

openBIM in der Wasserwirtschaft am Beispiel der Abwasserableitung

Mit dem openBIM-Datenmodell Industry Foundation Classes (IFC) können Einzelbauteile bis hin zu ganzen Kläranlageneinzugsgebieten beschrieben werden. Auch ein Austausch von herstellernerutralen Produkten mit herstellerspezifischen Produkten in einem Gesamtmodell ist möglich. Mit der Übertragung der Daten aus einem Gesamtmodell mit hinterlegter Leitungsplanung an eine Baggersteuerung wird ein Beispiel aus der Praxis vorgestellt.

Bernhard Simon Bock

Building Information Modelling (BIM) und das hier beschriebene openBIM mit offenen und weltweit einheitlichen Standards hatte sich zuerst im Hochbau etabliert. Nun zieht openBIM auch zunehmend in andere Bereiche des Bauwesens ein. Anhand von Beispielen aus der Abwasserableitung werden Anwendungsmöglichkeiten in der Wasserwirtschaft beschrieben.

Das IFC-Modell als Datencontainer für Kläranlageneinzugsgebiete

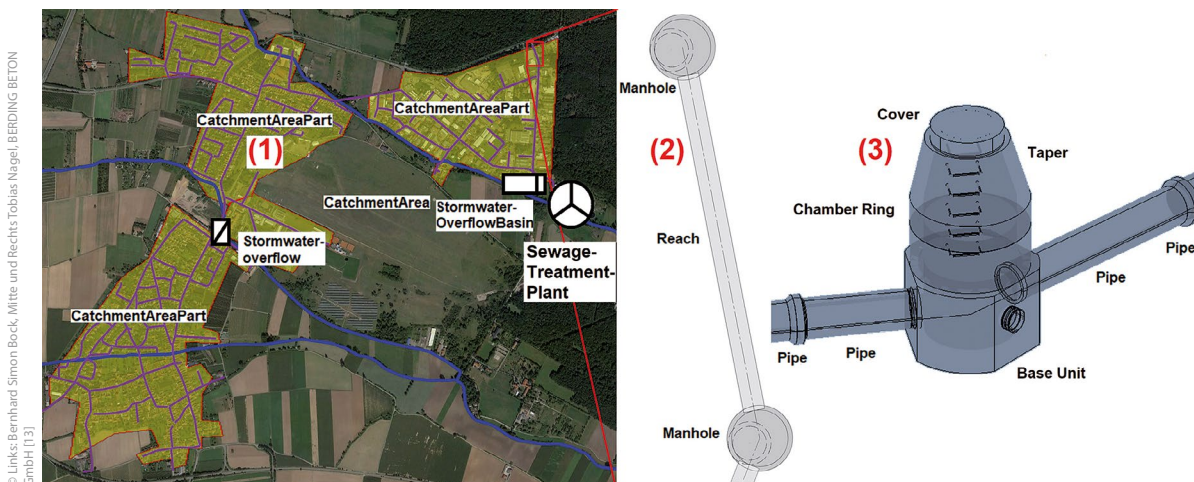
Bild 1 zeigt typische Informationsaggregationen von makroskopischer Sichtweise (z. B. bei Schmutzfrachtberechnungen) über mesoskopische Sichtweise (für Kanalinspektion oder hydraulische Berechnungen) bis hin zur mikroskopischen Betrachtung (z. B. für Vorhaltung von Ersatzbauteilen und detaillierte Informationssysteme). In **Bild 1** ist links das Einzugsgebiet (CatchmentArea in Ein-Wort-Schreibweise in Anlehnung an eine Datenmodellierungsbezeichnung) einer Kläranlage (SewageTreatmentPlant) mit drei Teileinzugsgebieten (CatchmentAreaPart), die durch örtliche Abgrenzung oder an Bauwerksschnittstellen, hier ein Regenüberlaufbecken (StormwaterOverflowBasin) und ein Regenüberlauf (StormwaterOverflow), abgegrenzt werden. In der

/ Kompakt /

- Neben dem Hochbau ziehen Building Information Modelling (BIM) und openBIM zunehmend auch in andere Bereiche des Bauwesens ein.
- Nicht nur Bauteile, sondern z. B. auch ganze Kläranlageneinzugsgebiete können mit dem openBIM-Standard IFC beschrieben werden.
- Bauteilnormen müssen für einen openBIM-Workflow maschinenlesbar sein und müssen nun dahingehend ergänzt und angepasst werden.

