen, inwieweit die Mapping-Methode P3 sinnvoll erscheint und wie möglicherweise einfachere Mapping-Methoden diese Herausforderung lösen könnten.

Ohne die Verwendung von «SPF-ID» als Primärschlüssel kam es zu zahlreichen Ausnahmen, da einige IFC-Entitäten weder über das Attribut «Name» noch über das Attribut «GlobalId» verfügten. Infolgedessen stellte sich die Frage, ob es nicht sinnvoller wäre, die SPF-ID als globalen Primärschlüssel in allen Tabellen im Datenbankmodell zu verwenden. Dies würde insbesondere praktisch sein, da die SPF-ID zwischen dem Stand des IFC-SPF und der Datenbank konsistent bleibt, weil die Datenbank bei jedem Öffnen neu aufgebaut wird.

Die kontinuierliche Initialisierung der Datenbank beim Öffnen vereinfachte die Anwendung, da das IFC-SPF als einzelne Datei aus der Autorensoftware exportiert wird. Es ist jedoch wichtig zu prüfen, ob diese Aktualisierung die Gesamtleistung der Anwendung beeinträchtigen könnte und ob es effizientere Methoden für den Umgang mit dem vollständigen Export des IFC-SPF gibt. Die Auswirkungen des aktuellen Vorgangs bei grossen IFC-SPFs sollten ebenfalls auf die Leistung der Anwendung untersucht werden. Weitere Forschungen in diesem Bereich könnten zusätzliche Erkenntnisse liefern.

## 11 Fazit und Ausblick

Die Modellierung digitaler Bauwerksmodelle ist aufgrund proprietärer Autorensoftware herausfordernd, da sie die Datenverwaltung erschwert und die Beteiligung fachkundiger Personen ohne spezifische Modellierkenntnisse behindert. Als Lösung wird ein Ansatz entwickelt, der die Trennung der Aufgaben im Modellierungsprozess eines KMUs ermöglicht. Dieser umfasst den Einsatz einer geeigneten Datenbanktechnologie und Mechanismen zur synchronen Zusammenarbeit, um die Datenkonsistenz zwischen der Autorensoftware und der Datenbank sicherzustellen. Die interoperable Dateiformat IFC-SPF dient als Brücke zwischen den beiden Systemen.

Studien haben datenbankbasierte Systeme als Alternative zum IFC-SPF zur Speicherung von Gebäudedaten aufgezeigt und die semantische Diskrepanz zwischen den Sprachen SQL und EXPRESS bei der Übersetzung des IFC-Datenmodells in Datenbanken thematisiert. In der Praxis existieren vergleichbare kommerzielle Lösungen, die jedoch für grosse Unternehmen entwickelt wurden. Daher werden zwölf Fachgespräche mit KMUs durchgeführt, um deren Modellierungspraktiken für DBM sowie den Einsatz von Datenbanktechnologien zu untersuchen. Erkennbar ist, dass kleine Unternehmen meist in den geringeren Modellierungsstufen verbleiben, während mittlere Unternehmen zu höheren Stufen zur semantischen Beschreibung der DBM neigen und häufiger eine klare Rollenzuweisung im Modellierungsprozess haben. DBMs werden hauptsächlich anhand von Eigenschaften, Mengen, Stückzahlen und der Klassifizierung von Bauteilen beschrieben. Obwohl die technische Umsetzung der fachlichen Beschreibung bei KMUs unbekannt ist, wird ein mögliches Mapping zu relevanten IFC-Konzepten hergeleitet. Im Modellierungsprozess werden Daten in die Autorensoftware eingegeben, exportiert und weiterverarbeitet. Ein Rückfluss erfolgt selten und manuell. Die meisten Unternehmen verwenden keine Datenbanken, sondern bevorzugen Excel, vermutlich aufgrund fehlenden Know-hows oder begrenzter Ressourcen. Grössere Unternehmen zeigen jedoch zunehmendes Interesse an Datenbanken.

