Diplomarbeit

Analyse und Erweiterung von Information Delivery Specification zur Verwendung für Prüfregeln

ausgeführt zum Zwecke der Erlangung des akademischen Grads
Diplom-Ingenieur
eingereicht an der TU Wien, Fakultät für Bau- und Umweltingenieurwesen

---

Diploma Thesis

Analysis and extension of Information Delivery Specification for use in checking rules

submitted in satisfaction of the requirements for the degree
Diplom-Ingenieur
of the TU Wien, Faculty of Civil and Environmental Engineering

Patrick Loibl, BSc

Matr.Nr.: 11802580

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

Wien, im Dezember 2023 _____

Kurzfassung

Building Information Modeling (*BIM*) nimmt in der Bauwirtschaft stetig an Bedeutung zu. Die interdisziplinäre Zusammenarbeit zwischen allen Projektbeteiligten an einem gemeinsamen BIM-Modell liefert in allen Phasen der Planung bis zum Betrieb des Bauwerks große Vorteile. In engem Zusammenhang mit der Entwicklung von BIM steht die stets zunehmende Anzahl an alphanumerischer Information in BIM-Modellen. Um mit dieser Information über die ganze Projektdauer umgehen zu können, werden Informationsanforderungen durch die einzelnen Beteiligten erstellt. Großteils erfolgt die Erstellung in textlicher oder tabellarischer Form, welche somit in der Regel nicht maschinenlesbar ist. Mit dem neuen Format *Information Delivery Specification* (*IDS*) von buildingSMART International soll ein offener Standard dafür geschaffen werden. Dieser ist maschinenlesbar und unabhängig von proprietärer Software. Die Prüfung alphanumerischer Information stellt nicht nur einen wichtigen Bestandteil der Planungsphase dar, sondern gewinnt auch mit der zunehmenden Digitalisierung im Baubewilligungsverfahren große Bedeutung. Bis zum jetzigen Zeitpunkt fand großteils proprietäre Software zur Testung auf Einhaltung der Rechtsmaterie Anwendung. Das Ziel dieser Arbeit liegt daher in der Analyse von IDS und der Beantwortung der Fragestellung, ob mit IDS ein offener Standard für Prüfregeln geschaffen werden kann.

In der Arbeit erfolgt eine Untersuchung von IDS anhand verschiedener Anwendungsfälle. Einerseits werden Informationsanforderungen für Modelle definiert, die für die Prüfung der Rechtsmaterie erforderlich sind. Dies wird am Beispiel der Fluchtwegsanalyse gezeigt. Andererseits finden direkt einige Punkte aus der Rechtsmaterie Anwendung. Im Speziellen wurden drei Tabellen der OIB-Richtlinie 2 betrachtet. Das Ergebnis der Untersuchung liefert die Anwendungsgrenzen von IDS im Jahr 2023 sowie Anforderungen an Weiterentwicklungen. Aufbauend darauf erfolgte eine Erweiterung von IDS, um die gewünschten Spezifikationen in den Dateien umsetzen zu können. Die Erweiterungen umfassen im Wesentlichen die Verwendung neuer Relationen des IFC-Schemas sowie die Verschachtelung von bestehenden Komponenten von IDS. Gleichzeitig wurde auch die Open-Source-Prüfsoftware *IfcOpenShell* mit dem enthaltenen *IfcTester* erweitert. Für beide Bearbeitungen erfolgte anhand von zwei erstellten BIM-Modellen eine Testung auf deren Funktionsfähigkeit. Darauf aufbauend wurde der Einsatz von IDS für bereits bestehende Prüfregeln aus dem Forschungsprojekt *BRISE-Vienna* analysiert. Diese Regeln beziehen sich auf die OIB-Richtlinien 2 und 4 und sind mit der Software *Solibri Office* zu verwenden.

Die Ergebnisse der Analyse zeigen derzeit eine beschränkte Eignung von IDS als neuen Prüfregel-Standard. Mit dem aktuellen Stand von IDS sind circa 30 % der untersuchten Prüfregeln im gleichen Ausmaß umsetzbar. Die Erweiterungen bringen jedoch großes Potential mit sich und ermöglichen die Umsetzung von insgesamt 60 % der Prüfregeln. Es hat sich gezeigt, dass vor allem bei der Verwendung von Relationen noch Erweiterungspotential liegt und sich dieses durch akzeptablen Programmieraufwand nutzen lässt. Im Jahr 2023 sind mit IDS großteils nur einfache Prüfungen möglich, aber durch verschiedene Erweiterungen kann IDS in der Zukunft effektiv für die Prüfung von Rechtsmaterie eingesetzt werden. Eine Abgrenzung zwischen alphanumerischer und geometrischer Information muss jedoch stets beachtet werden.

Abstract

Building Information Modeling (*BIM*) is becoming increasingly important in the construction industry. Interdisciplinary cooperation between all project participants delivers great benefits in all phases of planning through to the operation of the building. The increasing amount of alphanumerical information in BIM models is related to the development of BIM. To be able to deal with this information over the entire duration of the project, information requirements are created by the project participants. Most of the information is created in textual or tabular form, which is therefore not machine-readable. The new format *Information Delivery Specification* (*IDS*) from buildingSMART International aims to create an open standard for this. This new standard is machine-readable and independent of proprietary software. The checking of alphanumeric information is not only an important part of the planning phase, but is also becoming increasingly important with the increasing digitalisation of the building permit process. Up to now, proprietary software has largely been used for legal code checking. The aim of this thesis is to analyse IDS and answer the question, if IDS can be used to create an open standard for checking rules.

The thesis analyses IDS based on different use cases. On the one hand, information requirements are defined, that are necessary for legal code checking. This is shown using the example of escape route analysis. On the other hand, some parts from the legal documents are used directly. In particular, three tables of the OIB guideline 2 were analysed. The outcome of the analysis provides application limits of IDS in the year 2023 as well as requirements for further developments. Based on this, IDS was further developed in order to make the implementation of desired specifications possible. The extensions contain the use of new relations of the IFC schema as well as the nesting of existing components of IDS. At the same time, the open-source checking software *IfcOpenShell* with the included *IfcTester* was extended. Both extensions were tested for functionality using two BIM models. Based on this, existing checking rules from the research project *BRISE-Vienna* were analysed. These rules refer to the OIB guidelines 2 and 4 and are used with the software *Solibri Office*.

The results of the analysis show a limited suitability for the use of IDS as a new checking standard. The current status of IDS only allows approximately 30 % of the tested checking rules to be implemented. The extensions carry great potential and enable to implement a total of 60 % of the tested checking rules. Especially in the use of relations there is still potential for extension and that this can be used by acceptable programming effort. Finally, it should be noted that in 2023 IDS can only be used to do simple checks, but with some extensions IDS can be used for legal code checking in the future. However, a distinction between alphanumeric and geometric information must be observed.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Bauwirtschaft steht vor kontinuierlichen Herausforderungen und ist von ständigem Wandel und Entwicklungen geprägt. Seit einigen Jahren nimmt die Digitalisierung neben vielen anderen Bereichen auch in die Planungsprozesse von Bauwerken stets größer werdenden Einfluss [10]. Dies führt teilweise zu der Ansicht, dass die Digitalisierung als nächste große industrielle Revolution angesehen wird [13]. In der Bauwirtschaft wird dies vor allem durch die stetig ansteigende Verwendung von Building Information Modelling (BIM) sichtbar.

Die Definition von BIM ist weit gestreut und in der Fachliteratur nicht eindeutig definiert. Eine gute Definition liefern Borrmann et al. [4], welche unter Building Information Modeling die Erstellung und Bearbeitung von digitalen Bauwerksmodellen mit dafür geeigneter Software verstehen. Im Unterschied dazu ist das Building Information Model selbst zu sehen, welches das digitale Abbild des jeweiligen Bauwerks ist. Es enthält nicht nur die dreidimensionale Geometrie, sondern auch nicht-physische Objekte wie Räume und zusätzliche alphanumerische Information wie Materialangaben, Kosten oder technische Eigenschaften.

Digitale Bauwerksmodelle werden von verschiedensten Projektbeteiligten über den gesamten Projektzyklus verwendet [4]. Vor allem die Vielfalt der Akteure, darunter Architekt:innen, Ingenieur:innen und Auftragnehmer:innen erfordert einen reibungslosen Informationsaustausch. Die jeweils benötige Information muss vollständig im BIM-Modell enthalten sein und zusätzlich richtig verortet werden [33]. Die Anforderung an die notwendige Information erfolgt in der Regel mittels Informationsanforderungen (engl. *information requirements*, *IR*). Derzeit beschäftigt sich ein europäisches Normungsgremium mit dem Thema Informationsanforderungen und mit der Frage „*wer liefert was, wann, in welcher Qualität und wer hat es zu prüfen*“ [2]. Tomczak et al. [36] haben die aktuell verfügbaren und verwendeten Möglichkeiten untersucht und zusammengefasst (siehe Abb. 1.1).

○ – No ◑ – Partial ● – Yes * – under development	Standardised	Applicability	Fields					Value constraints				Content			Geom.		Metadata		
			Info. type	Data type	Unit of meas.	Description	References	Equality	Range	Enumeration	Patterns	Existence	Documents	Structure	Representation	Detailedness	Purpose	Actors	Process map
Spreadsheet	○	◑	◑	◑	◑	◑	◑	◑	○	○	○	○	○	○	○	○	◑	○	○
PDT*	●	◑	◑	●	●	◑	●	○	◑	●	○	◑	○	○	○	○	◑	○	○
Data Dict.	●	○	●	●	◑	●	●	●	◑	●	◑	○	○	○	○	○	○	○	○
IDS*	●	●	●	●	●	◑	◑	●	●	●	●	●	○	◑	○	○	●	◑	○
mvdXML	●	●	●	●	●	◑	○	●	●	◑	◑	◑	○	●	●	○	○	◑	○
idmXML	●	◑	◑	◑	◑	●							●		◑	◑	●	●	●
LOIN*	●	◑	◑	●	●	◑	◑	○	○	●	○	○	●	◑	◑	●	●	●	○
IFC P.T.	●	◑	◑	●	●	◑	○	○	○	●	○	○	○	○	○	○	○	○	○
LD+SHACL	○	●	●	●	●	◑	●	●	●	●	●	◑	◑	●	●	○	◑	◑	○

Abb. 1.1: Verschiedene Möglichkeiten von Informationsanforderungen [36, S. 8]

Die einfachste Anwendung sind textbasierte Dokumente wie PDF-Dateien. Sie bieten den Vorteil der einfachen Nutzung, vor allem für technische Laien, bilden aber oft wenig Verbindung zu dem BIM-Modell selbst und sind bei weitem nicht standardisiert. Eine etwas fortgeschrittene, aber dennoch mit BIM beziehungsweise dem offenen Datenformat Industry Foundation Classes (*IFC*) nicht zusammenhängende Möglichkeit, ist die Verwendung von Tabellenkalkulationsprogrammen (z. B. Excel). Beide Lösungen besitzen die fehlende Standardisierung als großen Nachteil. Ein zusätzlicher Nachteil bei PDF-Dateien ist die fehlende Maschinenlesbarkeit [43]. Eine genormte Möglichkeit ist die Nutzung von *Information Delivery Manuals* (*IDM*) [17, 18, 22]. Dies bringt wirtschaftliche Prozesse in Verbindung mit der Bauwerksmodellierung. Dabei wird weniger definiert, *was* geliefert werden muss, sondern *wer* etwas *wann* und *für wen* liefern muss. Die Definition der zu beinhaltenden Daten ist über eine normativ verankerte und mit IFC in Verbindung stehende Möglichkeit umsetzbar, der sogenannten *Model View Definition* (*MVD*) [36]. Diese definiert in der Autorensoftware Eingrenzungen des verwendeten IFC-Schemas für den jeweiligen Informationsempfänger. Die Eingrenzungen beziehen sich auf die Inhalte wie erlaubte Elementklassen oder Quantity Sets und Property Sets [9]. Die Funktion von MVD zielt jedoch sehr auf die Software ab und ist im Wesentlichen standardisiert und nicht veränderbar. Daher hat buildingSMART International mit *Information Delivery Specification* (*IDS*) eine geeignete Möglichkeit zur Beschreibung von Anforderungen an alphanumerische Information entwickelt [36].

IDS ist ein relativ neuer Standard und wurde von buildingSMART International aufbauend auf deren technischer Roadmap [8] im Jahre 2023 offiziell eingeführt. Im Unterschied zur MVD, welche sich mit Themen wie der Abbildung der Klassen-Hierarchie oder der Geometrie-Übertragung aus der Autorensoftware beschäftigt, wird IDS größtenteils von Auftraggebern zur Spezifikation von alphanumerischer Information verwendet. van Berlo und Fischer [41] sehen den Anwendungsbereich von IDS zweigeteilt. Erstens gibt es die Möglichkeit, IDS-Dateien als Konfigurationsdateien in die verwendete Autorensoftware einzuspielen und somit die einzuhaltende Informationsstruktur direkt bei Projekterstellung bereitzustellen. Eine häufig verwendete Autorensoftware ist *ArchiCAD*. Für Version 26 dieser Software erarbeiteten Mellenthin Filardo et al. [21] eine mögliche Lösung zum Einspielen von Informationsanforderungen im XML-Format. Zweitens kann mit IDS-Dateien in der Prüfsoftware der Aufbau und Inhalt der alphanumerischen Informationen automatisch geprüft werden. Diese Prüfsoftware lässt sich wiederum in zweierlei Kategorien einteilen. Einerseits gibt es proprietäre Softwarelösungen, die schon am kommerziellen Markt verfügbar sind. Beispielsweise sind *Solibri Office* [34] (kurz: *Solibri*) und *usBIM* [1] hier weit verbreitet. Andererseits gibt es Open-Source-Programme, die genauso zur Verwendung geeignet sind. Dazu zählt der *IfcTester* von *IfcOpenShell* [14]. Schrödter und Mellenthin Filardo [33] vergleichen beide Lösungen und zeigen deren Vor- und Nachteile bei der Testung von IDS auf.

Im Bauwesen sind verschiedenste rechtliche Dokumente stets präsent. Diese reichen von Richtlinien über Normen bis zu den Landes- und Bundesgesetzen. Im Zusammenhang damit steht die vor Baubeginn notwendige Baubewilligung der jeweiligen Behörden. Im Zuge dieser sind Informationsanforderungen zu definieren und die erhaltenen Unterlagen auf die Einhaltung der Anforderungen zu prüfen. Mit dem Aufkommen von BIM findet die Digitalisierung auch in diesen Prozess Einzug und bringt wesentliche Vorteile und Verbesserungen mit sich. Borrmann et al. [4] geben eine allgemeine Einführung in die Thematik. Gleichzeitig ist die digitale Baubewilligung Thema vieler Forschungsprojekte. Hier ist zum Beispiel das Projekt *BRISE-Vienna* [35, 38] zu nennen. Studien zur Einführung der digitalen Baubewilligung gibt es von Ullah et al. [39] anhand der Stadtverwaltung von Tallinn oder von Trebbi et al. [37], welche dafür Paragraphen aus verschiedenen europäischen Bauordnungen untersucht haben. Das Thema findet auch international Beachtung. Ismail et al. [16] führten einen Vergleich verschiedener Herangehensweisen zur Regelerstellung durch. Es können die bestehenden Programme und verschiedene objekt-orientierte

und logische Herangehensweisen unterschieden werden. Insgesamt wird durch die verschiedenen Untersuchungen die Bedeutung einer digitalen Baueinreichung gezeigt.

Diese Diplomarbeit beschäftigt sich mit der Frage, in welchen Grenzen IDS in der Version 0.9.6 zur Prüfung von alphanumerischer Information verwendbar ist und ob IDS als offener Standard zur Erstellung von Prüfregeln geeignet ist. Bei der digitalen Baubewilligung ist es wichtig, nicht auf proprietäre Programmlösungen zu setzten, sondern allen BauwerberInnen die Zugänglichkeit zu ermöglichen. Da IDS ein software-unabhängiger Standard ist und für die Anforderungen geeignet scheint, wird dieser in der vorliegenden Diplomarbeit auf dessen Eignung untersucht. Anhand ausgewählter praktischer Beispiele werden Anforderungen gestellt, die mit IDS umgesetzt werden sollen. Zum Aufbau des notwendigen Grundwissens handelt Kapitel 2 über die Grundlagen der Diplomarbeit. Das Kapitel beschäftigt sich mit XML und IFC, welche beide eine wesentliche technische Grundlage bilden. Neben der Erklärung von IDS werden in weiterer Folge zwei wesentliche Programme erklärt. In Kapitel 3 werden Anwendungsfälle aus der Forschung und der Praxis gezeigt und daraus Anforderungen an IDS abgeleitet. Kapitel 4 bildet den Kern der Arbeit. Aufbauend auf den Anforderungen aus dem vorhergehenden Kapitel wird in diesem Kapitel IDS um einige Funktionen erweitert. Zu Beginn wird die Struktur von IDS ergänzt. Danach wird die Funktionalität von *IfcOpenShell* erweitert, damit eine Verarbeitung der neuen IDS-Struktur möglich ist. Die Erweiterungen werden in weiterer Folge in Kapitel 5 genutzt, um IDS-Dateien zu erstellen und zu testen. Kapitel 6 beschreibt die Ergebnisse der Untersuchung und analysiert die Verwendung von IDS als Format für Prüfregeln aus dem Forschungsprojekt *BRISE-Vienna*. Das Kapitel 7 fasst die Ergebnisse zusammen und gibt eine Handlungsempfehlung zur Verwendung von IDS.

Kapitel 2

Grundlagen

2.1 XML und XSD

Extensible Markup Language (kurz *XML*) bildet die technische Basis für IDS. XML ist eine Sprache zur strukturierten Darstellung und zum universellen Austausch von Daten. Als Merkmale nennt Becher [3] die hierarchische Struktur, die sehr komplex, aber auch relativ einfach sein kann, und die Lesbarkeit durch Menschen und Maschinen. Somit ist XML zur Verwendung für IDS-Dateien gut geeignet.

Becher [3] beschreibt in ihrem Lehrbuch neben der Anwendung auch die Struktur von XML-Dokumenten. Im Regelfall beginnt jedes XML-Dokument mit dem Prolog, in dem einige wichtige Deklarationen und Metadaten verortet sind. Bei der Verwendung von IDS-Dateien sollte dieser Block aus einer Vorlage entnommen und nicht verändert werden. Nach dem Prolog kommt der eigentliche Inhalt der Datei, der innerhalb des Wurzelelements (hier `<ids>`) zu finden ist. Kommentare sind innerhalb sowie außerhalb des Wurzelelements mit der Zeichenkette `<!--` zu beginnen und mit `-->` zu beenden.

Einzelne Elemente sind die kleinsten Bausteine, aus denen XML-Dokumente aufgebaut werden. Sie besitzen jeweils einen Namen, einen Start- und Ende-Tag und einen optionalen Inhalt. Abb. 2.1 zeigt exemplarisch den Aufbau anhand des Tags `namen` mit dem Inhalt Maier.

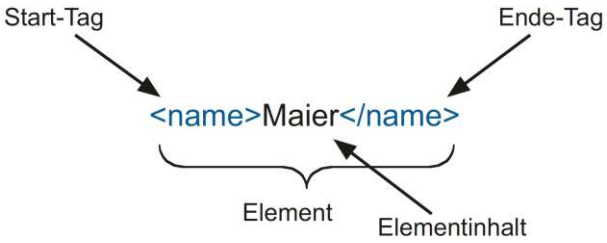

Abb. 2.1: Aufbau eines einfachen Elements [3, S. 11]

Elemente lassen sich in einfache Elemente, welche nur eine Zeichenkette enthalten, und in strukturierte Elemente unterteilen. Zweitere beinhalten eine Verschachtelung von einfachen oder strukturierten Elementen. Zur besseren Unterscheidung spricht man dabei auch von Eltern- sowie Kindelementen. Ein Beispiel dazu ist in Abb. 2.2 zu sehen, wo das strukturierte Elternelement `dozent` die beiden einfachen Kindelemente `name` und `vorname` enthält.

```
<dozent>
  <name>Maier</name>
  <vorname>Fritz</vorname>
</dozent>◀
```

Abb. 2.2: Beispiel eines strukturierten Elements [3, S. 12]

Elemente können neben ihrem Inhalt auch eine beliebige Anzahl von Attributen erhalten. Diese sind im jeweiligen Start-Tag anzugeben (siehe Abb. 2.3). Es können mehrere Attribute in beliebiger Reihenfolge durch Leerzeichen getrennt angegeben werden. Anwendung finden jene vor allem bei der Angabe von Zusatzinformation und Metadaten.

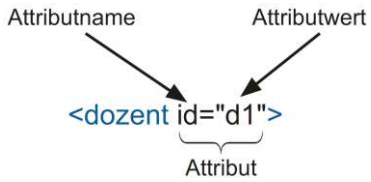

Abb. 2.3: Aufbau eines Start-Tags mit einem Attribut [3, S. 13]

Bei der Verwendung von XML ist darauf zu achten, dass nur sogenannte wohlgeformte Dokumente benutzt werden. Ausschließlich diese entsprechen den notwendigen Vorgaben und sind daher verwendbar. Einige wichtige Regeln für wohlgeformte Dokumente können ebenfalls im Lehrbuch von Becher [3] gefunden werden:

1. Jeder verwendete Start-Tag muss durch den richtigen Ende-Tag geschlossen werden, auch wenn das Element selbst leer ist.
2. Bei XML ist die Groß- und Kleinschreibung zu beachten.
3. Verschachtelung von Elementen ist erlaubt, eine Überlappung von Elementen jedoch nicht.
4. Attribute müssen mit Gleichheitszeichen und doppelten oder einfachen Anführungszeichen eindeutig definiert sein. In einem Element sind zwei gleichnamige Attribute nicht erlaubt.

Wohlgeformte XML-Dokumente entsprechen nur der allgemein vorgegebenen Syntax, jedoch nicht den eigenen Anforderungen des Anwenders zur gewünschten Struktur. Neben der *Document Type Definition* (kurz *DTD*), welche eine alte Möglichkeit zur Strukturdefinition bietet, gibt es als neue Möglichkeit die *XML Schema Definition Language* (kurz *XSD*). buildingSMART International verwendet XSD zur detaillierten Definition der Struktur von IDS. Die jeweils aktuelle XSD-Datei (derzeit Version 0.9.6) ist auf der GitHub-Seite von buildingSMART International [5] in dem Ordner *Development* zu finden. Die Datei kann inklusive der Erweiterungen aus Abschnitt 4.1 Anhang A entnommen werden.

Eine XSD-Datei selbst wird als ein XML-Dokument verfasst und besitzt als Wurzelelement das Element `schema`, wie Becher [3] beschreibt. In diesem Wurzelelement muss als Attribut der Namensraum angegeben werden, welcher alle Elemente, Attribute und vordefinierten Datentypen enthält. Die Definition des Namensraums führt zu einer Festlegung auf ein Namensraum-Präfix (meistens `xs`), welches bei allen Elementen verwendet wird. In weiterer Folge wird nun auf die Deklaration von denjenigen einfachen und komplexen Elementtypen näher eingegangen, welche in der XSD-Datei für IDS von buildingSMART verwendet werden. Die Deklaration von einfachen Elementen, die keine Kindelemente oder Attribute enthalten, erfolgt durch `<xs:element>`. Dem Tag können Attribute hinzugefügt werden (siehe Abb. 2.4), wobei `name` das einzige obligatorische davon ist. Eine Angabe eines Datentyps wird häufig durchgeführt.

```
<xs:element name="Elementname" type="Datentyp"/>
```

Abb. 2.4: Start-Tag mit zwei Attributen [3, S. 94]

Im Gegensatz dazu stehen die strukturierten Elemente, die andere Elemente oder Attribute enthalten können. Die Deklaration jener ist auf zwei Arten möglich. Einerseits kann innerhalb eines Elements `<xs:element>` der komplexe Typ `<xs:complexType>` gestartet werden. Innerhalb des komplexen Typs wird in unserem Fall in der Regel der Kompositor `<xs:sequence>` aufgerufen. Dieser gibt an, dass die Kindelemente eine vorgegebene Reihenfolge einhalten müssen. Zusätzlich dazu kann auch die Häufigkeit der Kindelemente festgelegt werden. Durch die Attribute `minOccurs` und `maxOccurs` erfolgt eine Angabe der Grenzen. Eine Deklaration eines strukturierten Elements mit enthaltenem komplexen Typ ist in Abb. 2.5 zu sehen.

```
<xs:element name="dozent">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name"    type="xs:string"/>
      <xs:element name="vorname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element> ◀
```

Abb. 2.5: Deklaration eines komplexen Elementtyps [3, S. 98]

Andererseits kann in einer XML Schema Definition (XSD) ein komplexer Datentyp durch `<xs:complexType>` zentral definiert werden. Dies wird mit dem Element `<xs:complexType name="Datentyp">` erreicht. Durch diese zentrale Definition können Elemente, die in verschiedenen Teilen des Schemas wiederholt genutzt werden, effizient und einheitlich festgelegt werden. Dies erleichtert die Konsistenz und Wartung des Schemas, da Änderungen an der Definition des Datentyps an einer zentralen Stelle vorgenommen werden können und sich automatisch auf alle Verwendungen dieses Typs auswirken.

2.2 IFC

Der IFC-Standard bildet eine wesentliche Basis von openBIM und wird von Eichler et al. [9] ausführlich beschrieben. Die Veröffentlichung der ersten Version des Standards erfolgte bereits im Jahre 1995. Nach vielen Zwischenschritten ist die aktuelle genormte Version IFC4 gemäß ISO 16739 [30] sowie die neueste veröffentliche Version IFC4.3. Für diese Arbeit dient die Version IFC4 als Grundlage. Mit dieser werden alle wesentlichen Gewerke des Hochbaus abgedeckt. Eine Erweiterung auf den Infrastrukturbau ist für die kommende Version IFC5 angedacht. Weitere mögliche Änderungen können bei van Berlo et al. [42] nachgelesen werden. Von den vielen Funktionalitäten, die das IFC-Schema besitzt, werden in diesem Abschnitt nur die für diese Arbeit wichtigen Punkte genannt.

Der wesentliche Vorteil von IFC liegt in der leichten Zusammenarbeit über verschiedene Softwareprogramme hinweg. Grundlage dafür bildet der offene Austausch der Dateien. Von Eichler et al. [9] wird das dafür meistens verwendete Format SPF (STEP Physical File) erläutert. Zusätzlich gibt es auch eine komprimierte Version (ifczip) sowie ein nutzbares XML-Format. Ein Beispiel einer IFC-Datei im SPF-Format ist im linken Teil der Abb. 2.6 zu sehen. Im Gegensatz dazu ist rechts eine native *Revit*-Datei zu sehen. Dies veranschaulicht den wesentlichen Unterschied und zeigt, dass IFC-Dateien im Texteditor geöffnet vom Menschen gelesen werden können. Dies ist für diese Arbeit notwendig, da die Ergebnisausgabe von *IfcOpenShell* den jeweiligen Zeilen der Elemente aus der IFC-Datei im SPF-Format entspricht.

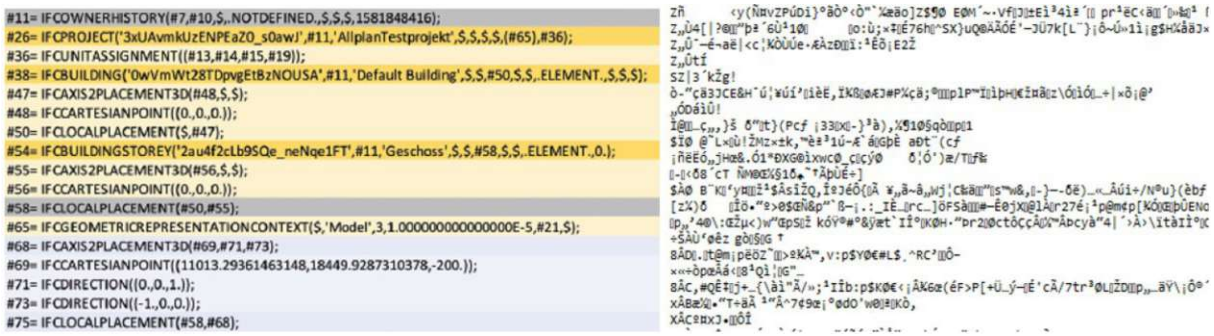

Abb. 2.6: STEP Physical File (links) und Revit-Datei (rechts) im Texteditor geöffnet [40, S. 3]

Neben geometrischen Informationen beinhaltet BIM vor allem auch alphanumerische Informationen. Deren Prüfung ist eine wesentliche Aufgabe von IDS. Daher erfolgt nun auf Basis der Beschreibungen von Eichler et al. [9] eine Begriffsdefinition der wichtigsten Bestandteile:

- **Objekt**: Ein Objekt ist ein Gegenstand, der entweder real greifbar oder begrifflich definierbar sein kann. Im IFC-Schema ist ein Objekt als eine Instanz der jeweiligen definierenden Klasse zu sehen.
- **Attribut**: Attribute sind Informationseinheiten, mit denen grundlegende Eigenschaften von Objekten angegeben werden.
- **Quantity** und **Quantity Set**: Quantities sind Kennzahlen von physikalischen Eigenschaften eines Objekts. Beispiele dafür sind Abmessungen eines Raums oder das Volumen eines Bauteils. Mehrere Quantities werden in einem Quantity Set zusammengefasst.
- **Property** und **Property Set**: Properties sind Informationseinheiten, die zusätzliche beschreibende Informationen und Merkmale von Objekten beinhalten. Beispielsweise sind dies Definitionen zum Feuerwiderstand oder zur Brennbarkeit. Properties werden in Property Sets zusammengefasst. Diese können durch buildingSMART vorgegeben oder selbst erstellt werden. Auch Properties können bei Bedarf selbst angelegt werden.

2.3 IDS

Dieser Abschnitt beschäftigt sich mit den technischen Grundlagen und Eigenschaften von IDS. Es erfolgt eine nähere Erklärung des allgemeinen Aufbaus sowie der verwendbaren Bestandteile (sogenannte *Facets*). van Berlo und Fischer [41] beschreiben die technischen Eigenschaften des neuen Standards ausführlich. Weiters können alle Informationen auch auf der zugehörigen GitHub-Seite von buildingSMART International [5] abgerufen werden. Als Basis für das Dateiformat IDS wurde von den Entwickler:innen die Sprache XML gewählt [33, 36]. IDS kann theoretisch mit jeder Art von Daten verwendet werden, funktioniert jedoch mit dem IFC-Schema am besten. Diese Tatsache wird in weiterer Folge anhand der erklärten Facets deutlich sichtbar.

Der Aufbau der Dateien ist durch die veröffentlichte XSD-Datei vorgegeben. Miyauchi [23] bietet einen kurzen Überblick sowie eine Einführung dazu. IDS-Dateien bestehen zu Beginn aus einem Block an Metadaten. Danach folgt der eigentliche Inhalt in Form von aneinandergereihten Spezifikationen (*Specifications*). Diese setzten sich wiederum aus Metadaten sowie dem Anwendungsbereich (*Applicability*) und den Anforderungen (*Requirements*) zusammen [9, 19, 33, 36]. Die Metadaten sind jeweils als XML-Attribute in den Start-Tags der Spezifikation enthalten. Zur

besseren Erkennbarkeit sind in Abb. 2.7 der Anwendungsbereich blau und die Anforderungen orange eingefärbt.

Im ersten Bereich werden diejenigen Elemente definiert, auf die die aktuelle Spezifikation anzuwenden ist. Die Definition erfolgt durch die Benennung einer notwendigen IFC-Klasse. Sollte das Element noch weiter spezifiziert werden, ist zum Beispiel zusätzlich die Angabe von zu erfüllenden Properties oder Attributen möglich. Der zweite Bereich regelt die Anforderungen an die Elemente und beinhaltet somit die eigentlichen Informationsanforderungen. Hier werden in der Regel erforderliche Properties oder Property Values angegeben. Das Beispiel aus Abb. 2.7 fordert von allen Wänden (`IFCWALL`) in dem Modell, dass sie das Property `LoadBearing` mit dem Datentyp `IfcBoolean` im Property Set `Pset_WallCommon` besitzen.

```
<specification name="IfcWall General" ifcVersion="IFC4">
<applicability>
    <entity>
      <name>
        <simpleValue>IFCWALL</simpleValue>
      </name>
    </entity>
  </applicability>
  <requirements>
    <property measure="IfcBoolean">
      <propertySet>
        <simpleValue>Pset_WallCommon</simpleValue>
      </propertySet>
      <name>
        <simpleValue>LoadBearing</simpleValue>
      </name>
    </property>
    <!-- further properties -->
  </requirements>
</specification>
```

Abb. 2.7: Beispielhafte Spezifikation für Wände aus einer IDS-Datei [41, S. 106]

Für beide Bestandteile einer Spezifikation finden sogenannte Facets Verwendung [33, 41]. Von diesen existieren insgesamt sechs Stück. Sie dienen der konkreteren Beschreibung der Information von IFC-Elementen, wie zum Beispiel der Entität, der Properties oder der Attribute. Die einzelnen Facets sind:

Entity Facet: Dieses Facet gibt die Klassen des IFC-Schemas an, die in der IDS-Datei beziehungsweise in der jeweiligen Spezifikation zu beachten sind. Die Verwendung dieses Facets ist optional, jedoch muss bei der Nutzung exakt eine Entität angegeben werden.

Attribute Facet: Das zweite Facet dient zur Angabe der Attribute und berücksichtigt die standardmäßig in IFC vorhandenen. Der Name des Attributs ist verpflichtend anzugeben. Ein entsprechender Wert kann optional hinzugefügt werden. Ist das der Fall, muss das Element den entsprechenden Wert besitzen, ansonsten muss nur das Attribut befüllt sein.

Classification Facet: Mit diesem Facet können zusätzliche eigene Klassifikationen abseits der IFC-Klassifikation genutzt werden.

Property Facet: Dieses Facet funktioniert ähnlich dem Attribute Facet. Es kann entweder nur ein Property mit dem zugehörigen Property Set angegeben oder auch ein Property Value

spezifiziert werden. Zusätzlich muss in den Attributen der Datentyp angeführt werden. Dieses Facet kann ebenfalls bei Quantities und Quantity Sets Anwendung finden.

Material Facet: Das fünfte verfügbare Facet dient zur Definition eines Materials, aus dem das Element bestehen muss. Es kann entweder direkt ein Material genannt werden, welches vorhanden sein muss, oder nur das Facet angelegt werden. Dann kann das Element ein beliebiges Material besitzen, darf jedoch nicht undefiniert sein.

PartOf Facet: Mit diesem Facet können, aufbauend auf den verschiedenen Relationen, die es im IFC-Schema gibt, Beziehungen zwischen Elementen vorgegeben werden. Dieses Facet wird in der vorliegenden Arbeit am häufigsten verwendet und wird auch erweitert.

In den beschriebenen Facets können zum Beispiel für Property oder Attribute Values nicht nur einzelne konkrete Werte vorgegebenen werden, sondern es sind auch bestimmte Einschränkungen möglich. Unter anderem kann eine Liste von erlaubten Werten, die minimale und maximale Länge eines Werts sowie dessen Muster, in Form einer Reihenfolge von erlaubten Zeichen, definiert werden. Die verschiedenen Möglichkeiten sind bei van Berlo und Fischer [41] nachzulesen beziehungsweise erfolgt in spezifischeren Beispielen in dieser Arbeit eine Erklärung.

In Abb. 2.8 ist ein weiteres Beispiel aus einer IDS-Datei angeführt. Diese Spezifikation ist umfangreicher als in Abb. 2.7 und farblich wieder in zwei Blöcke geteilt. Angewendet wird diese ebenfalls auf Wände (`IFCWALL`), die jedoch noch näher definiert werden. In diesem Fall sollen nämlich nur tragende Wände Beachtung finden. Dies wird durch das Property `LoadBearing` ausgedrückt, welches den Wert `true` besitzen muss. Als Anforderung an die tragenden Wände wird das Property `FireRating` gestellt. Hier sieht man eine der möglichen Einschränkungen, nämlich, dass das Element einen von fünf verschiedenen Werten besitzen kann, um die Anforderung zu erfüllen.

Die einzelnen Tags in einer IDS-Datei können verschiedene Attribute bekommen. Bei Property-Facets ist das Attribut *datatype* stets zu verwenden und mit einem Datentyp aus dem verwendeten IFC-Schema zu befüllen. Das PartOf-Facet kann das Attribut *relation* besitzen, in welchem eine entsprechende Relation (Subsubklasse von *IfcRelationship* [6]) angegeben werden soll. Im Start-Tag von Spezifikationen sind die Attribute *name* und *ifcVersion* zu finden. Weiters können für Spezifikationen sowie für alle Facets im Abschnitt der Requirements die beiden Attribute *minOccurs* und *maxOccurs* angegeben werden. Durch Verwendung dieser im Start-Tag einer Spezifikation kann die minimale und maximale Anzahl der entsprechenden Objekte in einem BIM-Modell vorgegebenen werden. Die Funktion bei den einzelnen Facets ist ähnlich dazu. In den Requirements kann mit den beiden Attributen vorgeschrieben werden, ob zum Beispiel ein Property vorhanden sein muss (`minOccurs="1"`), sein kann (`minOccurs="0"`) oder nicht vorhanden sein darf (`maxOccurs="0"`). In den meisten Fällen wird für beide Attribute der Wert `1` benutzt.

Es ist anzumerken, dass die beiden Spezifikationen aus den Abb. 2.7 und 2.8 nicht den Vorgaben der aktuellen XSD-Version 0.9.6 vom 20. Juni 2023 entsprechen. Mittlerweile wurde das Attribut `measure` zu `datatype` umbenannt, um dessen Verwendung besser gerecht zu werden. Allgemein ist auf die stetige Weiterentwicklung von IDS bei der Nutzung von verschiedensten Programmen zu achten, um jeweils mit dem aktuellen beziehungsweise funktionalen Stand zu arbeiten.

```
<specification name="IfcWall FireRating for LoadBearing walls" ifcVersion="IFC4">
  <applicability>
    <entity>
      <name>
        <simpleValue>IFCWALL</simpleValue>
      </name>
    </entity>
    <property measure="IfcBoolean">
      <propertySet>
        <simpleValue>Pset_WallCommon</simpleValue>
      </propertySet>
      <name>
        <simpleValue>LoadBearing</simpleValue>
      </name>
      <value>
        <simpleValue>true</simpleValue>
      </value>
    </property>
  </applicability>
  <requirements>
    <property measure="IfcLabel">
      <propertySet>
        <simpleValue>Pset_WallCommon</simpleValue>
      </propertySet>
        <name>
          <simpleValue>FireRating</simpleValue>
        </name>
        <value>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ND"/>
            <xs:enumeration value="REI 30"/>
            <xs:enumeration value="REI 60"/>
            <xs:enumeration value="REI 90"/>
            <xs:enumeration value="REI 120"/>
          </xs:restriction>
        </value>
      </property>
    </requirements>
  </specification>
```

Abb. 2.8: Beispielhafte Spezifikation für tragende Wände aus einer IDS-Datei [41, S. 106–107]

2.4 usBIM

usBIM ist ein cloud-basiertes BIM-Management-System des italienischen Softwareherstellers *ACCA software S.p.A.* [1] und beinhaltet 14 kostenlose sowie viele weitere kostenpflichtige Funktionen. Trebbi et al. [37] untersuchten die Anwendungsmöglichkeiten und Leistungsfähigkeit der Software im Bereich der Kollisionsprüfung sowie der Prüfung von Rechtsmaterie. Ersteres funktioniert sehr gut und ist mit anderen Standardprogrammen vergleichbar, wobei eine Ausgabe als BCF-Kommentare zum damaligen Zeitpunkt noch nicht möglich war. Bei der Prüfung der Rechtsmaterie haben die Autoren mehrere Beispiele aus unterschiedlichen Bauordnungen näher betrachtet. Eine Vielzahl von BIM-Modellen wurde mit der von Trebbi et al. [37] entwickelten App *usBIM.code* auf Einhaltung der ausgewählten Punkte getestet. Die Autoren sehen Vorteile bei

der Verwendung von *usBIM.code* vor allem für Planer:innen, da die BIM-Modelle schon vor dem Bewilligungsverfahren auf Einhaltung der Rechtsmaterie getestet werden können. *usBIM.code* ist jedoch derzeit nicht im kommerziellen Portfolio von ACCA enthalten.

Der Startbildschirm von *usBIM* ist in Abb. 2.9 dargestellt. Darin ist der Bereich der eigenen Dokumente mit dem bereits angelegten Ordner *Diplomarbeit* zu sehen. Eine Erstellung und Ablage von Dateien ist nur in Ordnern und nicht direkt am Startbildschirm möglich. Im unteren Bereich des rechten Rands sind häufig benutzte Funktionen zu sehen. Die drei wichtigsten sind von unten beginnend: *Neues Dokument*, *Datei hochladen* und *Neuer Ordner*. Am rechten oberen Rand unter *Anwendungen* sind weitere Apps zu finden, wie zum Beispiel *usBIM.platform* zum Aufrufen einer gemeinsamen Datenumgebung (CDE).

Für die vorliegende Arbeit sind aus der Vielzahl der Anwendungen nur die beiden Apps *usBIM.IDSeditor* und *usBIM.IDS* von Bedeutung. Erstere ist kostenlos und dient zur Erstellung von IDS-Dateien. Zweitere ist kostenpflichtig und wird zur Kontrolle von IFC-Dateien auf Einhaltung der Anforderungen verwendet. Der Editor *usBIM.IDSeditor* wird entweder bei Erstellung einer neuen IDS-Datei oder durch das Klicken auf eine bestehende geöffnet. Im Gegensatz dazu kann *usBIM.IDS* nicht direkt geöffnet werden, sondern wird ausgehend von einem IFC-Modell aufgerufen.

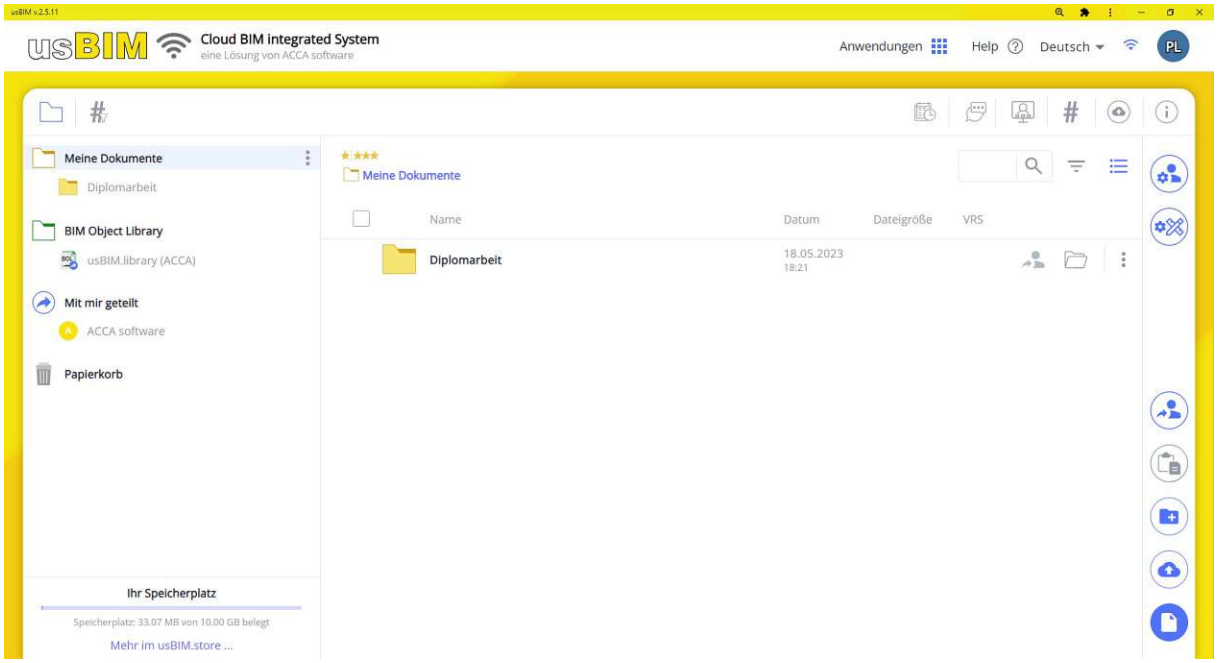

Abb. 2.9: Startbildschirm von usBIM

2.5 IfcOpenShell und IfcTester

IfcOpenShell [14] ist eine Bibliothek von verschiedenen Open-Source-Software-Paketen, die für eine Vielzahl an Anwendungen im Bereich BIM und IFC benutzt werden können. Das gesamte System basiert auf C++ beziehungsweise *Python* und ist in einer beliebigen Entwicklungsumgebung oder als Kommandozeilenwerkzeug auszuführen [33]. In der vorliegenden Arbeit wurde dafür *Visual Studio Code* gewählt. *IfcOpenShell* unterstützt alle aktuellen IFC-Versionen. Es werden Lösungen zum Öffnen und Erstellen von IFC-Dateien, zum Abrufen, Verändern und Prüfen von alphanumerischen (z. B. Entities und Properties) sowie geometrischen Informationen (z. B.