Mitte ist eine Abwasserkanalhaltung (Reach) zwischen zwei Schächten (Manhole) dargestellt. Rechts sind in **Bild 1** die Schachtbauteile Deckel (Cover), Konus (Taper), Schachtring (Chamber Ring) und Schachtunterteil (Base Unit) und angeschlossene Rohre (Pipe) abgebildet.

Werden die oben aufgezeigten Plandarstellungen auf eine Schema-Darstellung übertragen, so ergibt sich der Systemplan in **Bild 2**, der auch eine Zuordnung von IFC-Elementen enthält.



© Links: Bernhard Simon Bock, Mitte und Rechts: Tobias Nagel, BERDING BETON GmbH [13]

Bild 1: Makroskopische bis mikroskopische Betrachtung von Entwässerungssystemen [6]

Sowohl in **Bild 1** als auch in **Bild 2** ist ein Teileinzugsgebiet mit (1), eine Haltung mit (2) und ein Rohr mit (3) gekennzeichnet.

Bild 3 zeigt einen Auszug aus der Beschreibung eines Kläranlageneinzugsgebietes, wie es in **Bild 1** und **2** dargestellt ist, im IFC-STEP-Format. Das IFC-STEP-Format wurde bereits in [1] beschrieben (s. dort Bild 4). Zur besseren Lesbarkeit wurden Syntax-Hervorhebungen sowie Groß-/Kleinschreibung verwendet.

Für die nachfolgenden Beschreibungen wird die folgende Formatierung verwendet, bei der die Begriffe Attribut und Eigenschaft sowie die Begriffe Typ, Klasse und IFC-Entity hier gleichbedeutend betrachtet werden.

<i>IfcZone</i>	Name eines Typs/einer Klasse/eines IFC-Entity
<i>Name</i>	Name eines Attributes oder Eigenschaft eines Typs
„ <i>Reach</i> “	Wert eines Attributes
<i>ifcLabel</i>	Typ eines Attributes
<i>.SEWAGE.</i>	Aufzählungswert
<i>#274</i>	Kennziffer (Entity-Id) in einer IFC-STEP-Datei

Zur Erläuterung der nachfolgend verwendeten Begrifflichkeiten „Vererbung“ und „Ableitung“ zunächst ein Vergleich zu beispielhaften biologischen Typbeschreibungen:

Ein Typ *Nadelbaum* mit der Eigenschaft „hat Nadeln“ erbt die Eigenschaften eines Typs *Baum* mit der Eigenschaft „hat einen Stamm und Äste“. Man sagt auch, der Typ *Nadelbaum* ist vom Typ *Baum* „abgeleitet“. Ferner sagt man auch, der Typ *Baum* „erbt“ die Eigenschaften eines Typs *Pflanze* mit der Eigenschaft „hat Wurzeln“. Diese Vorgehensweise der Ableitung bzw. Vererbung hat den Vorteil, dass Eigenschaften von Typen nicht immer wieder neu definiert werden müssen. Ein weiterer Vorteil besteht darin, dass man verschiedene Typen zusammengefasst ansprechen kann, z. B. dass alle Arten des Typs *Pflanze* Wasser benötigen, womit dann auch die Typen *Baum* und *Nadelbaum* erfasst sind. Gleichzeitig kann man auch sagen: Ein *Nadelbaum* „ist ein“ *Baum*, ein *Baum*