Die Auswahl der geeigneten Datenbanktechnologie berücksichtigt die Anforderungen aus den Fachgesprächen. MySQL und SQLite werden als Optionen identifiziert, wobei SQLite aufgrund seiner Einfachheit und Effizienz bevorzugt wird. Die Systemarchitektur umfasst eine SQLite-Datenbank, eine interoperable Dateiformat IFC-SPF sowie Autodesk Revit als proprietäre Autorensoftware. Zwei Python-Skripte werden in der Anwendung verwendet: Ein Skript fungiert als Middleware zwischen der SQLite-Datenbank, einem Datenbank-Interface (DB Browser) und der IFC-SPF zur Bearbeitung von alphanumerischen Informationen der DBM. Das zweite Skript wird als pyRevit-Add-On in Autodesk Revit eingesetzt, um die alphanumerischen Informationen aus der Datenbank in die Autorensoftware zu übertragen und eine IFC-SPF aus der Autorensoftware zu exportieren. Dabei wird IfcOpenShell verwendet, um die IFC-SPF zu parsen und die Datenbankkommunikation erfolgt über die SQL-Sprache.

Bei der Mapping-Methode vom IFC-Datenmodell in die Datenbank wird ein Mittelweg zwischen Zugriffseffizienz und Datenkonsistenz bevorzugt, weshalb eine «Entitäten-zu-Tabellen» Mapping verwendet wird. Dabei werden nicht abstrakte Entitäten mit eigenen und vererbten Attributen in Tabellen übersetzt, was Datenabfragen erleichtert.

Die Datenkonsistenz wird von verschiedenen Faktoren beeinflusst, darunter die Wahl der Mapping-Methode des IFC-Datenmodells für Datenbanken, das Datenbankdesign und von den Einschränkungen wie Primär- und Fremdschlüssel sowie vom Mapping der Datentypen. Ebenso spielen Transaktionen von Daten, Datenzugriffe auf Dateien und Datenbanken sowie kritische Operationen in den Aktivitäten der Anwendung eine Rolle bei der Entstehung von Dateninkonsistenzen. Um Dateninkonsistenz zu verhindern, werden einerseits systemorientierte Integritätsbedingungen im Datenbankschema mit

Primärschlüssel und Einschränkungen von erlaubten Werten in Datentypen sowie anwendungsorientierte Mechanismen wie die formale Vereinbarung zur Datenpflege für geometrische- und alphanumerische Informationen und diverse technische Mechanismen eingesetzt. Als technische Mechanismen werden eine Sperrung durch die Regelung der Datenhoheit mit einer temporären Datei im IFC-SPF sowie der Einsatz von Zwischenspeichern und das Speichern von Logdateien im Dateisystem für durchgeführte Operationen zur Rückverfolgung bei Fehlern und die Regelung der Datenhoheit auf gemeinsam genutzten Ressourcen verwendet.

Bei der Synchronisation zwischen der Datenbank und der Autorensoftware ist die zeitliche Abfolge von Aktivitäten entscheidend. Dabei ist die zuletzt vorgenommenen Anpassungen am Bauwerksmodell zu identifizieren. Eine geordnete Abfolge von Aktivitäten und die Verwendung der IFC-Entität «IfcApplication» im digitalen Bauwerksmodell zur Bestimmung der letzten bearbeitenden Quelle stellt den Prozess der Synchronisation sicher. Dennoch können Ausnahmesituationen auftreten, welche vertiefter analysiert werden sollten.

Die Machbarkeit des Einsatzes eines relationalen Datenbanksystems im Modellierungsprozess wird durch ein Proof of Concept gezeigt. Dennoch wird festgestellt, dass weitere Entwicklungen erforderlich sind, um die praktische Anwendbarkeit dieses Ansatzes zu gewährleisten. Eine der Herausforderungen besteht darin, dass KMUs möglicherweise nicht über ausreichend kompetentes IT-Personal verfügen, um solche Systeme zu implementieren. Trotz dieser Herausforderungen bietet die Arbeit jedoch eine Chance für Unternehmer aus der Softwareentwicklungsbranche, die Bedürfnisse von KMUs im Modellierungsprozess zu erfüllen.