Kollisionsprüfung) angeboten. Weiters können 4D- und 5D-Tools und das buildingSMART Data Dictionary (bSDD) verwendet werden. Zur Überprüfung von IFC-Dateien auf Einhaltung der Anforderungen aus IDS-Dokumenten ist das Paket *IfcTester* vorhanden, welches in dieser Arbeit ausgiebig genutzt wird.

Die Verwendung von *IfcOpenShell* erfolgt durch Download der für das jeweilige Betriebssystem passenden Version. Gleichzeitig muss auch die kompatible *Python*-Version installiert werden. Für diese Arbeit werden *IfcOpenShell* v0.7.0 und *Python* 3.11.0 für Windows 64 Bit benutzt. Um den *IfcTester* verwenden zu können, müssen von der GitHub-Seite [15] unter *src/ifctester/ifctester* noch die dort vorhandenen Dateien heruntergeladen werden. Bei der XSD-Datei ist jedoch darauf zu achten, ob dies die aktuelle Version 0.9.6 (Stand: 2023) ist. Ansonsten muss sie von der GitHub-Seite von buildingSMART International [5] besorgt werden.

Der wesentliche Vorteil des *IfcTesters* von *IfcOpenShell* im Vergleich zum vorhin vorgestellten Programm *usBIM.IDS* liegt in der öffentlichen Verfügbarkeit des Programmcodes. So kann der gesamte Ablauf eines Tests nachvollzogen werden. Zusätzlich sind eine Weiterentwicklung, Verbesserung und darauf aufbauende Anwendung des Systems umsetzbar. Dadurch wird es erst möglich, Erweiterungen der XSD-Datei anzuwenden.

2.6 BRISE-Vienna

BRISE-Vienna [35, 38] ist ein im März 2023 abgeschlossenes Forschungsprojekt der Stadt Wien, in welchem gemeinsam mit Forscher:innen (TU Wien) und BIM-Expert:innen an einem zukunftsfähigen Bewilligungsverfahren gearbeitet wurde. Das zu erreichende Ziel war die teilautomatische Durchführung des Genehmigungsprozesses eines eingereichten openBIM-Modells [20, 40]. Aktuell findet das Bewilligungsverfahren noch mit 2D-Plänen statt, obwohl dies für viele Firmen einen zusätzlichen Aufwand bedeutet. Die Abwicklung vieler Bauprojekte erfolgt jedoch bereits als openBIM-Projekte. Diese basieren auf der Nutzung von offenen Formaten, wie dem IFC-Standard. Somit müssen aus den aufwendig erstellten digitalen Bauwerksmodellen wieder 2D-Pläne generiert werden. Dieser Vorgang produziert zusätzlichen Aufwand und verursacht einen Medienbruch [32]. Durch ein neues Bewilligungsverfahren sollen somit Zeit, Kosten und allgemeine Ressourcen gespart werden [12]. Zu den zwei Kernbereichen des Forschungsprojekts gehören die Entwicklung von openBIM-Prüfroutinen mit den dazu notwendigen Prüfregeln [11, 40] sowie der Einsatz von Augmented Reality [12, 32]. Auf ersteres wird in dieser Arbeit näher eingegangen.

Als wesentliche Grundlage zur Ermöglichung einer digitalen Baubewilligung, wie sie durch *BRISE* angestrebt wird, sehen Krischmann et al. [20] die Verwendung von openBIM. Das Gegenteil zu openBIM ist closedBIM. Dieses nutzt native Datenformate und bietet dadurch teilweise einen einfacheren Datenaustausch auf Projektebene, ist jedoch auf einzelne Programme beschränkt. Um keine Planer:innen zu benachteiligen oder ihnen eine Software aufzuzwingen, kann eine digitale Baubewilligung nur mit openBIM realisiert werden.

In einem openBIM-Bewilligungsverfahren müssen Bauwerber:innen ein Bauantragsmodell aufbauend auf den Anforderungen zum Informationsgehalt, dem *LOIN (Level of Information Need)*, erstellen [32]. Nach Urban et al. [40] sind die zwei Teile dieser Informationsanforderung die Detaillierungsgrade *LOG (Level of Geometry)* und *LOI (Level of Information)*. Abb. 2.10 zeigt ein Bauantragsmodell mit Beispielen zur Unterscheidung des LOG und LOI. Die Nutzung von IDS steht mit zweiterem in Verbindung, da in IDS-Dateien Anforderungen an den alphanumerischen Inhalt von BIM-Modellen gestellt werden [9]. Für beide Detaillierungsgrade ist bei der digitalen Baubewilligung in der Regel die Stufe 300 (LOG300 und LOI300) vorgesehen [11]. In Abb. 2.11 sind beispielsweise die benötigten Merkmale je LOI-Klasse aus den Anforderungen von buildingSMART dargestellt. Um gestellte Anforderungen zu erreichen, wird in der Regel

das bestehende Architekturmodell geometrisch verfeinert und mit den zusätzlich notwendigen alphanumerischen Informationen befüllt [20]. Diese zusätzlichen alphanumerischen Informationen betreffen vor allem bewilligungs-spezifische Merkmale [11].

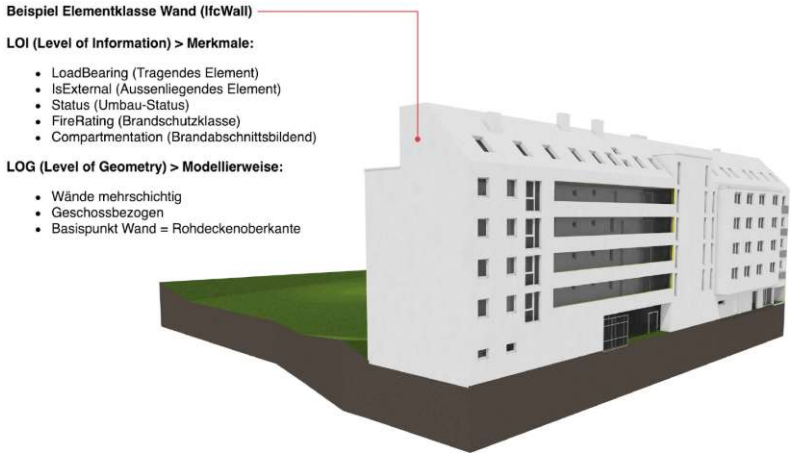

Abb. 2.10: Beispiel der Detaillierungsgrade für die Elementklasse Wand [40, S. 4]

Phasenbezug

LOI-Klasse	MERKMALE ÜBERSETZUNG DE	MERKMAL-NAMEN	EINHEITENTYP	EINHEIT	VERORTUNG	VERANTWORTUNG
LOI100	Aussenbauteil	IsExternal	Wahrheitswert	TRUE/FALSE	Pset_WallCommon	AR
	RaumhoheWand	ExtendToStructure	Wahrheitswert	TRUE/FALSE	Pset_WallCommon	AR
	Status	Status	Text (Optionen-Set[24])	-	Pset_WallCommon	AR
	TragendesElement	LoadBearing	Wahrheitswert	TRUE/FALSE	Pset_WallCommon	AR/TWP
LOI200	BrandabschnittsdefinierendesBauelement	Compartmentation	Wahrheitswert	TRUE/FALSE	Pset_WallCommon	BS
	BrennbaresMaterial	Combustible	Wahrheitswert	TRUE/FALSE	Pset_WallCommon	BS
	Feuerwiderstandsklasse	FireRating	Text (Optionen-Set[24])	-	Pset_WallCommon	BS
	UWert	ThermalTransmittance	Wärmedurchgangskoeffizient	positive Zahl [W/m²K]	Pset_WallCommon	PH
LOI300	Brandverhalten	SurfaceSpreadOfFlame	Text (Beispiel [25])	-	Pset_WallCommon	BS
	Schallschutzklasse	AcousticRating	Text (Beispiel [26])	-	Pset_WallCommon	PH
LOI400	Ausfuehrung	ConstructionMethod	Text (Optionen-Set[24])	-	Pset_ConcreteElementGeneral	AR/TWP
	Betonart	TypeOfConcrete	Text	-	Pset_WallSpecific	AR/TWP
	BewehrungsgradFlaeche	ReinforcementAreaRatio	Bewehrungsgrad	positive Zahl [kg/m²]	Pset_ConcreteElementGeneral	AR/TWP
LOI500	- Noch zu definieren. -					

Abb. 2.11: LOI-Klassen der Elementklasse Wand [9, S. 100]

Im Forschungsprojekt *BRISE-Vienna* kommt die Software *Solibri Office* [34] als Prüfsoftware zum Einsatz. Um damit BIM-Modelle automatisch prüfen zu können, braucht es eine computer-interpretierbare Version der Rechtsmaterie. Die in *BRISE-Vienna* betrachtete Rechtsmaterie besteht aus dem Wiener Garagengesetz und der Wiener Bauordnung für rechtliche Themen sowie den OIB-Richtlinien [31] für bautechnische Themen. Die Umsetzung eines Ausschnitts dieser Rechtsmaterie ist in Fischer et al. [11] erläutert. Allgemein können entweder bereits vorhandene Prüfschablonen verwendet oder mittels der API-Schnittstelle von *Solibri* neue erstellt werden [11, 40]. Dadurch werden die Abdeckung und Prüfung eines größeren Bereichs der Rechtsmaterie möglich. Im Laufe dieser Arbeit wird in Kapitel 6 nochmals auf die sogenannten *BRISE*-Prüfregeln eingegangen.

2.7 OIB-Richtlinien

Das Österreichische Institut für Bautechnik veröffentlicht im Abstand von vier Jahren Richtlinien, die zur Harmonisierung der bautechnischen Vorschriften dienen. Die aktuellen Versionen der Richtlinien stammen aus dem Mai 2023. Alle österreichischen Bundesländer erklären diese im

Allgemeinen durch ihre Bauordnungen bzw. Bautechnikverordnungen für verbindlich. Unterschiede gibt es in der Regel nur beim Datum des Inkrafttretens. Die Richtlinien regeln Anforderungen an viele Bereiche, wie zum Beispiel an die Tragkonstruktion, den Brandschutz oder die Barrierefreiheit [20]. Insgesamt gibt es die OIB-Richtlinien 1 bis 6 [31], wobei für das Thema Brandschutz vier Stück vorhanden sind. Diese besitzen die Nummern 2, 2.1, 2.2 und 2.3. Die OIB-Richtlinie 2 in der aktuellen Version aus 2023 ist teilweise Grundlage dieser Arbeit. Zusätzlich werden in Kapitel 6 alle vier Richtlinien zum Brandschutz sowie die OIB-Richtlinie 4 jeweils in der Version aus 2019 verwendet. Die genauen Titel der verwendeten Richtlinien lauten:

- *OIB-Richtlinie 2: Brandschutz* [24, 25]
  - *OIB-Richtlinie 2.1: Brandschutz bei Betriebsbauten* [26]
  - *OIB-Richtlinie 2.2: Brandschutz bei Garagen, überdachten Stellplätzen und Parkdecks* [27]
  - *OIB-Richtlinie 2.3: Brandschutz bei Gebäuden mit einem Fluchtniveau von mehr als* 22 m [28]
- *OIB-Richtlinie 4: Nutzungssicherheit und Barrierefreiheit* [29]

Kapitel 3

Anwendungsfälle und Anforderungen für IDS

Die Hauptanwendung von IDS ist die maschinenlesbare Definition von alphanumerischer Information von BIM-Modellen. In der Regel werden bestimmte Properties und Attribute vorgeschrieben, die von Elementen unter bestimmten Anwendungsbedingungen einzuhalten sind. Das können Vorgaben bezüglich eines geforderten Detaillierungsgrad der alphanumerischen Information sein, dem sogenannten LOI. Es sind auch Anforderungen aus der Rechtsmaterie denkbar, wie zum Beispiel den OIB-Richtlinien [31]. Dieses Kapitel beschreibt mit dem LOI-Check für die Fluchtwegsanalyse und den Tabellen 1b, 2a und 2b der OIB-Richtlinie 2 einige Beispiele für beide Fälle. Ein spezielles Augenmerk liegt auf den Möglichkeiten und Grenzen des derzeit definierten IDS-Schemas sowie deren notwendigen Erweiterungen, um möglichst viele Anforderungen umsetzen zu können.

3.1 Anwendungsfall LOI-Check für die Fluchtwegsanalyse

Der verwendete LOI stammt aus dem Forschungsprojekt *BRISE-Vienna* und bildet die Grundlage zur automatischen Durchführung einer Fluchtwegsanalyse, wie sie von Fischer et al. [11] entwickelt wurde. Dieser LOI-Check beinhaltet Anforderungen bezüglich der Properties von Räumen und Türen (siehe Tab. 3.1 und 3.2).

Die LOI-Anforderungen an Räume sind relativ einfach gehalten. Einerseits gibt es die beiden Properties *WidmungBehoerde* und *OccupancyNumberPeak*, welche von allen Räumen erfüllt werden müssen, andererseits das Property *ParkingSlot*. Dieses muss nur vorhanden sein, wenn als Widmung *Verkehrserschliessung und -sicherung* eingetragen ist. Als Konsequenz lassen sich daraus zwei Spezifikationen ableiten. Beide bekommen das Attribut `minOccurs="0"`, da es keine Elemente der Entität *IfcSpace* geben muss. Im Gegensatz dazu erhalten die Property-Facets jeweils das Attribut `minOccurs="1"`. Bei Erfüllung der Applicability, also Vorhandensein eines Raums, muss dieser auf jeden Fall die jeweiligen Properties besitzen.

Tab. 3.1: LOI-Anforderungen an die Entität IfcSpace

Property	Property Set	Datatype	Value	Applicability
Widmung Behoerde	Pset_SpaceSpecific	IfcLabel	Liste	alle
ParkingSlot	ZDB_Space_Escape RouteAnalysis	IfcBoolean	T / F	WidmungBehoerde = Verkehrsersch.
Occupancy NumberPeak	Pset_SpaceOccupancy Requirements	IfcCountMeasure	≥ 0	alle

1. **Anforderungen an alle IfcSpace**: Die erste Spezifikation umfasst die beiden für alle Räume verpflichtenden Properties. In der Applicability wird nur ein Entity-Facet mit `IFCSPACE` verwendet. Zwei Property-Facets werden für die Requirements benutzt. Bei beiden findet eine Einschränkung des erlaubten Werts statt. Für das Property `WidmungBehoerde` wird nur eine beschränkte Auswahl an Werten zugelassen, wobei im Codebeispiel aufgrund des Platzbedarfs die Liste auf einige Beispiele gekürzt wurde. Beim zweiten Property `OccupancyNumberPeak` wird eine mathematische Beschränkung vorgenommen. Der vorhandene Wert muss größer oder gleich Null sein. Die Requirements inklusive der beiden Property-Facets werden im Code 3.1 zum Nachlesen angeführt.

2. **Anforderung IfcSpace ParkingSlot**: Das Property `ParkingSlot` muss mit einer eigenen Spezifikation geprüft werden. In der Applicability ist nun neben einem Entity- auch ein Property-Facet vorhanden, welches die Einhaltung der Widmung überprüft. In den Requirements ist kein Property Value vorgegeben, sondern nur das Vorhandensein des Property `ParkingSlot` mit dem Datentyp `IFCBOOLEAN` gefordert.

Code 3.1: Property-Facets aus den Requirements für alle IfcSpace

```
<requirements>
   <property datatype ="IfcLabel" minOccurs="1" maxOccurs="1">
      <propertySet>
         <simpleValue>Pset_SpaceSpecific</simpleValue>
      </propertySet>
      <name>
         <simpleValue>WidmungBehoerde</simpleValue>
      </name>
      <value>
         <xs:restriction base="xs:string">
            <xs:enumeration value="ND"/>
            <xs:enumeration value="Sanitär- und Umkleideräume"/>
            <xs:enumeration value="Wohnen und Aufenthalt"/>
            <xs:enumeration value="Büroarbeit"/>
            <xs:enumeration value="Lagern, Verteilen und Verkaufen"/>
            <xs:enumeration value="Bildung, Unterricht und Kultur"/>
            <xs:enumeration value="Heilen und Pflegen"/>
            <xs:enumeration value="Sonstige Nutzungen"/>
            <xs:enumeration value="Betriebstechnische Anlagen"/>
            <xs:enumeration value="Loggien"/>
         </xs:restriction>
      </value>
   </property>
   <property datatype="IfcCountMeasure" minOccurs="1" maxOccurs="1">
      <propertySet>
         <simpleValue>Pset_SpaceOccupancyRequirements</simpleValue>
      </propertySet>
      <name>
         <simpleValue>OccupancyNumberPeak</simpleValue>
      </name>
      <value>
         <xs:restriction base="xs:integer">
            <xs:minInclusive value="0" />
         </xs:restriction>
      </value>
   </property>
</requirements>
```

Tab. 3.2: LOI-Anforderungen an die Entität IfcDoor

Property	Property Set	Datatype	Value	Applicability
IDDoor	ZDB_Door_ EscapeRouteAnalysis	IfcIdentifier	—	min = 1
IDNextDoor	ZDB_Door_ EscapeRouteAnalysis	IfcIdentifier	—	min = 1
Occupancy NumberFlat	ZDB_Door_ EscapeRouteAnalysis	IfcCountMeasure	> 0	WidmungBehoerde = Wohnung & DoorType = Entry
DoorType	ZDB_Door_ EscapeRouteAnalysis	IfcLabel	—	alle

Die Anforderungen an Türen im BIM-Modell sind etwas umfangreicher. Die vier Properties aus Tab. 3.2 lassen sich in drei Bereiche unterteilen, die auch die einzelnen Spezifikationen bilden:

1. **Anforderungen an alle IfcDoor (DoorType)**: Das Property *DoorType* wird von allen vorhandenen Türen gefordert. Ein genauer Property Value wird nicht angegeben. Die Spezifikation besteht somit aus einem Entity-Facet im Applicability- und einem Property-Facet im Requirements-Abschnitt.

2. **Anforderung an mindestens eine IfcDoor (IDDoor und IDNextDoor)**: Die beiden ersten Einträge der Tab. 3.2 unterscheiden sich von den bisherigen. Die Anforderung sieht vor, dass es mindestens ein Element geben muss, dass diese beiden Properties besitzt. Das führt zu einer Spezifikation mit dem Attribut `minOccurs="1"`. In der Applicability werden ein Entity-Facet sowie zwei Property-Facets für die beiden Properties benutzt. Eine weitere Besonderheit in diesem Fall ist, dass in der IDS-Datei keine Requirements angegeben werden müssen. Es gibt nämlich keine konkreten Anforderungen an die Informationen des Elements, außer dass mindestens ein Element mit den zwei gegebenen Properties vorhanden sein muss. Dies kann mit der Applicability schon ausreichend abgedeckt werden.

3. **Anforderungen IfcDoor als Eingangstür (OccupanyNumberFlat)**: Die letzte Spezifikation befasst sich mit dem Property *OccupancyNumberFlat*. Hier sind zwei Anforderungen bezüglich der Applicability gegeben. Erstens muss die Tür den Typ *Entry* in den Properties hinterlegt haben. Zweites muss sie auch Eingangstür zu einer Wohneinheit sein. Beide Voraussetzungen werden in Abb. 3.1 anhand einer Tür in einer Wand und zwei Räumen graphisch dargestellt. Dies führt zu einer Anforderung an IDS, welche nicht im derzeitigen Standard enthalten ist. Eine Beziehung zwischen zwei Elementen kann zwar mit dem PartOf-Facet hergestellt werden, alle derzeit verwendbaren Relationen lassen jedoch nicht von einer Tür auf einen angrenzenden Raum schließen. Des Weiteren ist es auch nicht möglich, von einem verbundenen Element bestimmte Properties zu fordern. Daher ist diese Spezifikation derzeit nicht umsetzbar.

Mit dem aktuellen Stand von IDS können somit sechs der sieben notwendigen Properties vollständig für den LOI-Check abgedeckt werden. Aufgrund der Länge der einzelnen Spezifikationen sind diese im Anhang B aufgeführt. Für die Verwendung der sechs möglichen Properties können

Standardprogramme Anwendung finden, wie *usBIM.IDSeditor*. Zur Realisierung der letzten Spezifikation mit dem Property *OccupancyNumberFlat* muss ein Texteditor oder eine Programmierumgebung verwendet werden. An dieser Stelle ist anzumerken, dass *usBIM.IDSeditor* nicht sofort nach Veröffentlichung einer neuen XSD-Datei aktualisiert wird. So wurde für circa zwei Monate nach der Veröffentlichung der Version 0.9.6 zum Beispiel der Datentyp der Properties nicht mit dem Attribut `dataytype`, sondern mit dem veralteten `measure` angegeben. Hier ist auf die jeweilige in *usBIM.IDSeditor* verwendete Version zu achten.

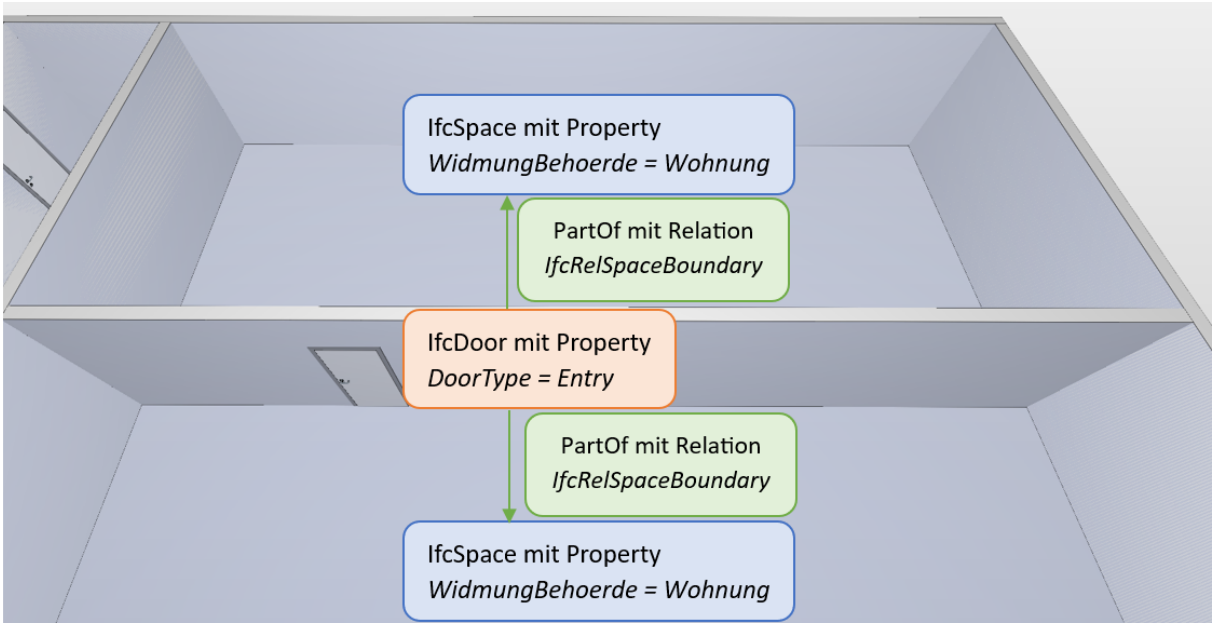

Abb. 3.1: Graphische Darstellung der Anforderungen IfcDoor als Eingangstür

Beispielhaft für alle umsetzbaren Properties wird nun die Erstellung der Spezifikation für das Property *ParkingSlot* mit dem Programm *usBIM.IDSeditor* beschrieben. Dieser wird entweder bei Erstellung einer neuen IDS-Datei oder durch das Klicken auf eine bestehende geöffnet. Im oberen linken Bereich des Startbildschirms sind stets die laufende App sowie die geöffnete Datei sichtbar. Der Editor besitzt fünf Reiter, die in der linken Spalte der Abb. 3.2 ersichtlich sind. Bei Auswahl des ersten Abschnitts können im Hauptteil des Programms *Allgemeine Daten* (Titel, Datum, ...) eingetragen werden. Diese erscheinen am Beginn der IDS-Datei im Block `info`. Der zweite Reiter (*Spezifikationen*) bietet einen Überblick über die schon angelegten Spezifikationen (siehe Abb. 3.3). In diesem Fall sind schon diejenigen für den LOI-Check der Fluchtwegsanalyse zu sehen. Gleichzeitig kann man über diese Ansicht direkt zu den Spezifikationen springen, sie kopieren, löschen oder neue anlegen. Der dritte Reiter (*Reports*) gibt die erstellten Spezifikationen in schriftlicher Form als ein Word-Dokument aus. Im nächsten Reiter (*XML*) kann die XML-Datei eingesehen und heruntergeladen werden. Der letzte Reiter *Audit* bietet als Beta-Funktion eine Überprüfung der Spezifikationen und meldet etwaige Fehler. Ausgehend von dieser Funktion stammt auch der rote Balken in den Abb. 3.2 und 3.3, der auf ein vorhandenes Problem hinweist.

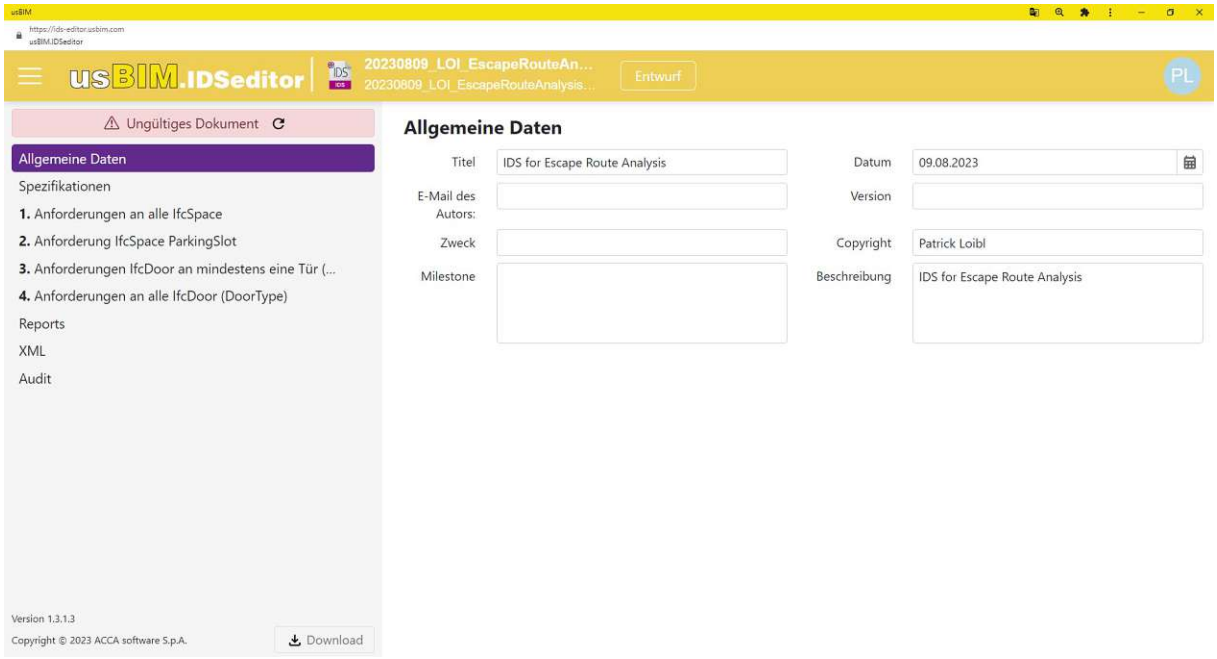

Abb. 3.2: Einstellung der allgemeinen Daten einer IDS-Datei im usBIM.IDSeditor

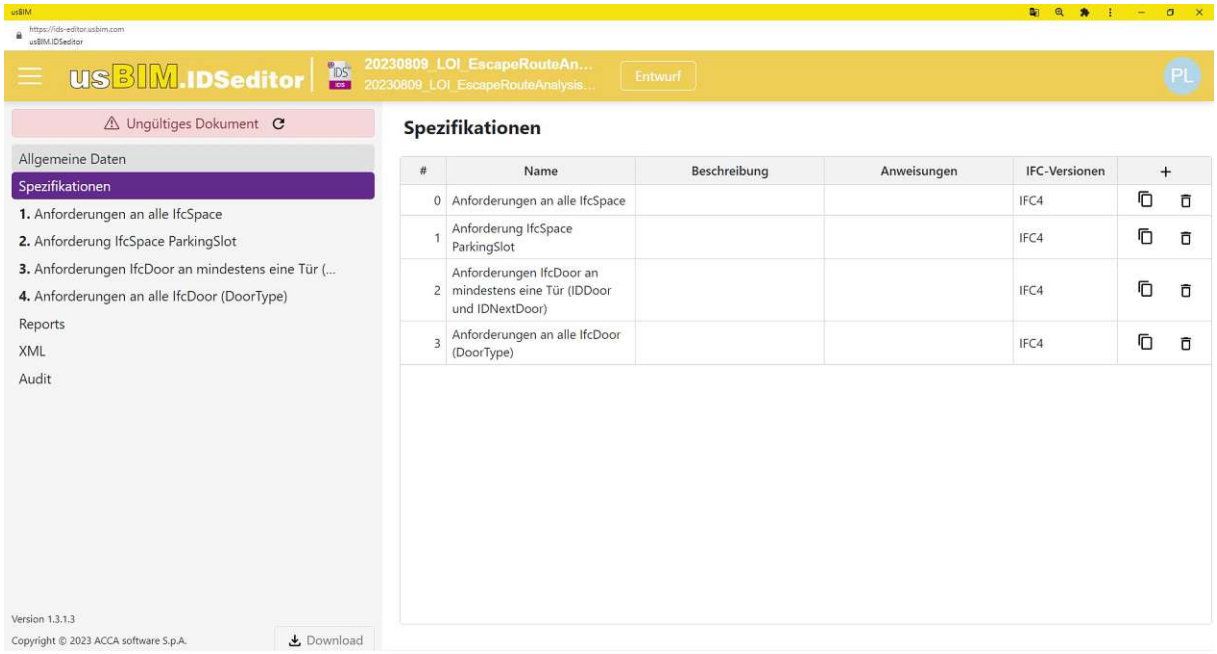

Abb. 3.3: Übersicht der angelegten Spezifikationen im usBIM.IDSeditor

In den Abb. 3.4 und 3.5 ist die Eingabe der Applicability sowie der Requirements für das Property *ParkingSlot* zu sehen. Großteils erfolgt die Auswahl der einzelnen Punkte mittels Dropdown-Menü. Die einzelnen Property- und Property-Set-Bezeichnungen müssen als Freitext eingefügt werden. In Abb. 3.6 können der Name der Spezifikation sowie die IFC-Version eingegeben werden. Zweiteres ist ein Pflichtfeld. Weiters kann im unteren Abschnitt noch die Zusammenfassung der Spezifikation in lesbarem Format betrachtet werden.

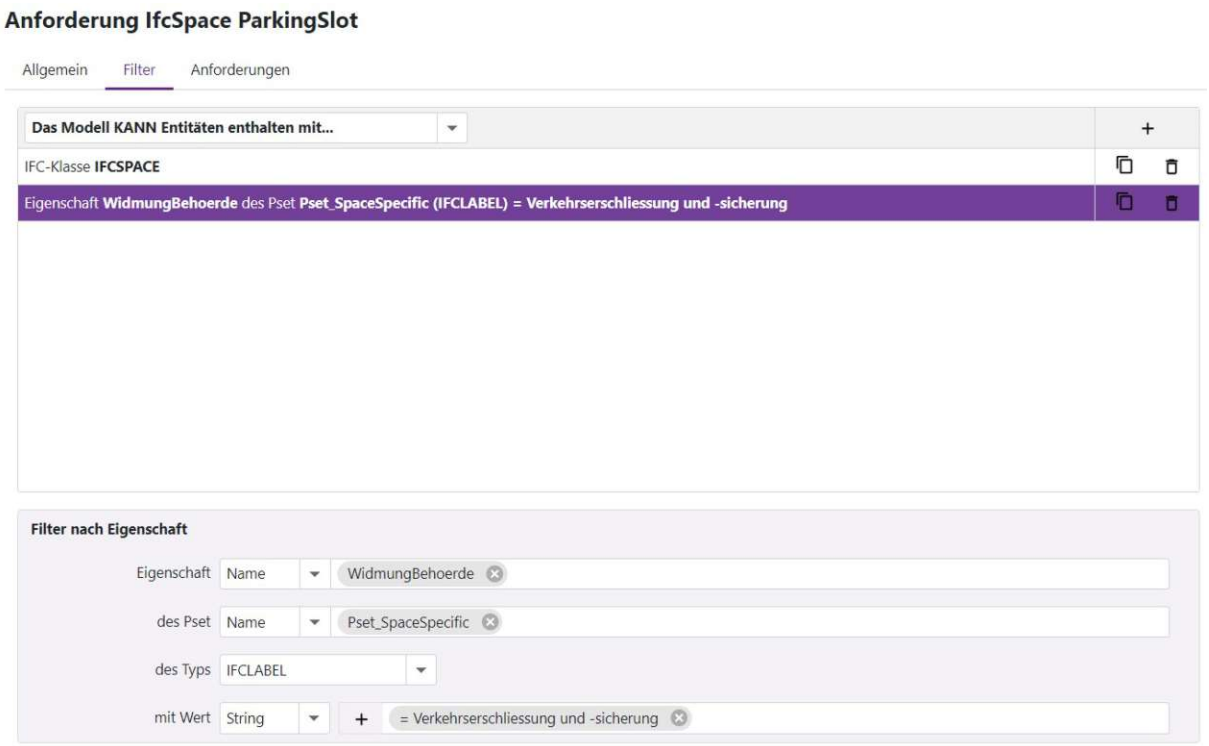

Abb. 3.4: Beispiel der Applicability in usBIM.IDSeditor

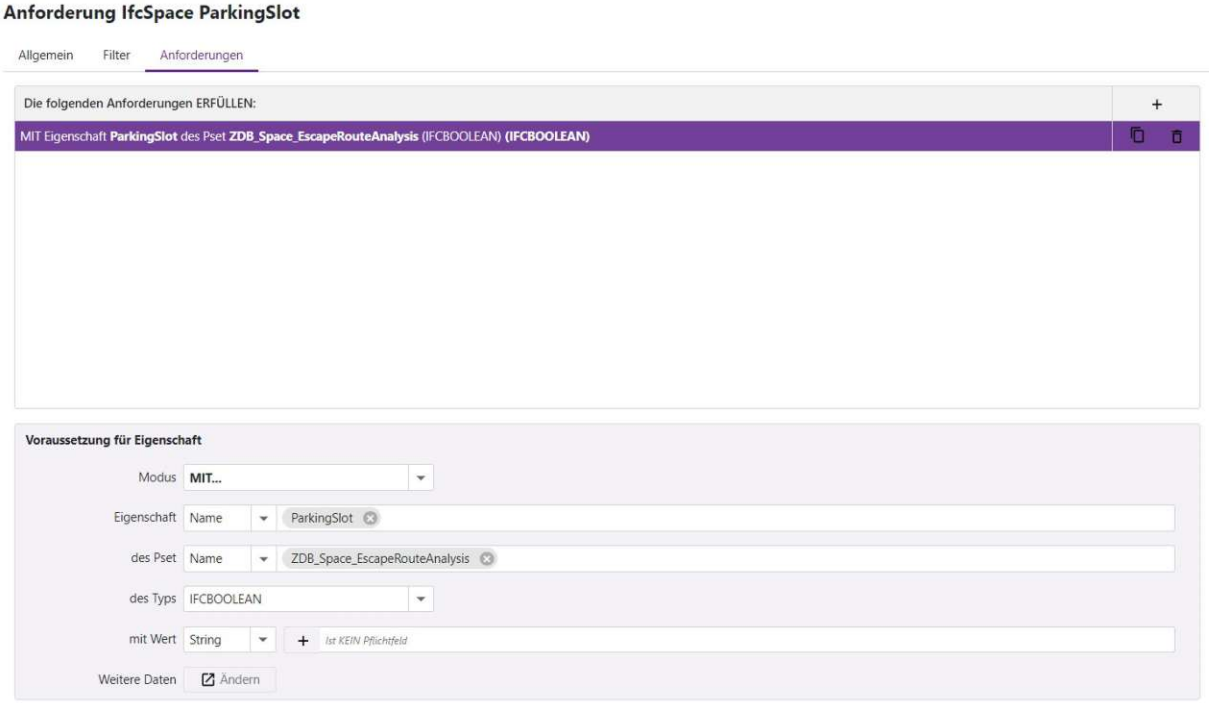

Abb. 3.5: Beispiel der Requirements in usBIM.IDSeditor

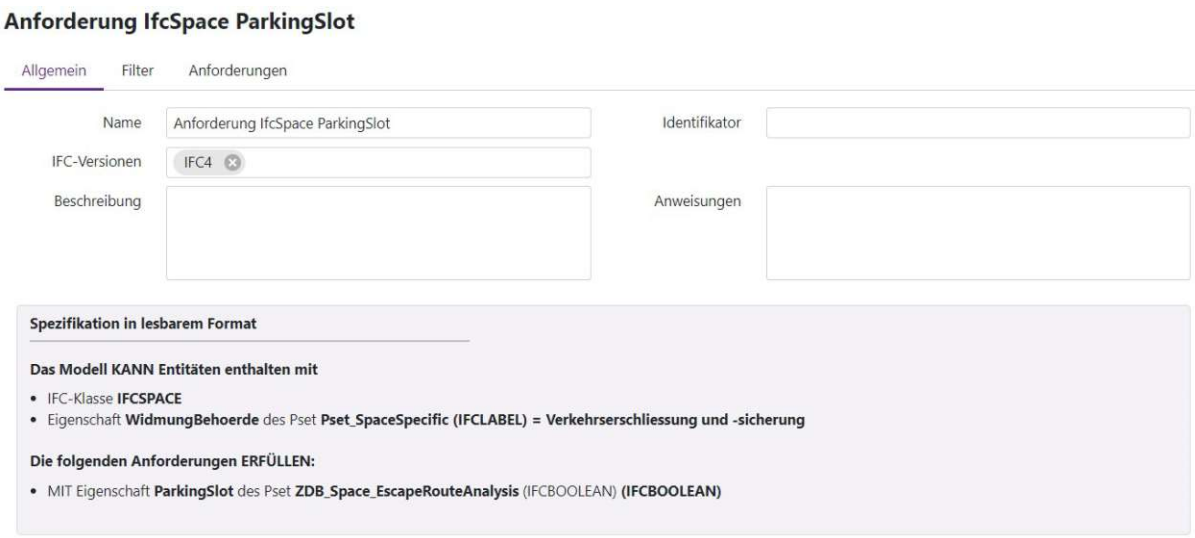

Abb. 3.6: Beispiel einer Spezifikation in usBIM.IDSeditor

Für die Testung der Spezifikationen der sechs bereits möglichen Properties sind ebenso die Standardprogramme geeignet. Das siebte Property *OccupancyNumberFlat* kann erst nach Erweiterung des *IfcTester* von *IfcOpenShell* geprüft werden. Abb. 3.7 zeigt einen Ausschnitt von *usBIM.IDS*. Hierfür wurde die IDS-Datei für den LOI-Check mit den sechs testbaren Properties in den vier erstellten Spezifikationen gewählt. Im linken Bereich der Abbildung ist das BIM-Modell transparent dargestellt. Die fehlerhaften Räume und Türen sind rot eingefärbt. Im rechten Bereich des Screenshots findet man alle vier Spezifikationen aufgelistet. Die danebenstehenden Ziffern geben die Anzahl der Elemente an, welche die Requirements nicht erfüllen (rot), beziehungsweise die Anzahl jener, die die Applicability erfüllen (blau). Durch einen Klick auf den Pfeil kann die Spezifikation ausgeklappt werden. Dabei wird für jedes Element, auf welches die Spezifikation anwendbar ist, eine Zeile sichtbar.

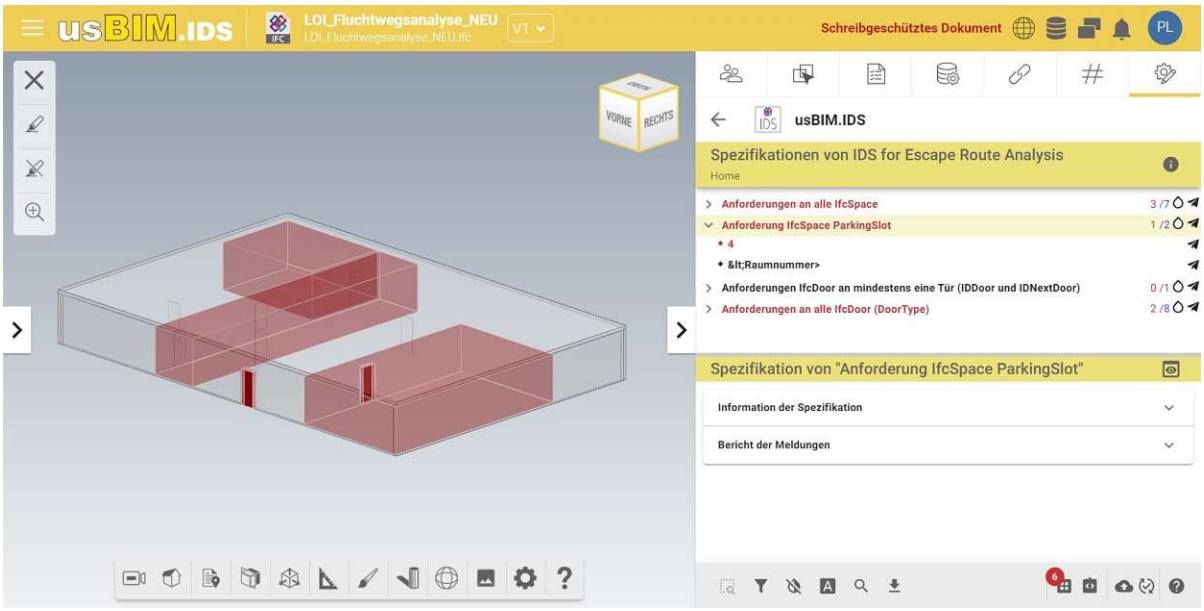

Abb. 3.7: Beispiel eines Ergebnisses in usBIM.IDS

In diesem Fall wird die Spezifikation für das Property *ParkingSlot* angezeigt, welches alle Räume mit der Widmung *Verkehrserschliessung und -sicherung* besitzen müssen. In dem Testmodell sind es zwei Räume mit der entsprechenden Widmung. Der Raum mit der Nummer `4` besitzt das Property *ParkingSlot* nicht und erfüllt somit die Anforderungen nicht. Daher ist dieser rot dargestellt. Zusätzlich kann im rechten unteren Bereich eine Erklärung nachgelesen werden. Der zweite in Frage kommende Raum, der in diesem Beispiel keine Raumnummer besitzt, erfüllt die Spezifikation und ist deshalb nicht hervorgehoben.