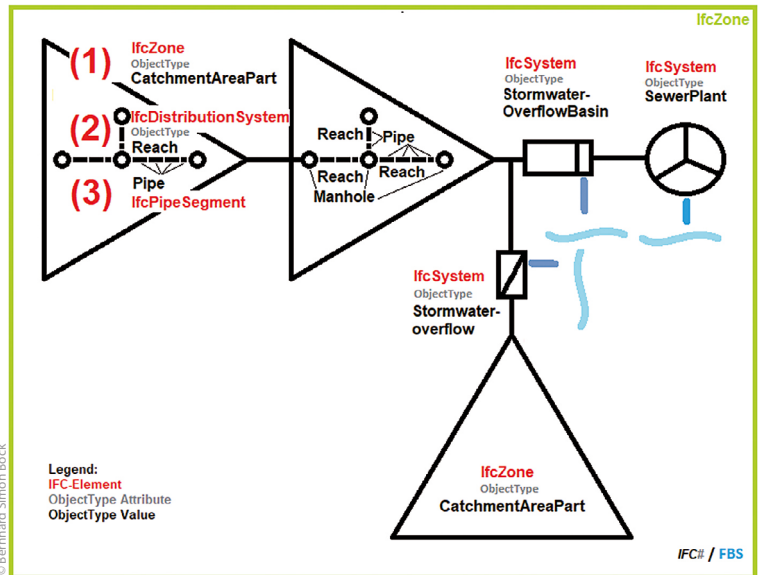


Bild 2: Entwässerungssystemdarstellung als Systemplan mit Zuordnung zu bereits bestehenden IFC-Elementen [6]

„ist eine“ *Pflanze* oder ein *Nadelbaum* „ist eine“ *Pflanze*. Tatsächlich gibt es zwar auch Pflanzen ohne Wurzeln (z. B. Moose). Für diesen vereinfachten Erläuterungszweck wollen wir aber darüber hinwegsehen.

Wenn man also einen neuen Typ von einem übergeordneten Typ ableitet, bedeutet das, dass man die Eigenschaften des übergeordneten Typs erbt. Die Systematik der Vererbung an abgeleitete Typen ist in der Programmierung ein wesentlicher Bestandteil der „Objektorientierung“. Dort werden die hier genannten Typen i. d. R. auch „Klassen“ genannt, die bei IFC im zugrundeliegenden EXPRESS-Schema (s. Erläuterung in [1]) mit „Entities“ bezeichnet werden (Einzahl „Entity“).

```

/* Define CatchmentArea */
#266=ifcZone('2iToC6jRPliQRqGR6rlrc6', $, 'CatchmentArea of Sewage Plant', $, 'CatchmentArea', $);
/* Define CatchmentAreaParts */
#267=ifcZone('2_OKsHlBn1lgtDg757jFcj', $, 'CatchmentAreaPart1', $, 'CatchmentAreaPart', $);
#268=ifcZone('3$ouJkZ8rlqxUoc8vMAf$d', $, 'CatchmentAreaPart2', $, 'CatchmentAreaPart', $);
#269=ifcZone('OpVsrCs0b4xxKLxhViZEgv', $, 'CatchmentAreaPart3', $, 'CatchmentAreaPart', $);
/* CatchmentAreaPart2,3 flow into CatchmentAreaPart1 */
#270=ifcRelNests('1SI4Vi7D9EXj75oULLkKp', $, 'ToCatchmentAreaPartFromOneOrManyCatchmentAreaParts', $, #267, (#268, #269));
/* CatchmentAreaPart1,2,3 are Part of CatchmentArea of Sewage Plant */
#271=ifcRelAssignsToGroup('3jPe6E19HDMgLLhfXIOc$P', $, 'CatchmentAreaOfCatchmentAreaParts', $, (#267, #268), .GROUP., #266);
/* Assign Reaches to CatchmentAreaPart */
#272=ifcRelAssignsToGroup('OLEY5Ta9TAiwtIdxDR_eZW', $, 'ReachToCatchmentAreaPart', $, (#273, #458), .GROUP., #267);
/* Reach */
#273=ifcDistributionSystem('1mCXZ13Xz7GgvT_fd_wZOV', $, '160.1-160', $, 'Reach', $, .SEWAGE.);
#274=ifcRelAssignsToGroup('0shUFYPSv8lPWZwaYShvAh', $, 'PipeToReach', $, (#283, #297, ...), .GROUP., #273);
/* pipe-instance: */
../../../../SewerLibrary/SewerLib/Pipes/doc/img/pipe_instances.png
#275=ifcCartesianPoint(28.6040, 9.0020, 0.1800);
#276=ifcDirection(-6.1330, -7.3807, 1598.7539);
#277=ifcDirection(25.5540, 30.7530, 0.2400);
#278=ifcAxis2Placement3D(#275, #276, #277); /* refer to the to-point of the reach, z-normal and direction */
#279=ifcLocalPlacement($, #278);
#280=ifcCartesianPoint(0.5000, 0.0000, 0.0000); /* pipe-distance from the to-point of the reach */
#281=ifcAxis2Placement3D(#280, #4, #2);
#282=ifcLocalPlacement(#279, #281);
#283=ifcPipeSegment('3qKhb3$Y53u8PwIk7Xhw$E', $, '160.1-160_0,50', $, 'Pipe', #282, #95, $, $);

