

Christoph Carl Eichler | Christian Schranz | Tina Krischmann | Harald Urban

# BIMcert Handbuch

Grundlagenwissen openBIM

Ausgabe 2023





### 3.7 IDS – Information Delivery Specification

Gastautoren: Léon van Berlo, Simon Fischer



IDS ist ein Standard von buildingSMART International zur Definition von computer-interpretierbaren Modellaustauschanforderungen. IDS ist ein verhältnismäßig junger Standard (2023), der als Ergänzung zu MVD verstanden werden kann. Während sich MVD mit grundlegenden Themen wie der korrekten Abbildung der Klassen-Hierarchie und der Übertragung der Geometrie beschäftigt, spezifiziert IDS den alphanumerischen Informationsgehalt von Modellen. Es definiert, mit welchen Informationen Objekte übertragen werden müssen. Aus diesem Grund ist IDS ein vielversprechendes Werkzeug für die Bereitstellung und Prüfbarkeit von Informationsanforderungen der Auftraggeber, definiert in den AIA. Es bindet die derzeit in Textform vorhandenen Informationsanforderungen in den automatisierten openBIM-Prozess ein. IDS kann für zwei Teilprozesse zur Anwendung kommen:

- Informationen definieren: Als Konfigurationsdatei für BIM-Autorensoftware, zur automatisierten Bereitstellung der geforderten Informationsstruktur und
- Informationen prüfen: Als Konfigurationsdatei für BIM-Prüfsoftware, zur automatisierten Prüfung des Aufbaus und Inhalts der Informationsstruktur.

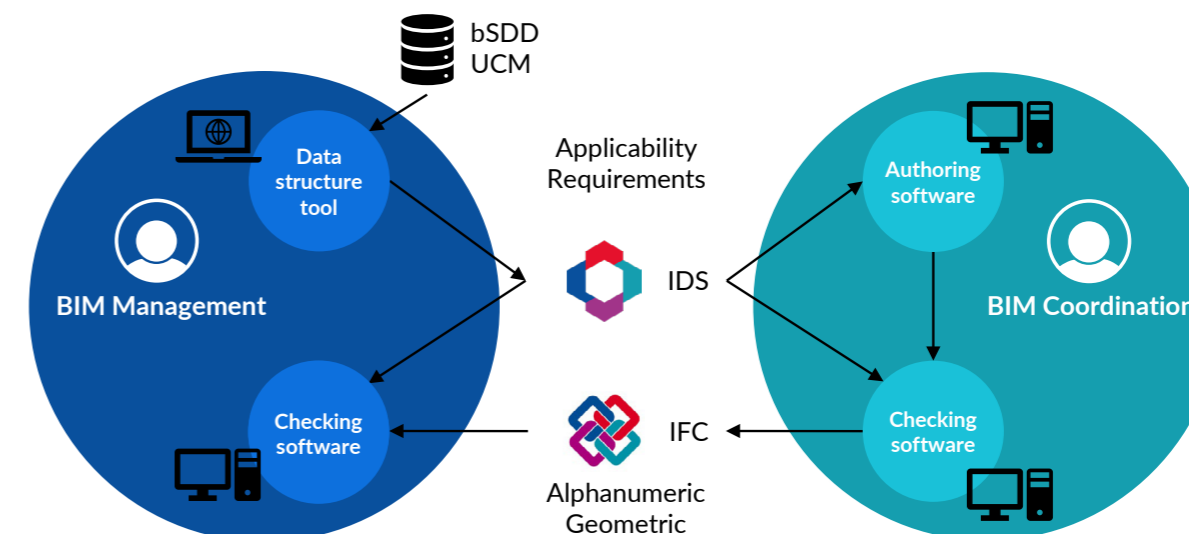
Der IDS-Workflow beginnt beim Auftraggeber (BIM-Management). Das BIM-Management definiert die gewünschten BIM-Anwendungsfälle und die dafür erforderlichen Informationen. Schauen wir uns zwei Beispiele für Informationsanforderungen an.