3.2 Anwendungsfall Anforderungen an den Feuerwiderstand von Bauteilen

Neben dem LOI-Check wird auch untersucht, wie weit IDS für die Prüfung von BIM-Modellen auf Einhaltung von Rechtsmaterie anwendbar ist. Für die Prüfung von Fluchtwegen selbst ist IDS nicht geeignet, da hierfür die Verwendung von Geometrie und Algorithmen notwendig wäre. Stattdessen kann IDS sehr gut zur Untersuchung von alphanumerischer Information im Brandschutz eingesetzt werden. Daher werden in diesem und in folgendem Abschnitt die Tabellen 1b sowie 2a und 2b der OIB-Richtlinie 2 [25] aus dem Jahr 2023 untersucht. Erstere ist in Tab. 3.3 abgebildet und regelt die Feuerwiderstandsklasse und das Brandverhalten. Die zugehörigen Properties sind *FireRating* und *SurfaceSpreadOfFlame*. Als Ziel wurde definiert, alle Unterpunkte der Punkte 1 bis 4 mit IDS prüfen zu können.

Mit dem derzeitigen Stand von IDS sind im Wesentlichen nur einfache Prüfungen möglich. Eine Prüfung der Feuerwiderstandsklasse von tragenden Bauteilen, mit jeweils einer Spezifikation für Wände, Träger und Stützen, lässt sich zum Beispiel leicht umsetzen. Schwierig wird es bei Abhängigkeiten zu anderen Elementen wie der Erkennung von Trennwänden in Abhängigkeit der angrenzenden Räume oder der Gebäudeklasse, die sich auf das gesamte Gebäude und nicht direkt das Element bezieht. Aufgrund der letzten zwei Gründen ist mit dem Originalstand von IDS keine einzige vollständige Prüfung von Punkten aus der Tabelle möglich. In weiterer Folge wird eine Unterteilung in zwei Kategorien vorgenommen: erstens in die Punkte, deren Umsetzung durch eine Erweiterung erstrebenswert ist, sowie zweitens in jene, die entweder nicht umsetzbar sind oder einen zu großen Aufwand mit zu wenig Nutzen mit sich bringen würden. Folgende Erweiterungen sind möglich:

- **Ermittlung der Gebäudeklasse**
  Die Gebäudeklasse ist in der Regel in einem Property des Gebäudes (IfcBuilding) vorhanden. Die jeweiligen Spezifikationen handeln jedoch von einzelnen Bauteilen. Somit muss zwischen diesen und dem Gebäude eine Verbindung geschaffen werden. Derzeit ist es maximal möglich, von einem Bauteil (z. B. IfcWall) auf das umschließende Geschoß (IfcBuildingStorey) zu kommen. Eine Erweiterung sollte ermöglichen, auch weiter auf das Gebäude zuzugreifen. Wie bereits im vorhergehenden Abschnitt bei dem Property *OccupancyNumberFlat* des LOI-Checks erwähnt, muss hier auch die Möglichkeit gegeben werden, von verbundenen Elementen gewisse Properties zu fordern.

- **Unterscheidung unter- und oberirdisch**
  Eine Unterscheidung zwischen unter- und oberirdischen Bauteilen ist in der Regel nicht direkt möglich, da diese Information in den jeweiligen Bauteilen (z. B. Wände, Decken) nicht existiert. Daher muss eine Verbindung zu den Bauteilen der Verortungsstruktur (z. B. Räume, Geschoße) hergestellt werden, wo die Information zur Höhenlage zu finden ist. Die verbundenen Bauteile sind dann einer Property- beziehungsweise Attributprüfung zu unterziehen.

Tab. 3.3: Ausschnitt der Tabelle 1b der OIB-Richtlinie 2 [25, S. 24]

Gebäudeklassen (GK)		GK 1	GK 2	GK 3	GK 4	GK 5	
						≤ 6 oberirdische Geschoße	> 6 oberirdische Geschoße
1	**tragende Bauteile (ausgenommen Decken und brandabschnittsbildende Wände)**						
1.1	im obersten Geschoß	-	R 30	R 30	R 30	R 60 (5)	R 60
1.2	in sonstigen oberirdischen Geschoßen	R 30 (1)	R 30	R 60	R 60	R 90	R 90 und A2
1.3	in unterirdischen Geschoßen	R 60	R 60	R 90 und A2	R 90 und A2	R 90 und A2	R 90 und A2
2	**Trennwände (ausgenommen Wände von Treppenhäusern)**						
2.1	im obersten Geschoß	-	REI 30 EI 30	REI 30 EI 30	REI 60 EI 60	REI 60 (5) EI 60	REI 60 EI 60
2.2	in oberirdischen Geschoßen	-	REI 30 EI 30	REI 60 EI 60	REI 60 EI 60	REI 90 EI 90	REI 90 und A2 EI 90 und A2
2.3	in unterirdischen Geschoßen	-	REI 60 EI 60	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2
2.4	zwischen Wohnungen bzw. Betriebseinheiten in Reihenhäusern	nicht zutreffend	REI 60 EI 60	nicht zutreffend	REI 60 EI 60	nicht zutreffend	nicht zutreffend
3	**brandabschnittsbildende Wände und Decken**						
3.1	brandabschnittsbildende Wände an der Nachbargrundstücks- bzw. Bauplatzgrenze	REI 60 EI 60	REI 90 (2) EI 90 (2)	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2
3.2	sonstige brandabschnittsbildende Wände oder Decken	nicht zutreffend	REI 90 EI 90	REI 90 EI 90	REI 90 EI 90	REI 90 EI 90	REI 90 und A2 EI 90 und A2
4	**Decken und Dachschrägen mit einer Neigung ≤ 60°**						
4.1	Decken über dem obersten Geschoß	-	R 30	R 30	R 30	R 60	R 60
4.2	Trenndecken über dem obersten Geschoß	-	REI 30	REI 30	REI 60	REI 60	REI 60
4.3	Trenndecken über sonstigen oberirdischen Geschoßen	-	REI 30	REI 60	REI 60	REI 90	REI 90 und A2
4.4	Decken innerhalb von Wohnungen bzw. Betriebseinheiten in oberirdischen Geschoßen	R 30 (1)	R 30	R 30	R 30	R 60	R 90 und A2
4.5	Decken über unterirdischen Geschoßen	R 60	REI 60 (3)	REI 90 und A2	REI 90 und A2	REI 90 und A2	REI 90 und A2

- **Trennwände und Trenndecken**
  Unter Trennwänden versteht man Wände, welche zwischen zwei Wohneinheiten liegen. Eine Unterscheidung von diesen kann durch das für Räume zu vergebende Property *Wohnungsnummer* ermöglicht werden. Dies führt jedoch zu drei Problemen. Erstens muss eine Verbindung zwischen Wand und Raum ermöglicht werden. Zweitens ist es erforderlich, an diese Räume Anforderungen bezüglich des Property *Wohnungsnummer* zu stellen. Drittens muss diese Anforderung nicht nur die Existenz fordern, sondern auch die Prüfung eines Unterschieds in den vorhandenen Werten ermöglichen.

- **Decken innerhalb von Wohnungen**
  In diesem Fall sollen keine Anforderungen an Trenndecken, sondern an Decken innerhalb von Wohnungen gestellt werden. Dabei treten jedoch dieselben Schwierigkeiten wie zuvor auf. Eine Verbindung zwischen Decke und Raum und eine Anforderung an den Raum bezüglich eines Property wären notwendig. Der dritte Punkt unterscheidet sich dahingehend, dass kein Unterschied in den Werten, sondern eine Gleichheit überprüft werden muss.

Neben den vier zu bearbeitenden Punkten gibt es noch einige, die nicht umgesetzt werden können. Diese müssen durch andere Möglichkeiten, meist manuelle Kontrolle, geprüft werden und sind nicht mit IDS abdeckbar. Folgende Punkte sind davon betroffen:

- **Ermittlung der Geschoßanzahl für GK5:** Eine Zählung von Objekten ist nicht möglich. Dafür müsste in IDS die Eingabe einer gewünschten Stückanzahl mit weiteren Infos (max/min) vorhanden sein.
- **Ermittlung des obersten Geschoßes:** Das oberste Geschoß enthält keine dafür verwendbare Information in sich. Eine Ermittlung wäre nur durch Vergleich mit allen anderen Geschoßen möglich.
- **Verwendung der Nachbargrundstücks- bzw. Bauplatzgrenze:** Die Nutzung geometrischer Information in IDS ist nicht möglich.

3.3 Anwendungsfall Anforderungen an den Feuerwiderstand von Treppenhäusern

Als weitere mögliche Anwendungen wurden die Tabellen 2a und 2b aus der OIB-Richtlinie 2 [25] aus dem Jahr 2023 angesehen. Diese regeln ebenso Anforderungen an die Feuerwiderstandsklasse sowie an das Brandverhalten, in diesem Fall jedoch hauptsächlich von Treppenhäusern und Schleusen. Aus der Tabelle 2a sollen die Punkte 1 bis 4 und aus der Tabelle 2b die Punkte 1 bis 5 abgedeckt werden. Die entsprechenden Ausschnitte sind in Tab. 3.4 und 3.5 zu sehen.

Tab. 3.4: Ausschnitt der Tabelle 2a der OIB-Richtlinie 2 [25, S. 25]

Gegenstand	GK 2 (1)	GK 3	GK 4
1 Wände von Treppenhäusern			
1.1 in oberirdischen Geschoßen (2)	REI 30 EI 30	REI 60 EI 60	REI 60 (3) EI 60 (3)
1.2 in unterirdischen Geschoßen	REI 60 EI 60	REI 90 und A2 EI 90 und A2	REI 90 und A2 EI 90 und A2
2 Decke über dem Treppenhaus (4)	REI 30 EI 30	REI 60 EI 60	REI 60 (3) EI 60 (3)
3 Türen in Wänden von Treppenhäusern			
3.1 zu Wohnungen, Betriebseinheiten sowie sonstigen Räumen	EI_2 30	EI_2 30-C	EI_2 30-C-S_{200}
3.2 zu Gängen in oberirdischen Geschoßen (5)	-	E 30-C	E 30-C
3.3 zu Gängen und Räumen in unterirdischen Geschoßen	EI_2 30	EI_2 30-C	EI_2 30-C-S_{200}
4 Treppenläufe und Podeste in Treppenhäusern	R 30	R 60	R 60 und A2

Es wird zwischen den umsetzbaren und den nicht umsetzbaren Erweiterungen unterschieden. Die bereits besprochenen Punkte, wie ober- und unterirdisch, die Gebäudeklasse oder die Verbindung zwischen Bauteilen und Räumen werden hier nicht mehr erwähnt. Somit liegen die einzigen offenen Probleme im Punkt 3 bei Bestimmung der Lage der Türen. Hierbei sind zwei Fälle zu unterscheiden.

Bei den Türen zu oberirdischen Treppenhäusern wird unterschieden, ob der sich auf der anderen Seite befindliche Raum ein Gang oder eine Wohnung beziehungsweise Betriebseinheit ist. Durch eine Erweiterung soll nicht nur die Verbindung zwischen Türen und Räumen möglich werden, sondern es soll auch die Anforderung gestellt werden können, dass beide angrenzenden Räume bestimmte Property Values erfüllen müssen. Im Fall von Punkt 3.2 muss zum Beispiel einer den Raumnamen *Treppenhaus* und der andere *Hauptgang* oder *Nebengang* besitzen. Bei Punkt 3.1 müssen im Gegensatz zu Gängen andere Widmungen in den Properties hinterlegt sein.

Sind die Türen mit unterirdischen Treppenhäusern verbunden, wird nicht zwischen den Nutzungen des auf der anderen Seite liegenden Raums unterschieden. Um die Anforderung

Tab. 3.5: Ausschnitt der Tabelle 2b der OIB-Richtlinie 2 [25, S. 26]

Gegenstand		GK5 mit mechanischer Belüftungsanlage	GK 5 mit automatischer Brandmeldeanlage und Rauchabzugseinrichtung	GK 5 mit Schleuse und Rauchabzugseinrichtung
1	**Wände von Treppenhäusern und Schleusen**			
1.1	in oberirdischen Geschoßen (1)	REI 90 und A2	REI 90 und A2	REI 90 und A2
1.2	in unterirdischen Geschoßen	REI 90 und A2	REI 90 und A2	REI 90 und A2
2	**Decke über dem Treppenhaus (2)**	REI 90 und A2	REI 90 und A2	REI 90 und A2
3	**Türen in Wänden von Treppenhäusern**			
3.1	zu Gängen in oberirdischen Geschoßen (3)	E 30-C	E 30-C-S_{200}	nicht zutreffend
3.2	zu Wohnungen, Betriebseinheiten sowie sonstigen Räumen	EI_2 30-C	EI_2 30-C-S_{200}	unzulässig
3.3	zu Gängen und Räumen in unterirdischen Geschoßen	EI_2 30-C	EI_2 30-C-S_{200}	nicht zutreffend
4	**Türen in Wänden von Schleusen**			
4.1	zu Gängen und Treppenhäusern (3)	nicht zutreffend	nicht zutreffend	E 30-C
4.2	zu Wohnungen, Betriebseinheiten sowie sonstigen Räumen	nicht zutreffend	nicht zutreffend	EI_2 30-C
5	**Treppenläufe und Podeste in Treppenhäusern**	R 90 und A2	R 90 und A2	R 60 und A2

zu erfüllen, reicht es somit aus, dass einer der beiden an die Tür angrenzenden Räume einen vorgegebenen Property Value erfüllt und der andere Wert davon unterschiedlich ist.

Ähnlich zu dem im vorherigen Abschnitt nicht nutzbaren obersten Geschoß ist auch diesmal die Ermittlung der Decke über dem Treppenhaus nicht möglich. Diese entspricht nämlich genau der Decke über dem letzten Geschoß. Daher wird dieser Punkt nicht weiter behandelt.

3.4 Zusammenfassende Anforderungen

Zusammenfassend lassen sich aus den Anforderungen vier notwendige und gleichzeitig umsetzbare Erweiterungen ableiten. Diese stehen jedoch nicht einzeln für sich, sondern sind teilweise voneinander abhängig.

1. **PartOf-Verschachtelung:** Um die Anforderungen der OIB-Richtlinien im Hinblick auf die verschiedenen Gebäudeklassen zu erfüllen, muss eine Verschachtelung von mehreren PartOf-Facets ermöglicht werden. Damit soll schlussendlich von einem Bauteil zum Geschoß und von diesem auf das Gebäude geschlossen werden können.

2. **SpaceBoundary-Relation:** In einigen Fällen reichen die vorhandenen Relationen des PartOf-Facets nicht aus. Dies betrifft vor allem die Verbindung zwischen Bauteilen und Räumen. Durch die Verwendung einer weiteren Relation soll ermöglicht werden, von Wänden, Türen oder Decken auf die angrenzenden Räume zuzugreifen.

3. **PartOf-Anforderungen:** Die Verwendung eines PartOf-Facets und somit die Angabe eines verbundenen Elements alleine entspricht in vielen Fällen nicht den Anforderungen. Es soll mit Hilfe einer Erweiterung möglich werden, an die verbundenen Elemente Anforderungen zu stellen. Konkret ist hier die Verwendung der Property- und Attribut-Facets vorgesehen.

4. **Property-Vergleich:** Aufbauend auf dem vorherigen Punkt genügt es teilweise nicht, nur ein Property-Facet anzugeben. In einigen Fällen, wie zum Beispiel bei Trenndecken oder Treppenhaustüren, müssen die Properties der verbundenen Elemente verglichen werden, um einen Unterschied oder eine Gleichheit feststellen zu können.

Der letzte Punkt steht mit den beiden vorhergehenden in Verbindung. Vor allem ist zu beachten, dass diese Erweiterung nur in Verbindung mit einer neuen Funktionalität möglich ist, wo zu einem Element mehrere verbundene Elemente gefunden werden können. Bei nur einem Element wäre kein Vergleich zwischen Property Values möglich. Im nächsten Kapitel wird IDS um die vier oben genannten Punkte erweitert.

Kapitel 4

Erweiterungen von IDS und IfcOpenShell

Im Wesentlichen sind zwei Bereiche zu betrachten, um die möglichen Anwendungen von IDS zu erhöhen und die Anforderungen aus dem vorherigen Kapitel zu erfüllen. Einerseits muss es ermöglicht werden, die gewünschten Spezifikationen und Erweiterungen in der IDS-Datei abzubilden. Um dies der Standardisierung entsprechend durchzuführen, wird die XSD-Datei dahingehend verändert. Andererseits ist eine Nutzung der Erweiterung mit den verfügbaren Standardprogrammen nicht mehr möglich. Dafür wird der Programmcode des *IfcTester* von *IfcOpenShell* passend zu den Änderungen der XSD-Datei überarbeitet und erweitert.

4.1 Erweiterung von IDS

Die gesamte Erweiterung der XSD-Datei in der Version 0.9.6 (Stand: 20.Juni 2023) besteht im Wesentlichen aus drei einzelnen Bestandteilen. Die ersten beiden beziehen sich auf bereits bestehende komplexe Typen. Die dritte Erweiterung führt ein neues Element mit einem eigenen Typ in den schon davor erweiterten Typ `partOfType` ein. Die vollständig erweiterte XSD-Datei kann Anhang A entnommen werden.

4.1.1 Erweiterung der Relationen im partOfType

Das PartOf-Facet entspricht dem komplexen Typ `partOfType` aus Code 4.1, welcher laut der XSD-Datei das verpflichtend anzugebende Element `entity` enthält. Dieses dient zur Beschreibung der Entität aus dem IFC-Modell. Weiters kann dem `partOfType` bei Bedarf noch das Attribut `relation` hinzugefügt werden, welches den Typ `relations` besitzt.

Code 4.1: Vorhandener `partOfType` aus der XSD-Datei

```
<xs:complexType name="partOfType">
   <xs:sequence>
      <xs:element name="entity" type="ids:entityType" minOccurs="1"/>
   </xs:sequence>
   <xs:attribute name="relation" type="ids:relations" />
</xs:complexType>
```

Theoretisch können alle im IFC-Schema vorhandenen Relationen verwendet werden. Diese sind in der Regel jeweils Subsubklassen von *IfcRelationship* [6] und werden teilweise von van Berlo und Fischer [41] beschrieben. Für jede einzelne Relation ist jedoch eine eigene Programmierung erforderlich, weshalb derzeit nur sechs bestimmte vorgegeben sind. Durch den Typ `relations` können für das Attribut `relation` somit die im Code 4.2 angeführten Schlüsselwörter gewählt werden.

Code 4.2: Vorhandene Relationen in der XSD-Datei

```
<xs:simpleType name="relations">
   <xs:restriction base="xs:string">
      <xs:enumeration value="IFCRELAGGREGATES"/>
      <xs:enumeration value="IFCRELASSIGNSTOGROUP"/>
      <xs:enumeration value="IFCRELCONTAINEDINSPATIALSTRUCTURE"/>
      <xs:enumeration value="IFCRELNESTS"/>
      <xs:enumeration value="IFCRELVOIDSELEMENT"/>
      <xs:enumeration value="IFCRELFILLSELEMENT"/>
   </xs:restriction>
</xs:simpleType>
```

Diese Schlüsselwörter ermöglichen es, stets auf eine Beziehung zwischen genau zwei IFC-Elementen zuzugreifen. Die Verortungsstruktur in IFC wird mittels `IFCRELAGGREGATES` abgebildet. Mit diesem kann zum Beispiel von einem Raum auf das Geschoß oder von diesem weiter auf das Gebäude zugegriffen werden. Die Relation `IFCRELASSIGNSTOGROUP` verbindet ein Element mit dessen zugehöriger Gruppe. Mit `IFCRELCONTAINEDINSPATIALSTRUCTURE` kann man von einem einfachen Bauteil (Wand, Stütze, ...) auf die Verortung (das zugehörige Geschoss) schließen. Die Relation `IFCRELNESTS` bildet eine Zuordnung zwischen Bauteilen und nicht-physischen Objekten (z. B. Kosten). Die Beziehungen `IFCRELVOIDSELEMENT` und `IFCRELFILLSELEMENT` stellen die Verbindung von einem Bauteil (z. B. *IfcWall*) zu einer Öffnung (*IfcOpeningElement*) beziehungsweise von dieser zu dem füllenden Element (z. B. *IfcDoor*, *IfcWindow*) dar. Die beiden Relationen zur Verortung von Bauteilen sind in Abb. 4.1 grafisch veranschaulicht. Die genauen Beziehungen aller sechs Relationen sind sowohl in der IFC-Dokumentation [6] als auch bei van Berlo und Fischer [41] nachlesbar.

Die Angabe einer Relation im PartOf-Facet ist laut Definition nicht verpflichtend. Wird in der IDS-Datei daher keines dieser Schlüsselwörter verwendet, werden in *IfcOpenShell* alle Beziehungen in der oben angegebenen Reihenfolge durchsucht. Dies läuft so lange, bis das in der IDS-Datei angegebene verbundene Element gefunden wird. Daher wird aus Effizienzgründen dazu geraten, stets eine gewünschte Relation anzugeben.

Das Ziel der Erweiterung der Relationen ist, von einem Bauteil auf den zugehörigen Raum schließen zu können. Daher wird als neues Schlüsselwort die Relation *IfcRelSpaceBoundary* eingeführt. buildingSMART International [7] beschreibt Spaceboundaries als physikalische Begrenzungen von Räumen in IFC-Modellen. Darunter versteht man eine Vielzahl von Elementen, wie Wände, Decken, Türen und Fenster. Diese sind genau die gewünschten Elemente. Somit wird der Typ `relations` schließlich um das neue Schlüsselwort erweitert (siehe Code 4.3).

Code 4.3: Erweiterte Relationen in der XSD-Datei

```
<xs:simpleType name="relations">
   <xs:restriction base="xs:string">
      <xs:enumeration value="IFCRELAGGREGATES"/>
      <xs:enumeration value="IFCRELASSIGNSTOGROUP"/>
      <xs:enumeration value="IFCRELCONTAINEDINSPATIALSTRUCTURE"/>
      <xs:enumeration value="IFCRELNESTS"/>
      <xs:enumeration value="IFCRELVOIDSELEMENT"/>
      <xs:enumeration value="IFCRELFILLSELEMENT"/>
      <!-- Extension Patrick Loibl -->
      <xs:enumeration value="IFCRELSPACEBOUNDARY"/>
   </xs:restriction>
</xs:simpleType>
```

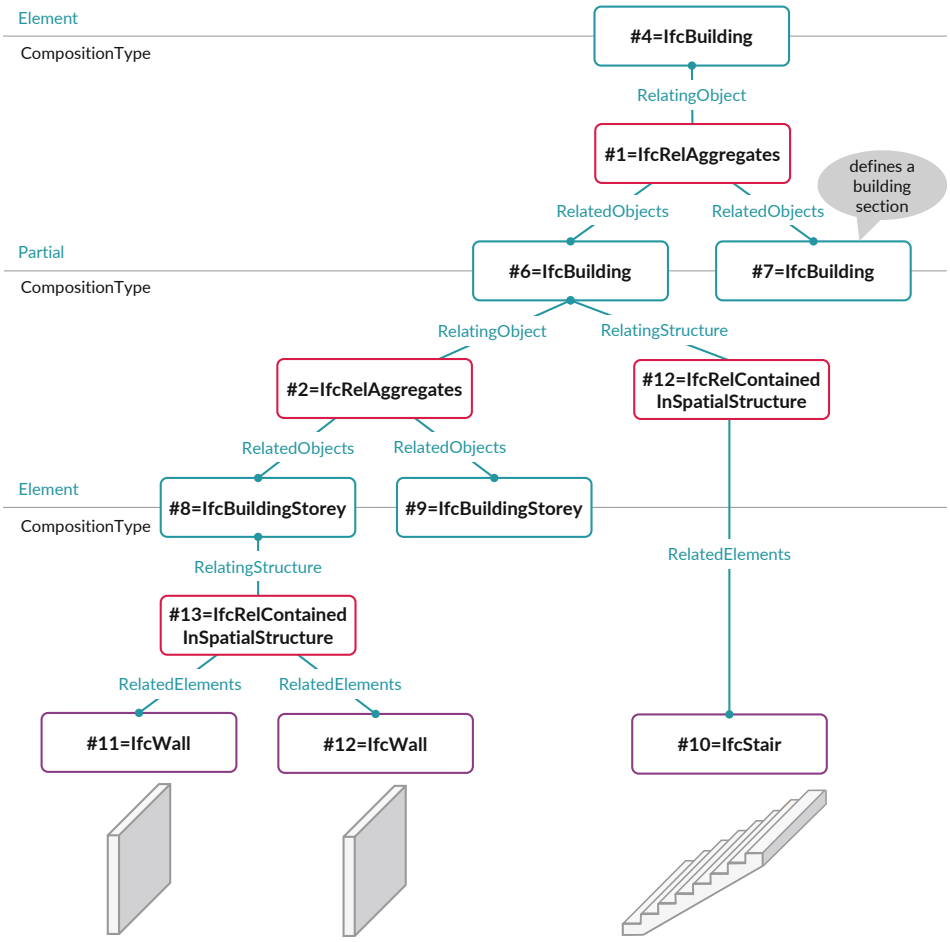

Abb. 4.1: Beispielhafte Abbildung von Beziehungen im IFC-Schema [9, S. 80]

Durch die neu eingeführte Relation können nun Beziehungen von einem Raum auf die begrenzenden Bauteile sowie von einem Bauteil auf die begrenzten Räume verwendet werden. Eine wichtige Anmerkung zur weiteren Verwendung des Schlüsselworts ist die Tatsache, dass durch die Relation *IfcRelSpaceBoundary* in der Regel mehrere Elemente in Verbindung stehen. Ein Raum besitzt nämlich stets einige Spaceboundaries in Form von z. B. mehreren Wänden, zwei Decken und einigen Fenstern und Türen. Auch eine Wand oder eine Tür selbst können mehreren Räumen zugeordnet sein. Dies ist bei der späteren Programmierung der Erweiterungen zu beachten. Eine einfache Applicability einer Wand, die einen Raum begrenzen muss, kann in einer IDS-Datei nun wie in Code 4.4 lauten.

Code 4.4: Applicability mit der neuen verwendbaren Relation

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCWALL</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELSPACEBOUNDARY">
      <entity>
         <name>
            <simpleValue>IFCSPACE</simpleValue>
```

```
            </name>
         </entity>
      </partOf>
   </applicability>
```

4.1.2 Erweiterung von partOfType durch Verschachtelung

Durch die Erweiterung um die SpaceBoundary-Relation im vorhergehenden Abschnitt ist es nun zum Beispiel möglich, in den Anforderungen der Applicability anzugeben, dass eine Wand in Verbindung mit einem Raum stehen muss. Diese Anforderung ist jedoch trivial, da sich normalerweise in jedem IFC-Modell auf mindestens einer Seite einer Wand ein Raum befindet. Um die aufgebrachten Anforderungen an IDS zu erfüllen, ist es wichtig, das verbundene Element noch weiter beschreiben zu können.

Neben der Entität soll zum Beispiel auch die Beschreibung eines Property oder eines Attributs möglich sein, wie es bei den PartOf-Anforderungen beschrieben wird. In der XSD-Datei lässt sich dies durch Hinzufügen zweier Elemente zum `partOfType` umsetzen, nämlich den bereits bestehenden Typen `attributeType` und `propertyType`. Der erweiterte Typ ist in Code 4.5 zu sehen. Beide hinzugefügten Elemente sind jedoch nur optional (im Gegensatz zur verpflichtend anzugebenden Entität), was durch `minOccurs="0"` ausgedrückt wird. Eine maximale Beschränkung, wie viele Properties oder Attribute gefordert werden dürfen, ist nicht vorgesehen. Dies erfolgt durch die Angabe von `maxOcccurs="unbounded"`.

Code 4.5: Teilweise erweiterter `partOfType` aus der XSD-Datei

```
<xs:complexType name="partOfType">
   <xs:sequence>
      <xs:element name="entity" type="ids:entityType" minOccurs="1"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="attribute" type="ids:attributeType" minOccurs="0
          " maxOccurs="unbounded"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="property" type="ids:propertyType" minOccurs="0"
          maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="relation" type="ids:relations" />
</xs:complexType>
```

Folgendes Beispiel in Code 4.6 zeigt eine durch die Erweiterungen aus dem vorherigen und dem aktuellen Abschnitt mögliche IDS-Struktur. In der Applicability wird eine Tür (`IFCDOOR`) spezifiziert, welche einen Raum begrenzt. Durch die Erweiterung wird nun zusätzlich vom Raum der Property Value `Wohnen und Aufenthalt` gefordert.

Code 4.6: Applicability mit dem erweiterten PartOf-Facet

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCDOOR</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELSPACEBOUNDARY">
      <entity>
```

```
            <name>
               <simpleValue>IFCSPACE</simpleValue>
            </name>
         </entity>
         <property datatype="IfcLabel" >
            <propertySet>
               <simpleValue>Pset_SpaceSpecific</simpleValue>
            </propertySet>
            <name>
               <simpleValue>WidmungBehoerde</simpleValue>
            </name>
            <value>
               <simpleValue>Wohnen und Aufenthalt</simpleValue>
            </value>
         </property>
      </partOf>
   </applicability>
```

Im Abschnitt 3.4 über die zusammenfassenden Anforderungen an IDS wird zusätzlich noch eine Verschachtelung von PartOf-Facets gefordert. Daher wird analog zu den beiden vorherigen Typen (`attributeType` und `propertyType`) auch der Typ `partOfType` hinzugefügt (siehe Code 4.7). Dies erzeugt eine Art rekursiver Verschachtelung, da das Element sich selbst beinhaltet. Durch Angabe der richtigen Relationen ist es nun möglich, eine Beziehung von einem Bauteil zu einem Raum und davon weiter zu einem Geschoß und einem Gebäude herzustellen.

Code 4.7: Teilweise erweiterter `partOfType` aus der XSD-Datei

```
<xs:complexType name="partOfType">
   <xs:sequence>
      <xs:element name="entity" type="ids:entityType" minOccurs="1"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="partOf" type="ids:partOfType" minOccurs="0"
          maxOccurs="unbounded"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="attribute" type="ids:attributeType" minOccurs="0
          " maxOccurs="unbounded"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="property" type="ids:propertyType" minOccurs="0"
          maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="relation" type="ids:relations" />
</xs:complexType>
```

Im folgenden Ausschnitt einer in Code 4.8 angegebenen IDS-Datei sind alle Erweiterungen dieses Abschnitts enthalten. Es können nun PartOf-Facets verschachtelt, die neu eingeführte Relation *IfcRelSpaceBoundary* verwendet und Anforderungen an die verbundenen Elemente gestellt werden. Die im Beispielcode angegebene Wand muss in einem oberirdischen Geschoß sein, welches die Anforderungen bezüglich des gegebenen Attributs erfüllen muss (`Elevation > 0`). Zusätzlich muss das Geschoß Teil eines Gebäudes sein, welches wiederum ein bestimmtes Property besitzen muss. Dieses Property fordert vom Gebäude nun die passende Gebäudeklasse `GK2`. Die einzelnen Relationen sind dieselben wie jene in Abb. 4.1.

Code 4.8: Applicability mit Verschachtelung von PartOf-Facets

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCWALL</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
      <entity>
         <name>
            <simpleValue>IFCBUILDINGSTOREY</simpleValue>
         </name>
      </entity>
      <partOf relation="IFCRELAGGREGATES">
         <entity>
            <name>
               <simpleValue>IFCBUILDING</simpleValue>
            </name>
         </entity>
         <property datatype="IfcLabel">
            <propertySet>
               <simpleValue>ZDB_Building_Infos</simpleValue>
            </propertySet>
            <name>
               <simpleValue>Gebaeudeklasse</simpleValue>
            </name>
            <value>
               <simpleValue>GK2</simpleValue>
            </value>
         </property>
      </partOf>
      <attribute>
         <name>
            <simpleValue>Elevation</simpleValue>
         </name>
         <value>
            <xs:restriction base="xs:integer">
               <xs:minInclusive value="0" />
            </xs:restriction>
         </value>
      </attribute>
   </partOf>
</applicability>
```

4.1.3 Erweiterung von partOfType um einen Property-Vergleich

Die letzte Erweiterung führt zur Durchführung eines Vergleichs von mehreren Property Values ein neues Element mit dem Namen `multiObjectProcessing` in `partOfType` ein (siehe Code 4.9). Dieses Element wird in der Reihenfolge hinter dem Property-Facet eingefügt, da es mit diesem im Zusammenhang steht. Der Typ des Elements ist `idsValue` und somit ein einzelner Wert. Die Erweiterung kann optional angewendet werden. Im Unterschied zu den anderen ist sie jedoch auf die einmalige Anwendung im PartOf-Facet beschränkt.

Aufgrund des gewählten Typs der Erweiterung wird dem Element in der IDS-Datei ein einfacher String als beliebiger Wert zugewiesen, wie in folgendem Beispiel zu sehen ist. Um jedoch die Funktionalität der Erweiterung zu gewährleisten, dürfen nur drei bestimmte Schlüsselwörter

angegeben werden. Diese werden in weiterer Folge noch erläutert und sind in Tab. 4.1 zusammengefasst. Wird ein nicht bekanntes Schlüsselwort angegeben, erfolgt die Auswertung der IDS-Datei, als gäbe es kein Element `multiObjectProcessing`. In einer zusätzlichen Erweiterung wäre es denkbar, den Typ des Elements in der XSD-Datei zu ändern, damit eine Überprüfung des eingegebenen Schlüsselworts auf seine Gültigkeit, ähnlich zu den Relationen, durchgeführt wird.

Code 4.9: Vollständig erweiterter `partOfType` aus der XSD-Datei

```
<xs:complexType name="partOfType">
   <xs:sequence>
      <xs:element name="entity" type="ids:entityType" minOccurs="1"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="partOf" type="ids:partOfType" minOccurs="0"
          maxOccurs="unbounded"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="attribute" type="ids:attributeType" minOccurs="0
          " maxOccurs="unbounded"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="property" type="ids:propertyType" minOccurs="0"
          maxOccurs="unbounded"/>
      <!-- Extension Patrick Loibl -->
      <xs:element name="multiObjectProcessing" type="ids:idsValue"
          minOccurs="0" maxOccurs="1"/>
   </xs:sequence>
   <xs:attribute name="relation" type="ids:relations" />
</xs:complexType>
```

Folgende Szenarien können im Zusammenhang mit dem Schlüsselwort *multiObjectProcessing* unterschieden werden:

1. **Keine Angabe von multiObjectProcessing**:
   Wird das Element nicht verwendet oder wird ein falsches oder kein Schlüsselwort angegeben, erfolgt eine im Ablauf veränderte Überprüfung des Property. Standardmäßig muss das vorhandene Element den Anforderungen des Property-Facet entsprechen. Da bei der Relation *IfcRelSpaceBoundary* jedoch mehrere Elemente ermittelt werden, muss nur eines die Spezifikation bezüglich des Property erfüllen. Ist zum Beispiel eine Wand gegeben, die in Verbindung zu einem Wohnraum stehen soll, reicht es, wenn einer der angrenzenden Räume das entsprechende Property besitzt. Eine Vorgabe von mehreren Properties ist möglich. Dann muss mindestens ein Element (in diesem Fall *IfcSpace*) alle gegebenen Properties erfüllen. In Zukunft ist eine Regelung der Anzahl der Elemente, welche alle Properties erfüllen sollen, über die beiden Attribute *min/maxOccurs* im PartOf-Facet denkbar.

2. **samePropertyValuesForAllObjects**:
   Dieses Schlüsselwort wird verwendet, um anzugeben, dass alle verbundenen Elemente den selben Wert des angegebenen Property zu besitzen haben. Wird zum Beispiel im PartOf-Facet zur Entität *Raum* das Property *Wohnungsnummer* angegeben, so müssen die Räume beidseits der ausgehenden Entität (z. B. Tür, Wand) dieselbe Wohnungsnummer haben. In der IDS-Datei ist die Angabe von einem oder mehreren Properties möglich. Dabei müssen die Elemente bei allen angegebenen Properties denselben Wert haben. Wichtig ist, dass kein Wert (Property Value) in der IDS-Datei vorgegeben wird, da dieser **nicht** überprüft wird. Für die Übereinstimmung des Property Values aller Elemente ist der tatsächliche Wert irrelevant.

3. **differentPropertyValuesForAnyObject**:
   Dieses Schlüsselwort findet Anwendung, wenn die Elemente einen Unterschied in den Property Values aufweisen sollen. Dabei muss nicht jeder einzelne Wert unterschiedlich sein, sondern es muss nur mindestens zwei verschiedene geben. Dieses Schlüsselwort kann mit einem oder mehreren Properties benutzt werden. Im Gegensatz zu vorhin ist es diesmal möglich, auch einen Property Value anzugeben. Dieser muss in dem Fall von mindestens einem Element erfüllt sowie von mindestens einem nicht erfüllt werden, sprich einen unterschiedlichen Wert haben.

4. **differentObjectsForEachPropertyValue**:
   Bei der Verwendung des letzten Schlüsselworts sind mehrere Property Values anzugeben. Für jeden dieser Werte muss es mindestens ein Element geben, das diese Anforderung erfüllt. Die Property Values können auch von unterschiedlichen Properties sein. Anzumerken ist jedoch, dass jedes Element nur zur Erfüllung einer Anforderung verwendet wird und nicht doppelt benutzt werden kann. Eine genauere Steuerung ist nicht möglich.

Tab. 4.1: Überblick über die Anwendungsmöglichkeiten von `multiObjectProcessing`

Schlüsselwort	Gleichheit oder Unterschied	ein oder mehrere Properties	Property Value angeben
keines oder falsches	—	mehrere möglich	möglich
samePropertyValuesForAllObjects	Gleichheit	mehrere möglich	nicht möglich
differentPropertyValuesForAnyObject	Unterschied	mehrere möglich	möglich
differentObjectsForEachPropertyValue	Unterschied	mehrere möglich	verpflichtend

Beispiele zu den einzelnen Anwendungen der Schlüsselwörter werden hier nicht gezeigt, sondern sind in Kapitel 5 mit Verbindungen zur Praxis erklärt. Der Unterschied im Programmablauf des *IfcTester* von *IfcOpenShell* wird im nächsten Abschnitt beschrieben.

4.2 Erweiterung von IfcOpenShell

Neben der Erweiterung der XSD-Datei, welche notwendig ist, um die gewünschten Anforderungen in einer IDS-Datei zu beschreiben, muss auch *IfcOpenShell* erweitert werden. Nur dadurch können die neuen Möglichkeiten auch getestet und genutzt werden. Bevor dies jedoch geschehen kann, wird ein Überblick über alle vorkommenden Klassen in den relevanten Dateien gegeben. Darauffolgend beschreibt der erste Abschnitt die Beseitigung von Bugs in den für den *IfcTester* von GitHub [15] heruntergeladenen Dateien. Die restlichen Abschnitte widmen sich den neuen Programmteilen und Erweiterungen aufbauend auf den vorausgehenden Änderungen der XSD-Datei. Die gesamten Beschreibungen beziehen sich auf die *IfcOpenShell*-Version v0.7.0 sowie den Stand des *IfcTesters* vom Juli 2023.

Die Datei *ids.py* beinhaltet die beiden Klassen `Ids` und `Specification`. Neben der Klasse `Restriction` beinhaltet die Datei *facet.py* noch alle Klassen aus Tab. 4.2. Die beiden Klassen `Facet` und `Result` werden allen anderen darunter stehenden vererbt. Alle veränderten oder näher beschriebenen Klassen sind in der Tabelle fett hervorgehoben.

Tab. 4.2: Klassen in der Datei *facet.py*

	Facet	**Result**
	Entity	EntityResult
	Attribute	AttributeResult
Restriction	Classification	ClassificationResult
	PartOf	PartOfResult
	Property	**PropertyResult**
	Material	MaterialResult

4.2.1 Behebung von Bugs in der verwendeten Version

Insgesamt wurden im *IfcTester* von *IfcOpenShell* zwei Bugs gefunden, die relativ einfach gelöst werden können. Beide Fehler befinden sich in der Klasse `PartOf` der Datei *facet.py* und werden nun in der vorkommenden Reihenfolge aufgezeigt.

Der erste Bug liegt im elif-Zweig der Relation `IFCRELFILLSELEMENT`. Hier lässt sich ein falscher Methodenaufruf finden. Die Methode `.filled_opening` ist in der Datei nicht vorhanden. Dies muss zur richtigen Methode `.get_filled_opening` geändert werden.

