

Diplomarbeit

Datenstrukturwerkzeuge im openBIM-Prozess – Prozessoptimierung – Fallstudie mit Allplan und BIMQ

ausgeführt zum Zwecke der Erlangung des akademischen Grads

Diplom-Ingenieur

eingereicht an der TU Wien, Fakultät für Bau- und Umweltingenieurwesen

Diploma Thesis

Data structure tools in the openBIM process – Process optimisation – case study with Allplan and BIMQ

submitted in satisfaction of the requirements for the degree

Diplom-Ingenieur

of the TU Wien, Faculty of Civil and Environmental Engineering

Klemens Huber, BSc

Matr.Nr.: 01426016

Betreuung: Associate Prof. Dipl.-Ing. Dr.techn. **Christian Schranz**, M.Sc.
Univ.Ass. Dipl.-Ing. Dr.techn. **Harald Urban**, BSc
Forschungsbereich Digitaler Bauprozess
Technische Universität Wien
Karlsplatz 13/235-03, 1040 Wien, Österreich

Wien, im Mai 2023

Kurzfassung

Im Rahmen eines Bauprozesses ist eine klare Kommunikation zwischen dem Auftraggeber und dem Projektteam von entscheidender Bedeutung, um die Projektziele erfolgreich zu erreichen. Insbesondere bei Projekten, die mit der Methode Building Information Modeling (BIM) umgesetzt werden, werden die Ziele des Auftraggebers und die Struktur der Bearbeitung in den sogenannten BIM-Regelwerken festgelegt. Dazu gehören die Auftraggeber-Informationsanforderungen (AIA) und der BIM-Projektentwicklungsplan (BAP).

Die AIA legen Vorgaben zur Detaillierung fest, von denen die erforderlichen Informationen abgeleitet werden. Die alphanumerischen Informationen sind durch den *Level of Information (LOI)* bestimmt, der konkret durch die erforderlichen *Property Sets* (Merkmallisten, *Psets*) und deren *Properties* (Merkmale, *Properties*) abgebildet wird. Diese Anforderungen werden oft in Form von umfangreichen Tabellen dem AIA-Dokument beigefügt und müssen vom BIM-Autor in der entsprechenden BIM-Software umgesetzt werden.

Datenstrukturwerkzeuge bieten in diesem Zusammenhang die Möglichkeit, die *Psets* und *Properties* zu verwalten und die zusammengestellten Strukturen und *LOI*-Anforderungen als direkt importierbare Vorlage in die BIM-Autorensoftware auszugeben. Auf diese Weise kann der Bauprozess effizienter gestaltet werden.

In dieser Diplomarbeit werden die Funktionalitäten von Datenstrukturwerkzeugen gezeigt und der bestehende Prozessablauf bei der Verwendung eines Datenstrukturwerkzeuges dargestellt. Die Entwicklung eines optimierten Prozesses basiert auf dem Ziel der Verbesserung der Datensicherheit und Effizienz (Einfachheit und Flexibilität) der strukturierten Daten. In einem definierten Anwendungsfall wird gezeigt, wie eine Standardkonfiguration an Informationsanforderungen erstellt werden kann. Am Beispiel *Allplan* ist dargestellt, wie die Interaktion zwischen Datenstrukturwerkzeug und Autorensoftware funktioniert. Die praktische Umsetzung des optimierten Prozesses wird in einer Fallstudie mit den Softwareprodukten *Allplan* und *BIMQ* evaluiert. Mithilfe der durchgeführten qualitativen und quantitativen Tests wird ersichtlich, ob die exportierte IFC-Datei mit der ursprünglichen Vorlage der Informationsanforderungen übereinstimmt, und wo es zu Abweichungen kommt.

Abstract

As part of a construction process, clear communication between the client and the project team is of vital importance in order to successfully achieve the project goals. Especially in projects that are executed by using the Building Information Modeling (BIM) method, the client's goals and the structure of the processing are defined by different types of information requirements. These include the Exchange Information Requirements (EIR) and the BIM-Execution Plan (BEP).

The EIR specify the levels of detail from which the necessary information is derived. The alphanumeric information is defined by the *Level of Information (LOI)*, which is specifically mapped by the required *Property Sets (Psets)* and its *Properties*. These requirements are often attached to the EIR document in the form of multi-page tables and must be implemented by the BIM-author in the BIM-software.

In this regard, data structure tools offer the possibility of managing *Psets* and *Properties* and exporting the compiled structures and *LOI*-requirements as templates that can be imported directly into the authoring software. In this way the construction process can be made more efficient.

In this diploma thesis, the functionalities of data structure tools and the existing process when using them is presented. The development of an optimized process is based on the goal of improving the data security and efficiency (simplicity and flexibility) of the structured data. For a defined use case it is shown how a standard configuration of information requirements can be created. The interaction between data structure tools and the authoring software is shown by the example of *Allplan*. The practical implementation of the optimized process is evaluated in a case study with the software products *Allplan* and *BIMQ*. The executed qualitative and quantitative tests show whether the exported IFC file matches the original template of the information requirements and where there are deviations.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Inhaltsverzeichnis

Kurzfassung	3
Abstract	5
Abkürzungen	9
1 Einleitung	11
1.1 Stand der Forschung und Forschungsfragen	11
1.2 Forschungsmethodik	12
1.3 Forschungsabgrenzung	13
2 Grundlagen	14
2.1 <i>Level of Information Need (LOIN)</i>	14
2.2 <i>Level of Geometry (LOG)</i> und <i>Level of Information (LOI)</i>	14
2.3 Auftraggeber-Informationsanforderungen (AIA), BIM-Projektentwicklungsplan (BAP)	15
2.4 BIM-Leistungsbilder und Projektphasen	15
2.5 <i>Properties</i> und <i>Property Sets (Psets)</i>	16
2.6 <i>Allplan</i>	16
2.7 Notationen in dieser Arbeit	17
3 Ausgangslage	18
3.1 Möglichkeiten zur Implementierung von Informationsanforderungen	18
3.1.1 Implementierung von Informationsanforderungen in <i>Allplan</i>	18
3.2 Datenstrukturwerkzeuge	20
3.2.1 Datenstrukturwerkzeuge im openBIM-Prozess	20
3.2.2 Arten von Datenstrukturwerkzeugen	20
3.3 Anforderungen an ein Datenstrukturwerkzeug	22
3.4 Definition des Anwendungsfalls	23
4 Prozessablauf bei der Verwendung eines Datenstrukturwerkzeuges	25
4.1 Beschreibung des bestehenden Prozesses bei Verwendung eines Datenstrukturwerkzeuges	25
4.2 Verbesserungspunkte beim bestehenden Prozess	28
4.3 Beschreibung des optimierten Prozesses	28
4.4 Vorteile durch den optimierten Prozess	29
5 Fallstudie	30
5.1 Prozessablauf bei der Verwendung von <i>BIMQ</i> und <i>Allplan</i>	30
5.2 Prozessschritte in <i>BIMQ</i>	32
5.2.1 Beschreibung von <i>BIMQ</i>	32
5.2.2 <i>Property-Silo</i> erstellen	33
5.2.3 <i>Pset-Silo</i> erstellen	46
5.2.4 <i>Properties</i> mit <i>Psets</i> verknüpfen	48

5.2.5	<i>Entities</i> erstellen	48
5.2.6	Mapping hinzufügen	49
5.2.7	Konfektionssilo mit <i>Entities</i> verknüpfen	49
5.2.8	Projektanforderung der Projektphase und Anwendungsfall zuweisen	51
5.2.9	Softwarevorlagen exportieren	51
5.3	Prozessschritte in <i>Allplan</i>	53
5.3.1	Softwarevorlage dem Projekt zuweisen	54
5.3.2	Modellierung	54
5.3.3	IFC-Datei exportieren	56
5.4	Formale Prüfung	57
5.5	Durchführung von Versuchen	58
5.5.1	Allgemeiner Test der Funktionstüchtigkeit der Prozessschritte	58
5.5.2	Qualitative und quantitative Tests zu <i>Properties</i> in <i>Allplan</i>	65
5.5.3	Tests von weiteren <i>Entities</i> und <i>PredefinedTypes</i>	70
6	Diskussion, Empfehlungen und Ausblick	78
6.1	Diskussion der durchgeführten Tests	78
6.2	Beantwortung der Forschungsfragen	79
6.3	Anmerkungen zur IFC-Spezifikation	79
6.4	<i>BIMQ</i> -Verbesserungen	80
6.5	<i>Bimplus</i> -Verbesserungen	81
6.6	<i>Allplan</i> -Verbesserungen	81
6.7	Ausblick	82
	Literaturverzeichnis	83
	A Anhang	86

Abkürzungen

AIA Auftraggeber-Informationsanforderungen

BAP BIM-Projektentwicklungsplan

BIM Building Information Modeling

BPMN Business Process Model and Notation

bSAT *buildingSMART Austria*

bSDD *buildingSMART Data Dictionary*

bSI *buildingSMART International*

CDE *Common Data Environment* (Kollaborationsplattform)

Entity *Entity* (Klasse, auch Entität)

GUID *Globally Unique Identifier*

IFC *Industry Foundation Classes*

LOG *Level of Geometry*

LOI *Level of Information*

LOIN *Level of Information Need* (Informationsbedarfstiefe)

Property *Property* (Merkmal)

Pset *Property Set* (Merkmalliste)

Kapitel 1

Einleitung

Die Auftraggeber-Informationsanforderungen (AIA) als BIM-Regelwerk legen Vorgaben zur Detaillierung fest, von denen die Informationsanforderungen abgeleitet werden. In der Literatur wird betont, dass textbasierte Dokumente die am häufigsten verwendete Methode zur Spezifizierung dieser Informationsanforderungen sind, wie Tomczak et al. [28] oder Mellenthin Filardo [19] beschreiben. Allerdings erwähnen Eichler et al. [8, S. 69] Datenstrukturwerkzeuge als eine Alternative in diesem Kontext:

„Die IFC-Datenstruktur erlaubt, in Ergänzung zu den bereits vorhandenen Vorgaben, die Definition individueller Ergänzungen. Diese können projektspezifisch in einem lokalen Rahmen definiert werden (bspw. mit einem Datenstrukturwerkzeug) und werden über BIM-Regelwerke an das Projektteam (mittels des AIA) kommuniziert bzw. (mittels des BAP) konsolidiert. [...]“

Das Definieren der Informationsanforderungen ist nach den BIM-Leistungsbildern von Eichler et al. [11] der Organisationseinheit *BIM-Management* oder der *BIM-Projektleitung* zugewiesen. Die Anforderungen in die BIM-Autorensoftware zu integrieren, liegt in der Sphäre des Auftragnehmers. Eichler [9, S. 12] schreibt dem Leistungsbild *BIM-Erstellung* die Leistung *„Umsetzung der Vorgaben aus dem BAP und Aufbau der entsprechend notwendigen Konfigurationen in der BIM-Autorensoftware“* zu. Ein Datenstrukturwerkzeug kann beim Definieren der Anforderung und beim Integrieren in die Autorensoftware unterstützen und könnte somit für mehrere BIM-Organisationseinheiten hilfreich sein.

1.1 Stand der Forschung und Forschungsfragen

Beim openBIM-Prozess ist, aufgrund der digitalen Zusammenarbeit in offenen Datenformaten, der Datenaustausch eines der wesentlichen Themen. Zu den teilweise standardisierten Methoden zur Spezifizierung von Informationsanforderungen wurde bereits Tomczak et al. [28] erwähnt. Tomczak et al. [28] vergleichen in bestimmten Punkten die Methoden *Spreadsheet*, *Product Data Templates*, *Data Dictionary*, *IDS*, *mvdXML*, *idmXML*, *LOIN*, *IfcPropertyTemplate* und *Linked Data + Shaps Constraint Language*. Als die aktuell am häufigsten verwendete Methode werden jedoch textbasierte Dokumente genannt. Textbasierte Dokumente sind jedoch nur schwer maschinenlesbar und die Übernahme der Informationsanforderungen in die BIM-Autorensoftware ist einzeln und manuell durchzuführen. Dies stellt einen sehr großen Arbeitsaufwand dar.

Datenstrukturwerkzeuge sind eine Entwicklung die immer häufiger genutzt wird. Aktuell gibt es jedoch noch wenige wissenschaftliche Publikationen, die Datenstrukturwerkzeuge oder Informationsmanagementsysteme behandeln. Erwähnt werden einzelne Datenstrukturwerkzeuge und konkret *BIMQ* von Mellenthin et al. [20]. Hier wird *BIMQ* als ein kommerzieller Dienst für die Definition von Informationsanforderungen beschrieben und der Export in Excel-Tabellen und spezifische Dateien für BIM-Autorensoftware erwähnt. Eichler et al. [8] beschreiben Datenstrukturwerkzeuge im BIMcert Handbuch. Sie ermöglichen das Erstellen und Bearbeiten von individuellen

Datenstrukturen, wie den *Property* (Merkmal, *Property*) und *Property Set* (Merkmalliste, *Pset*), die die alphanumerischen Informationsanforderungen abbilden. Die Datenstrukturen sind in einer Datenbank gespeichert, was die Basis für die Ausgabe der Informationsanforderungen in verschiedene Kanäle und Formate bildet. Üblicherweise ist als erste Option die Ausgabe im PDF-Format für den Anhang der AIA möglich. Weiter Kanäle können direkte Schnittstellen zu BIM-Autorensoftware oder Prüfsoftware sein. Eichler et al. [8] führt als Vorteile beispielsweise den geringeren Anpassungsaufwand oder das verbesserte Qualitätsmanagement an.

Von Tomczak et al. [28] werden Datenstrukturwerkzeuge als *proprietary software solutions* bezeichnet und einzelne kritische Aspekte angeführt. Es wird hervorgehoben, dass der Nachteil der vermeintlichen Flexibilität in der Definition von Workflows und Funktionen darin liegt, dass die Inhalte an das geistige Eigentum des Software-Anbieters gebunden sind. Da es sich um sogenannte unfreie Software handelt und der Code nicht einsehbar ist, ist die *validation logic* nicht nachvollziehbar. Das bedeutet, dass für den Nutzer nicht bekannt ist, wie weit ungültige Eingabe zulässig sind und ob diese erkannt werden.

Für die BIM-Autorensoftware *Allplan* existiert eine Dokumentation von *Allplan* [21], in denen die Prozesse in *Allplan* im Kontext von BIM beschrieben werden. Hier sind ebenfalls Angaben zur Handhabung von Informationsanforderungen in Form der *Psets* und *Properties* zu finden.

Forschungsfragen

Zu Datenstrukturwerkzeugen und deren Integrierung im openBIM-Prozess gibt es folglich noch Forschungsbedarf. In dieser Arbeit gilt es als Zielsetzung folgende Forschungsfragen zu beantworten:

1. Wie sieht die Integrierung eines Datenstrukturwerkzeuges im openBIM-Prozess aus?
2. Wie kann, innerhalb eines Anwendungsfalls, eine Standardkonfiguration mittels eines Datenstrukturwerkzeuges erstellt und in die BIM-Autorensoftware *Allplan* importiert werden?
3. Welche Herausforderungen entstehen bei der Implementierung eines Datenstrukturwerkzeuges in einem konkreten Fallbeispiel?

1.2 Forschungsmethodik

Um eine aussagekräftige und valide Untersuchung durchzuführen, ist es von entscheidender Bedeutung, eine angemessene Forschungsmethodik zu wählen und anzuwenden. In diesem Kapitel wird die Forschungsmethodik erläutert.

Zu Beginn wird eine Literaturrecherche durchgeführt, die das Ziel hat Information zur Ausgangslage und den Anforderungen an ein Datenstrukturwerkzeug zu gewinnen. Anhand dieser Informationen wird ein Anwendungsfall (engl. *use case*) festgelegt und darauf aufbauend der Prozessablauf für das Aufsetzen eines Datenstrukturwerkzeug und der Autorensoftware *Allplan* beschrieben. Eine Fallstudie¹ mit Versuchen dient der Evaluierung des Prozessablaufes. Zum Schluss stehen die Diskussion der Forschungsergebnisse und Empfehlungen zur Verbesserung von *BIMQ*, *Bimplus* und *Allplan*. Die Forschungsmethodik ist in Abbildung 1.1 dargestellt.

¹Die Herangehensweise bei der Fallstudie gleicht einer möglichen Herangehensweise von BIM-Anwendern in Unternehmen.

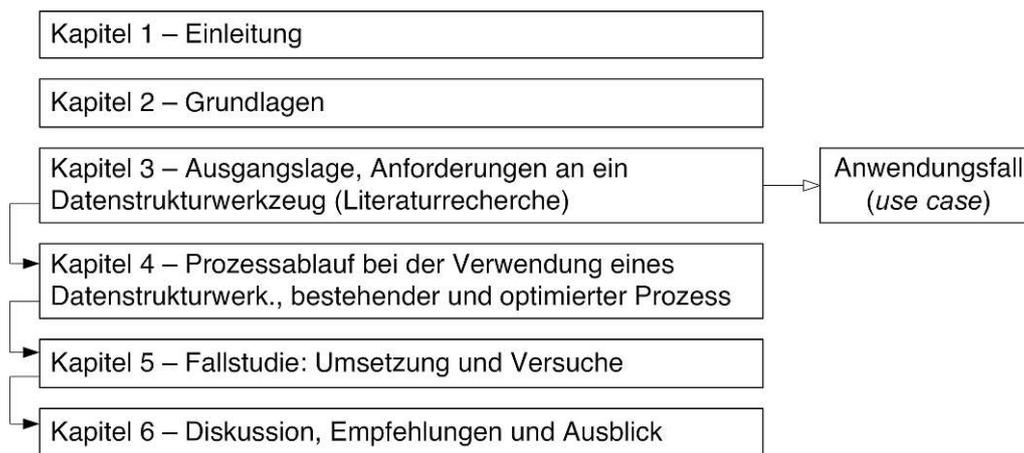


Abb. 1.1: Forschungsmethodik

1.3 Forschungsabgrenzung

Bei der durchgeführten Fallstudie liegt der Fokus auf der Funktionstüchtigkeit des Workflows zwischen *BIMQ* und *Allplan*. Auf die Leistungsbilder und Projektphasen wird daher nicht detailliert eingegangen. Die durchgeführten Versuche konzentrieren sich auf die im Hochbau häufig genutzten *Entities* (Klassen, geometrische Elemente) der Klasse *IfcBuiltElement*. Diese sind beispielsweise *IfcColumn*, *IfcBeam* oder *IfcSlab*, die die geometrischen Elemente der Stützen, Trägern und Decken abbilden. Als Quelle der Daten für *Entities*, *Psets* und *Properties*, gilt die *IFC Specifications* der Version 4.3.1.0. Bei den verwendeten Software-Produkten werden die aktuellsten Versionen genutzt:

- *BIMQ* Version 2.7.7-4
- *Allplan* Version 2023-0-4

Kapitel 2

Grundlagen

Für ein besseres Verständnis werden im Folgenden einzelne Fachbegriffe erklärt, die in dieser Arbeit wiederholt verwendet werden.

2.1 *Level of Information Need (LOIN)*

Das Informationsmanagement in BIM wird unter anderem mit der ISO-Norm ISO 19650-1 [18] standardisiert, siehe Eichler et al. [8]. Hier wird allgemein bestimmt, dass der *Level of Information Need* (Informationsbedarfstiefe, *LOIN*) jeder zu liefernden Information entsprechend ihrem Anwendungsziel¹ festgelegt werden soll. Die Mindestmenge an Informationen, die zur Erfüllung jeder relevanten Anforderung erforderlich ist, hat den *LOIN* zu bestimmen: „*Anything beyond this minimum is considered as waste.*“ [18, S.23]. Die deutsche Bezeichnung für *LOIN* lautet nach ÖNORM EN 17412-1 [23] *Informationsbedarfstiefe* und wird dort wie folgt definiert:

„Vorgabe, die den Umfang und die Tiefe der Informationen (3.11) [Daten mit Bedeutung] definiert“ [23, S. 6]

Die Festlegung der Informationsbedarfstiefe und -bereitstellung erfordert die Berücksichtigung bestimmter notwendiger Bedingungen, wie in ÖNORM EN 17412-1 [23] weiters erläutert wird. Dazu gehören die Anwendungsziele, die Meilensteine für die Informationsbereitstellung, die Akteure, die die Information benötigen, und die, die sie bereitstellen werden, sowie die Objekte, die in einer oder mehreren Gliederungsstrukturen organisiert sind. Diese Bedingungen beeinflussen die Definition der Informationsbedarfstiefe, sind jedoch nicht explizit in dieser enthalten. Eine weitere Beschreibung der Informationsbedarfstiefe lautet:

„Die Informationsbedarfstiefe ist der Rahmen, der den Umfang und die Granularität der auszutauschenden Information festlegt. In Übereinstimmung mit dieser Norm sollte die Informationsbedarfstiefe durch verschiedene Konzepte beschrieben werden: geometrische Informationen, alphanumerische Informationen und Dokumentation.“ [23, S.10]

Die genannten Konzepte werden meistens über die Detaillierungsgrade *Level of Geometry (LOG)* und *Level of Information (LOI)* umgesetzt.

2.2 *Level of Geometry (LOG) und Level of Information (LOI)*

Der *Level of Geometry (LOG)* und der *Level of Information (LOI)* zählen zu den sogenannten Detaillierungsgrade. Der *LOG* ist ein Maß für die geometrische Detaillierungstiefe der dreidimensionalen Bauteile im Modell. Nach demselben Prinzip steht der *LOI* für die Anzahl und den Umfang der verknüpften alphanumerischen Informationen. Sie werden verwendet, um die

¹Anwendungsziel ist ein Begriff aus ÖNORM EN 17412-1 [23].

Anforderungen an die jeweiligen Informationen abzubilden. In ÖNORM A 6241-2 [22] wird in der Definition des *LOI* zusätzlich dessen Abhängigkeit von der jeweiligen Projektphase verankert. Eichler et al. [8] beschreiben, dass *LOG* und *LOI* maßgebend für den technischen Teil in den BIM-Regelwerken AIA und BAP sind:

„Durch sie [LOG und LOI, Anm.] werden die konkreten Informations-Anforderungen des AG definiert, die durch die Projektbeteiligten umgesetzt werden müssen. Der AIA enthält dabei den Grundstamm an Informations-Anforderungen, die im BAP projektbezogen modifiziert werden können. Die Inhalte beider Detaillierungsgrade sollten zu Projektbeginn genau betrachtet und hinsichtlich der Umsetzbarkeit und Verantwortlichkeit besprochen werden.“ [8, S. 99]

2.3 Auftraggeber-Informationsanforderungen (AIA), BIM-Projektentwicklungsplan (BAP)

Die Auftraggeber-Informationsanforderungen (AIA) und der BIM-Projektentwicklungsplan (BAP) zählen zu den BIM-Regelwerken, in denen *„[...] die relevanten Ziele der AG, die Anforderungen an die Projektbeteiligten und die Vorgehensweise für eine erfolgreiche Umsetzung dieser Anforderungen“* [8, S. 42] festgelegt werden. Sie verankern die Informationsbedürfnisse des Auftraggebers, die die Auftragnehmer als Informationsanforderungen zu erfüllen haben. Die Inhalte der AIA stellen die Grundlage für den BAP dar:

„Der BAP ist ein Richtliniendokument, das die Grundlage einer BIM-basierten Zusammenarbeit definiert. [...] Der BAP legt weiterhin die projektbezogenen Ausprägungen fest und definiert das Maß der Informations- und Detaillierungstiefe und deren Qualitäten.“ [8, S. 43]

Der BAP wird während des gesamten Projektes kontinuierlich angepasst und erweitert.

2.4 BIM-Leistungsbilder und Projektphasen

In BIM-Leistungsbildern werden BIM-spezifische Leistungen bzw. Aufgaben den jeweiligen BIM-Organisationseinheiten zugewiesen. Eichler et al. [8, S. 39] beschreiben: *„Ein Leistungsbild beinhaltet immer die Einordnung der jeweiligen Organisationseinheit in das Gesamtgefüge, die Beschreibung der allgemeinen und projektphasenübergreifenden Leistungen sowie die konkreten projektphasenbezogenen Leistungen.“* Die BIM-Organisationseinheiten sind die BIM-Projektleitung, BIM-Projektsteuerung, BIM-Management, BIM-Gesamtkoordination, BIM-Fachkoordination und BIM-Erstellung. Die Organisationseinheit des BIM-Managements stellt die Zusammenführung der BIM-Projektleitung und BIM-Projektsteuerung dar.

Die einzelnen BIM-Leistungsbilder LM.BIM 2022 [11] von *buildingSMART Austria (bSAT)* werden folgenden Projektphasen zugewiesen: Initiierung, Planung, Beschaffung, Bauphase und Übergabe. In Abschnitt 2.2 wurde erwähnt, dass die Detaillierungsgrade *LOG* und *LOI* in Abhängigkeit zur jeweiligen Projektphase stehen. Dementsprechend gibt es je Projektphase unterschiedliche Anforderungen an die geometrischen (*LOG*) und alphanumerischen Informationen (*LOI*). Nach Eichler et al. [8] erhöht sich der *LOI* von Projektphase zu Projektphase, wohingegen der *LOG* schwanken kann.

2.5 Properties und Property Sets (Psets)

In ÖNORM A 6241-2 [22] wird ein Merkmal (*Property*) definiert als: „*alphanumerisches Element, bestehend aus einer immer gleich bleibenden Bezeichnung und variablen Inhalt [...]*“ [22, S. 5]. Von Eichler et al. [8] wird ein *Property Set (Pset)* beschrieben als „[...] ein Container, der *Properties* in einer *Eigenschaftsbaumstruktur* enthält.“ [8, S. 69] Für die technische Definition eines *Property* ist das Datenschema von IFC zu betrachten. Hier ist das Schemata *IfcPropertyResource* in dem *Resource Layer* enthalten. Dies bedeutet, dass es „[...] von mindestens einer Klasse eines der anderen drei Layer referenziert werden [muss]“ [8, S. 72] und nicht als eigenständiges Element existieren kann. In der Dokumentation zur *IFC Specification* [16] wird zu *IfcPropertyResource* erklärt, dass die hierdurch definierten *Property Types* durch *IfcPropertySet* mit Elementen (Subklassen von *IfcObject*) verknüpft werden.

Eichler et al. [8] erläutern weiters, dass die Definitionen der *Properties* mit den Subklassen von *IfcProperty* formuliert werden können. Diese Subklassen sind *IfcSimpleProperty* und *IfcComplexProperty*, welche für einzelne *Properties* oder solche die eine Liste an *Properties* abbilden, vorgesehen sind. Für die Definition eines einzelnen *Property* sind die Klasse *IfcSimpleProperty* und wiederum deren Subklassen relevant. Die davon am häufigsten genutzte ist *IfcPropertySingleValue*. Hier wird einem *Property* ein einzelner alphanumerischer Wert zugewiesen. Die essenzielle Angabe bei einem *Property* der Klasse *IfcPropertySingleValue* ist der *Name*. Optional sind weiters eine *Specification* (Beschreibung), der *NominalValue* (*Data Type* und Wert) und *Unit* (Einheit). Ein Beispiel ist das *Property ReinforcementVolumeRatio* (Bewehrungsgrad Volumen) mit einem *NominalValue* von z. B. 200, dem *Data Type* *IfcMassDensityMeasure* und der möglichen *Unit* kg/m³.

2.6 Allplan

Allplan ist ein Softwareprodukt der Nemetschek AG und kann als BIM-Autorensoftware für die Modellierung verwendet werden. Abbildung 2.1 zeigt, dass *Allplan* aktuell (29.04.2023) für *Architectural Reference Exchange* in der *Major Version* IFC4 und hier für den Export zertifiziert ist. Die Zertifizierung wurde von *buildingSMART International* [6] durchgeführt. *Allplan* bietet in der aktuellen Version 2023 Funktionen für den Export in IFC4.3.

IFC Certified Software

IFC Certification Participants

For the market to benefit the most from IFC, there must be a robust implementation in software available to users in their respective regions and marketplaces. buildingSMART International provides an ongoing platform and process to certify applications for IFC2x3 Coordination View 2.0 and IFC4 Reference View 1.2.

The table below is automatically updated on a daily basis from the [b-cert platform](#).

Search: Allplan

Vendor	Product	Schema	Exchange Requirement	Import / Export	Status	Started	Completed	Report (link)
Allplan GmbH	Allplan	IFC4	Architectural Reference Exchange	Export	Finished	2020-03-13	2022-05-06	
NEMETSCHKE Allplan GmbH	Allplan	IFC 2x3	CV 2.0	Import	Finished	2010-04-27	2014-05-07	https://ifc2x3.b-cert.org/ords/ifc/certification/getCertificationReport/341
NEMETSCHKE Allplan GmbH	Allplan	IFC 2x3	CV2.0-Arch	Export	Finished	2010-04-27	2013-04-16	https://ifc2x3.b-cert.org/ords/ifc/certification/getCertificationReport/102
Allplan GmbH	Allplan	IFC4	Architectural Reference Exchange	Import	In Progress	2020-03-13		

Abb. 2.1: *Allplan* IFC-Zertifizierungsstatus von *buildingSMART International* in [6]

2.7 Notationen in dieser Arbeit

Für eine klare Zuordnung gleicher Wörter mit unterschiedlicher fachlicher Verwendung werden folgende Hinweise gegeben und Festlegungen für diese Arbeit getroffen. Ein *Property* in der IFC-Datenstruktur ist laut Eichler et al. [8] eine dynamische Informationseinheit die vom BIM-Autor in der BIM-Autorensoftware (z. B. *Allplan*) befüllt wird und somit änderbar ist. In der Software *Allplan* werden diese Informationseinheiten *Attribute* genannt. Ein Attribut ist andererseits in der IFC-Datenstruktur als eine statische Informationseinheit, die von der Software selbst erstellt wird und nicht änderbar ist, definiert. In dieser Arbeit steht der Begriff Attribut für das Pendant zum *Property* in *Allplan*, wenn nicht explizit anders angegeben.

Der Begriff *Struktur* wird im Kontext eines Datenstrukturwerkzeuges verwendet. Er steht für die Tabellen-förmige Auflistung unterschiedlicher Einträge, wie *Entities*, *Psets* oder *Properties*.

Fachbegriffe in englischer Sprache, Bezeichnungen von Produkten oder Organisationen sowie Software-spezifische Bezeichnungen sind in Kursivschrift hervorgehoben.

Kapitel 3

Ausgangslage

In diesem Kapitel werden zuerst die Möglichkeiten zur Implementierung von Informationsanforderungen in eine BIM-Autorensoftware beschrieben. Danach wird anhand einer Recherche ermittelt, welche Datenstrukturwerkzeuge am Markt verfügbar sind und welche Funktionen diese abdecken. Außerdem werden die Anforderungen an ein Datenstrukturwerkzeug sowie der in dieser Arbeit weiter behandelte Anwendungsfall formuliert.

3.1 Möglichkeiten zur Implementierung von Informationsanforderungen in eine BIM-Autorensoftware

Für die Implementierung der Informationsanforderungen, beispielsweise aus den AIA, in eine BIM-Autorensoftware existieren im Allgemeinen zwei Möglichkeiten:

- Manuelle Eingabe direkt in der BIM-Autorensoftware
- Spezifischer Import je nach Möglichkeit der BIM-Autorensoftware und die mögliche Nutzung eines Datenstrukturwerkzeuges

Bei der manuellen Eingabe werden in der BIM-Autorensoftware während der Modellierung die festgelegten Anforderungen in Form von *Psets* und *Properties* eingegeben. Die zweite Möglichkeit ist der Software-spezifische Import von Vorlagen der Anforderungen. Mellenthin et al. [20] beschreiben, dass diese Vorlage bei der BIM-Autorensoftware *Autodesk Revit* in Form einer *txt*-Datei und bei *ArchiCAD* in Form einer *xml*-Datei importiert wird. Der Import einer solchen Vorlage in *Allplan* ist folgend beschrieben.

3.1.1 Implementierung von Informationsanforderungen in *Allplan*

In *Allplan* wird eine Vorlage der Informationsanforderungen *Attributsetvorlage* genannt. Diese ist im Allgemeinen in den *Projekteinstellungen* festzulegen, siehe das linke Fenster in Abbildung 3.1. Hier steht einerseits die Option *Bimplus* zur Verfügung, um eine in der zugehörigen *Common Data Environment* (Kollaborationsplattform, *CDE*) *Bimplus* vorhandene Vorlage zu importieren. Um als Quelle der Vorlage ein Datenstrukturwerkzeug zu nutzen, wird beispielsweise beim Datenstrukturwerkzeug *BIMQ* die Vorlage nach *Bimplus* exportiert und so für das *Allplan*-Projekt zur Verfügung gestellt. Eine weitere Option ist, die *Attributsetvorlage* manuell zu definieren. Hier können *Attribute* zu *Psets* zusammengefasst und diese direkt mit den *Entities* verknüpft werden, siehe Abbildung 3.1 rechts. Die zur Verfügung stehenden *Attribute* müssen jedoch bereits vorher in *Allplan* vorhanden sein. Ein Erstellen und Bearbeiten der *Attribute* ist hier nicht möglich.

Abseits der Projekteinstellungen existiert eine weitere Möglichkeit zur Implementierung von Informationsanforderungen. Hierbei sind mehrere Einzelschritte erforderlich, die manuell durchzuführen bzw. zu initiieren sind. Die Möglichkeit basiert auf einem sogenannten *PythonPart* namens

3.1 Möglichkeiten zur Implementierung von Informationsanforderungen

*AttributeWorkflow*¹. Hier können innerhalb einer Excel-Datei *Allplan-Attribute* definiert und im weiteren Sinne *Allplan-Objekten* (geometrischen Elementen, wie eine Wand) zugewiesen werden. Diese Excel-Datei wird, unter Zuhilfenahme des *PythonPart*, in *Allplan* eingespielt. Hiermit sind die vordefinierten Attribute integriert. Abbildung 3.2 zeigt das *PythonPart* und dessen Funktionen.

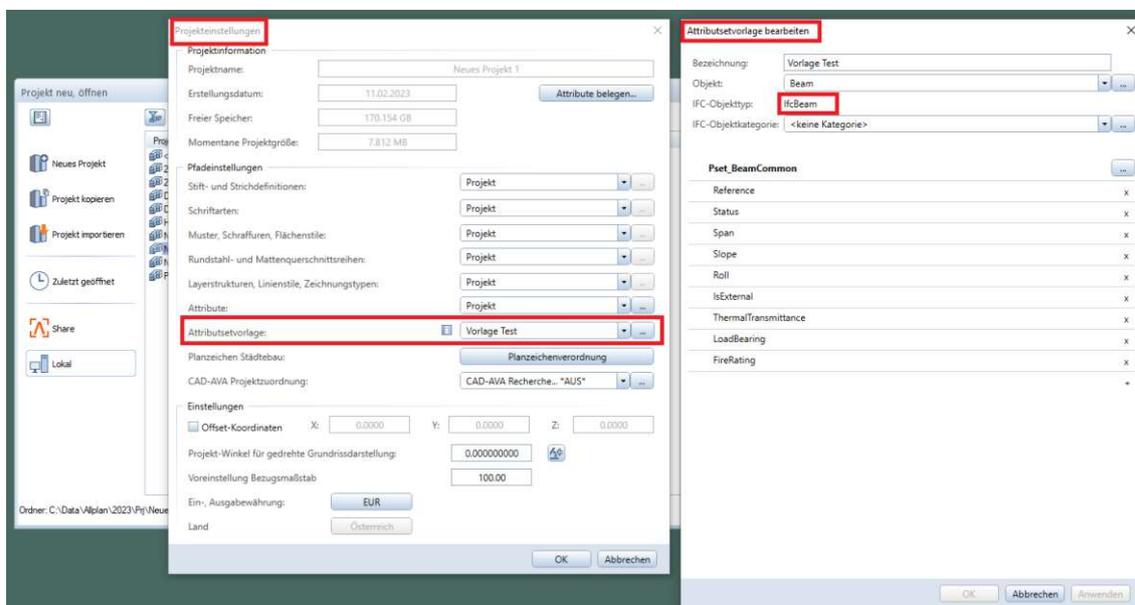


Abb. 3.1: Fenster *Projekteinstellungen* und *Attributsetvorlage bearbeiten* in *Allplan*

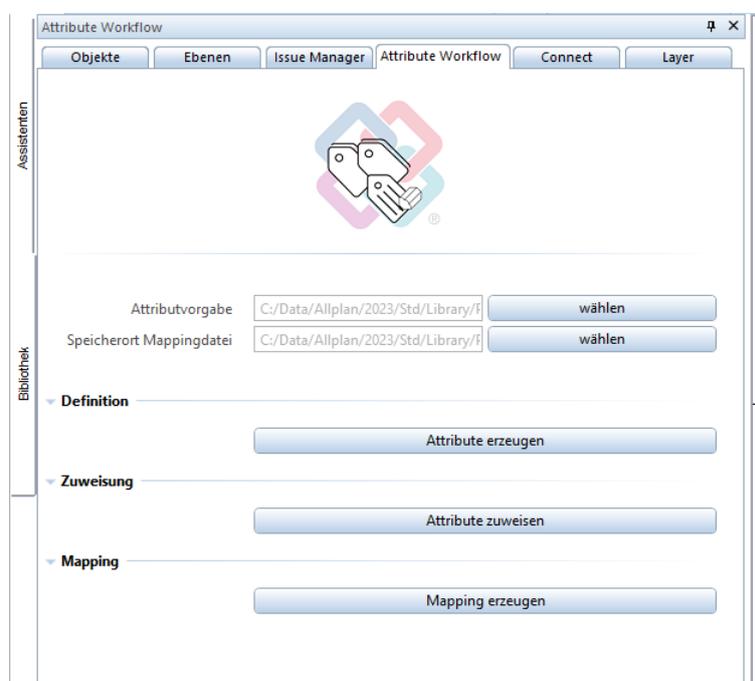


Abb. 3.2: *PythonPart AttributeWorkflow* in *Allplan*

¹Dieses *PythonPart* kann auf der Webseite *Allplan Connect* [3] heruntergeladen werden.

3.2 Datenstrukturwerkzeuge

Eichler et al. [8, S. 32] beschreiben Datenstrukturwerkzeuge wie folgt:

„Datenstrukturwerkzeuge zählen [...] zu den BIM-Werkzeugen. Sie sind webbasierte Dienste zur Erstellung und Modifikation von individuellen Datenstrukturen sowie darauf basierten Detaillierungsgraden. Für diese Tätigkeiten bieten sie eine zentrale Moderation und integrierte Distribution in verschiedene Kanäle (BIM-Applikation, BIM-Regelwerke etc.). Damit minimieren sie den jeweiligen individuellen Anpassungsaufwand. [...]“

Ein weiterer, verwendeter Begriff für ein Datenstrukturwerkzeug ist Informationssystem. Bei einem Informationssystem können zusätzlich zu Funktionen zur Datenstrukturierung Funktionen zur Prozessdarstellung enthalten sein.

Wie von Tomczak et al. [28] beschrieben, handelt es sich bei Datenstrukturwerkzeugen meistens um *proprietary software solutions*. Eine deutschsprachige Bezeichnung hierfür ist Unfreie Software. Der Software-Code ist demzufolge nicht einsehbar.

3.2.1 Datenstrukturwerkzeuge im openBIM-Prozess

Eichler et al. [8] zeigen einen Vergleich einer konventionellen Herangehensweise mit der Verwendung eines Datenstrukturwerkzeugs aus Sicht der Modellerstellung im openBIM-Prozess. Die Modellanforderungen, wie beispielsweise der *LOIN*, sind über den BAP für die Fachplaner vorgeschrieben. Bevor der Fachplaner mit der Modellerstellung beginnt, muss er die Informationsanforderungen in seine verwendete Software übernehmen. Bei den alphanumerischen Informationen sind dies die geforderten *Properties*, die in der Software anzulegen sind. Bei der konventionellen Herangehensweise wird das Anlegen manuell und von jedem Fachplaner durchgeführt. Die *Properties* werden angelegt und in einem weiteren Schritt mit der IFC-Datenstruktur verknüpft (*Mapping*). Durch das manuelle Eingeben ist diese Herangehensweise wenig effizient und fehleranfällig.