Erstens: Ein Kunde möchte, dass alle Räume in einem Modell mit einem bestimmten Code klassifiziert werden und eine Reihe von Eigenschaften haben. Die Anforderung könnte wie folgt beschrieben werden: *»Alle Raumdaten in einem Modell müssen als [AT]Zimmer klassifiziert sein und NetFloorArea und GrossFloorArea (beide im Set BaseQuantities) und ein Property namens AT\_Zimmernummer im Property Set Austria\_example haben.«* Dies ist nur ein Beispiel. Es kann sich um jede Art von Anforderung handeln. Benutzer können die Anforderungen auch weiter verfeinern, so dass sie nicht für alle Räume gelten, sondern nur für Räume mit bestimmten Eigenschaften – zum Beispiel für Räume mit einer bestimmten Eigenschaft und/oder einem bestimmten Eigenschaftswert oder für Räume, die Teil einer bestimmten Hierarchie sind, oder für Räume, die auf eine bestimmte Weise klassifiziert sind. Dies gilt für alle Objekte, nicht nur für Räume. Die Anforderungen von Klassifizierungscodes, Materialien, Sets, Attributen, Properties und einigen Beziehungen können auch für die Auswahl (manchmal auch als Filterung bezeichnet; im IDS formal als Anwendungsbereich bezeichnet) von Objekten angegeben werden.

Der Anwendungsbereich ist im zweiten Beispiel enthalten, einer Spezifikation bestimmter Eigenschaften für Wände: *»Alle Wände müssen die Properties LoadBearing und FireRating haben (beide in einem Property Set namens Pset\_WallCommon). Wände mit einem Wert von true für die Property LoadBearing benötigen*

einen Wert für die Property FireRating aus der folgenden Liste (ND, REI 30, REI 60, REI 90, REI 120).« Das Raumbeispiel wird später verwendet, um verschiedene Möglichkeiten der Visualisierung von IDS zu zeigen. Das Wandbeispiel ist in der Beschreibung der Datenstruktur von IDS im nächsten Abschnitt enthalten.

Die Definition der Informationsanforderungen erfolgt üblicherweise mithilfe eines Datenstrukturwerkzeugs und unter Berücksichtigung von Daten aus dem bSDD und dem UCM. Anschließend exportiert das BIM-Management die Informationsanforderungen im IDS-Format und stellt sie der BIM-Koordination bzw. BIM-Erstellung des Auftragnehmers zur Verfügung. Diese nutzt IDS als Konfigurationsdatei sowohl für die BIM-Autorensoftware als auch für die BIM-Prüfsoftware. Die Autorensoftware kann dadurch die geforderten Merkmale automatisch objektspezifisch anlegen. In der BIM-Prüfsoftware bewirkt die Konfigurationsdatei eine automatische Auswahl und Befüllung von Prüfregeln. Die geprüfte IFC-Datei wird schließlich dem BIM-Management übermittelt, das ebenfalls die eigens erstellte IDS-Datei zur Konfiguration ihrer Prüfsoftware verwendet. IDS koppelt damit die Informationsanforderungen des Auftraggebers mit dem BIM-Modell und ermöglicht damit eine automatisierte Prüfung genau jener Informationen, die definiert wurden.



#### 3.7.1 Datenstruktur

Das Dateiformat IDS basiert auf dem XML-Schema. Konkret ist es eine standardisierte Form davon. Das bedeutet, dass der Aufbau und die Syntax einer IDS-Datei genauer spezifiziert ist als für eine allgemeine XML-Datei. Dazu nutzt buildingSMART International das Format XSD (XML Schema Definition). Darin ist definiert, welche Elemente in einer IDS-Datei enthalten sein müssen und dürfen.