```

Bild 3: Beschreibung eines Kläranlageneinzugsgebietes im IFC-STEP-Format (Auszug)

Abbildung von Einzugsgebieten in IFC-STEP

Für die Beschreibung von Einzugsgebieten wurde in **Bild 3** in Zeile #266 der Typ *ifcZone* verwendet. *IfcZone* dient dazu, Bereiche zu gruppieren und wurde hier für die Beschreibung von Einzugsgebieten und untergeordneten Teileinzugsgebieten verwendet.

Die Formulierung „wurde verwendet“ resultiert aus der Tatsache, dass es durchaus unterschiedliche Möglichkeiten der Abbildung von Objekten (hier: Einzugsgebiet) innerhalb von IFC gibt, wenn sie noch nicht explizit in dem Standard aufgenommen sind. Da es im Bereich der Entwässerung noch keine verbindliche Festlegung hierzu gibt, wurde jeweils die in diesem Beitrag beschriebene Verfahrensweise gewählt, die damit auch ein Vorschlag zur Abbildung von Abwassersystemen im IFC-Datenmodell ist.

Der *IfcZone*-Eintrag wird in **Bild 4** näher beschrieben. Im Kasten unterhalb des Eintrags #266=*IfcZone* in **Bild 4** sind die Vererbungsstufen vom Typ *IfcZone* dargestellt und das Attribut 6 *LongName* vom Typ *Label*, das *IfcZone* zugeordnet ist. In den letzten beiden Zeilen ist rechts dargestellt, dass *IfcZone* von *IfcSystem* abgeleitet ist.

Bei *IfcZone* gibt es wie beim o. g. Beispiel *Nadelbaum* auch eine mehrstufige Ableitung bzw. Vererbung.

Bei der Ableitung *IfcZone* von *IfcSystem* werden zunächst keine direkten Eigenschaften von *IfcSystem* vererbt, da *IfcSystem* keine eigenen Eigenschaften hat. Was aber damit dennoch zum Ausdruck gebracht wird: *IfcZone* „ist ein“ *IfcSystem*-Typ. Gleiches gilt auch für die Ableitung *IfcSystem* von *IfcGroup* in der drittletzten rechten Zeile des unteren Beschreibungskastens in **Bild 4**. Auch hier bedeutet es: *IfcZone* „ist ein“ *IfcGroup*-Typ. Auf diese Eigenschaft kommen wir später noch bei der Verknüpfung von Typen zurück. Weiterhin ist *IfcGroup* von *IfcObject* abgeleitet. *IfcObject* besitzt das Attribut 5 *ObjectType* vom Typ *Label*, welches nun *IfcZone* von *IfcObject* erbt. *IfcObject* ist von *IfcObjectDefiniton* ohne Eigenschaften abgeleitet und *IfcObjectDefiniton* von *IfcRoot*, welches die Eigenschaften 1 – 4 besitzt, die dann *IfcZone* auch erbt.

Diese sehr vielschichtige Typendefinition ist erforderlich, um die zwischenzeitlich über tausend IFC-Typen (Entities) sinnvoll zu organisieren.