```
elif self.relation == "IFCRELFILLSELEMENT":
    opening = self.get_filled_opening(inst) # Bugfix Patrick Loibl
```

Der zweite Fehler tritt ebenfalls in der Klasse `PartOf` auf, kann aber in der Methode `parse` der Klasse `Facet` gelöst werden. Im PartOf-Facet soll bei jedem Zweig der elif-Verzweigung der Relationen das ermittelte Element (z. B. `container`) mit der Entität (`self.entity`) verglichen werden, die in der IDS-Datei angegeben wurde. Folgend ist ein Beispiel aus dem Ast der Relation *IfcRelContainedInSpatialStructure* angegeben:

```
if container.is_a().upper() != self.entity:
```

Dem PartOf-Facet wird jedoch nie die richtige Entität aus der IDS-Datei unter `self.entity` zugewiesen, sondern es behält stets die initiale Entität `IFCWALL`. Die richtige Entität wird allein unter `name` gespeichert. Die Lösung liegt in Veränderung der vorhin erwähnten Methode `parse` der Klasse `Facet`. Jedes Facet aus der IDS-Datei hat ein XML-Attribut *name*, dass die jeweilige Entität enthält. In der Iteration (Zeile 4) werden alle Attribute durchgegangen und in die Variable `name` gespeichert (Zeile 5). Ist also das Attribut *name* an der Reihe und in der Variable `name` vorhanden, wird der im Attribut enthaltene Wert (`value["simpleValue"]`) dem PartOf-Facet durch eine if-Verzweigung in die Variable `entity` hinzugefügt (Zeile 14 und 15). Die entsprechende Änderung ist im Code 4.10 angeführt.

Code 4.10: Methode `parse` der Klasse `Facet`

```
1     def parse(self, xml):
2         setattr(self, "minOccurs", 1)
```

```
3         setattr(self, "maxOccurs", 1)
4         for name, value in xml.items():
5            name = name.replace("@", "")
6            if isinstance(value, dict) and "simpleValue" in value.keys():
7               setattr(self, name, value["simpleValue"])
8            elif isinstance(value, dict) and "restriction" in value.keys():
9               setattr(self, name, Restriction().parse(value["restriction"
                    ][0]))
10              # TODO handle more than one restriction: return [restriction(r)
                    for r in v["restriction"]]
11           else:
12              setattr(self, name, value)
13           # Bugfix Patrick Loibl
14           if name == "name":
15              setattr(self,"entity",value['simpleValue'])
16        return self
```

4.2.2 Verschachtelung der Facets in ids.py

In Abschnitt 4.1.2 wird die Erweiterung des `PartOfType` um bestehende Typen beschrieben. In *IfcOpenShell* finden diese zusätzlichen Angaben jedoch noch keine Berücksichtigung. Nur die Elemente der ersten Stufe innerhalb der Applicability beziehungsweise der Requirements werden verwendet. In folgenden Abschnitten wird der Ablauf der beteiligten Methoden und die darauf aufbauende Erweiterung näher beschrieben.

Für jede Spezifikation aus der IDS-Datei wird in der Datei *ids.py* ein eigenes Objekt der Klasse `Specification` erzeugt. Diesem Objekt werden durch die Methode `parse` aus Code 4.11 zuerst grundlegende Infos (`Name, minOccurs, maxOccurs, ifcVersion`) zugewiesen. Danach folgt die Erstellung und Zuordnung der Applicability und der Requirements. Dies erfolgt durch den Aufruf der weiteren Methode `parse_clause` (Zeile 6 und 7).

Code 4.11: Methode `parse` der Klasse `Specification`

```
1     def parse(self, ids_dict):
2        self.name = ids_dict.get("@name", "")
3        self.minOccurs = ids_dict["@minOccurs"]
4        self.maxOccurs = ids_dict["@maxOccurs"]
5        self.ifcVersion = ids_dict["@ifcVersion"]
6        self.applicability = self.parse_clause(ids_dict["applicability"])
             if "applicability" in ids_dict else []
7        self.requirements = self.parse_clause(ids_dict["requirements"])
             if "requirements" in ids_dict else []
8        return self
```

Die Methode `parse_clause` (siehe Code 4.12) bekommt das aktuelle Objekt der Spezifikation (`self`) sowie das Dictionary `clause` übergeben. In diesem sind die Elemente der ersten Stufe der Applicability beziehungsweise der Requirements enthalten. Die Elemente sind selbst wieder Dicitionaries und beinhalten ihre Kindelemente. Diese Tatsache wird verwendet, um mit dem jeweiligen Element der ersten Stufe, welches durch eine for-Schleife in `facet_xml` gespeichert ist, wieder die Funktion `parse_clause` aufzurufen. So werden dem erstellten Facet die weiteren enthaltenen Facets unter der Variable `extendedSpecification` angefügt. Die Rekursion wird so lange ausgeführt, bis ein Element keine Kindelemente mehr besitzt.

Code 4.12: Methode `parse_clause` der Klasse `Specification`

```
def parse_clause(self, clause):
    results = []
    for name, facets in clause.items():
        if name not in ["entity", "attribute", "classification", "partOf",
            "property", "material"]:
            continue
        if not isinstance(facets, list):
            facets = [facets]
        for facet_xml in facets:
            name_capitalised = name[0].upper() + name[1:]
            facet = globals()[name_capitalised]().parse(facet_xml)

            # Extension Patrick Loibl
            facet.extendedSpecification = self.parse_clause(facet_xml)

            results.append(facet)
    return results
```

Nun folgt ein Beispiel, um den beschriebenen Ablauf besser nachvollziehen zu können. Ausgehend von der Applicability der IDS-Anforderung im Code 4.13 werden die Unterschiede gezeigt. Wird das Programm ohne die Erweiterung durchgeführt, wird eine Spezifikation erstellt, welche in der Applicability die Elemente `Entity` und `PartOf` als jeweilige Objekte ihrer Klassen (Facets) enthält (siehe Abb. 4.2). Im zweiten Objekt (PartOf) ist ersichtlich, dass die Informationen der beiden Kindelemente `partOf` und `property` zwar vorhanden sind, jedoch nicht als verwendbares Objekt der jeweiligen Klassen, sondern als Dictionaries.

Im Unterschied dazu ist in Abb. 4.3 zu erkennen, dass bei Verwendung der Erweiterung in *ids.py* dem PartOf-Facet dessen Kindelemente als weitere Facets unter `extendedSpecification` angefügt wurden. Das verdeutlicht, dass die Erweiterung nicht nur in der IDS-Datei angegeben werden kann, sondern auch in der *IfcOpenShell* Verwendung findet.

Code 4.13: Ausschnitt aus einer IDS-Anforderung inklusive Verschachtelung

```
<applicability>
    <entity>
        <name>
            <simpleValue>IFCWALL</simpleValue>
        </name>
    </entity>
    <partOf relation="IFCRELSPACEBOUNDARY">
        <entity>
            <name>
                <simpleValue>IFCSPACE</simpleValue>
            </name>
        </entity>
        <partOf relation="IFCRELAGGREGATES">
            <entity>
                <name>
                    <simpleValue>IFCBUILDINGSTOREY</simpleValue>
                </name>
            </entity>
            <attribute>
                <name>
                    <simpleValue>Elevation</simpleValue>
                </name>
```

```
23                <value>
24                   <xs:restriction base="xs:integer">
25                      <xs:minInclusive value="0" />
26                   </xs:restriction>
27                </value>
28             </attribute>
29          </partOf>
30          <property datatype="IfcLabel">
31             <propertySet>
32                <simpleValue>Pset_SpaceSpecific</simpleValue>
33             </propertySet>
34             <name>
35                <simpleValue>SpaceName</simpleValue>
36             </name>
37             <value>
38                <simpleValue>Treppe</simpleValue>
39             </value>
40          </property>
41       </partOf>
42    </applicability>
```

Abb. 4.2: Screenshot einer Spezifikation ohne Verschachtelung in Visual Studio Code

Abb. 4.3: Screenshot einer Spezifikation mit Verschachtelung in Visual Studio Code

4.2.3 Verwendung von IfcRelSpaceBoundary

Die nächste Erweiterung der XSD-Datei ist die Verwendung der Relation *IfcRelSpaceBoundary*. Wird diese Beziehung in der nicht erweiterten *IfcOpenShell* verwendet, läuft das Programm zwar durch, ermittelt aber kein verbundenes Element. Dadurch gilt die PartOf-Anforderung dann nicht als erfüllt. In diesem Abschnitt wird die Erweiterung der Klasse *PartOf* beschrieben, damit die Relation *IfcRelSpaceBoundary* sinnvoll verwendet werden kann.

Ein Objekt der Klasse `PartOf` besitzt stets das Attribut `relation`, welches zu Beginn erstellt und mit `None` belegt wird. Ist in der IDS-Datei ein passender Wert angegeben, wird dieser als String übernommen und existiert somit in der *IfcOpenShell*-Struktur. Verwendung findet dieser Wert beim Aufruf eines PartOf-Facets mit einem zu prüfenden Element. Je nach angegebener Relation wird ein anderer Weg der Verzweigung aus Code 4.14 eingeschlagen. Der letzte Block umfasst den Fall der neuen Relation und wurde im Zuge dieser Arbeit eingefügt. Der Ablauf innerhalb dieses Blocks entspricht in großen Teilen dem der anderen Fälle und wird im Code 4.15 angeführt und in Abb. 4.4 dargestellt. Drei wesentliche Unterschiede sind jedoch vorhanden, welche bei der Programmierung besondere Beachtung erfordern:

1. Bei der Verwendung der Relation *IfcRelSpaceBoundary* werden zu einem gegebenen Element meistens mehrere verbundene Elemente ermittelt. Zu jedem Raum gibt es auf jeden Fall mehrere Wände und Decken, die ihn begrenzen. Auch eine Decke hat in der Regel oberhalb und unterhalb jeweils mindestens einen Raum, ausgenommen sie liegt im obersten Geschoß.

2. Die Relation funktioniert in zwei Richtungen, nämlich von einem Raum zu den Grenzen sowie von einem Bauteil zu den begrenzten Räumen. Bei den anderen gibt es stets nur eine Richtung. Daher sind zum Beispiel bei Öffnungselementen zwei Relationen vorhanden, um einmal auf das umgebende und einmal auf das füllende Bauteil zu schließen.

3. Da durch die vorhandene Implementierung in *IfcOpenShell* allein bei dieser Relation, wie im ersten Punkt genannt, eine Liste an Elementen ermittelt wird, kann auch nur hier die Erweiterung *multiObjectProcessing* verwendet werden. Daher unterscheidet sich die Überprüfung der Properties wesentlich von den anderen Abläufen.

Code 4.14: elif-Verzweigung der Relationen in IfcOpenShell

```
if not self.relation: ...
elif self.relation == "IFCRELAGGREGATES": ...
elif self.relation == "IFCRELASSIGNSTOGROUP": ...
elif self.relation == "IFCRELCONTAINEDINSPATIALSTRUCTURE": ...
elif self.relation == "IFCRELNESTS": ...
elif self.relation == "IFCRELVOIDSELEMENT": ...
elif self.relation == "IFCRELFILLSELEMENT": ...
# Extension Patrick Loibl
elif self.relation == "IFCRELSPACEBOUNDARY": ...
```

Die Funktionalität in beide Richtungen wird zu Beginn im Code 4.15 ersichtlich. Im ersten Schritt wird der Typ des gegebenen Elements ermittelt. Im Wesentlichen ist nur interessant, ob es sich um einen Raum handelt oder nicht. Beim Typ `IfcSpace` wird der erste folgende Block (Zeilen 6–7) aufgerufen, bei jedem anderen Typ der zweite Block (Zeilen 9–10). Der Grund dafür liegt im Unterschied beim Abrufen der verbundenen Elemente. Ausgehend von einem Raum (`inst`) wird ermittelt, welche Spaceboundaries ihn begrenzen (`inst.BoundedBy`) sowie welchen Bauteilen diese entsprechen (`inst.BoundedBy[i].RelatedBuildingElement`). Diese

werden dann in der Liste `elements` gespeichert. Die Syntax bei anderen gegebenen Elementen ist gleich, nur werden die vorhandenen Boundaries und die dazugehörigen Räume abgerufen (`inst.ProvidesBoundaries[i].RelatingSpace`) und in der Liste `elements` gespeichert.

Der restliche Ablauf der Methode ist unabhängig von dem Typ des Elements. Nach der Ermittlung der Liste `elements` wird geprüft, ob verbundene Elemente vorhanden sind (Zeile 11). Ist dies nicht der Fall, wird die Prüfung beendet und als Grund (`reason`) `NOVALUE` gespeichert. Wenn die Bedingung aber erfüllt ist, wird als nächstes der Typ der verbundenen Elemente geprüft (Zeile 18). Dieser muss der in der IDS-Datei angegebenen Entität entsprechen. Bestandene Elemente werden in dem neuen Set `elements_to_check` gespeichert, welches im weiteren Verlauf statt der ursprünglichen Liste Verwendung findet. Ist in diesem Set mindestens ein Element, wird die Methode `checkSpaceBoundary` aufgerufen. In dieser findet die weitere Überprüfung der Elemente statt. Existiert jedoch kein Element mit dem richtigen Typ, wird der Ablauf beendet und als Grund der falsche Typ des letzten Elements genannt.

Die Methode `checkSpaceBoundary` ist im Code 4.16 angeführt und schematisch in Abb. 4.5 abgebildet. Sie bekommt das aktuelle Objekt von PartOf (`self`), die Liste der zu prüfenden Elemente sowie den Status `is_pass` und den Grund übergeben. Da die Überprüfung der Properties, wie schon vorher erwähnt, anhand der Schlüsselwörter unterschiedlich stattfindet, wird zuerst die Prüfung der Attribute in der eigenen Methode `attribute_check` durchgeführt. Sobald ein Element die Anforderung nicht erfüllt, wird abgebrochen und das Ergebnis zurückgegeben. Nur wenn jedes Element das Attribut-Facet erfüllt, geht das Programm weiter zur Überprüfung der Properties (Code 4.16, ab Zeile 7). Diese wird in folgendem Abschnitt mit spezieller Betrachtung des Unterschieds der Schlüsselwörter beschrieben. Nach jener Überprüfung findet auch der Aufruf der Rekursion statt (Code 4.16, Zeile 40).

Code 4.15: elif-Block der Relation `IFCRELSPACEBOUNDARY` in der Klasse `PartOf`

```
1    #  Extension Patrick Loibl
2    elif self.relation == "IFCRELSPACEBOUNDARY":
3        type = inst.wrapped_data.is_a(True).split(".")[1]
4        elements = []
5        if type == "IfcSpace":
6            for i in range(0,len(inst.BoundedBy)):
7                elements.append(inst.BoundedBy[i].RelatedBuildingElement)
8        else:
9            for i in range(0,len(inst.ProvidesBoundaries)):
10               elements.append(inst.ProvidesBoundaries[i].RelatingSpace)
11       is_pass = bool(len(elements))
12       if not is_pass:
13           reason = {"type": "NOVALUE"}
14       else:
15           elements_to_check = set()
16           for element in elements:
17               elementtyp = element.wrapped_data.is_a(True).split(".")[1]
18               if elementtyp.upper() == self.name:
19                   elements_to_check.add(element)
20           if len(elements_to_check) > 0:
21               is_pass=True
22               (is_pass,reason) = self.checkSpaceBoundary(elements_to_check,
                     is_pass,reason)
23           else:
24               is_pass = False
25               reason = {"type": "ENTITY", "actual": elementtyp}
```

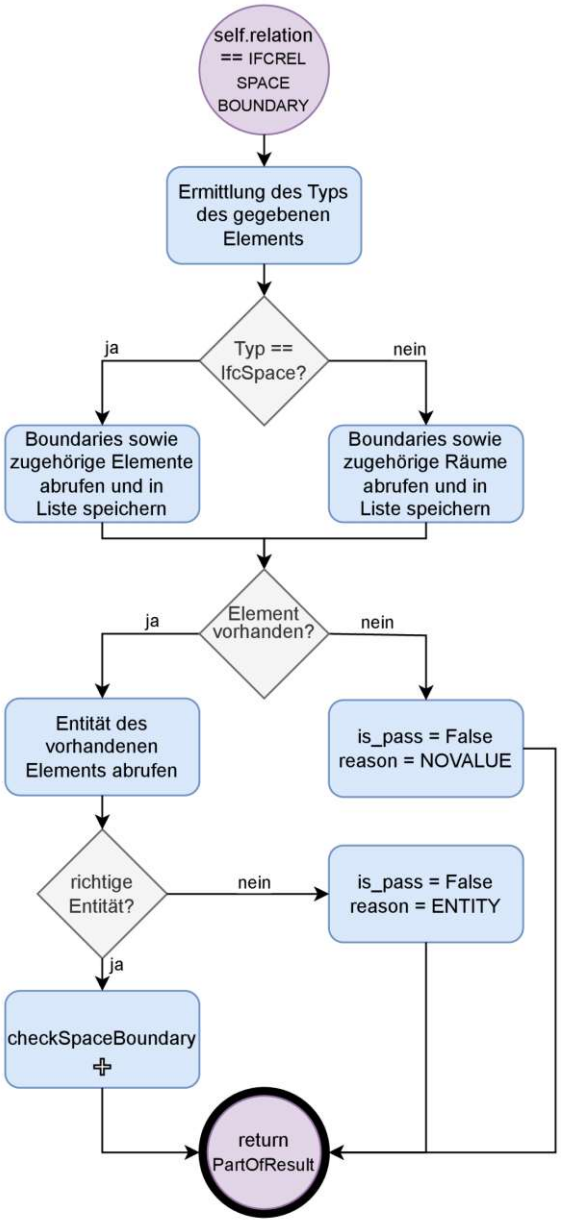

Abb. 4.4: Ablauf bei der Relation `IFCRELSPACEBOUNDARY`

Code 4.16: Methode `checkSpaceBoundary` der Klasse `PartOf`

```
1   def checkSpaceBoundary(self, elements_to_check, is_pass, reason):
2       for element in elements_to_check:
3           (is_pass,reason) = self.attribute_check(element,is_pass,reason)
4           if not is_pass:
5               break
6       if is_pass:
7           if (not hasattr(self,"multiObjectProcessing")) or (hasattr(self,"
                multiObjectProcessing") and self.multiObjectProcessing not in
                ["samePropertyValuesForAllObjects", "
                differentPropertyValuesForAnyObject","
                differentObjectsForEachPropertyValue"]):
8               for element in elements_to_check:
9                   is_pass = True
10                  (is_pass,reason) = self.property_check(element,is_pass,
                        reason)
11                  if is_pass:
12                      break
13          elif hasattr(self,"multiObjectProcessing") and self.
                multiObjectProcessing == "samePropertyValuesForAllObjects":
14              (is_pass,reason)=self.propertyValueComparisonOfMultipleObjects(
                    elements_to_check, "same")
15          elif hasattr(self,"multiObjectProcessing") and self.
                multiObjectProcessing == "differentPropertyValuesForAnyObject"
                :
16              for element in elements_to_check:
17                  is_pass=True
18                  (is_pass,reason) = self.property_check(element,is_pass,
                        reason)
19                  if is_pass:
20                      break
21              if is_pass:
22                  (is_pass,reason)=self.
                        propertyValueComparisonOfMultipleObjects(
                        elements_to_check, "different")
23          elif hasattr(self,"multiObjectProcessing") and self.
                multiObjectProcessing == "differentObjectsForEachPropertyValue
                ":
24              properties_to_check = list()
25              for specification in self.extendedSpecification:
26                  properties_to_check.append(specification) if isinstance(
                        specification, Property) else []
27              for element in elements_to_check:
28                  for property_to_check in properties_to_check:
29                      if property_to_check(element).is_pass:
30                          properties_to_check.remove(property_to_check)
31                          break
32                      else:
33                          continue
34              if len(properties_to_check) == 0:
35                  is_pass = True
36              else:
37              is_pass = False
38                  reason = {"type": "ENTITY", "actual": "there are no
                        different objects"}
39      for element in elements_to_check:
40          (is_pass,reason) = self.partOf_recursion(element,is_pass,reason)
41          if not is_pass:
42              break
```

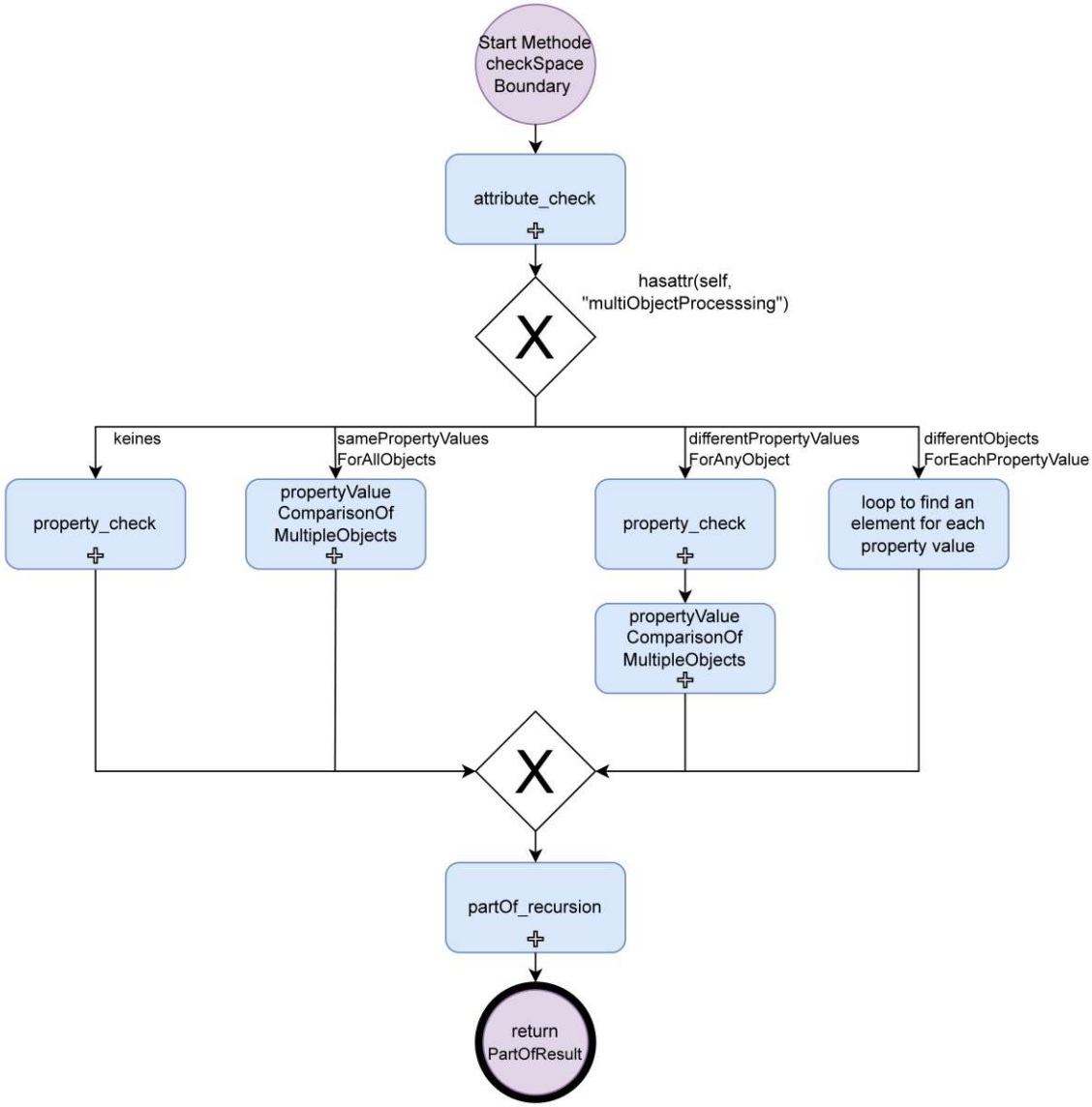

Abb. 4.5: Ablauf der Methode `checkSpaceBoundary`

4.2.4 Unterschiede bei den Arten von multiObjectProcessing

Nachdem der im vorherigen Abschnitt erwähnte Attribut-Check bestanden wurde, handelt dieser Teil vom weiteren Verlauf der Methode `checkSpaceBoundary` (siehe Code 4.16). Das Augenmerk liegt hierbei vor allem auf den vier verschiedenen Szenarien im Zusammenhang mit dem eingeführten Tag *multiObjectProcessing*.

4.2.4.1 Keine Verwendung von `multiObjectProcessing`

Der erste Fall (Zeilen 7–12) aus Code 4.16 tritt ein, wenn das Element `multiObjectProcessing` entweder nicht oder mit einem falschen Schlüsselwort verwendet wird. In beiden Fällen erfolgt ein einfacher Property-Check. Wie in den aufgestellten Anforderungen an IDS genannt, gilt die Prüfung als erfüllt, sobald es ein Element gibt, dass die Methode erfolgreich (`is_pass=True`) durchlaufen hat. Dann kann die Schleife abgebrochen werden.

4.2.4.2 Verwendung von `samePropertyValuesForAllObjects`

Bei der Verwendung dieses Schlüsselworts soll die Gleichheit von Property Values abgefragt werden. Die Prüfung findet in der eigenen Methode `propertyValueComparisonOfMultipleObjects` statt, die im Code 4.17 zu sehen ist. Sie wird mit dem aktuellen Objekt der Klasse `PartOf` aufgerufen und bekommt die zu prüfenden Elemente sowie den Wert `same` übergeben. Letzterer dient als Schlüsselwort und ist notwendig, da die Methode mehrere Verwendungen hat.

Der gesamte Inhalt der Methode läuft innerhalb einer Schleife über alle übergebenen Property-Facets, da mehrere davon in der IDS-Datei angegeben werden können. Danach folgt eine weitere Iteration über alle zu prüfenden Elemente. Mit diesen wird jeweils nacheinander das Property-Facet aufgerufen (siehe Zeile 8: `facet(element)`). Dabei wird die Methode `__call__` der Klasse `Property` gestartet. Innerhalb der Methode findet der gesamte Ablauf der Prüfung von Properties statt. Hierbei ist auf eine kleine Erweiterung in dessen Implementierung hinzuweisen. Normalerweise wird in der Ergebnisrückgabe der Property Value nur dann gespeichert, wenn er nicht den Anforderungen entspricht. Da aber in diesem Fall der Wert allgemein ermittelt werden soll, bewirkt eine Änderung der Klasse `Property`, dass auch beim Bestehen der Prüfung ein Grund in folgender Form angelegt wird:

```
# Extension Patrick Loibl:
reason = {"type": "VALUE", "actual": props[pset_name][self.name]}
```

Zusätzlich dazu muss auch die Klasse `Result` geändert werden, damit `reason` mit einem Wert belegt ist und nicht nur mit `None` initialisiert wird.

```
def __init__(self, is_pass, reason={"type": None}): # Extension Patrick
    Loibl: Reason ist immer vorhanden
    self.is_pass = is_pass
    self.reason = reason
```

Somit kann von der Klasse *Property* zur Methode `propertyValueComparisonOfMultiple-Objects` (Code 4.17) zurückgekehrt werden. Bei Vorhandensein eines Property Value wird dieser nach der Rückgabe (`result_property`) in ein angelegtes Set `different_values` gespeichert (Zeile 11). Sollte ein Element keinen Wert besitzen, wird dies in einem Boolean-Wert festgelegt (Zeile 13). Im nächsten Schritt tritt nun die Unterscheidung durch den übergebenen Wert `same` in Verwendung. Der zweite Block der elif-Verzweigung (Zeilen 22–29) prüft, ob die Gleichheit der Property Values gegeben ist. Dies trifft zu, wenn das Set nur einen Wert enthält und kein Element ohne gefordertem Property Value existiert. Die Prüfung gilt hingegen als nicht bestanden, wenn mehrere Property Values vorhanden sind oder einem betroffenen Element der Property Value fehlt. Als Fehlergrund wird ein ausgewählter Satz gespeichert. Zusätzlich kann auch die Schleife über alle Facets abgebrochen werden. Schlussendlich wird ein Tupel als Ergebnis in die Methode `checkSpaceBoundary` (Code 4.16) zurückgegeben und die Prüfung ist beendet.

4.2.4.3 Verwendung von `differentPropertyValuesForAnyObject`

Der Ablauf dieses Blocks (Zeilen 15–22, Code 4.16) ist nahezu gleich zum vorherigen. Der erste Unterschied liegt in der Möglichkeit, einen Property Value in der IDS-Datei anzugeben. Daher wird in der ersten Schleife ein Property-Check durchgeführt. Bevor der nächste Schritt startet, muss mindestens ein Element diese Anforderung erfüllen. Danach wird die Methode `propertValueComparisonOfMultipleObjects` mit dem Wert `different` aufgerufen, damit ein möglicher Unterschied in den Property Values festgestellt werden kann. Im Code 4.17 wird somit der erste Block (Zeilen 14–21) der elif-Verzweigung eingeschlagen. Hier liegt der zweite

Unterschied und es wird nun auf ein Vorhandensein mehrerer Property Values überprüft. Dies tritt einerseits auf, wenn mehr als ein Wert im Set `different_values` vorhanden ist. Andererseits zählt auch als Unterschied, wenn ein Element ohne gefordertem Property Value und mindestens ein Element mit Wert vorhanden ist. Darauf aufbauend erfolgt wieder die Erstellung des jeweiligen Ergebnisses und die Rückgabe als Tupel.

Code 4.17: Methode `propertyValueComparisonOfMultipleObjects`

```
1    def propertyValueComparisonOfMultipleObjects(self, elements_to_check,
         value):
2        is_pass=False
3        elementsWithoutProperty = False
4        for facet in self.extendedSpecification:
5            if isinstance(facet,Property):
6                different_values = set()
7                for element in elements_to_check:
8                    result_property = facet(element)
9                    if result_property.reason["type"] == "VALUE":
10                       element_property_value = result_property.reason["actual"]
11                       different_values.add(element_property_value)
12                   else:
13                       elementsWithoutProperty = True
14               if value == "different":
15                   if (len(different_values) > 1) or (len(different_values) >=
                         1 and elementsWithoutProperty):
16                       is_pass = True
17                       reason = None
18                   else:
19                       is_pass = False
20                       reason = {"type": "ENTITY", "actual": "the entities have
                             no difference between their properties"}
21                       break
22               elif value == "same":
23                   if len(different_values) == 1 and not
                         elementsWithoutProperty:
24                       is_pass = True
25                       reason = None
26                   else:
27                       is_pass = False
28                       reason = {"type": "ENTITY", "actual": "the entities have
                             no same property"}
29                       break
30       return is_pass,reason
```

4.2.4.4 Verwendung von `differentObjectsForEachPropertyValue`

Das dritte Schlüsselwort und somit der letzte Block im Code 4.16 (Zeilen 23–38) umfasst die Überprüfung, ob es für jeden vorgegebenen Property Value mindestens ein Element gibt. Zuerst werden alle Property-Facets ermittelt und in einer neuen Liste gespeichert (Zeile 26). Dann startet eine Schleife sowohl über die Elemente als auch die Property-Facets. Genau hier werden die Elemente auf Vorhandensein eines geforderten Property Values geprüft. Trifft dies zu, wird das Property-Facet aus der Liste entfernt und mit den nächsten Elementen fortgesetzt. Sind beide Schleifen durchgelaufen, erfolgt eine Überprüfung, ob die Liste `properties_to_check` noch

Property-Facets beinhaltet. Ist dies nicht der Fall, existiert zu jedem geforderten Property Value mindestens ein Element mit dem passenden Wert und die Prüfung gilt somit als bestanden.

4.2.5 Zusätzliche eigene Methoden

Im bisherigen Programmablauf wurden drei wesentliche Funktionen in Methoden ausgelagert, die in diesem Abschnitt näher erklärt werden. Es handelt sich um die beiden ähnlichen Methoden `property_check` und `attribute_check` sowie die dritte Methode `partOf_recursion`.

Der Property- und Attribut-Check (siehe Code 4.18) werden jeweils mit dem aktuellen Objekt der Klasse `PartOf` aufgerufen und bekommen zusätzlich das zu überprüfende Element sowie `is_pass` und `reason` übergeben. Als erstes wird überprüft, ob alle Prüfungen bis jetzt bestanden wurden und das Objekt überhaupt ein Property- beziehungsweise Attribut-Facet besitzt. Ist ein Punkt davon nicht erfüllt, werden die übergebenen Variablen sofort wieder als Tupel zurückgegeben. Wenn es jedoch erfüllt ist, läuft die Prüfung weiter. In einer Schleife werden die einzelnen Facets mit dem jeweiligen Element aufgerufen und die Rückgabewerte gespeichert. Dies wird für alle Facets wiederholt, bis entweder ein Element die Prüfung nicht besteht oder die Schleife zu Ende ist. Dann wird das Ergebnis wieder als Tupel zurückgegeben. Anzumerken ist, dass die von den Facets erstellten Fehlergründe nicht verwendet werden können, da dies sonst zu Problemen bei der Ergebnisausgabe führt. Der Grund liegt in den möglichen Ergebnistypen, welche bei Property- und Attribut-Facets andere sind als beim PartOf-Facet. Dies wird durch einen eigenen erstellen Fehlergrund gelöst, der das Ergebnis auch besser beschreibt.

Code 4.18: Methoden `property_check` und `attribute_check` der Klasse `PartOf`

```
1   def property_check(self, element,is_pass,reason):
2       if is_pass and hasattr(self,"property") and self.property !=[]:
3           for facet in self.extendedSpecification:
4               if isinstance(facet,Property):
5                   result = facet(element)
6                   is_pass = result.is_pass
7                   reason = {"type": "ENTITY", "actual": "the entity has one or
                        more false properties"}
8                   if not is_pass:
9                       break
10          return is_pass,reason
11      else:
12          return is_pass,reason
13
14  def attribute_check(self, element,is_pass,reason):
15      if is_pass and hasattr(self,"attribute") and self.attribute !=[]:
16          for facet in self.extendedSpecification:
17              if isinstance(facet,Attribute):
18                  result = facet(element)
19                  is_pass = result.is_pass
20                  reason = {"type": "ENTITY", "actual": "the entity has one or
                        more false attributes"}
21                  if not is_pass:
22                      break
23          return is_pass,reason
24      else:
25          return is_pass,reason
```

Die Methode `partOf_recursion` im Code 4.19 ist eine Rekursion und dient zur Umsetzung der Verschachtelung der PartOf-Facets in IDS-Dateien. Der Aufruf und die übergebenen Variablen

sind ident zu den beiden vorhergehenden Methoden. Ebenso findet zuerst die Überprüfung von `is_pass` sowie des Vorhandenseins weiterer PartOf-Facets statt. Ist dies erfüllt, werden diese PartOf-Facets mit dem Element aufgerufen und damit die Rekursion gestartet. Das Ergebnis wird gespeichert und als Tupel zurückgegeben.

Code 4.19: Methode `partOf_recursion` der Klasse `PartOf`

```
1   def partOf_recursion(self,element,is_pass,reason):
2       if is_pass and hasattr(self,"partOf"):
3           for facet in self.extendedSpecification:
4               if isinstance(facet,PartOf):
5                   result = facet(element)
6                   is_pass = result.is_pass
7                   reason = result.reason
8               else:
9                   continue
10          return is_pass,reason
11      else:
12          return is_pass,reason
```

Diese drei behandelten Methoden werden bei allen vorhandenen Relationen aufgerufen (siehe Code 4.14). Die Verwendung bei der eingefügten Relation `IfcRelSpaceBoundary` wurde schon in den vorherigen Abschnitten erläutert. Bei den anderen Relationen wird zuerst der normale Ablauf der Prüfung durchgeführt. Anschließend werden die eingefügten Methoden in folgender Reihenfolge aufgerufen:

```
# Extension Patrick Loibl
(is_pass,reason) = self.attribute_check(parent,is_pass,reason)
(is_pass,reason) = self.property_check(parent,is_pass,reason)
(is_pass,reason) = self.recursion(parent,is_pass,reason)
```

4.2.6 Änderung der Ergebnisausgabe und -rückgabe

Zusätzlich zu den auf den Änderungen der XSD-Datei aufbauenden Erweiterungen wurde die Ergebnisausgabe beziehungsweise die Ergebniserstellung der Klasse `PropertyResult` angepasst. In der ursprünglichen Version wird bei den Gründen `NOPSET` und `NOVALUE` nur ausgegeben, dass das gesuchte Property Set oder das gewünschte Property nicht vorhanden ist. Abb. 4.6 zeigt dazu zwei Beispiele. Die Änderungen sollen in beiden Fällen auch die zugehörigen Namen ausgeben, um die Ergebnisse besser verständlich zu machen.

```
#7990=IfcDoor('2rL90XeUL4LudlL19YqQ6D',#12,'Tür-001',$,$,#7978,#7987,'B5549021-A1E5-4457-89EF-541262D1A18D',2.04,0.88,.DOOR.,$,$)
The property set does not contain the required property

#20478=IfcDoor('1dpLLUFiPAjhkkB1dP5k8K',#12,'Tür-006',$,$,#17723,#20469,'67CD555E-3EC6-4AB6-BBAE-2C19D916E214',2.04,0.88,.DOOR.,$,$)
The required property set does not exist
```

Abb. 4.6: Screenshot der ursprünglichen Ergebnisausgabe

Um die gewünschten Ergebnisse zu erreichen, wird zuerst die Erstellung der Gründe angepasst. Sowohl bei den Properties als auch bei den Property Sets wird nun deren Name mit angelegt. Beide folgenden Änderungen sind in der Methode `__call__` der Klasse `Property` zu finden.

```
reason = {"type": "NOPSET", "actual": self.propertySet} # Extension
    Patrick Loibl: Name des gesuchten Property Set wird angegeben
```

```
reason = {"type": "NOVALUE", "actual": self.name} # Extension Patrick
    Loibl: Name des gesuchten Property wird angegeben
```

Auch die im Code 4.20 dargestellte Klasse `PropertyResult` muss angepasst werden, damit bei den ersten zwei Gründen der hinzugefügte Wert `self.reason['actual']` mit ausgegeben wird.

Code 4.20: Klasse `PropertyResult` der Datei *facet.py*

```
1   class PropertyResult(Result):
2       def to_string(self):
3           if self.reason["type"] == "NOPSET":
4               return f"The required property set \"{str(self.reason['actual
                    '])}\" does not exist" # Extension Patrick Loibl: Name des
                    Property Set wird mit ausgegebenen
5           elif self.reason["type"] == "NOVALUE":
6               return f"The property set does not contain the required
                    property \"{str(self.reason['actual'])}\"" # Extension
                    Patrick Loibl: Name des Property wird mit ausgegeben
7           elif self.reason["type"] == "DATATYPE":
8               return f"The data type \"{str(self.reason['actual'])}\" does
                    not match the requirements"
9           elif self.reason["type"] == "VALUE":
10              if isinstance(self.reason["actual"], list):
11                  if len(self.reason["actual"]) == 1:
12                      return f"The property value \"{str(self.reason['actual
                            '][0])}\" does not match the requirements"
13                  else:
14                      return f"The property values \"{str(self.reason['actual
                            '])}\" do not match the requirements"
15              else:
16                  return f"The property value \"{str(self.reason['actual'])}\"
                         does not match the requirements"
17          elif self.reason["type"] == "PROHIBITED":
18              return f"The property should not have met the requirement"
```

Schlussendlich wird das gewünschte Ergebnis wie in Abb. 4.7 ausgegeben. Eine genauere Beschreibung der Ergebnisausgabe findet in folgendem Abschnitt statt.

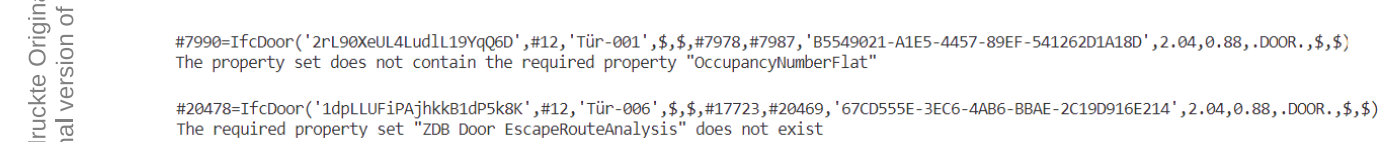

Abb. 4.7: Screenshot der neuen Ergebnisausgabe

4.3 Gesamter Programmablauf inklusive Erweiterung

In diesem Abschnitt wird ein gesamter Programmablauf vom Aufruf der notwendigen Dateien bis zur Ergebnisausgabe beschrieben. Hierbei werden das selbst programmierte Skript zur Verwendung von *IfcOpenShell* sowie die beiden wesentlichen Schritte, das Öffnen der IDS-Datei sowie die Prüfung der IFC-Datei, näher betrachtet. In weiterer Folge wird in diesem Abschnitt und allen folgenden Kapiteln das eigene Skript und *IfcOpenShell* mit den gesamten Erweiterungen als *AdvancedIfcTester* bezeichnet, um für Klarheit bei Nennung der Programme zu sorgen.

4.3.1 Verwendung eines eigenen Skripts

Um *IfcOpenShell* zu benutzen, muss eine eigenes *Python*-Skript in *Visual Studio Code* angelegt werden. Dessen gesamter Inhalt ist im Code 4.21 enthalten. Am Anfang werden die beiden benötigten Module geladen. Diese sind die allgemeine *IfcOpenShell* mittels `import ifcopenshell` sowie die speziellen Funktionen des *IfcTester* aus der Datei *ids.py* mittels `import ifcopenshell.ids as ids`.

Danach folgt das Öffnen einer gewählten IDS-Datei. Dafür wird die Methode `open` aus der Datei *ids.py* mit dem Pfad der zu öffnenden IDS-Datei an erster Stelle aufgerufen. Als zweites wird der Boolean-Wert `True` übergeben. Dieser führt zu Beginn der Methode dazu, dass vor der Erstellung eines Objekts der Klasse `IDS` direkt eine Überprüfung auf Einhaltung der gegebenen XSD-Datei durchgeführt wird. Wenn der Wert `False` beträgt, wird die Validierung erst zu einem späteren Zeitpunkt abgehandelt. Das Ergebnis bei einer falschen Syntax führt in beiden Fällen zum Abbruch der Prüfung. Nach dem Öffnen der Datei erstellt die Methode `open` im weiteren Verlauf ein Objekt der Klasse `IDS` und ordnet diesem den Inhalt der IDS-Datei als Objektvariable zu. Schließlich wird das Objekt in der Variable `ids_file` gespeichert. Eine detailliertere Beschreibung des Ablaufs der Erstellung zeigt der nächste Abschnitt.

Nach einigen Zeilen zur Formatierung der Ausgabe findet der zweite Schritt statt. In diesem erfolgt das Öffnen der gewählten IFC-Datei. Hier wird nicht die Datei *ids.py*, sondern die allgemeine Funktionsweise von *IfcOpenShell* benötigt. Es startet die dort enthaltene Methode `open` durch Aufruf mit dem Pfad zur gewünschten Datei. Eine Speicherung erfolgt in der Variable `ifc_file`.

Im dritten Schritt findet die eigentliche Prüfung statt. Ausgehenden von der IDS-Datei wird die Methode `validate` mit der zu prüfenden IFC-Datei aufgerufen. Näheres dazu wird im dritten Abschnitt noch genannt. Nach Abschluss der Prüfung sollen die Ergebnisse ausgegeben werden. Hier ist viel Freiraum mit unterschiedlichen Möglichkeiten gegeben. In dieser Arbeit wurde folgende Ausgabe gewählt.

Zuerst wird der Name der jeweiligen Spezifikation aus der IDS-Datei ausgegeben. Danach erfolgt eine Auflistung der Entitäten, welche die Applicability erfüllen und auf die die Anforderungen anwendbar sind. Zusätzlich werden noch die gesamte Stückzahl und die Entitäten ausgegeben, welche die Anforderungen nicht erfüllen. Dazu wird auch jeweils der passende Grund genannt. Ein Beispiel dafür war in Abb. 4.7 zu sehen. In Kapitel 5 werden noch mehrere Ergebnisausgaben gezeigt. Als Ausgabe einer Entität dient die jeweilige Zeile aus der IFC-Datei im STEP Physical File Format. Wie in Eichler et al. [9] beschrieben, steht zu Beginn der jeweilige dateiinterne Identifikator in Form einer Zahl mit davorstehendem Rautezeichen. Danach folgt die Angabe der IFC-Klasse. In der Klammer wird die jeweilige GUID angeführt, sowie weitere Informationen und Verweise auf andere Objekte genannt. Eine genauere Erklärung der einzelnen Inhalte ist in Eichler et al. [9] zu finden.

Code 4.21: Inhalt der Datei zur Verwendung der IfcOpenShell

```
1   import ifcopenshell
2   import ifcopenshell.ids as ids
3
4   # Erster Schritt: Öffnen einer gewünschten IDS-Datei
5   ids_file = ids.open('./idsFiles/20230809
        _LOI_EscapeRouteAnalysis_ifcOpenShell.ids',True)
6
7   print("Opening file: Success")
8   print()
9   print("Checking")
```

```
10
11      # Zweiter Schritt: Öffnen der gewünschten IFC-Datei
12      ifc_file = ifcopenshell.open('./ifcFiles/LOI_Fluchtwegsanalyse_NEU.ifc')
13
14      # Dritter Schritt: Validierung ausführen
15      ids_file.validate(ifc_file)
16
17      print()
18      print("Checking file: Success")
19
20      for spec in ids_file.specifications:
21          print()
22          print()
23          print(spec.name)
24          print()
25          print("Applicable entities:" , len(spec.applicable_entities))
26          a=0
27          for a in range(0,len(spec.applicable_entities)):
28              print(spec.applicable_entities[a])
29              a+=1
30          print()
31          print("Failed entities:" , len(spec.failed_entities))
32          f=0
33          for r in range(0,len(spec.requirements)):
34              for f in range(0,len(spec.requirements[r].failed_entities)):
35                  print()
36                  print(list(spec.requirements[r].failed_entities)[f])
37                  print(list(spec.requirements[r].failed_reasons)[f])
38                  f+=1
39              r+=1
```

4.3.2 Methode open der Datei ids.py

Dieser Abschnitt erläutert die Zuordnung der Attribute der IDS-Datei zum Objekt der Klasse `IDS` durch die Methode `open` der Datei *ids.py*. Nach der Validierung zur XSD-Datei wird die Methode `parse` aufgerufen. Diese bekommt die dekodierte Information aus der IDS-Datei als Dictionary übergeben. In der Methode werden dem Objekt zuerst einige Metadaten zugewiesen, bevor eine Schleife über die Spezifikationen beginnt. Innerhalb dieser wird jeweils ein Objekt der Klasse `Specification` erzeugt. Ausgehend von diesem erfolgt ein Aufruf der Methode `parse` (siehe Code 4.11 im Abschnitt 4.2.2, Seite 46). Darin erfolgt die Zuordnung der Informationen bezüglich der Applicability und der Requirements. Nach Ablauf der Schleife über alle Spezifikationen ist das Objekt fertig angelegt.

4.3.3 Methoden validate der Klassen IDS und Specification

Ziel dieses Abschnitts ist die detailliertere Erklärung der Validierung mit speziellem Blick auf die Erweiterungen. Ausgehend von der geöffneten IDS-Datei wird die Methode `validate` aus dem Code 4.22 mit der IFC-Datei aufgerufen (siehe Zeile 15, Code 4.21). Innerhalb der Methode läuft eine Schleife über alle enthaltenen Spezifikationen. Nun wird ausgehend von diesen Spezifikationen die gleichnamige Methode `validate` der Klasse `Specification` ebenfalls mit der IFC-Datei gestartet. Der Beginn dieser ist im Code 4.23 angeführt.

Die Methode startet eine Schleife über alle in der Applicability der IDS-Datei enthaltenen Facets (Zeile 8). Ziel jener ist die Ermittlung der richtigen Elemente, auf die die Anforderungen danach angewendet werden können. Dafür wird die Methode `filter` des jeweiligen Facets mit

Code 4.22: Methode `validate` der Klasse `IDS`