Grundsätzlich ist eine IDS-Datei in zwei Bereiche gegliedert: »Header« und Liste an Spezifikationen. Der Header enthält allgemeine Metadaten zur Datei. Diese sind innerhalb des Elements info gesammelt. Mögliche Informationen darin sind title, copyright, version, description, author, date, purpose und milestone. Verpflichtend vorgeschrieben ist davon nur der Titel. Alle anderen Informationen sind optional. Die Zeilen vor den Metadaten sind einerseits der XML-Prolog zur

3.7 IDS – Information Delivery Specification

Definition der XML-Version und der Codierung sowie das *Root element* (<ids ...>) mit der Definition von Namensräumen für das Dokument.

```
<?xml version="1.0" encoding="UTF-8"?>
<ids xmlns="http://standards.buildingsmart.org/IDS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://standards.buildingsmart.org/IDS/ids_09.xsd">
  <info>
    <title>IDS for BIMcert</title>
    <copyright>Simon Fischer</copyright>
    <description>Created to describe IDS for BIMcert</description>
    <date>2023-01-11</date>
  </info>
```

Nach den allgemeinen Metadaten folgt der eigentliche Inhalt einer IDS-Datei: eine Liste an Spezifikationen. Spezifikationen beschreiben Informationsanforderungen an Elemente in IFC. Sie sind so aufgebaut, dass sie einerseits von Menschen einfach verstanden werden können und andererseits auch maschinenlesbar sind. Eine Spezifikation besteht aus drei Teilen: Metadaten, Anwendungsbereich (*Applicability*) und Anforderungen (*Requirements*).

Die **Metadaten** sind als XML-Attribute im *Specification element* enthalten. Im nachfolgenden Beispiel sind das die beiden verpflichtenden Informationen *name* und *ifcVersion*. Darüber hinaus können zusätzlich die Notwendigkeit (*occurs*), eine ID (*identifier*), eine Beschreibung (*description*) und Anweisungen (*instructions*) definiert werden. Die *description* and *instructions* sind Optionen, um die Anforderungen um eine für den Menschen lesbare Dokumentation zu ergänzen. IDS ist zwar für die Interpretation durch Computer ausgelegt, aber in vielen Fällen werden Menschen unweigerlich Informationen zum BIM-Datensatz hinzufügen müssen. Der Ersteller einer IDS kann daher Anweisungen hinterlassen, die klarstellen, dass auch ein Mensch Daten eingeben muss. Als zweiter Bestandteil der Spezifikation folgt der **Anwendungsbereich** (*Applicability*). Dieser Filter definiert, für welche Elemente die aktuelle Spezifikation relevant bzw. anzuwenden ist. Diese Einschränkung kann auf der Ebene von IFC-Klassen, aber auch deutlich spezifischer über *Predefined Types*, *Properties*, *Materialien* usw. erfolgen. Der dritte Bestandteil der Spezifikation sind die **Anforderungen** (*Requirements*). Diese enthalten die eigentlichen Informationsanforderungen an Objekte. Die Kombination von Anwendungsbereich und Anforderungen bildet die maschinenlesbare Definition von Informationsanforderungen. Beide Bestandteile verwenden zur Spezifizierung ihres Inhalts sogenannte *Facets*. *Facets* bedeuten im Zusammenhang mit XML Einschränkungen für XML-Elemente. Im IDS-Schema beschreiben *Facets* Informationen, die ein Element im IFC-Modell haben kann. Es werden dabei 6 exakt definierte Facet-Parameter verwendet, um die Anforderungen maschinenlesbar zu machen. Die Facet-Parameter beziehen sich auf verschiedene Inhalte im IFC-Schema:

- Entity Facet
- Attribute Facet
- Classification Facet
- Property Facet
- Material Facet
- PartOf Facet

3.7 IDS – Information Delivery Specification

Bei Verwendung der *Facets* zur Definition des Anwendungsbereichs können Elemente sehr gezielt gefiltert werden (z.B. nur Elemente, die ein bestimmtes *Property* mit einem bestimmten Wert besitzen). Dabei ist es auch möglich, mehrere *Facets* zu kombinieren, was die Möglichkeiten zur individuellen Definition von Anforderungen erhöht.