Bei der Nutzung eines Datenstrukturwerkzeugs erfolgt die Definition der *Properties* und das Mapping zentral. Dabei können Mappings für unterschiedliche Software berücksichtigt werden. Die *Properties* werden direkt mit den Leistungsbildern, Projektphasen und Anwendungsfällen (*Use cases*) verknüpft. Als Ergebnis erhält der Fachplaner eine Softwarevorlage für die Modellierungs- und Prüfsoftware, womit die manuelle Eingabe entfällt. Das Datenstrukturwerkzeug kann je nach Projekt und Organisation vom Auftraggeber zur Verfügung gestellt werden, oder die Projektbeteiligten verwenden ihr eigenes Datenstrukturwerkzeug. Abbildung 3.3 zeigt die zugehörige Darstellung aus [8, S. 57]. Im unteren Teil ist hier der openBIM-Prozess zwischen BIM-Koordination und Fachplanung ersichtlich. Rechts oben ist das Datenstrukturwerkzeug und dessen Funktionen dargestellt. Links oben wird angedeutet, dass möglicherweise Daten aus dem *buildingSMART Data Dictionary (bSDD)* oder dem *Use Case Management (UCM)* direkt übernommen werden können. Das Datenstrukturwerkzeug unterstützt bei den dargestellten Inhalten, wie der *Erstellung LOI/LOG* oder der *Erstellung Softwarevorlagen*. Die Pfeiler zwischen dem Feld Datenstrukturwerkzeug und dem grauen Feld *CDE* deuten die Einbindung des Datenstrukturwerkzeug im openBIM-Prozess an. Die Inhalte im Datenstrukturwerkzeug dienen als Grundlagen für den BAP und als Vorlagen, die beispielsweise im Form von *xml* oder *IDS*-Dateien zur Verfügung gestellt werden.

3.2.2 Arten von Datenstrukturwerkzeugen

Nach Mellenthin et al. [20] ist im deutschsprachigen Raum ein häufig genutztes Datenstrukturwerkzeug *BIMQ*. Eichler et al. [8, S. 32] bezeichnen *BIMQ* als ein „*typisches Beispiel für*

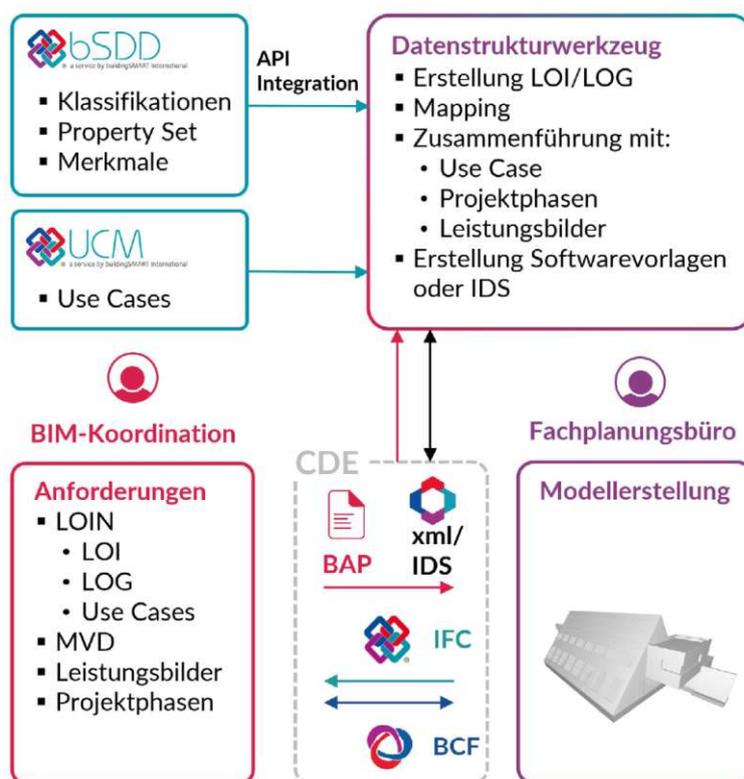


Abb. 3.3: Verwendung eines Datenstrukturwerkzeugs während der Phase der Modellerstellung im openBIM-Prozess, aus Eichler et al. [8, S. 57]

ein aktuelles Datenstrukturwerkzeug“. *BIMQ* ist eine Webapplikation, mit der Datenstrukturen mit einer Verknüpfung (Mappings) zu Projektphasen, Anwendungsfällen, der IFC-Datenstruktur und Software-spezifische Datenstrukturen erstellt werden können. *BIMQ* bietet die Funktion des Exports und Re-Imports der erstellten Datenbankinhalte in XLS-Dateien. Dies ermöglicht die Erstellung und Bearbeitung in Tabellenbearbeitungsprogrammen. Die erstellten Inhalte können als Dokumente für den Anhang der AIA, Softwarevorlagen bzw. Konfigurationsdateien für Modellierungssoftware oder als Grundlagen für Prüfsoftware ausgegeben werden.

Von Mellenthin et al. [20, S. 21] wird im Kontext von AIA-Inhalten als Werkzeug *CAFM-Connect Editor* erwähnt, und erläutert, dass: „Dort können Informationen nach vordefinierten Katalogen (DIN 277-2, DIN 276 oder GEFMA 198), teilweise mit einem inbegriffenen IFC-Mapping, in sogenannten Profilen erstellt werden. [...] Der Austausch kann über IFC-basierte Formate erfolgen (ifcXML oder ifcZIP)“. Der Fokus liegt hier auf Anwendungsfällen im Facility-Management, weshalb eine Verknüpfung der Inhalte mit unterschiedlichen Projektphasen nicht möglich ist. Außerdem wird ein *AIA-Editor* von Ekkodale erwähnt, „welcher Autodesk Revit-kompatible Parameterdateien im Sinne von Vorgaben aus AIA-Dokumenten erzeugt“. Hierbei handelt es sich jedoch um kein Datenstrukturwerkzeug, sondern eine Übersetzung von Inhalten aus den AIA in Software-spezifische Dateien.

Weitere existierende Werkzeuge für das Informationsmanagement legen den Fokus auf Informationen von Produktdaten. Ein Beispiel hierfür sind die Angebote des Unternehmens *BIMsystems* [5], wie die *BIM Platform waya*. Diese kann jedoch aufgrund der nicht vorhandenen Möglichkeit, individuelle *Properties* zu definieren, nicht als Datenstrukturwerkzeug gesehen werden. Ein Informationsmanagementsystem, das neben Funktionen zur Datenstrukturierung, Funktionen auf

Prozessebene abbildet, ist *BIM-Profil-Server* [4]. Hier sind jedoch Features zum Exportieren von Vorlagen für die BIM-Autorensoftware nicht vorhanden.

Einzelne Funktionalitäten von Datenstrukturwerkzeugen sind manchmal in *CDE* integriert. Diese beschränken sich jedoch am Beispiel *Bimplus* auf das Anlegen und Verknüpfen von *Psets* und *Properties* zu den Software-spezifischen Bezeichnungen der hinterlegten *Entities*. Eine Definition der Daten mit Bezug zu einem bestimmten Anwendungsfall etc. wird nicht abgebildet.

3.3 Anforderungen an ein Datenstrukturwerkzeug

In diesem Abschnitt werden die Anforderungen an ein Datenstrukturwerkzeug auf Grundlage einer Literaturrecherche formuliert. Ein Datenstrukturwerkzeug sollte die in den AIA üblicherweise festgelegten Informationsanforderungen abbilden können. Da AIA noch nicht standardisiert sind, werden die Anforderungen an ein Datenstrukturwerkzeug von einem durchgeführten Pilotprojekt abgeleitet. Beispielsweise wurde das Projekt *Bio-Institut der HBLFA Raumberg-Gumpenstein* wissenschaftlich begleitet [25] und die AIA öffentlich zugänglich gemacht. Im Anhang [26] sind die *LOI-Definitionen* festgelegt. Als Erstes wird die Abhängigkeit der Informationsanforderungen zu den Elementklassen (*Entities*) erwähnt. In weiterer Folge wird eine Unterscheidung nach *Fachmodellen* vorgenommen und der eigentliche Inhalt der Informationsanforderungen, nämlich die *Psets* und deren *Properties* angegeben. In den angehängten Tabellen werden je *Entity* die erforderlichen *Properties* aufgeschlüsselt. Die Überschriften der einzelnen Spalten lauten: *LOI-Klasse*, *Merkmale Übersetzung DE*, *Merkmal-Namen*, *Einheitentyp*, *Einheit*, *Verortung (Pset)* und *Verantwortung* (Projektbeteiligte). Mit der *LOI-Klasse* erfolgt eine Verknüpfung mit dem jeweiligen Fachmodell und der jeweiligen Projektphase. Die für ein Datenstrukturwerkzeug abgeleiteten, allgemeinen Funktionen werden folgend zusammengefasst:

- Erstellen von Strukturen an *Entities*, *Psets* und standardisierten sowie benutzerdefinierten *Properties*
- Korrektes IFC-*Mapping* der erstellten Strukturen
- Zuordnen der Einträge der Strukturen zu Fachmodellen und Projektphasen
- Ausgabe der erstellten Strukturen, zumindest im PDF-Format

In Tabelle 3.1 werden die in Abschnitt 3.2.2 genannten Datenstrukturwerkzeuge auf Vorhandensein dieser Funktionen verglichen². In weiterer Folge gilt die qualitative Anforderung der Datensicherheit in einem Datenstrukturwerkzeug grundsätzlich als vorausgesetzt. Datensicherheit steht hier dafür, dass Änderungen einfach und fehlerfrei möglich sind und die Änderungen in der Datenstruktur konsistent weitergegeben werden.

²Die verglichenen Versionen sind: *BIMQ* – 2.7.9, *CAFM-Connect Editor* – 3.0.142.891, *BIM-Profil-Server* – frontend version 1.2.14 – backend version 1.2.12, *Bimplus* – 15.02.2023.

Tab. 3.1: Vergleich der erforderlichen, allgemeinen Funktionen verschiedener Datenstrukturwerkzeuge

Datenstrukturwerkzeug	<i>Entities, Psets, Properties</i> (benutzerdefinierte)	IFC-Mapping	Zuordnung zu Fachmodellen und Projektphasen	Ausgabe
<i>BIMQ</i>	■	■	■	■
<i>CAFM-Connect Editor</i>	■	■	□	■
<i>BIM-Profil-Server</i>	■	■	■	■
<i>Bimplus</i>	■	■	□	□

■ Funktion enthalten

□ Funktion nicht enthalten

3.4 Definition des Anwendungsfalls

Zusätzlich zu den allgemeinen Anforderungen an ein Datenstrukturwerkzeug können weitere Anforderungen abhängig vom zu erfüllenden Anwendungsfall sein. Um weitere Anforderungen zielführend formulieren zu können, muss folglich zuerst ein Anwendungsfall festgelegt werden. Der, in dieser Arbeit ausgewählte und als Beispiel festgelegte, Anwendungsfall ist die

Erstellung einer Standardkonfiguration für Architektur und Tragwerksplanung für die Verwendung in der BIM-Autorensoftware *Allplan*.

Die Informationsanforderungen an geometrischen und alphanumerischen Informationen resultieren aus dem Bedarf im konkreten Anwendungsfall, siehe Eichler et al. [8]. Diese Anforderungen werden durch den *LOIN* abgebildet, der mit den Normen ISO 19650-1 [18] und ÖNORM EN 17412-1 [23] standardisiert ist. Die zugehörigen *LOI* und *LOG*, die Teil des *LOIN* sind, werden wie bei einem BIM-Anwendungsfall getroffen. Da im Anwendungsfall eine Standardkonfiguration abgebildet werden soll, werden als Inhalt der alphanumerischen Informationsanforderungen (*LOI*) die standardisierten *Psets* (Suffix *Common*) und deren zugehörigen *Properties* festgelegt. Ein Beispiel ist das *Pset_BeamCommon* und dessen *Properties Reference, Status, Span, Slope* etc. Außerdem wird als Grundlage für die Auswahl an *Entities* (geometrische Elemente) die Klasse *IfcBuiltElement*³ herangezogen. Die Klasse *IfcBuiltElement* ist Teil der funktionalen Struktur der IFC-Datenstruktur und nach Eichler et al. [8] im Kontext des Hochbaus essenziell. Die ausgewählten *Entities* sind die Hochbau-spezifischen Unterklassen von *IfcBuiltElement*, beispielsweise *IfcBeam, IfcChimney, IfcColumn, IfcCovering* etc. In Abbildung 3.4 ist die Vererbungsstruktur der Klasse *IfcBuiltElement* dargestellt. Die Unterklassen sind schwarz hinterlegt, wobei die Hochbau-spezifischen zusätzlich rot umrandet sind.

Beim genannten Anwendungsfall und den, in dieser Arbeit beschriebenen, Prozessabläufen liegt der Fokus auf den alphanumerischen Informationen. Für die geometrischen Informationen sowie die Dokumentationen, die ebenfalls Teil des *LOIN* sind, werden deswegen keine Anforderungen formuliert.

Da der festgelegte Anwendungsfall die Kompatibilität mit der BIM-Autorensoftware *Allplan* einschließt, wird ein Datenstrukturwerkzeug gewählt, das eine Schnittstelle zu *Allplan* zur Verfügung stellt. Die Datenstrukturwerkzeuge, die sämtliche in Tabelle 3.1 aufgelisteten

³Mit der Version IFC4.3.0.0 wurde die Klasse von *IfcBuildingElement* zu *IfcBuiltElement* umbenannt.

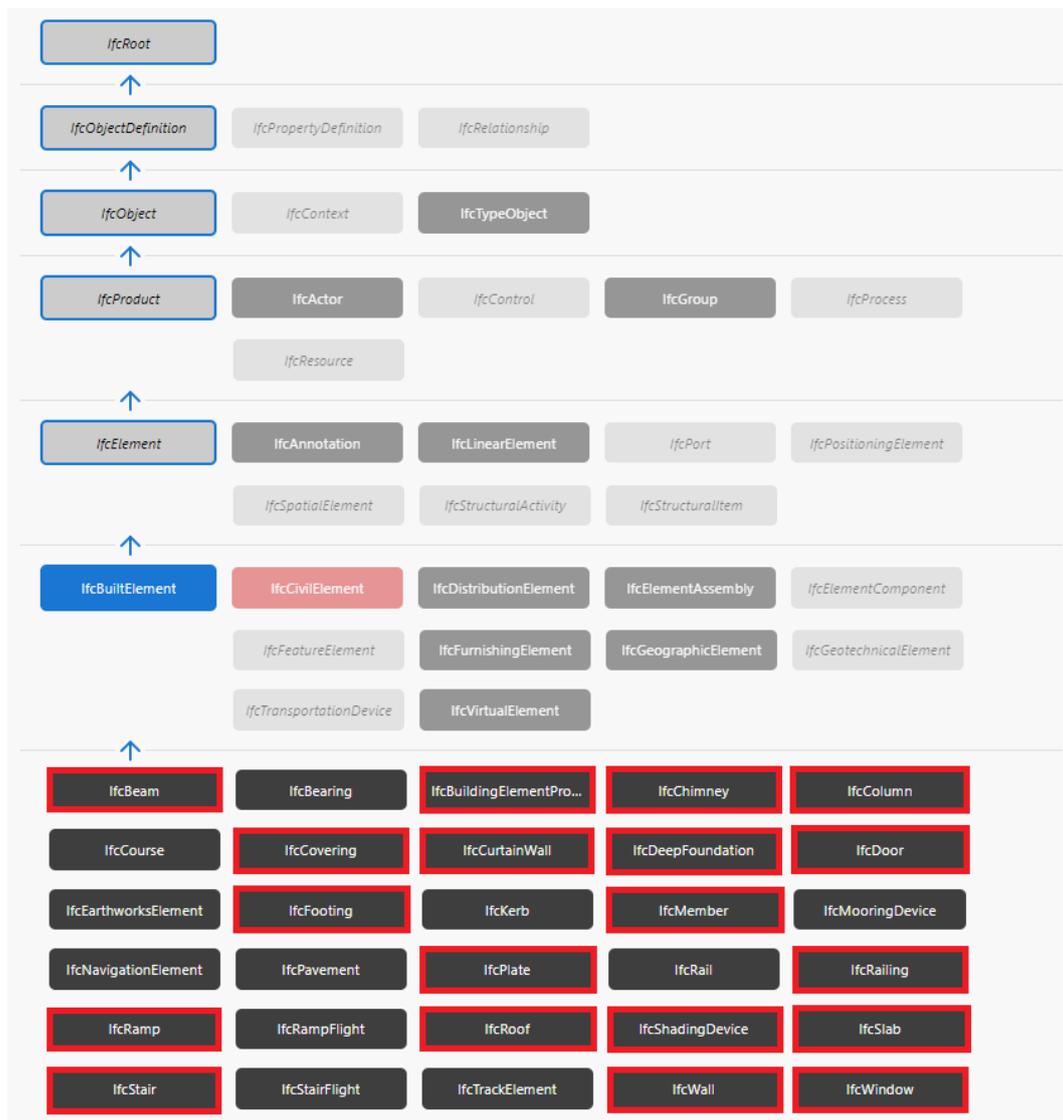


Abb. 3.4: Vererbungsstruktur der Klasse *IfcBuiltElement* [15]

allgemeinen Anforderungen erfüllen sind *BIMQ* und *BIM-Profil-Server*. Da jedoch lediglich *BIMQ* eine Schnittstelle zu *Allplan* bietet, wird *BIMQ* als für am geeignetsten für den definierten Anwendungsfall beurteilt und in der folgenden Fallstudie in Kapitel 5 verwendet.

Kapitel 4

Prozessablauf bei der Verwendung eines Datenstrukturwerkzeuges

Im Kapitel 3 wurden die Ausgangslage, der Anwendungsfall und die Anforderungen an Datenstrukturwerkzeuge formuliert. Damit sind die Randbedingungen für den Prozess bei Verwendung eines Datenstrukturwerkzeuges festgelegt. Zu Beginn dieses Kapitels wird der bestehende Prozess skizziert und mögliche Verbesserungspunkte aufgezeigt. Darauf aufbauend wird ein optimierter Prozess dargestellt und dessen Vorteile erläutert.

4.1 Beschreibung des bestehenden Prozesses bei Verwendung eines Datenstrukturwerkzeuges

Der bestehende Prozess bei Verwendung eines Datenstrukturwerkzeuges wurde gemeinsam mit den Experten des Flughafens Wien und der TU Wien im Zuge eines Workshops evaluiert. Dabei wurden folgende Charakteristika der Strukturierung der Daten identifiziert:

- Der Fokus liegt auf den *Psets* und die Einträge der *Psets* sind einmalig.
- Die Ordnung erfolgt nach den *Psets*.
- Die *Properties* werden den *Psets* direkt und fest angefügt.
- Dadurch kommen mehrere, nicht verknüpfte Einträge für gleichnamige *Properties* vor.
- Die *Entities* werden getrennt von den *Psets* und *Properties* angelegt.
- Die *Psets*, mit den angefügten *Properties*, werden als Paket den *Entities* zugeordnet.

Ein Beispiel für die angeführten Charakteristika zeigt eine exportierte Excel-Datei aus einem bestehenden Projekt von einem der Experten. In Abbildung 4.1 ist ersichtlich, dass im Datenstrukturwerkzeug (hier *BIMQ*) die *Psets* wie folgt strukturiert wurden: In der 1. Ebene (mittlere Spalte) sind die *Psets* aufgelistet. Die *Properties* sind in der 2. Ebene (rechte Spalte) platziert und sind somit direkt den *Psets* der 1. Ebene angefügt. In Abbildung 4.1 ist ebenfalls zu sehen, dass gleichnamige *Properties* (hier *Reference* und *Status* [blau markiert]) als idente Einträge bei unterschiedlichen *Psets* existieren.

Diese Vorgehensweise wird in Abbildung 4.2 mithilfe der Business Process Model and Notation (BPMN) visualisiert. Der erste Prozessschritt ist das Erstellen der Einträge für die *Psets* und deren *Properties*. Diese werden direkt und zusammen erstellt. Das bedeutet, dass zuerst der Eintrag für das *Pset* angelegt wird und direkt in der Struktur darunter die zugehörigen *Properties*. Die *Psets* und *Properties* sind mit Name, *Property Type*, Datentyp und gegebenenfalls der Einheit und dem Optionen-Set anzulegen. Darauffolgend können die *Entities* der geometrischen Elemente, wie Stütze, Decke, etc. erstellt werden. Das Hinzufügen der Mappings zur IFC-Datenstruktur

Hauptgruppe (Navigation)	1.Ebene	2.Ebene
Merkmale	Pset_ManufacturerOccurrence	Beschaffungsdatum
		Chargennummer
		SerienNr
	Pset_AlarmTypeCommon	Reference
		Status
	Pset_AudioVisualApplianceTypeCommon	Reference
		Status
	Pset_BoilerTypeCommon	EnergySource
		HeatTransferSurfaceArea
		IsWaterStorageHeater
		NominalEnergyConsumption
		NominalPartLoadRatio
		OperatingMode
		OutletTemperatureRange
		PressureRating
		Reference
		Status
WaterInletTemperatureRange		
WaterStorageCapacity		

Abb. 4.1: Excel-Export aus dem Bereich *Vorlage* in *BIMQ* aus einem vergangenen Projekt eines Workshop-Teilnehmers

und der spezifischen BIM-Autorensoftware erfolgt im nächsten Schritt. Beim Mapping spezifisch für die Autorensoftware werden die Namen der *Psets* oder *Properties* angegeben, die in der Autorensoftware angezeigt werden sollen. Danach ist das zusammengesetzte Paket aus *Pset* und *Properties* mit den *Entities* zu verknüpfen. Beispielsweise ist das *Pset_BeamCommon* mit dessen zugehörigen *Properties Span* etc. der *Entity IfcBeam* zugeordnet. Die Sammlung dieser Zuordnungen wird, unter anderem im Datenstrukturwerkzeug *BIMQ* [1], als Projektanforderungen bezeichnet. Im nächsten Schritt werden die Anforderungen der zugehörigen Projektphase und dem Anwendungsfall zugewiesen. Die Projektanforderungen können als Softwarevorlage aus dem Datenstrukturwerkzeug exportiert und in die BIM-Autorensoftware importiert werden. Zusätzlich können diese als fertige Anhänge für die AIA oder als Grundlage für Prüfregeln ausgegeben werden. In der BIM-Autorensoftware ermöglicht nun die importierte Vorlage, dass die festzulegenden *Properties* eines Bauteils bereits strukturiert vorhanden sind und die zugehörigen Werte direkt eingegeben werden können. Beispielsweise kann bei der Modellierung eines Balkens (*IfcBeam*) beim nun vorhandenen *Pset_BeamCommon* und weiters dem *Property Status* einer der möglichen Optionen, z. B. *New* (Neu), *Existing* (Bestand) etc. ausgewählt werden. Nachdem die Modellierung abgeschlossen ist, wird die IFC-Datei exportiert. Zur Kontrolle des Vorhandenseins aller Anforderungen kann eine formale Prüfung mithilfe einer Prüfsoftware erfolgen. Hierfür können die exportierten Prüfregeln verwendet werden. Die Prüfung fällt negativ aus, wenn in der IFC-Datei z. B. nicht sämtliche *Properties*, oder deren festgelegten Optionen, in einem *Pset* und der verknüpften *Entity* vorhanden sind. Weitere Gründe können auch von den Anforderungen abweichende *Data Type* oder Einheiten der *Properties* sein. Im Falle einer nicht erfolgreichen Prüfung wird eine Feedbackschleife aktiviert. Zuerst sollte eine Überprüfung der Export-Einstellungen in der BIM-Autorensoftware erfolgen. Falls die Prüfung wiederholt negativ ausfällt, ist die Ursache, für die in der Prüfung aufgezeigten Abweichungen, im Datenstrukturwerkzeug zu suchen und im jeweiligen Prozessschritt zu korrigieren. Sobald die Prüfung erfolgreich ist, ist der Prozessablauf beendet.

4.1 Beschreibung des bestehenden Prozesses bei Verwendung eines Datenstrukturwerkzeuges

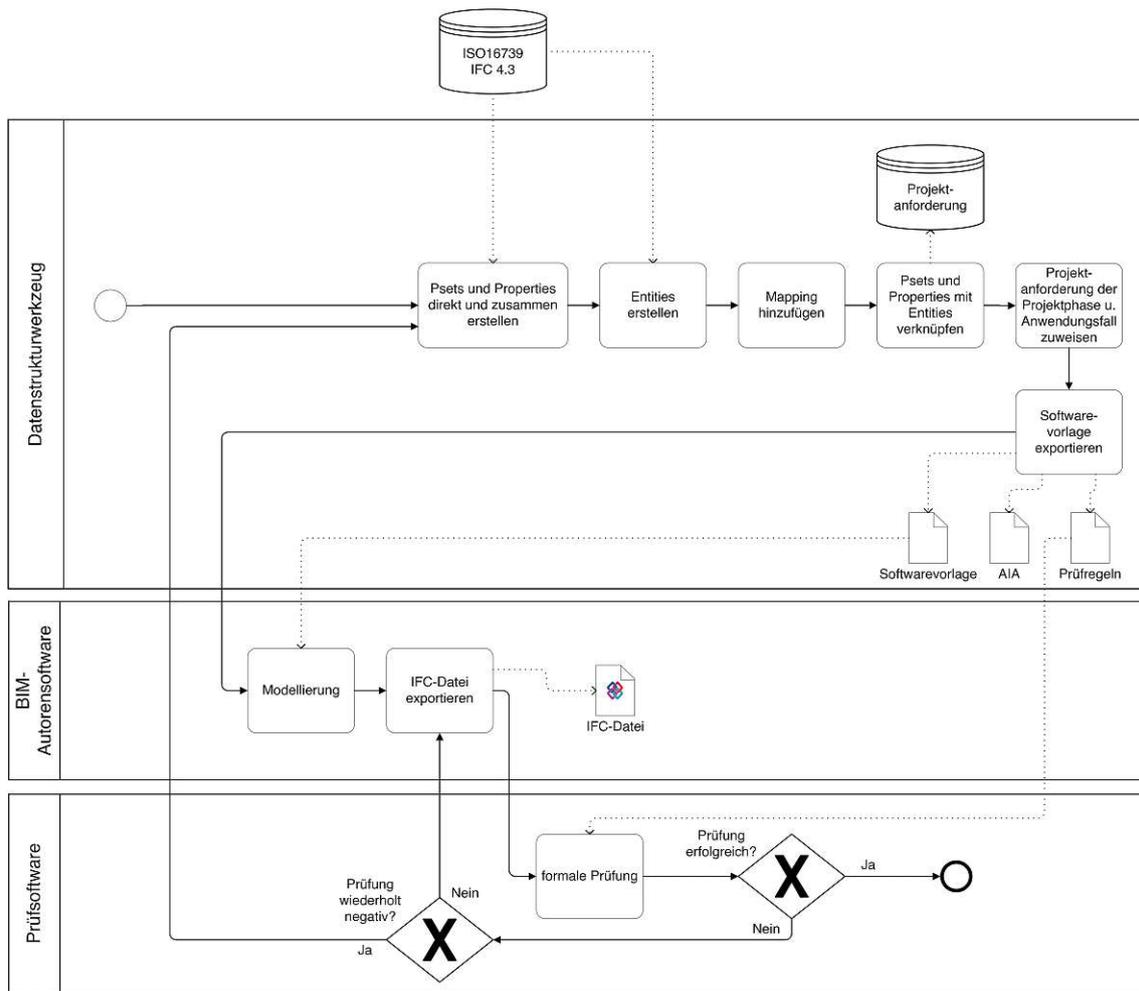


Abb. 4.2: Bestehender Prozessablauf bei Verwendung eines Datenstrukturwerkzeuges

Aufgrund der genannten Charakteristika und dem dargestellten Ablauf des bestehenden Prozesses ergeben sich mehrere Herausforderungen:

- Da Einträge an gleichnamigen *Properties* im Datenstrukturwerkzeug mehrfach vorkommen, sind diese nicht eindeutig.
- Eine hohe Fehleranfälligkeit, da die Definitionen der gleichnamigen *Properties*, wie der *Property Type* oder die Einheit, wiederholt angegeben werden müssen.
- Erhöhter Aufwand, da bei der Eingabe neuer *Psets* alle zugehörigen *Properties* neu eingegeben werden müssen, obwohl eventuell einzelne oder alle bereits existieren würden.
- Erhöhter Aufwand und Fehleranfälligkeit beim Ändern von Definitionen der *Properties*, da die Änderung bei allen gleichnamigen *Properties* durchgeführt werden muss.
- Geringere Flexibilität bei der Änderung von *Psets*, da beispielsweise beim Löschen einzelner *Properties*, die entsprechenden Einträge aus der Tabellenstruktur entfernt werden müssen (da *Pset* und *Properties* eine fixe Einheit bilden).
- Größere Datenmenge und Unübersichtlichkeit in der Tabellenstruktur, aufgrund der mehrfach vorhandenen Einträge gleichnamiger *Properties*.

4.2 Verbesserungspunkte beim bestehenden Prozess

Bei dem, in Abschnitt 4.1, erwähnten Workshop wurde erarbeitet, wie der bestehende Prozess optimiert werden kann. Die Ziele der Optimierung waren eine Verbesserung der Datensicherheit und Effizienz (Einfachheit und Flexibilität) der strukturierten Daten. Um diese zu erreichen wurden mehrere Verbesserungspunkte festgehalten. Beim bestehenden Prozess sind die *Psets* mit deren *Properties* getrennt von den *Entities* abgelegt. Diese fragmentierte Strukturierung ist auch für die *Psets* und *Properties* umzusetzen. Dies bedeutet, dass *Properties* von *Psets* losgelöst und getrennt abgelegt werden. Die *Properties* werden einzeln und zentral angelegt, in einem sogenannten Silo. Der Begriff des Silos wurde von den Workshop-Teilnehmern für die Sammlung von *Psets* oder *Properties* gewählt. Dies ist kein standardisierter Begriff. Im jeweiligen Silo soll ein *Pset* oder *Property* nur einmal vorhanden und damit einzigartig sein. Diese einmaligen Einträge werden im weiteren Prozess lediglich referenziert, was die Konsistenz der Daten gewährleistet. Die Silos stellen dementsprechend die Quelle der *Psets* und *Properties* dar und werden ausschließlich hier definiert. Das Zusammenführen der *Properties* mit den *Psets* erfolgt in einem sogenannten *Konfektionssilo*, der auf die Einträge im *Property*- und *Psets*-Silo referenziert. Die *Properties* sind demnach lediglich ein Mal, nämlich im *Property*-Silo, mit deren Name, *Property Type*, Datentyp, und gegebenenfalls der Einheit und dem Optionen-Set definiert. Damit ist im weiteren Sinne ein *Single Source of Truth*-Prinzip umgesetzt.

4.3 Beschreibung des optimierten Prozesses

Der optimierte Prozessablauf kann skizziert werden, indem aufbauend auf dem, in Abschnitt 4.1 beschriebenen, bestehenden Prozess die genannten Verbesserungspunkte berücksichtigt werden. Als Inhalt für den Prozess gilt der Anwendungsfall der Erstellung einer Standardkonfiguration, der in Abschnitt 3.4 erläutert wurde. Die beschriebene Auswahl an *Psets*, *Properties* und *Entities* stellen die Input-Daten für den Prozess dar.

Im Unterschied zum bestehenden Prozess sind die ersten Schritte das getrennte Erstellen des *Property*- und *Pset*-Silo im Datenstrukturwerkzeug. Ob zuerst der *Property*- oder der *Pset*-Silo erstellt wird, ist frei wählbar. Bei beiden Prozessschritten werden die *Psets* mit Name und die *Properties* mit Name, *Property Type*, Datentyp und gegebenenfalls der Einheit und dem Optionen-Set angelegt. Beim gewählten Anwendungsfall werden für den *Pset*-Silo die Standard-*Psets* gemäß ISO 16739 [17] verwendet. Im nächsten Schritt sind die *Properties* aus dem *Property*-Silo mit den entsprechenden *Psets* aus dem *Pset*-Silo zu verknüpfen. Diese Strukturierung wird im sogenannten *Konfektionssilo* gespeichert. Das Hinzufügen der Mappings zur IFC-Datenstruktur und der spezifischen BIM-Autorensoftware erfolgt gleich wie beim bestehenden Prozess. Beispielsweise ist folglich der *Entity IfcBeam* das *Pset_BeamCommon* und diesem wiederum die zugehörigen *Properties Span* etc. zugeordnet. Im nächsten Schritt werden die Anforderungen der zugehörigen Projektphase und dem Anwendungsfall zugewiesen. Die folgenden Prozessschritte, beginnend mit *Softwarevorlage exportieren*, sind ident zum bestehenden Prozess.

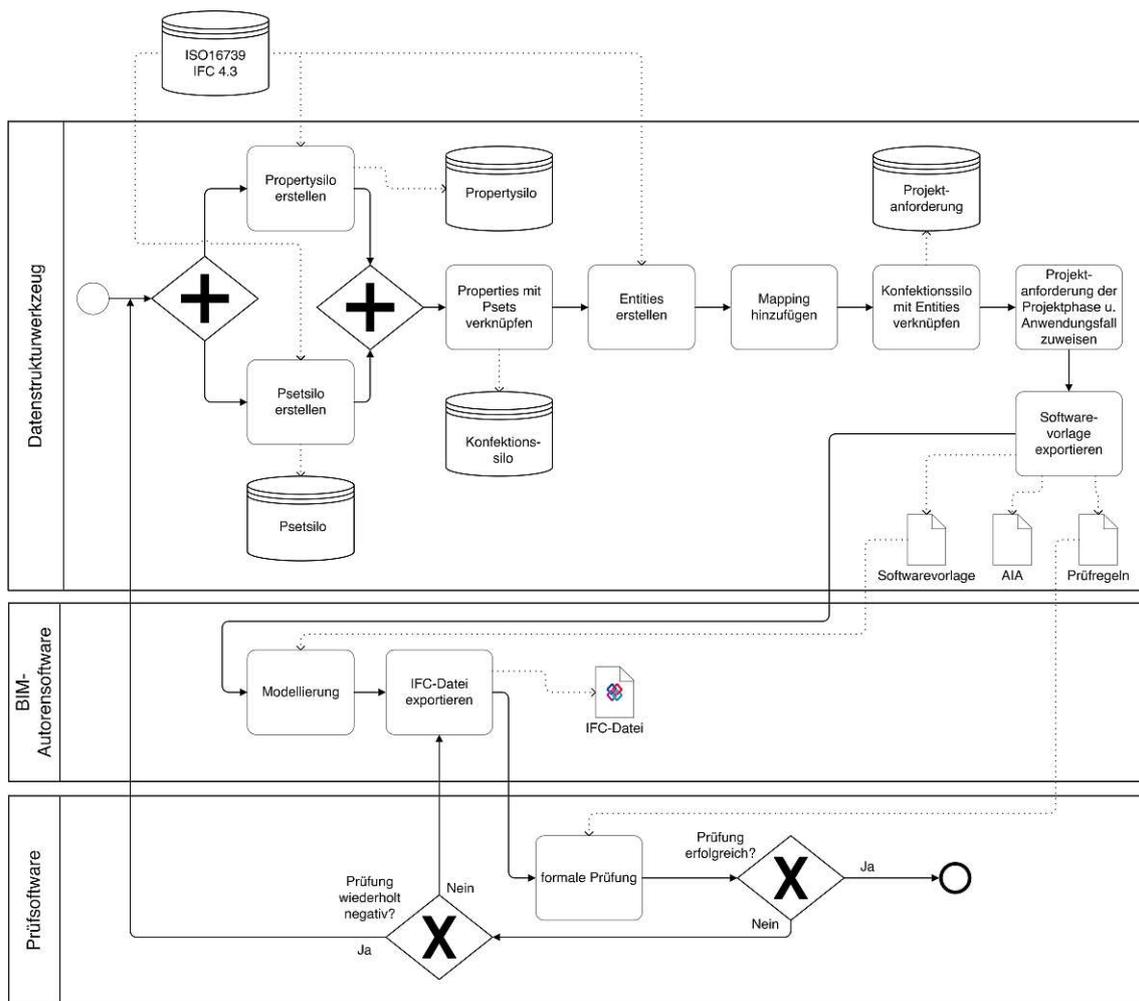


Abb. 4.3: Optimierter Prozessablauf bei Verwendung eines Datenstrukturwerkzeuges

4.4 Vorteile durch den optimierten Prozess

Die Umsetzung der Verbesserungspunkte im optimierten Prozess bringt, im Vergleich zum bestehenden Prozess, folgende Vorteile:

- Große Übersichtlichkeit, da *Entities*, *Psets* und *Properties* an nur einem Ort spezifiziert werden und hier zu finden sind.
- Hohe Datensicherheit durch die Konsistenz der Daten, da jedes *Property* etc. nur einmal angelegt ist. Bei einer Änderung dieses einmaligen Eintrages wird diese automatisch in allen referenzierten Strukturen, wie dem Konfektionssilo und den Projektanforderungen, übernommen.
- Auf Basis der Silos können neue Konfigurationen, spezifische *Psets* und projekt- oder nutzerspezifische Konfigurationen effizient (schnell und übersichtlich) erstellt werden. Dies gilt beispielsweise für die Erstellung von Informationsanforderungen für neue Projekte oder zusätzliche Anwendungsfälle.

Kapitel 5

Fallstudie

Die Umsetzbarkeit des optimierten Prozessablaufes aus Abschnitt 4.3 wird mithilfe einer Fallstudie analysiert. Die konkreten Forschungsfragen, die hier untersucht werden, lauten:

1. Wie kann, innerhalb eines Anwendungsfalls, eine Standardkonfiguration mittels eines Datenstrukturwerkzeuges erstellt und in die BIM-Autorensoftware *Allplan* importiert werden?
2. Welche Herausforderungen entstehen bei der Implementierung eines Datenstrukturwerkzeuges in einem konkreten Fallbeispiel?

Als Datenstrukturwerkzeug wird *BIMQ* und als BIM-Autorensoftware *Allplan* gewählt. *BIMQ* erfüllt die für den Anwendungsfall gestellten Anforderungen, siehe Abschnitt 3.4. In der Fallstudie wird gemäß dem definierten Anwendungsfall die Entwicklung einer Standardkonfiguration verfolgt. Dementsprechend sind die zu strukturierenden Daten die Standard-*Psets* und deren zugehörigen *Properties*.

In den folgenden Abschnitten wird zuerst der konkrete Prozess beschrieben, der sich aus dem Adaptieren des optimierten Prozessablaufes aus Abschnitt 4.3 auf die Softwareprodukte *BIMQ* und *Allplan* ergibt. Darauf folgend werden die einzelnen Schritte in der Umsetzung erläutert. Diese werden nach den Softwareprodukten in die Prozessschritte in *BIMQ* (Abschnitt 5.2) und die Prozessschritte in *Allplan* (Abschnitt 5.3) gegliedert. In Abschnitt 5.5 wird die Umsetzung durch qualitative und quantitative Tests überprüft. Die Analyse und die Diskussion der Ergebnisse erfolgen in Kapitel 6.

5.1 Prozessablauf bei der Verwendung von *BIMQ* und *Allplan*

Der in Abschnitt 4.3 beschriebene, optimierte Prozess muss bei der Verwendung der konkreten Softwareprodukte angepasst bzw. um einzelne Zwischenschritte ergänzt werden. Abbildung 5.1 zeigt diesen konkreten Prozess. Ergänzend zum optimierten Prozess (vgl. Abbildung 4.3) ist ersichtlich, dass die entsprechenden Namen der Softwaremenüs angegeben sind. Beispielsweise erfolgt das Erstellen des *Property-Silos* und *Pset-Silos* in *BIMQ* im Menü *Vorlagen*. Der Prozessschritt *Properties mit Psets verknüpfen* mit Ergebnis des Konfektionssilos ist in *BIMQ* nicht vollumfänglich umsetzbar. In *BIMQ* wäre hier ein zusätzliches Menü erforderlich. Stattdessen wären die *Properties* und *Psets* im Menü *Projektkonfiguration* zusammenzuführen. Hier ergibt sich jedoch der Nachteil, dass die *Properties* bei sich wiederholenden *Entities* oder *Psets* wiederholt und einzeln zu verknüpfen sind. Im optimierten Prozess ist vorgesehen, dass *Properties* und *Psets* über den Konfektionssilo miteinander verknüpft sind und als Paket den *Entities* zugewiesen werden. Diese Problematik wird im späteren Abschnitt 5.2.4 detaillierter erläutert.

Eine Adaption des optimierten Prozesses ergibt sich bei der Aufgabe *Softwarevorlage exportieren*. Bei der BIM-Autorensoftware *Allplan* kann die Softwarevorlage von *BIMQ* nicht direkt nach *Allplan* übertragen werden. Hier ist ein Zwischenschritt mit der *CDE Bimplus*¹ notwendig.

¹ *Bimplus* ist eine *Common Data Environment* (Kollaborationsplattform, *CDE*) von *Allplan*.