In **Bild 4** folgt nach der Typenbezeichnung *IfcZone* nach der Klammer das erste Attribut (Eigenschaft) *GloballyUniqueId* mit dem Wert „2iToC6jRP1iQRqGR6rlrc6“. Dieser für praktische Zwecke als global eindeutig angenommene Wert [17], der von der erzeugenden Software generiert wird, kann bei Bedarf als Zugriffsschlüssel oder Identifikator verwendet werden. In unserem Anwendungsfall wollen wir aber die Kombination der Attribut-Werte (Eigenschaftswerte) von *ObjectType* und *Name* als eindeutig ansehen. Das zweite Attribut von *IfcZone* in **Bild 4** heißt *OwnerHistory* und ist optional. Da es hier nicht verwendet wird, wird es mit \$ (bedeutet im STEP-Format NULL oder „ohne

Wert“) belegt. Das dritte Attribut *Name* wird hier für eine eindeutige Bezeichnung des Einzugsgebietes verwendet. Das vierte Attribut *Description* ist optional und wird hier nicht verwendet und daher mit \$ belegt. Die Information, dass es sich bei dem *IfcZone*-Eintrag um ein Einzugsgebiet handelt, wird hier durch die Verwendung des fünften Attributes *ObjectType* mit dem Wert „CatchmentArea“ festgelegt. Das sechste Attribut *LongName* ist optional und wird – da es hier nicht verwendet wird – mit \$ belegt.

Eine vollständige Beschreibung von *IfcZone* und allen anderen IFC-Entities befindet sich in der IFC-Dokumentation [8]. Dort ist auch eine umfangreiche Darstellung der oben beschriebenen Vererbungshierarchie zu finden.

Abbildung von Einzugsgebieten in IFC mit weiteren Methoden und Formaten

Das in **Bild 3** verwendete STEP-Format ist das am Häufigsten verwendete Format für IFC-Modelle. Der Vorteil der Verwendung des STEP-Formates (eine Zeile je Entity mit vorangestellter Raute „#“) ist die sehr kompakte, aber dennoch menschenlesbare Schreibweise. In [1] wurde bereits beschrieben, dass es sich bei den Industrie Foundation Classes (IFC) nicht um ein Dateiformat handelt, sondern um ein Datenmodell, welches über verschiedene Datenformate oder Methoden beschrieben werden kann, die nachfolgend auszugsweise angesprochen werden. Tatsächlich gibt es deutlich mehr Formate [18] und Methoden [19].

Abbildung von Einzugsgebieten in IFC#

Die Abbildung des Einzugsgebiets mit *IfcZone* in der Programmierschnittstelle IFC# [4] sieht so aus:

```
new ifc.Zone(GlobalId:null,Name:"CatchmentArea of Sewage Plant",ObjectType:"CatchmentArea");
```

Mit IFC# kann z.B. auch eine IFC-Datei im o. g. STEP-Format erzeugt werden. Vorteil der Verwendung von IFC# zur Erzeugung von IFC-Dateien ist, dass nur Objekte erzeugt und miteinander kombiniert werden können, die das IFC-Schema zulässt, da das IFC-Schema in der IFC#-Klassenbibliothek abgebildet ist. Das hier beschriebene Beispiel wurde mit der Programmierschnittstelle IFC# erzeugt.

Abbildung von Einzugsgebieten in ifcXML

Mit ifcXML, welches von buldingSMART standardisiert ist, sähe der Eintrag für *IfcZone* so aus:

```
<IfcZone> (hier stehen die Attributeinträge) </IfcZone>
```

Vorteil von ifcXML ist, dass XML wegen seiner weit größeren Verbreitung gegenüber dem STEP-Format von den meisten Standard-Programmibliotheken gelesen werden kann.



Bild 4: Verwendung des *IfcZone*-Elementes zur Abbildung von Einzugsgebieten

Abbildung von Einzugsgebieten in ifcSQL

ifcSQL [10] ist eine Möglichkeit, IFC-Modelle in einer Datenbank zu speichern. ifcSQL ist kein Standard von buildingSMART, ermöglicht aber das schemasichere Speichern von IFC-Modellen. Das Einfügen von *IfcZone* via SQL beginnt so:

```
DECLARE @TypeId int = (SELECT [TypeId] FROM [cs].[Type] WHERE [TypeName]='Zone')
INSERT INTO [ifcInstance].[Entity] ([GlobalEntityInstanceId], [EntityTypeId]) VALUES (@CurrentGlobalId,@TypeId)... (danach folgen Attributeinträge)
```

Vorteil von ifcSQL ist, dass mehrere Modelle in einer Datenbank gespeichert, abgefragt und bei Bedarf über alle Modelle hinweg modifiziert oder ergänzt werden können.