Durch all diese Funktionalitäten kann IDS fortgeschrittene Definitionen von Anforderungen bieten. Es ermöglicht Benutzern, Eigenschaften zu verlangen, die mit einer bestimmten Art von Maßnahme gemeinsam genutzt werden. Es gibt auch umfangreiche Möglichkeiten, Einschränkungen für Werte zu definieren. So kann beispielsweise der Wert einer Eigenschaft nur aus einer Liste zulässiger Werte ausgewählt werden. Handelt es sich bei dem Wert um eine Zahl, so kann er ein bestimmtes Minimum, Maximum oder einen Bereich haben. Auch der Mustervergleich ist eine in IDS verfügbare Option. IDS verwendet hierfür die XSD-Einschränkungen, um die Zuverlässigkeit der Implementierung zu verbessern. Einschränkungen für Spezifikationen sind ein weiteres Beispiel für eine erweiterte Funktion. Mit den XML-Attributen *minOccurs* und *maxOccurs* können Benutzer ein Minimum, Maximum, einen Bereich oder eine genaue Anzahl von Objekten definieren, die im BIM-Datensatz enthalten sein müssen. Benutzer können mit dem *PartOf Facet* bestimmte Strukturen im BIM-Datensatz vorgeben, die typisch für die Verwendung von Industry Foundation Classes (IFC) sind. Diese Funktionalität ermöglicht die Definition der Anforderungen, dass ein Objekt Teil einer Baugruppe oder Teil einer Gruppe sein soll.

Das nachfolgende Beispiel zeigt Vorgaben für Objekte der Klasse *IfcWall* als IDS und im Vergleich dazu als klassischen Text in einer PDF-Datei. Die erste Spezifikation gibt vor, dass jede Wand ein *Property LoadBearing* im *Pset\_WallCommon* benötigt. Die zweite Spezifikation regelt mögliche Werte für die Feuerwiderstandsklasse von tragenden Wänden (die Liste ist als Ausschnitt möglicher Werte zu verstehen). Der Anwendungsbereich beider Spezifikationen ist hellblau hervorgehoben, die Anforderungen sind hellorange markiert.

LOI – Level of Information (IfcWall)

Property	Data type	Unit of value	Location	Selection set	Note
LoadBearing	IfcBoolean	Logical value	Pset_WallCommon	-	Default value: FALSE
FireRating	IfcLabel	Text	Pset_WallCommon	Selection set	Default value: ND; Example: REI 60
...					

Selection sets IfcWall FireRating

load bearing	non-bearing	...
ND	ND	
REI 30	EI 30	
REI 60	EI 60	
REI 90	EI 90	
REI120	EI120	
...	...	

```

<specifications>
  <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>
  <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>
      </property>
    </requirements>
  </specification>
</ids>

```

```

    <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>
</specifications>
</ids>

```

### 3.7.2 Bezug zum buildingSmart Data Dictionary

Erhält ein Benutzer eine IDS von einem Kunden, kann er seine eigenen Daten mit den in der IDS definierten Anforderungen abgleichen. Die IDS kann für den Empfänger lesbare Erklärungen und Anweisungen enthalten, damit er die Anforderungen besser versteht. IDS ermöglicht das Hinzufügen eines Links (formal als *Uniform Resource Identifier* URI bezeichnet) mit weiteren Informationen über eine Eigenschaft oder einen Klassifizierungscode. Hier kommt der Bezug zum bSDD ins Spiel. Ein URI, der mit *identifier.buildingsmart.org* beginnt, verweist auf ein Objekt, das im bSDD zu finden ist. Wenn der Benutzer diesem URI folgt, erhält er mehr Informationen über ein Property, die über den Detailgrad hinausgehen, der in der IFC angegeben werden kann. Die bSDD enthält detaillierte, standardisierte Informationen über Definitionen, Einheiten, Beziehungen zu anderen Objekten usw. Dies gilt für Klassifizierungscodes, Eigenschaften (einschließlich Attribute und Mengen) und Materialien, sowohl für internationale als auch für landesspezifische Normen. Die Optionen zur Definition von Werteschränkungen in IDS sind die gleichen, die auch bSDD unterstützt. Dies ermöglicht eine nahtlose Interaktion zwischen IDS und bSDD. Durch Hinzufügen des URI zu einer Eigenschaft oder einem Klassifizierungscode (oder einem System) können Benutzer (und in einigen Fällen sogar Computer) mehr Informationen über die Anforderung und die typische Verwendung von Objekten erhalten.