5.1 Prozessablauf bei der Verwendung von BIMQ und Allplan

Die Softwarevorlage ist von *BIMQ* nach *Bimplus* zu exportieren und dort dem *Allplan*-Projekt zuzuweisen, welches mit *Bimplus* verknüpft ist. Die Vorlage wird darauffolgend in *Allplan* automatisch von *Bimplus* heruntergeladen. Bei der Aufgabe *Softwarevorlage exportieren* ist zusätzlich eine sogenannte *Zuordnungsdatei* auszugeben. Diese wird beim Exportieren der IFC-Datei in *Allplan* eingespielt und enthält die Zuordnung der *Globally Unique Identifier (GUID, Zahl zur Identifizierung)* der *Allplan-Attribute* zu den entsprechenden *Properties* der Vorlage. In den folgenden Abschnitten werden die einzelnen Prozessschritte detailliert beschrieben.

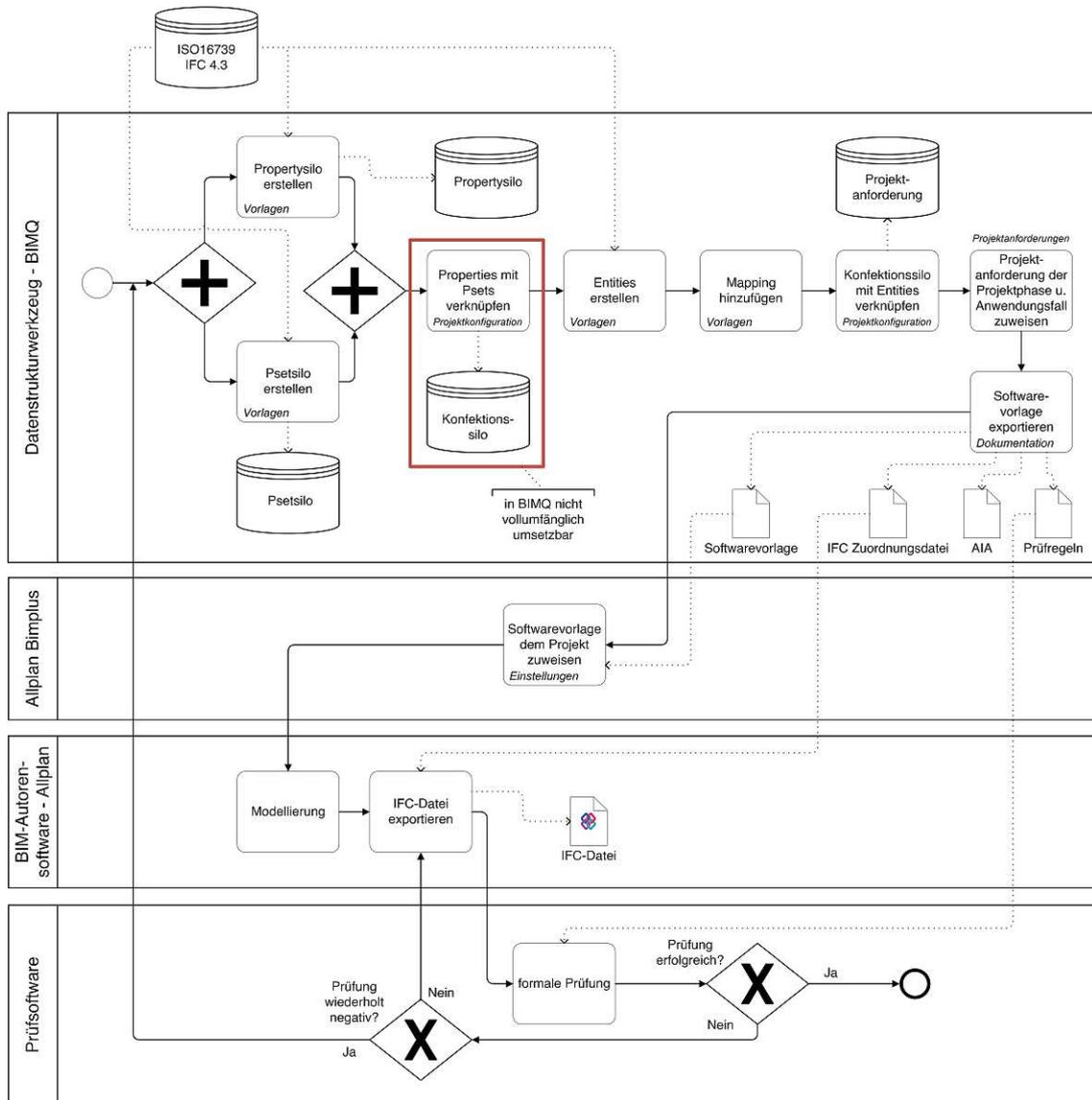


Abb. 5.1: Konkreter Prozess bei Verwendung des Datenstrukturwerkzeuges *BIMQ* und der BIM-Autorensoftware *Allplan*

5.2 Prozessschritte in *BIMQ*

Die Umsetzung des konkreten Prozesses startet im Datenstrukturwerkzeug *BIMQ*. In den folgenden Abschnitten wird die Umsetzung der einzelnen Schritte des konkreten Prozesses beschrieben. Dem vorangestellt werden allgemein die Funktionen und die Struktur von *BIMQ* erläutert.

5.2.1 Beschreibung von *BIMQ*

BIMQ ist ein webbasiertes Datenstrukturwerkzeug. Es wird unter dem Begriff Informationsmanagementsystem vermarktet. *BIMQ* ist kostenpflichtig und bietet unterschiedliche Nutzungspakete, die sich jedoch nicht im Umfang an Funktionen unterscheiden. Die Unterschiede liegen in der Anzahl an zur Verfügung stehenden Nutzern, Projekten, etc. [1].

Die Gliederung der Webapplikation bilden Menüs, die sich an den einzelnen, durchführbaren Schritten bei der Definition von Informationsanforderungen orientieren. Diese Menüs bauen aufeinander auf und sind in der 2023 aktuellen Version 2.7.9-3 mit *Meine Projekte*, *Einstellungen*, *Vorlagen*, *Projektkonfiguration*, *Projektanforderungen* und *Dokumentation* benannt. Auf der Startseite bzw. im Menü *Meine Projekte* kann das zu bearbeitende Projekt ausgewählt werden. Im geöffneten Projekt sind die weiteren, genannten Menüs ersichtlich.

Beim Menü *Einstellungen* gibt es einerseits das Untermenü *Projektübersicht*. Dies ist in Abbildung 5.2 dargestellt. Festlegungen zum Projektname, einer Beschreibung etc. und das Hinzufügen von Projektteilnehmer, unter Angabe der *Rolle im Projekt*, sind hier möglich. Andererseits gibt es das Untermenü *Projekteinstellungen*, bei dem unter anderem eine BIM-Autorensoftware angegeben werden kann, für die in weiterer Folge spezifische Mappings definiert werden können. Aktuell gibt es Schnittstellen zu *ArchiCAD*, *Revit*, *Vectorworks*, *Allplan* und *ProVI*. Im gleichen Untermenü sind weiters die zu verwendende IFC-Version und die Sprachen festzulegen sowie Quellen für mögliche Klassifikationssysteme, wie aus dem *buildingSMART Data Dictionary (bSDD)*, anzugeben.

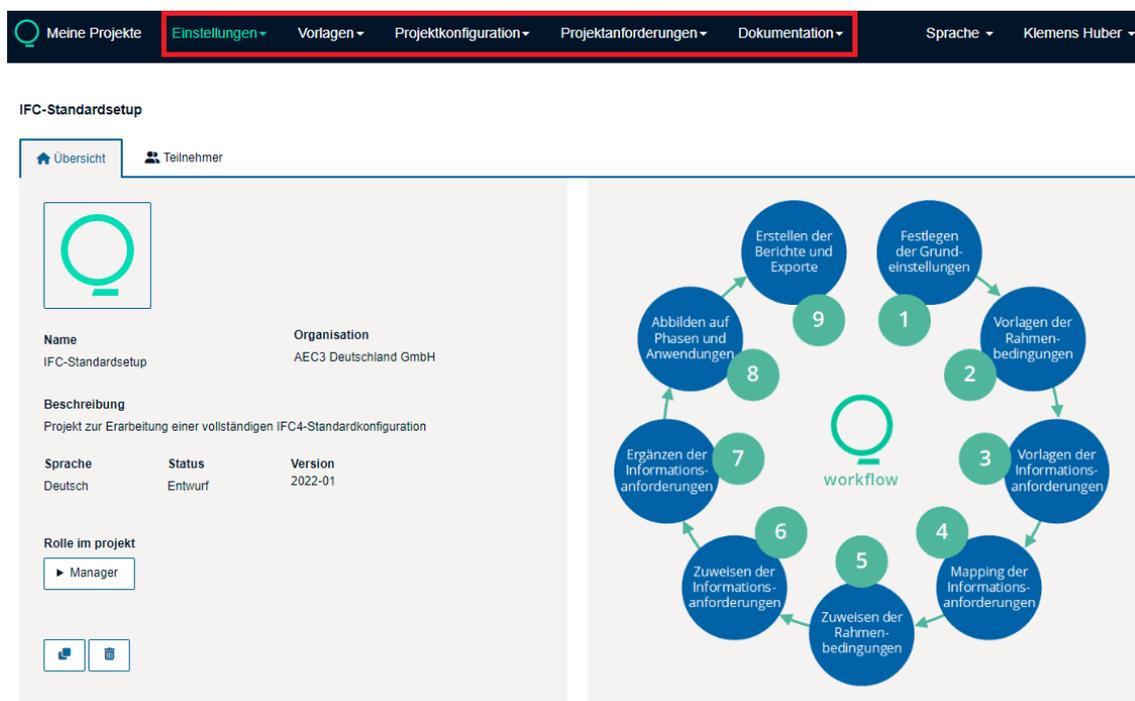


Abb. 5.2: *BIMQ*-Untermenü *Projektübersicht* im Menü *Einstellungen*

Im Menü *Vorlagen* werden die Datensätze angelegt und bearbeitet. Abbildung 5.3 zeigt die Kategorisierung der Datensätze gemäß den Namen der verfügbaren Untermenüs *Leistungsbilder*, *Projektphasen*, *Anwendungsfälle* und *Alle Komponentenvorlagen*. *Alle Komponentenvorlagen* ist vorgesehen, um in Tabellenform die Fachmodelle, *Entities*, *Psets* und *Properties* inkl. ihrer Definitionen, wie Name, Code, ggf. *Mapping*, etc. anzulegen. Bei einem sogenannten *Mapping* wird zum entsprechenden Datensatz die zugehörige Definition in der IFC-Datenstruktur und später auch der BIM-Autorensoftware angegeben. Beispielsweise gibt es den Datensatz namens *Unterzug*, für den als zugehöriges IFC-*Mapping* *IfcBeam* und als Software-*Mapping* für *Allplan Balken* möglich ist. In jedem dieser Untermenüs gibt es unterstützende Funktionen, wie *Suchen & Filtern*, *Mehrfachzuweisung*, *Import aus Vorlage*, den *Excel Import/Export* etc.

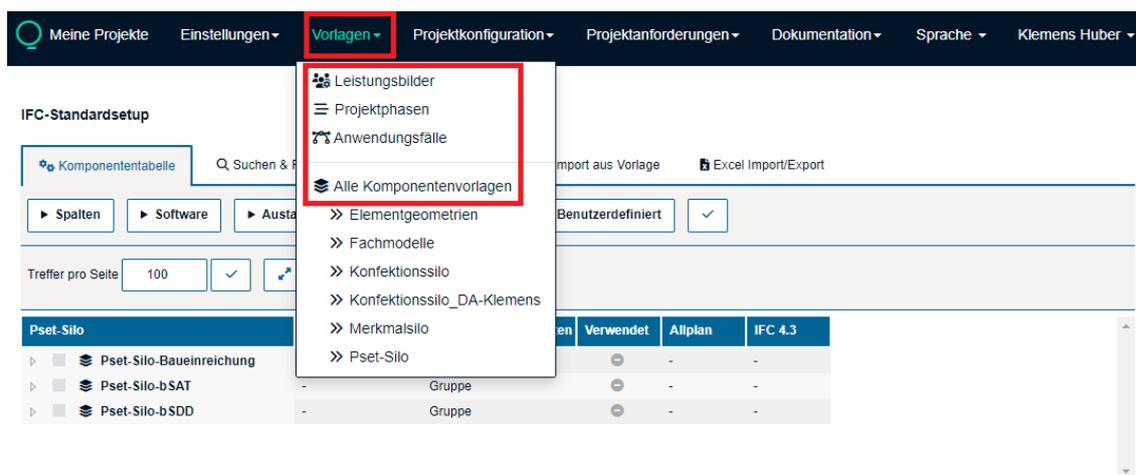


Abb. 5.3: BIMQ-Menü Vorlagen

Das Menü *Projektkonfiguration* enthält die Untermenüs *Zuweisung der Anwendungsfälle* und *Zuweisung Fachmodelle*. Hier erfolgt die Verknüpfung der definierten Anwendungsfälle mit den zugehörigen Projektphasen. Das Untermenü *Zuweisung Fachmodelle* ermöglicht die Verknüpfung der einzelnen Datensätze aus dem Untermenü *Alle Komponentenvorlagen*. Die Verknüpfung erfolgt hierarchisch, das heißt, dass zuerst dem jeweiligen Fachmodell die geforderten *Entities* zugewiesen werden. Den *Entities* untergeordnet sind die zugehörigen *Psets* und diesen wiederum deren *Properties*.

Im Menü *Projektanforderungen* werden die, einem Fachmodell verknüpften, Strukturen den jeweiligen Projektphasen zugewiesen. Konkret wird angegeben, welche *Properties* in welcher Projektphase ein Teil der Informationsanforderungen sind. Auch in diesem Menü gibt es unterstützende Funktionen wie *Suchen & Filtern*, *Mehrfachzuweisung* etc.

Das Menü *Dokumentation* ermöglicht das Exportieren der erstellten Anforderungen. Die möglichen Formate lauten *Berichte*, *Softwarevorlagen* und *Prüfregeln*. Weitere Erläuterungen und Screenshots werden in den folgenden Abschnitten, im Zuge der Beschreibung der Umsetzung der einzelnen Prozessschritte gezeigt.

5.2.2 Property-Silo erstellen

Am Anfang des, in Abschnitt 5.1 definierten, konkreten Prozesses steht die Aufgabe *Property-Silo erstellen*. Als Input-Daten sind hierfür die gewünschten *Properties* inkl. deren Definitionen erforderlich. Das Anlegen von Daten erfolgt in *BIMQ* im Menü *Vorlagen*. Dieses Menü ermöglicht eine tabellarische Auflistung der *Properties*.

Für das Anlegen der Daten stehen mehrere Möglichkeiten zur Verfügung:

- manuelles Eingeben in jeder Zelle
- die Funktion *Import aus Vorlage*
- die Funktion *Excel Import/Export*

Das manuelle Eingeben wird nicht weiter verfolgt, da dies aufgrund der großen Anzahl einzugebender Daten (mehrere Tausend Zellen) dem Ziel der möglichst zeiteffizienten Umsetzung widerspricht. Bei der Funktion *Import aus Vorlage* können bestehende Datensätze aus anderen *BIMQ*-Projekten importiert werden. Da hier eine eigene Vorlage erstellt und der Prozess dafür dargestellt wird, wird diese Funktion vernachlässigt. Die Funktion *Excel Import/Export* bietet für diesen Prozessschritt mehrere Vorteile. In Excel können Funktionen sehr einfach auf eine Vielzahl an Zellen angewendet werden. Zum Beispiel kann mit einer simplen Funktion der Name der *Properties* von einer Spalte in eine andere übertragen werden. Die Vorgehensweise mit Excel bildet eine mögliche Vorgehensweise von BIM-Anwendern ab, die Informationsanforderungen in Excel zur Verfügung haben und diese in *BIMQ* integrieren möchten.

Bei der Nutzung der *BIMQ*-Funktion *Excel Import/Export* werden die Daten zuerst in einer Excel-Datei erstellt, die in weiterer Folge nach *BIMQ* importiert wird. Das *BIMQ*-Menü *Vorlagen* fordert eine gewisse Struktur der hier angelegten Daten. Das heißt, dass die einzelnen Spalten eine feste Reihenfolge aufweisen müssen und für einen bestimmten Inhalt vorgesehen sind. Diese Struktur ist ebenfalls in der Excel-Datei zu berücksichtigen. Um diese Strukturierung zu erfüllen, empfiehlt *BIMQ* die Funktion *Export Vorlagen* zu verwenden. Dabei wird die, allenfalls noch inhaltslose, Tabelle aus dem *BIMQ*-Menü *Vorlagen* in eine Excel-Datei exportiert². In der exportierten Excel-Datei ist dadurch die erforderliche Struktur der Spalten bereits vorhanden. Abbildung 5.4 zeigt die möglichen Einstellungen beim Exportieren³. Hier kann ausgewählt werden welche Spalten exportiert werden sollen. Die Spalten *Code* und *Typ* werden automatisch exportiert. Für die Erstellung des *Property*-Silos empfehlen sich zusätzlich die Spalten *Einheiten*, *Allplan* und *IFC 4.3*. Unter *Einheiten* wird die Einheit des jeweiligen *Property* angegeben. In den Spalten *Allplan* und *IFC 4.3* sind die sogenannten *Mappings* einzutragen.

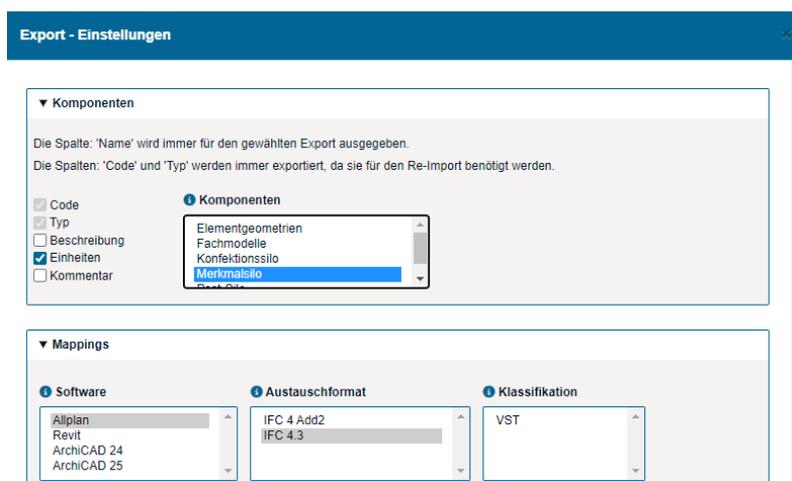
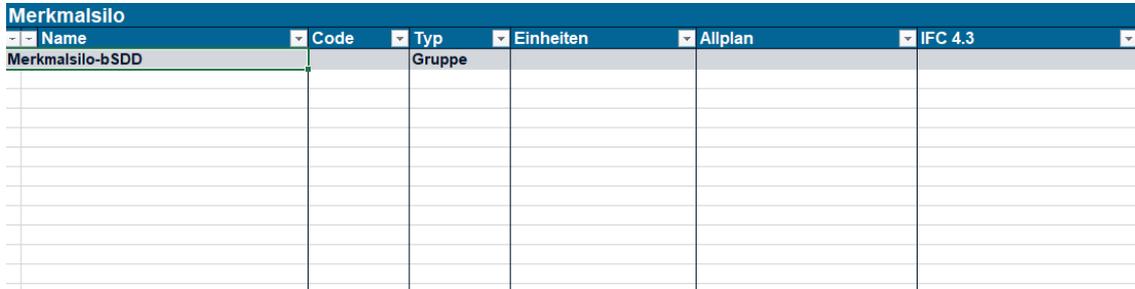


Abb. 5.4: Funktion *Export Vorlagen* im *BIMQ*-Menü *Vorlagen*

²Es empfiehlt sich den Eintrag der obersten Hierarchieebene zuvor in *BIMQ* hinzuzufügen. Dadurch kann dieser beim Export ausgewählt werden. Hier ist dies der Eintrag *Merkmalsilo* mit dem Typ *Gruppe*.

³*BIMQ*-Version 2.7.5-1, 2023-01-12

Die exportierte Excel-Datei hat nun die in Abbildung 5.5 dargestellte Form. Es ist ersichtlich, dass die in den Export-Einstellungen ausgewählten Spalten nun in Excel als Spalten vorhanden sind. In den Zeilen können die *Properties* und deren Definitionen eingefügt werden.



Name	Code	Typ	Einheiten	Allplan	IFC 4.3
Merkmalsilo-bSDD		Gruppe			

Abb. 5.5: Aus BIMQ exportierte Excel-Datei

Die Quelle dieser Daten kann beispielsweise das *bSDD* sein. Da in dieser Diplomarbeit gemäß dem Anwendungsfall eine Standardkonfiguration entwickelt wird und die standardisierten *Psets* im Fokus stehen, wird als Quelle die *IFC Specifications Database* [13] von *buildingSMART International (bSI)* verwendet. Hier stehen unter anderem alphabetische Listen aller *Properties* und *Psets* der IFC 4.3.1.0-Version zur Verfügung. In der Liste in [13] sind jedoch lediglich die Namen der *Properties* und nicht deren vollständigen Definitionen angeführt. Insgesamt sind folgende Angaben bzw. Definitionen eines *Property* in BIMQ essenziell:

- *Name*
- *Typ*
- *Einheit*
- *Allplan-Mapping*
- *IFC-Mapping*

Als *Name* des *Property* wird unverändert die englische, von *bSI* festgelegte, Bezeichnung verwendet. Unter *Typ* wird in BIMQ bestimmt, ob der Eintrag einer *Gruppe*, *Modell*, *Element*, *Eigenschaft*, *Geometrie* oder *Wert* entspricht. Ein *Property* wird als *Eigenschaft* klassifiziert.

Für die Festlegung der Einheit sind die Erläuterungen in Abschnitt 2.5 hilfreich. Die essenzielle Angabe bei einem *Property* der Klasse *IfcPropertySingleValue* ist der *Name*. Optional sind weiters eine *Specification* (Beschreibung), der *NominalValue* (*Data Type* und Wert) und *Unit* (Einheit). In Tabelle 5.1 sind diese Definitionen durch 3 Beispiele veranschaulicht. Das *Property LoadBearing* besitzt den *Data Type* bzw. *NominalValue IfcBoolean*, was die Werte *TRUE* oder *FALSE* zulässt. Der *Data Type IfcLabel* beim *Property FireRating*⁴ ermöglicht die freie Eingabe eines alphanumerischen Wertes, z. B. *R 90*. Für einen numerischen Wert, wie beim *Property YoungModulus* (Elastizitätsmodul), stehen eine Vielzahl an entsprechenden *Data Type* zur Verfügung. Dessen Namen beinhalten meistens das Suffix *Measure*, wie beispielsweise *IfcModulusOfElasticityMeasure*. Hier kann eine reelle Zahl als *NominalValue* eingegeben werden. Die zugehörigen *Units* (Einheiten) orientieren sich an den SI-Einheiten.

Neben der Definition eines *Property* anhand der Klasse *IfcPropertySingleValue* existiert die Klasse *IfcPropertyEnumeratedValue*. Bei einem *Property* dieser Klasse kann der Wert aus einer

⁴Das *Property FireRating* wird häufig auch als *IfcPropertyEnumeratedValue* abgebildet. Dabei wird eine Liste an möglichen Optionen für den Wert des *Property* definiert.

Tab. 5.1: Beispiele für *Properties* der Klasse *IfcPropertySingleValue*

Name	Data Type bzw. NominalValue	Wert (Beispiel)	Unit (Einheit)
<i>LoadBearing</i>	<i>IfcBoolean</i>	<i>TRUE</i>	—
<i>FireRating</i>	<i>IfcLabel</i>	R 90	—
<i>YoungModulus</i>	<i>IfcModulusOfElasticityMeasure</i>	210 000 000 000	N/m ²

festgelegten Liste möglicher Werte ausgewählt werden. Beim *Property Status* ist diese Liste mit *PEnum_ElementStatus* benannt und bietet folgende, standardisierte Werte: *Demolish* (Abbruch), *Existing* (Bestand), *New* (Neu), *Temporary* (Temporär), *Other* (Andere), *Notknown* (Nicht bekannt) oder *Unset* (Nicht definiert). Die weiteren Subklassen von *IfcSimpleProperty* sind *IfcPropertyBoundedValue*, *IfcPropertyListValue*, *IfcPropertyReferenceValue* und *IfcPropertyTableValue*. Mithilfe dieser können entsprechend deren Namen Grenzwerte, Listenwerte, Objektreferenzierungen oder Tabellen als Werte der *Properties* festgelegt werden. In *BIMQ* ist es lediglich möglich, *Properties* nach den Klassen *IfcPropertySingleValue* und *IfcPropertyEnumeratedValue* zu definieren.

Die Definitionen *Data Type* und *Unit* werden bei *IfcPropertySingleValue* in *BIMQ* in der Spalte *Einheiten* festgelegt. Die Definition eines *IfcPropertyEnumeratedValue Property* erfolgt durch Freilassen der Spalte *Einheiten* und dem Hinzufügen von untergeordneten Einträgen, die in der Spalte *Typ* mit *Wert* gekennzeichnet werden.

Die noch nicht erläuterten beiden essenziellen Angaben für die Definition eines *Property* in *BIMQ* sind die sogenannten *Mappings*. Diese legen die zugehörige Definition des *Property* in der BIM-Autorensoftware (hier *Allplan-Mapping*) bzw. im IFC-Schema fest und werden hier ident mit dem *Property*-Namen gewählt. Für das *IFC-Mapping* ist zusätzlich das Präfix #. zu ergänzen. Es ermöglicht später die korrekte Verknüpfung mit dem zugehörigen *Pset*.

Quelle der Daten

Die erforderlichen Angaben für die Definition eines *Property* in *BIMQ* wurden erläutert. Ziel ist es, diese Daten in die zuvor exportierte Excel-Datei einzufügen. Als Quelle der Daten steht der Anhang der *IFC 4.3 Dokumentation* [12] zur Verfügung. In *Annex B Alphabetical listings* sind alphabetisch geordnete Listen unter anderem der *Standard-Properties*, *Types* und *Property Enumerations* vorhanden. Die Zuordnung untereinander ist hier jedoch nicht angegeben bzw. in Tabellenform nicht kopierbar. In *Annex A Computer interpretable listings* können die *IFC property and quantity set definition* als XML-Dateien heruntergeladen werden. Diese beinhalten die Definitionen der *Psets* sowie deren zugehörigen *Properties* und können nach Excel importiert werden. Code 5.1 zeigt einen Ausschnitt der XML-Datei *Pset_ColumnCommon*. Exemplarisch sind die Definitionen des *Property Slope* (Neigungswinkel, grau hinterlegt) dargestellt. Diese zeigen, dass es sich um ein *Property* der Klasse *IfcPropertySingleValue* handelt. Der *Data Type* ist *IfcPlaneAngleMeasure*.

Code 5.1: Ausschnitt aus der XML-Datei *Pset_ColumnCommon* [12]

```

1  [...]
2  <Name> Pset_ColumnCommon </Name>
3  [...]
4  <PropertyDefs >
5      [...]
6      <PropertyDef >
7          <Name> Slope </Name >

```

5.2 Prozessschritte in BIMQ

```

8      <Definition>Slope angle - relative to horizontal (0.0 degrees).The
      shape information is provided in addition to the shape
      representation and the geometric parameters used within. In
      cases of inconsistency between the geometric parameters and the
      shape properties, provided in the attached property, the
      geometric parameters take precedence. For geometry editing
      applications, like CAD: this value should be write-only.</
      Definition>
9      <PropertyType>
10     <Type PropertySingleValue >
11     <DataType type=" IfcPlaneAngleMeasure " />
12     </TypePropertySingleValue >
13 </PropertyType>
14 </PropertyDef >
15 [...]

```

Für jedes der Standard-*Psets* existiert eine solche XML-Datei, die nach Excel importiert werden kann. Abbildung 5.6 zeigt die Nutzung der Funktion *XML-Quelle* in der Registerkarte *Entwicklertools*⁵. Im nächsten Schritt ist eine *XML-Zuordnung* zu erstellen (siehe Abbildung 5.6 unten).

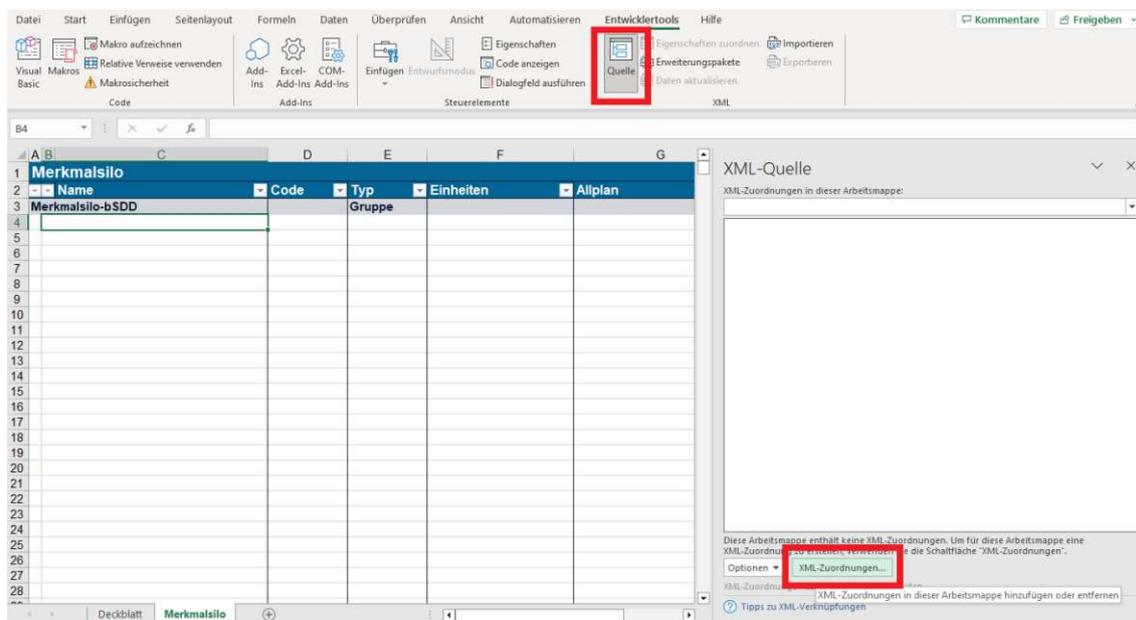


Abb. 5.6: Funktion *XML-Quelle* in Excel

Bei dieser Funktion sollte, für die beispielhafte Zuordnung, eine XML-Datei eines *Pset* ausgewählt werden, das *Properties* der Klassen *IfcPropertySingleValue* und *IfcPropertyEnumeratedValue* enthält. Bei der Zuordnung werden die zu importierenden Datenelemente angegeben. In Abbildung 5.7 wurde die XML-Datei *Pset_ColumnCommon.xml* hinzugefügt. Es ist ersichtlich, dass rechts die Datenelemente *Name (Pset_ColumnCommon)*, *Name (Reference)*, *type (IfcIdentifier)*, *name (PEnum_ElementStatus)* und *EnumItem (DEMOLISH)* ausgewählt wurden. Diese werden bei gehaltener STRG-Taste zu der gewünschten Zelle der Tabelle gezogen, hier die Zelle *B4*⁶. Dies ist erforderlich, damit die Positionierung der einzelnen Datenelemente untereinander erhalten

⁵Diese Registerkarte muss möglicherweise zuerst in den *Optionen* unter *Menüband anpassen* eingeblendet werden.

⁶Zuvor wurde die Funktion *Filtern* in der Registerkarte *Daten* sowie die Verbindung der Spalten *B* und *C* aufgehoben.

bleibt. Es wurden zusätzlich die Namen des *Pset* und der *Enumeration* eingefügt, um eine Überprüfung der Daten durchführen zu können. Die beiden Spalten werden zu einem späteren Zeitpunkt wieder entfernt werden.

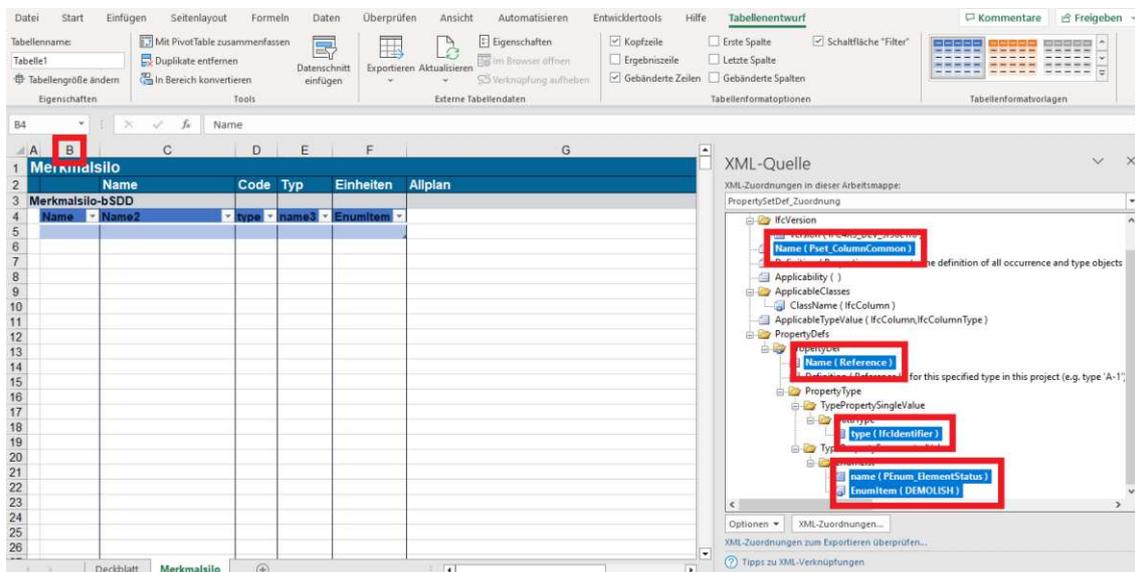


Abb. 5.7: Zuordnen in Excel

Die einzelnen Datenelemente sind nun in nebeneinanderliegenden Spalten angeordnet und nach dem Importieren aller XML-Dateien in die vorgesehene Spalte zu verschieben⁷. Unter der Registerkarte *Entwicklertools* können mit der Funktion *Importieren* alle XML-Dateien der *Psets* importiert werden. In Abbildung 5.8 ist diese Funktion ersichtlich (rot markiert). Ebenfalls ist das Ergebnis der importierten Dateien zu sehen.

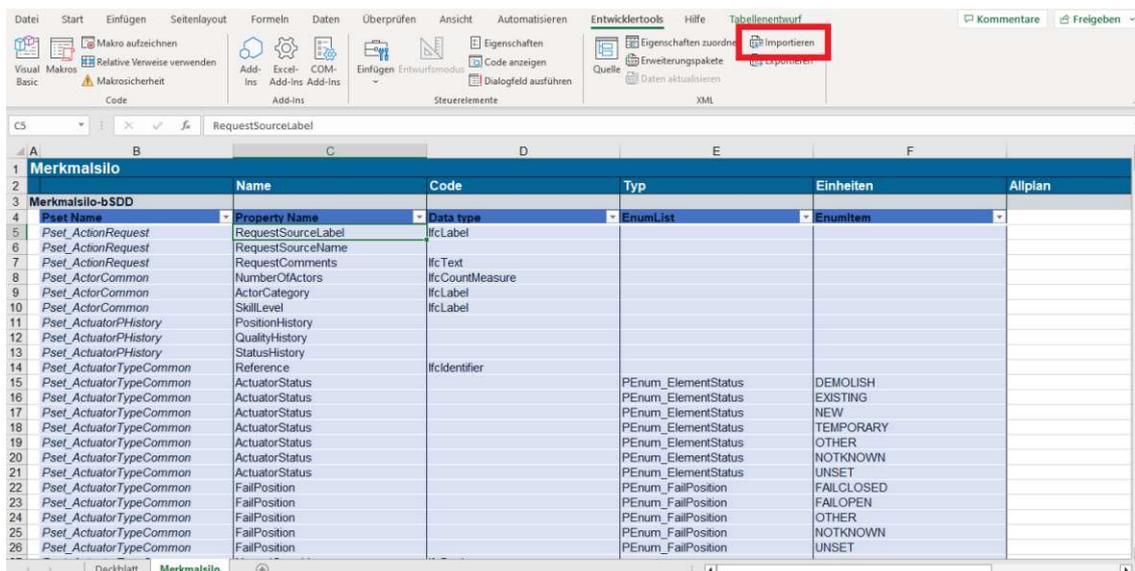


Abb. 5.8: Importieren der XML-Dateien nach Excel

⁷In der sich öffnenden Registerkarte *Tabellenentwurf* kann unter *Tabellenformatoptionen* die Option *Gebänderte Zeilen* deaktiviert werden. Dadurch entfällt die farbliche Hervorhebung der Zeilen.

In weiterer Folge werden lediglich die *Properties* der Klassen *IfcPropertySingleValue* und *IfcPropertyEnumeratedValue* berücksichtigt, da lediglich diese beiden in *BIMQ* abgebildet werden können. Dementsprechend sind importierte *Properties*, die in den Spalten *Data Type*⁸ und *EnumItem* keinen Eintrag enthalten, *Properties* mit Definitionen nach anderen Klassen. Um diese zu entfernen, können die Spalten *Data type* und *EnumItem* nacheinander nach leeren Zellen gefiltert werden. Zeilen, die nach dieser Filterung übrig bleiben, werden gelöscht.

Im nächsten Schritt wird die Spalte *Property Name* von *A bis Z* sortiert. Dies macht mehrfache Einträge gleichnamiger *Properties* sichtbar, was zu erwarten ist, da gleichnamige *Properties* unterschiedlichen *Psets* angefügt sind. In Abbildung 5.9 ist beispielsweise das *Property AccessCoverLoadRating* markiert, das 3 Mal bei unterschiedlichen *Psets* vorkommt.

Merkmal	Name	Code	Typ	Einheiten
Merkmal: silo-bSDD				
Pset_BerthCommon	AbnormalBerthingFactor	IfcPositiveRatioMeasure		
Pset_MarineVehicleCommon	AboveDeckProjectedWindEnd	IfcAreaMeasure		
Pset_MarineVehicleCommon	AboveDeckProjectedWindSide	IfcAreaMeasure		
Pset_BuildingStoreyCommon	AboveGround	IfcLogical		
Pset_InstallationOccurrence	AccessCoverLoadRating	IfcText		
Pset_DistributionChamberElementTypeFormedDuct	AccessCoverLoadRating	IfcText		
Pset_DistributionChamberElementTypeInspectionChamber	AccessCoverLoadRating	IfcText		
Pset_DistributionChamberElementTypeManhole	AccessLengthOrRadius	IfcPositiveLengthMeasure		
Pset_DistributionChamberElementTypeInspectionChamber	AccessLengthOrRadius	IfcPositiveLengthMeasure		
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	LOOSECOVER
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	MANHOLE
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	NONE
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	SECUREDCOVER
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	SECUREDCOVERWITHMANHOLE
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	OTHER
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	NOTKNOWN
Pset_TankTypeCommon	AccessType		PEnum_TankAccessType	UNSET
Pset_DistributionChamberElementTypeInspectionChamber	AccessWidth	IfcPositiveLengthMeasure		
Pset_DistributionChamberElementTypeManhole	AccessWidth	IfcPositiveLengthMeasure		
Pset_MaintenanceStrategy	AccidentResponse		PEnum_AccidentResponse	EMERGENCYINSPECTION
Pset_MaintenanceStrategy	AccidentResponse		PEnum_AccidentResponse	EMERGENCYPROCEDURE

Abb. 5.9: Darstellung gleichnamiger *Properties* der *IFC Specification* bei der Strukturierung in Excel

Bevor die mehrfachen Einträge gleichnamiger *Properties* entfernt werden, wird überprüft, ob *Properties* existieren, die zwar denselben Namen, aber einen unterschiedlichen *Data Type* haben. Mit Eingabe der Funktion

`=WENN(C5=C6;WENN(D5=D6;"";"Achtung");WENN(C5=C4;WENN(D5=D4;"";"Achtung");""))`

in der Spalte *G* wird getestet, ob im ersten Schritt in der Spalte *Property Name* in der Zeile davor oder danach ein identer Wert vorhanden ist. Falls ein identer Wert vorhanden ist, wird zusätzlich getestet, ob ebenfalls die Werte in der Spalte *Data type* ident sind. Falls diese nicht ident sind, wird *Achtung* ausgegeben und damit angezeigt, dass die *Properties* zwar denselben Namen, aber unterschiedliche *Data Type* aufweisen. In Abbildung 5.10 ist zu sehen, dass die Funktion 21 Mal (von 6239 Zeilen) den Wert *Achtung* ausgegeben hat. Diese Zeilen wurden gefiltert. Zwei Arten von Differenzen werden dadurch sichtbar. Es existieren gleichnamige *Properties*

- mit Definitionen nach unterschiedlichen Klassen. In Abbildung 5.10 ist beispielsweise das *Property AntennaType* hervorgehoben. Dies existiert einmal mit *Enumeration*-Werten und einmal als *IfcPropertySingleValue* mit *Data Type IfcLabel*.
- mit unterschiedlichem *Data Type*. Das *Property NominalHeight* gibt es mit *Data Type IfcPositiveLengthMeasure* und *IfcNonNegativeLengthMeasure*.