```
1     def validate(self, ifc_file, filter_version=False):
2       for specification in self.specifications:
3          specification.reset_status()
4          specification.validate(ifc_file, filter_version=filter_version)
```

der IFC-Datei sowie einem Set von Elementen aufgerufen. Die Reihenfolge basiert hier auf der Anordnung in der IDS-Datei. Im ersten Schleifendurchlauf ist das Set noch ohne Elemente, wodurch beim ersten Facet alle in der IFC-Datei enthaltenen Objekte in Frage kommen. Nach der Filterung wird das Set mit denjenigen Elementen befüllt, welche die erste Anforderung erfüllen. Im zweiten Schleifendurchgang bekommt das nächste Facet somit nur mehr eine begrenzte Anzahl an zu prüfenden Objekten. So nähert man sich Schritt für Schritt dem Ende und erhält die Elemente, welche die gesamte Applicability erfüllen.

Code 4.23: Beginn der Methode `validate` der Klasse `Specification`

```
1   def validate(self, ifc_file, filter_version=False):
2      if filter_version and ifc_file.schema not in self.ifcVersion:
3         return
4
5      elements = None
6      # This is a broadphase filter of applicability. We almost never want
7      # to test every single class in an IFC model.
8      for i, facet in enumerate(self.applicability):
9         elements = facet.filter(ifc_file, elements)
10
11     for element in elements or []:
12        is_applicable = True
13        for facet in self.applicability:
14           if isinstance(facet, Entity):
15              continue
16           if not bool(facet(element)):
17              is_applicable = False
18              break
19        if not is_applicable:
20           continue
21        self.applicable_entities.append(element)
22        for facet in self.requirements:
23           result = facet(element)
24           if not bool(result):
25              self.failed_entities.add(element)
26              facet.failed_entities.append(element)
27              facet.failed_reasons.append(str(result))
```

Beim Aufruf der Methode `filter` (Zeile 9) wechselt man von der Datei *ids.py* in die Datei *facet.py* zur Klasse des jeweiligen Facets. So unterschiedlich diese sind, so unterschiedlich ist auch der Ablauf der jeweiligen Methode. Die Filterung nach der Entität funktioniert sehr schnell, da hier die IFC-Datei einfach nach der übergebenen Klasse gesucht wird. Beim Aufruf des Property-Facets dauert der Ablauf schon etwas länger und ist auch differenzierter. So wird zuerst

die Existenz des richtigen Property Sets geprüft. Danach wird das angegebene Property gesucht. Wenn zusätzlich ein Property Value angegeben ist, läuft die Prüfung weiter. Hierbei wird noch zwischen den verschiedenen Datentypen unterschieden, bis ein Element fertig überprüft ist.

Der Ablauf des PartOf-Facets bildet inklusive der Erweiterungen den umfangreichsten Bereich. Ohne auf die zusätzlichen Methoden einzugehen, wird in der ursprünglichen Version je nach gegebener Relation das entsprechende Element abgerufen. Ist keines vorhanden, wird dies als Fehlergrund gespeichert. Wenn ein Element existiert, findet als nächstes eine Überprüfung der Entität statt. Hier entscheidet sich in der Originalversion, ob die Prüfung bestanden ist oder nicht. Durch die Erweiterung um die drei eigenen Methoden `property_check`, `attribute_check` und `partOf_recursion` wird die endgültige Entscheidung noch von diesen abhängig gemacht. Abb. 4.8 stellt den erweiterten Ablauf der ursprünglich vorhandenen Relationen dar. Nun wird ersichtlich, dass die Erweiterung mit der Relation *IfcRelSpaceBoundary* einen sehr ähnlichen Ablauf hat, was auch ein wesentliches Ziel darstellte. Wie in Abb. 4.4 (Seite 51) zu sehen war, findet auch am Beginn ein Abruf der Elemente sowie eine Prüfung der Entität statt. Die drei weiteren Methoden finden erst in `checkSpaceBoundary` Anwendung.

Sind alle Facets in der Applicability abgearbeitet, gibt es schlussendlich ein erstes Set an Elementen. Über diese läuft im nächsten Schritt noch eine Schleife (Zeile 11). In dieser werden bis auf das Entity-Facet alle Facets der Applicability mit dem jeweiligen Element nochmal aufgerufen (Zeile 16). Dies stellt eine Redundanz her, damit kein nicht anwendbares Element verwendet wird. Ist diese zweite Prüfung bestanden, folgt eine Schleife über alle Requirements (Zeile 22). Darin findet der Aufruf der Facets der Requirements und somit die Überprüfung der Anforderungen statt. Wird eine davon nicht bestanden, wird das jeweilige Element mit zugehörigem Fehlergrund abgespeichert. Die gesamte Validierung endet, sobald die Schleifen über die Facets sowie die Elemente abgelaufen sind.

Eine direkte Rückgabe mit `return` erfolgt nicht, da die Ergebnisse im jeweiligen Facet abgespeichert sind. Darauf wird zur Ergebnisausgabe mit dem eigenen Skript aus dem Abschnitt 4.3.1 zugegriffen. Anzumerken ist, dass bei den Elementen, welche die Applicability nicht bestehen, zwar ein Fehlergrund erstellt wird, dieser aber nirgendwo gespeichert wird. Ebenso erfolgt keine Speicherung der nicht verwendbaren Elemente. Dies würde nämlich ins Unermessliche führen, da dies bis auf ein paar gewünschte beinahe alle Objekte der IFC-Datei betreffen würde.

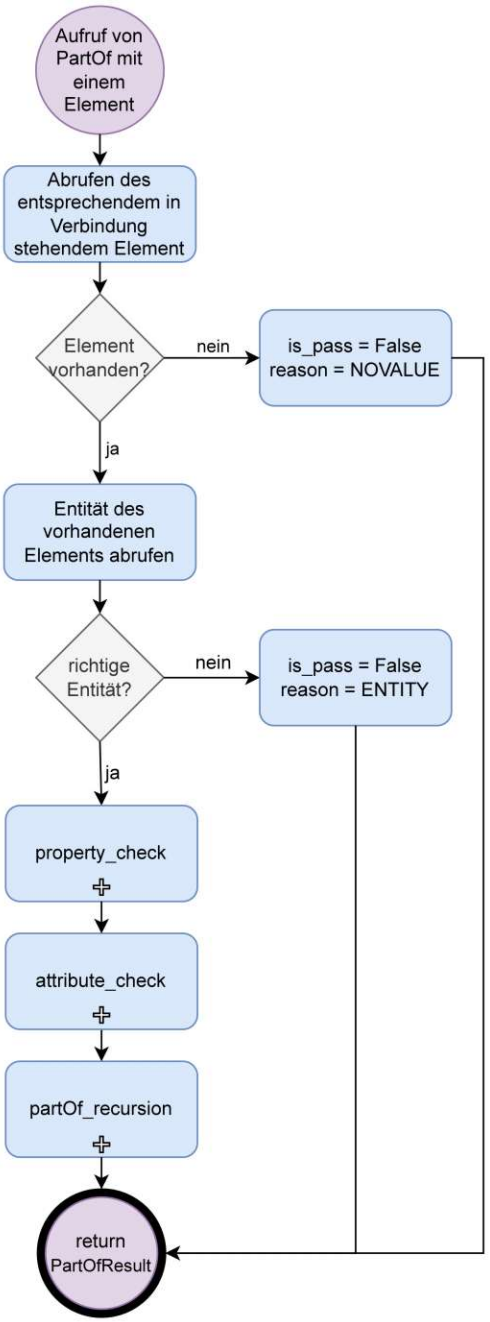

Abb. 4.8: Ablauf der Klasse `PartOf` beim Aufruf mit einem Element

Kapitel 5

Anwendung der Erweiterungen

Dieses Kapitel beinhaltet die Anwendung der durch die Erweiterungen möglich gewordenen Spezifikationen für den LOI-Check der Fluchtwegsanalyse sowie der behandelten Tabellen der OIB-Richtlinie 2. In jedem Abschnitt wird zuerst die Erstellung der einzelnen IDS-Dokumente beschrieben, danach erfolgt eine Anwendung und Testung mit dem eigens geschriebenen Programm *AdvancedIfcTester*.

Die Erstellung der verwendeten IDS-Dateien erfolgte mit dem Programm *Visual Studio Code*. Damit ist eine Erstellung von XML-Dokumenten im Vergleich zu einem einfachen Texteditor leichter möglich. Ausschnitte aus den IDS-Dateien dienen im Laufe des Kapitels zum besseren Verständnis. Alle erstellten IDS-Dateien befinden sich im Anhang. Bei der Beschreibung wird besonderes Augenmerk auf die korrekte Verwendung der Erweiterungen gelegt.

Bei der Anwendung und Testung mit dem *AdvancedIfcTester* wird jeweils für den LOI-Check und die OIB-Richtlinie 2 ein eigens erstelltes Testmodell verwendet. Die Anwendung von Standardprogrammen (z. B. *usBIM.IDS*) ist aufgrund der veränderten XSD-Datei und der fehlenden Funktionen nicht mehr möglich. Im Abschnitt über den LOI-Check werden die Ausgabe und der Ablauf der Testung nochmals kurz beschrieben. In den darauffolgenden Abschnitten über die OIB-Richtlinie 2 wird vor allem auf die Besonderheiten der Erweiterungen eingegangen.

5.1 Umsetzung des LOI-Check für die Fluchtwegsanalyse

Mit der Standardversion von IDS können beim LOI-Check schon sechs der sieben geforderten Properties abgedeckt werden. Die Erweiterungen ermöglichen nun auch eine Erstellung zur Prüfung von *OccupancyNumberFlat*. Die Einführung der Relation *IfcRelSpaceBoundary* sowie die Möglichkeit, für verbundene Elemente Properties vorzuschreiben, helfen bei der Umsetzung. Im Anhang B ist die erstellte IDS-Datei mit allen enthaltenen Spezifikationen angeführt. Der Inhalt wurde zur Überprüfung mit dem *AdvancedIfcTester* erstellt und ist aufgrund der Erweiterung nicht mehr mit *usBIM.IDS* verwendbar.

Die entsprechende Spezifikation für das Property *OccupancyNumberFlat*, in welcher die Erweiterungen Anwendung finden, ist zur besseren Übersicht zusätzlich auch im Code 5.1 dargestellt. In der ersten Stufe der Applicability befinden sich ein Entity-, ein PartOf- und ein Property-Facet. Ersteres gibt die Entität `IFCDOOR` an. Das zweite Facet dient zur Beschreibung des in Verbindung stehenden Elements, welches ein Raum sein muss. Hier wird mit der Relation `IFCRELSPACEBOUNDARY` bereits die erste Erweiterung benutzt. In der zweiten Stufe des PartOf-Facets sind ein Entity- und ein Property-Facet vorhanden. Zuerst wird die Entität `IFCSPACE` angeführt. Danach wird das vom Raum geforderte Property `WidmungBehoerde` mit dem einzuhaltenden Wert angeführt. Mit der hier verwendeten Konkretisierung findet die zweite Erweiterung Anwendung. Nochmals ist zu betonen, dass jede Tür nur einen verbundenen Raum mit der entsprechenden Widmung haben muss und nicht alle Räume den angegebenen Wert `Wohnen und Aufenthalt` besitzen müssen. Das dritte Facet der ersten Stufe behandelt den Türtyp und schreibt eine Eingangstür (`Entry`) vor.

Code 5.1: Spezifikation für das Property *OccupancyNumberFlat*

```
<specification minOccurs="0" name="Anforderungen IfcDoor Eingangstür (OccupanyNumberFlat)" ifcVersion="IFC4">
  <applicability>
    <entity>
      <name>
        <simpleValue>IFCDOOR</simple>
      </name>
    </entity>
    <partOf relation="IFCRELSPACEBOUNDARY">
      <entity>
        <name>
          <simpleValue>IFCSPACE</simpleValue>
        </name>
      </entity>
      <property datatype="IfcLabel" >
        <propertySet>
          <simpleValue>Pset_SpaceSpecific</simpleValue>
        </propertySet>
        <name>
          <simpleValue>WidmungBehoerde</simpleValue>
        </name>
        <value>
          <simpleValue>Wohnen und Aufenthalt</simpleValue>
        </value>
      </property>
    </partOf>
    <property datatype="IfcLabel" >
      <propertySet>
        <simpleValue>ZDB_Door_EscapeRouteAnalysis</simpleValue>
      </propertySet>
      <name>
        <simpleValue>DoorType</simpleValue>
      </name>
      <value>
        <simpleValue>Entry</simpleValue>
      </value>
    </property>
  </applicability>
  <requirements>
    <property datatype="IfcCountMeasure" minOccurs="1" maxOccurs="1">
      <propertySet>
        <simpleValue>ZDB_Door_EscapeRouteAnalysis</simpleValue>
      </propertySet>
      <name>
        <simpleValue>OccupancyNumberFlat</simpleValue>
      </name>
      <value>
        <xs:restriction base="xs:integer">
          <xs:minExclusive value="0" />
        </xs:restriction>
      </value>
      </property>
  </requirements>
</specification>
```

In den Requirements ist ein Property-Facet angegeben, welches das geforderte Property `OccupancyNumberFlat` prüft. Dieses besitzt eine Einschränkung auf Werte größer `0`. Wie ersicht-

lich ist, kann die gesamte Spezifikation in wenigen Zeilen angegeben werden. Die verwendeten Erweiterungen passen sich der Struktur der Standardversion entsprechend an.

Mit der Erstellung der Spezifikation für das letzte Property umfasst der LOI-Check für die Fluchtwegsanalyse nun insgesamt alle sieben Properties. Sechs Properties waren bereits ohne Erweiterung prüfbar, was beispielhaft anhand des Property *ParkingSlot* in Kapitel 3 demonstriert wurde. In diesem Abschnitt wird nun die Anwendung des *AdvancedIfcTester* für eben dieses Property sowie die Funktionsfähigkeit der Erweiterung anhand des siebten Property *OccupancyNumberFlat* gezeigt. Als Testobjekt für den LOI-Check wurde mit *ArchiCAD 26* ein einstöckiges Gebäude als BIM-Modell erstellt, welches eine Wohnung widerspiegeln soll (siehe Abb. 5.1). Insgesamt gibt es sieben Räume mit unterschiedlichen Widmungen sowie acht Türen. Weiters sind entsprechende Wände und zwei Decken vorhanden. Auf weitere für den LOI-Check nicht relevante Objekte wurde verzichtet.

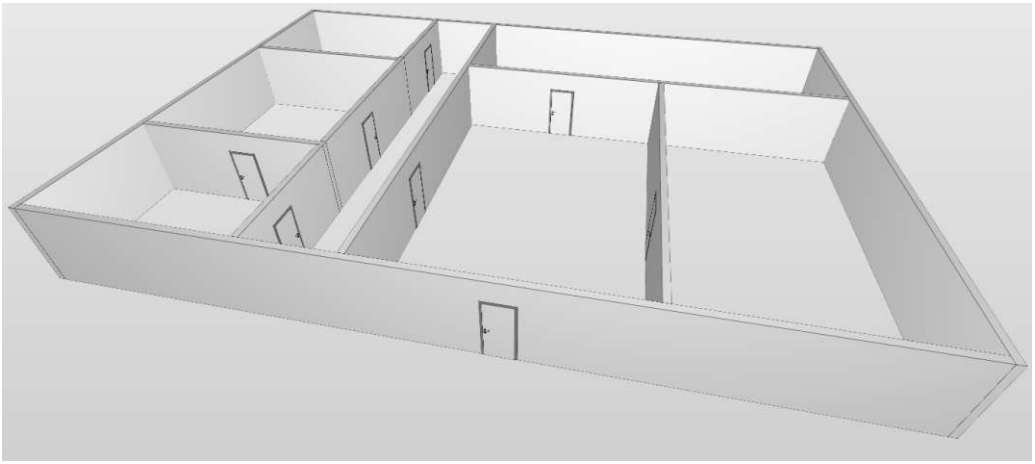

Abb. 5.1: Axonometrie des Testmodells für den LOI-Check

Für den LOI-Check sind die Widmungen der einzelnen Räume von Bedeutung, genauer gesagt das Property *WidmungBehoerde*, weshalb Abb. 5.2 die jeweiligen Property Values zur Übersicht angibt. Der rechte untere Raum besitzt das Property und das zugehörige Property Set *Pset_SpaceSpecific* nicht.

Das Property *ParkingSlot* wird von allen Räumen gefordert, welche die Widmung *Verkehrserschliessung und -sicherung* besitzen. Dies entspricht den beiden mittleren Räumen des Modells. Die Ausgabe der Ergebnisse des *AdvancedIfcTester* erfolgt in der Konsole von *Visual Studio Code*. Zuerst erfolgt eine Ausgabe des Namens der Spezifikation. Danach werden die einzelnen Elemente angeführt, welche die Applicability erfüllen. In diesem Fall sind das die beiden genannten Räume. Zur besseren Übersicht, vor allem bei großen Modellen, ist auch direkt die Gesamtanzahl der Elemente ersichtlich. Einer der beiden Räume besitzt das Property Set *ZDB_Space_EscapeRouteAnalysis* nicht und somit auch das darin enthaltene und geforderte Property *ParkingSlot* nicht. Der *AdvancedIfcTester* gibt diese Tatsache als Ergebnis zum entsprechenden Element aus. In den Requirements wird eigentlich das Property gefordert, jedoch erfolgt eine Ausgabe des nicht vorhandene Property Set als Fehler. Das liegt daran, dass das fehlende Property Set die Existenz des Property ausschließt. Es könnte zufälligerweise ein Property mit dem richtigen Namen geben, da dieses dann aber nicht im richtigen Property Set sein kann, liegt der Fehler wie beschrieben im nicht vorhandenen Property Set. Schlussendlich gleicht das Ergebnis des *AdvancedIfcTester* dem von *usBIM.IDS*.

Das siebte (und somit letzte) geforderte Property *OccupancyNumberFlat* wird von allen Eingangstüren (*DoorType = Entry*) gefordert, welche in einen Wohnraum (*WidmungBehoerde*

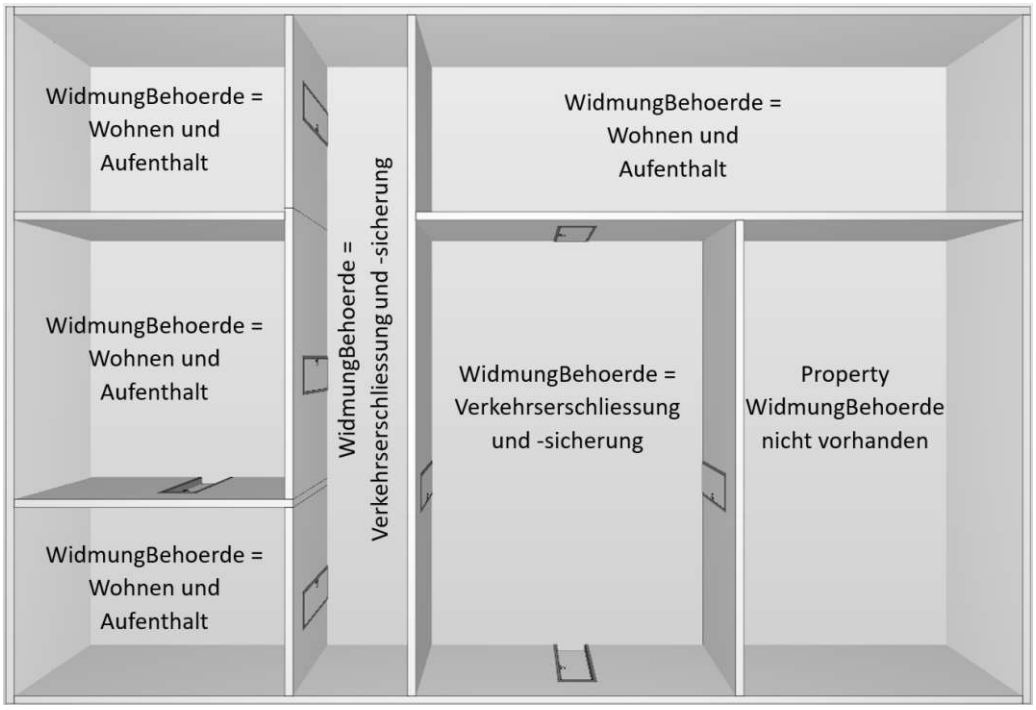

Abb. 5.2: Widmungen der Räume des Testmodells für den LOI-Check

```
Anforderung IfcSpace ParkingSlot

Applicable entities: 2
#27628=IfcSpace('22_WTqYQ55TBitxs79iJNw',#12,'4',$,$,#27574,#27624,'Gang',.ELEMENT.,.NOTDEFINED.,$)
#31499=IfcSpace('32qMaYoLv0exlpIb9ybPHF',#12,'<Raumnummer>',$,$,#31445,#31495,'<Raumname>',.ELEMENT.,.NOTDEFINED.,$)

Failed entities: 1

#27628=IfcSpace('22_WTqYQ55TBitxs79iJNw',#12,'4',$,$,#27574,#27624,'Gang',.ELEMENT.,.NOTDEFINED.,$)
The required property set "ZDB_Space_EscapeRouteAnalysis" does not exist
```

Abb. 5.3: Ergebnis des Property *ParkingSlot*

= *Wohnen und Aufenthalt*) führen. Mit dieser Spezifikation wird nun die Erweiterung von *IfcOpenShell* um die Relation *IfcRelSpaceBoundary* und die Spezifizierung des verbundenen Raums getestet. Die Spezifikation handelt von Türen, weshalb die Prüfung auch von diesen ausgeht. Im ersten Schritt der Applicability wird das Entity-Facet abgearbeitet. Dabei werden alle Türen des Modells ermittelt (hier: 8 Stück). Der zweite Schritt kümmert sich um das PartOf-Facet. Durch dieses werden aus den vorher erhaltenen Türen jene bestimmt, welche zu einem Raum mit der richtigen Widmung verbunden sind. Dies ist bei vier Räumen aus Abb. 5.2 der Fall. Somit werden die fünf angrenzenden Türen weiter untersucht. Der letzte Schritt der Applicability bestimmt noch deren Typ (*DoorType*). Schließlich entsprechen drei Türen dem Anwendungsbereich (*Entry*) und werden im Ergebnis ausgegeben (siehe Abb. 5.4). Die Prüfung der Requirements ist unverändert zur Standardversion abgelaufen.

Die erhaltenen Ergebnisse entsprechen den Erwartungen. Bei der Modellerstellung wurden absichtlich verschiedene Fälle eingebaut, damit nicht alle Räume beziehungsweise Türen die Applicability erfüllen. Somit können die Elemente entsprechend gefiltert und damit die Funktionsweise von Teilen der Erweiterung gezeigt werden.

```
Anforderungen IfcDoor Eingangstür (OccupanyNumberFlat)

Applicable entities: 3
#7990=IfcDoor('2rL90XeUL4LudlL19YqQ6D',#12,'Tür-001',$,$,#7978,#7987,'B5549021-A1E5-4457-89EF-541262D1A18D',2.04,0.88,.DOOR.,$,$)
#9665=IfcDoor('1bkMaXZtr5QxtqRoIpKXKw',#12,'Tür-008',$,$,#9653,#9662,'65B96921-8F7D-456B-BDF4-6F24B352153A',2.04,0.88,.DOOR.,$,$)
#36556=IfcDoor('3Cgxi8RWT539eokn3qhoxI',#12,'Tür-003',$,$,#33785,#36547,'CCABBB08-6E07-450C-9A32-BB10F4AF2ED2',2.04,0.88,.DOOR.,$,$)

Failed entities: 2

#7990=IfcDoor('2rL90XeUL4LudlL19YqQ6D',#12,'Tür-001',$,$,#7978,#7987,'B5549021-A1E5-4457-89EF-541262D1A18D',2.04,0.88,.DOOR.,$,$)
The property set does not contain the required property "OccupancyNumberFlat"

#9665=IfcDoor('1bkMaXZtr5QxtqRoIpKXKw',#12,'Tür-008',$,$,#9653,#9662,'65B96921-8F7D-456B-BDF4-6F24B352153A',2.04,0.88,.DOOR.,$,$)
The property value "-2.0" does not match the requirements
```

Abb. 5.4: Ergebnis des Property *OccupancyNumberFlat*

5.2 Umsetzung der Anforderungen an den Feuerwiderstand von Bauteilen

Mit den Erweiterungen kann eine Prüfung von IFC-Dateien auf Einhaltung der Tabelle 1b zum großen Teil durchgeführt werden. Mit der Standardversion von IDS wäre nichts davon möglich gewesen. Insgesamt werden die möglichen Punkte in den folgenden elf Spezifikationen aus Tab. 5.1 umgesetzt. In dieser sind jeweils die Nummern aus Tabelle 1b der OIB-Richtlinie 2, die betreffenden Entitäten und die verwendeten Erweiterungen angeführt. Für die ersten beiden Punkte aus Tab. 5.1 gibt es je Entität eine eigene Spezifikation. Alle umgesetzten Anforderungen beinhalten die Verschachtelung von PartOf-Facets sowie die Angabe von Properties und Attributen der verbundenen Elemente. Teilweise finden auch die Relation *IfcRelSpaceBoundary* und der neue Tag *multiObjectProcessing* Verwendung.

Bei der Beschreibung der Erstellung der jeweiligen Spezifikationen wird vor allem auf die Besonderheiten der Erweiterungen näher eingegangen. Um die Erklärung verständlich zu gestalten, werden in weiterer Folge zwei Ausschnitte aus der erstellten IDS-Datei als Anwendungsbeispiele gezeigt. Die Spezifikationen ohne wesentliche Besonderheiten werden im Gegenzug nur kurz beschrieben. Die gesamte IDS-Datei mit den elf genannten Spezifikationen ist im Anhang C angeführt. Die einzelnen Spezifikationen wurden jeweils nur für die Gebäudeklasse 2 erstellt, können jedoch analog für die weiteren Klassen geschrieben werden.

Es wurde ein zu dem BIM-Modell vom LOI-Check ähnliches Modell erstellt, um die Testung der Erweiterung durchführen zu können. Diesmal besteht es aus einem Untergeschoß, Erdgeschoß und einem Obergeschoß. Die einzelnen Grundrisse sind alle ident mit dem vorherigen Modell. Zusätzlich gibt es nun mehrere Treppenläufe, die vom Untergeschoß ins Erdgeschoß beziehungsweise vom Erdgeschoß ins Obergeschoß führen. Das Gebäude (*IfcBuilding*) des BIM-Modells hat passend zu den Spezifikationen im Property Set *ZDB_Building_Infos* die Gebäudeklasse *GK2* hinterlegt (siehe Abb. 5.5), unabhängig davon, dass es in der Realität nicht den dahingehenden Anforderungen der OIB-Richtlinie 1 entspricht. Es wurde jedoch bewusst die Gebäudeklasse 2

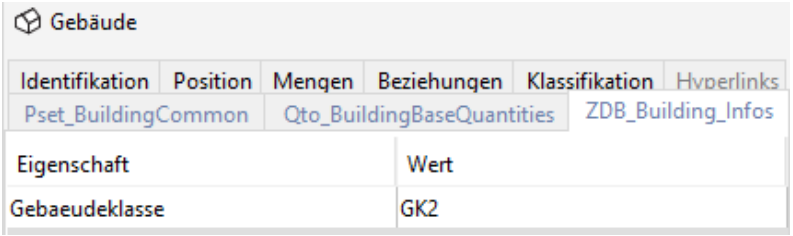

Abb. 5.5: Property *Gebauedeklasse* des Gebäudes (*IfcBuilding*)

Tab. 5.1: Spezifikationen für Tabelle 1b der OIB-Richtlinie 2

Anforderung	Entität	verwendete Erweiterungen
1.2 tragende Bauteile oberirdisch	Wände, Träger, Stützen	Verschachtelung, Properties und Attribute für verbundene Elemente
1.3 tragende Bauteile unterirdisch	Wände, Träger, Stützen	Verschachtelung, Properties und Attribute für verbundene Elemente
2.2 Trennwände oberirdisch	Wände	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, differentPropertyValuesForAnyObject
2.3 Trennwände unterirdisch	Wände	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, differentPropertyValuesForAnyObject
4.3 Trenndecken oberirdisch	Decken	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, differentPropertyValuesForAnyObject
4.4 Decken innerhalb von Wohnungen	Decken	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, samePropertyValuesForAllObjects
4.5 Decken unterirdisch	Decken	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente

gewählt, da mit einer niedrigeren Klasse einige Anforderungen laut der OIB-Richtlinie 2 nicht zu testen wären.

Die ersten Spezifikationen der Tabelle 1b sind insgesamt sechs Stück (je drei Stück für Punkt 1.2 und Punkt 1.3) und handeln von tragenden Bauteilen in ober- und unterirdischen Geschoßen. Anhand dieser Spezifikationen erfolgt die Beschreibung der Ermittlung von ober- und unterirdischen Bauteilen sowie der Bestimmung der Gebäudeklasse. Code 5.2 zeigt dafür die beispielhafte Umsetzung in der Applicability einer Spezifikation für Punkt 1.2. Konkret soll eine tragende Wand betrachtet werden, die in einem oberirdischen Geschoß und in einem Gebäude der Gebäudeklasse 2 liegt. Für den ersten Schritt wird die Relation `IFCRELCONTAINEDINSPATIALSTRUCTURE` verwendet, welche von der Wand auf das Geschoß schließt. Dieses wird durch das angegebene Attribut `Elevation` näher beschrieben und muss oberirdisch liegen. Von dem Geschoß wird mit einem weiteren PartOf-Facet mit Hilfe der Relation `IFCRELAGGREGATES` auf das Gebäude geschlossen. Ein Property-Facet fordert von diesem die richtige Gebäudeklasse. Damit ist die Verschachtelung abgeschlossen. Nach dem PartOf-Facet der ersten Stufe folgt noch ein Property-Facet, welches von der Wand die Eigenschaft tragend zu sein fordert. An dieser Stelle ist anzumerken, dass das verwendete Attribut `Elevation` im IFC-Schema nur optional ist. Dieses muss somit in der BIM-Autorensoftware bei der Modellierung extra angelegt werden. Ebenfalls ist darauf hinzuweisen, dass in IFC4.3 das Attribut nicht mehr vorhanden sein wird. Stattdessen können hier die beiden Properties *ElevationOfSSLRelative* und *ElevationOfFFLRelative* verwendet werden.

Code 5.2: Applicability der Spezifikation für tragende oberirdische Wände (Unterpunkt 1.2.1, Tabelle 1b, OIB 2)

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCWALL</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
      <entity>
         <name>
            <simpleValue>IFCBUILDINGSTOREY</simpleValue>
         </name>
      </entity>
      <partOf relation="IFCRELAGGREGATES">
         <entity>
            <name>
               <simpleValue>IFCBUILDING</simpleValue>
            </name>
         </entity>
         <property datatype="IfcLabel">
            <propertySet>
               <simpleValue>ZDB_Building_Infos</simpleValue>
            </propertySet>
            <name>
               <simpleValue>Gebaeudeklasse</simpleValue>
            </name>
            <value>
               <simpleValue>GK2</simpleValue>
            </value>
         </property>
      </partOf>
      <attribute>
         <name>
            <simpleValue>Elevation</simpleValue>
         </name>
         <value>
            <xs:restriction base="xs:integer">
               <xs:minInclusive value="0" />
            </xs:restriction>
         </value>
      </attribute>
   </partOf>
   <property datatype="IfcBoolean">
      <propertySet>
         <simpleValue>Pset_WallCommon</simpleValue>
      </propertySet>
      <name>
         <simpleValue>Compartmentation</simpleValue>
      </name>
      <value>
         <simpleValue>False</simpleValue>
      </value>
   </property>
</applicability>
```

In der Applicability kommen somit zwei Erweiterungen zur Anwendung. Durch die Verschachtelung der Facets kann von einer Wand schlussendlich auf das Gebäude geschlossen werden. Mit

Hilfe der Erweiterung um die bestehenden Typen sind die Anforderungen bezüglich Gebäudeklasse und ober-/unterirdisch umsetzbar. Beide Erweiterungen finden bei allen drei behandelten Tabellen der OIB-Richtlinie 2 Anwendung. Teilweise wird jedoch von einem Bauteil zuerst auf einen Raum, dann erst auf das Geschoß und das Gebäude geschlossen. Ein Beispiel dafür sind die Spezifikationen für Punkt 2 der Tabelle 1b (Trennwände). Bei manchen Punkten wird keine Unterscheidung in ober-/unterirdisch benötigt, weshalb hier das Attribut `Elevation` wegfällt. Die Angabe des Geschoßes ist trotzdem notwendig, um die Verortung richtig abbilden zu können.

In weiterer Folge können nun die sechs Spezifikationen von Punkt 1 getestet werden. Das Modell hat jedoch eine Vielzahl an entsprechenden Wänden, weshalb deren Ergebnisse aufgrund des Umfangs nicht angeführt werden. Stattdessen werden die Ergebnisse der Stützen und der Träger gezeigt, welche sich im Erd- und Obergeschoß befinden. Die genaue Position ist in verschiedenen Abbildungen dieses Kapitels ersichtlich. Alle vier Bauteile wurden durch die Applicability richtig gefunden und ausgegeben (siehe Abb. 5.6). Sie besitzen jedoch nicht alle das Property *FireRating*, weshalb sie auch teilweise durch die Requirements gefallen sind. Trotzdem konnte mit den vier Bauteilen und allen Wänden die Funktionsweise der Gebäudeklasse, der Unterscheidung zwischen ober-/unterirdisch sowie der Verschachtelung der PartOf-Facets allgemein getestet werden. Ebenso wird die richtige Verortung der Bauteile gezeigt.

```
1.2.2 Tragende Bauteile in sonstigen oberirdischen Geschoßen - Träger

Applicable entities: 2
#40270=IfcBeam('28wbh1DZH86eheCYpQZIQo',#12,'Träger-001',$,$,#40221,#40261,'88EA5AC1-3634-481A-8AE8-322CDA8D26B2',.NOTDEFINED.)
#92695=IfcBeam('30MFIOrOL0lQkuYHS54vmt',#12,'Träger-001',$,$,#92615,#92686,'C058F498-D585-40BD-ABB8-891705139C37',.NOTDEFINED.)

Failed entities: 1

#40270=IfcBeam('28wbh1DZH86eheCYpQZIQo',#12,'Träger-001',$,$,#40221,#40261,'88EA5AC1-3634-481A-8AE8-322CDA8D26B2',.NOTDEFINED.)
The property set does not contain the required property "FireRating"


1.2.3 Tragende Bauteile in sonstigen oberirdischen Geschoßen - Stützen

Applicable entities: 2
#64613=IfcColumn('0zGnKhLsv2a8mwifD2nhuT',#12,'Stütze-001',$,$,#64565,#64604,'3D43152B-576E-4290-8C3A-B29342C6BE1D',.NOTDEFINED.)
#95496=IfcColumn('0uFdU$ZKDEj8eTXQ0oR0hx',#12,'Stütze-001',$,$,#95450,#95487,'383E77BF-8D43-4EB4-8A1D-85A0326C0AFB',.NOTDEFINED.)

Failed entities: 1

#95496=IfcColumn('0uFdU$ZKDEj8eTXQ0oR0hx',#12,'Stütze-001',$,$,#95450,#95487,'383E77BF-8D43-4EB4-8A1D-85A0326C0AFB',.NOTDEFINED.)
The property set does not contain the required property "FireRating"
```

Abb. 5.6: Ergebnis der oberirdischen Träger und Stützen

Die nächste bei Tabelle 1b angewendete Erweiterung ist der neue Tag *multiObjectProcessing*. Dabei kommen zwei Schlüsselwörter zum Einsatz: einerseits *differentPropertyValuesForAnyObject* für Trennwände aus Punkt 2 und Trenndecken aus Punkt 4.3, andererseits *samePropertyValuesForAllObjects* für Decken innerhalb eines Geschoßes aus Punkt 4.4.

Insgesamt gibt es zwei Spezifikation zur Überprüfung der Trennwände, welche sich jedoch nur durch den Unterschied im Wert des Attributs *Elevation* unterscheiden. Daher wird in weiterer Folge eine zusammenfassende Beschreibung durchgeführt. Die Definition von Trennwänden erfolgt über Unterschiede in den angrenzenden Räumen. Der notwendige Unterschied liegt im eigens dafür erstelltem Property *Wohnungsnummer*. Von diesem müssen bei den angrenzenden Räumen mindestens zwei verschiedene Werte vorhanden sein, unabhängig davon, wie viele Elemente es gibt. Im Code 5.3 ist die entsprechende Applicability angeführt. Die Wand steht in Verbindung zu einem Raum, welcher durch die Verschachtelung mittels `IFCRELAGGREGATES` mit einem ober- oder unterirdischen Geschoß und einem Gebäude verbunden ist, worin die relevante Gebäudeklasse definiert ist. Da dieser Bereich gleich wie im vorherigen Code ist, wurde aus

Gründen der Lesbarkeit dieses Facet inklusive der Verschachtelung im Code 5.3 eingeklappt (Zeilen 13–14) und nicht dargestellt. Nun wird das Property-Facet mit dem Property `Wohnungsnummer` angegeben. Danach folgt das Schlüsselwort `differentPropertyValuesForAnyObject`, welches aus dem vorhergehenden normalen Property-Facet eine Anforderung bezüglich des Unterschieds der Werte macht.

Code 5.3: Applicability der Spezifikation für Trennwände (Punkt 2, Tabelle 1b, OIB 2)

```
1   <applicability>
2      <entity>
3         <name>
4            <simpleValue>IFCWALL</simpleValue>
5         </name>
6      </entity>
7      <partOf relation="IFCRELSPACEBOUNDARY">
8         <entity>
9            <name>
10              <simpleValue>IFCSPACE</simpleValue>
11           </name>
12        </entity>
13        <partOf relation="IFCRELAGGREGATES"> ...
14        </partOf>
15        <property datatype="IfcInteger">
16           <propertySet>
17              <simpleValue>Pset_SpaceSpecific</simpleValue>
18           </propertySet>
19           <name>
20              <simpleValue>Wohnungsnummer</simpleValue>
21           </name>
22        </property>
23        <multiObjectProcessing>
24           <simpleValue>differentPropertyValuesForAnyObject</simpleValue>
25        </multiObjectProcessing>
26     </partOf>
27  </applicability>
```

Bevor zu den Testungen der entsprechenden Spezifikationen übergegangen werden kann, sind an dieser Stelle die Unterschiede bei der Modellierung von Wänden und Decken anzumerken. Im BIM-Modell begrenzen einige Wände einerseits nur einzelne Räume zum direkten Vergleich von zwei Property Values sowie andererseits auch mehrere Räume, wo dadurch mehrere Property Values zu untersuchen sind. Der Unterschied ist in den Abb. 5.7 und 5.8 gut zu erkennen. Der längliche Raum in der Mitte im Erdgeschoß wird zu den drei linken Räumen durch eine einzelne gemeinsame Wand begrenzt. Im Gegensatz dazu bestehen zu den Räumen auf der rechten Seite zwei getrennte Wände. Im Obergeschoß ist dies genau umgekehrt der Fall. Ebenso liegen einige Decken nur zwischen zwei Räumen und manche erstrecken sich über mehrere. Die Unterschiede im Testdurchlauf werden entsprechend in den Spezifikationen erklärt.