### 3.7.3 Facet-Parameter

Dieser Abschnitt behandelt die Funktionalität und Möglichkeiten der sechs Facet-Parameter. Für *Facets* kann wie für die Spezifikationen die Notwendigkeit (occurs) als XML-Attribut angegeben werden. Das *Property Facet* und das *PartOf Facet* bieten darüber hinaus weitere spezifische XML-Attribute. Die folgende Beschreibung enthält einen Beispiel Code für jedes Facet. Die beiden ersten Ausschnitte sind jeweils in den Anwendungsbereich einer Spezifikation eingebunden. Die weiteren Ausschnitte können auf gleiche Weise in den Anwendungsbereich oder den Anforderungsbereich einer Spezifikation eingebunden werden.



**Entity Facet**

Das *Entity Facet* bezieht sich auf die Klassen im IFC-Schema. Es ist daher besonders wichtig zur Definition des Anwendungsbereichs, da es beschreibt, für welche IFC-Klasse eine Spezifizierung relevant ist. Neben dem verpflichtenden Namen der IFC-Klasse kann im *Entity Facet* auch optional ein *Predefined Type* eines Elements festgelegt werden. Folgender Code-Ausschnitt zeigt die Verwendung des *Entity Facet* zur Festlegung des Anwendungsbereichs einer Spezifikation auf alle Elemente der IFC-Klasse *IfcDoor*.

```
<applicability>
  <entity>
    <name>
      <simpleValue>IFCDOOR</simpleValue>
    </name>
  </entity>
</applicability>
```

**Attribute Facet**

Das *Attribute Facet* ermöglicht die Berücksichtigung von Attributen, die standardmäßig in IFC-Klassen enthalten sind, bspw. der Name eines Elements oder die GUID. Zur Verwendung des *Facet* muss der Name des Attributs angegeben werden, der Wert des Attributs ist optional. Wird nur ein Name ohne einen Wert definiert, muss das Element ein Attribut mit Namen und beliebigen, definierten (nicht leeren) Wert enthalten. Der Code-Ausschnitt zeigt die Verwendung des *Attribute Facet* zur Festlegung des Anwendungsbereichs einer Spezifikation auf alle Elemente der IFC-Klasse *IfcDoor* mit dem Namen *Entry*.

```
<applicability>
  <entity>
    <name>
      <simpleValue>IFCDOOR</simpleValue>
    </name>
  </entity>
  <attribute minOccurs="1" maxOccurs="1">
    <name>
      <simpleValue>Name</simpleValue>
    </name>
    <value>
      <simpleValue>Entry</simpleValue>
    </value>
  </attribute>
</applicability>
```

**Classification Facet**

Werden neben den Klassen des IFC-Schemas weitere Klassifikationssysteme verwendet, können diese mit dem *Classification Facet* berücksichtigt werden. Solche externen Klassifikationssysteme sind bspw. *Uniclass2015* oder nationale Systeme. Das *Classification Facet* ermöglicht die Angabe eines Klassifizierungssystems und eines Referenzcodes (wie ist ein Objekt innerhalb des Systems

klassifiziert). Beide Parameter sind optional. Falls kein Parameter angegeben ist, muss ein Objekt mit beliebigem Referenzcode in einem beliebigen System klassifiziert sein. Darüber hinaus kann ein URI als XML-Attribut des *Classification element* hinzugefügt werden, um auf weitere Informationen zu verweisen. Hier ist die Vorgabe des Systems *Uniclass2015* mit beliebigem Referenzcode dargestellt.

```
<classification minOccurs="1" maxOccurs="1">
  <system>
    <simpleValue>Uniclass2015</simpleValue>
  </system>
</classification>
```