In weiterer Folge werden beide der unterschiedlichen Definitionen weiterverwendet, um später die zugehörigen *Psets* korrekt abbilden zu können. Die mehrfach vorhandenen und identen Einträge können nun jedoch entfernt werden. Unter Berücksichtigung der Spalten *Property Name*, *Data type* und *EnumItem* kann dafür die Funktion *Duplikate entfernen* verwendet werden.

⁸Die Kopfzeilen der Spalten wurde nach dem Import für eine bessere Nachvollziehbarkeit unbenannt.

Merkmalsilo	Name	Code	Typ	Einheiten	Allplan	IFC 4.3
Merkmalisilo-bSDD	Property Name	Data type	EnumList	EnumItem	Überprüfung	
Pset_CommunicationsApplianc	AntennaType	IfcLabel	PEnum_AntennaType	UNSET	Achtung	
Pset_MobileTeleCommunicatio	AntennaType	IfcLabel	PEnum_AntennaType	UNSET	Achtung	
Pset_ControllerTypeTwoPositi	ControlType	IfcLabel	PEnum_ControllerTwoPositionType	UNSET	Achtung	
Pset_DamperTypeSmokeDamp	ControlType	IfcLabel	PEnum_ControllerTwoPositionType	UNSET	Achtung	
Pset_FurnitureTypeCommon	NominalHeight	IfcPositiveLengthMeasure			Achtung	
Pset_KerbStone	NominalHeight	IfcNonNegativeLengthMeasure			Achtung	
Pset_MarkerGeneral	NominalHeight	IfcPositiveLengthMeasure			Achtung	
Pset_SwitchingDeviceTypeRel	NominalHeight	IfcNonNegativeLengthMeasure			Achtung	
Pset_CourseCommon	NominalLength	IfcNonNegativeLengthMeasure			Achtung	
Pset_FittingTransition	NominalLength	IfcPositiveLengthMeasure			Achtung	
Pset_DistributionPortTypeDuct	NominalWidth	IfcNonNegativeLengthMeasure			Achtung	
Pset_ElementSize	NominalWidth	IfcPositiveLengthMeasure			Achtung	
Pset_Address	Purpose	IfcLabel	PEnum_AddressType	UNSET	Achtung	
Pset_OpeningElementCommon	Purpose	IfcLabel	PEnum_AddressType	UNSET	Achtung	
Pset_RampFlightCommon	Reference	IfcIdentifier			Achtung	
Pset_ReinforcementBarPitchO	Reference	IfcLabel			Achtung	
Pset_MechanicalBeamOutOfPl	TensileStrength	IfcPressureMeasure			Achtung	
Pset_SectionInsulator	TensileStrength	IfcForceMeasure			Achtung	
Pset_SpringTensioner	TensileStrength	IfcPressureMeasure			Achtung	
Pset_ProtectiveDeviceTypeFus	TransformationRatio	IfcRatioMeasure			Achtung	
Pset_SwitchingDeviceTypeEme	TransformationRatio	IfcPositiveRatioMeasure			Achtung	

Abb. 5.10: Darstellung gleichnamiger *Properties* der *IFC Specification*, die einen unterschiedlichen *Data Type* oder *Property type* aufweisen

Bevor die Tabelle passend für *BIMQ* umstrukturiert wird, wird überprüft, ob gleichnamige *Properties* mit unterschiedlichen *Property Enumerations* vorkommen. Hierfür wird ebenfalls eine kurze Funktion ausgeführt:

```
=WENN(C5=C6;WENN(E5<>E6;"Achtung";WENN(C5=C4;WENN(E5<>E4;"Achtung";"");""));
WENN(C5=C4;WENN(E5<>E4;"Achtung";"");""))
```

Die Funktion überprüft im ersten Schritt, ob in der Spalte *Property Name* in der Zeile davor oder danach ein identer Wert vorhanden ist. Falls ein identer Wert vorhanden ist, wird getestet, ob die Werte in der Spalte *EnumList* voneinander verschieden sind. Falls diese verschieden sind, wird *Achtung* ausgegeben und damit angezeigt, dass die *Properties* zwar denselben Namen, aber unterschiedliche *Property Enumerations* aufweisen. Die Überprüfung fällt negativ aus. Es werden 18 Zeilen mit dem Wert *Achtung* ausgegeben. In Abbildung 5.11 sind diese Differenzen ersichtlich. Beim *Property AntennaType* zeigt die Funktion an, dass, wie bekannt, Definitionen nach unterschiedlichen Klassen existieren. Das *Property ControlType* gibt es 4 Mal mit unterschiedlichen *Property Enumerations* und ein weiteres Mal mit einer Definition als *IfcPropertySingleValue*. Die unterschiedlichen Definitionen werden vorerst weiterverwendet, um später die zugehörigen *Psets* korrekt abbilden zu können.

Merkmalsilo	Name	Code	Typ	Einheiten	Allplan	IFC 4.3
Merkmalisilo-bSDD	Property Name	Data type	EnumList	EnumItem	Überprüfung	
Pset_CommunicationsApplianc	AntennaType	IfcLabel	PEnum_AntennaType	UNSET	Achtung	
Pset_MobileTeleCommunicatio	AntennaType	IfcLabel	PEnum_AntennaType	UNSET	Achtung	
Pset_DistributionPortTypeDuct	ConnectionType	IfcLabel	PEnum_DuctConnectionType	NOTDEFINED	Achtung	
Pset_DistributionPortTypePipe	ConnectionType	IfcLabel	PEnum_PipeEndStyleTreatment	BRAZED	Achtung	
Pset_ControllerTypeFloating	ControlType	IfcLabel	PEnum_ControllerTypeFloating	UNSET	Achtung	
Pset_ControllerTypeProgramm	ControlType	IfcLabel	PEnum_ControllerTypeProgrammable	PRIMARY	Achtung	
Pset_ControllerTypeProgramm	ControlType	IfcLabel	PEnum_ControllerTypeProgrammable	SECONDARY	Achtung	
Pset_ControllerTypeProportion	ControlType	IfcLabel	PEnum_ControllerProportionType	PROPORTIONAL	Achtung	
Pset_ControllerTypeProportion	ControlType	IfcLabel	PEnum_ControllerProportionType	PROPORTIONALINTEGRALDE	Achtung	
Pset_ControllerTypeTwoPositi	ControlType	IfcLabel	PEnum_ControllerTwoPositionType	AND	Achtung	
Pset_ControllerTypeTwoPositi	ControlType	IfcLabel	PEnum_ControllerTwoPositionType	XOR	Achtung	
Pset_DamperTypeSmokeDamp	ControlType	IfcLabel	PEnum_ControllerTwoPositionType	UNSET	Achtung	
Pset_BoilerTypeCommon	EnergySource	IfcLabel	PEnum_EnergySource	UNSET	Achtung	
Pset_EngineTypeCommon	EnergySource	IfcLabel	PEnum_EngineEnergySource	BIFUEL	Achtung	
Pset_CoolingTowerTypeComm	FlowArrangement	IfcLabel	PEnum_CoolingTowerFlowArrangement	UNSET	Achtung	
Pset_HeatExchangerTypeCom	FlowArrangement	IfcLabel	PEnum_HeatExchangerArrangement	MULTIPASS	Achtung	
Pset_Address	Purpose	IfcLabel	PEnum_AddressType	UNSET	Achtung	
Pset_OpeningElementCommon	Purpose	IfcLabel	PEnum_AddressType	UNSET	Achtung	

Abb. 5.11: Darstellung gleichnamiger *Properties* der *IFC Specification*, die unterschiedliche *Property Enumerations* aufweisen

Formatierung

Damit die Tabelle in *BIMQ* importiert werden kann, muss diese eine bestimmte Struktur erfüllen. Da die Spalte *Pset Name* nur zur Überprüfung diente und hier rein die *Properties* angelegt werden, kann diese nun entfernt werden. Die Spalte *EnumList* bleibt für den kommenden Schritt noch erhalten. Die Zelle *A3* mit dem Wert *Merkmalsilo-bSDD* gilt als übergeordnete Gruppe. Dementsprechend müssen die *Properties* dieser Zelle untergeordnet sein. Dies wird realisiert in dem die Spalte mit den *Property Namen* um eine Spalte, im Vergleich zur Zelle *Merkmalsilo-bSDD*, eingerückt wird. Ebenfalls müssen die *Enumerations* in der Spalte *EnumItem* den *Property Namen* untergeordnet sein. Die *Property Namen* werden also in Spalte *B Name* verschoben, die *Enumerations* in Spalte *C*. Die Spalte der *Data type* entspricht in *BIMQ* den Einheiten der *Properties* und wird demzufolge in Spalte *F Einheiten* verschoben. In der Registerkarte *Tabellenentwurf* kann mithilfe der Funktion *In Bereich konvertieren* die Tabelle in einen normalen Zellenbereich geändert und damit die Kopfzeile entfernt werden. Abbildung 5.12 zeigt den Zwischenstand dieser Formatierungsschritte. In Abbildung 5.12 ist außerdem ersichtlich, dass bei *Properties* der Klas-

Merkmalsilo		Code	Typ	Einheiten	Allplan	IFC 4.3
Name						
Merkmalsilo-bSDD						
AbnormalBerthingFactor				IfcPositiveRatioMeasure		
AboveDeckProjectedWindEnd				IfcAreaMeasure		
AboveDeckProjectedWindSide				IfcAreaMeasure		
AboveGround				IfcLogical		
AcceptanceDate				IfcDate		
AccessCoverLoadRating				IfcText		
AccessLengthOrRadius				IfcPositiveLengthMeasure		
AccessType	LOOSECOVER		PEnum_TankAccessType			
AccessType	MANHOLE		PEnum_TankAccessType			
AccessType	NONE		PEnum_TankAccessType			
AccessType	SECUREDCOVER		PEnum_TankAccessType			
AccessType	SECUREDCOVERWITHMANHOLE		PEnum_TankAccessType			
AccessType	OTHER		PEnum_TankAccessType			
AccessType	NOTKNOWN		PEnum_TankAccessType			
AccessType	UNSET		PEnum_TankAccessType			
AccessWidth				IfcPositiveLengthMeasure		
AccidentResponse	EMERGENCYINSPECTION		PEnum_AccidentResponse			
AccidentResponse	EMERGENCYPROCEDURE		PEnum_AccidentResponse			

Abb. 5.12: Zwischenstand der Formatierung in Excel

se *IfcPropertyEnumeratedValue*, z. B. *AccessType*, die *Enumeration Items* (Spalte *C*) in derselben Zeile wie die *Property Namen* stehen. Dies würde beim Import in *BIMQ* zu einer Fehlermeldung führen, da in jeder Zeile nur ein Eintrag mit einem eindeutigen *Typ*, wie *Eigenschaft* für *Properties* oder *Wert* für die *Enumeration Items*, enthalten sein darf. Um dies Umzusetzen muss über den *Property Enumerations* eine Zeile eingefügt werden, in der allein der zugehörige *Property*-Name steht. In Excel kann dies mit einem *Makro* realisiert werden. Code 5.2 zeigt den dafür verwendeten Code. Im Code wird im ersten Schritt überprüft, ob in der Spalte *Property Name* in der Zeile danach ein identer Wert vorhanden ist. Falls ein identer Wert vorhanden ist, bedeutet das, dass es sich um ein *Property* mit *Property Enumerations* handelt und es wird darüber eine Zeile eingefügt. In die eingefügte Zeile wird lediglich der *Property Name* geschrieben. Im Code wird mittels des Zählers *i* und *If*-Schleifen berücksichtigt, dass je *Property* nur ein Mal eine Zeile eingefügt wird. Der Code kann in Excel ausgeführt werden, indem in der Registerkarte *Entwicklertools* die Funktion *Makros* verwendet wird. In dem sich öffnenden Fenster kann ein neuer Makro mit Code 5.2 erstellt und anschließend ausgeführt werden.

Code 5.2: Makro zum Einfügen einer Zeile in Excel

```

1 Sub einf_Zeile()
2 ' Einfügen einer Zeile bei doppeltem Property Name
3 ' z entspricht der ersten Zeile des Property Namen
4     z = 4
5     ' s entspricht der ersten Spalte des Property Namen
6     s = 2

```

```

7     i = 0
8     Do
9     If Cells(z, s).Value = Cells(z + 1, s).Value And i = 0 Then
10    Rows(z).Insert Shift:=xlDown
11    Cells(z, s).Value = Cells(z + 1, s).Value
12    z = z + 1
13    i = i + 1
14    ElseIf Cells(z, s).Value <> Cells(z + 1, s).Value Then
15    z = z + 1
16    i = 0
17    ElseIf Cells(z, s).Value = Cells(z - 1, s).Value Then
18    z = z + 1
19    i = i + 1
20    Else
21    z = z + 1
22    i = 0
23    End If
24    Loop Until z = ActiveSheet.UsedRange.Rows.Count
25 End Sub

```

Nach Ausführen des Makros sieht die Tabelle wie in Abbildung 5.13 aus. Die rot markierten Zeilen wurden durch Ausführen des Makro ergänzt. Es ist ersichtlich, dass der *Property*-Name nun in einer eigenen Zeile steht. Das Einfügen einer Zeile muss bei den vorher erkannten, gleichnamigen *Properties* mit unterschiedlichen *Enumeration list* manuell durchgeführt werden. Die betroffenen 3 *Properties* sind *ConnectionType*, *ControlType* und *FlowArrangement*.

Merkmalsilo						
Name		Code	Typ	Einheiten	Allplan	IFC 4.3
Merkmalsilo-bSDD						
AbnormalBerthingFactor				IfcPositiveRatioMeasure		
AboveDeckProjectedWindEnd				IfcAreaMeasure		
AboveDeckProjectedWindSide				IfcAreaMeasure		
AboveGround				IfcLogical		
AcceptanceDate				IfcDate		
AccessCoverLoadRating				IfcText		
AccessLengthOrRadius				IfcPositiveLengthMeasure		
AccessType						
AccessType	LOOSECOVER		PEnum_TankAccessType			
AccessType	MANHOLE		PEnum_TankAccessType			
AccessType	NONE		PEnum_TankAccessType			
AccessType	SECUREDCOVER		PEnum_TankAccessType			
AccessType	SECUREDCOVERWITHMANHOLE		PEnum_TankAccessType			
AccessType	OTHER		PEnum_TankAccessType			
AccessType	NOTKNOWN		PEnum_TankAccessType			
AccessType	UNSET		PEnum_TankAccessType			
AccessWidth				IfcPositiveLengthMeasure		
AccidentResponse						
AccidentResponse	EMERGENCYINSPECTION		IEnum_AccidentResponse			

Abb. 5.13: Tabelle nach Ausführen des Makros

Im nächsten Schritt wird die Spalte mit den *Property*-Namen um doppelte Einträge bereinigt, ohne dabei die entsprechenden Zeilen zu löschen. Hierfür wird eine *WENN*-Funktion mit Definition

$$=WENN(C4="" ; B4 ; "")$$

temporär in Spalte *D* eingetragen. Die Funktion übernimmt den *Property*-Namen aus Spalte *B*, wenn in der Spalte der *Property Enumerations* (Spalte *C*) kein Eintrag vorhanden ist. Die Hilfsspalte *D*, die nun rein die *Property*-Namen enthält, kann anschließend in die Spalte *Name* (Spalte *B*) kopiert werden. Bei den *Einfügeoptionen* ist *Werte* auszuwählen. Die Inhalte der Hilfsspalte *D* können nun gelöscht werden. Ebenso kann die Spalte mit den *Enumeration lists*, wie *PEnum_TankAccessType*, entfernt werden.

Um die Definitionen zu vervollständigen sind noch die Spalten *Code*, *Typ*, *Allplan* und *IFC 4.3* zu befüllen. Die Zuweisung eines Codes in *BIMQ* ist für den Prozess nicht essenziell, erleichtert jedoch die Handhabung. In ÖNORM EN ISO 23386 [24], die den Untertitel *Methodik zur Beschreibung, Erstellung und Pflege von Merkmalen in miteinander verbundenen Datenkatalogen* trägt,

wird unter den festgelegten Attributen für die Verwaltung eines Merkmals ein Code angeführt. Dieser soll der Identifizierung dienen. Außerdem werden bei Fehlermeldungen, wie z. B. in der exportierten Zuordnungsdatei, die von Fehlern betroffenen Einträge mit ihren jeweiligen Codes ausgegeben. Mit der Festlegung eines Codes kann also die Fehlerbeseitigung erleichtert werden, da der Code maßgebend für die Identifizierung ist und so der von Fehlern betroffene Eintrag schnell gefunden und korrigiert werden kann. Damit jedem *Property* ein fortlaufender Code und die *Enumerations* dem untergeordnet sind, wurde folgende Funktion entwickelt⁹:

```
=WENN(LÄNGE(B4)>0;VERKETTEN("M-";
TEXT(ANZAHL2(B$4:$B4)-ANZAHLLEEREZELLEN(B$4:$B4);"0000"));
WENN(LÄNGE(B3)>0;VERKETTEN("M-";
TEXT(ANZAHL2(B$4:$B4)-ANZAHLLEEREZELLEN(B$4:$B4);"0000");
"-";TEXT(1;"00"));VERKETTEN("M-";
TEXT(ANZAHL2(B$4:$B4)-ANZAHLLEEREZELLEN(B$4:$B4);"0000");
"-";TEXT(ZAHLENWERT(TEIL(D3;8;2))+1;"00"))))
```

Die Funktion generiert einen Code mit vorangestelltem Buchstaben M (für Merkmal) und einer fortlaufenden Nummer für jedes *Property*. Den *Enumerations* wird ebenfalls eine fortlaufende Nummer abhängig vom zugehörigen *Property* zugewiesen. Das Zwischenergebnis ist in Abbildung 5.14 dargestellt.

Name	Code	Typ	Einheiten	Allplan	IFC 4.3
AbnormalBerthingFactor	M-0001		IfcPositiveRatioMeasure		
AboveDeckProjectedWindEnd	M-0002		IfcAreaMeasure		
AboveDeckProjectedWindSide	M-0003		IfcAreaMeasure		
AboveGround	M-0004		IfcLogical		
AcceptanceDate	M-0005		IfcDate		
AccessCoverLoadRating	M-0006		IfcText		
AccessLengthOrRadius	M-0007		IfcPositiveLengthMeasure		
AccessType	M-0008				
	LOOSECOVER	M-0008-01			
	MANHOLE	M-0008-02			
	NONE	M-0008-03			
	SECUREDCOVER	M-0008-04			
	SECUREDCOVERWITHMANH	M-0008-05			
	OTHER	M-0008-06			
	NOTKNOWN	M-0008-07			
	UNSET	M-0008-08			
AccessWidth	M-0009		IfcPositiveLengthMeasure		
AccidentResponse	M-0010				
	EMERGENCYINSPECTION	M-0010-01			
	EMERGENCYPROCEDURE	M-0010-02			

Abb. 5.14: Zwischenergebnis der Tabelle in Excel

In der Spalte *Typ* ist für *Properties* der Wert *Eigenschaft* und für *Enumerations* Wert hinzuzufügen. Dafür wird ebenfalls eine *WENN*-Funktion verwendet, die prüft ob in der Spalte *Name* ein Eintrag vorhanden ist und dementsprechend entweder *Eigenschaft* oder *Wert* ausgibt:

```
=WENN(LÄNGE(B4)>0;"Eigenschaft";"Wert")
```

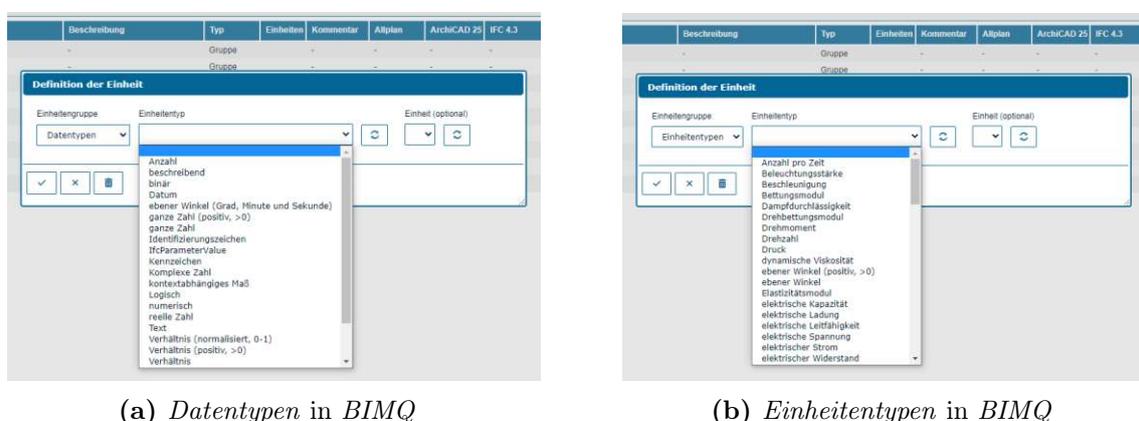
Die *Data Type* bzw. die *Einheit* des *Property* werden beim Import in *BIMQ* nicht als Angabe in einer IFC-Klasse (z. B. *IfcBoolean*) erkannt. Aus diesem Grund müssen die Einheiten manuell von der Definition im IFC-Schema zu der Definition von *BIMQ* geändert werden. Hierfür wird eine Liste mit den vorhandenen *Data Type* und den zugehörigen Schreibweise in *BIMQ* erstellt.

⁹Die Formel ist nur gültig, wenn die Zellen den Wert " " enthalten, da auf diese Zellen bereits zuvor Formeln angewendet wurden. Falls die Zellen *echt leer* sind, kann der Teil *-ANZAHLLEEREZELLEN(B\$4:\$B4)* entfallen.

Die in der Tabelle vorhandenen *Data Type* lassen sich mit folgender Funktion aus Spalte *F Einheiten* auslesen und alphabetisch ordnen:

```
=SORTIEREN(EINDEUTIG(F4:F4732))
```

Die zugehörigen Einheiten in *BIMQ* werden manuell ausgelesen. In *BIMQ* im Menü *Vorlagen* erscheint beim Anklicken einer Zelle in der Spalte *Einheiten* ein Fenster mit mehreren Auswahlmöglichkeiten. Zuerst kann bei den sogenannten *Einheitengruppen* zwischen *Datentypen* und *Einheitentypen* gewählt werden. Abbildung 5.15 zeigt diese Möglichkeiten. Mit den *Datentypen* werden grundsätzliche *Data Type* abgebildet. Beispielsweise der *Data Type IfcBoolean* wird in *BIMQ* mit *Wahr/Falsch* abgebildet. *Data Type* mit Suffix *Measure* sind in *BIMQ* unter den sogenannten *Einheitentypen* zu finden. Ein Beispiel ist *IfcForceMeasure* mit dem zugehörigen *Einheitentyp Kraft*.

(a) *Datentypen* in *BIMQ*(b) *Einheitentypen* in *BIMQ*Abb. 5.15: *Einheitengruppen* in *BIMQ*

Von den unterschiedlichen Auswahlmöglichkeiten in *BIMQ* werden die zugehörigen *Data Type* der *IFC Specification* abgeleitet¹⁰. In Tabelle 5.2 ist ein Ausschnitt der wichtigsten *Data Type* und deren zugehörigen Schreibweise bei der Eingabe in *BIMQ* angeführt. Die vollständige Tabelle aller *Data Type* ist im Anhang beigefügt, siehe Tabelle A.1.

Tab. 5.2: Einheiten-Zuweisung in *BIMQ*

<i>Data Type</i>	Schreibweise bei der Eingabe in <i>BIMQ</i>
IfcAreaMeasure	Fläche.m2
IfcBoolean	Wahr/Falsch
IfcCountMeasure	Anzahl
IfcForceMeasure	Kraft.N
IfcIdentifier	Identifizierungszeichen
IfcInteger	ganze Zahl
IfcLabel	Kennzeichen
IfcLengthMeasure	Länge
IfcLogical	Logisch

¹⁰Im späteren Prozessschritt *Softwarevorlage exportieren* wird eine sogenannte *Zuordnungsdatei* exportiert. In dieser Datei sind beispielsweise die *Data Type* der festgelegten *Properties* ersichtlich. Damit kann die korrekte Zuordnung der *Data Type* zur Schreibweise in *BIMQ* überprüft werden.

5.2 Prozessschritte in BIMQ

IfcModulusOfElasticityMeasure	Elastizitätsmodul.N/m ²
IfcNonNegativeLengthMeasure	Länge (nicht-negativ, >=0)
IfcPlanarForceMeasure	Flächenkraft.N/m ²
IfcPlaneAngleMeasure	ebener Winkel
IfcPositivePlaneAngleMeasure	ebener Winkel (positiv, >0)
IfcReal	reelle Zahl
IfcText	Text
IfcThermalTransmittanceMeasure	Wärmedurchgangskoeffizient.Watt / m ² Kelvin

Diese tabellarische Zuweisung der Einheiten kann in der vorliegenden Excel-Datei erstellt werden. Hierfür werden die Werte der beiden Spalten in eigenen Zellen in der Excel-Datei eingefügt, siehe Abbildung 5.16 rechts. Im nächsten Schritt wird die Zuweisung mittels XVERWEIS-Funktion übernommen. Die verwendete Formel überprüft im ersten Schritt, ob die Spalten *Name* und *Einheiten* einen Eintrag enthalten. Damit ist gewährleistet, dass es sich um eine Zeile mit einem *Property* handelt. Danach wird die XVERWEIS-Funktion angewendet, die den Wert aus der Spalte *Einheiten* mit der erstellten Zuweisung in Spalte *J* abgleicht und die zugehörige Schreibweise in *BIMQ*, aus Spalte *K* zurückgibt. Die eingegebene Formel lautet:

```
=WENN(LÄNGE(B4)>0;WENN(LÄNGE(F4)>0;XVERWEIS(F4;$J$4:$J$76;$K$4:$K$76;
"NICHT_GEFUNDEN");"Kennzeichen");"
```

Die Formel wurde temporär in der Spalte *I* eingetragen und auf alle belegten Zeilen angewendet. Bei einem *Property* der Klasse *IfcPropertyEnumeratedValue*, wie z. B. *AccessType*, ist in der Spalte *Einheiten* per Definition kein Wert vorhanden, da solche *Properties* eine Liste an auswählbaren Optionen zur Verfügung stellen. Hier ist in *BIMQ* die Vorgehensweise, dass solche *Properties* als *Einheit* den Wert *Kennzeichen* zugewiesen bekommen. Abbildung 5.16 zeigt den Zwischenstand der Excel-Datei mit den zugewiesenen Schreibweise für *BIMQ* in Spalte *I*. Die Werte der Spalte *I* können nun in die vorgesehene Spalte *F* *Einheiten* kopiert werden.

A	B	C	D	E	F	G	H	I	J	K
1	Merkmalslo									
2	Name	Code	Typ	Einheiten	Allplan	IFC 4.3				
3	Merkmalslo-bSDD									
4	AbnormalBerthngFactor	M-0001	Eigenschaft	#fPositiveRatioMeasure				Verhältnis (positiv, >0)	#fAccelerationMeasure	BIMQ Einheit
5	AboveDeckProjectedWindCnd	M-0002	Eigenschaft	#fArcAngleMeasure				Fläche m ²	#fAngularVelocityMeasure	Beschleunigung m/s ²
6	AboveDeckProjectedWindSide	M-0003	Eigenschaft	#fAreaMeasure				Fläche m ²	#fAreaDensityMeasure	Winkelgeschwindigkeit radian/s
7	AboveGround	M-0004	Eigenschaft	#fLogical				Logisch	#fAreaMeasure	Flächendichte kg/m ²
8	AcceptanceDate	M-0005	Eigenschaft	#fDate				Datum, Jahr, Monat und	#fBoolean	Wahr/Falsch
9	AccessCoverLoadRating	M-0006	Eigenschaft	#fText				Text	#fComplexNumber	Komplexe Zahl
10	AccessLengthOrRadius	M-0007	Eigenschaft	#fPositiveLengthMeasure				Länge (positiv, >0)	#fCountMeasure	Anzahl
11	AccessType	M-0008	Eigenschaft					Kennzeichen	#fDate	Datum, Jahr, Monat und Tag (YYYY-MM-DD)
12	LOOSECOVER	M-0008-01	Wert						#fDate Time	Zeitpunkt (Datum und Zeit)
13	MANHOLE	M-0008-02	Wert						#fDuration	Kennzeichen
14	NONE	M-0008-03	Wert						#fDynamicViscosityMeasure	dynamische Viskosität Pa s
15	SECUREDCOVER	M-0008-04	Wert						#fElectricCapacitanceMeasure	elektrische Kapazität F
16	SECUREDCOVERWITHMANH	M-0008-05	Wert						#fElectricChargeMeasure	elektrische Ladung C
17	OTHER	M-0008-06	Wert						#fElectricCurrentMeasure	elektrischer Strom A
18	NOTKNOWN	M-0008-07	Wert						#fElectricResistanceMeasure	elektrischer Widerstand V/A
19	UNSET	M-0008-08	Wert						#fElectricVoltageMeasure	elektrische Spannung V
20	AccessWidth	M-0009	Eigenschaft	#fPositiveLengthMeasure				Länge (positiv, >0)	#fEnergyMeasure	Energie J
21	AccidentResponse	M-0010	Eigenschaft					Kennzeichen	#fForceMeasure	Kraft N
22		EMERGENCYINSPECTION	M-0010-01	Wert					#fFrequencyMeasure	Frequenz Hz
23		EMERGENCYPROCEDURE	M-0010-02	Wert					#fHeatFluxDensityMeasure	Wärmestromdichte W/m ²

Abb. 5.16: Zwischenstand der Tabelle in Excel

In den übrigen beiden Spalten *Allplan* und *IFC 4.3* sind die sogenannten *Mappings* einzutragen. Diese werden ident zum Namen des *Property* gewählt, wobei beim *IFC 4.3-Mapping* das Präfix #. hinzugefügt wird. Für Einträge vom *Typ Wert (Enumerations)* ist kein *Mapping* erforderlich. Das Präfix wird mittels der Funktion VERKETTEN ergänzt. Verwendet werden folgende Formeln, die in die Spalte *Allplan* bzw. *IFC4.3* eingegeben werden:

```
=B4
```

```
=WENN(LÄNGE(B4)>0;VERKETTEN("#. ";B4);"
```

Ein Ausschnitt der finalen Excel-Datei des *Property-Silos* ist in Abbildung 5.17 ersichtlich.

Merkmalsilo						
Name	Code	Typ	Einheiten	Allplan	IFC 4.3	
Merkmalsilo-bSDD						
AbnormalBerthingFactor	M-0001	Eigenschaft	Verhältnis (positiv, >0)	AbnormalBerthingFactor	# AbnormalBerthingFactor	
AboveDeckProjectedWindEnd	M-0002	Eigenschaft	Fläche.m2	AboveDeckProjectedWindEnd	# AboveDeckProjectedWindEnd	
AboveDeckProjectedWindSide	M-0003	Eigenschaft	Fläche.m2	AboveDeckProjectedWindSide	# AboveDeckProjectedWindSide	
AboveGround	M-0004	Eigenschaft	Logisch	AboveGround	# AboveGround	
AcceptanceDate	M-0005	Eigenschaft	Datum.Jahr, Monat und Tag (YY)	AcceptanceDate	# AcceptanceDate	
AccessCoverLoadRating	M-0006	Eigenschaft	Text	AccessCoverLoadRating	# AccessCoverLoadRating	
AccessLengthOrRadius	M-0007	Eigenschaft	Länge (positiv, >0)	AccessLengthOrRadius	# AccessLengthOrRadius	
AccessType	M-0008	Eigenschaft	Kennzeichen	AccessType	# AccessType	
	LOOSECOVER	M-0008-01	Wert			
	MANHOLE	M-0008-02	Wert			
	NONE	M-0008-03	Wert			
	SECUREDCOVER	M-0008-04	Wert			
	SECUREDCOVERWITHMANH	M-0008-05	Wert			
	OTHER	M-0008-06	Wert			
	NOTKNOWN	M-0008-07	Wert			
	UNSET	M-0008-08	Wert			
AccessWidth	M-0009	Eigenschaft	Länge (positiv, >0)	AccessWidth	# AccessWidth	
AccidentResponse	M-0010	Eigenschaft	Kennzeichen	AccidentResponse	# AccidentResponse	
	EMERGENCYINSPECTION	M-0010-01	Wert			
	EMERGENCYPROCEDURE	M-0010-02	Wert			

Abb. 5.17: Finale Excel-Datei

In *BIMQ* kann die Excel-Datei im Menü *Vorlagen*, dem Untermenü *Alle Komponentenvorlagen* und unter *Excel Import/Export* mit der Funktion *Import Komponentenvorlagen* importiert werden, siehe Abbildung 5.18.



Abb. 5.18: Funktion *Import Komponentenvorlagen* für den Import von Excel-Dateien in *BIMQ*

5.2.3 Pset-Silo erstellen

Die Vorgehensweise beim Erstellen des *Pset-Silos* ist ebenfalls diesen zuerst in Excel aufzubereiten und danach in *BIMQ* zu importieren. Ein *Pset* ist eine Gruppierung von *Properties*. In diesem Sinn wird in *BIMQ* dem Eintrag eines *Pset* der *Typ Gruppe* zugewiesen. Die weiteren obligatorischen Definitionen sind die *Mappings*, also Software- und IFC-*Mapping*. Zur einfacheren Handhabung wird zusätzlich, und wie beim *Property-Silo*, ein *Code* für jeden Eintrag festgelegt. Die Standard-*Psets* können aus dem Anhang der *IFC 4.3 Dokumentation* [14] kopiert und in eine Excel-Tabelle eingefügt werden. Hier gibt es 640 Einträge an unterschiedlichen *Psets*. Abbildung 5.19 zeigt das Kopieren aus [14]. Als Vorlage für die Formatierung der Excel-Datei kann ebenfalls eine zuvor aus *BIMQ* exportierte Datei verwendet werden. *Code*, *Typ* und *Mappings* werden nach derselben Vorgehensweise wie in Abschnitt 5.2.2 eingefügt.

Der Code für die einzelnen *Psets* kann mit folgender Formel, die den Buchstaben *P* und eine fortlaufende Nummer generiert, erstellt werden:

```
=VERKETTEN("P-" ; TEXT(ANZAHL2(B$4 : $B4) ; "000"))
```

In Abbildung 5.20 ist ein Ausschnitt der fertiggestellten Excel-Datei dargestellt. Es ist ersichtlich, dass die Spalte *Einheiten* definitionsgemäß leer bleibt. Die *Allplan*- und IFC-*Mappings* werden ident zum *Pset*-Namen gewählt.

5.2 Prozessschritte in BIMQ

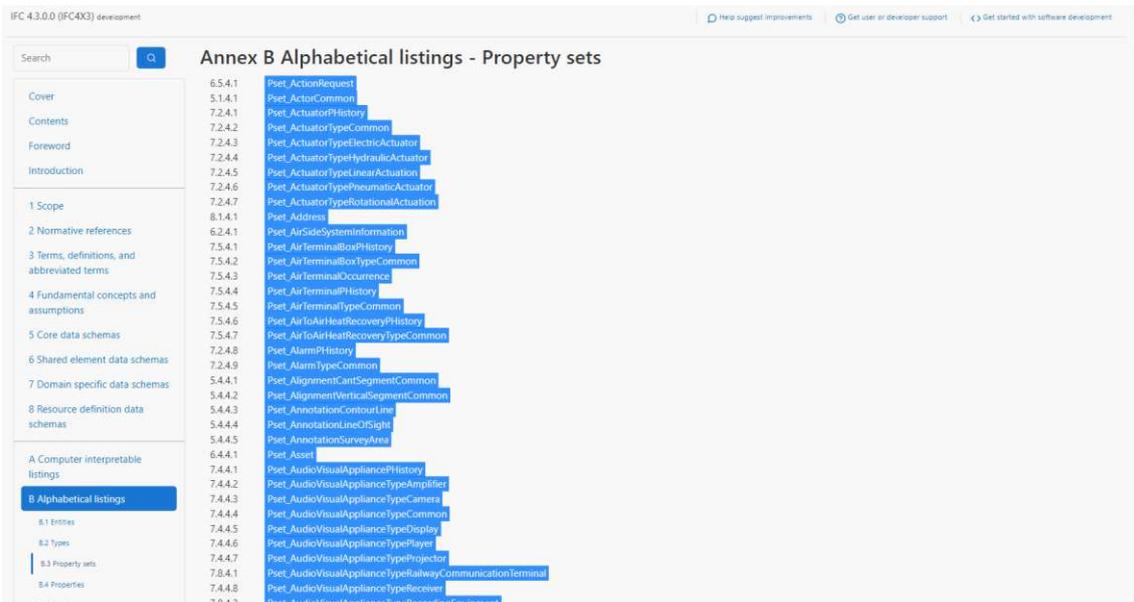


Abb. 5.19: Kopieren der Standard-*Psets* aus [14]

Name	Code	Typ	Einheiten	Allplan	IFC 4.3
Pset-Silo					
Pset_ActionRequest	P-001	Gruppe		Pset_ActionRequest	Pset_ActionRequest
Pset_ActorCommon	P-002	Gruppe		Pset_ActorCommon	Pset_ActorCommon
Pset_ActuatorPHistory	P-003	Gruppe		Pset_ActuatorPHistory	Pset_ActuatorPHistory
Pset_ActuatorTypeCommon	P-004	Gruppe		Pset_ActuatorTypeCommon	Pset_ActuatorTypeCommon
Pset_ActuatorTypeElectricActuator	P-005	Gruppe		Pset_ActuatorTypeElectricActuator	Pset_ActuatorTypeElectricActuator
Pset_ActuatorTypeHydraulicActuator	P-006	Gruppe		Pset_ActuatorTypeHydraulicActuator	Pset_ActuatorTypeHydraulicActuator
Pset_ActuatorTypeLinearActuation	P-007	Gruppe		Pset_ActuatorTypeLinearActuation	Pset_ActuatorTypeLinearActuation
Pset_ActuatorTypePneumaticActuator	P-008	Gruppe		Pset_ActuatorTypePneumaticActuator	Pset_ActuatorTypePneumaticActuator
Pset_ActuatorTypeRotationalActuation	P-009	Gruppe		Pset_ActuatorTypeRotationalActuation	Pset_ActuatorTypeRotationalActuation
Pset_Address	P-010	Gruppe		Pset_Address	Pset_Address
Pset_AirSideSystemInformation	P-011	Gruppe		Pset_AirSideSystemInformation	Pset_AirSideSystemInformation
Pset_AirTerminalBoxPHistory	P-012	Gruppe		Pset_AirTerminalBoxPHistory	Pset_AirTerminalBoxPHistory
Pset_AirTerminalBoxTypeCommon	P-013	Gruppe		Pset_AirTerminalBoxTypeCommon	Pset_AirTerminalBoxTypeCommon
Pset_AirTerminalOccurrence	P-014	Gruppe		Pset_AirTerminalOccurrence	Pset_AirTerminalOccurrence
Pset_AirTerminalPHistory	P-015	Gruppe		Pset_AirTerminalPHistory	Pset_AirTerminalPHistory
Pset_AirTerminalTypeCommon	P-016	Gruppe		Pset_AirTerminalTypeCommon	Pset_AirTerminalTypeCommon
Pset_AirToAirHeatRecoveryPHistory	P-017	Gruppe		Pset_AirToAirHeatRecoveryPHistory	Pset_AirToAirHeatRecoveryPHistory
Pset_AirToAirHeatRecoveryTypeCommon	P-018	Gruppe		Pset_AirToAirHeatRecoveryTypeCommon	Pset_AirToAirHeatRecoveryTypeCommon
Pset_AlarmPHistory	P-019	Gruppe		Pset_AlarmPHistory	Pset_AlarmPHistory
Pset_AlarmTypeCommon	P-020	Gruppe		Pset_AlarmTypeCommon	Pset_AlarmTypeCommon

Abb. 5.20: *Pset-Silo* in Excel

Silo-*bSAT*

Zu der Standardkonfiguration zählen auch die länderspezifischen Standardisierungen. Für den *Pset-Silo-bSAT* werden die auf der *buildingSMART Austria (bSAT)*-Seite downloadbaren Standard-AIA [10] verwendet. Daraus werden die *Psets* aller Architektur Klassen und bis *LOI 300* übernommen, jedoch nur solche, die noch nicht im zuvor erstellten Silo vorhanden sind. Dies sind *bSAT*-spezifische *Psets* wie *Pset_BeamSpecific*. Die entsprechenden Einträge werden manuell aus dem Dokument kopiert und ebenfalls in eine Vorlage-Excel für den Import in *BIMQ* aufbereitet.