Zusammenfassend deutet die Arbeit darauf hin, dass weitere Studien erforderlich sind, um den Modellierungsprozess von DBM in der Praxis im grösserem Umfang zu untersuchen und die beobachteten Trends zu bestätigen. Es ist auch zu untersuchen, inwieweit Unternehmen in der Baubranche für die Datenhaltung in Datenbanken sensibilisiert sind und welche Kenntnisse sie darüber besitzen. Das Proof of Concept hat die Machbarkeit nachgewiesen und kann nun als Grundlage dienen, um die Anwendung um eine GUI zu erweitern sowie ihre praktische Anwendung in einem Unternehmen zu evaluieren. Dadurch können die in dieser Master-Thesis getroffenen Entscheidungen bezüglich der Mapping-Methode und der Datenbanktechnologie hinsichtlich ihrer Praktikabilität bewertet werden.

## 12 Literaturverzeichnis

- Bogner, A., Littig, B., Menz, W., 2014. *Interviews mit Experten: Eine praxisorientierte Einführung*. Springer Fachmedien Wiesbaden, Wiesbaden. <https://doi.org/10.1007/978-3-531-19416-5>
- Borrmann, A., König, M., Koch, C., Beetz, J. (Eds.), 2021. *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*, VDI-Buch. Springer Fachmedien Wiesbaden, Wiesbaden. <https://doi.org/10.1007/978-3-658-33361-4>
- Broy, M., Kuhrmann, M., 2021. *Einführung in die Softwaretechnik*, Xpert.press. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-50263-1>
- BuildingSMART International, 2023. IFC String Encoding [WWW Document]. buildingSMART Technical. URL <https://technical.buildingsmart.org/resources/ifcimplementationguidance/string-encoding/> (accessed 4.28.24).
- BuildingSMART International, 2022. IFC Formats [WWW Document]. buildingSMART Technical. URL <https://technical.buildingsmart.org/standards/ifc/ifc-formats/> (accessed 9.17.23).
- BuildingSMART International, 2020. IFC4 Documentation [WWW Document]. URL <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/FINAL/HTML/> (accessed 3.30.24).
- BuildingSMART International (Ed.), n.d. *The EXPRESS Definition Language for IFC Development*.
- Bundesamt für Statistik, 2023. Kleine und mittlere Unternehmen [WWW Document]. URL <https://www.bfs.admin.ch/bfs/de/home/statistiken/industrie-dienstleistungen/unternehmen-beschaeftigte/wirtschaftsstruktur-unternehmen/kmu.html> (accessed 9.22.23).
- Buttgereit, L., Emara, M., 2023. *An updated object-relational approach for database-centered IFC-models*, Ruhr-Universität Bochum. Ruhr-Universität Bochum. <https://doi.org/10.13154/294-10104>
- Digital Construction Management, 2023. *Digitalisierung der Baubranche*. Digital Construction Management.
- Dion Moul, 2023. IFC stored as SQLite and MySQL [WWW Document]. OSArch. URL <https://community.osarch.org/discussion/1535/ifc-stored-as-sqlite-and-mysql> (accessed 9.1.23).
- DiRoots, n.d. *Revit to Excel and Google Sheets | Import and Export | SheetLink Add-in*. DiRoots. URL <https://diroots.com/revit-plugins/revit-to-excel-sheetlink/> (accessed 5.1.24).
- dRofus Software [WWW Document], 2023. URL <https://www.nemetschek.com/de/marke/drofus> (accessed 10.4.23).
- Frédéric Beaupère (ERNE Holzbau AG), 2021. Monthly Meetup #20: Saturday 13th of November @ 20:00 UTC [WWW Document]. OSArch. URL <https://community.osarch.org/discussion/794/monthly-meetup-20-saturday-13th-of-november-20-00-utc> (accessed 9.16.23).
- GeeksforGeeks, 2023. How to Build a Web App using Flask and SQLite in Python [WWW Document]. GeeksforGeeks. URL <https://www.geeksforgeeks.org/how-to-build-a-web-app-using-flask-and-sqlite-in-python/> (accessed 4.4.24).
- Helfferich, C., 2011. *Die Qualität qualitativer Daten: Manual für die Durchführung qualitativer Interviews*, 4. Aufl. ed. VS, Verl. für Sozialwiss, Wiesbaden.
- ISO 10303-21, 2023. . Wikipedia.
- Jablonski, S., Ruf, T., 1991. *Datenkonsistenz in verteilten Systemen / Data Consistency in Distributed Systems*. it - Information Technology 33, 175–184. <https://doi.org/10.1524/itit.1991.33.4.175>
- Krcmar, H., 2015. *Informationsmanagement*. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-45863-1>
- Lee, G., Jeong, J., Won, J., Cho, C., You, S., Ham, S., Kang, H., 2014. *Query Performance of the IFC Model Server Using an Object-Relational Database Approach and a Traditional Relational Database Approach*. J. Comput. Civ. Eng. 28, 210–222. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000256](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000256)
- Postle, B., Moul, D., 2022. *Native IFC*, Unpublished. <https://doi.org/10.13140/RG.2.2.15614.56648>
- PricewaterhouseCoopers GmbH (Ed.), 2019. *Digitalisierung der deutschen Bauindustrie*.
- Rosenthaler, C., Schildknecht, L., 2010. *Objektorientierte Modellierung von Strasseninformationen*.
- Runebook.dev, 2022. SQLite ist serverlos [WWW Document]. SQLite ist serverlos. URL <https://runebook.dev/de/docs/sqlite/serverless> (accessed 12.16.23).
- Schicker, E., 2017. *Datenbanken und SQL*, Informatik & Praxis. Springer Fachmedien Wiesbaden, Wiesbaden. <https://doi.org/10.1007/978-3-658-16129-3>