**Property Facet**

Das *Property Facet* ist das Gegenstück zum *Attribute Facet* und bezieht sich auf die nicht standardmäßig in IFC enthaltenen Eigenschaften, die *Properties*. Darüber hinaus kann es auch zur Vorgabe von *Quantities* dienen. Zur Definition einer Anforderung kommen die Parameter *Property Set (Quantity Set)*, *Property Name (Quantity Name)*, Wert und Datentyp zum Einsatz. Der Wert des *Property* ist auch hier ein optionaler Parameter und verhält sich wie bei den vorigen *Facets*. Alle anderen Parameter sind verpflichtend, wobei der Datentyp als XML-Attribut des *Property element* anzugeben ist, nicht wie die anderen Parameter als eigenes XML-Element. Ein URI kann ebenfalls als XML-Attribut hinzugefügt werden, um z.B. auf das *bSDD* zu verweisen. Als Beispiel ist hier eine Spezifikation angeführt, die ein *Property LoadBearing* mit dem Wert *true* und dem Datentyp *IfcBoolean* im *Property Set Pset\_WallCommon* verlangt.

```
<property measure="IfcBoolean" minOccurs="1" maxOccurs="1">
  <propertySet>
    <simpleValue>Pset_WallCommon</simpleValue>
  </propertySet>
  <name>
    <simpleValue>LoadBearing</simpleValue>
  </name>
  <value>
    <simpleValue>true</simpleValue>
  </value>
</property>
```

**Material Facet**

Bei Verwendung von Einschränkungen bezüglich Materialien ist zu beachten, dass ein Objekt aus einem oder mehreren Materialien bestehen kann. Mit dem *Material Facet* wird geprüft, ob eines der Materialien des entsprechenden Objekts mit dem vorgegebenen Material übereinstimmt. Bei diesem *Facet* gibt es nur einen optionalen Parameter für das Material. Falls nicht definiert, muss eine beliebige Materialangabe vorhanden sein. Ein URI kann als XML-Attribut des *Material element* verwendet werden, um auf zusätzliche Informationen über das Material zu verweisen.

```
<material minOccurs="1" maxOccurs="1">
  <value>
    <simpleValue> ExampleMaterial</simpleValue>
  </value>
</material>
```

#### PartOf Facet

Mit dem *PartOf Facet* können Beziehungen zwischen Objekten vorgegeben werden. Beziehungen (*Relations*) werden in IFC über eigene Klassen beginnend mit *IfcRel...* definiert. Im *PartOf Facet* kann eine *Relation* über eine solche Relation-Klasse und die IFC-Klasse, auf welche die *Relation* verweisen soll, angegeben werden. Die *Relation* ist dabei als XML-Attribut des *PartOf Element* anzugeben. Eine mögliche Anforderung ist, dass ein Element einem Geschoss zugeordnet sein muss. Dafür ist die Beziehung *IfcRelContainedInSpatialStructure* mit der Klasse *IfcBuildingStorey* zu wählen.

```
<partOf relation="IfcRelContainedInSpatialStructure" minOccurs="1" maxOccurs="1">
  <entity>
    <name>
      <simpleValue>IFCBUILDINGSTOREY</simpleValue>
    </name>
  </entity>
</partOf>
```

#### 3.7.4 Einfache und komplexe Einschränkungen

Neben der Möglichkeit über die *Facets* Anforderungen für verschiedene Inhalte des IFC-Schemas festzulegen, können auch die Anforderungen selbst unterschiedlich definiert werden. Dazu unterscheidet IDS zuerst zwischen einfachen und komplexen Einschränkungen. Einfache Einschränkungen sind einzelne Werte in Form eines Texts, einer Zahl oder eines Wahrheitswerts (wahr/falsch). Komplexe Einschränkungen ermöglichen hingegen die Vorgabe mehrerer zulässiger Werte und können in vier Unterkategorien eingeteilt werden:

##### Aufzählung (Enumeration)

Die Aufzählung dient zur Angabe einer Liste zulässiger Werte. Die Liste kann sowohl Texte als auch Zahlenwerte enthalten. Nachfolgend ist ein Beispiel für die Angabe von Feuerwiderstandsklassen für tragende Wände gegeben (Ausschnitt aus möglichen Werten).

```
<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>
```