Für den *Property-Silo-bSAT* werden die *Properties* aus der Standard-AIA [10] übernommen. Zum Beispiel das *bSAT*-spezifische *Property ElementMainMateriality*. Die entsprechenden Einträge werden manuell aus dem Dokument kopiert und in eine Vorlage-Excel für den Import in *BIMQ* aufbereitet. Die Zuweisung von *Code*, *Typ*, *Einheit* und *Mappings* entspricht der Vorgehensweise des vorangehenden Abschnittes 5.2.2.

5.2.4 Properties mit Psets verknüpfen

Der nächste Schritt im, in Abschnitt 4.3 dargestellten, optimierten Prozess ist die Verknüpfung der *Properties* mit den zugehörigen *Psets*. Diese Verknüpfung wäre in einer eigenen Struktur vorgesehen, die lediglich zum jeweiligen *Property*- und *Pset*-Silo referenziert. Die, in dieser Struktur zusammengestellten, *Psets* könnten in weiterer Folge mit den *Entities* verknüpft werden. In *BIMQ* ist diese Vorgehensweise nicht umsetzbar. Es muss auf das Menü *Projektkonfiguration* zurückgegriffen werden, wo jedes *Property* einzeln dem zugehörigen *Pset* zuzuweisen ist, um als Ursprung den *Property*-Silo zu behalten. Im Falle der hier erstellten Standardkonfiguration wären dies über 6000 manuelle Schritte. Da dies einem zeiteffizienten Prozess widerspricht und sehr fehleranfällig ist, wird auf dieselbe Vorgehensweise wie bei der Erstellung des *Property*- und *Pset*-Silo gesetzt. Die *Psets* werden dabei bereits im Menü *Vorlagen* als feste Einheit mit den zugehörigen *Properties* als untergeordnete Einträge angelegt. Dadurch gibt es idente Einträge an *Properties*, die mehrfach vorkommen. Der dabei generierte Silo wird Konfektionssilo genannt. Er bildet jedoch nicht den Gedanken aus dem optimierten Prozess ab, dass die *Properties* und *Psets* im Konfektionssilo lediglich zum jeweiligen *Property*- und *Pset*-Silo referenzieren.

Abbildung 5.21 zeigt die in *BIMQ* hochgeladene Excel-Datei des Konfektionssilos. Dabei ist das *Pset_ActuatorTypeCommon* und ein Ausschnitt dessen *Properties* und *EnumeratedValue* hervorgehoben.

Konfektionssilo	Code	Typ	Einheiten	Allplan	IFC 4.3
Pset_ActuatorCommon	KS-002	Gruppe		Pset_ActuatorCommon	Pset_ActuatorCommon
Pset_ActuatorPHistory	KS-003	Gruppe		Pset_ActuatorPHistory	Pset_ActuatorPHistory
Pset_ActuatorTypeCommon	KS-004	Gruppe		Pset_ActuatorTypeCommon	Pset_ActuatorTypeCommon
ActuatorApplication	KS-004-01	Eigenschaft	Kennzeichen	ActuatorApplication	#.ActuatorApplication
DAMPERACTUATOR	KS-004-01-01	Wert			
ENTRYEXITDEVICE	KS-004-01-02	Wert			
FIRESMOKEDAMPERACTUATC	KS-004-01-03	Wert			
LAMPACTUATOR	KS-004-01-04	Wert			
SUNBLINDACTUATOR	KS-004-01-05	Wert			
VALVEPOSITIONER	KS-004-01-06	Wert			
OTHER	KS-004-01-07	Wert			
NOTKNOWN	KS-004-01-08	Wert			
UNSET	KS-004-01-09	Wert			
ActuatorStatus	KS-004-02	Eigenschaft	Kennzeichen	ActuatorStatus	#.ActuatorStatus
FailPosition	KS-004-03	Eigenschaft	Kennzeichen	FailPosition	#.FailPosition
ManualOverride	KS-004-04	Eigenschaft	Wahr/Falsch	ManualOverride	#.ManualOverride
Reference	KS-004-05	Eigenschaft	Identifizierungszeichen	Reference	#.Reference

Abb. 5.21: Konfektionssilo in *BIMQ* im Menü *Vorlagen*

5.2.5 Entities erstellen

Die Einträge der *Entities* sind in *BIMQ* im Menü *Vorlagen* mit *Typ Element* anzulegen. Die Vorgehensweise ist grundsätzlich wie bei den *Properties* und *Psets*. Bei den *Mappings* sind lediglich vorgegebene Werte möglich, da beim IFC-*Mapping* per Definition nur die in der IFC-Struktur vorhandenen *Entities* zuordenbar sind. In *BIMQ* sind diese möglichen *Mappings* bereits hinterlegt und können direkt ausgewählt werden. Als individuelle Erweiterung der vorgegebenen *Entities* ist die Angabe eines *PredefinedType* möglich. Durch die Verwendung von *Allplan* entstehen hier jedoch Einschränkungen. Die *Allplan*-*Mappings* von *Entities* können nämlich nur aus einer sogenannten *Referenzrichtlinie* ausgewählt werden. Die Werte aus der *Referenzrichtlinie* bilden die *Objekte* aus der *CDE Bimplus* ab. In Tabelle 5.3 sind für eine Auswahl an *Entities* die zugehörigen *Mappings* angegeben. Außerdem sind die zugehörigen *Objekte* in *Allplan* ergänzt. Die Bezeichnungen der *Objekte* in *Bimplus* und *Allplan* weichen stellenweise voneinander ab, womit

die Nachvollziehbarkeit nicht vollständig gegeben ist. Zum Beispiel wird die *Entity IfcBeam* in *Bimplus* als *Balken* bezeichnet, jedoch hat das zugehörige *Objekt* in *Allplan* den Namen *Unterzug*, *Überzug*.

Tab. 5.3: Zusammenhang von *Entity*, *Bimplus-Objekt* und *Allplan-Objekt*

<i>Entity</i>	<i>Bimplus-Objekt</i> und <i>Mapping</i> in <i>BIMQ</i>	<i>Allplan-Objekt</i>
<i>IfcBeam</i>	<i>Balken</i>	<i>Unterzug</i> , <i>Überzug</i>
<i>IfcColumn</i>	<i>Stütze</i>	<i>Stütze</i>
<i>IfcCovering</i>	<i>Bodenbelag</i> , etc.	<i>Bodenfläche</i> bzw. <i>Bodenbelag</i> , etc.
<i>IfcDoor</i>	<i>Tür</i>	<i>Tür</i> , <i>Tor-SmartPart</i>
<i>IfcFooting</i>	<i>Fundament</i>	<i>Plattenfundament</i>
<i>IfcRoof</i>	<i>Dach</i>	<i>Dachhaut</i>
<i>IfcSlab</i>	<i>Deckenplatte</i>	<i>Decke</i>
<i>IfcSpace</i>	<i>Raum</i>	<i>Raum</i>
<i>IfcStair</i>	<i>Treppe</i>	<i>Gerade Treppe</i> , <i>Wendeltreppe</i> etc.
<i>IfcWall</i>	<i>Wand</i>	<i>Wand</i>
<i>IfcWindow</i>	<i>Fenster</i>	<i>Fenster-SmartPart</i>

In Abbildung 5.22 ist beispielhaft die Definition der *Entity IfcBeam*¹¹ im Menü *Vorlagen* dargestellt.

Elementgeometrien	Code	Typ	Einheiten	Allplan	IFC 4.3
 Standard Träger	EL-001-05-01	Element		Balken	IfcBeam

Abb. 5.22: Definition der *Entity IfcBeam* in *BIMQ* im Menü *Vorlagen*

5.2.6 Mapping hinzufügen

Der Prozessschritt *Mapping hinzufügen* wurde in der Fallstudie vorgezogen und in den Schritten *Propertyset erstellen* (Abschnitt 5.2.2) und *Psetset erstellen* (Abschnitt 5.2.3) integriert. Die Erläuterung hierzu sind in den genannten Abschnitten angeführt.

5.2.7 Konfektionssilo mit *Entities* verknüpfen

Die Verknüpfung der *Properties* und *Psets* aus dem Konfektionssilo mit den zugehörigen *Entities* wird in *BIMQ* im Menü *Projektkonfiguration* erstellt. Eine Voraussetzung hierfür ist das Vorhandensein eines *Fachmodells*. Technisch gesehen ist in *BIMQ* ein *Fachmodell* ein Eintrag im Menü *Vorlagen*, dem der *Typ Modell* zugewiesen ist. Anschließend kann in das Menü *Projektkonfiguration* und dem Untermenü *Zuweisung Fachmodelle* gewechselt werden. Hier sind zwei getrennte Bereiche zu sehen, vgl. Abbildung 5.23. Links der Bereich *Vorlagen*, in dem alle Einträge aus dem Menü *Vorlagen* aufscheinen. Rechts befinden sich der Bereich *Projektanforderungen*. Um die Anforderungen zu erstellen, sind die einzelnen Einträge hierarchisch aus dem Bereich *Vorlage* zum Bereich *Projektanforderungen* zu verschieben. Das Verschieben erfolgt per *Drag&Drop*. Das heißt, dass die einzelnen Einträge im Bereich *Vorlagen* anzuklicken sind und unter Halten des Klicks zur gewünschten Gliederung im Bereich *Projektanforderungen* zu ziehen sind. Die Hierarchie dabei ist

¹¹Der Name des Eintrages ist hier *Standard Träger*, da dieser aus einem bestehenden Projekt übernommen ist. Für den Prozessablauf und ist lediglich die genaue Bezeichnung bei den *Mappings* relevant.

absteigend nach dem *Typ* des jeweiligen Eintrages gegliedert. Diese ist: *Fachmodell – Element – Pset – Property – Wert*. Zu Beginn muss die oberste Hierarchie-Ebene, das *Fachmodell*, festgelegt werden. Dafür ist im Bereich *Vorlagen* ein Eintrag mit *Typ Modell* mit der rechten Maustaste anzuklicken und *Als Anforderung hinzufügen* auszuwählen. Das Modell erscheint dann rechts als Hierarchie-Ebene. Danach können hintereinander die *Entities*, *Psets* sowie die *Properties* und deren zugehörigen *Enumerations* verknüpft werden. In Abbildung 5.24 ist ein Ausschnitt der verknüpften Einträge zu sehen.

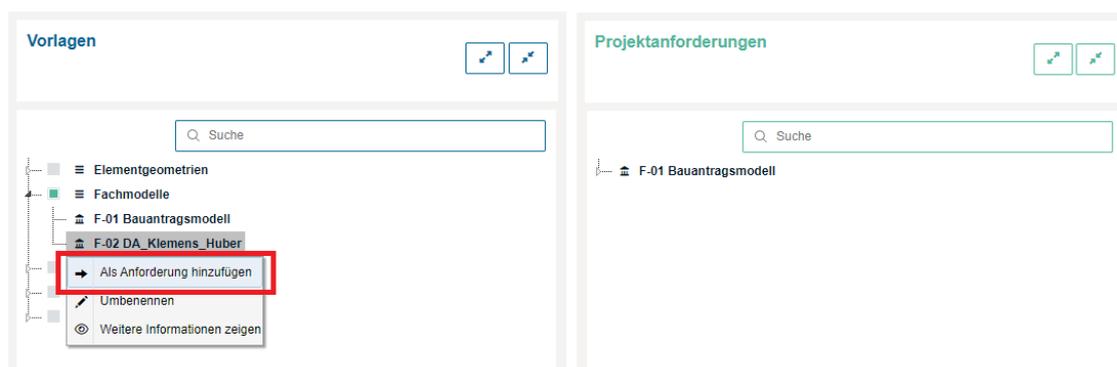


Abb. 5.23: *Fachmodell* mithilfe der Funktion *Als Anforderung hinzufügen* im Menü *Projektkonfiguration* anlegen

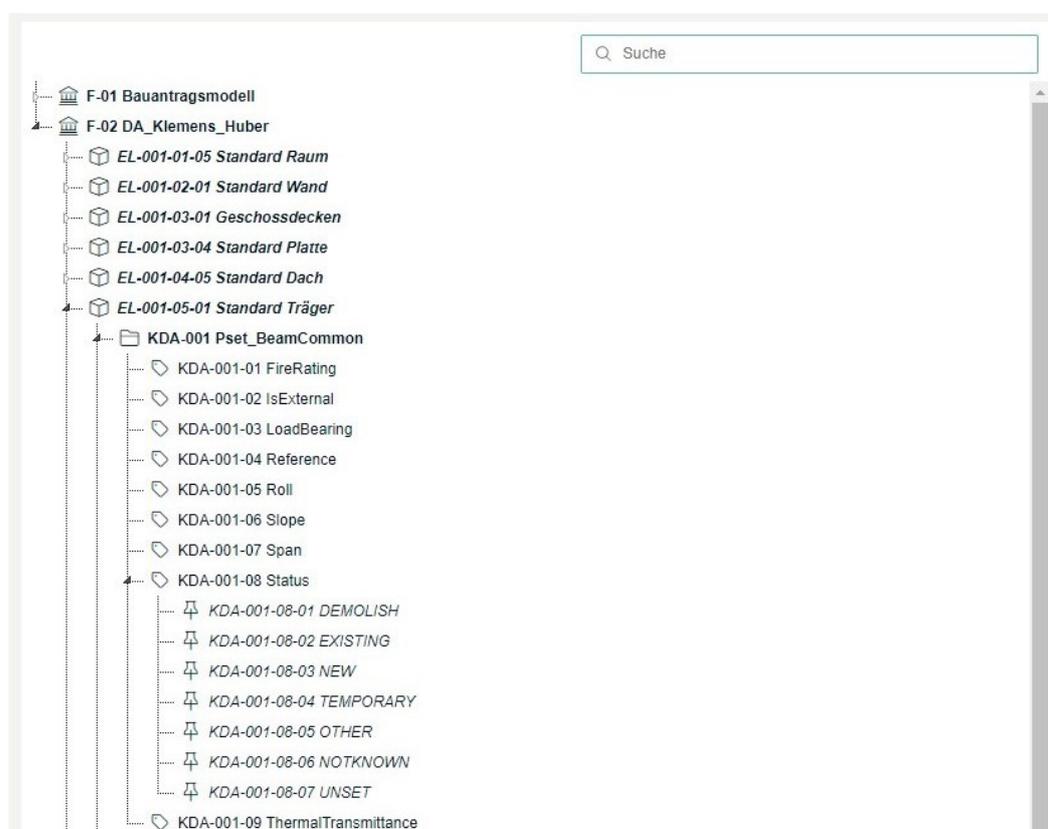


Abb. 5.24: *Projektanforderungen* im Menü *Projektkonfiguration*

5.2.8 Projektanforderung der Projektphase und Anwendungsfall zuweisen

Die zuvor definierte *Projektkonfigurationen* wird in *BIMQ* im Menü *Projektanforderungen* den entsprechenden *Projektphasen* und *Anwendungsfällen* zugewiesen. Das Anlegen der *Leistungsbilder*, *Projektphasen* und *Anwendungsfälle* erfolgt wie bei den *Psets* und *Properties* im Menü *Vorlagen*. In der Fallstudie wird eine beispielhafte *Projektphase* mit Code *LOI XXX* als Platzhalter angelegt. Im Prozessschritt ist zuerst das Untermenü *Zuweisung Leistungsbild* im Menü *Projektanforderungen* aufzurufen. Hier wird das *Fachmodell*, dessen Anforderungen zuvor definiert wurden, dem *Leistungsbild* zugewiesen. In Abbildung 5.25 ist dies das *Fachmodell DA_Klemens_Huber* und ein beispielhaftes *Gesamtleistungsbild*. Im Untermenü zum jeweiligen *Fachmodell* kann nun die Zuweisung der *Properties* zur entsprechenden *Projektphase* und *Anwendungsfall* erfolgen. Hierfür sind die zugehörigen Spalten einzublenden. In Abbildung 5.26 sind die *Projektanforderungen* und die Spalte *LOI XXX-XXX* ersichtlich. Die Abkürzung der Spalte wird gebildet aus dem Code der ausgewählten Projektphase *LOI XXX* und dem Code des Anwendungsfalls *XXX*. Die *Properties* werden nun zugewiesen, indem die Zelle in der Zeile des *Property* und der Spalte *LOI XXX-XXX* angeklickt wird. Dadauch verändert sich, zur Bestätigung, der dargestellte Bindestrich in einen Hacken. Dies ist für alle gewünschten *Properties* durchzuführen. Als Unterstützung stehen Funktionen wie *Mehrfachzuweisung* und *Excel Import/Export* zur Verfügung.

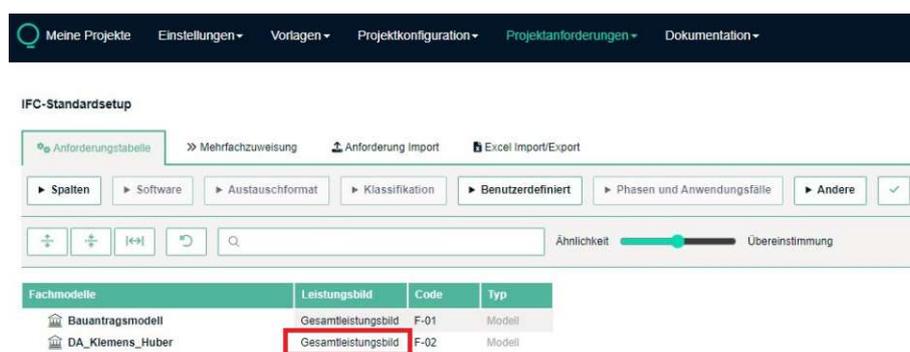


Abb. 5.25: Zuweisung des *Fachmodells* zum *Leistungsbild* in *BIMQ* im Untermenü *Zuweisung Leistungsbild*

5.2.9 Softwarevorlagen exportieren

In *BIMQ* können im Menü *Dokumentation* jeweils *Berichte*, *Softwarevorlagen* oder *Prüfregeln* der zuvor definierten *Projektanforderungen* exportiert werden. *Berichte* bilden die Funktion ab die Anforderungen im PDF-Format zu exportieren, um diese beispielsweise dem AIA-Dokument anzuhängen. Unter *Softwarevorlagen* kann die erstellte Konfiguration an *Properties* und *Psets* direkt oder indirekt in eine BIM-Autorensoftware exportiert werden. Das Feature *Prüfregeln* ermöglicht das Ausgeben der Anforderung in Formaten, die als Grundlage für Prüfregel verwendet werden können. Entsprechend der Ziel-Software können unterschiedliche Dateiformate ausgegeben werden.

Der Export der *Softwarevorlagen* und *Prüfregeln* erfolgt immer unter Angabe des Leistungsbildes, Fachmodells, Projektphase und des Anwendungsfalls. Demzufolge wird eine spezifisch festgelegte Anforderung exportiert. Für den Anwendungsfall der Standardkonfiguration wird entsprechend die *Softwarevorlage* für die Software *Allplan* exportiert. Im Untermenü *Softwarevorlagen* scheint

DA_Klemens_Huber	Code	Typ	Einheiten	Allplan	IFC 4.3	LOI XXX-XXX
Standard Raum	EL-001-01-05	Element		Raum	IfcSpace.PredefinedType	
Standard Wand	EL-001-02-01	Element		Wand	IfcWall.PredefinedType	
Geschossdecken	EL-001-03-01	Element		Deckenplatte	IfcSlab.PredefinedType	
Standard Platte	EL-001-03-04	Element		Platte	IfcPlate.PredefinedType	
Standard Dach	EL-001-04-05	Element		Dach	IfcRoof.PredefinedType	
Standard Träger	EL-001-05-01	Element		Balken	IfcBeam.PredefinedType	
Pset_BeamCommon	KDA-001	Gruppe		Pset_BeamCommon	Pset_BeamCommon	
FireRating	KDA-001-01	Eigenschaft	Kennzeichen	FireRating	#FireRating	✓
IsExternal	KDA-001-02	Eigenschaft	Wahr/Falsch	IsExternal	#IsExternal	✓
LoadBearing	KDA-001-03	Eigenschaft	Wahr/Falsch	LoadBearing	#LoadBearing	✓
Reference	KDA-001-04	Eigenschaft	Identifizierungszeichen	Reference	#Reference	✓
Roll	KDA-001-05	Eigenschaft	ebener Winkel	Roll	#Roll	✓
Slope	KDA-001-06	Eigenschaft	ebener Winkel	Slope	#Slope	✓
Span	KDA-001-07	Eigenschaft	Länge (positiv, >0)	Span	#Span	✓
Status	KDA-001-08	Eigenschaft	Kennzeichen	Status	#Status	✓
DEMOLISH	KDA-001-08-01	Wert				
EXISTING	KDA-001-08-02	Wert				
NEW	KDA-001-08-03	Wert				
TEMPORARY	KDA-001-08-04	Wert				
OTHER	KDA-001-08-05	Wert				
NOTKNOWN	KDA-001-08-06	Wert				
UNSET	KDA-001-08-07	Wert				
ThermalTransmittance	KDA-001-09	Eigenschaft	Wärmedurchgangskoeffi	ThermalTransmittance	#ThermalTransmittance	✓
Pset_BeamSpecific	KS-642	Gruppe		Pset_BeamSpecific	Pset_BeamSpecific	
ElementMainMateriality	KS-642-01	Eigenschaft	Kennzeichen	ElementMainMateriality	#ElementMainMateriality	✓
Beton	KS-642-01-01	Wert				
Gipskarton	KS-642-01-02	Wert				
Holz	KS-642-01-03	Wert				
Stahl	KS-642-01-04	Wert				
Ziegel	KS-642-01-05	Wert				
Glas	KS-642-01-06	Wert				
andere	KS-642-01-07	Wert				
ND	KS-642-01-08	Wert				
WienBV_BeamSpecific	KS-644	Gruppe		WienBV_BeamSpecific	WienBV_BeamSpecific	

Abb. 5.26: Zuweisung der *Properties* zur *Projektphase* und dem *Anwendungsfall* in *BIMQ* im Menü *Projektanforderungen*

bei der Auswahl von *Allplan* ein zusätzliches Feld namens *Typ* auf, das die Auswahlmöglichkeiten *Bimplus Export* und *IFC Zuordnungsdatei* anbietet, siehe Abbildung 5.27. Bei *Bimplus Export* wird die Softwarevorlage zu der *CDE Bimplus* exportiert. Unter *IFC Zuordnungsdatei* wird die Zuordnung zwischen den *Allplan*-Attributen und den gewünschten *Properties* der definierten Anforderung exportiert. Die beiden Abläufe werden nachfolgend in Abschnitt 5.3 beschrieben. Für den Export der Softwarevorlage der Standardkonfiguration ist im ersten Schritt *Bimplus Export* auszuwählen. Abbildung 5.27 zeigt die dabei getroffenen Einstellungen für das *Leistungsbild*, *Fachmodell* etc.

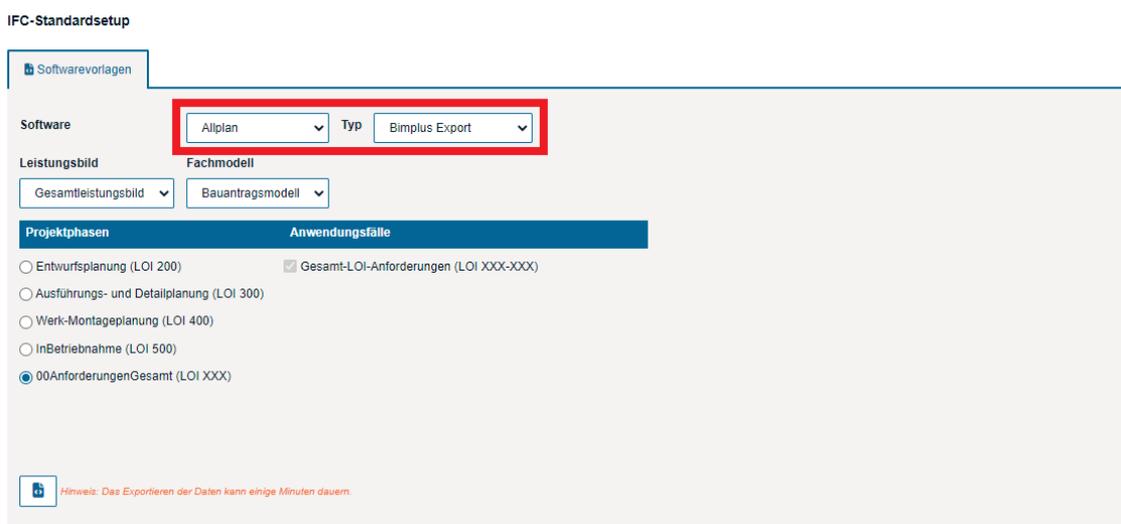


Abb. 5.27: Exportieren der Softwarevorlage in BIMQ

5.3 Prozessschritte in Allplan

Spezifisch bei *Allplan* wird die Softwarevorlage nicht direkt von *BIMQ* in die Modellierungssoftware *Allplan* exportiert, sondern zuerst zu der *CDE Bimplus*. Nachdem im Menü *Dokumentation* die gewünschte Auswahl des Leistungsbildes etc. getroffen wurde, kann auf den Button *Export* geklickt werden, siehe Abbildung 5.27 links unten. Dadurch erscheint ein weiteres Eingabefenster. Dieses ist in Abbildung 5.28 dargestellt. Hier sind einerseits ein Name für die Softwarevorlage und andererseits die Benutzerdaten (E-Mail und Passwort) und der *Projektteamname* aus *Bimplus* anzugeben. Dementsprechend ist ein Benutzeraccount in *Bimplus* erforderlich.

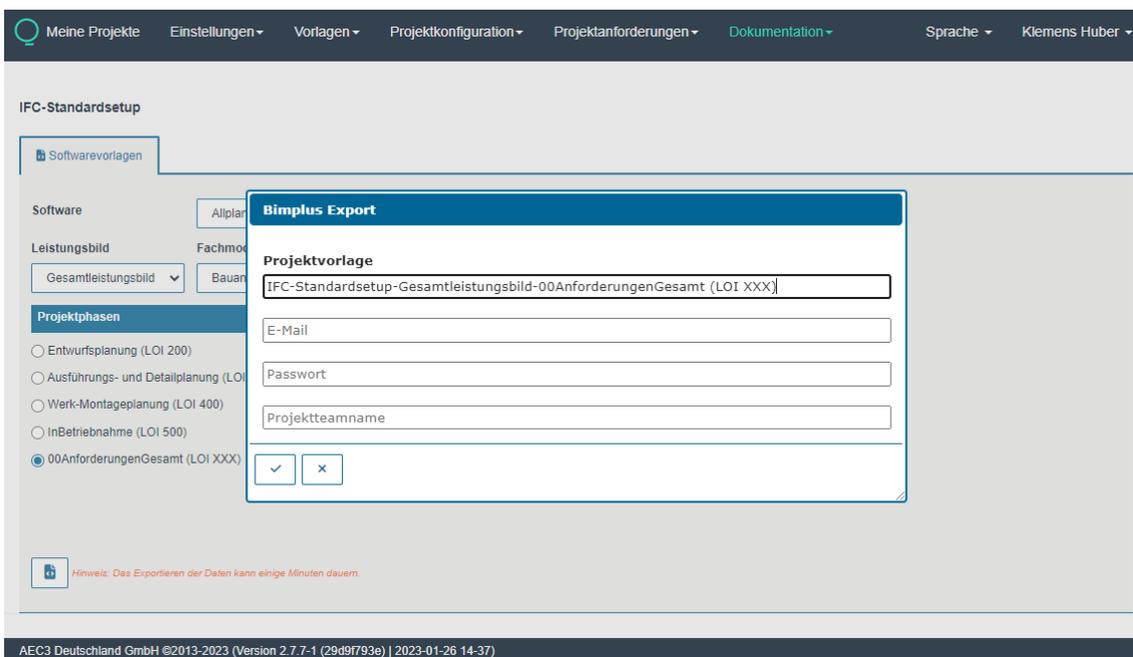


Abb. 5.28: Fenster *Bimplus Export* in BIMQ

5.3.1 Softwarevorlage dem Projekt zuweisen

Im nächsten Prozessschritt ist in *Bimplus* die exportierte Softwarevorlage dem gewünschten Projekt zuzuweisen. Dies setzt voraus, dass das Projekt in *Bimplus* bereits angelegt ist. Für die Zuweisung sind die Projekteinstellungen aufzurufen. Hier kann im Dropdown-Menü unter *Attributsetvorlage* und *Projektvorlage* der Name der exportierten Vorlage ausgewählt werden. In Abbildung 5.29 ist diese Option markiert. Die Einstellungen sind durch Klicken auf den Button

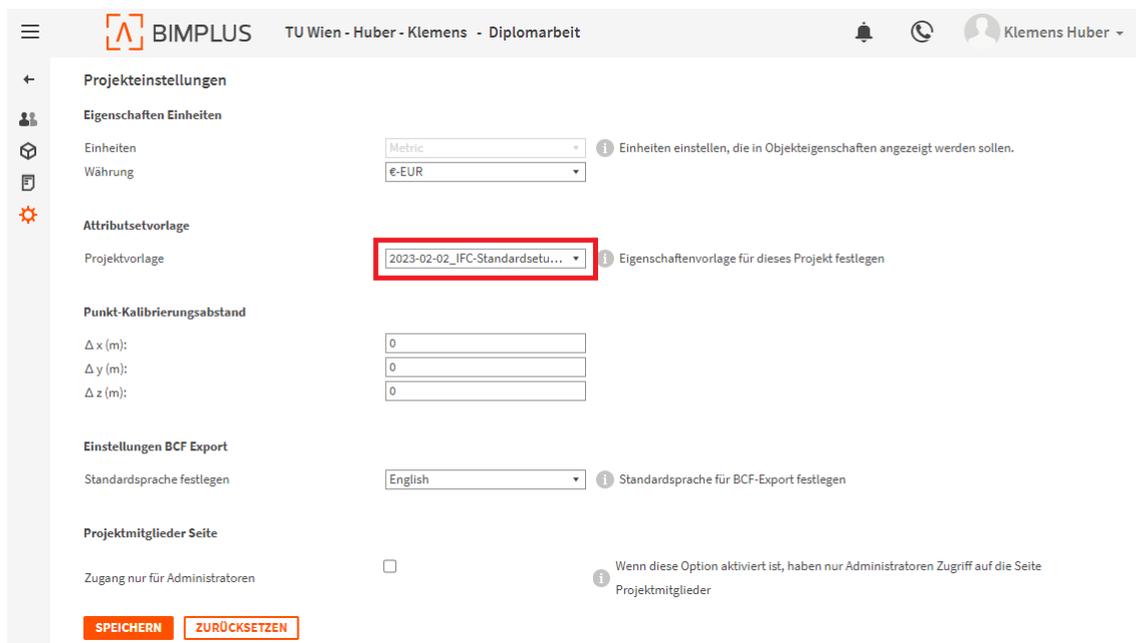


Abb. 5.29: Projekteinstellungen in *Bimplus*

Speichern zu sichern. Die erforderlichen Schritte in *Bimplus* sind hiermit abgeschlossen.

In *Bimplus* gibt es zusätzlich die Möglichkeit den Inhalt der importierten Vorlage anzusehen. Dafür steht die Funktion *Attributmanager* zur Verfügung, siehe Abbildung 5.30. Im Dropdown-Menü links oben kann die Vorlage ausgewählt werden. Im Bereich *Objekt* sind Objekte aufgelistet, die mit den zugehörigen *Entities* hinterlegt sind. Nach Anklicken des gewünschten Objektes werden im mittleren Bereich die durch die Vorlage verknüpften *Psets* und deren *Properties* aufgelistet. In Abbildung 5.30 ist das Objekt *Stütze* und das zugehörige *Pset_ColumnCommon* dargestellt.

5.3.2 Modellierung

Im nächsten Schritt kann in die BIM-Autorensoftware *Allplan* gewechselt werden. Um die Vorlage aus *Bimplus* mit dem entsprechenden *Allplan*-Projekt zu synchronisieren, muss das *Allplan*-Projekt mit dem Projekt in *Bimplus* verknüpft sein. Es ist hier zu unterscheiden, ob das entsprechende *Allplan*-Projekt bereits vorhanden ist oder neu erstellt wird. Bei der Erstellung eines neuen Projektes kann direkt in den zu treffenden *Projekteinstellungen* für die Option *Attributsetvorlage* der Wert *Bimplus* ausgewählt werden. Dadurch öffnet sich automatisch ein Fenster zum *Login* in *Bimplus*. Im darauffolgenden Fenster *Projekt mit Bimplus verknüpfen* kann das zuvor gewählte *Bimplus*-Team und *Bimplus*-Projekt ausgewählt werden. Zusätzlich kann ein bereits bestehendes *Bimplus*-Modell verwendet oder ein neues erstellt werden, siehe Abbildung 5.31.

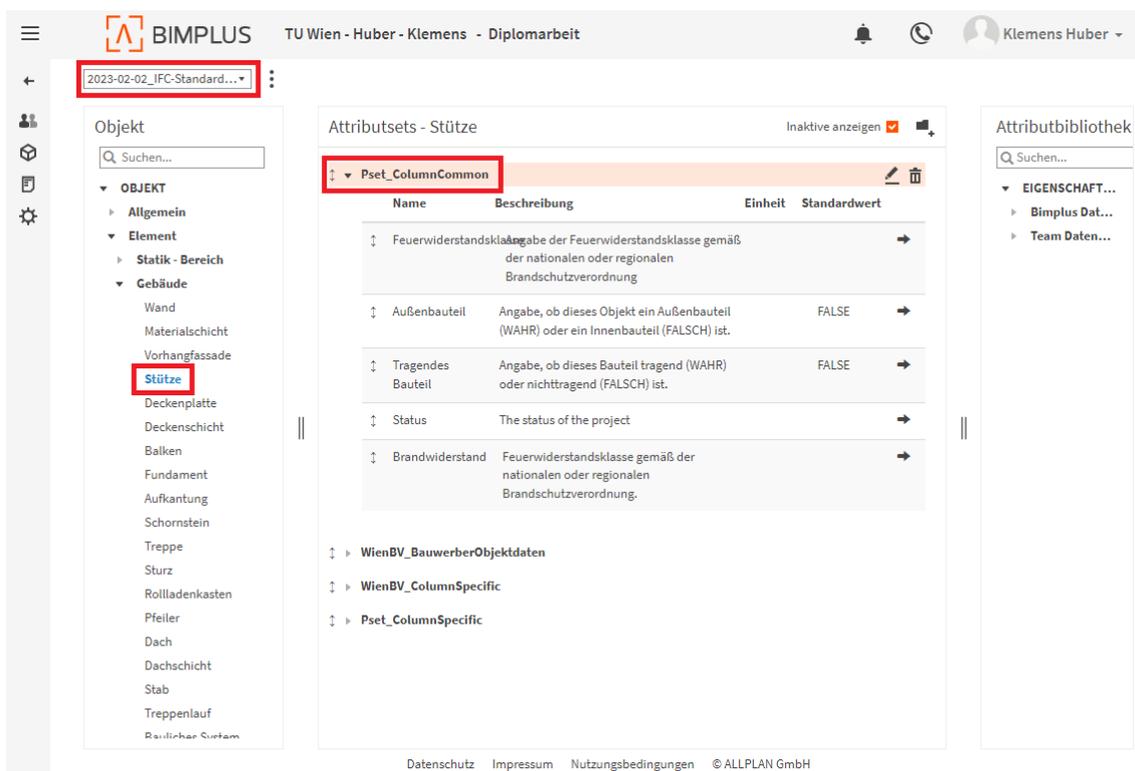


Abb. 5.30: Attributmanager in *Bimplus*

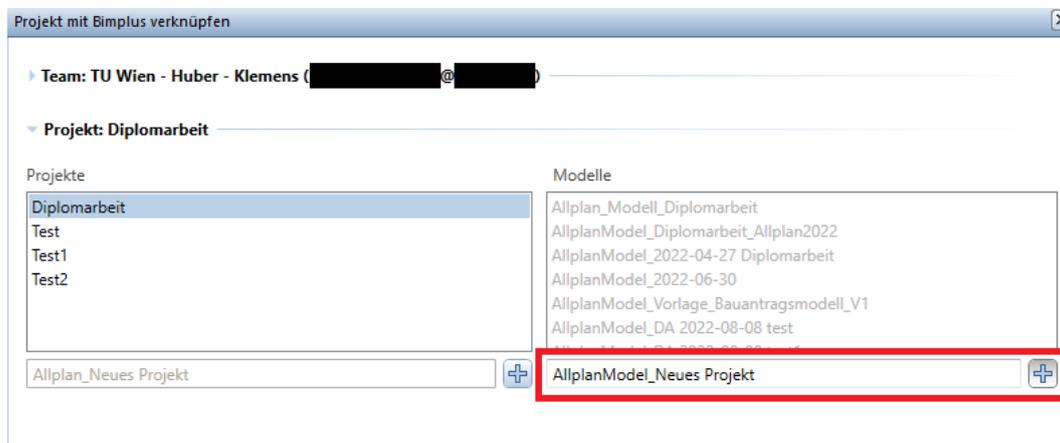
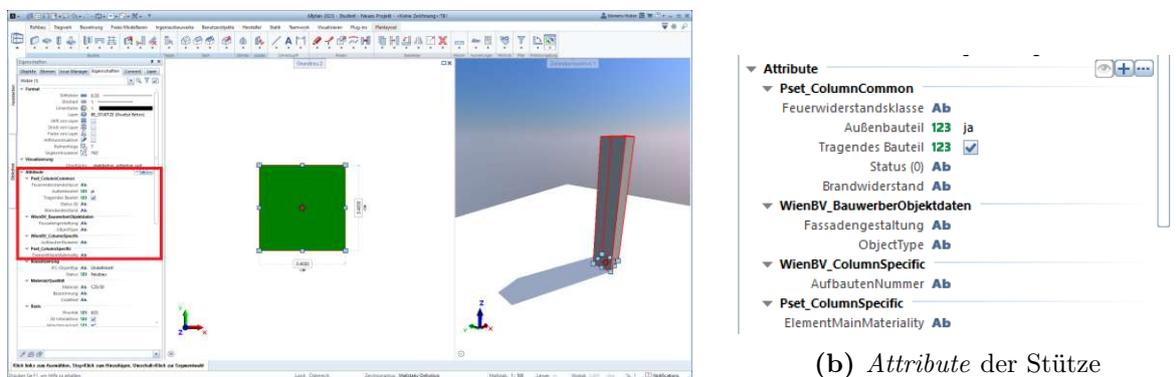


Abb. 5.31: Fenster *Projekt mit Bimplus verknüpfen* in *Allplan*

Nach dem Abschließen dieser Einstellungen wird die Vorlage im Hintergrund heruntergeladen. Die Vorlage wird in einem Unterordner des Projektordners namens /Xml/AttrSet (hier: C:\Data\Allplan\2023\Prj\2023-02-08_DA-Test.prj\Xml\AttrSet) abgespeichert. Je *Entity* ist hier eine xml-Datei angelegt. In einer zusätzlichen Datei namens *bim+AttrSetDownload.txt* wird über den Erfolg des Download berichtet und allenfalls Warnungen aufgelistet. Eine Warnung lautet hier: **WARNING:AttributeTitle: 'Status' isnotunique.**

Nun sind, beispielsweise beim Modellieren einer Stütze, die in der Vorlage definierten *Psets* und *Properties* ersichtlich. In Abbildung 5.32 ist in den *Eigenschaften* einer Stütze (Abbildung 5.32a und

vergrößert in 5.32b) das *Pset* *Pset_ColumnCommon* mit dessen *Properties* *Feuerwiderstandsklasse*, *Außenbauteil* etc. erkennbar.



(a) Modellierte Stütze in *Allplan* und deren *Eigenschaften*

(b) *Attribute* der Stütze

Abb. 5.32: *Psets* und *Properties* der Vorlage bei einer Stütze in *Allplan*

Bei einem bereits in *Allplan* vorhandenen Projekt kann die Verknüpfung mit *Bimplus* und der Import der Vorlage in gleicher Weise über die *Projekteinstellungen* induziert werden. Diese werden über die Funktion *Projekt neu, öffnen*, einem Rechtsklick auf dem Projektnamen und der Auswahl von *Eigenschaften* aufgerufen. Hier ist beim Feld *Attributsetvorlage* ebenfalls *Bimplus* auszuwählen. Die zu treffenden Einstellungen sind wie, in den vorangehenden Abschnitten, beschrieben anzugeben.

Die Vorlage ist nun in *Allplan* integriert. Bei der Modellierung der Bauteile bzw. der *Allplan-Objekte*, wie z. B. *Wand*, *Decke*, *Stütze* etc. können nun die Werte der durch die Vorlage definierten *Attribute* (*Properties*) angegeben werden.