Die Testung der Spezifikationen der ober- und unterirdischen Trennwände (Punkt 2, Tabelle 1b) steht als nächstes an der Reihe. Da die Definition von Trennwänden über einen Unterschied im Property *Wohnungsnummer* der einzelnen Räume erfolgt, sind die jeweiligen Werte in den Abb. 5.7 und 5.8 dargestellt. Der *AdvancedIfcTester* sollte im Erdgeschoß neun und im Obergeschoß elf Wände als Trennwände erkennen. Der Unterschied in der verschiedenen Anzahl liegt in der unterschiedlichen Modellierung der Wände des mittleren Raums. Es kommen alle Wände außer diejenigen innerhalb der Wohnungen 1 und 11 sowie der Wand im Erdgeschoß zwischen den

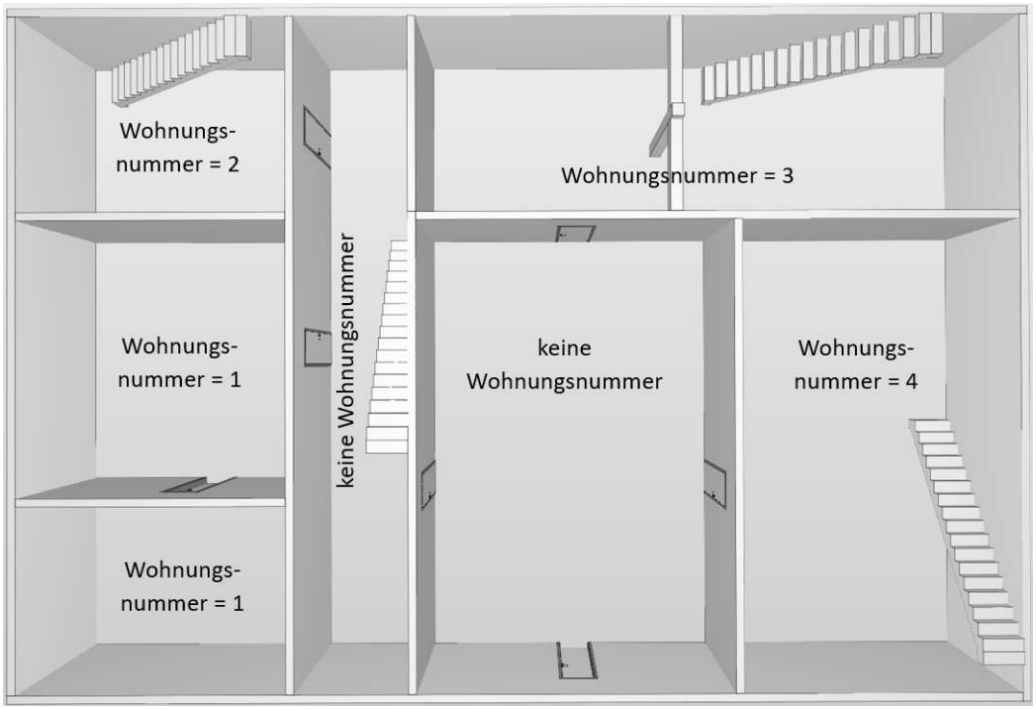

Abb. 5.7: Grundriss und Wohnungsnummern des Erdgeschoßes

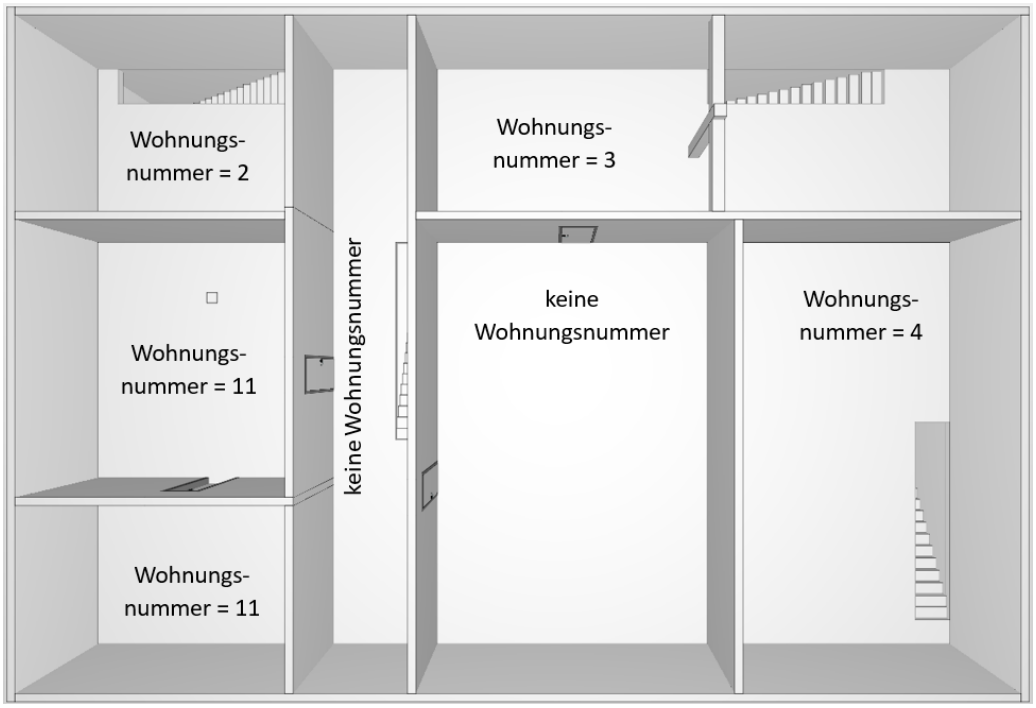

Abb. 5.8: Grundriss und Wohnungsnummern des Obergeschoßes

Räumen ohne Wohnungsnummer in Frage. Die Außenwände sind, obwohl nur auf einer Seite ein Raum liegt, dennoch zu beachten, da sie mehrere Räume berühren. Nach der Testung wird ersichtlich (siehe Abb. 5.9 und 5.10), dass der *AdvancedIfcTester* alle entsprechenden Wände erkennt.

```
2.2 Trennwände in oberirdischen Geschoßen

Applicable entities: 20
#40605=IfcWall('2Jpxnq8sv4A99IQUXKCv9z',#12,'Wand-004',$,$,#40528,#40595,'93CFBC74-236E-4428-9252-69E85433927D',.NOTDEFINED.)
#42245=IfcWall('1bHseELkf9Pu_YLQ7mvqNF',#12,'Wand-003',$,$,#42186,#42235,'65476A0E-56EA-4967-8FA2-55A1F0E745CF',.NOTDEFINED.)
#42563=IfcWall('21Nzv8xWr7$RnWMptqYI5C',#12,'Wand-002',$,$,#42502,#42553,'815FDE48-EE0D-47FD-BC60-5B3DF489214C',.NOTDEFINED.)
...

Failed entities: 20

#40605=IfcWall('2Jpxnq8sv4A99IQUXKCv9z',#12,'Wand-004',$,$,#40528,#40595,'93CFBC74-236E-4428-9252-69E85433927D',.NOTDEFINED.)
The property value "REI120" does not match the requirements

#42245=IfcWall('1bHseELkf9Pu_YLQ7mvqNF',#12,'Wand-003',$,$,#42186,#42235,'65476A0E-56EA-4967-8FA2-55A1F0E745CF',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#42563=IfcWall('21Nzv8xWr7$RnWMptqYI5C',#12,'Wand-002',$,$,#42502,#42553,'815FDE48-EE0D-47FD-BC60-5B3DF489214C',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

...
```

Abb. 5.9: Ausschnitt des Ergebnisses der Trennwände in oberirdischen Geschoßen

```
2.3 Trennwände in unterirdischen Geschoßen

Applicable entities: 11
#3364=IfcWall('2n6L0n3DH4jOJZ_N$0pkUB',#12,'Wand-004',$,$,#3303,#3354,'B1195031-0CD4-44B5-84E3-F97FC0CEE78B',.NOTDEFINED.)
#3686=IfcWall('17dgX1BaP4tfRJ3hwQ7Kxy',#12,'Wand-003',$,$,#3629,#3676,'479EA841-2E46-44DE-96D3-0EBE9A1D4EFC',.NOTDEFINED.)
#4000=IfcWall('3PyBESU4TAkOTdbJYEUcNX',#12,'Wand-002',$,$,#3943,#3990,'D9F0B39C-7847-4AB9-8767-95388E7A65E1',.NOTDEFINED.)
...

Failed entities: 11

#3364=IfcWall('2n6L0n3DH4jOJZ_N$0pkUB',#12,'Wand-004',$,$,#3303,#3354,'B1195031-0CD4-44B5-84E3-F97FC0CEE78B',.NOTDEFINED.)
The property value "REI60" does not match the requirements

#3686=IfcWall('17dgX1BaP4tfRJ3hwQ7Kxy',#12,'Wand-003',$,$,#3629,#3676,'479EA841-2E46-44DE-96D3-0EBE9A1D4EFC',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#4000=IfcWall('3PyBESU4TAkOTdbJYEUcNX',#12,'Wand-002',$,$,#3943,#3990,'D9F0B39C-7847-4AB9-8767-95388E7A65E1',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

...
```

Abb. 5.10: Ausschnitt des Ergebnisses der Trennwände in unterirdischen Geschoßen

Die nächsten Spezifikationen befassen sich mit Trenndecken (Punkt 4.3, Tabelle 1b) und Decken innerhalb von Wohnungen (Punkt 4.3, Tabelle 1b). Die Definition der Decken ist von den Wohnungsnummern der darüber und darunter befindlichen Räumen abhängig. Bevor die beiden Spezifikationen genauer erklärt werden, zeigt Abb.5.11 die unterschiedlichen auftretenden Fälle sowie bereits die entsprechenden Ergebnisse der Spezifikationen. Neben den drei Decken des Modells werden zusätzlich blau strichliert die Trennungen der Räume dargestellt. Es lassen sich dabei drei Fälle unterscheiden:

1. Eine Decke begrenzt mehrere Räume, welche unterschiedlichen Wohnungen zugeordnet sind: Dieser Fall entspricht der Decke über den drei linken Räumen. Hier liegen einerseits Wohnung 2 (oben und unten) sowie Wohnung 1 (unten) und Wohnung 11 (oben). Somit liegt diese Decke nicht in einer Wohnung.

2. Eine Decke begrenzt mehrere Räume, wobei einige Räume keine Wohnungsnummer haben: Im Testmodell teilt sich die zweistöckige Wohnung 3 mit den vier Räumen (je zwei oben und unten) ohne Wohnungsnummer eine Decke. Aufgrund der fehlenden Properties zählt diese Decke nicht als innerhalb einer Wohnung liegend.

3. Eine Decke begrenzt oben und unten jeweils einen Raum aus einer Wohnung: Dieser Deckenfall tritt im Modell in Wohnung 4 auf und entspricht einer Wohnungsdecke.

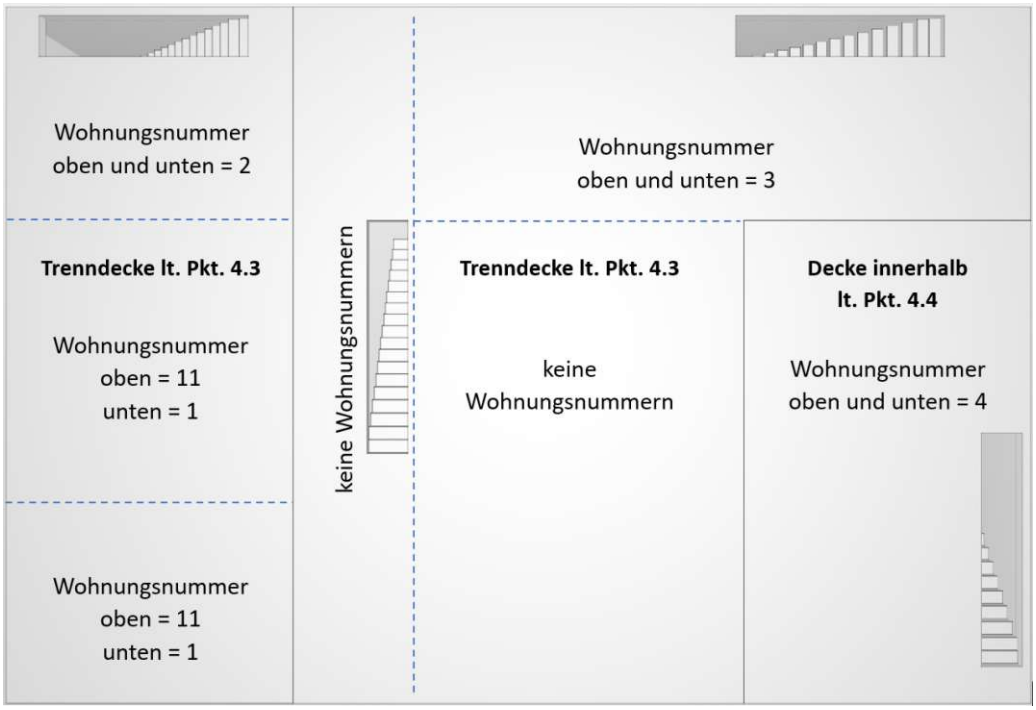

Abb. 5.11: Decken über dem Erdgeschoß

Für die Spezifikation der Trenndecken (Punkt 4.3, Tabelle 1b), deren Definition bis auf die geänderte Entität im ersten Entity-Facet analog zu der Spezifikation der Trennwände erfolgt, kann daher auf eine Erklärung der Erstellung verzichtet werden. Ebenso liefert die Testung der Spezifikation mit dem *AdvancedIfcTester* Ergebnisse, die auf den Erkenntnissen der vorhergehenden Testungen der Trennwände aufbauen und somit keine Besonderheiten mit sich bringen (siehe Abb. 5.12). Die beiden der Applicability entsprechenden Trenndecken zwischen dem Erd- und Obergeschoß sind auch in Abb. 5.11 zu sehen.

```
4.3 Trenndecken in oberirdischen Geschoßen

Applicable entities: 3
#63013=IfcSlab('2xvqifBlr0_AlV5r8jFY7s',#12,'Decke-001',$,$,#62911,#63004,'BBE74B29-2EFD-40F8-ABDF-17522D3E21F6',.NOTDEFINED.)
#64073=IfcSlab('2kzR5d24L0jvq$5x_iq_Ec',#12,'Decke-001',$,$,#63981,#64064,'AEF5B167-0845-40B7-9D3F-17BFACD3E3A6',.NOTDEFINED.)
#95239=IfcSlab('2CgiacVBjC7BSd3GWSNA_l',#12,'Decke-003',$,$,#95169,#95230,'8CAAC926-7CBB-4C1C-B727-0D081C5CAFAF',.NOTDEFINED.)

Failed entities: 3

#63013=IfcSlab('2xvqifBlr0_AlV5r8jFY7s',#12,'Decke-001',$,$,#62911,#63004,'BBE74B29-2EFD-40F8-ABDF-17522D3E21F6',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#64073=IfcSlab('2kzR5d24L0jvq$5x_iq_Ec',#12,'Decke-001',$,$,#63981,#64064,'AEF5B167-0845-40B7-9D3F-17BFACD3E3A6',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#95239=IfcSlab('2CgiacVBjC7BSd3GWSNA_l',#12,'Decke-003',$,$,#95169,#95230,'8CAAC926-7CBB-4C1C-B727-0D081C5CAFAF',.NOTDEFINED.)
The property set does not contain the required property "FireRating"
```

Abb. 5.12: Ergebnis der Trenndecken

Die letzte fehlende zu testende Erweiterung wird bei Decken innerhalb einer Wohnung benutzt (Punkt 4.4, Tabelle 1b). Der Aufbau der Applicability ist analog zum Beispiel aus Code 5.3 (Trennwände), weshalb kein Ausschnitt aus der IDS-Datei dargestellt wird. Der Unterschied liegt in der anderen Entität und in dem geänderten Schlüsselwort, welches nun *samePropertyValues-*

ForAllObjects ist. Dadurch wird kein Unterschied, sondern eine Gleichheit in den Property Values gesucht.

Bei der Testung der Spezifikation ist wieder auf die Modellierung und die drei vorkommenden Fälle hinzuweisen (siehe Abb. 5.12). Der *AdvancedIfcTester* erkennt fehlerfrei, dass nur eine Decke die Applicability erfüllt (siehe Abb. 5.13). Damit konnte die Funktionsweise des Schlüsselworts *samePropertyValuesForAllObjects* gezeigt werden.

```
4.4 Decken innerhalb in oberirdischen Geschoßen

Applicable entities: 1
#64354=IfcSlab('0_i6QsGNH7oO5fkWQhC83c',#12,'Decke-002',$,$,#64284,#64345,'3EB066B6-4174-47C9-8169-BA06AB3080E6',.NOTDEFINED.)

Failed entities: 1

#64354=IfcSlab('0_i6QsGNH7oO5fkWQhC83c',#12,'Decke-002',$,$,#64284,#64345,'3EB066B6-4174-47C9-8169-BA06AB3080E6',.NOTDEFINED.)
The property set does not contain the required property "FireRating"
```

Abb. 5.13: Ergebnis der Decken innerhalb von Wohnungen

Die letzte Spezifikation der Tabelle 1b (Punkt 4.5) beschreibt unterirdische Decken. Hier kommen nur bereits erklärte Erweiterung zum Einsatz. Daher wird auf den Inhalt und die Erstellung dieser Spezifikation nicht näher eingegangen. Der *AdvancedIfcTester* wurde auch mit dieser Spezifikation getestet. Da die Ergebnisausgabe wie erwartet funktioniert hat und keine zusätzlichen Besonderheiten im Vergleich zu den vorherigen Fällen vorkommen, wird die Testung hier nicht beschrieben.

5.3 Umsetzung der Anforderungen an den Feuerwiderstand von Treppenhäusern

Dieser Abschnitt beschreibt die umgesetzten Spezifikationen der Tabelle 2a der OIB-Richtlinie 2 näher. Für die Tabelle 2b wurden keine Spezifikationen erstellt, da sie denjenigen für Tabelle 2a sehr ähnlich sind. Der einzige Unterschied liegt in zusätzlichen Properties des Gebäudes, worin Informationen bezüglich der mechanischen Belüftung, Brandmeldeanlage und so weiter hinterlegt sein müssen. Für die Tabelle 2a konnten sechs Spezifikationen umgesetzt werden, welche in Tab. 5.2 aufgelistet sind. Alle enthalten wieder die beiden Erweiterungen zur Verschachtelung und Konkretisierung der verbundenen Elemente. Zusätzlich kommt noch ein weiteres Schlüsselwort zur Anwendung. Die gesamte IDS-Datei für Tabelle 2a kann dem Anhang D entnommen werden. Die einzelnen Spezifikationen beziehen sich wieder auf die Gebäudeklasse 2.

Die Spezifikationen für Wände von Treppenhäusern (Punkte 1.1 und 1.2 der Tabelle 2a) sind relativ kurz. Hier gibt es ein Entity-Facet, das die Wände angibt, sowie eine Verschachtelung von PartOf-Facets. Das erste gibt die verbundenen Räume an, wobei einer davon den gewünschten Property Value *Treppe* besitzen muss. Dies wird durch ein standardmäßiges Property-Facet ohne einem weiteren Schlüsselwort angegeben. Die weiteren PartOf-Facets dienen wieder zur Beschreibung des ober- oder unterirdischen Geschoßes sowie der Gebäudeklasse.

Zu den zu beachtenden Wänden zählen alle, an die mindestens ein Raum mit dem Raumnamen *Treppe* angrenzt. Dies trifft in allen Geschoßen auf den mittleren länglichen Raum zu. Der *AdvancedIfcTester* gibt in der Spezifikation alle vorhandenen Wände richtig aus. Einerseits sind dies die oberen und unteren Außenwände sowie andererseits die Trennwände zu den umliegenden Räumen. Damit kann bestätigt werden, dass das Property-Facet bei einem verbundenen Element auch ohne einem der neuen Schlüsselwörter richtig funktioniert (siehe Abb. 5.14 und 5.15).

Tab. 5.2: Spezifikationen für Tabelle 2a der OIB-Richtlinie 2

Anforderung	Entität	verwendete Erweiterungen
1.1 Wände Treppenhäuser oberirdisch	Wände	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente
1.2 Wände Treppenhäuser unterirdisch	Wände	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente
3.1 Türen Treppenhäuser – Räume	Türen	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, differentObjectsForEachPropertyValue
3.2 Türen Treppenhäuser – Gänge	Türen	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, differentObjectsForEachPropertyValue
3.3 Türen Treppenhäuser unterirdisch	Türen	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente, differentPropertyValuesForAnyObject
4 Treppen in Treppenhäuser	Treppen	Verschachtelung, Spaceboundary, Properties und Attribute für verbundene Elemente

```
1.1 Wände von Treppenhäusern - oberirdisch

Applicable entities: 11
#40605=IfcWall('2Jpxnq8sv4A99IQUXKCv9z',#12,'Wand-004',$,$,#40528,#40595,'93CFBC74-236E-4428-9252-69E85433927D',.NOTDEFINED.)
#42563=IfcWall('21Nzv8xWr7$RnWMptqYI5C',#12,'Wand-002',$,$,#42502,#42553,'815FDE48-EE0D-47FD-BC60-5B3DF489214C',.NOTDEFINED.)
#45649=IfcWall('0I5z5byCj5$fmJ4IEsxcYp',#12,'Wand-009',$,$,#45570,#45639,'1217D165-F0CB-45FE-9C13-1123B6EE68B3',.NOTDEFINED.)
...

Failed entities: 6

#42563=IfcWall('21Nzv8xWr7$RnWMptqYI5C',#12,'Wand-002',$,$,#42502,#42553,'815FDE48-EE0D-47FD-BC60-5B3DF489214C',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#45649=IfcWall('0I5z5byCj5$fmJ4IEsxcYp',#12,'Wand-009',$,$,#45570,#45639,'1217D165-F0CB-45FE-9C13-1123B6EE68B3',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#62654=IfcWall('2eGpJwFST3gQ8oVoS$ajjX',#12,'Wand-013',$,$,#62597,#62644,'A84334FA-3DC7-43A9-A232-7F273F92DB61',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

...
```

Abb. 5.14: Ausschnitt des Ergebnisses der Wände von oberirdischen Treppenhäusern

Die Spezifikationen des Punkts 3 beschäftigen sich mit Türen in Treppenhäuser und sind komplizierter, da hier zwei Schlüsselwörter verwendet werden. Bei den Punkten 3.1 und 3.2 findet das dritte und letzte Schlüsselwort *differentObjectsForEachPropertyValue* Anwendung, da die beiden Elemente an beiden Seiten der Tür unterschiedliche Property Values benötigen. Im Code 5.4 ist dazu die Applicability des Punkts 3.1 angegeben. Hier werden im PartOf-Facet bei der Beschreibung des Raums zwei Property-Facets verwendet, die einerseits den Wert *Treppe* und andererseits eine Auswahl anderer Werte angeben. In diesem Beispiel können es theoretisch alle Widmungen außer verschiedene Arten von Gängen sein. Zur besseren Übersichtlichkeit werden jedoch praktischerweise nur sechs Werte in dem Beispiel aufgeführt. Bei Punkt 3.2 wären in der

```
1.2 Wände von Treppenhäusern - unterirdisch

Applicable entities: 6
#3364=IfcWall('2n6L0n3DH4jOJZ_N$0pkUB',#12,'Wand-004',$,$,#3303,#3354,'B1195031-0CD4-44B5-84E3-F97FC0CEE78B',.NOTDEFINED.)
#4000=IfcWall('3PyBESU4TAkOTdbJYEUcNX',#12,'Wand-002',$,$,#3943,#3990,'D9F0B39C-7847-4AB9-8767-95388E7A65E1',.NOTDEFINED.)
#4648=IfcWall('0VuAN23YH2QvZbsJ4fUII6',#12,'Wand-006',$,$,#4571,#4638,'1FE0A5C2-0E24-426B-98E5-D93129792486',.NOTDEFINED.)
...

Failed entities: 3

#4000=IfcWall('3PyBESU4TAkOTdbJYEUcNX',#12,'Wand-002',$,$,#3943,#3990,'D9F0B39C-7847-4AB9-8767-95388E7A65E1',.NOTDEFINED.)
The property set does not contain the required property "FireRating"

#4648=IfcWall('0VuAN23YH2QvZbsJ4fUII6',#12,'Wand-006',$,$,#4571,#4638,'1FE0A5C2-0E24-426B-98E5-D93129792486',.NOTDEFINED.)
The property value "REI30" does not match the requirements

#11101=IfcWall('1W2nbRHgPFAALyyUnJ6iUV',#12,'Wand-009',$,$,#11022,#11091,'600B195B-46A6-4F28-A57C-F1EC531AC79F',.NOTDEFINED.)
The property set does not contain the required property "FireRating"
```

Abb. 5.15: Ausschnitt des Ergebnisses der Wände von unterirdischen Treppenhäusern

Auswahlliste die verschiedenen Arten von Gängen (*Haupt-* oder *Nebengang*) angegeben. Zu jedem der Property-Facets muss es aufgrund des danach angegebenen Schlüsselworts ein Objekt geben, dass den vorgeschriebenen Wert besitzt.

Code 5.4: Applicability der Spezifikation für Türen in oberiridischen Treppenhäusern (Punkt 3.1, Tabelle 2a, OIB 2)

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCDOOR</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELSPACEBOUNDARY">
      <entity>
         <name>
            <simpleValue>IFCSPACE</simpleValue>
         </name>
      </entity>
      <partOf relation="IFCRELAGGREGATES"> ...
      </partOf>
      <property datatype="IfcLabel">
         <propertySet>
            <simpleValue>Pset_SpaceSpecific</simpleValue>
         </propertySet>
         <name>
            <simpleValue>SpaceName</simpleValue>
         </name>
         <value>
            <simpleValue>Treppe</simpleValue>
         </value>
      </property>
      <property datatype="IfcLabel">
         <propertySet>
            <simpleValue>Pset_SpaceSpecific</simpleValue>
         </propertySet>
            <name>
         <simpleValue>SpaceName</simpleValue>
            </name>
         <value>
            <xs:restriction base="xs:string">
```

```
35                    <xs:enumeration value="Wohnen"/>
36                    <xs:enumeration value="Toilette"/>
37                    <xs:enumeration value="Küche"/>
38                    <xs:enumeration value="Einlagerungsraum"/>
39                    <xs:enumeration value="Kinderwagen-Abstellraum"/>
40                    <xs:enumeration value="Toilette"/>
41                 </xs:restriction>
42              </value>
43           </property>
44           <multiObjectProcessing>
45              <simpleValue>differentObjectsForEachPropertyValue</simpleValue>
46           </multiObjectProcessing>
47        </partOf>
48     </applicability>
```

Aufgrund der Verwendung der beiden Schlüsselwörtern waren für die beiden Spezifikationen umfangreiche Testungen notwendig. Wie vorhin erwähnt, ist in jedem Geschoß der mittlere längliche Raum als Treppenhaus modelliert (siehe Abb. 5.16 und 5.17). Die beiden linken angrenzenden Räume besitzen verschiedene Namen, wie zum Beispiel *Wohnen* oder *Küche*. Der rechts vom Treppenhaus liegende Raum ist ein Gang, je nach Geschoß entweder ein *Hauptgang* oder ein *Nebengang*.

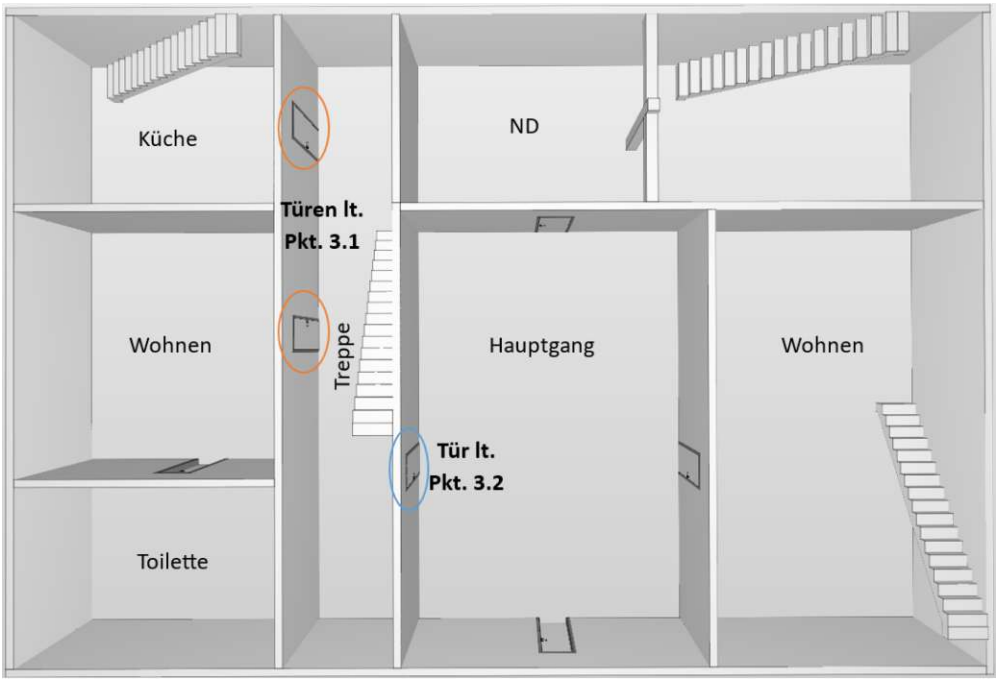

Abb. 5.16: Raumnamen im Erdgeschoß

Die ersten beiden Spezifikationen des dritten Punkts (Punkte 3.1 und 3.2) beschäftigen sich mit oberirdischen Treppenhäusern. Die Erweiterung funktioniert bei den verschiedenen Widmungen ohne Probleme, wie Abb. 5.18 anhand der Ergebnisausgabe zeigt. Es werden die Türen zu den verschiedenen Räumen auf der linken Seite (orange markiert) sowie die Türen zu den Gängen auf der rechten Seite (blau markiert) erkannt und in der Applicability richtig angegeben. Bei den Requirements fallen zwei Stück durch, da einmal das Property nicht existiert und einmal ein falscher Wert angegeben wird. Bei Punkt 3.2 gibt es an die Türen keine Anforderungen, weshalb dadurch keine durchgefallenen Elemente existieren (siehe Abb. 5.18). Dennoch wurde entschieden,

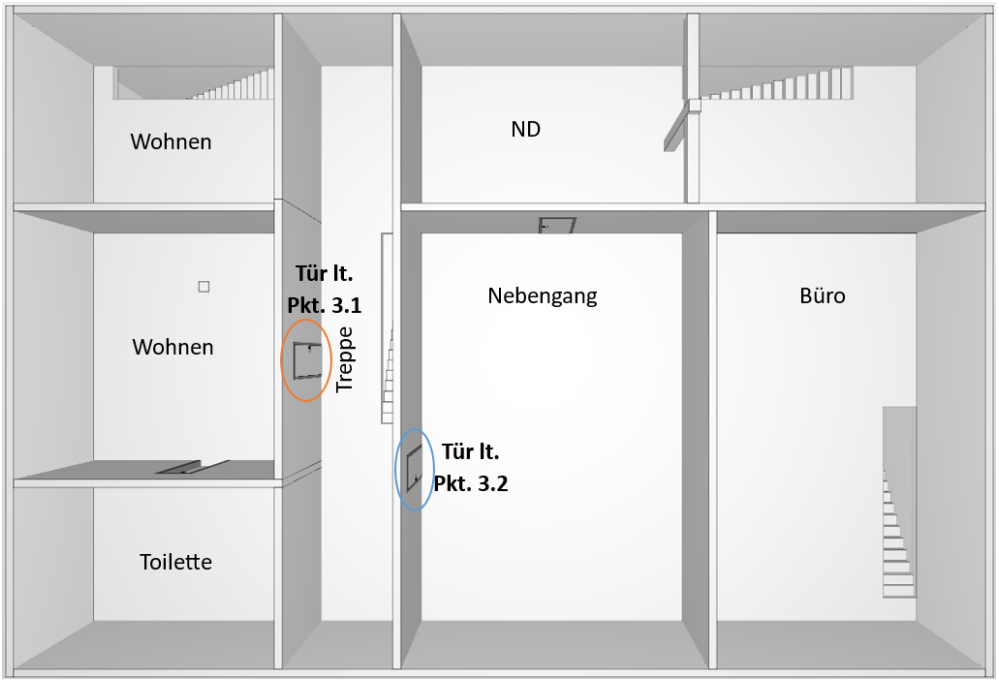

Abb. 5.17: Raumnamen im Obergeschoß

die Spezifikation in die IDS-Datei aufzunehmen, da vor allem die Testung der Applicability das gewünschte Ziel war. Schlussendlich kann das dritte Schlüsselwort als ausreichend getestet und bestanden betrachtet werden.

```
3.1 Türen in Wänden von Treppenhäusern - zu Wohnungen, Betriebseinheiten sowie sonstigen Räumen (oberirdisch)

Applicable entities: 3
#68079=IfcDoor('1lxXleIfD948TtHE5LZB6l',#12,'Tür-001',$,$,#65324,#68070,'6FEE1BE8-4A93-4910-8777-44E1558CB1AF',2.04,0.9,.DOOR.,$,$)
#69363=IfcDoor('0umzE4KcXCOQFcxwcbjEtt',#12,'Tür-002',$,$,#69351,#69360,'38C3D384-5268-4C61-A3E6-EFA9A5B4EDF7',2.04,0.88,.DOOR.,$,$)
#85481=IfcDoor('22qJn_GCf9vxwbkiUw_$dZ',#12,'Tür-002',$,$,#85469,#85478,'82D13C7E-40CA-49E7-BEA5-BAC7BAFBF9E3',2.04,0.88,.DOOR.,$,$)

Failed entities: 3

#68079=IfcDoor('1lxXleIfD948TtHE5LZB6l',#12,'Tür-001',$,$,#65324,#68070,'6FEE1BE8-4A93-4910-8777-44E1558CB1AF',2.04,0.9,.DOOR.,$,$)
The property set does not contain the required property "FireRating"

...


3.2 Türen in Wänden von Treppenhäusern - zu Gängen oberirdisch

Applicable entities: 2
#45979=IfcDoor('2cm5$hP6L5UQrl8IsO_20e',#12,'Tür-004',$,$,#45967,#45976,'A6C05FEB-6465-4579-AD6F-212D98F82028',2.04,0.88,.DOOR.,$,$)
#77921=IfcDoor('06Yw2JAGbD1QpTpHgxKZp2',#12,'Tür-004',$,$,#77909,#77918,'068BA093-2909-4D05-ACDD-CD1ABB523CC2',2.04,0.88,.DOOR.,$,$)

Failed entities: 0
```

Abb. 5.18: Ausschnitt der Ergebnisse der Türen in oberirdischen Treppenhäusern

Für Punkt 3.3 wird das schon zuvor beschriebene *differentPropertyValuesForAnyObject* benutzt. Damit soll ein Unterschied zwischen den an die Tür angrenzenden Räumen festgestellt werden. Zusätzlich wird aber der Property Value *Treppe* vorgegeben, der diesmal durch einen Raum einzuhalten ist. Im Code 5.5 ist die Applicability zu sehen. Die Verschachtelung der PartOf-Facets wurde zur besseren Lesbarkeit wieder ausgeblendet (zwischen Zeilen 13–14). Das Property-Facet beinhaltet den von einem Raum geforderten Property Value. Danach folgt wieder die Angabe des Schlüsselworts.

Code 5.5: Applicability der Spezifikation für Türen in unterirdischen Treppenhäusern (Punkt 3.3, Tabelle 2a, OIB 2)

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCDOOR</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELSPACEBOUNDARY">
      <entity>
         <name>
            <simpleValue>IFCSPACE</simpleValue>
         </name>
      </entity>
      <partOf relation="IFCRELAGGREGATES"> ...
      </partOf>
      <property datatype="IfcLabel">
         <propertySet>
            <simpleValue>Pset_SpaceSpecific</simpleValue>
         </propertySet>
         <name>
            <simpleValue>SpaceName</simpleValue>
         </name>
         <value>
            <simpleValue>Treppe</simpleValue>
         </value>
      </property>
      <multiObjectProcessing>
         <simpleValue>differentPropertyValuesForAnyObject</simpleValue>
      </multiObjectProcessing>
   </partOf>
</applicability>
```

Bei der Testung des Punkts 3.3 wurde neben dem bereits getesteten Schlüsselwort *differentPropertyValuesForAnyObject* zusätzlich noch getestet, dass von einem verbundenen Element ein Property Value gefordert wird. In diesem Fall muss ein angrenzender Raum ein Treppenhaus sein. Auch hier findet der *AdvancedIfcTester* alle drei beteiligten Türen rund um den mittleren Raum ohne Probleme (siehe Abb. 5.19).

```
3.3 Türen in Wänden von Treppenhäusern - zu Gängen und Räumen unterirdisch

Applicable entities: 3
#7749=IfcDoor('0JsxoS8AXEGe43sIEQyZ2l',#12,'Tür-001',$,$,#4968,#7740,'13DBBC9C-20A8-4E42-8103-D9239AF230AF',2.04,0.88,.DOOR.,$,$)
#11431=IfcDoor('3zVm1ZQPP1t9QLaSSppJYn',#12,'Tür-004',$,$,#11419,#11428,'FD7F0063-6996-41DC-9695-91C733CD38B1',2.04,0.88,.DOOR.,$,$)
#28605=IfcDoor('3zCnN$PXX28PC4n9pqSpyh',#12,'Tür-002',$,$,#25834,#28596,'FD3315FF-6618-4221-9304-C49CF4733F2B',2.04,0.88,.DOOR.,$,$)

Failed entities: 1

#7749=IfcDoor('0JsxoS8AXEGe43sIEQyZ2l',#12,'Tür-001',$,$,#4968,#7740,'13DBBC9C-20A8-4E42-8103-D9239AF230AF',2.04,0.88,.DOOR.,$,$)
The property set does not contain the required property "FireRating"
```

Abb. 5.19: Ergebnis der Türen in unterirdischen Treppenhäusern

Die letzte Spezifikation definiert Treppen in Treppenhäusern (Punkt 4) und ist sehr kurz. Die Applicability dazu wird in Code 5.6 angegeben. Eine Verbindung zwischen Treppe und Raum lässt sich durch keine Relation herstellen. Deshalb ist durch das Property *TypeOfStairs* anzugeben, dass es sich um eine Treppe in einem Treppenhaus handelt und nicht um eine Wohnungstreppe. Weiters wird über eine Verschachtelung von PartOf-Facets noch die Gebäudeklasse überprüft.

Code 5.6: Applicability der Spezifikation Treppenläufe (Punkt 4, Tabelle 2a, OIB 2)

```
<applicability>
   <entity>
      <name>
         <simpleValue>IFCSTAIR</simpleValue>
      </name>
   </entity>
   <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
      <entity>
         <name>
            <simpleValue>IFCBUILDINGSTOREY</simpleValue>
         </name>
      </entity>
      <partOf relation="IFCRELAGGREGATES">
         <entity>
            <name>
               <simpleValue>IFCBUILDING</simpleValue>
            </name>
         </entity>
         <property datatype="IfcLabel">
            <propertySet>
               <simpleValue>ZDB_Building_Infos</simpleValue>
            </propertySet>
            <name>
               <simpleValue>Gebaeudeklasse</simpleValue>
            </name>
            <value>
               <simpleValue>GK2</simpleValue>
            </value>
         </property>
      </partOf>
   </partOf>
   <property datatype="IfcLabel">
      <propertySet>
         <simpleValue>Pset_StairSpecific</simpleValue>
      </propertySet>
      <name>
         <simpleValue>TypeOfStair</simpleValue>
      </name>
      <value>
         <xs:restriction base="xs:string">
         <xs:enumeration value="Haupttreppe"/>
         <xs:enumeration value="Nebentreppe"/>
         </xs:restriction>
      </value>
   </property>
</applicability>
```

Die Testung dieser Spezifikation erfolgte anhand der fünf im Modell vorhandenen Treppen. Der *AdvancedIfcTester* erkennt, dass die beiden Treppen in der Mitte der Geschoße den Property Value *Haupttreppe* haben und in einem Gebäude der richtigen Gebäudeklasse liegen, und gibt beide im Ergebnis an (siehe Abb. 5.20). Damit kann die richtige Verortung der Treppen bewiesen werden.

```
4. Treppenläufe und Podeste in Treppenhäusern

Applicable entities: 2
#564=IfcStair('1Lf_e8m815a9T8r1D6xGZw',#12,'Treppe - 001',$,$,#562,$,'55A7EA08-C080-4590-9748-D41346ED08FA',.NOTDEFINED.)
#63225=IfcStair('1ZP$PQA49CZOdVG14Sgiuy',#12,'Treppe - 001',$,$,#63224,$,'6367F65A-2842-4C8D-89DF-40111CAACE3C',.NOTDEFINED.)

Failed entities: 1

#564=IfcStair('1Lf_e8m815a9T8r1D6xGZw',#12,'Treppe - 001',$,$,#562,$,'55A7EA08-C080-4590-9748-D41346ED08FA',.NOTDEFINED.)
The property set does not contain the required property "FireRating"
```

Abb. 5.20: Ergebnis der Treppen

Kapitel 6

Potential von IDS für Prüfregeln

Im Forschungsprojekt *BRISE-Vienna* werden verschiedenste bestehende oder eigens programmierte Prüfregeln mit der Software *Solibri* verwendet. Die Prüfung der Rechtsmaterie erfolgt entweder durch alleinstehende Regeln oder durch den Ablauf von Regelkaskaden. Urban et al. [40] beschreiben eine mögliche Einteilung der daraus entwickelten Prüfroutinen in drei Typen. Diese sind automatische, teilautomatische und unterstützende Abläufe. Beim ersten Typ ist im Gegensatz zu den anderen beiden keine Sichtung oder Entscheidung durch einen Referenten notwendig.

Für dieses Kapitel wurden die in *BRISE-Vienna* vorhandenen Prüfregeln der OIB-Richtlinien 2 [24], 2.1 [26], 2.2 [27], 2.3 [28] und 4 [29] aus dem Jahr 2019 näher untersucht, um die potentielle Verwendung von IDS für Prüfregeln zu analysieren. Zuerst wurden die teilautomatischen und unterstützenden Prüfregeln in den Typ *teilautomatisch oder unterstützend* zusammengefasst. Danach wurde ein Vergleich zwischen *Solibri* und IDS durchgeführt. Dessen Ziel war die Feststellung, wie viele der vorhandenen Prüfregeln mit IDS (Stand 2023) ohne Erweiterungen sowie den eigenen Erweiterungen umsetzbar sind. Gleichzeitig wurden auch die Grenzen und Unterschiede der beiden Bereiche aufgezeigt.

Aus den vier Richtlinien entstammen zusammen 131 Prüfroutinen, welche jeweils eigene zu prüfende Punkte abdecken. Die Routinen bestehen aus mindestens einer bis hin zu fünf einzelnen aneinandergereihten Regeln. Ziel der Prüfroutinen stellt die Prüfung der bautechnischen Themen bezüglich des Brandschutzes und der Barrierefreiheit dar. Baurechtliche Themen sind nicht Bestandteil der OIB-Richtlinien, wie Krischmann et al. [20] erläutern. Nach Sichtung der einzelnen vorhandenen Routinen lässt sich feststellen, dass etwas mehr als die Hälfte automatisch abläuft und die andere Hälfte teilautomatisch oder unterstützend funktioniert (siehe Abb. 6.1).

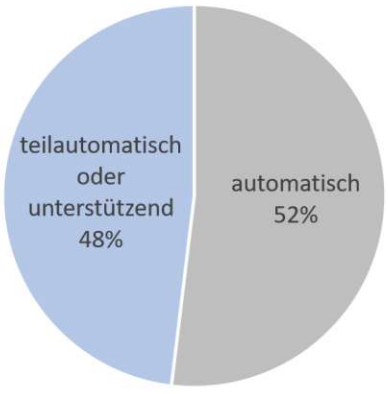

Abb. 6.1: Prozentuale Aufteilung nach Automatisierungsgrad der untersuchten Prüfroutinen

In weiterer Folge werden die Prüfroutinen erneut in zwei beziehungsweise drei Kategorien eingeteilt. Diese beschreiben nun die Umsetzbarkeit mit IDS. Hier wird einerseits in *möglich* und *nicht möglich* unterschieden. Alle möglichen werden weiter bezüglich der Verwendung der Erweiterung oder der Original-IDS unterteilt. Die Ergebnisse sind graphisch in Abb. 6.2

dargestellt. Mit dem Originalstand lassen sich bereits 37 Prüfroutinen umsetzten. Diese umfassen großteils nur teilautomatische oder unterstützende Anwendungsfälle in Form von graphischen Hilfestellungen (28 Stück). Durch die in dieser Arbeit beschriebenen Erweiterungen kommen zusätzlich noch 41 hinzu. Diese Prüfroutinen benötigen zum Großteil die eingebrachte Relation *IfcRelSpaceBoundary* oder die Vorgabe von Properties bei verbundenen Elementen. Insgesamt ist somit die Erstellung von ca. 60 % der aus *Solibri* stammenden Prüfroutinen möglich. Aufbauend auf die vorhin getroffene Unterscheidung zwischen automatischen und teilautomatischen oder unterstützenden Prüfroutinen ist anzumerken, dass diese Unterscheidung auch in IDS fortgesetzt wurde. Für teilautomatische oder unterstützende Prüfregeln aus *Solibri* wurde nicht untersucht, ob sie mit IDS automatisch möglich wären, sondern nur, ob ebenfalls eine teilautomatische oder unterstützende Prüfung umsetzbar ist, wie zum Beispiel die Filterung von Elementen. Abb. 6.2 zeigt, dass durch die Erweiterungen vor allem bei den automatischen Prüfregeln ein großer Teil ermöglicht wird.

Anzumerken ist an dieser Stelle, dass die Erweiterung vor allem bei den Richtlinien zum Brandschutz viele Punkte umsetzbar macht. Bei den Prüfroutinen der OIB-Richtlinie 4 kommt nur ein einziges Mal eine Erweiterung zum Einsatz. Der Rest wird entweder schon von der Original-IDS abgedeckt oder ist nicht möglich.