- Schumann-Giesler, U., 1997. *Datenkonsistenz bei heterogener Datenspeicherung*. Deutscher Universitätsverlag, Wiesbaden. <https://doi.org/10.1007/978-3-663-08344-3>
- SIA 2051, 2017. *Building Information Modelling (BIM) - Grundlagen zur Anwendung der BIM-Methode*. solid IT, 2023. historical trend of relational DBMS popularity [WWW Document]. URL [https://db-engines.com/en/ranking\\_trend/relational+dbms](https://db-engines.com/en/ranking_trend/relational+dbms) (accessed 10.18.23).
- SQLite, 2024a. SQLite Home Page [WWW Document]. URL <https://www.sqlite.org/index.html> (accessed 5.23.24).
- SQLite, 2024b. Appropriate Uses For SQLite [WWW Document]. Appropriate Uses For SQLite. URL <https://sqlite.org/whentouse.html> (accessed 4.13.24).
- SQLite, 2023. SQLite Frequently Asked Questions [WWW Document]. URL <https://www.sqlite.org/faq.html#q5> (accessed 3.30.24).
- SQLite, 2022. Datatypes In SQLite [WWW Document]. URL <https://sqlite.org/datatype3.html> (accessed 3.21.24).
- Starke, G., Hruschka, P., 2011. *Software-Architektur kompakt: angemessen und zielorientiert*, 2. Aufl. ed, IT kompakt. Spektrum Akademischer Verlag, Heidelberg.
- Suhr, J., 1999. *The choosing by advantages decisionmaking system*. Quorum, Westport, Conn.
- Wei, Q., Duan, G., Cai, J., Zhou, G., 2009. *Mapping Technique of STEP Data Model in Relational Database Based on Data Storage*, IEEE, in: 2009 Second International Conference on Intelligent Networks and Intelligent Systems. Presented at the 2009 Second International Conference on Intelligent Networks and Intelligent Systems (ICINIS), IEEE, Tianjian, China, pp. 657–660. <https://doi.org/10.1109/ICINIS.2009.170>
- You, S.-J., Yang, D., 2004. *Relational Db Implementation of Step Based Product Model*.
- Zhu, Z., Shide, K., 2021. *The study of Building Information Model server focusing on IFC to RDB conversion*. Architectural Informatics Society 22.

## **13 Anhang**

**13.1 Anhang A: Literaturrecherche – Begriffstabellen**

**13.2 Anhang B: Datenerhebung – Interview-Leitfaden**

**13.3 Anhang C: Datenerhebung – Transkription der Fachgespräche**

**13.4 Anhang D: Datenerhebung – Anforderungen aus den Fachgesprächen**

**13.5 Anhang E: Entscheidungsmatrix RDBMS**

**13.6 Anhang F: Proof of Concept – Quellcode**