5.3.3 IFC-Datei exportieren

Für den Export der modellierten Inhalte als IFC-Datei ist die Funktion *IFC Daten exportieren* vorgesehen. Im sich öffnenden Fenster können unter *Teilbilder* die zu exportierenden Inhalte angegeben werden. Außerdem sind ein Dateipfad und -name anzugeben. Über den Button *Einstellungen* werden zusätzliche Optionen verfügbar. Abbildung 5.33 stellt die beim Exportieren getroffenen Einstellungen dar. Unter *Attribute Mapping* ist es essenziell, die aus *BIMQ* exportierte *Zuordnungsdatei* zu laden und anzuwählen. Die *Zuordnungsdatei* wird in *BIMQ* ebenfalls im Menü *Dokumentation* ausgegeben, lokal am PC gespeichert und hat die Dateierweiterung *.cfg*. Der Code 5.3 zeigt einen beispielhaften Ausschnitt dieser Datei. Es ist für die *Entity IfcColumn* ersichtlich welche *GUID* der *Allplan-Attribute* welchen *Properties* zuzuordnen sind.

Code 5.3: Ausschnitt aus einer Zuordnungsdatei

```

1 @Column
2 ATT_GUID: bcb38045 -8c4b -4dee -ae4c -7090facbbf8a ->Pset_ColumnCommon:FireRating
   (IfcLabel)
3 ATT_GUID: 51544853 -5092 -4d18 -8076 -97f419457581 ->Pset_ColumnCommon:IsExternal
   (IfcBoolean)
4 ATT_GUID: aa7a0ffe -9dc8 -4a04 -bd4e -5c305767ebff ->
   Pset_ColumnCommon:LoadBearing (IfcBoolean)
5 ATT_GUID: f7db1ccb -deef -4f0b -8731 -eb909de8788f ->Pset_ColumnCommon:Reference (
   IfcIdentifier)

```

```

6 ATT_GUID:b7be7b7f-047a-4bd3-b299-80ddb1ce0f1f->Pset_ColumnCommon:Roll(
  IfcPlaneAngle)
7 ATT_GUID:f8d365c4-a940-46c3-b845-6e6b5b197ac6->Pset_ColumnCommon:Slope(
  IfcPlaneAngle)
8 ATT_GUID:df78f97d-c7e6-405a-99f7-18e1d61236b6->Pset_ColumnCommon:Status(
  IfcLabel)
9 ATT_GUID:4cb05ba2-fbfb-4b9d-9470-29a22b51fce8->
  Pset_ColumnCommon:ThermalTransmittance(IfcThermalTransmittance)

```

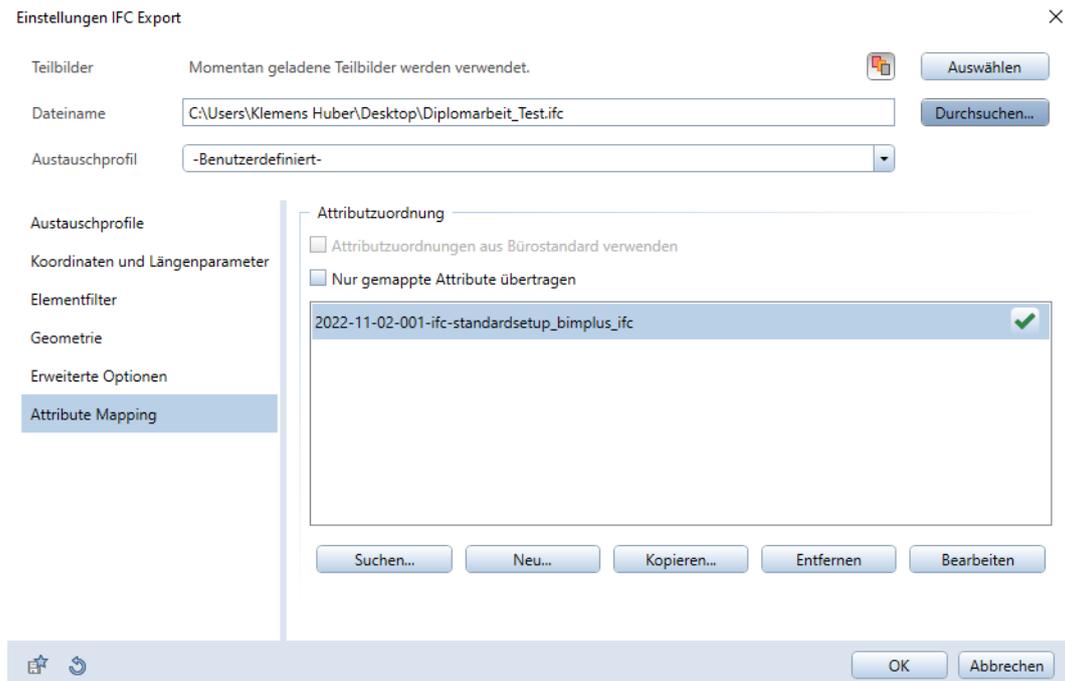


Abb. 5.33: Fenster *IFC Daten exportieren* in *Allplan*

Nach dem Exportieren wird ein Fenster *Protokolldatei* eingeblendet, das über die Anzahl der exportierten *Allplan-Objekte* und *IFC-Objekte* informiert.

5.4 Formale Prüfung

Am Ende des Prozessablaufes kann überprüft werden, ob die entsprechenden *Psets* und *Properties* der Vorlage auch in der IFC-Datei korrekt vorhanden sind. Dies stellt eine formale Prüfung dar. *BIMQ* bietet die Möglichkeit für den Export von sogenannten *Prüfregeln* an. Dabei werden Software-spezifische Dateien aus *BIMQ* exportiert, die dann in die Prüfungssoftware importiert werden können. Eine dafür unterstützte Software ist *Solibri*.

Für die Prüfung, mit den aus *BIMQ* exportierten *Prüfregeln*, ist in *Solibri* ein neuer *Regelsatz* anzulegen. Mit diesem *Regelsatz* wird dann die Funktion *Modell überprüfen* ausgeführt. Abbildung 5.34 zeigt, dass bei der beispielhaft durchgeführten Prüfung der IFC-Datei *Fehlende Eigenschaften (Properties)* und *Fehlende Eigenschaftensets (Psets)* erkannt wurden.

Im Prozessablauf ist an dieser Stelle eine Feedbackschleife integriert. Zuerst sollte eine Überprüfung der modellierten Objekte oder der Export-Einstellungen in *Allplan* erfolgen. Falls die Prüfung wiederholt negativ ausfällt, ist die Ursache, für die in der Prüfung aufgezeigten Abweichungen, in *BIMQ* zu suchen und im jeweiligen Prozessschritt zu korrigieren.

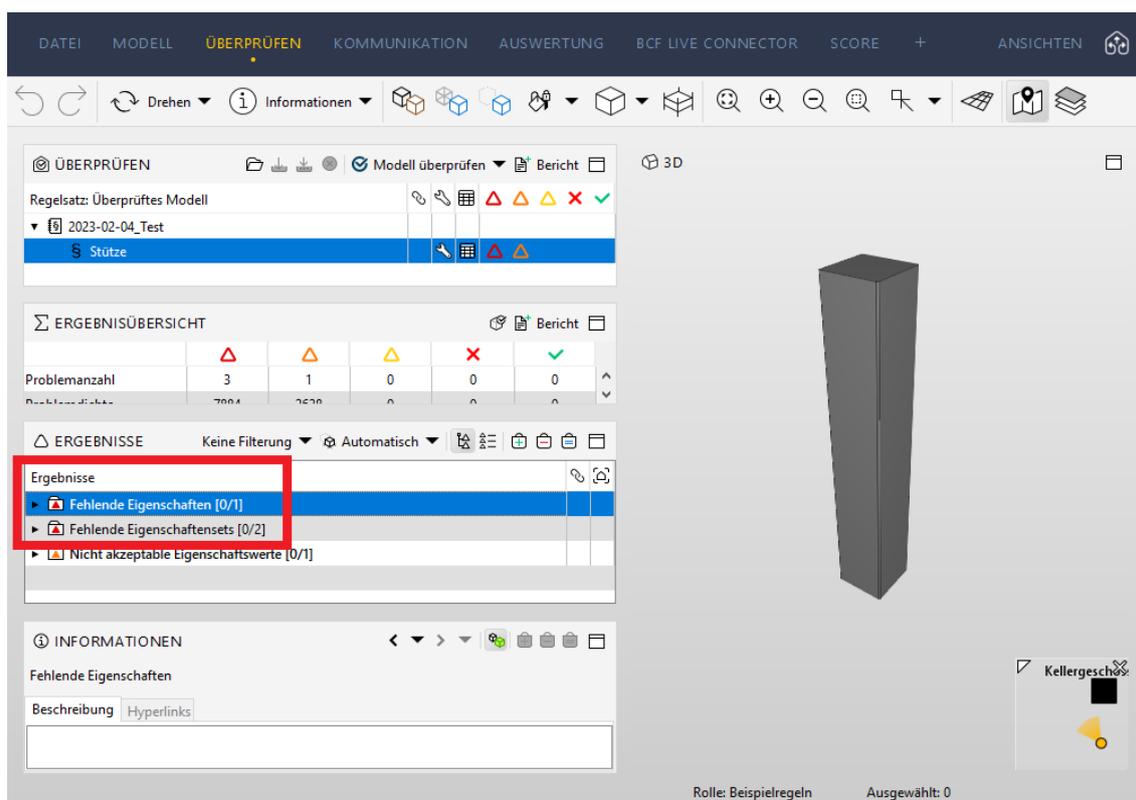


Abb. 5.34: Überprüfung der IFC-Datei in Solibri

5.5 Durchführung von Versuchen

Im Folgenden wird die Umsetzung des dargestellten optimierten Prozessablaufs mit *BIMQ* und *Allplan* qualitativ und quantitativ getestet. In Abschnitt 5.5.1 wird ein qualitativer Test durchgeführt und geprüft, ob alle Prozessschritte wie gewünscht funktionieren. Im Abschnitt 5.5.2 wird das Leistungsvermögen im Prozessablauf von *Allplan* ausgetestet und zusätzlich quantitative Tests durchgeführt. Beispielsweise wird in einem definierten Beispielmmodell ausgetestet wie viele *Properties* und deren Definitionen korrekt in der IFC-Datei vorhanden sind. Die Tests wurden im Februar 2023 durchgeführt. Die dabei verwendeten, aktuellsten Softwareversionen sind:

- *BIMQ* Version 2.7.7-4
- *Allplan* Version 2023-0-4

5.5.1 Allgemeiner Test der Funktionstüchtigkeit der Prozessschritte

Die allgemeinen Tests haben die Funktionstüchtigkeit der Prozessschritte im Fokus.

Messziele und Messgrößen

Das Ziel des ersten Versuchs ist es qualitative Aussagen zu der Übereinstimmung der verarbeiteten Daten in den Prozessschritten in *BIMQ* und *Allplan* zu erhalten. Qualitative Aussagen werden von einem Vergleich abgeleitet. Als quantitative Messgröße wird zusätzlich die Zeit der einzelnen Prozessschritte gemessen, um einen Überblick zu erhalten. Aussagen sind von dieser Messgröße nicht ableitbar, da nur ein Versuchsdurchgang durchgeführt wird, da der Fokus auf den qualitativen Aussagen liegt.

Versuchsdurchführung

Für den Versuch wird der Prozess mit mehreren ausgewählten *Psets* und deren *Properties* durchlaufen. Als *Entities* werden 5 im Hochbau häufig verwendete gewählt: *IfcFooting*, *IfcWall*, *IfcColumn*, *IfcSlab* und *IfcSpace*. Für die *Psets* werden die Standard-*Psets* aus der erstellten Standardkonfiguration und deren zugehörigen *Properties* gewählt.

Konfektionssilo mit *Entities* verknüpfen

Die *Projektkonfiguration* des Versuchs wird in *BIMQ* mithilfe der vorhandenen Standardkonfiguration erstellt. Dies bildet den Prozessschritt *Konfektionssilo mit Entities verknüpfen* (Abschnitt 5.2.7) ab. Dieser Schritt nahm für die 5 *Entities* und jeweils einem *Pset* 4 Minuten und 19 Sekunden in Anspruch, was einem Durchschnitt von 52 Sekunden pro *Entity* entspricht. Es wurden die *Psets* aus dem Konfektionssilo verwendet, bei dem die zugehörigen *Properties* bereits den *Psets* untergeordnet sind.

Im Menü *Projektanforderungen* müssen den *Eigenschaften (Properties)* ein *Anwendungsfall* zugewiesen werden¹². Dies dauerte 1 Minute und 29 Sekunden. Die Bedingungen sind jedoch vereinfacht, da sämtlichen im *Fachmodell* vorhandenen *Properties* derselbe *Anwendungsfall* zugewiesen werden konnte. Hier wird für die *Projektphase* und den *Anwendungsfall* der Platzhalter *LOI XXX-XXX* verwendet, siehe Abbildung 5.35.

DA_Klemens_Huber (Gesamtleistungsbild)	Code	Typ	Einheiten	Allplan	IFC 4.3
Standard Raum	EL-001-01-05	Element		Raum	IfcSpace.*
Pset_SpaceCommon	KDA-005	Gruppe		Pset_SpaceCommon	Pset_SpaceCommon
GrossPlannedArea	KDA-005-01	Eigenschaft	Fläche.m2	GrossPlannedArea	#.GrossPlannedArea
HandicapAccessible	KDA-005-02	Eigenschaft	Wahr/Falsch	HandicapAccessible	#.HandicapAccessible
IsExternal	KDA-005-03	Eigenschaft	Wahr/Falsch	IsExternal	#.IsExternal
NetPlannedArea	KDA-005-04	Eigenschaft	Fläche.m2	NetPlannedArea	#.NetPlannedArea
PubliclyAccessible	KDA-005-05	Eigenschaft	Wahr/Falsch	PubliclyAccessible	#.PubliclyAccessible
Reference	KDA-005-06	Eigenschaft	Identifizierungszeichen	Reference	#.Reference
Standard Wand	EL-001-02-01	Element		Wand	IfcWall.*
Pset_WallCommon	KDA-006	Gruppe		Pset_WallCommon	Pset_WallCommon
AcousticRating	KDA-006-01	Eigenschaft	Kennzeichen	AcousticRating	#.AcousticRating
Combustible	KDA-006-02	Eigenschaft	Wahr/Falsch	Combustible	#.Combustible
Compartmentation	KDA-006-03	Eigenschaft	Wahr/Falsch	Compartmentation	#.Compartmentation
ExtendToStructure	KDA-006-04	Eigenschaft	Wahr/Falsch	ExtendToStructure	#.ExtendToStructure
FireRating	KDA-006-05	Eigenschaft	Kennzeichen	FireRating	#.FireRating
IsExternal	KDA-006-06	Eigenschaft	Wahr/Falsch	IsExternal	#.IsExternal
LoadBearing	KDA-006-07	Eigenschaft	Wahr/Falsch	LoadBearing	#.LoadBearing
Reference	KDA-006-08	Eigenschaft	Identifizierungszeichen	Reference	#.Reference
Status	KDA-006-09	Eigenschaft	Kennzeichen	Status	#.Status
DEMOLISH	KDA-006-09-01	Wert			
EXISTING	KDA-006-09-02	Wert			
NEW	KDA-006-09-03	Wert			
TEMPORARY	KDA-006-09-04	Wert			
OTHER	KDA-006-09-05	Wert			
NOTKNOWN	KDA-006-09-06	Wert			
UNSET	KDA-006-09-07	Wert			

Abb. 5.35: *Projektanforderungen* beim allgemeinen Test in BIM

Softwarevorlage exportieren

Das Exportieren der Softwarevorlage von *BIMQ* nach *Bimplus* dauerte 1 Minute und 32 Sekunden. Im *Attributmanager* in *Bimplus* können die importierten *Psets* und *Properties* betrachtet werden.

¹²Bei diesem Versuch wird allen *Eigenschaften* derselbe *Anwendungsfall* zugewiesen. Dabei kann die Funktion *Suchen & Filtern* genutzt werden, um nach allen Einträgen des Typs *Eigenschaft* zu filtern. Im nächsten Schritt kann mit *Mehrfachzuweisung* allen markierten Einträgen gleichzeitig ein *Anwendungsfall* zugewiesen werden. Dem *Fachmodell* (hier *DA_Klemens_Huber*) muss ein *Leistungsbild* zugewiesen werden, hier *Gesamtleistungsbild*. Dies ist möglich, wenn beim Button *Projektanforderungen Alle Fachmodelle* ausgewählt wird. Möglicherweise müssen bei beiden Schritten die entsprechenden Spalten eingeblendet werden.

Abbildung 5.36 zeigt die *Attributsets (Psets)* für das *Objekt Wand*. Die Namen des *Pset* und der *Properties* stimmen mit *BIMQ* überein. Es fällt auf, dass beispielsweise beim *Property ThermalTransmittance* die Einheit mit m/mK angezeigt wird. In *BIMQ* wurde sie jedoch mit $W/(m^2 K)$ festgelegt. Diese Abweichung tritt ebenfalls bei den gleichnamigen *Properties* bei *IfcColumn* und *IfcSlab* auf. Beim Anklicken eines *Property* erscheint in der *Attributbibliothek* rechts der zugehörige Eintrag. Die Funktion *Bearbeiten* macht die Definitionen des *Property* ersichtlich. Abbildung 5.37 zeigt das entsprechende Fenster, in dem ebenfalls die falsch übernommene Einheit ersichtlich ist.

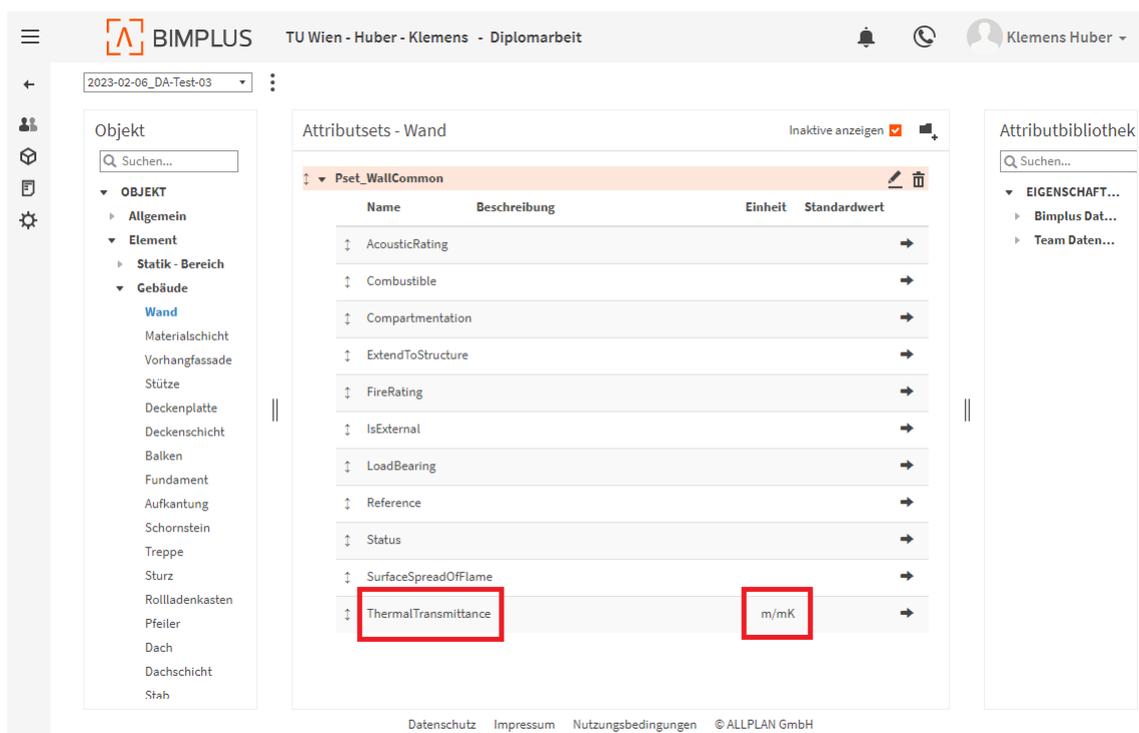


Abb. 5.36: *Attributsets* für das *Objekt Wand* in *Bimplus*

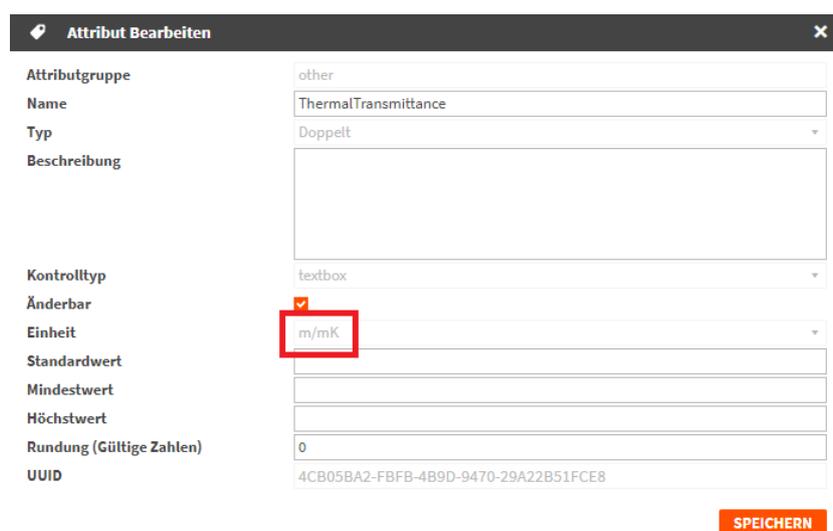


Abb. 5.37: Funktion *Attribut Bearbeiten* in *Bimplus*

Softwarevorlage dem Projekt zuweisen

Die Vorlage wird nun in den Einstellungen des *Bimplus*-Projektes als *Projektvorlage* festgelegt. Dieser Vorgang dauerte während des Versuch 51 Sekunden.

Modell erstellen

Bei einem bestehenden Projekt in *Allplan*, das bereits mit dem *Bimplus*-Projekt verknüpft ist, erscheint automatisch ein Hinweis, dass die *Attributsets* in *Bimplus* geändert wurde. Gleichzeitig wird die Frage gestellt, ob die *Attributsets* *aktualisiert* werden sollen. Das Laden der *Attributsets* in *Allplan* hat im Versuch 32 Sekunden gedauert. Nachdem die Vorlage verfügbar ist, wird für den Versuch ein Modell erstellt, das Objekte aller 5 *Entities*, die im Abschnitt *Versuchsdurchführung* ausgewählt wurden, enthält.

Im Abschnitt 5.2.7 wurde erwähnt, dass die *Allplan-Mappings* zu den *Entities* nicht eindeutig sind. Diese hängen nämlich mit der Bezeichnung der *Objekte* in *Bimplus* zusammen. In *Allplan* fällt auf, dass die Bezeichnung der *Objekte* stellenweise wiederum von denen in *Bimplus* abweicht. Beispielsweise hat die *Entity IfcBeam* in *BIMQ* das *Mapping Balken* (diese Bezeichnung hat ihren Ursprung in *Bimplus*) und wird in *Allplan* mit dem *Objekt Unterzug, Überzug* modelliert.

Die *Psets* und *Properties* sind bei modellierten *Objekten* in den *Eigenschaften* ersichtlich und werden als *Attribute* bezeichnet. Abbildung 5.38 zeigt die *Eigenschaften* und *Attribute* einer Wand. Im Vergleich mit *BIMQ* fällt auf, dass bei den Namen der *Properties* *Reference* und *Status* der Text (0) ergänzt ist. Beim *Attribut Reference* ist keine Eingabe eines Wertes möglich. Bei *Status* sind die *EnumerationsValues*, wie in *BIMQ* definiert, auswählbar. Der *Data Type* der *Properties* ist korrekt übernommen worden. Beim *Objekt Raum* fällt auf, dass anstatt dem *Pset*-Namen die Bezeichnung *Benutzer* als Gliederung der *Properties* angeführt wird.

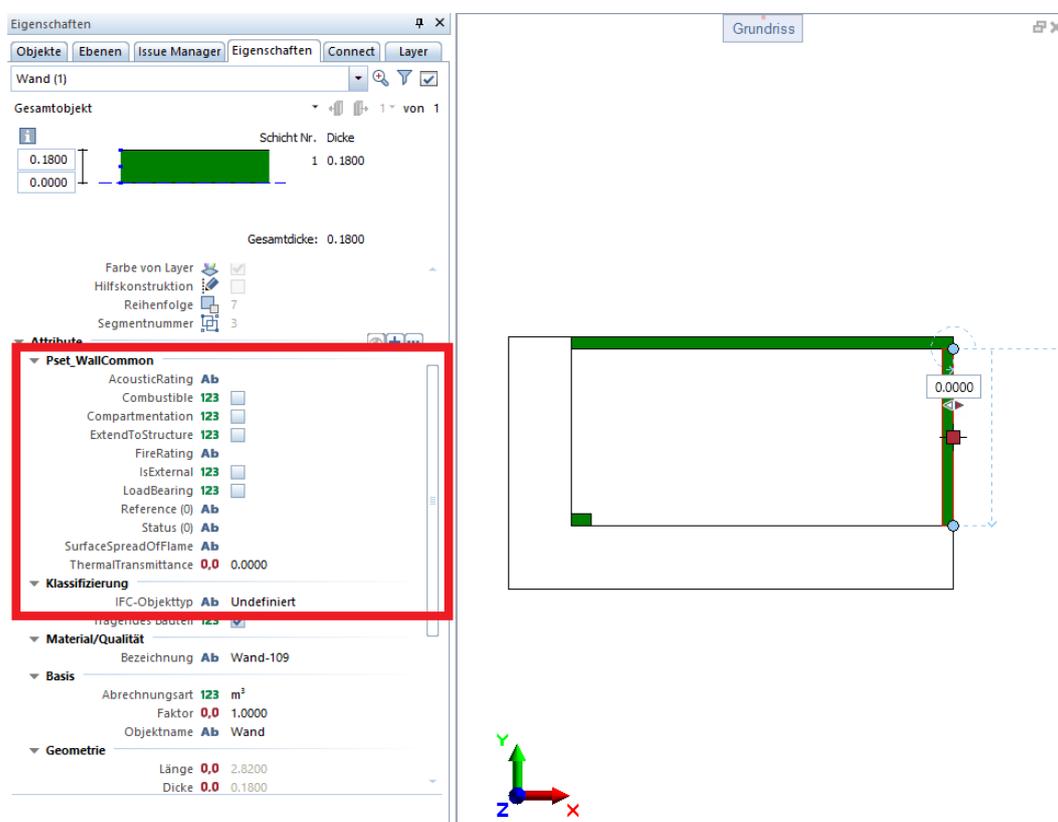


Abb. 5.38: Attribute einer Wand in Allplan

Die Attribute eines Bauteils können ebenfalls mit der Funktion *Attribute modifizieren* detailliert angezeigt werden. Bei der Nutzung dieser Funktion werden allfällige Einheiten der *Properties* sichtbar. Bei der modellierten *Stütze* ist erkenntlich, dass bei *ThermalTransmittance* ebenfalls die abweichende Einheit m/mK angezeigt wird, siehe Abbildung 5.39 rechts. Das Attribut *Reference (0)* ist hier mit grauer Schrift dargestellt und eine Eingabe eines Wertes ist nicht möglich.

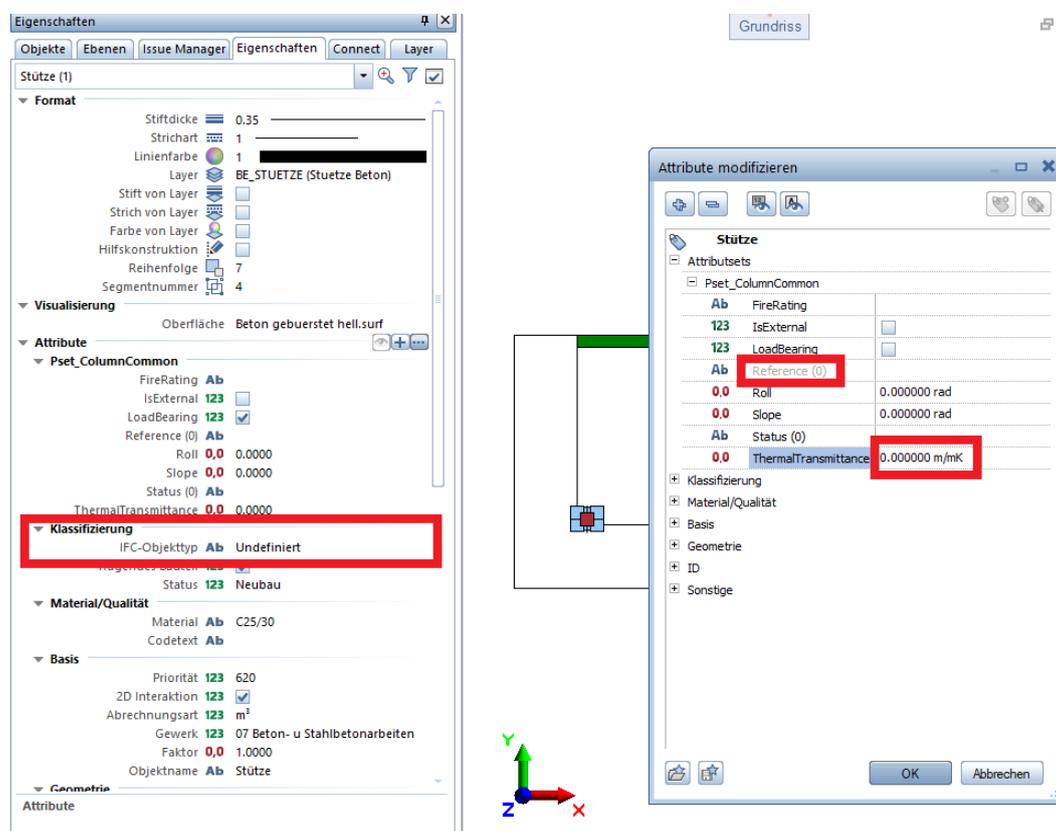


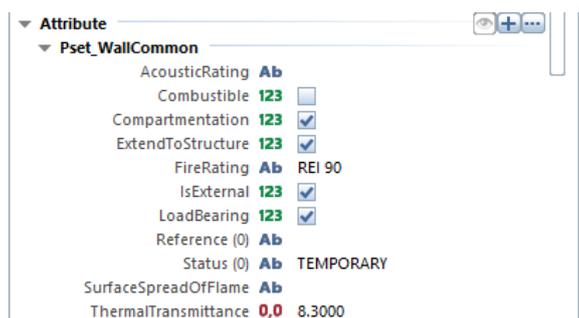
Abb. 5.39: Attribute einer *Stütze* in *Allplan*

Unter den in *Allplan* vordefinierte *Attributen* gibt es das *Attribut IFC-Objektyp*. Dieses wurde ebenfalls in den Abbildungen 5.38 und 5.39 (links) markiert. Bei diesem *Attribut* ist eine manuelle Angabe der *Entity* des jeweiligen *Objektes* möglich. Eine Angabe ist jedoch nicht zwingend erforderlich, da beim IFC-Export die *Entities* automatisch, gemäß dem *Objekt*, zugewiesen wird. Ein *Objekt Wand* bekommt beispielsweise die *Entity IfcWall* zugewiesen. In Abbildung 5.39 (links) ist ersichtlich, dass das *Attribut* den Wert *Undefiniert* aufweist. Lediglich bei der modellierten *Decke* ist der Wert *IfcSlab* angegeben.

Für einen ausführlicheren Vergleich mit der exportierten IFC-Datei werden nun einzelne *Property*-Werte willkürlich angegeben. In Abbildung 5.40 sind die angegebenen Werte bei einer *Wand* dargestellt. Der folgende Prozessschritt ist das Exportieren der IFC-Datei. Hierfür wird in *BIMQ* die *Zuordnungsdatei* exportiert und beim Export in *Allplan* angegeben, siehe Abschnitt 5.3.3.

IFC-Datei

Im Folgenden werden die Beobachtungen in der exportierten IFC-Datei festgehalten. Bezüglich dem modellierten *Fundament* fällt auf, dass in der IFC-Datei keine *Entity IfcFooting* vorhanden ist. Das *Fundament* (Bodenplatte) wurde als *IfcSlab* mit dem *PredefinedType Baseslab* ausgegeben. Die in der Vorlage definierten *Properties* des *Pset_FootingCommon* wurden scheinbar automatisch in ein *Pset* mit Namen *Pset_SlabCommon* übertragen.

Abb. 5.40: Festgelegte *Attribute*-Werte bei einer *Wand*

Bei den Wänden sind die *Properties*, bei denen ein Wert angegeben wurde, korrekt vorhanden. Bei *ThermalTransmittance* wurde die Einheit nicht exportiert, und auch der *Data Type* als *IfcReal* angegeben, siehe Code 5.4. Hier ist außerdem ersichtlich, dass *Allplan* gleichnamige *Properties* einer *Entity* doppelt ausgibt und mit dem gleichen *Pset* verknüpft. Beispielsweise das *Property LoadBearing* in Zeile #182 und #189. Einerseits sind dies die *Properties* aus der definierten Vorlage, andererseits werden gleichnamige, in *Allplan* vordefinierte *Attribute*, ausgegeben. Allgemein wurden *Properties* von Typ *IfcLabel*, bei denen kein Wert angegeben wurde, nicht exportiert. Dies betrifft beispielsweise die *Properties AcousticRating*, *Reference (0)* und *SurfaceSpreadOfFlame*, die in Abbildung 5.40 ersichtlich sind, jedoch nicht in der IFC-Datei (Code 5.4) aufscheinen.

Code 5.4: Ausschnitt der exportierten IFC-Datei

```

1  [...]
2  #182=IFCPROPERTY SINGLEVALUE('LoadBearing',,$,IFCBOOLEAN(.T.),$);
3  #183=IFCPROPERTY SINGLEVALUE('IsExternal',,$,IFCBOOLEAN(.T.),$);
4  #184=IFCPROPERTY SINGLEVALUE('ThermalTransmittance',,$,
5      IFC THERMAL TRANSMITTANCE MEASURE(0.),$);
6  #185=IFCPROPERTY SINGLEVALUE('Compartmentation',,$,IFCBOOLEAN(.T.),$);
7  #186=IFCPROPERTY SINGLEVALUE('ExtendToStructure',,$,IFCBOOLEAN(.T.),$);
8  #187=IFCPROPERTY SINGLEVALUE('IsExternal',,$,IFCBOOLEAN(.T.),$);
9  #188=IFCPROPERTY SINGLEVALUE('FireRating',,$,IFCLABEL('REI 90'),$);
10 #189=IFCPROPERTY SINGLEVALUE('LoadBearing',,$,IFCBOOLEAN(.T.),$);
11 #190=IFCPROPERTY SINGLEVALUE('Status',,$,IFCLABEL('TEMPORARY'),$);
12 #191=IFCPROPERTY SINGLEVALUE('ThermalTransmittance',,$,IFCREAL(8.3),$);
13 #192=IFCPROPERTY SINGLEVALUE('Combustible',,$,IFCBOOLEAN(.F.),$);
14 #193=IFCPROPERTY SINGLEVALUE('Compartmentation',,$,IFCBOOLEAN(.T.),$);
15 #194=IFCPROPERTY SINGLEVALUE('ExtendToStructure',,$,IFCBOOLEAN(.T.),$);
16 [...]
17 #211=IFCPROPERTY SET('1L1ARoWln0PeSFQbNtF9_s',#5,'Pset_WallCommon',,$,(
18     #182,#183,#184,#185,#186,#187,#188,#189,#190,#191,#192,#193,#194));
19 [...]

```

Das *Property FireRating* der Stütze ist in der IFC-Datei ebenfalls nicht vorhanden. Das *Property Reference* hat den Wert *AR_ST_C20/25*. Diesen leitet *Allplan* scheinbar von einem vordefinierten *Attribut* ab, das den gleichen Namen besitzt. Dies könnte der Grund sein, warum der Name des *Property Reference* mit (0) ergänzt und teilweise sogar ausgegraut wird.

Bei der Decke ist eine zusätzliche Abweichung erkennbar. Beim *Property AcousticRating* aus der Vorlage wurde in *Allplan* kein Wert angegeben. In der IFC-Datei scheint jedoch der Wert 1

auf. Dieser hat den Ursprung bei einem vordefinierten *Attribut* namens *Schallschutzklasse*. Das nicht definierte *Property Reference* ist bei der *Entity IfcSpace* ebenfalls nicht vorhanden.

Tabelle 5.4 stellt zusammenfassend den durchgeführten Vergleich dar. Das Symbol ✓ zeigt an, dass die vergleichbaren Daten ident zu denen aus *BIMQ* sind. Abweichungen sind textlich beschrieben.

Tab. 5.4: Vergleich der Daten in den einzelnen Prozessschritten zu *BIMQ*

<i>Entity</i>	<i>Bimplus</i>	<i>Allplan</i>	IFC
<i>IfcFooting</i>	✓	beim <i>Property Reference</i> kann kein Wert eingetragen werden	wird als <i>IfcSlab.Baseslab</i> ausgegeben
<i>IfcWall</i>	die Einheit des <i>Property ThermalTransmittance</i> wird falsch angezeigt	die Einheit des <i>Property ThermalTransmittance</i> wird falsch angezeigt, beim <i>Property Reference</i> kann kein Wert eingetragen werden	<i>Properties</i> ohne definierten Wert wurden nicht exportiert, bei <i>ThermalTransmittance</i> wurde keine Einheit ausgegeben, es kommt zu Überlagerungen mit in <i>Allplan</i> vordefinierten <i>Attributen</i>
<i>IfcColumn</i>	die Einheit des <i>Property ThermalTransmittance</i> wird falsch angezeigt	die Einheit des <i>Property ThermalTransmittance</i> wird falsch angezeigt, beim <i>Property Reference</i> kann kein Wert eingetragen werden	<i>Properties</i> ohne definierten Wert wurden nicht exportiert, es kommt zu Überlagerungen mit in <i>Allplan</i> vordefinierten <i>Attributen</i>
<i>IfcSlab</i>	die Einheit des <i>Property ThermalTransmittance</i> wird falsch angezeigt	die Einheit des <i>Property ThermalTransmittance</i> wird falsch angezeigt, beim <i>Property Reference</i> kann kein Wert eingetragen werden	<i>Properties</i> ohne definierten Wert wurden nicht exportiert, es kommt zu Überlagerungen mit in <i>Allplan</i> vordefinierten <i>Attributen</i>
<i>IfcSpace</i>	✓	beim <i>Property Reference</i> kann kein Wert eingetragen werden	<i>Properties</i> ohne definierten Wert wurden nicht exportiert

Ableitbare Aussagen

Basierend auf den angeführten Beobachtungen während des Versuchs können folgende qualitative Aussagen abgeleitet werden:

- Der Prozessablauf funktioniert.
- Die Übereinstimmung der Daten von der Vorlage in *BIMQ* bis zur, aus *Allplan* exportierten, IFC-Datei ist nicht vollständig gegeben.
- Vor allem beim Prozessschritt *IFC-Datei exportieren* in *Allplan* kommt es zu Abweichungen.
- Abweichungen sind hier eine abweichend exportierte *Entity*, teilweise nicht exportierte *Properties* und Einheiten derer sowie stellenweise die Überlagerung von *Properties* mit von *Allplan* vordefinierten *Attributen*.

5.5.2 Qualitative und quantitative Tests zu *Properties* in *Allplan*

Da bei dem allgemeinen Test Abweichungen bei den Prozessschritten in *Allplan* festgestellt wurden, werden hier weitere spezifische Tests durchgeführt. In Abschnitt 5.5.1 wurde bereits ein Test zu *Allplan* durchgeführt und davon qualitative Aussagen abgeleitet. Für eine ausführlichere Evaluierung des Prozessschrittes *IFC-Datei exportieren* wird hier ein zusätzlicher quantitativer Test durchgeführt. Dabei werden weitere *Entities* berücksichtigt: *IfcRoof*, *IfcBeam*, *IfcStair*, *IfcCovering*, *IfcDoor* und *IfcWindow*. Insgesamt werden 11 unterschiedliche *Entities* getestet. Die *Psets* und *Properties* sind weiterhin aus der Standardkonfiguration. Die Messziele sind quantitative Aussagen zu der Übereinstimmung der *Properties* in der IFC-Datei mit der Vorlage in *BIMQ*. Als Messgrößen werden dafür festgelegt:

- Anzahl an *Properties* mit korrekter Übereinstimmung des Namens, des Wertes, des *Data Type* und falls vorhanden auch der Einheit
- Anzahl an *Properties* mit Abweichungen von der korrekten Übereinstimmung
- Anzahl der abweichenden *Properties*, bei denen ein falscher *Data Type* und falls vorhanden keine oder eine falsche Einheit vorhanden sind
- Anzahl der Abweichenden, bei denen ein falscher Wert vorhanden ist
- Anzahl der Abweichenden, die nicht vorhanden sind

Von jeder *Entity* wird zumindest ein Element modelliert. Bei allen *Properties* wird, wo möglich, bei der Modellierung ein Wert angegeben. Dementsprechend wird bei jedem *Property* mit dem *Data Type* *IfcLabel* ein Wert angegeben. Abbildung 5.41 zeigt das erstellte Modell im *BIM-Viewer BIM-collabZOOM*. Es sind die modellierten *Entities* und beispielhaft das *Pset_BeamCommon* des ausgewählten Unterzuges sichtbar.