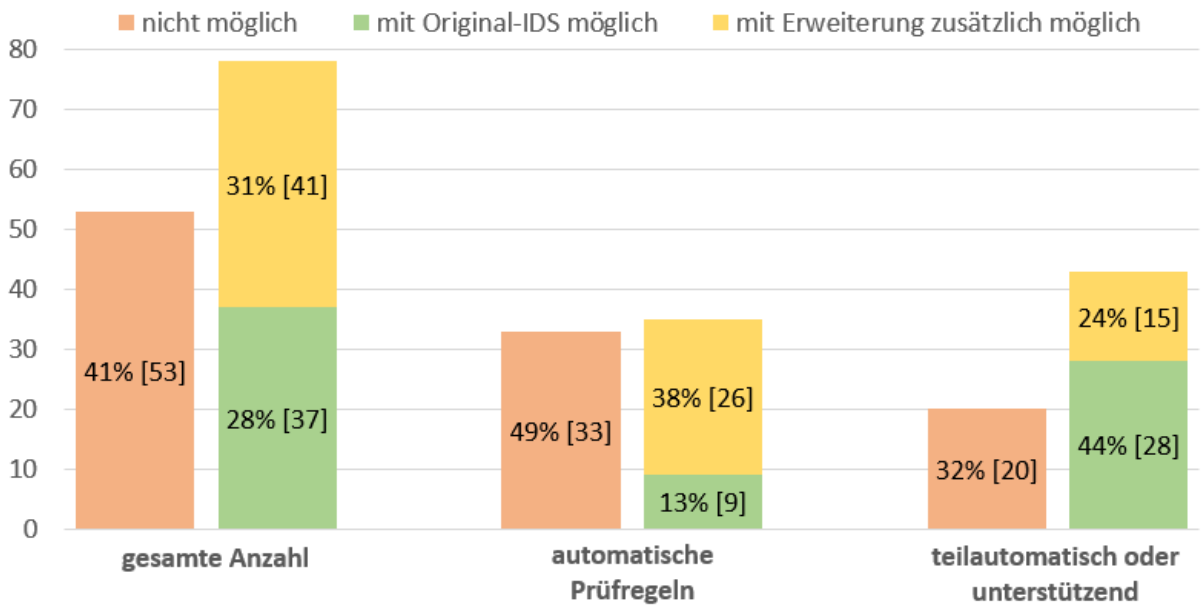

Abb. 6.2: Einteilung der untersuchten Prüfroutinen in die Umsetzbarkeit mit IDS

Ungefähr 40 % der bestehenden Prüfregeln können nicht oder nur mit weiterem großen Programmieraufwand mit IDS umgesetzt werden. IDS dient hauptsächlich zur Prüfung von alphanumerischer Information, die in der Regel in Properties und Attributen gespeichert ist. Geometrische sowie topologische Informationen können mit der Original-IDS nicht abgedeckt werden. Hintergründe sowie eine mögliche Erweiterung dafür untersuchten Kremer und Beetz [19]. Eine von ihnen beschriebene Lösung sieht die Einbindung externer Programme in IDS vor. Dafür wird in der XSD-Datei ein eigenes Facet eingeführt. Dieses enthält die notwendigen Informationen für die externen Programme (z. B. min/max-Angaben, Entitäten) sowie den URI (Uniform Resource Identifier) zum Aufruf der Programme. Somit werden die eigentliche Berechnung und Verwendung der Geometrie extern verwendet und IDS nicht direkt um diese Funktionalität erweitert.

Tab. 6.1 listet die Probleme, welche zu den 53 nicht möglichen Regeln führen und in dieser Arbeit nicht gelöst wurden. Zusätzlich werden die absolute Anzahl angegeben und einige Beispiele aufgeführt. Ein paar Prüfroutinen scheitern an mehreren Problemen und wurden in der Tabelle dem maßgebenderen Fall zugeordnet. Wenn eine Zuordnung nicht möglich war, sind die Prüfroutinen in der Kategorie *Sonstige* zu finden. Weiters enthält diese Kategorie noch unterschiedliche Probleme, die teilweise nur einmal vorgekommen, und deshalb hier zusammengefasst wurden.

Tab. 6.1: Probleme mit IDS

Problemgrund	Anzahl	Beispiele
Abstandsmessung	7	Abstandsmessung zwischen Bauwerken oder zwischen Fenstern und brandabschnittsbildenden Bauteilen
Fehlende Relationen/Beziehungen	7	Verbindung zwischen Stütze und Raum oder Rauchfang und Raum
Zählung von Elementen	8	Zählung von Obergeschoßen, Räumen oder Aufzügen
Oberstes Geschoß	3	Bestimmung des obersten Geschoßes oder der Decke über dem Treppenhaus
Berechnungen	8	Berechnung verschiedener Flächengrößen bzw. Aufsummierung, Prozentrechnung
Geometrische Eigenschaften	13	Gebäudehöhe zu umgebendem Gelände, Geometrie von Treppen, Verwendung eines Wendekreises
Sonstiges	7	Öffnungsrichtung von Türen, Verwendung der Bauplatzgrenze

Kapitel 7

Fazit und Ausblick

Der derzeitige Stand von IDS deckt mit den sechs vorhandenen Facets bereits viele Anwendungsbereiche ab. Es können alle im IFC-Schema vorhandenen Entitäten verwendet und bezüglich deren Attribute, Properties, Material oder Klassifikation spezifiziert werden. Mit dem PartOf-Facet ist die Abbildung von einigen Beziehungen zwischen einzelnen Elementen möglich. In Kapitel 3 wurden zwei Anwendungsmöglichkeiten von IDS bei der Prüfung von Properties gezeigt. Einerseits kann es für einen LOI-Check vor einer geometrisch anspruchsvollen Prüfung eingesetzt werden, wie der Fluchtwegsanalyse von Fischer et al. [11]. Andererseits findet IDS auch bei der Prüfung von Rechtsmaterie in Form der OIB-Richtlinie 2 Anwendung. Aus beiden Anwendungsbereichen haben sich einige Grenzen von IDS ergeben, die durch Erweiterungen in Kapitel 4 im Wesentlichen überwunden wurden. Diese Erweiterungen stehen großteils mit dem PartOf-Facet in Verbindung. Dadurch können nun an verbundene Elemente einzelne Anforderungen (z. B. Properties) spezifisch ausgedrückt werden. Weiters ermöglicht die eingeführte Relation *IfcRelSpaceBoundary* eine neue Beziehung in IDS zu nutzen. Anhand zweier BIM-Modelle wurden die Erweiterungen getestet und deren Funktionsfähigkeit bestätigt. Zusätzlich konnten aus der anschließenden Analyse weitere Erkenntnisse bezüglich der Umsetzung von Prüfregeln gezogen werden.

Ziel dieser Diplomarbeit war die Analyse des derzeitigen Stands von IDS sowie die Weiterentwicklung in Richtung eines offenen Formates für Prüfregeln. Zusammenfassend lässt sich sagen, dass IDS ein Potential zur Verwendung als Prüfstandard besitzt. Dabei muss jedoch stets auf die gewünschte zu prüfende Rechtsmaterie geachtet werden. Zum derzeitigen Stand (2023) beschränkt sich IDS nämlich eindeutig auf die Prüfung von alphanumerischer Information, wie sie in den bearbeiteten Stellen der OIB-Richtlinie 2 zu finden ist. Bei der Prüfung von geometrischen Eigenschaften kann IDS in den meisten Fällen nur als unterstützende Hilfe eingesetzt werden. Darunter ist zum Beispiel die graphische Hervorhebung zu untersuchender Elemente zu verstehen. In Kapitel 6 wurde auch eine entsprechende Einteilung vorgenommen. Diese hat aber gezeigt, dass ebenso bereits etablierte Software, wie zum Beispiel *Solibri*, nicht alle erdenklichen Regeln abdecken kann.

Die in dieser Arbeit erstellten Erweiterungen können teilweise mit der Arbeit von Kremer und Beetz [19] verglichen werden. In beiden Fällen wird die XSD-Datei um einen oder mehrere Tags ergänzt, der später die neuen Anweisungen an die Software weitergeben soll. Beide Erweiterungen beziehen sich auf die fehlende Verarbeitung topologischer Beziehungen und Informationen. In der vorliegenden Diplomarbeit wurde IDS um bestehende Teile des IFC-Schemas erweitert, damit die Applicability und Requirements spezifischer ausgeführt werden können. Im Gegensatz dazu bringen Kremer und Beetz [19] eine externe Software in Verbindung mit IDS, um geometrische Eigenschaften zu prüfen. Dennoch wird mit beiden Erweiterungen gezeigt, dass IDS zum derzeitigen Stand das Ende der Weiterentwicklung noch nicht erreicht hat.

Neben der Untersuchung von IDS wurde in dieser Arbeit auch der *IfcTester* von *IfcOpenShell* ausgiebig getestet, verwendet und erweitert. Schrödter und Mellenthin Filardo [33] untersuchten diesen sowie die Software *Solibri* mit Blick auf die Verwendung von IDS. Hierbei sticht der *IfcTester* beziehungsweise *IfcOpenShell* allgemein mit der großen Flexibilität hervor. Durch den verfügbaren

Programmcode können die eigenen Änderungen und Anpassungen schnell vorgenommen werden. Dadurch kann die Prüfsoftware stets auf dem zur jeweiligen XSD-Datei aktuellen Stand gehalten werden. Auch zusätzliche Erweiterungen sind jederzeit möglich, wie die vorliegende Arbeit zeigt. Als Nachteil ist vor allem die langsame Performance des *IfcTester* anzuführen. Im Unterschied dazu bietet *Solibri* bei der Verwendung laut Schrödter und Mellenthin Filardo [33] als kommerzielle und etablierte Software eine leistungsstarke Oberfläche und Benutzerfreundlichkeit. Als Nachteile werden die fehlende Flexibilität und der große Zeitaufwand zur Regelerstellung genannt.

Aus der Analyse in Kapitel 6 lassen sich weitere mögliche Erweiterungen und Handlungsempfehlungen zur Verwendung von IDS in der Zukunft ableiten. Die Erweiterungen können einerseits sehr speziell einzelne Punkte der OIB-Richtlinien betreffen oder allgemein bei der Nutzung von IDS Anwendung finden.

Die erste mögliche Erweiterung umfasst die Verfeinerung des derzeitigen Stands der XSD-Datei. Der neue Tag *multiObjectProcessing* ist derzeit nur als einfacher Typ in der XSD-Datei hinterlegt. Um die Benutzerfreundlichkeit zu erhöhen, könnte ein eigener komplexer Typ erstellt werden, welcher nur die drei in *IfcOpenShell* eingearbeiteten Schlüsselwörter zulässt. Damit wäre in der IDS-Datei keine Angabe von falschen Anweisungen möglich.

Eine zweite konkrete Ergänzung der aktuellen Erweiterung des PartOf-Facets wäre die Möglichkeit zur Angabe, wie viele Elemente mit einem gegebenen Element verbunden sein müssen. Beispielsweise wird in der OIB-Richtlinie 4 [29] gefordert, dass ein Raum mit mehr als 120 Personen maximaler Belegung zwei Fluchttüren besitzen muss. Mit der neu eingeführten Relation *IfcRelSpaceBoundary* kann von dem Raum bereits auf die Türen geschlossen werden. Durch die Verwendung der Attribute *min/maxOccurs* könnte zum Beispiel eine Vorgabe der Anzahl ermöglicht werden.

Weiters ist ein Beispiel für eine umsetzbare Erweiterung bei den bestehenden Relationen *IfcRelAggregates* und *IfcRelContainedInSpatialStructure* zu finden. Derzeit sind die beiden Relationen nur in eine Richtung verwendbar, also von dem jeweils untergeordneten Bauteil zum übergeordneten. Es kann zum Beispiel von einer Wand auf das zugehörige Geschoß geschlossen werden und nicht umgekehrt. Die Relation selbst funktioniert im IFC-Schema bereits in beide Richtungen und auch in *IfcOpenShell* werden zu der Relation die Elemente in beide Richtungen ausgewertet. Allein der *IfcTester* müsste dahingehend erweitert werden. Eine konkrete Anwendung für diese Erweiterung ist ebenfalls in OIB-Richtlinie 4 [29] zu finden. Diese schreibt für jedes Geschoß eine Treppe oder Rampe zur vertikalen Erschließung vor. Mit einer möglichen Erweiterung sollte dann gefordert werden können, dass in jedem Geschoß eine Treppe vorhanden sein muss.

Eine allgemeine denkbare Erweiterung ist bei der Verwendung der Schlüsselwörter von *multiObjectProcessing* zu finden. Wird im derzeitigen Stand ein Schlüsselwort nach einem oder mehreren Property-Facets angegeben, bezieht sich dieses Schlüsselwort stets auf alle Facets. Daher ist es nicht möglich, von zwei verbundenen Elementen zum Beispiel die Gleichheit in der Widmung und einen Unterschied in der Wohnungsnummer zu fordern. Eine Erweiterung könnte dies durch eine genaue Zuordnung des Schlüsselworts zu einem der angeführten Property-Facets ermöglichen.

Unabhängig von bereits durchgeführten oder noch möglichen Erweiterungen ist bei der Prüfung von Rechtsmaterie die Software stets durch eine Person der Baubehörde zu bedienen, welche auch endgültige Entscheidungen treffen muss. Wie die Testungen in Kapitel 5 gezeigt haben, treten teilweise unklare Fälle auf, wo die Software zum derzeitigen Stand keine eindeutigen Entscheidungen treffen kann. Ebenso ist die Entscheidung einer Person in solchen Fällen notwendig, wo auch bereits bestehende Prüfsoftware wie *Solibri* an ihre Grenzen stößt, wie es auch im Forschungsprojekt *BRISE-Vienna* schon beachtet wurde [40].

Ein ständiger Begleiter bei der Arbeit mit BIM sind die laufenden Verbesserungen und Entwicklungen. Dies gilt auch im Zusammenhang mit IDS. Einerseits wird IDS durch buildingSMART International stets weiterentwickelt und verbessert. Im ersten Halbjahr 2023 gab es alle paar

Wochen einen neuen Stand der XSD-Datei. Es ist daher nötig, stets am aktuellen Stand zu bleiben, unabhängig davon, ob man IDS anwendet oder es selbst weiterentwickelt. Andererseits ist auch das IFC-Schema stetigen größeren und kleineren Änderungen unterworfen. Wie beim Attribut *Elevation* erläutert, bringen neue IFC-Versionen veränderte Attribute und Properties mit sich. Somit können einmal erstellte IDS-Dateien nicht als starr betrachtet werden, sondern benötigen kontinuierliche Aktualisierungstätigkeiten an neue Standards. Nach Versionsupdates müssen alle enthaltenen Punkte mit dem neuen IFC-Schema verglichen werden. Die neuste Version des IFC-Schemas ist IFC 4.3 und soll wahrscheinlich Anfang 2024 als neue ISO-Norm erscheinen.

Abschließend ist festzuhalten, dass IDS eine gut geeignete Struktur zum maschinenlesbaren Abbilden von Informationsanforderungen ist. Somit kann es auch praktikabel zur Prüfung von Rechtsmaterie eingesetzt werden. Wichtig ist jedoch stets eine klare Abgrenzung der Möglichkeiten. Mit den derzeitigen Erweiterungen kann IDS nicht für geometrische Anforderungen, sondern nur für einfache Spezifikation bezüglich der Lage oder Verbindung einzelner Elemente verwendet werden. Diese Spezifikationen sind eher topologischer Natur und nur innerhalb der Grenzen der verfügbaren Relationen des IFC-Schemas möglich. Durch zusätzliche Erweiterungen können jedoch weitere Relationen in IDS eingebracht werden, um damit zusätzliche Beziehungen in den Spezifikationen auszudrücken. Wirkliche geometrische Anforderungen, wie zum Beispiel die Überprüfung eines Wendekreises oder die Berechnung der Absturzhöhe, sind schwierig mit Erweiterungen umsetzbar. Ab einem gewissen Ausmaß der Erweiterungen würde der Umfang der IDS-Dateien zu stark zunehmen und sie werden aufgrund des Verlusts ihrer klaren Struktur zu unübersichtlich. Mit den derzeitigen Erweiterungen hat sich die Struktur von IDS-Dateien kaum verändert, da nur die bestehenden Facets ergänzt wurden. Die Beibehaltung der grundlegenden Struktur sollte stets bei der Erweiterung berücksichtigt werden. Die Umsetzung von geometrischen Anforderungen scheint derzeit nur mit speziellen Softwarelösungen möglich zu sein.

Literatur

[1] ACCA software S.p.A. *usBIM. BIM-MANAGEMENT-SYSTEM*. 2023. URL: https://www.accasoftware.com/de/bim-management-system (Zugriff am 31. 08. 2023).

[2] Austrian Standards International. *Building Information Modeling*. URL: https://www.austrian-standards.at/de/themengebiete/bau-immobilien/building-information-modeling/alles-zu-bim (Zugriff am 11. 11. 2023).

[3] M. Becher. *XML: DTD, XML-Schema, XPath, XQuery, XSL-FO, SAX, DOM*. 2. Auflage. Wiesbaden [Heidelberg]: Springer Vieweg, 2021. ISBN: 978-3-658-35434-3.

[4] A. Borrmann, M. König, C. Koch und J. Beetz. *Building information modeling: Technologische Grundlagen und industrielle Praxis*. 2., aktualisierte Auflage. Wiesbaden: Springer Vieweg, 2021. ISBN: 978-3-658-33361-4.

[5] buildingSMART International. *buildingSMART/IDS*. 2023. URL: https://github.com/buildingSMART/IDS (Zugriff am 22. 06. 2023).

[6] buildingSMART International. *IfcRelationship*. URL: https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/link/ifcrelationship.htm (Zugriff am 08. 09. 2023).

[7] buildingSMART International. *IfcSpaceBoundary*. URL: https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/link/ifcrelspaceboundary.htm (Zugriff am 08. 09. 2023).

[8] buildingSMART International. *Technical Roadmap buildingSMART*. Apr. 2020. URL: https://www.buildingsmart.org/about/technical-roadmap/ (Zugriff am 31. 08. 2023).

[9] C. C. Eichler, C. Schranz, T. Krischmann und H. Urban. *BIMcert Handbuch. Grundlagenwissen openBIM. Ausgabe 2023*. Niederfrohna: Mironde-Verlag, 2023. DOI: 10.34726/4161.

[10] S. Fischer, C. Schranz und H. Urban. „Bewertung von openBIM-Projekten: Indikatoren für die Nutzungsintensität von openBIM“. In: *Bauingenieur* 97 (6) (2022), S. 206–214. DOI: 10.37544/0005-6650-2022-06-66.

[11] S. Fischer, C. Schranz, H. Urban und D. Pfeiffer. „Automation of escape route analysis for BIM-based building code checking“. In: *Automation in Construction* 156 105092 (2023). DOI: 10.1016/j.autcon.2023.105092.

[12] A. Gerger, H. Urban und C. Schranz. „Augmented Reality for Building Authorities: A Use Case Study in Austria“. In: *Buildings* 13 (6) 1462 (2023). DOI: 10.3390/buildings13061462.

[13] G. Goger, M. Piskernik und H. Urban. *Potenziale der Digitalisierung im Bauwesen*. Studie. WKO, BMVIT, 2018.

[14] IfcOpenShell. *IfcOpenShell – The open source IFC toolkit and geometry engine*. 2023. URL: https://ifcopenshell.org/ (Zugriff am 31. 08. 2023).

[15] IfcOpenShell. *IfcOpenShell/IfcOpenShell*. 2023. URL: https://github.com/IfcOpenShell/IfcOpenShell (Zugriff am 22. 06. 2023).

[16] A. S. Ismail, K. N. Ali und N. A. Iahad. „A Review on BIM-based automated code compliance checking system“. In: *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*. Langkawi, 2017. DOI: 10.1109/ICRIIS.2017.8002486.

[17] K. Jeon, G. Lee, S. Kang, H. Roh, J. Jung, K. Lee und M. Baldwin. „A relational framework for smart information delivery manual (IDM) specifications“. In: *Advanced Engineering Informatics* 49 (2021). DOI: 10.1016/j.aei.2021.101319.

[18] B. Klusmann, Z. Meng, N. Kremer, A. Meins-Becker und M. Helmus. „BIM Based Information Delivery Controlling System“. In: *2020 Proceedings of the 37th ISARC*. Kitakyushu, 2020, S. 215–222. DOI: 10.22260/ISARC2020/0032.

[19] N. C. Kremer und J. Beetz. „Extending – information delivery specification – for linking distributed model checking services“. In: *Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference*. Heraklion, 2023. DOI: 10.35490/EC3.2023.266.

[20] T. Krischmann, H. Urban und C. Schranz. „Entwicklung eines openBIM-Bewilligungsverfahrens“. In: *Bauingenieur* 95 (9) (2020), S. 335–344. DOI: 10.37544/0005-6650-2020-09-61.

[21] M. Mellenthin Filardo, P. Debus, J. Melzner und H.-J. Bargstädt. „XML-based Automated Information Requirement Import to a Modelling Environment“. In: *EG-ICE 2023 Conference Papers*. London, 2023.

[22] M. Mellenthin Filardo, L. H. Wiesner, J. Melzner und H.-J. Bargstädt. „Automated supplement of information requirements for tendering data“. In: *Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference*. Heraklion, 2023. DOI: 10.35490/EC3.2023.247.

[23] Y. Miyauchi. *What is the power of IDS for the correct operation of IFC data?* URL: https://open.substack.com/pub/yoshiyukimiyauchi/p/what-is-the-power-of-ids-for-the?utm_campaign=post&utm_medium=web (Zugriff am 23. 09. 2023).

[24] *OIB-Richtlinie 2: Brandschutz*. Wien: Österreichisches Institut für Bautechnik, Apr. 2019.

[25] *OIB-Richtlinie 2: Brandschutz*. Wien: Österreichisches Institut für Bautechnik, Mai 2023.

[26] *OIB-Richtlinie 2.1: Brandschutz bei Betriebsbauten*. Wien: Österreichisches Institut für Bautechnik, Apr. 2019.

[27] *OIB-Richtlinie 2.2: Brandschutz bei Garagen, überdachten Stellplätzen und Parkdecks*. Wien: Österreichisches Institut für Bautechnik, Apr. 2019.

[28] *OIB-Richtlinie 2.3: Brandschutz bei Gebäuden mit einem Fluchtniveau von mehr als 22 m*. Wien: Österreichisches Institut für Bautechnik, Apr. 2019.

[29] *OIB-Richtlinie 4: Nutzungssicherheit und Barrierefreiheit*. Wien: Österreichisches Institut für Bautechnik, Apr. 2019.

[30] *ÖNORM EN ISO 16739-1:2020-11-01: Industry Foundation Classes (IFC) für den Datenaustausch in der Bauwirtschaft und im Anlagenmanagement – Teil 1: Datenschema*. Wien: Austrian Standards, Nov. 2020.

[31] Österreichisches Institut für Bautechnik. *OIB-Richtlinien*. URL: https://www.oib.or.at/de/oib-richtlinien (Zugriff am 01. 09. 2023).

[32] C. Schranz, H. Urban und A. Gerger. „Potentials of Augmented Reality in a BIM based building submission process“. In: *Journal of Information Technology in Construction* 26 (2021), S. 441–457. DOI: 10.36680/j.itcon.2021.024.

[33] F. Schrödter und M. Mellenthin Filardo. „Open Source und BIM: Prüfung von IDS-basierten Informationsanforderungen anhand von Open-Source-Tools“. In: *34. Forum Bauinformatik*. Ruhr-Universität Bochum, 2023, S. 3–10. DOI: 10.13154/294-10109.

[34] Solibri Inc. *Solibri Office*. URL: https://www.solibri.com/solibri-office (Zugriff am 31. 08. 2023).

[35] Stadt Wien. *BRISE-VIENNA*. URL: https://digitales.wien.gv.at/projekt/brisevienna/ (Zugriff am 11. 11. 2023).

[36] A. Tomczak, L. van 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 092024 (2022). DOI: 10.1088/1755-1315/1101/9/092024.

[37] C. Trebbi, M. Cianciulli, F. Matarazzo, C. Mirarchi, G. Cianciulli und A. Pavan. „Clash Detection and Code Checking BIM Platform for the Italian Market“. In: *Digital Transformation of the Design, Construction and Management Processes of the Built Environment*. Hrsg. von B. Daniotti, M. Gianinetto und S. Della Torre. Cham: Springer International Publishing, 2020, S. 115–125. DOI: 10.1007/978-3-030-33570-0_11.

[38] UIA. -– Urban Innovative Actions. *BRISE-Vienna – Building Regulations Information for Submission Envolvement*. 2019. URL: https://www.uia-initiative.eu/en/uiacities/vienna-call4 (Zugriff am 31. 08. 2023).

[39] K. Ullah, E. Witt und I. Lill. „The BIM-Based Building Permit Process: Factors Affecting Adoption“. In: *Buildings* 12 (1) 45 (2022). DOI: 10.3390/buildings12010045.

[40] H. Urban, C. Schranz, T. Krischmann, H. Asmera und B. Pinter. „Einsatz von openBIM und KI im Bewilligungsverfahren der Stadt Wien“. In: *ÖIAZ – Österreichische Ingenieur- und Architektenzeitschrift* 166 (2021), S. 1–9.

[41] L. van Berlo und S. Fischer. „IDS – Information Delivery Specification“. In: *BIMcert Handbuch: Grundlagenwissen openBIM. Ausgabe 2023*. Niederfrohna: Mironde-Verlag, 2023, S. 102–115. DOI: 10.34726/4843.

[42] L. van Berlo, T. Krijnen, H. Tauscher, T. Liebich, A. van Kranenburg und P. Paasiala. „Future of the Industry Foundation Classes: towards IFC 5“. In: *Proceedings of the 38th International Conference of CIB W78* (2021), S. 123–137.

[43] L. van Berlo, P. Willems und P. Pauwels. „Creating information delivery specifications using linked data“. In: *Proceedings of the 36th International Conference of CIB W78* (2019), S. 637–650. DOI: 1854/LU-8633671.

Anhang A

ids.xsd-Datei inklusive Erweiterung

```
<!-- June 20, 2023 - DRAFT  -->
<xs:schema xmlns:ids="http://standards.buildingsmart.org/IDS" xmlns:xs="
    http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
    XMLSchema-instance" xmlns:altova="http://www.altova.com/
    xml-schema-extensions" targetNamespace="http://standards.buildingsmart.
    org/IDS" elementFormDefault="qualified" attributeFormDefault="
    unqualified" version="0.9.6">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xs:import namespace="http://www.w3.org/2001/XMLSchema" schemaLocation="
        http://www.w3.org/2001/XMLSchema.xsd"/>
    <xs:import namespace="http://www.w3.org/2001/XMLSchema-instance"
        schemaLocation="http://www.w3.org/2001/XMLSchema-instance"/>
    <xs:element name="ids">
       <xs:complexType>
          <xs:sequence>
             <xs:element name="info">
                <xs:complexType>
                   <xs:sequence>
                      <xs:element name="title" type="xs:string"/>
                      <xs:element name="copyright" type="xs:string"
                          minOccurs="0"/>
                      <xs:element name="version" type="xs:string" minOccurs=
                          "0"/>
                      <xs:element name="description" type="xs:string"
                          minOccurs="0"/>
                      <xs:element name="author" minOccurs="0">
                         <xs:simpleType>
                            <xs:restriction base="xs:string">
                               <xs:pattern value="[^@]+@[^\.]+\..+"/>
                            </xs:restriction>
                         </xs:simpleType>
                      </xs:element>
                      <xs:element name="date" type="xs:date" minOccurs="0"/>
                      <xs:element name="purpose" type="xs:string" minOccurs=
                          "0"/>
                      <xs:element name="milestone" type="xs:string"
                          minOccurs="0"/>
                   </xs:sequence>
                </xs:complexType>
             </xs:element>
             <xs:element name="specifications" type="ids:specificationsType"
                 />
          </xs:sequence>
       </xs:complexType>
    </xs:element>
    <xs:complexType name="entityType">
```

```
34          <xs:sequence>
35             <xs:element name="name" type="ids:idsValue"/>
36             <xs:element name="predefinedType" type="ids:idsValue" minOccurs="0
                   "/>
37          </xs:sequence>
38       </xs:complexType>
39       <xs:complexType name="idsValue">
40          <xs:choice minOccurs="1">
41             <!-- place for potential additional rules for idsValue -->
42             <xs:element name="simpleValue" type="xs:string" minOccurs="1"
                   maxOccurs="1"/>
43             <xs:element ref="xs:restriction" minOccurs="1" maxOccurs="
                   unbounded"/>
44          </xs:choice>
45       </xs:complexType>
46       <xs:complexType name="classificationType">
47          <xs:sequence>
48             <xs:element name="value" type="ids:idsValue" minOccurs="0"/>
49             <xs:element name="system" type="ids:idsValue" minOccurs="0"/>
50          </xs:sequence>
51       </xs:complexType>
52       <xs:complexType name="partOfType">
53          <xs:sequence>
54             <xs:element name="entity" type="ids:entityType" minOccurs="1"/>
55             <!-- Extension Patrick Loibl -->
56             <xs:element name="attribute" type="ids:attributeType" minOccurs="0
                   " maxOccurs="unbounded"/>
57             <!-- Extension Patrick Loibl -->
58             <xs:element name="property" type="ids:propertyType" minOccurs="0"
                   maxOccurs="unbounded"/>
59             <!-- Extension Patrick Loibl -->
60             <xs:element name="multiObjectProcessing" type="ids:idsValue"
                   minOccurs="0" maxOccurs="1"/>
61             <!-- Extension Patrick Loibl -->
62             <xs:element name="partOf" type="ids:partOfType" minOccurs="0"
                   maxOccurs="unbounded"/>
63          </xs:sequence>
64          <xs:attribute name="relation" type="ids:relations" />
65       </xs:complexType>
66       <xs:complexType name="applicabilityType">
67          <xs:sequence>
68             <xs:element name="entity" type="ids:entityType" minOccurs="0"/>
69             <xs:element name="partOf" type="ids:partOfType" minOccurs="0"
                   maxOccurs="unbounded"/>
70             <xs:element name="classification" type="ids:classificationType"
                   minOccurs="0" maxOccurs="unbounded"/>
71             <xs:element name="attribute" type="ids:attributeType" minOccurs="0
                   " maxOccurs="unbounded"/>
72             <xs:element name="property" type="ids:propertyType" minOccurs="0"
                   maxOccurs="unbounded"/>
73             <xs:element name="material" type="ids:materialType" minOccurs="0"
                   />
74          </xs:sequence>
75       </xs:complexType>
76       <xs:complexType name="propertyType">
77          <xs:sequence>
78             <xs:element name="propertySet" type="ids:idsValue"/>
79             <xs:element name="name" type="ids:idsValue"/>
80             <xs:element name="value" type="ids:idsValue" minOccurs="0"/>
81          </xs:sequence>
```

```
82          <xs:attribute name="datatype" use="required">
83             <xs:annotation>
84                <xs:documentation>This is the name of an IFC Defined Type. See
                      the full list for IFC 4 on https://standards.buildingsmart.
                      org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/link/
                      alphabeticalorder-defined-types.htm  Documentation and
                      default units on https://github.com/buildingSMART/IDS/blob/
                      master/Documentation/units.md</xs:documentation>
85             </xs:annotation>
86             <!-- renamed 'measure' to data type to better represent reality --
                   >
87          </xs:attribute>
88       </xs:complexType>
89       <xs:complexType name="attributeType">
90          <xs:sequence>
91             <xs:element name="name" type="ids:idsValue"/>
92             <xs:element name="value" type="ids:idsValue" minOccurs="0"/>
93          </xs:sequence>
94       </xs:complexType>
95       <xs:complexType name="materialType">
96          <xs:sequence>
97             <xs:element name="value" type="ids:idsValue" minOccurs="0"/>
98          </xs:sequence>
99       </xs:complexType>
100      <xs:complexType name="requirementsType">
101         <xs:sequence maxOccurs="unbounded">
102            <xs:element name="entity" minOccurs="0">
103               <xs:annotation>
104                  <xs:documentation>Make sure 'Name' value of requirements
                         entity is the same as the 'applicability' node, or a
                         wildcard (inclusive pattern).</xs:documentation>
105               </xs:annotation>
106               <xs:complexType>
107                  <xs:complexContent>
108                     <xs:extension base="ids:entityType">
109                        <xs:attribute name="instructions" type="xs:string" use
                              ="optional">
110                           <xs:annotation>
111                              <xs:documentation>Author of the IDS can leave
                                    instructions for the authors of the IFC.
                                    This text could/should be displayed in the
                                    BIM/IFC authoring tool.</xs:documentation>
112                           </xs:annotation>
113                        </xs:attribute>
114                     </xs:extension>
115                  </xs:complexContent>
116               </xs:complexType>
117            </xs:element>
118            <xs:element name="partOf" minOccurs="0" maxOccurs="unbounded">
119               <xs:complexType>
120                  <xs:complexContent>
121                     <xs:extension base="ids:partOfType">
122                        <xs:attributeGroup ref="xs:occurs"/>
123                        <xs:attribute name="instructions" type="xs:string" use
                              ="optional">
124                           <xs:annotation>
125                              <xs:documentation>Author of the IDS can leave
                                    instructions for the authors of the IFC.
                                    This text could/should be displayed in the
                                    BIM/IFC authoring tool.</xs:documentation>
```

```
126                     </xs:annotation>
127                  </xs:attribute>
128               </xs:extension>
129            </xs:complexContent>
130         </xs:complexType>
131      </xs:element>
132      <xs:element name="classification" minOccurs="0" maxOccurs="
             unbounded">
133         <xs:complexType>
134            <xs:complexContent>
135               <xs:extension base="ids:classificationType">
136                  <xs:attribute name="uri" type="xs:anyURI" use="
                       optional"/>
137                  <xs:attributeGroup ref="xs:occurs"/>
138                  <xs:attribute name="instructions" type="xs:string" use
                       ="optional">
139                     <xs:annotation>
140                        <xs:documentation>Author of the IDS can leave
                             instructions for the authors of the IFC.
                             This text could/should be displayed in the
                             BIM/IFC authoring tool.</xs:documentation>
141                     </xs:annotation>
142                  </xs:attribute>
143               </xs:extension>
144            </xs:complexContent>
145         </xs:complexType>
146      </xs:element>
147      <xs:element name="attribute" minOccurs="0" maxOccurs="unbounded">
148         <xs:complexType>
149            <xs:complexContent>
150               <xs:extension base="ids:attributeType">
151                  <xs:attribute name="instructions" type="xs:string" use
                       ="optional">
152                     <xs:annotation>
153                        <xs:documentation>Author of the IDS can leave
                             instructions for the authors of the IFC.
                             This text could/should be displayed in the
                             BIM/IFC authoring tool.</xs:documentation>
154                     </xs:annotation>
155                  </xs:attribute>
156               </xs:extension>
157            </xs:complexContent>
158         </xs:complexType>
159      </xs:element>
160      <xs:element name="property" minOccurs="0" maxOccurs="unbounded">
161         <xs:complexType>
162            <xs:complexContent>
163               <xs:extension base="ids:propertyType">
164                  <xs:attribute name="uri" type="xs:anyURI" use="
                       optional"/>
165                  <xs:attributeGroup ref="xs:occurs"/>
166                  <xs:attribute name="instructions" type="xs:string" use
                       ="optional">
167                     <xs:annotation>
168                        <xs:documentation>Author of the IDS can leave
                             instructions for the authors of the IFC.
                             This text could/should be displayed in the
                             BIM/IFC authoring tool.</xs:documentation>
169                     </xs:annotation>
170                  </xs:attribute>
```

```
171                  </xs:extension>
172               </xs:complexContent>
173            </xs:complexType>
174         </xs:element>
175         <xs:element name="material" minOccurs="0">
176            <xs:complexType>
177               <xs:complexContent>
178                  <xs:extension base="ids:materialType">
179                     <xs:attribute name="uri" type="xs:anyURI" use="
                           optional"/>
180                     <xs:attributeGroup ref="xs:occurs"/>
181                     <xs:attribute name="instructions" type="xs:string" use
                           ="optional">
182                        <xs:annotation>
183                           <xs:documentation>Author of the IDS can leave
                                 instructions for the authors of the IFC.
                                 This text could/should be displayed in the
                                 BIM/IFC authoring tool.</xs:documentation>
184                        </xs:annotation>
185                     </xs:attribute>
186                  </xs:extension>
187               </xs:complexContent>
188            </xs:complexType>
189         </xs:element>
190      </xs:sequence>
191   </xs:complexType>
192   <xs:complexType name="specificationType">
193      <xs:sequence>
194         <xs:element name="applicability" type="ids:applicabilityType"/>
195         <xs:element name="requirements" minOccurs="0">
196            <xs:complexType>
197               <xs:complexContent>
198                  <xs:extension base="ids:requirementsType">
199                     <xs:attribute name="description" type="xs:string" use=
                           "optional"/>
200                  </xs:extension>
201               </xs:complexContent>
202            </xs:complexType>
203         </xs:element>
204      </xs:sequence>
205      <xs:attribute name="name" type="xs:string" use="required"/>
206      <xs:attributeGroup ref="xs:occurs"/>
207      <xs:attribute name="ifcVersion" use="required">
208         <xs:simpleType>
209            <xs:list>
210               <xs:simpleType>
211                  <xs:restriction base="xs:string">
212                     <xs:minLength value="1"/>
213                     <xs:enumeration value="IFC2X3"/>
214                     <xs:enumeration value="IFC4"/>
215                     <xs:enumeration value="IFC4X3"/>
216                  </xs:restriction>
217               </xs:simpleType>
218            </xs:list>
219         </xs:simpleType>
220      </xs:attribute>
221      <xs:attribute name="identifier" type="xs:string">
222         <xs:annotation>
223            <xs:documentation>Author of the IDS can provide an identifier
                  to the specification. This is intended to be a machine
```

```
                readable identifier. Beware: because of the possibility to
                combine different 'requirement' elements from several ids
                files this cannot be enforced/assumed as (global) unique.</
                xs:documentation>
224          </xs:annotation>
225       </xs:attribute>
226       <xs:attribute name="description" type="xs:string" use="optional"/>
227       <xs:attribute name="instructions" type="xs:string" use="optional">
228          <xs:annotation>
229             <xs:documentation>Author of the IDS can leave instructions for
                the authors of the IFC. This text could/should be displayed
                 in the BIM/IFC authoring tool.</xs:documentation>
230          </xs:annotation>
231       </xs:attribute>
232    </xs:complexType>
233    <xs:complexType name="specificationsType">
234       <xs:sequence>
235          <xs:element name="specification" type="ids:specificationType"
              minOccurs="1" maxOccurs="unbounded"/>
236       </xs:sequence>
237    </xs:complexType>
238    <xs:simpleType name="relations">
239       <xs:restriction base="xs:string">
240          <xs:enumeration value="IFCRELAGGREGATES"/>
241          <xs:enumeration value="IFCRELASSIGNSTOGROUP"/>
242          <xs:enumeration value="IFCRELCONTAINEDINSPATIALSTRUCTURE"/>
243          <xs:enumeration value="IFCRELNESTS"/>
244          <xs:enumeration value="IFCRELVOIDSELEMENT"/>
245          <xs:enumeration value="IFCRELFILLSELEMENT"/>
246          <!-- Extension Patrick Loibl -->
247          <xs:enumeration value="IFCRELSPACEBOUNDARY"/>
248       </xs:restriction>
249    </xs:simpleType>
250 </xs:schema>
```

Anhang B

IDS-Datei für den LOI-Check der Fluchtwegsanalyse

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- IDS (INFORMATION DELIVERY SPECIFICATION) CREATED USING IFCOPENSHELL -->
<ids xmlns="http://standards.buildingsmart.org/IDS" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://standards.buildingsmart.org/IDS/ids_05.xsd">
    <info>
        <title>IDS for Escape Route Analysis</title>
        <copyright>Patrick Loibl</copyright>
        <description>IDS for Escape Route Analysis</description>
        <date>2023-05-19</date>
    </info>
    <specifications>
        <specification minOccurs="0" name="Anforderungen an alle IfcSpace" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCSPACE</simpleValue>
                    </name>
                </entity>
            </applicability>
            <requirements>
                <property datatype ="IfcLabel" minOccurs="1" maxOccurs="1">
                    <propertySet>
                        <simpleValue>Pset_SpaceSpecific</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>WidmungBehoerde</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="ND"/>
                            <xs:enumeration value="Sanitär- und Umkleideräume"/>
                            <xs:enumeration value="Wohnen und Aufenthalt"/>
                            <xs:enumeration value="Büroarbeit"/>
                            <xs:enumeration value="Produktion, Hand- und Maschinenarbeit, Experimente"/>
                            <xs:enumeration value="Lagern, Verteilen und Verkaufen"/>
                            <xs:enumeration value="Bildung, Unterricht und Kultur"/>
                            <xs:enumeration value="Heilen und Pflegen"/>
```

```
                        <xs:enumeration value="Sonstige Nutzungen"/>
                        <xs:enumeration value="Betriebstechnische
                            Anlagen"/>
                        <xs:enumeration value="Verkehrserschliessung
                            und -sicherung"/>
                        <xs:enumeration value="Loggien"/>
                        <xs:enumeration value="Terrassen, Balkone"/>
                        <xs:enumeration value="Dachböden"/>
                        <xs:enumeration value="Lufträume"/>
                        <xs:enumeration value="Kellerräume mindere
                            Qualität"/>
                        <xs:enumeration value="Unverwendbare Grundflä
                            che"/>
                    </xs:restriction>
                </value>
            </property>
            <property datatype="IfcCountMeasure" minOccurs="1"
                maxOccurs="1">
                <propertySet>
                    <simpleValue>Pset_SpaceOccupancyRequirements</
                        simpleValue>
                </propertySet>
                <name>
                    <simpleValue>OccupancyNumberPeak</simpleValue>
                </name>
                <value>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="0" />
                    </xs:restriction>
                </value>
            </property>
        </requirements>
    </specification>
    <specification minOccurs="0" name="Anforderung IfcSpace ParkingSlot
        " ifcVersion="IFC4">
        <applicability>
            <entity>
                <name>
                    <simpleValue>IFCSPACE</simpleValue>
                </name>
            </entity>
            <property datatype ="IfcLabel" >
                <propertySet>
                    <simpleValue>Pset_SpaceSpecific</simpleValue>
                </propertySet>
                <name>
                    <simpleValue>WidmungBehoerde</simpleValue>
                </name>
                <value>
                    <simpleValue>Verkehrserschliessung und -sicherung</
                        simpleValue>
                </value>
            </property>
        </applicability>
        <requirements>
            <property datatype="IfcBoolean" minOccurs="1" maxOccurs
                ="1">
                <propertySet>
                    <simpleValue>ZDB_Space_EscapeRouteAnalysis</
                        simpleValue>
```

```
87                      </propertySet>
88                      <name>
89                          <simpleValue>ParkingSlot</simpleValue>
90                      </name>
91                  </property>
92              </requirements>
93          </specification>
94          <specification minOccurs="1" name="Anforderungen IfcDoor an
                mindestens eine Tür (IDDoor und IDNextDoor)" ifcVersion="IFC4">
95              <applicability>
96                  <entity>
97                      <name>
98                          <simpleValue>IFCDOOR</simpleValue>
99                      </name>
100                 </entity>
101                 <property datatype="IfcIdentifier">
102                     <propertySet>
103                         <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                                simpleValue>
104                     </propertySet>
105                     <name>
106                         <simpleValue>IDDoor</simpleValue>
107                     </name>
108                 </property>
109                 <property datatype="IfcIdentifier">
110                     <propertySet>
111                         <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                                simpleValue>
112                     </propertySet>
113                     <name>
114                         <simpleValue>IDNextDoor</simpleValue>
115                     </name>
116                 </property>
117             </applicability>
118             <!-- Folgender Bereich ist eigentlich überflüssig!-->
119             <!-- <requirements>
120                 <property datatype="IfcIdentifier">
121                     <propertySet>
122                         <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                                simpleValue>
123                     </propertySet>
124                     <name>
125                         <simpleValue>IDDoor</simpleValue>
126                     </name>
127                 </property>
128                 <property datatype="IfcIdentifier">
129                     <propertySet>
130                         <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                                simpleValue>
131                     </propertySet>
132                     <name>
133                         <simpleValue>IDNextDoor</simpleValue>
134                     </name>
135                 </property>
136             </requirements> -->
137         </specification>
138         <specification minOccurs="0" name="Anforderungen an alle IfcDoor (
                DoorType)" ifcVersion="IFC4">
139             <applicability>
140                 <entity>
```