#### Muster (Pattern)

Ein Muster beschreibt in welcher Reihenfolge verschiedene Zeichen aneinandergereiht werden dürfen. Diese Funktionalität ist vor allem für Namenskonventionen bzw. Namensschemata anwendbar. Eine weit verbreitete und auch für IDS verwendete Methode zur Definition von solchen Mustern sind *Regular Expressions (Regex)*. Als Beispiel ist eine Konvention für Raumnamen angegeben. `[A-Z]` bedeutet der Name beginnt mit einem Großbuchstaben. `[0-9]{2}` legt fest, dass darauf zwei Ziffern zwischen 0 und 9 folgen. Durch `-[0-9]{2}` ist der Name nach einem Bindestrich mit zwei Ziffern zwischen 0 und 9 abzuschließen. Gültige Namen sind demnach beispielsweise `W01-01` oder `B18-74`.

```
<value>
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Z][0-9]{2}-[0-9]{2}"/>
  </xs:restriction>
</value>
```

#### Grenzen (Bounds)

Grenzen legen ein Intervall gültiger Werte fest. Dabei ist es möglich, entweder eine untere, eine obere oder beide Grenzen festzulegen. Die Grenzen können weiters durch die Symbole `</>` exklusiv oder `<=/>` inklusiv definiert werden.

#### Länge (Length)

Abschließend ist es möglich, die Länge eines Werts festzulegen, also die Anzahl der einzelnen Zeichen. Es können eine exakte sowie eine minimale oder maximale Länge eines Werts vorgegeben werden.

#### 3.7.5 Umfang und Einsatz von IDS

Eine IDS-Datei kann mehrere Anforderungen enthalten. Diese Anforderungen sind unabhängige Blöcke und haben keinen Bezug zu anderen Anforderungen in der Datei. Diese Struktur schafft die Möglichkeit, Anforderungen zwischen Dateien zu kopieren und einzufügen. Derzeit (2023) entwickeln Softwarehersteller erste IDS-Editoren und IDS-Autorentools, um den Benutzern die Erstellung von IDS-Dateien zu erleichtern. Für die Zukunft sieht *buildingSMART* das Vorhandensein von IDS-Bibliotheken vor, in denen Beispiele für einzelne Anforderungen für alle zur Verfügung stehen. Benutzer können IDS-Anforderungen suchen und sie in einen Auswahlkorb ziehen, um ihre eigene IDS-Datei zu erstellen. Eine wichtige Definition des Anwendungsbereichs von IDS ist, dass sie sich nur auf »Spezifikationen für die Informationsbereitstellung« konzentriert. Das bedeutet, dass die strukturierten IDS-Anforderungen definieren können, welche Informationen benötigt werden und wie sie strukturiert sein sollten.

Für automatisierte Arbeitsabläufe und Skripte ist es wichtig, Informationen so zu erhalten, dass sie automatisch verarbeitet werden können, und dies ist das Ziel von IDS. IDS kann jedoch nicht verwendet werden, um Designanforderungen oder sogenannte »*Rules*« zu definieren. So ist die Anforderung, dass alle Fenster in einem Toilettenraum undurchsichtiges Glas haben müssen, im Rahmen von IDS nicht möglich. Eine gültige IDS-Definition ist jedoch die Anforderung,

dass alle Fenster ein Property haben müssen, das die erforderliche Glas-Art im Fenster vorgibt. Mit einem Regelprüfprogramm oder einem anderen Algorithmus sollte dann überprüft werden, ob Fenster in Toilettenräumen undurchsichtiges Glas haben oder nicht. Hier gibt es eine Grauzone, da IDS Einschränkungen der Werte zulässt. Zukünftige Versionen von IDS werden diesen Bereich weiter verfeinern oder die Möglichkeiten von IDS zur Definition von Regeln erweitern. Praktische Anwendungsfälle werden die künftigen Möglichkeiten von IDS bestimmen.

### 3.7.6 Neue Möglichkeiten mit IDS

IDS bietet neben der Einbindung der Informationsanforderungen in den automatisierten openBIM-Prozess auch neue Möglichkeiten zur gezielten Definition dieser Anforderungen mithilfe eines Anwendungsbereichs. Klassische AIA definieren Informationsanforderungen auf Basis von IFC-Klassen und für *Predefined Types*. IDS kann Informationsanforderungen dagegen in Abhängigkeit aller beschriebenen *Facet-Parameter* definieren. Beispielsweise kann dadurch ein bestimmtes *Property* in einem bestimmten *Property Set* erst notwendig werden, wenn ein anderes *Property* in einem anderen *Property Set* einen bestimmten Wert annimmt. Das ermöglicht Auftraggebern, sehr gezielt Informationen zu fordern und vor allem diese abzufragen.