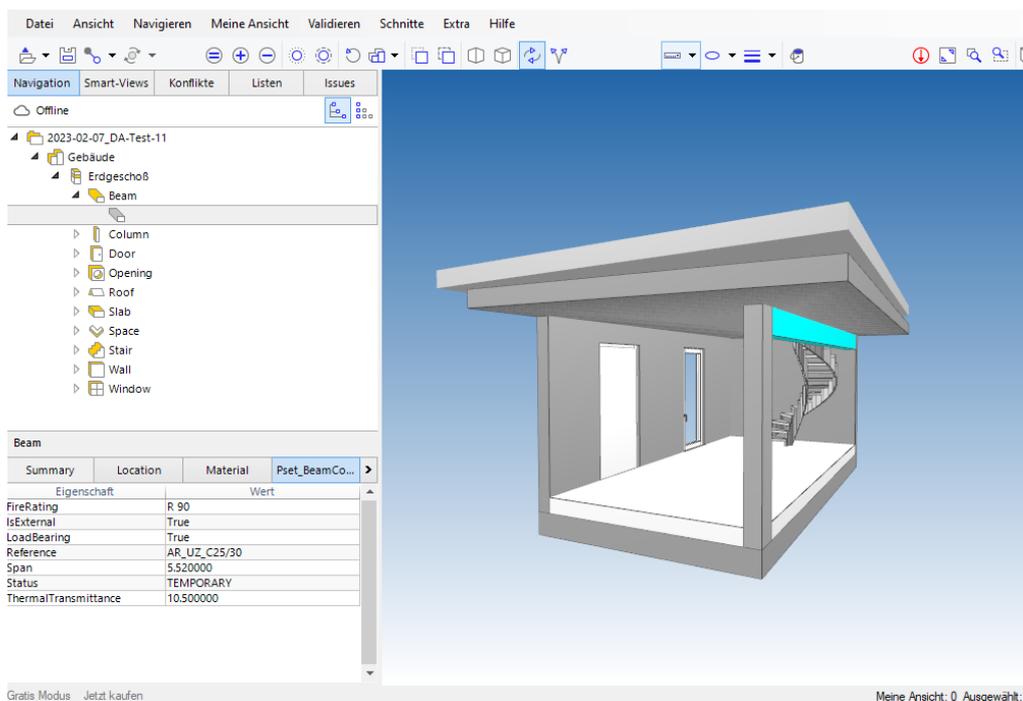


Abb. 5.41: Test-Modell im *Allplan-Test Properties*

Für die Auswertung wurde die IFC-Datei in einem *Viewer* geöffnet und die *Properties* auf Übereinstimmung kontrolliert. Der korrekte *Data type* und die Einheit wurde zusätzlich direkt in der IFC-Datei mithilfe eines Text-Editors überprüft.

Das Diagramm in Abbildung 5.42 stellt die *korrekt* exportierten *Properties* denen mit *Abweichungen* in Anzahl und Prozent gegenüber. Die Ergebnisse sind je *Entity* aufgeschlüsselt. Es ist ersichtlich, dass bei keiner *Entity* alle *Properties* korrekt exportiert wurden. Der Prozentsatz an abweichenden *Properties* reicht von 18% bis zu einem Extremwert von 100%. Der Mittelwert, der korrekt exportierten *Properties* je *Entity* beträgt 57%. Die *Entity IfcFooting* wurde, wie im Test in Abschnitt 5.5.1, nicht exportiert und bei der Angabe von Mittelwerten nicht berücksichtigt. Bei 5 der 10 *Entities* wurden ca. 75% der *Properties* korrekt exportiert, dies sind *IfcWindow*, *IfcWall*, *IfcSlab*, *IfcDoor* und *IfcCovering*.

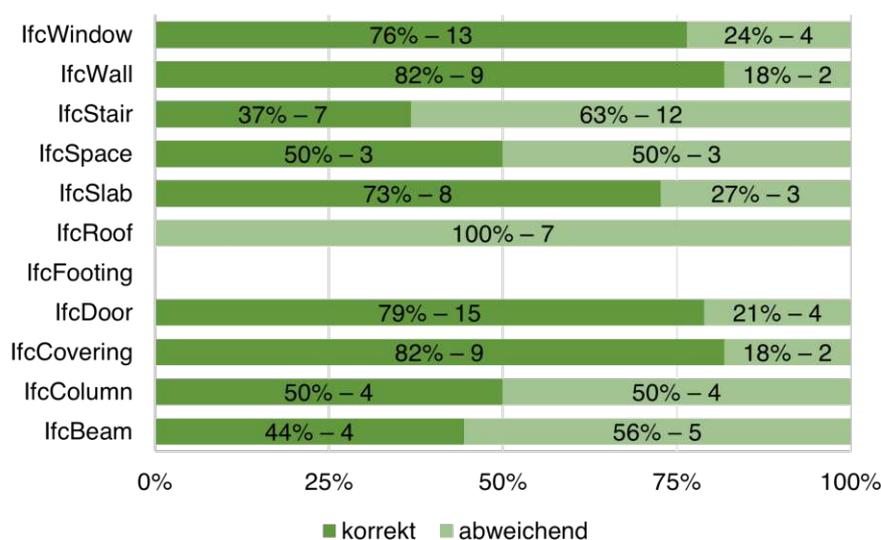


Abb. 5.42: Vergleich der *Properties* nach *Entity*

Bei der *Entity IfcRoof* wurden 0% korrekt exportiert. Code 5.5 zeigt den Ausschnitt der IFC-Datei beginnend bei der Zeile #3740 für *IfcRoof*. Es ist ersichtlich, dass zum *Pset_RoofCommon* die *Properties* in Zeile Nummer #3741 bis #3743 verknüpft sind. Dies sind jedoch von *Allplan* vordefinierte *Attribute*. Die *Properties* aus der Vorlage, mit den Zeilennummern #3754 bis #3759, wurden nicht mit dem *Pset* verknüpft.

Code 5.5: *Entity IfcRoof* im Ausschnitt der exportierten IFC-Datei

```

1  [...]
2  #3740= IFCROOF ('0aLN1Bbn5ACP9poUkpMmSc',#5,' ',,$,$,#3702,#3703,$,
    NOTDEFINED.);
3  #3741 = IFCPROPERTYENUMERATEDVALUE('Status',$(IFCLABEL('NEW')),$);
4  #3742 = IFCPROPERTYSINGLEVALUE('IsExternal',$(IFCBOOLEAN(.F.)),$);
5  #3743 = IFCPROPERTYSINGLEVALUE('ThermalTransmittance',$,
    IFC THERMALTRANSMITTANCEMEASURE(0.)),$);
6  [...]
7  #3754 = IFCPROPERTYSINGLEVALUE('IsExternal',$(IFCBOOLEAN(.T.)),$);
8  #3755 = IFCPROPERTYSINGLEVALUE('FireRating',$(IFCTEXT('3')),$);
9  #3756 = IFCPROPERTYSINGLEVALUE('LoadBearing',$(IFCBOOLEAN(.F.)),$);
10 #3757 = IFCPROPERTYSINGLEVALUE('Status (0)',$(IFCTEXT('TEMPORARY')),$);

```

```

11 #3758 =IFCPROPERTY SINGLEVALUE('ThermalTransmittance', $, IFCREAL(2.2), $);
12 #3759 =IFCPROPERTY SINGLEVALUE('AcousticRating', $, IFCTEXT('1'), $);
13 [...]
14 #3766 =IFCPROPERTY SET('3894gqq5n42BdS0t4jeLgD', #5, 'Pset_RoofCommon', $, (
    #3741, #3742, #3743));
15 [...]

```

Abbildung 5.43 stellt die Zusammensetzung der abweichenden *Properties* in Anzahl und Prozent über alle *Entities* dar. 65% der Abweichungen kommen zustande, weil der *Data Type* und, falls vorhanden, die Einheit des *Property* nicht dem der Vorlage entspricht. Der *Name* und *Wert* werden jedoch korrekt exportiert. Bei 9% wird ein falscher Wert – ein anderer als angegeben – exportiert. 24% der abweichenden *Properties* wurden als abweichend markiert, weil diese in der IFC-Datei nicht vorhanden bzw. nicht korrekt mit dem zugehörigen *Pset* verknüpft waren. Bei einem der abweichenden *Properties* wurde ein falscher Wert und eine falsche Einheit exportiert.

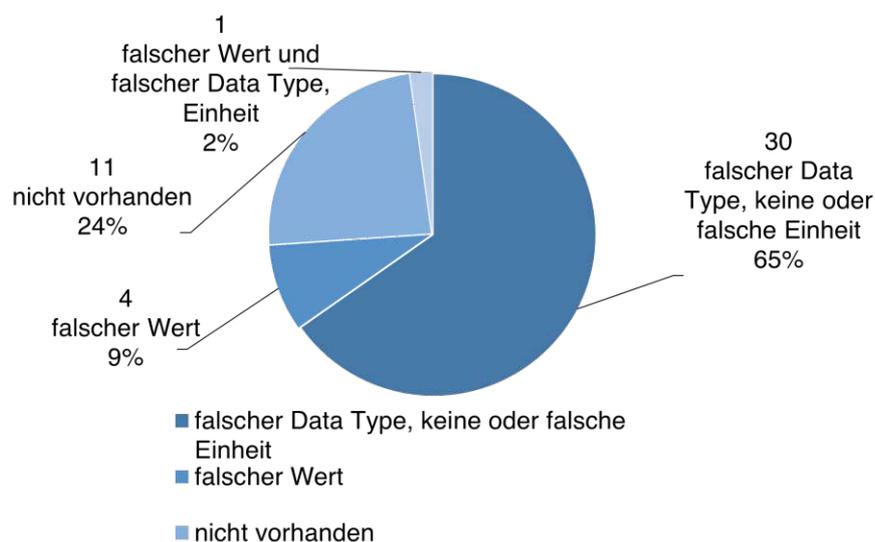


Abb. 5.43: Zusammensetzung der abweichenden *Properties*

Ein weiteres Diagramm über die Ergebnisse des Tests ist in Abbildung 5.44 dargestellt. Hier ist die Übereinstimmung der *Properties* nach ihrem *Data Type* bzw. deren Einheit aufgeschlüsselt und die Anzahl und Prozent angegeben. Es ist deutlich erkennbar, dass *Properties* mit *Data Type* *IfcBoolean* und *IfcLabel* in 95% bzw. 92% der Fälle mit korrektem *Namen*, *Wert* und *Data Type* exportiert wurden. Bei den übrigen, evaluierten *Data Type* kommt es in 100% der Fälle zu Abweichungen, da hier oft der *Data Type* oder die *Einheit* falsch exportiert wurden. Die *Data Type* mit der Endung *Measure*, wie z. B. *IfcThermalTransmittanceMeasure*, werden meistens und fälschlicherweise als *IfcReal* und ohne Einheit exportiert (siehe auch Code 5.5 und Zeilennummer #3758). Bei *Properties* mit dem *Data Type* *IfcIdentifier* kommt es zu 100% zu Abweichungen. Dies sind die *Properties Reference* bei denen in *Allplan* kein Wert angegeben werden konnte, oder ein Wert von einem vordefinierten *Allplan-Attribut* ausgegeben wurde.

Im allgemeinen Test in Abschnitt 5.5.1 wurde bereits gezeigt, dass manche Abweichungen zustande kommen, weil die getesteten Standard-*Properties* (Suffix *Common*) mit vordefinierten *Allplan-Attributen* überlagert werden. Im quantitativen Test kann konkreter evaluiert werden, dass diese Problematik zu den *Properties* mit falschem Wert, zu den nicht vorhandenen und zu denen mit falschem Wert und falscher Einheit führt. Dementsprechend sind in Summe 35% der

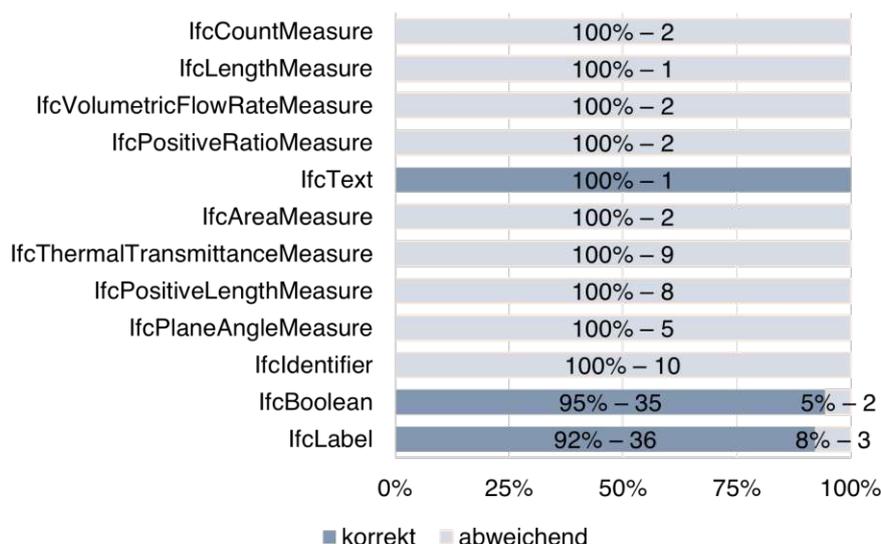


Abb. 5.44: Vergleich der *Properties* nach *Data Type* bzw. Einheit

abweichenden *Properties* dieser Problematik zuweisbar (vgl. Abbildung 5.43). Auf Basis dieser Ergebnisse wird folgende Hypothese aufgestellt, die mit einer Weiterführung des Tests überprüft wird:

Die Problematik der Überlagerung mit von *Allplan* vordefinierten *Attributen* ist bei individuell definierten *Properties* im Vergleich zu den Standard-*Properties* nicht vorhanden.

Überprüfung der Hypothese zur Problematik der Überlagerung mit von *Allplan* vordefinierten *Attributen*

Für die Überprüfung der Hypothese werden zu den modellierten *Entities* und den Standard-*Psets* die *Specific-Psets* von *bsAT* und die zur Verfügung stehenden *Psets* aus dem Forschungsprojekt *BRISE-Vienna* [27, 29] ergänzt. Da diese nicht standardisiert sind, kann das Vorhandensein in *Allplan* ausgeschlossen werden. Dies entspricht 65 weiteren *Properties* im Modell. In Abbildung 5.45 sind beispielsweise die zusätzlichen *Properties* der *Entity IfcDoor* dargestellt.

Ein Vergleich zwischen den korrekt exportierten und abweichenden je *Entity* ist hier nicht zielführend, weil die Verteilung der *Data Type* der zusätzlichen *Properties* abweichend zu jener der Standard-*Properties* ist. Wie der vorherige Test gezeigt hat, ist die Übereinstimmung der exportierten *Properties* mit denen der Vorlage vom jeweiligen *Data Type* abhängig. Zielführende Aussagen über die Ergebnisse können nur bei gleichzeitiger Betrachtung von Abbildung 5.46a und 5.46b gemacht werden. Die dargestellten Diagramme beziehen sich auf die 65 zusätzlichen *Properties*. Bei den zusätzlichen *Properties* kommt es bei 11,5¹³ von 65 *Properties* zu Abweichungen. In Abbildung 5.46a ist ersichtlich, dass hier der Anteil der *Properties* mit falschem Wert, und falschem *Data Type* und falscher Einheit bei 0% liegt. Abweichungen dieser Kategorie waren im vorhergehenden Test durch die Problematik der Überlagerung mit vordefinierten *Allplan-Attributen* verursacht worden. 2,5 *Properties* wurden in die IFC-Datei nicht exportiert (*nicht vorhanden*). 1,5 *Properties* davon sind vom *Data Type IfcBoolean*. Das eine weitere, mit *nicht vorhanden* markierte, *Property* ist vom *Data Type IfcLabel* und hat den Namen *Status*. Der Name

¹³Der Wert weicht von einer ganzen Zahl ab, weil im Modell 2 Wandobjekte vorhanden sind und bei der Anzahl der *Properties* der Mittelwert je verknüpfter *Entity* gebildet wurde.

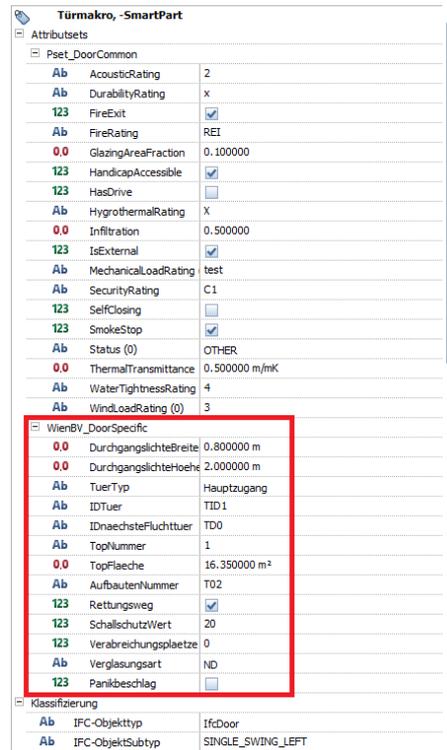
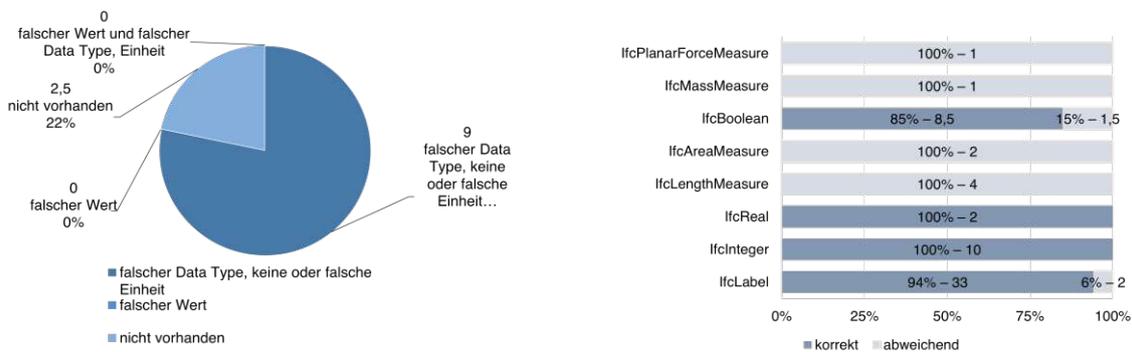


Abb. 5.45: Properties der Entity IfcDoor mit eingegebenen Werten

ist ident mit einem der Standard-Properties, was wieder zu der Problematik der Überlagerung führt. Die abweichenden Properties, die einen falschen Data Type und, falls vorhanden, keine Einheit aufweisen, stellen hier einen Großteil (78%) der abweichenden Properties dar. Abbildung 5.46b zeigt, dass die Data Type mit Suffix Measure weiterhin zu 100% nicht korrekt exportiert werden.

In Abbildung 5.46b ist ersichtlich, dass ergänzend zu den Properties mit Data Type IfcText, IfcBoolean und IfcLabel aus dem vorhergehenden Versuch, auch die mit Data Type IfcReal und IfcInteger zu über 90% korrekt exportiert wurden. Die Ergebnisse zeigen, dass die Hypothese bestätigt werden kann. Bei individuell definierten Properties tritt die Problematik der Überlagerung mit von Allplan vordefinierten Attributen nicht auf.



(a) Zusammensetzung der abweichenden Properties der individuellen Psets (b) Vergleich der Properties der individuellen Psets nach Data Type bzw. Einheit

Abb. 5.46: Diagramme zu den zusätzlichen, individuellen Psets

5.5.3 Tests von weiteren *Entities* und *PredefinedTypes*

In den folgenden Abschnitten werden weitere Features getestet, die von *BIMQ* und *Allplan* angeboten werden und für den Prozessablauf nützlich sein können. Diese umfassen weitere *Entities* und *PredefinedTypes*. Hierfür werden Funktionstests durchgeführt, bei denen überprüft wird, ob das Feature, im Zusammenspiel mit *BIMQ* und *Allplan*, wie gewünscht funktioniert.

Test von weiteren *Entities*

Bei den Versuchen in den Abschnitten 5.5.1 und 5.5.2 wurden primär *Entities* der Klasse *IfcBuiltElement* getestet. Im Folgenden wird evaluiert, ob der Prozessablauf unter der Verwendung von *BIMQ* und *Allplan* auch mit beliebigen *Entities* anderer Klassen funktioniert.

Hierfür werden jeweils eine *Entity* aus den Klassen *IfcFurnishingElement*, *IfcSpatialStructureElement* und *IfcDistributionElement* ausgewählt. Diese, innerhalb der Klasse beliebig, ausgewählten *Entities* sind: *IfcFurniture*, *IfcSolarDevice*, *IfcSite*. Als *Allplan*-Mapping in *BIMQ* liegen entsprechend *Möbel*, *Solarbetriebenes Gerät* und *Gelände* nahe. Die zu verknüpfenden *Psets* der Standardkonfiguration sind: *Pset_FurnitureTypeCommon*, *Pset_SolarDeviceTypeCommon* und *Pset_SiteCommon*.

Das Erstellen der *Projektanforderungen* in *BIMQ* funktioniert wie in Abschnitt 5.2.7 beschrieben. Nach dem Export aus *BIMQ* ist in *Bimplus* erkennbar, dass die *Psets* und *Properties* vollumfänglich exportiert wurden, wie in Abbildung 5.47 für das *Pset_SiteCommon* ersichtlich.

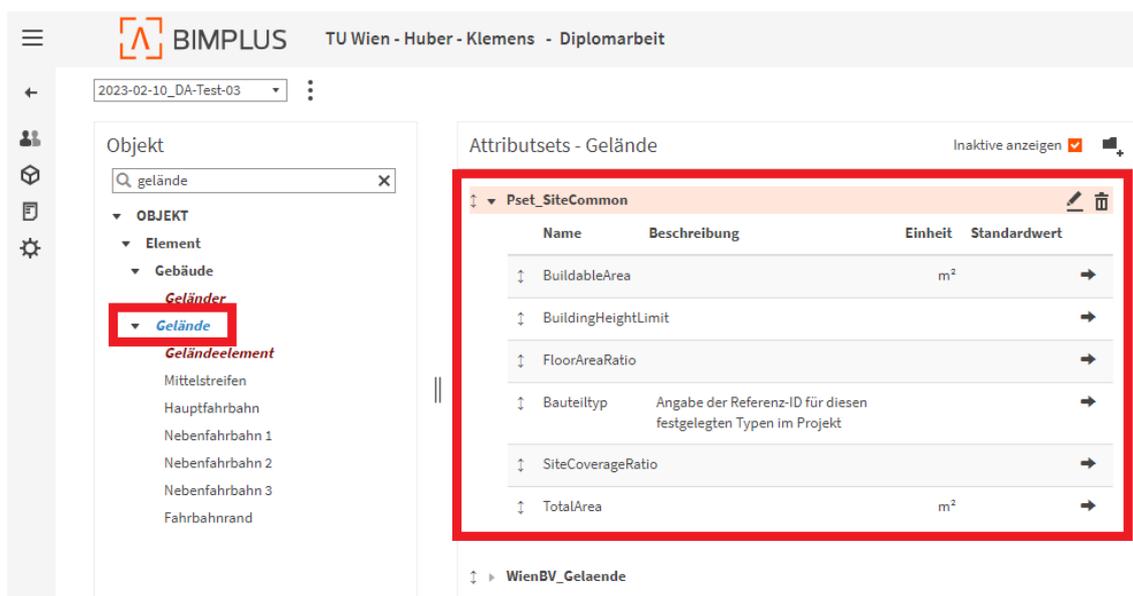


Abb. 5.47: Attributsets beim *Bimplus*-Objekt *Gelände*

Nachdem die Vorlage in *Allplan* geladen wurde, stellt sich die Frage wie ein *Möbel*, *Solarbetriebenes Gerät* und *Gelände* modelliert werden können. Bei intuitiven Verwenden der *Suchen*-Funktion zum Finden möglicher Befehle oder *Objekte* wird angezeigt, dass der Begriff *Möbel* in *Allplan* nicht gefunden werden kann, siehe Abbildung 5.48. Im von *Allplan* herausgegebenen *BIM-Kompendium* [21, S. 219] wird zu *IfcFurniture* folgendes erläutert:

„[...] Die Objekte können in 3D frei modelliert und beim Zusammensetzen aus mehreren Einzelteilen anschließend zu einem **Makro** bzw. **SmartPart** zusammengefasst werden. Das Verwenden und gegebenenfalls Anpassen von (Hersteller-)Content ist ebenfalls möglich. Unabhängig davon, aus welcher Quelle das Objekt stammt, erfolgt die

5.5 Durchführung von Versuchen

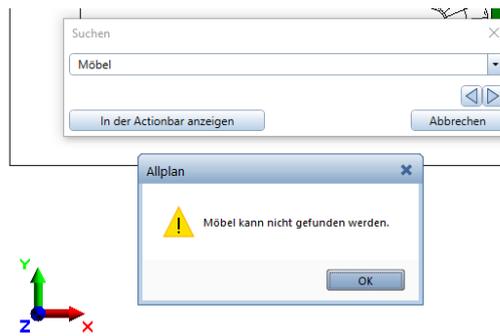


Abb. 5.48: Ergebnis der Funktion *Suchen* nach dem Begriff *Möbel*

*Definition als Mobiliar ausschließlich über das Attribut **IfcObjectType**, der manuell zugewiesen werden muss.“*

Dies bedeutet, dass kein spezifischer Befehl bzw. *Allplan-Objekt* für die *Entity IfcFurniture* vorhanden ist. Die Zuweisung eines *Objektes* zu dieser *Entity* muss manuell durchgeführt. Für den Funktionstest wird ein *3D-Objekt Zylinder* modelliert und beim *Attribut IFC-Objekttyp* der Wert *IfcFurniture* ausgewählt, siehe Abbildung 5.49. Hier sind die angegebenen auswählbaren *Entities* ersichtlich. Eine manuelle Eingabe einer *Entity* ist ebenfalls möglich.

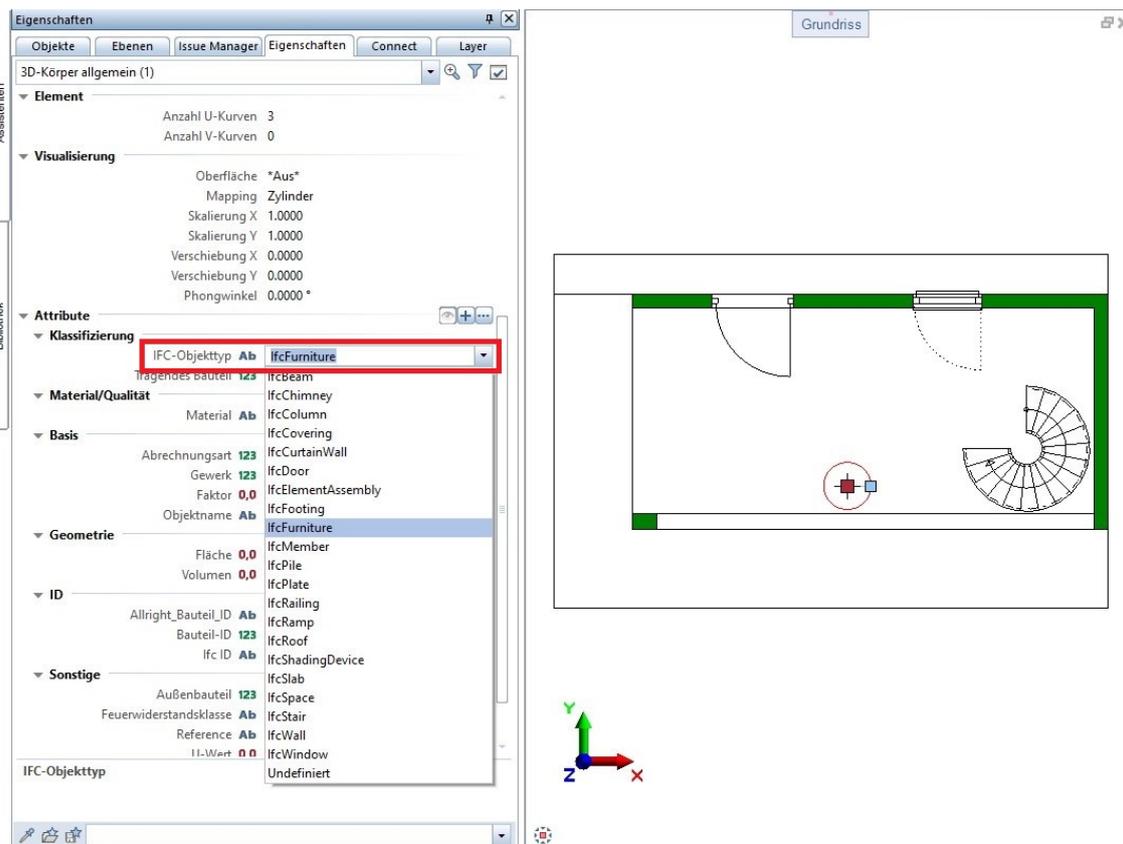


Abb. 5.49: *Allplan-Attribut IFC-Objekttyp*

Nach dieser Auswahl werden jedoch die, in der Vorlage definierten, *Psets* und *Properties* nicht angezeigt. Auch bei der Verwendung von den vorgeschlagenen *Makro* oder *SmartPart* werden diese nicht hinzugefügt. Hier ist die Funktionstüchtigkeit nicht gegeben.

Für die *Entity IfcSolarDevice* ist ebenfalls kein zugehöriges *Allplan*-Objekt verfügbar. *Allplan* gibt hier in [21] an, dass *TGA-Komponenten* in einem Zusatzmodul *Technische Gebäudeausrüstung* vorhanden sind. Im Regelfall ist die Zuweisung der *Entity* jedoch auch über das *Attribut IFC-Objekttyp* zu steuern. Um dies zu testen, wird ein weiterer *3D-Körper* modelliert und dem entsprechenden *Attribut* der Wert *IfcSolarDevice* manuell eingegeben. In der vordefinierten Auswahlliste, siehe Abbildung 5.49, wird dieser nämlich nicht vorgeschlagen. Die in der Vorlage verknüpften *Psets* und *Properties* werden auch hier nicht angezeigt. Es kann keine Aussage getroffen werden, ob mit dem erwähnten Zusatzmodul die korrekte Anzeige der definierten *Psets* gegeben wäre, da dem Verfasser dieses Zusatzmodul nicht zur Verfügung stand.

Bezüglich *IfcSite* wird von *Allplan* in [21, S. 163] darauf hingewiesen, dass:

„Bei der Typzuweisung an ein Zeichnungsobjekt sollten *IfcBuilding*, *IfcSite* und *IfcBuildingStorey* generell nicht verwendet werden, auch wenn **Allplan** Ihnen hierzu im Modul **Städtebau** entsprechende Objekte zur Verfügung stellt, auf die diese Typzuweisung passen würde. Sie sind innerhalb der IFC-Definitionen und Vorgaben ausschließlich für die einzelnen Stufen der Gliederungsstruktur verantwortlich und stellen darin keine realen Objekte dar.“

Es wird weiter angegeben, dass in der *Gliederungsstruktur* die *Strukturstufe Liegenschaft* der *Entity IfcSite* entspricht. Demzufolge ist das in *BIMQ* gewählte *Allplan*-Mapping *Gelände* falsch gewählt. Innerhalb der in *BIMQ* auswählbar Mappings befindet sich jedoch kein Mapping mit der Bezeichnung *Liegenschaft*. Dadurch kann nicht bestimmt werden, wie und ob eine Zuweisung und in weiterer Folge die *Psets* und *Properties* der Vorlage angegeben werden können. Zur Vollständigkeit wird auch die exportierte IFC-Datei überprüft. Hier sind die entsprechenden Elemente mit angegebener *Entity* vorhanden, jedoch ebenfalls keine *Psets* oder *Properties* aus der Vorlage.

Die Ergebnisse des Funktionstests zu den *Entities IfcFurniture*, *IfcSolarDevice* und *IfcSite* sind somit negativ zu bewerten. Die in der *Projektanforderung* angegebenen *Psets* und *Properties* dieser *Entities* sind in *Allplan* nicht vorhanden. Das negative Testergebnis führt zu der Frage, bei welchen *Entities* der Prozessablauf grundsätzlich funktioniert. Aus den vorangegangenen Versuchen ist bekannt, dass bei folgenden *Entities* die *Psets* und *Properties* der Vorlage in *Allplan* korrekt angezeigt werden: *IfcFooting*, *IfcWall*, *IfcColumn*, *IfcSlab*, *IfcSpace*, *IfcRoof*, *IfcBeam*, *IfcStair*, *IfcCovering*, *IfcDoor* und *IfcWindow*.

Basierend auf dem erkennbaren Schema kann folgende Hypothese formuliert werden:

Die *Entities*, bei denen die *Psets* und *Properties* der Vorlage in *Allplan* korrekt angezeigt werden, sind jene zu denen ein direktes *Allplan*-Objekt vorhanden ist.

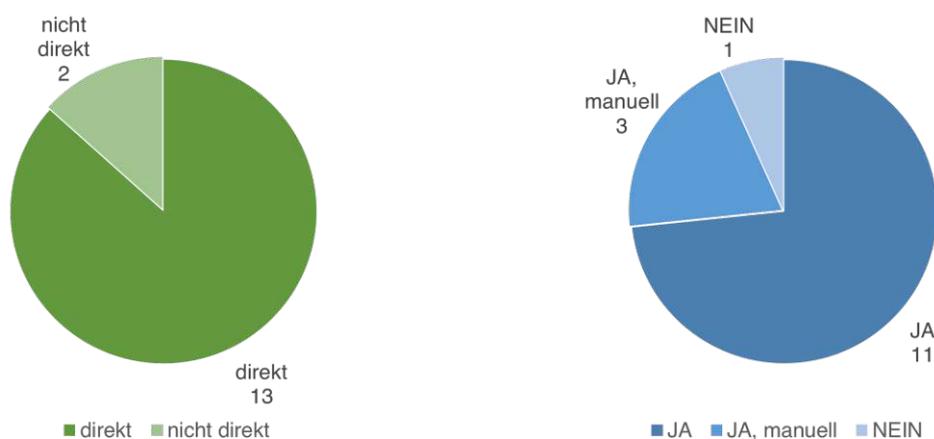
Da eine detaillierte Überprüfung aller möglichen *Entities* den Umfang dieser Diplomarbeit übersteigen würde, wird für die Überprüfung der Hypothese eine im Hochbau häufig verwendete Auswahl an *Entities* getroffen und diese auf die Klasse *IfcBuiltElement* beschränkt. Die Auswahl stellen die *Entities* dar, wie sie auch im BIM-Pilotprojekt [26] von *bsAT* verwendet wurden. Dies sind folgende 15 *Entities*: *IfcBeam*, *IfcColumn*, *IfcCovering*, *IfcCurtainWall*, *IfcPile*, *IfcDoor*, *IfcFooting*, *IfcPlate*, *IfcRailing*, *IfcRamp*, *IfcRoof*, *IfcSlab*, *IfcStair*, *IfcWall* und *IfcWindow*. Das Messziel ist hier eine qualitative Aussage zur aufgestellten Hypothese formulieren zu können.

Als Messgrößen werden festgelegt:

- die Anzahl der ausgewählten *Entities* zu denen ein oder kein direktes *Allplan-Objekt* vorhanden ist – *direkt* und *nicht direkt*
- die Anzahl, bei denen die *Psets* und *Properties* korrekt und automatisch angezeigt werden – *JA*
- die Anzahl, an *Entities* bei denen kein direktes *Allplan-Objekt* vorhanden ist, aber die *Psets* und *Properties* korrekt angezeigt werden, wenn manuell die gewünschte *Entity* zugewiesen wird – *JA, manuell*
- die Anzahl, bei denen die *Psets* und *Properties* in keinem Fall angezeigt werden – *NEIN*
- die Anzahl, der Kombination aus *direkt* und *nicht direkt* sowie *JA*, *JA, manuell* und *NEIN*

Für die Versuchsdurchführung werden die *Entities* in *BIMQ* im Menü *Projektkonfiguration*, wie in den vorangegangenen Test, mit den entsprechenden Standard-*Psets* und -*Properties* verknüpft. Nachdem die Vorlage exportiert und in *Allplan* geladen ist, werden die entsprechenden *Allplan-Objekte* Schritt für Schritt ausgewählt und ein beispielhaftes Element derer modelliert. In den Eigenschaften der Elemente ist ersichtlich, ob die *Psets* und *Properties* der Vorlage korrekt geladen wurden und hier sichtbar sind. Bei *Entities*, bei denen kein direktes *Allplan-Objekt* vorhanden ist, wird beispielsweise ein *3D-Körper* modelliert und beim *Attribut IFC-Objekttyp* die gewünschte *Entity* eingegeben. Darauffolgend wird kontrolliert, ob die *Psets* und *Properties* geladen werden.

Zu den Ergebnissen können nur qualitative Aussagen getroffen werden, da die Auswahl von 15 *Entities* nicht repräsentativ für alle *Entities* ist. Die Ergebnisse zu den ausgewählten *Entities* werden in Tabelle 5.5 aufgelistet und in den folgenden Abbildungen übersichtlich dargestellt und erläutert. In Abbildung 5.50a ist ersichtlich, dass für 13 der ausgewählten *Entities* ein *direktes Allplan-Objekt* vorhanden ist. Die beiden *Entities* auf die dies nicht zutrifft sind: *IfcPile* und *IfcPlate*. In Abbildung 5.50b ist erkennbar, dass bei 11 *Entities* die *Psets* und *Properties* der Vorlage automatisch geladen und korrekt angezeigt werden. Bei 3 *Entities* ist dies der Fall, nachdem die *Entity* im *Attribut IFC-Objekttyp* manuell angegeben wurde. Bei einer *Entity* werden die *Psets* und *Properties* auch dann nicht angezeigt.



(a) Anzahl an *Entities* mit *direkt* und *nicht direkt* vorhandenem *Allplan-Objekt* (b) Anzahl an *Entities* je nach angezeigten *Psets* und *Properties*

Abb. 5.50: Diagramme zu den getesteten *Entities*

In Abbildung 5.51 sind die Einzelergebnisse kombiniert dargestellt. Bei 11 der ausgewählten *Entities* sind bei *direkt* vorhandenen *Allplan-Objekten* auch die *Psets* und *Properties* aus der Vorlage automatisch vorhanden. Die Kategorie *direkt + JA, manuell* zeigt, dass in einem Fall zwar ein *Allplan-Objekt* zur *Entity IfcRamp* vorhanden ist, die *Properties* jedoch erst nach manuellem Zuweisen der *Entity* angezeigt werden. Bei einer *Entity IfcRailing* ist zwar ein *Allplan-Objekt* vorhanden, die zugewiesenen *Properties* der Vorlage werden jedoch nicht angezeigt. Per Definition ist ausgeschlossen, dass es eine *Entity* gibt, zu der kein direktes *Allplan-Objekt* vorhanden ist, jedoch trotzdem die *Psets* und *Properties* automatisch vorhanden sind (*nicht direkt + JA*). Die *Psets* und *Properties* können abseits von *Allplan-Objekten* nur angezeigt werden, wenn zuerst die gewünschte *Entity* manuell zugewiesen wird. Dies wird durch die Bezeichnung *nicht direkt + JA, manuell* abgebildet. Bei 2 *Entities* ist zwar kein *Allplan-Objekt* vorhanden, jedoch werden die *Properties* automatisch geladen, nachdem die *Entity* manuell zugewiesen wurde. Es gibt unter den hier ausgewählten *Entities* keine, bei der kein *Allplan-Objekt* vorhanden ist und auch nach manuellem Zuweisen keine *Properties* der Vorlage angezeigt werden. In diese Kategorie wäre zumindest die *Entity IfcFurniture* aus dem vorherigen Test einzustufen.