```
                    <name>
                        <simpleValue>IFCDOOR</simpleValue>
                    </name>
                </entity>
            </applicability>
            <requirements>
                <property datatype="IfcLabel" minOccurs="1" maxOccurs="1">
                    <propertySet>
                        <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                            simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>DoorType</simpleValue>
                    </name>
                </property>
            </requirements>
        </specification>
        <specification minOccurs="0" name="Anforderungen IfcDoor Eingangstü
            r (OccupanyNumberFlat)" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCDOOR</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELSPACEBOUNDARY">
                    <entity>
                        <name>
                            <simpleValue>IFCSPACE</simpleValue>
                        </name>
                    </entity>
                    <property datatype="IfcLabel" >
                        <propertySet>
                            <simpleValue>Pset_SpaceSpecific</simpleValue>
                        </propertySet>
                        <name>
                            <simpleValue>WidmungBehoerde</simpleValue>
                        </name>
                        <value>
                            <simpleValue>Wohnen und Aufenthalt</simpleValue
                                >
                        </value>
                    </property>
                </partOf>
                <property datatype="IfcLabel" >
                    <propertySet>
                        <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                            simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>DoorType</simpleValue>
                    </name>
                    <value>
                        <simpleValue>Entry</simpleValue>
                    </value>
                </property>
            </applicability>
            <requirements>
                <property datatype="IfcCountMeasure" minOccurs="1"
                    maxOccurs="1">
```

```
196                         <propertySet>
197                             <simpleValue>ZDB_Door_EscapeRouteAnalysis</
                                    simpleValue>
198                         </propertySet>
199                         <name>
200                             <simpleValue>OccupancyNumberFlat</simpleValue>
201                         </name>
202                         <value>
203                             <xs:restriction base="xs:integer">
204                                 <xs:minExclusive value="0" />
205                             </xs:restriction>
206                         </value>
207                     </property>
208                 </requirements>
209             </specification>
210         </specifications>
211 </ids>
```

Anhang C

IDS-Datei für die Tabelle 1b der OIB-Richtlinie 2

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- IDS (INFORMATION DELIVERY SPECIFICATION) CREATED USING IFCOPENSHELL -->
<ids xmlns="http://standards.buildingsmart.org/IDS" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://standards.buildingsmart.org/IDS http://standards.buildingsmart.org/IDS/0.9.6/ids.xsd">
    <info>
        <title>IDS for OIB 2 Tabelle 2a</title>
        <copyright>Patrick Loibl</copyright>
        <description>IDS for OIB 2 Tabelle 2a</description>
        <date>2023-05-19</date>
    </info>
    <specifications>
        <!-- 1.1 Die bestimmung der tragenden Bauteile im obersten Geschoß ist nicht möglich!-->
        <specification minOccurs="0" name="1.2.1 Tragende Bauteile in sonstigen oberirdischen Geschoßen - Wände" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCWALL</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
                    <entity>
                        <name>
                            <simpleValue>IFCBUILDINGSTOREY</simpleValue>
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDING</simpleValue>
                            </name>
                        </entity>
                        <property datatype="IfcLabel">
                            <propertySet>
                                <simpleValue>ZDB_Building_Infos</simpleValue>
                            </propertySet>
                            <name>
                                <simpleValue>Gebaeudeklasse</simpleValue>
                            </name>
                            <value>
```

```
                                <simpleValue>GK2</simpleValue>
                            </value>
                        </property>
                    </partOf>
                    <attribute>
                        <name>
                            <simpleValue>Elevation</simpleValue>
                        </name>
                        <value>
                            <xs:restriction base="xs:integer">
                                <xs:minInclusive value="0" />
                            </xs:restriction>
                        </value>
                    </attribute>
                </partOf>
                <property datatype="IfcBoolean">
                    <propertySet>
                        <simpleValue>Pset_WallCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>Compartmentation</simpleValue>
                    </name>
                    <value>
                        <simpleValue>FALSE</simpleValue>
                    </value>
                </property>
            </applicability>
            <requirements>
                <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                    ="1">
                    <propertySet>
                        <simpleValue>Pset_WallCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>FireRating</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="R180"/>
                            <xs:enumeration value="R120"/>
                            <xs:enumeration value="R90"/>
                            <xs:enumeration value="R60"/>
                            <xs:enumeration value="R30"/>
                        </xs:restriction>
                    </value>
                </property>
            </requirements>
        </specification>
        <specification minOccurs="0" name="1.2.2 Tragende Bauteile in
            sonstigen oberirdischen Geschoßen - Träger" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCBEAM</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
                    <entity>
                        <name>
                            <simpleValue>IFCSPACE</simpleValue>
```

```
                    </name>
                </entity>
                <partOf relation="IFCRELAGGREGATES">
                    <entity>
                        <name>
                            <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                >
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDING</simpleValue>
                            </name>
                        </entity>
                        <property datatype="IfcLabel">
                            <propertySet>
                                <simpleValue>ZDB_Building_Infos</
                                    simpleValue>
                            </propertySet>
                            <name>
                                <simpleValue>Gebaeudeklasse</
                                    simpleValue>
                            </name>
                            <value>
                                <simpleValue>GK2</simpleValue>
                            </value>
                        </property>
                    </partOf>
                    <attribute>
                        <name>
                            <simpleValue>Elevation</simpleValue>
                        </name>
                        <value>
                            <xs:restriction base="xs:integer">
                                <xs:minInclusive value="0" />
                            </xs:restriction>
                        </value>
                    </attribute>
                </partOf>
            </partOf>
        </applicability>
        <requirements>
            <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                ="1">
                <propertySet>
                    <simpleValue>Pset_BeamCommon</simpleValue>
                </propertySet>
                <name>
                    <simpleValue>FireRating</simpleValue>
                </name>
                <value>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="R180"/>
                        <xs:enumeration value="R120"/>
                        <xs:enumeration value="R90"/>
                        <xs:enumeration value="R60"/>
                        <xs:enumeration value="R30"/>
                    </xs:restriction>
                </value>
```

```
153                 </property>
154             </requirements>
155         </specification>
156         <specification minOccurs="0" name="1.2.3 Tragende Bauteile in
                sonstigen oberirdischen Geschoßen - Stützen" ifcVersion="IFC4">
157             <applicability>
158                 <entity>
159                     <name>
160                         <simpleValue>IFCCOLUMN</simpleValue>
161                     </name>
162                 </entity>
163                 <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
164                     <entity>
165                         <name>
166                             <simpleValue>IFCBUILDINGSTOREY</simpleValue>
167                         </name>
168                     </entity>
169                     <partOf relation="IFCRELAGGREGATES">
170                         <entity>
171                             <name>
172                                 <simpleValue>IFCBUILDING</simpleValue>
173                             </name>
174                         </entity>
175                         <property datatype="IfcLabel">
176                             <propertySet>
177                                 <simpleValue>ZDB_Building_Infos</
                                        simpleValue>
178                             </propertySet>
179                             <name>
180                                 <simpleValue>Gebaeudeklasse</simpleValue>
181                             </name>
182                             <value>
183                                 <simpleValue>GK2</simpleValue>
184                             </value>
185                         </property>
186                     </partOf>
187                     <attribute>
188                         <name>
189                             <simpleValue>Elevation</simpleValue>
190                         </name>
191                         <value>
192                             <xs:restriction base="xs:integer">
193                                 <xs:minInclusive value="0" />
194                             </xs:restriction>
195                         </value>
196                     </attribute>
197                 </partOf>
198             </applicability>
199             <requirements>
200                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
201                     <propertySet>
202                         <simpleValue>Pset_ColumnCommon</simpleValue>
203                     </propertySet>
204                     <name>
205                         <simpleValue>FireRating</simpleValue>
206                     </name>
207                     <value>
208                         <xs:restriction base="xs:string">
209                             <xs:enumeration value="R180"/>
```

```
                            <xs:enumeration value="R120"/>
                            <xs:enumeration value="R90"/>
                            <xs:enumeration value="R60"/>
                            <xs:enumeration value="R30"/>
                        </xs:restriction>
                    </value>
                </property>
            </requirements>
        </specification>
        <specification minOccurs="0" name="1.3.1 Tragende Bauteile in
            unterirdischen Geschoßen - Wände" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCWALL</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
                    <entity>
                        <name>
                            <simpleValue>IFCBUILDINGSTOREY</simpleValue>
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDING</simpleValue>
                            </name>
                        </entity>
                        <property datatype="IfcLabel">
                            <propertySet>
                                <simpleValue>ZDB_Building_Infos</
                                    simpleValue>
                            </propertySet>
                            <name>
                                <simpleValue>Gebaeudeklasse</simpleValue>
                            </name>
                            <value>
                                <simpleValue>GK2</simpleValue>
                            </value>
                        </property>
                    </partOf>
                    <attribute>
                        <name>
                            <simpleValue>Elevation</simpleValue>
                        </name>
                        <value>
                            <xs:restriction base="xs:integer">
                                <xs:maxExclusive value="0" />
                            </xs:restriction>
                        </value>
                    </attribute>
                </partOf>
                <property datatype="IfcBoolean">
                    <propertySet>
                        <simpleValue>Pset_WallCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>Compartmentation</simpleValue>
                    </name>
```

```
268                     <value>
269                         <simpleValue>FALSE</simpleValue>
270                     </value>
271                 </property>
272             </applicability>
273             <requirements>
274                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
275                     <propertySet>
276                         <simpleValue>Pset_WallCommon</simpleValue>
277                     </propertySet>
278                     <name>
279                         <simpleValue>FireRating</simpleValue>
280                     </name>
281                     <value>
282                         <xs:restriction base="xs:string">
283                             <xs:enumeration value="R180"/>
284                             <xs:enumeration value="R120"/>
285                             <xs:enumeration value="R90"/>
286                             <xs:enumeration value="R60"/>
287                             <xs:enumeration value="R30"/>
288                         </xs:restriction>
289                     </value>
290                 </property>
291             </requirements>
292         </specification>
293         <specification minOccurs="0" name="1.3.2 Tragende Bauteile in
                unterirdischen Geschoßen - Träger" ifcVersion="IFC4">
294             <applicability>
295                 <entity>
296                     <name>
297                         <simpleValue>IFCBEAM</simpleValue>
298                     </name>
299                 </entity>
300                 <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
301                     <entity>
302                         <name>
303                             <simpleValue>IFCSPACE</simpleValue>
304                         </name>
305                     </entity>
306                     <partOf relation="IFCRELAGGREGATES">
307                         <entity>
308                             <name>
309                                 <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                        >
310                             </name>
311                         </entity>
312                         <partOf relation="IFCRELAGGREGATES">
313                             <entity>
314                                 <name>
315                                     <simpleValue>IFCBUILDING</simpleValue>
316                                 </name>
317                             </entity>
318                             <property datatype="IfcLabel">
319                                 <propertySet>
320                                     <simpleValue>ZDB_Building_Infos</
                                            simpleValue>
321                                 </propertySet>
322                                 <name>
```

```
323                                     <simpleValue>Gebaeudeklasse</
                                            simpleValue>
324                                 </name>
325                                 <value>
326                                     <simpleValue>GK2</simpleValue>
327                                 </value>
328                             </property>
329                         </partOf>
330                         <attribute>
331                             <name>
332                                 <simpleValue>Elevation</simpleValue>
333                             </name>
334                             <value>
335                                 <xs:restriction base="xs:integer">
336                                     <xs:maxExclusive value="0" />
337                                 </xs:restriction>
338                             </value>
339                         </attribute>
340                     </partOf>
341                 </partOf>
342             </applicability>
343             <requirements>
344                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
345                     <propertySet>
346                         <simpleValue>Pset_BeamCommon</simpleValue>
347                     </propertySet>
348                     <name>
349                         <simpleValue>FireRating</simpleValue>
350                     </name>
351                     <value>
352                         <xs:restriction base="xs:string">
353                             <xs:enumeration value="R180"/>
354                             <xs:enumeration value="R120"/>
355                             <xs:enumeration value="R90"/>
356                             <xs:enumeration value="R60"/>
357                             <xs:enumeration value="R30"/>
358                         </xs:restriction>
359                     </value>
360                 </property>
361             </requirements>
362         </specification>
363         <specification minOccurs="0" name="1.3.3 Tragende Bauteile in
                unterirdischen Geschoßen - Stützen" ifcVersion="IFC4">
364             <applicability>
365                 <entity>
366                     <name>
367                         <simpleValue>IFCCOLUMN</simpleValue>
368                     </name>
369                 </entity>
370                 <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
371                     <entity>
372                         <name>
373                             <simpleValue>IFCBUILDINGSTOREY</simpleValue>
374                         </name>
375                     </entity>
376                     <partOf relation="IFCRELAGGREGATES">
377                         <entity>
378                             <name>
379                                 <simpleValue>IFCBUILDING</simpleValue>
```

```
                        </name>
                    </entity>
                    <property datatype="IfcLabel">
                        <propertySet>
                            <simpleValue>ZDB_Building_Infos</
                                simpleValue>
                        </propertySet>
                        <name>
                            <simpleValue>Gebaeudeklasse</simpleValue>
                        </name>
                        <value>
                            <simpleValue>GK2</simpleValue>
                        </value>
                    </property>
                </partOf>
                <attribute>
                    <name>
                        <simpleValue>Elevation</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:integer">
                            <xs:maxExclusive value="0" />
                        </xs:restriction>
                    </value>
                </attribute>
            </partOf>
        </applicability>
        <requirements>
            <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                ="1">
                <propertySet>
                    <simpleValue>Pset_ColumnCommon</simpleValue>
                </propertySet>
                <name>
                    <simpleValue>FireRating</simpleValue>
                </name>
                <value>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="R180"/>
                        <xs:enumeration value="R120"/>
                        <xs:enumeration value="R90"/>
                        <xs:enumeration value="R60"/>
                        <xs:enumeration value="R30"/>
                    </xs:restriction>
                </value>
            </property>
        </requirements>
    </specification>
    <specification minOccurs="0" name="2.2 Trennwände in oberirdischen
        Geschoßen" ifcVersion="IFC4">
        <applicability>
            <entity>
                <name>
                    <simpleValue>IFCWALL</simpleValue>
                </name>
            </entity>
            <partOf relation="IFCRELSPACEBOUNDARY">
                <entity>
                    <name>
                        <simpleValue>IFCSPACE</simpleValue>
```

```
                    </name>
                </entity>
                <partOf relation="IFCRELAGGREGATES">
                    <entity>
                        <name>
                            <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                >
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDING</simpleValue>
                            </name>
                        </entity>
                        <property datatype="IfcLabel">
                            <propertySet>
                                <simpleValue>ZDB_Building_Infos</
                                    simpleValue>
                            </propertySet>
                            <name>
                                <simpleValue>Gebaeudeklasse</
                                    simpleValue>
                            </name>
                            <value>
                                <simpleValue>GK2</simpleValue>
                            </value>
                        </property>
                    </partOf>
                    <attribute>
                        <name>
                            <simpleValue>Elevation</simpleValue>
                        </name>
                        <value>
                            <xs:restriction base="xs:integer">
                                <xs:minInclusive value="0" />
                            </xs:restriction>
                        </value>
                    </attribute>
                </partOf>
                <property datatype="IfcInteger">
                    <propertySet>
                        <simpleValue>Pset_SpaceSpecific</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>Wohnungsnummer</simpleValue>
                    </name>
                </property>
                <multiObjectProcessing>
                    <simpleValue>differentPropertyValuesForAnyObject</
                        simpleValue>
                </multiObjectProcessing>
            </partOf>
        </applicability>
        <requirements>
            <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                ="1">
                <propertySet>
                    <simpleValue>Pset_WallCommon</simpleValue>
                </propertySet>
```

```
492                     <name>
493                         <simpleValue>FireRating</simpleValue>
494                     </name>
495                     <value>
496                         <xs:restriction base="xs:string">
497                             <xs:enumeration value="EI 30"/>
498                             <xs:enumeration value="EI 60"/>
499                             <xs:enumeration value="EI 90"/>
500                             <xs:enumeration value="EI 120"/>
501                             <xs:enumeration value="EI 180"/>
502                             <xs:enumeration value="REI 30"/>
503                             <xs:enumeration value="REI 60"/>
504                             <xs:enumeration value="REI 90"/>
505                             <xs:enumeration value="REI 120"/>
506                             <xs:enumeration value="REI 180"/>
507                         </xs:restriction>
508                     </value>
509                 </property>
510             </requirements>
511         </specification>
512         <specification minOccurs="0" name="2.3 Trennwände in unterirdischen
                Geschoßen" ifcVersion="IFC4">
513             <applicability>
514                 <entity>
515                     <name>
516                         <simpleValue>IFCWALL</simpleValue>
517                     </name>
518                 </entity>
519                 <partOf relation="IFCRELSPACEBOUNDARY">
520                     <entity>
521                         <name>
522                             <simpleValue>IFCSPACE</simpleValue>
523                         </name>
524                     </entity>
525                     <partOf relation="IFCRELAGGREGATES">
526                         <entity>
527                             <name>
528                                 <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                        >
529                             </name>
530                         </entity>
531                         <partOf relation="IFCRELAGGREGATES">
532                             <entity>
533                                 <name>
534                                     <simpleValue>IFCBUILDING</simpleValue>
535                                 </name>
536                             </entity>
537                             <property datatype="IfcLabel">
538                                 <propertySet>
539                                     <simpleValue>ZDB_Building_Infos</
                                            simpleValue>
540                                 </propertySet>
541                                 <name>
542                                     <simpleValue>Gebaeudeklasse</
                                            simpleValue>
543                                 </name>
544                                 <value>
545                                     <simpleValue>GK2</simpleValue>
546                                 </value>
547                             </property>
```

```
548                         </partOf>
549                         <attribute>
550                             <name>
551                                 <simpleValue>Elevation</simpleValue>
552                             </name>
553                             <value>
554                                 <xs:restriction base="xs:integer">
555                                     <xs:maxExclusive value="0" />
556                                 </xs:restriction>
557                             </value>
558                         </attribute>
559                     </partOf>
560                     <property datatype="IfcInteger">
561                         <propertySet>
562                             <simpleValue>Pset_SpaceSpecific</simpleValue>
563                         </propertySet>
564                         <name>
565                             <simpleValue>Wohnungsnummer</simpleValue>
566                         </name>
567                     </property>
568                     <multiObjectProcessing>
569                         <simpleValue>differentPropertyValuesForAnyObject</
                            simpleValue>
570                     </multiObjectProcessing>
571                 </partOf>
572             </applicability>
573             <requirements>
574                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
575                     <propertySet>
576                         <simpleValue>Pset_WallCommon</simpleValue>
577                     </propertySet>
578                     <name>
579                         <simpleValue>FireRating</simpleValue>
580                     </name>
581                     <value>
582                         <xs:restriction base="xs:string">
583                             <xs:enumeration value="EI 60"/>
584                             <xs:enumeration value="EI 90"/>
585                             <xs:enumeration value="EI 120"/>
586                             <xs:enumeration value="EI 180"/>
587                             <xs:enumeration value="REI 60"/>
588                             <xs:enumeration value="REI 90"/>
589                             <xs:enumeration value="REI 120"/>
590                             <xs:enumeration value="REI 180"/>
591                         </xs:restriction>
592                     </value>
593                 </property>
594             </requirements>
595         </specification>
596         <!-- 3. Nicht möglich, da auf die Grundstücks- und Bauplatzgrenze
                nicht zugegriffen werden kann. -->
597         <!-- 4.1 & 4.2 Decken über dem obersten Geschoß sind nicht
                erfassbar. -->
598         <specification minOccurs="0" name="4.3 Trenndecken in oberirdischen
                 Geschoßen" ifcVersion="IFC4">
599             <applicability>
600                 <entity>
601                     <name>
602                         <simpleValue>IFCSLAB</simpleValue>
```

```
603                     </name>
604                 </entity>
605                 <partOf relation="IFCRELSPACEBOUNDARY">
606                     <entity>
607                         <name>
608                             <simpleValue>IFCSPACE</simpleValue>
609                         </name>
610                     </entity>
611                     <partOf relation="IFCRELAGGREGATES">
612                         <entity>
613                             <name>
614                                 <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                    >
615                             </name>
616                         </entity>
617                         <partOf relation="IFCRELAGGREGATES">
618                             <entity>
619                                 <name>
620                                     <simpleValue>IFCBUILDING</simpleValue>
621                                 </name>
622                             </entity>
623                             <property datatype="IfcLabel">
624                                 <propertySet>
625                                     <simpleValue>ZDB_Building_Infos</
                                        simpleValue>
626                                 </propertySet>
627                                 <name>
628                                     <simpleValue>Gebaeudeklasse</
                                        simpleValue>
629                                 </name>
630                                 <value>
631                                     <simpleValue>GK2</simpleValue>
632                                 </value>
633                             </property>
634                         </partOf>
635                         <attribute>
636                             <name>
637                                 <simpleValue>Elevation</simpleValue>
638                             </name>
639                             <value>
640                                 <xs:restriction base="xs:integer">
641                                     <xs:minInclusive value="0" />
642                                 </xs:restriction>
643                             </value>
644                         </attribute>
645                     </partOf>
646                     <property datatype="IfcInteger">
647                         <propertySet>
648                             <simpleValue>Pset_SpaceSpecific</simpleValue>
649                         </propertySet>
650                         <name>
651                             <simpleValue>Wohnungsnummer</simpleValue>
652                         </name>
653                     </property>
654                     <multiObjectProcessing>
655                         <simpleValue>differentPropertyValuesForAnyObject</
                            simpleValue>
656                     </multiObjectProcessing>
657                 </partOf>
658                 <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
```

```
659                     <entity>
660                         <name>
661                             <simpleValue>IFCBUILDINGSTOREY</simpleValue>
662                         </name>
663                     </entity>
664                     <attribute>
665                         <name>
666                             <simpleValue>Elevation</simpleValue>
667                         </name>
668                         <value>
669                             <xs:restriction base="xs:integer">
670                                 <xs:minInclusive value="0" />
671                             </xs:restriction>
672                         </value>
673                     </attribute>
674                 </partOf>
675             </applicability>
676             <requirements>
677                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
678                     <propertySet>
679                         <simpleValue>Pset_SlabCommon</simpleValue>
680                     </propertySet>
681                     <name>
682                         <simpleValue>FireRating</simpleValue>
683                     </name>
684                     <value>
685                         <xs:restriction base="xs:string">
686                             <xs:enumeration value="REI 30"/>
687                             <xs:enumeration value="REI 60"/>
688                             <xs:enumeration value="REI 90"/>
689                             <xs:enumeration value="REI 120"/>
690                             <xs:enumeration value="REI 180"/>
691                         </xs:restriction>
692                     </value>
693                 </property>
694             </requirements>
695         </specification>
696         <specification minOccurs="0" name="4.4 Decken innerhalb in
                oberirdischen Geschoßen" ifcVersion="IFC4">
697             <applicability>
698                 <entity>
699                     <name>
700                         <simpleValue>IFCSLAB</simpleValue>
701                     </name>
702                 </entity>
703                 <partOf relation="IFCRELSPACEBOUNDARY">
704                     <entity>
705                         <name>
706                             <simpleValue>IFCSPACE</simpleValue>
707                         </name>
708                     </entity>
709                     <partOf relation="IFCRELAGGREGATES">
710                         <entity>
711                             <name>
712                                 <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                        >
713                             </name>
714                         </entity>
715                         <partOf relation="IFCRELAGGREGATES">
```

```
716                         <entity>
717                             <name>
718                                 <simpleValue>IFCBUILDING</simpleValue>
719                             </name>
720                         </entity>
721                         <property datatype="IfcLabel">
722                             <propertySet>
723                                 <simpleValue>ZDB_Building_Infos</
                                        simpleValue>
724                             </propertySet>
725                             <name>
726                                 <simpleValue>Gebaeudeklasse</
                                        simpleValue>
727                             </name>
728                             <value>
729                                 <simpleValue>GK2</simpleValue>
730                             </value>
731                         </property>
732                     </partOf>
733                     <attribute>
734                         <name>
735                             <simpleValue>Elevation</simpleValue>
736                         </name>
737                         <value>
738                             <xs:restriction base="xs:integer">
739                                 <xs:minInclusive value="0" />
740                             </xs:restriction>
741                         </value>
742                     </attribute>
743                 </partOf>
744                 <property datatype="IfcInteger">
745                     <propertySet>
746                         <simpleValue>Pset_SpaceSpecific</simpleValue>
747                     </propertySet>
748                     <name>
749                         <simpleValue>Wohnungsnummer</simpleValue>
750                     </name>
751                 </property>
752                 <multiObjectProcessing>
753                     <simpleValue>samePropertyValuesForAllObjects</
                            simpleValue>
754                 </multiObjectProcessing>
755             </partOf>
756             <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
757                 <entity>
758                     <name>
759                         <simpleValue>IFCBUILDINGSTOREY</simpleValue>
760                     </name>
761                 </entity>
762                 <attribute>
763                     <name>
764                         <simpleValue>Elevation</simpleValue>
765                     </name>
766                     <value>
767                         <xs:restriction base="xs:integer">
768                             <xs:minInclusive value="0" />
769                         </xs:restriction>
770                     </value>
771                 </attribute>
772             </partOf>
```

```
773             </applicability>
774             <requirements>
775                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
776                     <propertySet>
777                         <simpleValue>Pset_SlabCommon</simpleValue>
778                     </propertySet>
779                     <name>
780                         <simpleValue>FireRating</simpleValue>
781                     </name>
782                     <value>
783                         <xs:restriction base="xs:string">
784                             <xs:enumeration value="REI 30"/>
785                             <xs:enumeration value="REI 60"/>
786                             <xs:enumeration value="REI 90"/>
787                             <xs:enumeration value="REI 120"/>
788                             <xs:enumeration value="REI 180"/>
789                         </xs:restriction>
790                     </value>
791                 </property>
792             </requirements>
793         </specification>
794         <specification minOccurs="0" name="4.5 Decken über unterirdischen
                Geschoßen" ifcVersion="IFC4">
795             <applicability>
796                 <entity>
797                     <name>
798                         <simpleValue>IFCSLAB</simpleValue>
799                     </name>
800                 </entity>
801                 <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
802                     <entity>
803                         <name>
804                             <simpleValue>IFCBUILDINGSTOREY</simpleValue>
805                         </name>
806                     </entity>
807                     <partOf relation="IFCRELAGGREGATES">
808                         <entity>
809                             <name>
810                                 <simpleValue>IFCBUILDING</simpleValue>
811                             </name>
812                         </entity>
813                         <property datatype="IfcLabel">
814                             <propertySet>
815                                 <simpleValue>ZDB_Building_Infos</
                                        simpleValue>
816                             </propertySet>
817                             <name>
818                                 <simpleValue>Gebaeudeklasse</simpleValue>
819                             </name>
820                             <value>
821                                 <simpleValue>GK2</simpleValue>
822                             </value>
823                         </property>
824                     </partOf>
825                     <attribute>
826                         <name>
827                             <simpleValue>Elevation</simpleValue>
828                         </name>
829                         <value>
```

```
                            <xs:restriction base="xs:integer">
                                <xs:maxExclusive value="0" />
                            </xs:restriction>
                        </value>
                    </attribute>
                </partOf>
            </applicability>
            <requirements>
                <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                    ="1">
                    <propertySet>
                        <simpleValue>Pset_SlabCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>FireRating</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="REI 60"/>
                            <xs:enumeration value="REI 90"/>
                            <xs:enumeration value="REI 120"/>
                            <xs:enumeration value="REI 180"/>
                        </xs:restriction>
                    </value>
                </property>
            </requirements>
        </specification>

    </specifications>
</ids>
```

Anhang D

IDS-Datei für die Tabelle 2a der OIB-Richtlinie 2

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- IDS (INFORMATION DELIVERY SPECIFICATION) CREATED USING IFCOPENSHELL -->
<ids xmlns="http://standards.buildingsmart.org/IDS" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://standards.buildingsmart.org/IDS http://standards.buildingsmart.org/IDS/0.9.6/ids.xsd">
    <info>
        <title>IDS for OIB 2 Tabelle 2a</title>
        <copyright>Patrick Loibl</copyright>
        <description>IDS for OIB 2 Tabelle 2a</description>
        <date>2023-05-19</date>
    </info>
    <specifications>
        <specification minOccurs="0" name="1.1 Wände von Treppenhäusern - oberirdisch" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCWALL</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELSPACEBOUNDARY">
                    <entity>
                        <name>
                            <simpleValue>IFCSPACE</simpleValue>
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDINGSTOREY</simpleValue>
                            </name>
                        </entity>
                        <partOf relation="IFCRELAGGREGATES">
                            <entity>
                                <name>
                                    <simpleValue>IFCBUILDING</simpleValue>
                                </name>
                            </entity>
                            <property datatype="IfcLabel">
                                <propertySet>
                                    <simpleValue>ZDB_Building_Infos</simpleValue>
```

```
                                </propertySet>
                                <name>
                                    <simpleValue>Gebaeudeklasse</
                                        simpleValue>
                                </name>
                                <value>
                                    <simpleValue>GK2</simpleValue>
                                </value>
                            </property>
                        </partOf>
                        <attribute>
                            <name>
                                <simpleValue>Elevation</simpleValue>
                            </name>
                            <value>
                                <xs:restriction base="xs:integer">
                                    <xs:minInclusive value="0" />
                                </xs:restriction>
                            </value>
                        </attribute>
                    </partOf>
                    <property datatype="IfcLabel" >
                        <propertySet>
                            <simpleValue>Pset_SpaceSpecific</simpleValue>
                        </propertySet>
                        <name>
                            <simpleValue>SpaceName</simpleValue>
                        </name>
                        <value>
                            <simpleValue>Treppe</simpleValue>
                        </value>
                    </property>
                </partOf>
            </applicability>
            <requirements>
                <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                    ="1">
                    <propertySet>
                        <simpleValue>Pset_WallCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>FireRating</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="REI180"/>
                            <xs:enumeration value="REI120"/>
                            <xs:enumeration value="REI90"/>
                            <xs:enumeration value="REI60"/>
                            <xs:enumeration value="REI30"/>
                            <xs:enumeration value="EI180"/>
                            <xs:enumeration value="EI120"/>
                            <xs:enumeration value="EI90"/>
                            <xs:enumeration value="EI60"/>
                            <xs:enumeration value="EI30"/>
                        </xs:restriction>
                    </value>
                </property>
            </requirements>
        </specification>
```

```
97          <specification minOccurs="0" name="1.1 Wände von Treppenhäusern -
                unterirdisch" ifcVersion="IFC4">
98              <applicability>
99                  <entity>
100                     <name>
101                         <simpleValue>IFCWALL</simpleValue>
102                     </name>
103                 </entity>
104                 <partOf relation="IFCRELSPACEBOUNDARY">
105                     <entity>
106                         <name>
107                             <simpleValue>IFCSPACE</simpleValue>
108                         </name>
109                     </entity>
110                     <partOf relation="IFCRELAGGREGATES">
111                         <entity>
112                             <name>
113                                 <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                        >
114                             </name>
115                         </entity>
116                         <partOf relation="IFCRELAGGREGATES">
117                             <entity>
118                                 <name>
119                                     <simpleValue>IFCBUILDING</simpleValue>
120                                 </name>
121                             </entity>
122                             <property datatype="IfcLabel">
123                                 <propertySet>
124                                     <simpleValue>ZDB_Building_Infos</
                                            simpleValue>
125                                 </propertySet>
126                                 <name>
127                                     <simpleValue>Gebaeudeklasse</
                                            simpleValue>
128                                 </name>
129                                 <value>
130                                     <simpleValue>GK2</simpleValue>
131                                 </value>
132                             </property>
133                         </partOf>
134                         <attribute>
135                             <name>
136                                 <simpleValue>Elevation</simpleValue>
137                             </name>
138                             <value>
139                                 <xs:restriction base="xs:integer">
140                                     <xs:maxExclusive value="0" />
141                                 </xs:restriction>
142                             </value>
143                         </attribute>
144                     </partOf>
145                     <property datatype="IfcLabel">
146                         <propertySet>
147                             <simpleValue>Pset_SpaceSpecific</simpleValue>
148                         </propertySet>
149                         <name>
150                             <simpleValue>SpaceName</simpleValue>
151                         </name>
152                         <value>
```

```
153                         <simpleValue>Treppe</simpleValue>
154                     </value>
155                 </property>
156             </partOf>
157         </applicability>
158         <requirements>
159             <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                    ="1">
160                 <propertySet>
161                     <simpleValue>Pset_WallCommon</simpleValue>
162                 </propertySet>
163                 <name>
164                     <simpleValue>FireRating</simpleValue>
165                 </name>
166                 <value>
167                     <xs:restriction base="xs:string">
168                         <xs:enumeration value="REI180"/>
169                         <xs:enumeration value="REI120"/>
170                         <xs:enumeration value="REI90"/>
171                         <xs:enumeration value="REI60"/>
172                         <xs:enumeration value="EI180"/>
173                         <xs:enumeration value="EI120"/>
174                         <xs:enumeration value="EI90"/>
175                         <xs:enumeration value="EI60"/>
176                     </xs:restriction>
177                 </value>
178             </property>
179         </requirements>
180     </specification>
181     <!-- 2. Decek über dem Treppenhaus: Folgende Anforderung kann nicht
             umgesetzt werden, da nicht die oberste Decke aus dem
            Treppenhaus bestimmt werden kann.-->
182     <specification minOccurs="0" name="3.1 Türen in Wänden von Treppenh
        äusern - zu Wohnungen, Betriebseinheiten sowie sonstigen Räumen
         (oberirdisch)" ifcVersion="IFC4">
183         <applicability>
184             <entity>
185                 <name>
186                     <simpleValue>IFCDOOR</simpleValue>
187                 </name>
188             </entity>
189             <partOf relation="IFCRELSPACEBOUNDARY">
190                 <entity>
191                     <name>
192                         <simpleValue>IFCSPACE</simpleValue>
193                     </name>
194                 </entity>
195                 <partOf relation="IFCRELAGGREGATES">
196                     <entity>
197                         <name>
198                             <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                >
199                         </name>
200                     </entity>
201                     <partOf relation="IFCRELAGGREGATES">
202                         <entity>
203                             <name>
204                                 <simpleValue>IFCBUILDING</simpleValue>
205                             </name>
206                         </entity>
```

```
207                         <property datatype="IfcLabel">
208                             <propertySet>
209                                 <simpleValue>ZDB_Building_Infos</
                                        simpleValue>
210                             </propertySet>
211                             <name>
212                                 <simpleValue>Gebaeudeklasse</
                                        simpleValue>
213                             </name>
214                             <value>
215                                 <simpleValue>GK2</simpleValue>
216                             </value>
217                         </property>
218                     </partOf>
219                     <attribute>
220                         <name>
221                             <simpleValue>Elevation</simpleValue>
222                         </name>
223                         <value>
224                             <xs:restriction base="xs:integer">
225                                 <xs:minInclusive value="0" />
226                             </xs:restriction>
227                         </value>
228                     </attribute>
229                 </partOf>
230                 <property datatype="IfcLabel">
231                     <propertySet>
232                         <simpleValue>Pset_SpaceSpecific</simpleValue>
233                     </propertySet>
234                     <name>
235                         <simpleValue>SpaceName</simpleValue>
236                     </name>
237                     <value>
238                         <simpleValue>Treppe</simpleValue>
239                     </value>
240                 </property>
241                 <property datatype="IfcLabel">
242                     <propertySet>
243                         <simpleValue>Pset_SpaceSpecific</simpleValue>
244                     </propertySet>
245                     <name>
246                         <simpleValue>SpaceName</simpleValue>
247                     </name>
248                     <value>
249                         <xs:restriction base="xs:string">
250                             <xs:enumeration value="Wohnen"/>
251                             <xs:enumeration value="Küche"/>
252                             <xs:enumeration value="Wohnküche"/>
253                             <xs:enumeration value="Einlagerungsraum"/>
254                             <xs:enumeration value="Kinderwagen-
                                    Abstellraum"/>
255                             <xs:enumeration value="Müllraum"/>
256                             <xs:enumeration value="Abstellraum"/>
257                             <xs:enumeration value="Toilette"/>
258                         </xs:restriction>
259                     </value>
260                 </property>
261                 <multiObjectProcessing>
262                     <simpleValue>differentObjectsForEachPropertyValue</
                        simpleValue>
```

```
                    </multiObjectProcessing>
                </partOf>
            </applicability>
            <requirements>
                <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                    ="1">
                    <propertySet>
                        <simpleValue>Pset_DoorCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>FireRating</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="EI2 30"/>
                            <xs:enumeration value="EI2 60"/>
                            <xs:enumeration value="EI2 90"/>
                            <xs:enumeration value="EI2 30-C"/>
                            <xs:enumeration value="EI2 60-C"/>
                            <xs:enumeration value="EI2 90-C"/>
                            <xs:enumeration value="EI2 30-C-Sm"/>
                            <xs:enumeration value="EI2 60-C-Sm"/>
                            <xs:enumeration value="EI2 90-C-Sm"/>
                        </xs:restriction>
                    </value>
                </property>
            </requirements>
        </specification>
        <specification minOccurs="0" name="3.2 Türen in Wänden von Treppenh
            äusern - zu Gängen oberirdisch" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCDOOR</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELSPACEBOUNDARY">
                    <entity>
                        <name>
                            <simpleValue>IFCSPACE</simpleValue>
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                    >
                            </name>
                        </entity>
                        <partOf relation="IFCRELAGGREGATES">
                            <entity>
                                <name>
                                    <simpleValue>IFCBUILDING</simpleValue>
                                </name>
                            </entity>
                            <property datatype="IfcLabel">
                                <propertySet>
                                    <simpleValue>ZDB_Building_Infos</
                                        simpleValue>
                                </propertySet>
```

```
                                <name>
                                    <simpleValue>Gebaeudeklasse</
                                        simpleValue>
                                </name>
                                <value>
                                    <simpleValue>GK2</simpleValue>
                                </value>
                            </property>
                        </partOf>
                        <attribute>
                            <name>
                                <simpleValue>Elevation</simpleValue>
                            </name>
                            <value>
                                <xs:restriction base="xs:integer">
                                    <xs:minInclusive value="0" />
                                </xs:restriction>
                            </value>
                        </attribute>
                    </partOf>
                    <property datatype="IfcLabel">
                        <propertySet>
                            <simpleValue>Pset_SpaceSpecific</simpleValue>
                        </propertySet>
                        <name>
                            <simpleValue>SpaceName</simpleValue>
                        </name>
                        <value>
                            <simpleValue>Treppe</simpleValue>
                        </value>
                    </property>
                    <property datatype="IfcLabel">
                        <propertySet>
                            <simpleValue>Pset_SpaceSpecific</simpleValue>
                        </propertySet>
                        <name>
                            <simpleValue>SpaceName</simpleValue>
                        </name>
                        <value>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="Hauptgang"/>
                                <xs:enumeration value="Nebengang"/>
                            </xs:restriction>
                        </value>
                    </property>
                    <multiObjectProcessing>
                        <simpleValue>differentObjectsForEachPropertyValue</
                            simpleValue>
                    </multiObjectProcessing>
                </partOf>
            </applicability>
        </specification>
        <specification minOccurs="0" name="3.3 Türen in Wänden von Treppenh
            äusern - zu Gängen und Räumen unterirdisch" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCDOOR</simpleValue>
                    </name>
                </entity>
```

```
376                 <partOf relation="IFCRELSPACEBOUNDARY">
377                     <entity>
378                         <name>
379                             <simpleValue>IFCSPACE</simpleValue>
380                         </name>
381                     </entity>
382                     <partOf relation="IFCRELAGGREGATES">
383                         <entity>
384                             <name>
385                                 <simpleValue>IFCBUILDINGSTOREY</simpleValue
                                        >
386                             </name>
387                         </entity>
388                         <partOf relation="IFCRELAGGREGATES">
389                             <entity>
390                                 <name>
391                                     <simpleValue>IFCBUILDING</simpleValue>
392                                 </name>
393                             </entity>
394                             <property datatype="IfcLabel">
395                                 <propertySet>
396                                     <simpleValue>ZDB_Building_Infos</
                                            simpleValue>
397                                 </propertySet>
398                                 <name>
399                                     <simpleValue>Gebaeudeklasse</
                                            simpleValue>
400                                 </name>
401                                 <value>
402                                     <simpleValue>GK2</simpleValue>
403                                 </value>
404                             </property>
405                         </partOf>
406                         <attribute>
407                             <name>
408                                 <simpleValue>Elevation</simpleValue>
409                             </name>
410                             <value>
411                                 <xs:restriction base="xs:integer">
412                                     <xs:maxExclusive value="0" />
413                                 </xs:restriction>
414                             </value>
415                         </attribute>
416                     </partOf>
417                     <property datatype="IfcLabel">
418                         <propertySet>
419                             <simpleValue>Pset_SpaceSpecific</simpleValue>
420                         </propertySet>
421                         <name>
422                             <simpleValue>SpaceName</simpleValue>
423                         </name>
424                         <value>
425                             <simpleValue>Treppe</simpleValue>
426                         </value>
427                     </property>
428                     <multiObjectProcessing>
429                         <simpleValue>differentPropertyValuesForAnyObject</
                                simpleValue>
430                     </multiObjectProcessing>
431                 </partOf>
```

```
            </applicability>
            <requirements>
                <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                    ="1">
                    <propertySet>
                        <simpleValue>Pset_DoorCommon</simpleValue>
                    </propertySet>
                    <name>
                        <simpleValue>FireRating</simpleValue>
                    </name>
                    <value>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="EI2 30"/>
                            <xs:enumeration value="EI2 60"/>
                            <xs:enumeration value="EI2 90"/>
                            <xs:enumeration value="EI2 30-C"/>
                            <xs:enumeration value="EI2 60-C"/>
                            <xs:enumeration value="EI2 90-C"/>
                            <xs:enumeration value="EI2 30-C-Sm"/>
                            <xs:enumeration value="EI2 60-C-Sm"/>
                            <xs:enumeration value="EI2 90-C-Sm"/>
                        </xs:restriction>
                    </value>
                </property>
            </requirements>
        </specification>
        <specification minOccurs="0" name="4. Treppenläufe und Podeste in
            Treppenhäusern" ifcVersion="IFC4">
            <applicability>
                <entity>
                    <name>
                        <simpleValue>IFCSTAIR</simpleValue>
                    </name>
                </entity>
                <partOf relation="IFCRELCONTAINEDINSPATIALSTRUCTURE">
                    <entity>
                        <name>
                            <simpleValue>IFCBUILDINGSTOREY</simpleValue>
                        </name>
                    </entity>
                    <partOf relation="IFCRELAGGREGATES">
                        <entity>
                            <name>
                                <simpleValue>IFCBUILDING</simpleValue>
                            </name>
                        </entity>
                        <property datatype="IfcLabel">
                            <propertySet>
                                <simpleValue>ZDB_Building_Infos</
                                    simpleValue>
                            </propertySet>
                            <name>
                                <simpleValue>Gebaeudeklasse</simpleValue>
                            </name>
                            <value>
                                <simpleValue>GK2</simpleValue>
                            </value>
                        </property>
                    </partOf>
                </partOf>
```

```
489                 <property datatype="IfcLabel">
490                     <propertySet>
491                         <simpleValue>Pset_StairSpecific</simpleValue>
492                     </propertySet>
493                     <name>
494                         <simpleValue>TypeOfStair</simpleValue>
495                     </name>
496                     <value>
497                         <xs:restriction base="xs:string">
498                             <xs:enumeration value="Haupttreppe"/>
499                             <xs:enumeration value="Nebentreppe"/>
500                         </xs:restriction>
501                     </value>
502                 </property>
503             </applicability>
504             <requirements>
505                 <property datatype ="IfcLabel" minOccurs ="1" maxOccurs
                        ="1">
506                     <propertySet>
507                         <simpleValue>Pset_StairCommon</simpleValue>
508                     </propertySet>
509                     <name>
510                         <simpleValue>FireRating</simpleValue>
511                     </name>
512                     <value>
513                         <xs:restriction base="xs:string">
514                             <xs:enumeration value="R180"/>
515                             <xs:enumeration value="R120"/>
516                             <xs:enumeration value="R90"/>
517                             <xs:enumeration value="R60"/>
518                             <xs:enumeration value="R30"/>
519                         </xs:restriction>
520                     </value>
521                 </property>
522             </requirements>
523         </specification>
524     </specifications>
525 </ids>
```