### 3.7.7 IDS im Detail

Alle technischen Informationen über IDS sind auf GitHub zu finden, wo Codeentwicklung, Dokumentation und Beispiele gespeichert sind. IDS wurde international als die vorteilhafteste Methode für die automatisierte Prüfung der Konformität durch Validierung der alphanumerischen Informationsanforderungen identifiziert. Es unterstützt die Erstellung von Informationsanforderungen, indem es den Benutzern eine Reihe von Möglichkeiten bietet, was von den Modellen verlangt werden kann.

### 3.7.8 Beziehung zu anderen Initiativen

Es gibt viele Möglichkeiten, den Informationsbedarf zu definieren. Excel scheint die gängigste zu sein, hat aber seine Grenzen. Andere Initiativen sind die Product Data Templates (PDTs), Level of Information Need (LOIN), Exchange (oder Employer) Information Requirements (EIR/AIA), BIM-Abwicklungsplan (BAP), der *exchange*-Teil von mvdXML, SHACL in den Linked Data Domains und andere. Alle diese Initiativen haben Vorteile und Grenzen. Je nach Anwendungsfall können andere Standards oder Initiativen die bessere Wahl sein. Ein von Tomczak et al. erstellter Vergleich ist hier zu finden (siehe QR-Code und Tabelle).

Für die meisten Anwendungsfälle in openBIM ist IDS die empfohlene Lösung, um Informationsanforderungen zu definieren. Es schafft ein Gleichgewicht zwischen Kompatibilität mit IFC und bSDD einerseits und Benutzerfreundlichkeit und Zuverlässigkeit auf der anderen Seite. Es gibt verschiedene Software-Tools, um eine IFC-Datei mit den Anforderungen einer IDS-Datei zu vergleichen. In der Regel werden die Ergebnisse in einem Viewer angezeigt. Für die gemeinsame



Nutzung der Ergebnisse wird die Verwendung des BIM Collaboration Format (BCF) empfohlen. BCF ist eine strukturierte Methode zum Austausch von Informationen über IFC-Objekte mit Projektpartnern.

○ – No  
 ◐ – Partial  
 ● – Yes  
 \* – under development  
 © 2022 Tomczak, van Berlo, Krijnen, Borrmann, Bolpagni

	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	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

### 3.7.9 Möglichkeiten zur Visualisierung von IDS

In diesem Abschnitt wird das Beispiel der Informationsanforderung für Räume aus der Einleitung verwendet, um verschiedene Möglichkeiten zur Visualisierung von IDS aufzuzeigen. Die Anforderung lautet: »Alle Raumdaten in einem Modell müssen als [AT]Zimmer klassifiziert sein und die Properties *NetFloorArea* und *GrossFloorArea* (beide im Set *BaseQuantities*) und ein Property namens *AT\_Zimmernummer* im *PropertySet Austria\_example* haben.« Die Formatierung dieser menschenlesbaren Anforderung in einem IDS sieht wie folgt aus:

```
<ids:ids xmlns:xs="https://www.w3.org/2001/XMLSchema" xmlns:ids="http://standards.buildingsmart.org/IDS">
  <ids:info>
    <ids:title>Austria example</ids:title>
    <ids:copyright>buildingSMART</ids:copyright>
    <ids:version>0.0.3</ids:version>
    <ids:description>A few example checks</ids:description>
    <ids:author>contact@buildingSMART.org</ids:author>
    <ids:date>2023-01-16+01:00</ids:date>
  </ids:info>
  <ids:specifications>
    <ids:specification minOccurs="1" ifcVersion="IFC2X3 IFC4" name="Spaces">
      <ids:applicability>
        <ids:entity>
          <ids:name>
            <ids:simpleValue>IFCSPACE</ids:simpleValue>
          </ids:name>
        </ids:entity>
      </ids:applicability>
    </ids:specification>
  </ids:specifications>
</ids:ids>
```