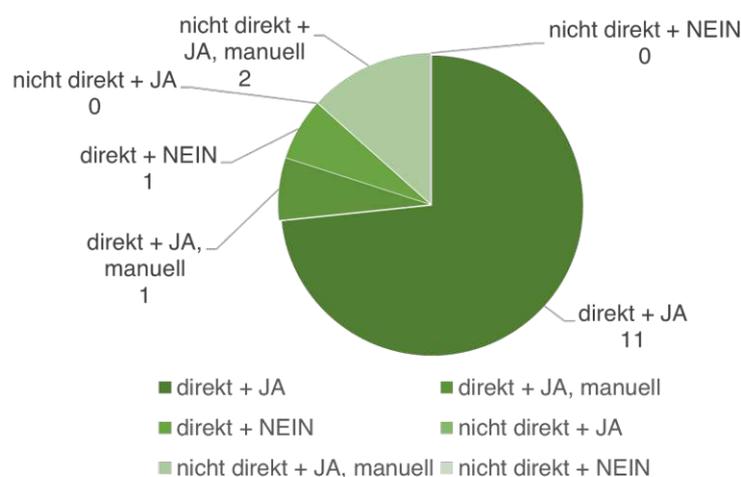


Abb. 5.51: Zusammensetzung an *Entities*, jeweils mit *direkt* oder *nicht direkt* vorhandenen *Allplan-Objekt* und korrekt automatisch (JA), korrekt bei manueller Zuweisung (JA, manuell) und in keinem Fall angezeigten *Psets* und *Properties*

Um den Test vollständig in einen Kontext mit dem Prozessablauf stellen zu können, wird auch die exportierte IFC-Datei überprüft. In Tabelle 5.5 ist ergänzend angegeben, ob die *Entity* (Spalte *IFC Entity*) sowie die *Psets* und *Properties* (Spalte *IFC Psets und Properties*) in die IFC-Datei exportiert wurden. Wenn der Name mit dem in der Vorlage übereinstimmt, wurde in der Tabelle die Bezeichnung *Ja* angegeben. Falls *Psets* oder *Properties* in der IFC-Datei nicht vorhanden sind, wird die Tabellenzelle mit *Abw.* (Abweichung) markiert.

Eine mögliche, qualitative Aussage ist, dass es, entgegen der aufgestellten Hypothese, mindestens eine *Entity* (*IfcRailing*) gibt, für die zwar ein *direktes Allplan-Objekt* vorhanden ist, jedoch die *Psets* und *Properties* der Vorlage trotzdem nicht angezeigt werden. Die Hypothese kann somit nicht bestätigt werden. Hervorzuheben ist, dass es *Entities* gibt für die kein direktes *Allplan-Objekt* vorhanden ist, bei denen aber trotzdem nach manueller Zuweisung der *Entity* die vorgesehenen *Psets* und *Properties* geladen und angezeigt werden. Diese *Entities* werden ebenfalls, inklusive der *Psets* und *Properties*, in die IFC-Datei exportiert. In diesem Test stand die Überprüfung der aufgestellten Hypothese im Vordergrund. Eine repräsentative, quantitative Aussagen ist nur über die hier getätigte Einschränkung auf die, im Hochbau häufig verwendeten, *Entities* der Klasse

IfcBuiltElement möglich. Abbildung 5.52 zeigt die quantitative Zusammensetzung der Spalte *IFC Psets und Properties* aus Tabelle 5.5. Es ist ersichtlich, dass bei 11 der 15 *Entities* (73%) die *Psets* und *Properties* korrekt in die IFC-Datei exportiert wurden.

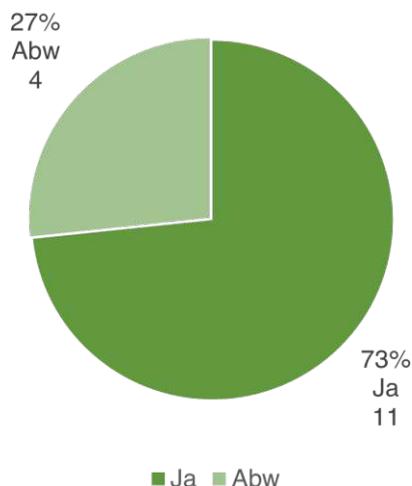


Abb. 5.52: Zusammensetzung der Anzahl an *Entities* der Spalte *IFC Psets und Properties* aus Tabelle 5.5 an korrekt (JA) oder mit Abweichungen (Abw) in die IFC-Datei exportierten *Psets* und *Properties*

Tab. 5.5: Vergleich der *Entities*

<i>Entity</i>	<i>Mapping</i>	<i>Allplan-Objekt</i>	<i>Allplan-Objekt vorhanden</i>	<i>Psets und Properties angezeigt</i>	<i>IFC Entity</i>	<i>IFC Psets und Properties</i>
IfcBeam	Balken	Unterzug, Überzug	direkt	JA	Ja	Ja
IfcColumn	Stütze	Stütze	direkt	JA	Ja	Ja
IfcCovering	Bodenbelag etc.	Bodenfläche, Bodenbelag etc.	direkt	JA	Ja	Ja
IfcCurtainWall	Vorhangfassade	Fassade	direkt	JA	Ja	Ja
IfcPile	Gründungspfahl		nicht direkt	JA, manuell	Ja	Ja
IfcDoor	Tür	Tür, Tor-SmartPart	direkt	JA	Ja	Ja
IfcFooting	Fundament	Plattenfundament etc.	direkt	JA	Ja	Abw.
IfcPlate	Platte		nicht direkt	JA, manuell	Ja	Ja
IfcRailing	Geländer	Geländer	direkt	NEIN	Ja	Abw.

IfcRamp	Rampe	Gerade Rampen etc.	direkt	JA, manuell	Nein	Abw.
IfcRoof	Dach	Dachhaut	direkt	JA	Ja	Abw.
IfcSlab	Deckenplatte	Decke	direkt	JA	Ja	Ja
IfcStair	Treppe	Gerade Treppe etc.	direkt	JA	Ja	Ja
IfcWall	Wand	Wand	direkt	JA	Ja	Ja
IfcWindow	Fenster	Fenster- SmartPart	direkt	JA	Ja	Ja

Test zu *PredefinedTypes*

In *BIMQ* besteht die Möglichkeit bei den *IFC-Mappings* der *Entities* auch den *PredefinedType* anzugeben. Durch diese Funktion können einzelnen *PredefinedTypes* gezielt *Psets* und *Properties* zugewiesen werden. *PredefinedTypes* sind optionale Subtypen von *Entities*. Ein *PredefinedType* der *Entity IfcBeam* ist *LINTEL* (englisch für Sturz).

Hier stellt sich die Frage, ob dies mit *Allplan* genutzt werden kann. Neben dem angegebenen *IFC-Mapping* wäre auch das *Allplan-Mapping* an den *PredefinedType* anzupassen. Die *Allplan-Mappings* können jedoch, wie bereits beschrieben, nicht manuell festgelegt werden, sondern sind an die Auswahlmöglichkeiten der *Referenzrichtlinie* gebunden. Diese haben ihren Ursprung in den *Objekten* in *Bimplus*. Für den Funktionstest kann als Beispiel die *Entity IfcBeam* herangezogen werden. Die hier in der Klasse *IfcBeamTypeEnum* definierten *PredefinedTypes* sind: *BEAM*, *CORNICE*, *DIAPHRAGM*, *EDGEBEAM*, *GIRDER_SEGMENT*, *HATSTONE*, *HOLLOWCORE*, *JOIST*, *LINTEL*, *PIERCAP*, *SPANDREL*, *T_BEAM*, *USERDEFINED*, *NOTDEFINED*. Für einzelne dieser Definitionen können mögliche *Allplan-Mappings* gefunden werden. Beispielsweise das *Mapping Sturz* für *LINTEL* oder *Gitterträger* für *T_BEAM*. Um diese beiden *PredefinedTypes* testen zu können, wird wieder der Prozessablauf in *BIMQ* und *Bimplus* durchlaufen und die exportierte Vorlage in *Allplan* geladen.

Zum *PredefinedType LINTEL* ist ein *Allplan-Objekt* namens *Sturz* vorhanden und die *Psets* und *Properties* der Vorlage werden in den *Eigenschaften* angezeigt. Zu *Gitterträger* existiert kein *Allplan-Objekt*. Hierfür wird eine manuelle Zuweisung getestet. Neben dem *Allplan-Attribut IFC-Objektyp* muss für den *PredefinedType* das *Attribut IFC-ObjektSubtyp* [21] angegeben werden. Dieses Attribut ist meistens in den *Eigenschaften* eines Bauteils nicht sofort ersichtlich, sondern muss mit der Funktion *Neues Attribut zuweisen* ergänzt werden. Im Versuch wird ein *3D-Körper* modelliert und zuerst bei *IFC-Objektyp* der Wert *IfcBeam* angegeben. Hierbei werden direkt die mit der *Entity IfcBeam* verknüpften *Psets* und *Properties* der Vorlage geladen. Beim *Attribut IFC-ObjektSubtyp* ist keine Auswahlliste vorhanden, sondern der Wert *T_BEAM* muss manuell eingegeben werden. Danach erfolgte jedoch keine Anzeige der, spezifisch für diesen *PredefinedType* verknüpften, *Psets* und *Properties*.

In der *IFC-Datei* ist ersichtlich, dass der modellierte *Sturz* nicht mit dem *PredefinedType LINTEL* exportiert wurde, sondern mit *NOTDEFINED*. Die zugewiesenen *Psets* und *Properties* sind jedoch verknüpft, siehe auch den Code 5.6. Beim *3D-Körper* sind *IfcBeam* und *PredefinedType T_BEAM* korrekt vorhanden, jedoch scheinen hier, die in der Vorlage zugewiesenen, *Psets* und *Properties* nicht auf. Die Funktionstüchtigkeit des Zuweisens von *Psets* und *Properties* an einzelne *PredefinedTypes* ist in *Allplan* somit nicht gegeben.

Code 5.6: Ausschnitt der exportierten IFC-Datei

```
1 [...]
2 #1372= IFCBEAM ('2eIYc325fBtPIxnm0eU2E$',#5,' ',,$,$,#1351,#1352,$,.
   NOTDEFINED.);
3 #1373=IFCPROPERTYENUMERATEDVALUE('Status',$(IFCLABEL('NEW')),,$);
4 #1374=IFCPROPERTYSINGLEVALUE('AufbautenNummer',,$,IFCLABEL('gdf'),,$)
   ;
5 #1375=IFCPROPERTYSINGLEVALUE('Allright_Bauteil_ID',,$,IFCTEXT('0011
   Srz0000001237'),,$);
6 #1376=IFCPROPERTYSINGLEVALUE('Abrechnungsart',,$,IFCTEXT('m'),,$);
7 #1377=IFCPROPERTYSINGLEVALUE('Objektname',,$,IFCTEXT('Sturz'),,$);
8 #1378=IFCPROPERTYSET('0By1xanoH8Lw4pZwvpqIhP',#5,'AllplanAttributes',,$
   ,(#1375,#1376,#1377));
9 #1379=IFCRELDEFINESBYPROPERTIES('10LvwaCDv9F8XE$VT0Ei5i',#5,$,$,(#1372)
   ,#1378);
10 #1380=IFCPROPERTYSET('2jccvACGD2RxNMsueaaPNM',#5,'Pset_BeamCommon',,$
   ,(#1373));
11 #1381=IFCRELDEFINESBYPROPERTIES('0STncJg7LE4Qh3EKsYhP0v',#5,$,$,(#1372)
   ,#1380);
12 #1382=IFCPROPERTYSET('0I$xyAidX6LQUR_Tc84r5f',#5,'
   WienBV_BeamSpecific',,$,(#1374));
13 [...]
14 #61666= IFCBEAM ('3IxiUwc1DDmuyUb5oBA00v',#5,' ',,$,$,#61633,#61634,$,.
   T_BEAM.);
15 [...]
```

Kapitel 6

Diskussion, Empfehlungen und Ausblick

In diesem Kapitel werden zu Beginn die Ergebnisse der durchgeführten Tests diskutiert. Im Anschluss werden die Forschungsfragen beantwortet. Darauf aufbauend werden Empfehlungen zur Verbesserung der genutzten Software-Produkte *BIMQ*, *Bimplus* und *Allplan* formuliert. Diese beziehen sich auf die Inhalte der Arbeit und sind teilweise um einen Ausblick auf zukünftige Entwicklungen im Kontext des openBIM-Prozesses erweitert.

6.1 Diskussion der durchgeführten Tests

In den Tests wurden der Prozessablauf unter der Verwendung von *BIMQ* und *Allplan* überprüft. Der qualitative Test in Abschnitt 5.5.1 hat gezeigt, dass der Prozessablauf mit *BIMQ* und *Allplan* funktioniert. Ein beobachtetes Ergebnis war, dass es zu Abweichungen in der Übereinstimmung der *Properties* zwischen der Vorlage in *BIMQ* und in *Allplan* sowie der exportierten IFC-Datei kommt. Die quantitative Evaluierung dieser Abweichungen hat ergeben, dass bezogen auf die *Entities* im Mittel 57% der *Properties* korrekt bis in die IFC-Datei weitergegeben werden, siehe Abschnitt 5.5.2. Ein *Property* wurde bei diesem Test als *korrekt* exportiert markiert, wenn in der IFC-Datei *Name*, *Wert*, *Data Type*, und falls vorhanden auch die *Einheit*, mit der Vorlage aus *BIMQ* übereinstimmen. Bei den hiervon abweichenden *Properties* ist es im Test in 65% der Fälle der *Data Type*, der in der IFC-Datei von dem in der Vorlage abweicht. Weitere Abweichungen kommen zustande, weil beim Exportieren der IFC-Datei *Psets* und *Properties* mit von *Allplan* vordefinierte *Attributen* überlagert werden. Bei einem weiterführenden Test konnte jedoch bestätigt werden, dass diese Problematik nur bei den Standard-*Properties* (Suffix *Common*) auftritt. Bei in der Vorlage definierten, spezifischen *Properties* kommt es beim Export der IFC-Datei ebenfalls zu Abweichungen des *Data Type*. In den Ergebnissen war ersichtlich, dass der Prozentsatz an korrekt exportierten *Properties* von dessen *Data Type* abhängig ist. *Properties* mit dem *Data Type* *IfcText*, *IfcBoolean*, *IfcLabel*, *IfcReal* und *IfcInteger* wurden in über 90% der Fälle korrekt exportiert. Die Abweichungen kommen vermehrt bei *Data Type* mit Endung *Measure*, wie *IfcThermalTransmittanceMeasure*, vor.

In Abschnitt 5.5.3 hat ein qualitativer Test gezeigt, dass die vollständige Funktionstüchtigkeit des Prozessablaufes nur für bestimmte *Entities* möglich ist. Diese sind *IfcBeam*, *IfcColumn*, *IfcCovering*, *IfcCurtainWall*, *IfcPile*¹, *IfcDoor*, *IfcPlate*, *IfcSlab*, *IfcStair*, *IfcWall* und *IfcWindow*. Ein weiterer Test hat ergeben, dass die Funktion der Zuweisung von *Psets* und *Properties* spezifisch an *PredefinedTypes*, von *Allplan* nicht unterstützt wird.

Allgemein kann bestätigt werden, dass der Prozessablauf, unter der Verwendung von *BIMQ* und *Allplan*, für bestimmte *Entities* funktioniert, es jedoch beim Export der IFC-Datei vereinzelt zu Abweichungen kommt. Aus Sicht des Verfassers ist folgende Aussage zur Bewertung der Ergebnisse möglich: Bei 11 der getesteten 15 *Entities*, der Klasse *IfcBuiltElement* und die häufig im Hochbau verwendet werden, ist die Funktionstüchtigkeit des Prozessablaufes durch *Allplan* gegeben. Für *All-*

¹Die Funktionstüchtigkeit ist gegeben bei manueller Zuweisung der entsprechenden *Entity* in *Allplan*.

plan stellt der Hochbau den Hauptanwendungsbereich der Standardversion (*Paket Architecture*) [2] dar.

6.2 Beantwortung der Forschungsfragen

Die folgenden Antworten führen die Inhalte dieser Diplomarbeit mit den in der Einleitung gestellten Forschungsfragen zusammen.

Wie sieht die Integrierung eines Datenstrukturwerkzeuges im *openBIM*-Prozess aus?

Datenstrukturwerkzeuge und deren Möglichkeiten im *openBIM*-Prozess wurden im Abschnitt 3.2 beschrieben. Mit ihnen können Strukturen an alphanumerische Informationsanforderungen angelegt und verwaltet werden. Dies erfolgt zentral in einer Datenbank. Die erstellten Strukturen können maschinenlesbar oder in Software-spezifischen Formaten für BIM-Autorensoftware oder Prüfsoftware ausgegeben werden.

Wie kann, innerhalb eines Anwendungsfalls, eine Standardkonfiguration mittels eines Datenstrukturwerkzeuges erstellt und in die BIM-Autorensoftware *Allplan* importiert werden?

In Kapitel 5 wurden der Prozessablauf dargestellt und die einzelnen Schritte beschrieben. Die Daten an *Entities*, *Psets* und *Properties* können von der IFC-Spezifikation übernommen und in Excel aufbereitet werden. Im Datenstrukturwerkzeug *BIMQ* wird die Strukturierung und Verknüpfung dieser Daten zu Informationsanforderungen erstellt sowie die Zuordnung zum Fachmodell, der Projektphase und dem Anwendungsfall durchgeführt. Bei der BIM-Autorensoftware *Allplan* ist ein Zwischenschritt in der *CDE Bimplus* erforderlich, wo die aus *BIMQ* exportierte Vorlage dem Projekt zugewiesen wird. Durch die Verknüpfung der Projekte von *Bimplus* und *Allplan* wird die Vorlage automatisch geladen. Anschließend kann in *Allplan* mit den *Psets* und *Properties* der Vorlage modelliert werden. Beim Export einer IFC-Datei ist eine aus *BIMQ* exportierte Zuordnungsdatei zu laden.

Welche Herausforderungen entstehen bei der Implementierung eines Datenstrukturwerkzeuges in einem konkreten Fallbeispiel?

Bei der Nutzung des Datenstrukturwerkzeuges *BIMQ* und der BIM-Autorensoftware *Allplan* konnte der optimierte Prozess nicht vollständig umgesetzt werden. In *BIMQ* kann die Idee des *Konfektionssilos*, in dem *Properties* mit *Psets* verknüpft und lediglich referenziert werden, nicht abgebildet werden. Die Versuche haben gezeigt, dass es in *Allplan* beim Export der IFC-Datei zu Abweichungen bei den Definitionen der *Properties* kommt. Abweichungen sind anders exportierte *Entities*, abweichende *Data Type* bzw. nicht vorhandene Einheiten und Überlagerungen von *Properties* mit von *Allplan* vordefinierten *Attributen*. Die Funktionstüchtigkeit des vollständigen Prozesses ist nur für bestimmte *Entities* gegeben.

6.3 Anmerkungen zur IFC-Spezifikation

Im Abschnitt 5.2.2 wurde erkannt, dass in der IFC-Spezifikation gleichnamige *Properties* existiert, die entweder nach unterschiedlichen Klassen definiert sind oder einen unterschiedlichen *Data Type* aufweisen. Außerdem wurde erkannt, dass in IFC4.3 gleichnamige *Properties* zwar mit identem *Property Type* jedoch unterschiedlichem *Enumeration* existieren. In Tabelle 6.1 ist beispielsweise das *Property EnergySource* mit jeweils unterschiedlichen *Data Type* dargestellt. In allen Fällen wird empfohlen, die *Properties* unterschiedlich zu benennen, sodass nur eindeutige Definitionen der Standard-*Properties* existieren.

Tab. 6.1: Gleichnamige *Properties* mit unterschiedlichem *Enumeration*

<i>Pset</i>	<i>Property</i> - Name	<i>Property Type</i>	<i>Data-Type</i>
Pset_BoilerType- Common	EnergySource	IfcProperty- EnumeratedValue	PEnum_EnergySource
Pset_EngineType- Common	EnergySource	IfcProperty- EnumeratedValue	PEnum_EngineEnergySource

6.4 BIMQ-Verbesserungen

Während dem Bearbeitungszeitraum dieser Arbeit, von Jänner 2022 bis März 2023, war erkennbar, dass *BIMQ* kontinuierlich weiterentwickelt wird und neue Features integriert werden. Vor allem gab es Verbesserungen in der Usability. Beispielsweise wurde durch ein Update ermöglicht, dass bei einem Export von Excel-Dateien, Vorlagen etc. automatisch ein neuer Browser-Tab geöffnet wird. In diesem Fall kann im ursprünglichen Tab im *BIMQ* weitergearbeitet werden und die Ladezeit des Exports von allenfalls mehreren Minuten passiert im Hintergrund. Auch die Ladezeiten der Daten in den einzelnen Menüs wurden stetig geringer.

Bei der Erläuterung des optimierten Prozesses in Kapitel 4.3 wurde die sogenannte *Pset*- und *Property*-Silo sowie der Konfektionssilo erläutert. Dafür wurden *Psets* und *Properties* jeweils getrennt voneinander im Menü *Vorlagen* angelegt. Im optimierten Prozess ist vorgesehen, dass *Psets* und *Properties* über den Konfektionssilo miteinander verknüpft sind und als Paket den *Entities* zugewiesen werden. Im Konfektionssilo werden die Einträge aus dem *Pset*- oder *Property*-Silo lediglich referenziert. Dies würde eine entsprechende Datensicherheit mit sich bringen, da jeder Eintrag nur einmal angelegt wird und in Folge lediglich referenziert wird. Damit wäre im weiteren Sinne ein *Single Source of Truth*-Prinzip umgesetzt. Dieser Schritt kann in *BIMQ* jedoch nicht abgebildet werden. Hier wäre ein zusätzliches Menü erforderlich. Stattdessen werden die *Properties* und *Psets* im Menü *Projektkonfiguration* zusammengeführt. Hier ergibt sich jedoch der Nachteil, dass die *Properties* bei sich wiederholenden *Entities* oder *Psets* wiederholt und einzeln zu verknüpfen sind. Mit dem Konfektionssilo wäre es ausreichend, einzelne *Properties* nur einmal mit deren *Psets* zu verknüpfen. Zusätzlich könnte das gleiche *Psets* unterschiedlichen *Entities* zugewiesen werden. Die *Properties* haben ihren Ursprung weiterhin im zugehörigen *Property*-Silo, wo dessen Eintrag ebenfalls einmalig ist.

Ein Verbesserungsvorschlag des Verfassers ist das Hinzufügen von Kontrollmechanismen. Beispielsweise wäre es hilfreich beim Verwenden der Funktion *Mehrfachzuweisung* ein Feedback zu erhalten, ob sämtliche Zuweisungen korrekt übernommen wurden. Hier wäre ein Hinweis vorteilhaft, der anzeigt, wie vielen *Properties* ein Wert zugewiesen wurde und optional wie viele davor gefiltert wurden und eventuell wie viele insgesamt vorhanden sind. Beim erfolgreichen Import einer Excel-Datei im Menü *Vorlagen* wird bereits ein Hinweis angezeigt, bei dem aufgeschlüsselt wird, wie viele Einträge *unverändert*, *aktualisiert* und *hinzugefügt* wurden.

Beim Import von Excel-Dateien wurden in einzelnen Fällen Fehlermeldungen angezeigt, da die Formatierung nicht dem von *BIMQ* gewünschten Format entspricht. Hier wäre ein Hinweis hilfreich, auf welche Einträge sich die Fehlermeldung bezieht, beispielsweise eine Meldung *Formatierung der Excel-Datei unpassend in Zelle xy*.

Während des Bearbeitungszeitraumes wurden zwischenzeitlich im Menü *Vorlagen* beim Excel-Import die Software- und IFC-*Mapping* nicht korrekt importiert. Diese wurden nach dem Import mit dem Wert - angezeigt. Mit der Version 2.7.9-6 wurde dieser Bug behoben und die *Mappings* werden korrekt importiert.

Im Abschnitt 5.2.2 ist ersichtlich, dass es erforderlich ist die Definition der *Data Type* bzw. Einheiten nach der spezifischen Schreibweise in *BIMQ* zu erstellen. Hier wäre es hilfreich, wenn beim Excel-Import von *Properties* die *Data Type* der IFC-Spezifikation, wie *IfcBoolean*, erkannt werden. Diese Funktionalität wäre im Vergleich zur durchgeführten manuellen Zuweisung weniger fehleranfällig.

Tomczak et al. [28] bemängeln bei Datenstrukturwerkzeugen die nicht einsehbare *validation logic*. Das bedeutet, dass für den Nutzer nicht bekannt ist, wie weit ungültige Eingabe zulässig sind und ob diese erkannt werden. Für *BIMQ* wäre es wünschenswert, dass Einblicke in die Validierungs- und Prüfprozesse gewährt werden. Damit wären für den Nutzer der Umfang und die Grundlagen, die für einen allfälligen Vergleich zur Validierung herangezogen werden, ersichtlich. Dies ist wesentlich für die vom Nutzer an das Datenstrukturwerkzeug zuschreibbare Zuverlässigkeit und den erforderlichen Bedarf an Eigenprüfung.

6.5 Bimplus-Verbesserungen

Der Zwischenschritt des Exports der Vorlage zuerst nach *Bimplus* stellt keinen zeitlichen Mehraufwand dar. Ein Vorteil ist hier, dass es eine zentrale und immer verfügbare Ablage der Vorlage gibt. Diese könnte vom BIM-Management bzw. der BIM-Projektleitung oder BIM-Projektsteuerung verwaltet und verteilt werden.

Es stellt sich jedoch die Frage des Zusammenspiels mit den unterschiedlichen Projektphasen etc. In *Bimplus* kann man lediglich dem gesamten Projekt eine Vorlage zuweisen. Wenn man die Projektphasen und unterschiedliche BIM-Ersteller einbinden und verwalten will, wäre eine Gliederung nach diesen notwendig. Ein erster Schritt könnte sein, dass man nicht nur dem gesamten Projekt eine Vorlage zuweisen kann, sondern auch den einzelnen Modellen in *Bimplus*. Im Folgenden werden einzelne Kritikpunkte zu *Bimplus* genannt:

- Die Bezeichnung der Objekte ist nicht eindeutig. Es ist nicht ersichtlich, welchem Objekt welche *Entity* und Objekt in *Allplan* zugewiesen ist.
- Im *Attributmanager* werden die Einheiten und Werte der *Properties* nicht angezeigt.

6.6 Allplan-Verbesserungen

In der folgenden Auflistung werden Empfehlungen und Verbesserungen für die BIM-Autorensoftware *Allplan*, im Kontext der Inhalte dieser Arbeit, formuliert.

- Die Definitionen und Werte der *Properties* sollten vollständig und korrekt in die IFC-Datei exportiert werden.
- Bei der Modellierung muss ersichtlich und steuerbar sein, welches *Allplan-Objekt* jeweils beim IFC-Export welcher *Entity* zugewiesen wird. *Entity* und *PredefinedType* werden beim Export teilweise automatisch und ohne Einblick festgelegt. Konkret sollten diese Festlegungen bei den vorgesehenen Allplan-Attributen *IFC-Objekttyp* und *IFC-ObjektSubtyp* ersichtlich sein.
- Die Nomenklatur der Objekte in *Allplan* und *Bimplus* sollte einheitlich sein. Außerdem wäre es wünschenswert, dass sämtliche Objekte in beiden Softwareprodukten vorhanden sind und genutzt werden können.
- Um die Einschränkung auf bestimmte *Allplan*-Mappings in *BIMQ* aufzulösen, müssen beliebig definierbare Objekte bzw. Mappings möglich sein. Diese Möglichkeit ist in *Bimplus*

zu schaffen, um in *BIMQ* beliebige *Entity* und *PredefinedType* als Mappings zuweisen zu können. In *Allplan* können diese individuellen Objekte beispielsweise durch das Attribut *IFC-Objektyp* zusammenhängen und gegebenenfalls wiederum mit den *PredefinedTypes* korrelieren.

- Eine zusätzliche Option zum Einspielen einer Vorlage als Datei direkt in *Allplan* und ohne dem Zwischenschritt in *Bimplus* wäre wünschenswert.

6.7 Ausblick

Im Bereich der Informationsanforderungen wird von *bSI* aktuell der Standard *Information Delivery Specification (IDS)* entwickelt. Dieser Standard ist seit 10. März 2023 in der Phase *Out for Review*. *bSI* schreibt in [7]: „*This is the standard to use to define your Level of Information Needs. It brings validation of IFC to the client, the modeler and the Software Tools that perform (automated) analyses.*“ Der Standard ist für Menschen und Maschinen lesbar und wird hierfür in *Extensible Markup Language (XML)* abgebildet. Im Datenstrukturwerkzeug *BIMQ* gibt es bereits, als Entwicklungsstand, die Möglichkeit des Exports der *Projektanforderungen* in Form des *IDS*.

Literaturverzeichnis

- [1] AEC3 Deutschland GmbH. *BIMQ*. URL: <https://www.bimq.de/> (Zugriff am 04. 03. 2023).
- [2] ALLPLAN Deutschland GmbH. *Allplan Paket Vergleich*. URL: <https://www.allplan.com/de/paket-vergleich/> (Zugriff am 12.02. 2023).
- [3] ALLPLAN Österreich GmbH. *Allplan Connect Aktueller Content*. URL: <https://connect.allplan.com/at/content/aktueller-content.html> (Zugriff am 10.03. 2023).
- [4] *BIM-Profil-Server*. URL: <https://www.crb.ch/BIM-Profil-Server.html> (Zugriff am 18.02. 2023).
- [5] *BIMsystems*. URL: <https://bimsystems.de/de/> (Zugriff am 18.02. 2023).
- [6] buildingSMART International Limited. *IFC Certified Software*. IFC Certified Software. URL: <https://www.buildingsmart.org/compliance/redevelopment/certified-software/> (Zugriff am 29.04. 2023).
- [7] buildingSMART International Limited. *Information Delivery Specification IDS*. Information Delivery Specification IDS. URL: <https://technical.buildingsmart.org/projects/information-delivery-specification-ids/> (Zugriff am 07.03. 2023).
- [8] C. C. Eichler, C. Schranz, T. Krischmann und H. Urban. *BIMcert Handbuch Grundlagenwissen openBIM. Ausgabe 2023*. Niederfrohna: Mironde-Verlag, 2023. ISBN: 978-3-96063-052-4.
- [9] C. C. Eichler. *BIM Leistungsbilder für Hoch- und Tiefbau*. Hrsg. von buildingSMART Austria. 20. Sep. 2019. URL: <https://cloud.buildingsmart.co.at/s/25YbmKc6xxX5qdW>.
- [10] C. C. Eichler. *Standard-AuftraggeberInformationsanforderungen für Hoch- und Tiefbau*. 24. Sep. 2019. URL: <https://cloud.buildingsmart.co.at/s/EmBt7dcJWnt8sCr> (Zugriff am 10.05. 2022).
- [11] C. C. Eichler, D. Gaudart, T. Krones und W. Malzer. *LM.BIM 2022 BIM Leistungsbilder*. Ausgabe 2022. Niederfrohna: Mironde-Verlag, 2022. ISBN: 978-3-96063-049-4.
- [12] *IFC 4.3.0.0 (IFC4X3) HTML Documentation*. IFC 4.3.0.0 (IFC4X3) HTML Documentation. Unter Mitarb. von L. van Berlo, D. Moulton und T. Krijnen. URL: <http://ifc43-docs. standards.buildingsmart.org/> (Zugriff am 10.04. 2022).
- [13] *IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, Annex B Alphabetical listings – Properties*. IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, Annex B Alphabetical listings - Properties. Unter Mitarb. von L. van Berlo, D. Moulton und T. Krijnen. URL: <http://ifc43-docs. standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex-b4.html> (Zugriff am 17.01. 2023).
- [14] *IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, Annex B Alphabetical listings – Property sets*. IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, Annex B Alphabetical listings - Property sets. Unter Mitarb. von L. van Berlo, D. Moulton und T. Krijnen. URL: <http://ifc43-docs. standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex-b3.html> (Zugriff am 07.01. 2023).

- [15] *IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, IfcBuiltElement*. IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, IfcBuiltElement. Unter Mitarb. von L. van Berlo, D. Moulton und T. Krijnen. URL: <http://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcBuiltElement.htm> (Zugriff am 07.01.2023).
- [16] *IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, IfcPropertyResource*. IFC 4.3.1.0 (IFC4X3_ADD1) HTML Documentation, IfcPropertyResource. Unter Mitarb. von D. Moulton und buildingSMART International. URL: <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/ifcpropertyresource/content.html> (Zugriff am 17.02.2023).
- [17] ISO 16739-1. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema*. Internationale Organisation für Normung. 2018.
- [18] ISO 19650-1. *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles*. Internationale Organisation für Normung. 2018.
- [19] M. Mellenthin Filardo. „Praxisrelevante Auftraggeber-Informations-Anforderungen für Infrastrukturprojekte als Indikatoren des Einführungsgrades von BIM in Deutschland“. In: *31. Forum Bauinformatik: 11. bis 13. September 2019 in Berlin: Proceedings*. Hrsg. von M. Sternal, L.-C. Ungureanu, L. Böger und C. Bindal-Gutsche. Berlin: Universitätsverlag der TU Berlin, 2019. ISBN: 978-3-7983-3105-1.
- [20] M. Mellenthin Filardo, C. Nürnberger und H.-J. Bargstädt. „XML-basierte Übernahme von Inhalten der Auftraggeber-Informationsanforderungen in eine BIM-Autorensoftware“. In: *Proceedings of 33. Forum Bauinformatik*. Sep. 2022.
- [21] A. Niedermaier und R. Bäck. *ALLPLAN BIM-Kompendium: Theorie und Praxis*. 4. München: ALLPLAN GmbH, Dez. 2018.
- [22] ÖNORM A6241-2:2020-02-01. *Digitale Bauwerksdokumentation Teil 2: Building Information Modeling (iBIM) – Level 3*. Austrian Standards Institute. Wien, 2020.
- [23] ÖNORM EN 17412-1:2021-05-15. *Bauwerksinformationsmodellierung – Informationsbedarftiefe – Teil 1: Konzepte und Grundsätze*. Austrian Standards Institute. Wien, 2021.
- [24] ÖNORM EN ISO 23386:2020-09-01. *Bauwerksinformationsmodellierung und andere digitale Prozesse im Bauwesen — Methodik zur Beschreibung, Erstellung und Pflege von Merkmalen in miteinander verbundenen Datenkatalogen*. Austrian Standards Institute. Wien, 2020.
- [25] C. Schranz, A. Gerger, S. Fischer und H. Urban, Hrsg. *Digitalisierung und Standardisierung der Immobilienwirtschaft unter Anwendung von BIM am Beispiel eines Neubaus: Bio-Institut der HBLFA Raumberg-Gumpenstein*. Endbericht von buildingSMART Austria. Wien: TU Verlag, 2021. ISBN: 978-3-903311-28-2.
- [26] C. Schranz, A. Gerger, S. Fischer und H. Urban, Hrsg. *Digitalisierung und Standardisierung der Immobilienwirtschaft unter Anwendung von BIM am Beispiel eines Neubaus: Bio-Institut der HBLFA Raumberg-Gumpenstein*. Anhang: AIA & BAP von buildingSMART Austria. Wien: TU Verlag, 2021.
- [27] Stadt Wien. *BRISE-VIENNA Gemeinsam planen und leben*. DigitalesWien. URL: <https://digitales.wien.gv.at/projekt/brisevienna/> (Zugriff am 14.05.2023).

-
- [28] A. Tomczak, L. v. Berlo, T. Krijnen, A. Borrmann und M. Bolpagni. „A review of methods to specify information requirements in digital construction projects“. In: *IOP Conference Series: Earth and Environmental Science* 1101.9 (1. Nov. 2022), S. 092024. DOI: 10.1088/1755-1315/1101/9/092024.
- [29] Urban Innovative Actions (UIA). *BRISE-Vienna - Building Regulations Information for Submission Envolvement*. URL: <https://uia-initiative.eu/en/uia-cities/vienna-call14> (Zugriff am 14.05.2023).

Anhang A

Anhang

Tab. A.1: Einheiten-Zuweisung

<i>Data Type</i>	Schreibweise der Einheiten bei der Eingabe in <i>BIMQ</i>
IfcAccelerationMeasure	Beschleunigung.m/s ²
IfcAngularVelocityMeasure	Winkelgeschwindigkeit.radians/s
IfcAreaDensityMeasure	Flächendichte.kg/m ²
IfcAreaMeasure	Fläche.m ²
IfcBoolean	Wahr/Falsch
IfcComplexNumber	Komplexe Zahl
IfcCountMeasure	Anzahl
IfcDate	Datum.Jahr, Monat und Tag (YYYY-MM-DD)
IfcDateTime	Zeitpunkt (Datum und Zeit)
IfcDuration	Kennzeichen
IfcDynamicViscosityMeasure	dynamische Viskosität.Pa s
IfcElectricCapacitanceMeasure	elektrische Kapazität.F
IfcElectricChargeMeasure	elektrische Ladung.C
IfcElectricCurrentMeasure	elektrischer Strom.A
IfcElectricResistanceMeasure	elektrischer Widerstand.V/A
IfcElectricVoltageMeasure	elektrische Spannung.V
IfcEnergyMeasure	Energie.J
IfcForceMeasure	Kraft.N
IfcFrequencyMeasure	Frequenz.Hz
IfcHeatFluxDensityMeasure	Wärmestromdichte.W/m ²
IfcHeatingValueMeasure	Heizwert.MJ/kg
IfcIdentifier	Identifizierungszeichen
IfcIlluminanceMeasure	Beleuchtungsstärke.lx
IfcInductanceMeasure	Induktivität.H
IfcInteger	ganze Zahl
IfcIntegerCountRateMeasure	Anzahl pro Zeit
IfcIonConcentrationMeasure	Ionenkonzentration
IfcIsothermalMoistureCapacityMeasure	isotherme Feldkapazität.m ³ /kg
IfcLabel	Kennzeichen
IfcLengthMeasure	Länge
IfcLinearVelocityMeasure	lineare Geschwindigkeit.m/s
IfcLogical	Logisch
IfcLuminousFluxMeasure	Lichtstrom.lm
IfcMassDensityMeasure	Massendichte.kg/m ³
IfcMassFlowRateMeasure	Massenstrom.kg/s

IfcMassMeasure	Masse
IfcMassPerLengthMeasure	Masse pro Längeneinheit.kg/m
IfcModulusOfElasticityMeasure	Elastizitätsmodul.N/m ²
IfcMoistureDiffusivityMeasure	Feuchtediffusivität.m ³ /s
IfcMolecularWeightMeasure	Molekülmasse.g/mole
IfcMomentOfInertiaMeasure	Trägheitsmoment.m ⁴
IfcNonNegativeLengthMeasure	Länge (nicht-negativ, >=0)
IfcNormalisedRatioMeasure	Verhältnis (normalisiert, 0-1)
IfcNumericMeasure	numerisch
IfcPHMeasure	pH-Wert
IfcPlanarForceMeasure	Flächenkraft.N/m ²
IfcPlaneAngleMeasure	ebener Winkel
IfcPositiveLengthMeasure	Länge (positiv, >0)
IfcPositivePlaneAngleMeasure	ebener Winkel (positiv, >0)
IfcPositiveRatioMeasure	Verhältnis (positiv, >0)
IfcPowerMeasure	Leistung.W
IfcPressureMeasure	Druck
IfcRatioMeasure	Verhältnis
IfcReal	reelle Zahl
IfcRotationalFrequencyMeasure	Drehzahl.cycles/s
IfcSectionModulusMeasure	Widerstandsmoment.m ³
IfcSoundPowerLevelMeasure	Schalleistungspegel.dB
IfcSpecificHeatCapacityMeasure	spezifische Wärmekapazität.J / kg Kelvin
IfcTemperatureRateOfChangeMeasure	Temperaturänderungsrate.K/s
IfcText	Text
IfcThermalConductivityMeasure	Wärmeleitfähigkeit.Watt / m Kelvin
IfcThermalExpansionCoefficientMeasure	Wärmeausdehnungskoeffizient.1/K
IfcThermalResistanceMeasure	Wärmeleitwiderstand.m ² Kelvin/Watt
IfcThermalTransmittanceMeasure	Wärmedurchgangskoeffizient.Watt / m ² Kelvin
IfcThermodynamicTemperatureMeasure	thermodynamische Temperatur.K
IfcTime	Zeit
IfcTimeMeasure	Zeitmaß
IfcTorqueMeasure	Drehmoment.Nm
IfcURIReference	Kennzeichen
IfcVaporPermeabilityMeasure	Dampfdurchlässigkeit.kg / s m Pascal
IfcVolumeMeasure	Volumen.m ³
IfcVolumetricFlowRateMeasure	Volumenstrom.m ³ /s
IfcWarpingConstantMeasure	Wölbwiderstand.m ⁶