Teileinzugsgebiete

Gehen wir nun in der Beschreibung des Beispiels in **Bild 3** weiter. Ein Beispiel für ein Teileinzugsgebiet ist in den **Bilder 1** und **2** mit (1) gekennzeichnet.

Teileinzugsgebiete wurden analog zu Einzugsgebieten in den Zeilen #267, #268 und #269 in **Bild 3** mit *IfcZone* abgebildet. Die Information, dass es sich bei dem *IfcZone*-Eintrag um ein Teileinzugsgebiet handelt, wurde durch die Verwendung des Attributes *ObjectType* mit dem Wert „*CatchmentAreaPart*“ festgelegt.

Gerichtete Verknüpfung von Teileinzugsgebieten

Die gerichtete Verknüpfung von Teileinzugsgebieten (von welchen anderen Teileinzugsgebieten erhält ein Teileinzugsgebiet Zufluss?) wurde in Zeile #270 (**Bild 3**) mit dem *IfcRelNests*-Typ abgebildet. Die Information, dass es sich bei dem *IfcRelNests*-Eintrag um eine gerichtete Verknüpfung von Teileinzugsgebieten handelt, wurde durch die Verwendung des Attributes *Name* mit dem Wert „*ToCatchmentAreaPartFromOneOrManyCatchmentAreaParts*“ festgelegt.

Gruppierung von Teileinzugsgebieten zu Einzugsgebieten

Die Gruppierung von *Teileinzugsgebieten* zu *Einzugsgebieten* wurde in Zeile #271 mit dem *ifcRelAssignsToGroup*-Typ abgebildet. Die Information, dass es sich bei dem *ifcRelAssignsToGroup*-Eintrag um eine Gruppierung von Teileinzugsgebieten zu einem Einzugsgebiet handelt, wird hier durch die Bezeichnung des Attributes *Name* mit „*CatchmentAreaOfCatchmentAreaParts*“ festgelegt.

Zusätzlich wurde das Attribut *RelatedObjectsType* mit dem *ifc-ObjectTypeEnum*-Aufzählungswert *.GROUP*. belegt, um klarzustellen, dass es sich um eine Gruppenzuordnung handelt. Das ist erforderlich, da *RelatedObjectsType* auch Werte wie *.PRODUCT*. oder *.PROJECT*. annehmen kann.

Gruppierung von Haltungen zu einem Teileinzugsgebiet

Auch die Gruppierung von Haltungen zu einem Teileinzugsgebiet wurde mit dem *ifcRelAssignsToGroup*-Typ abgebildet, hier in Zeile #272. Die Information, dass es sich bei dem *ifcRelAssignsToGroup*-Eintrag um eine Gruppierung von Haltungen zu einem Teileinzugsgebiet handelt, wurde durch die Bezeichnung des Attributes *Name* mit „*ReachToCatchmentAreaPart*“ festgelegt.

Abbildung von Haltungen

Eine Haltung (2) (**Bilder 1** und **2**) wurde in Zeile #273 mit dem *IfcDistributionSystem*-Typ abgebildet. *IfcDistributionSystem* dient dazu, funktionale Bereiche zu gruppieren und wird hier für die Abbildung von Haltungen verwendet. Die Bezeichnung „DistributionSystem“ (Verteilungssystem) ist hier zwar keine treffende Beschreibung, wird aber innerhalb von IFC sowohl für Verteilungssysteme als auch Sammlungssysteme – wie im hier vorliegenden Fall – verwendet. Es ist denkbar, dass in einer künftigen IFC-Version ein Element „*IfcCollectingSystem*“ ergänzt wird, was auch in der Entwässerung innerhalb von Gebäuden im Bereich der technischen Gebäudeausrüstung (TGA) sinnvoll ist. Bis dahin muss man aber auf *ifcDistributionSystem* zurückgreifen. Die mit *IfcDistributionSystem* abgebildete Haltung (2) wird in **Bild 5** näher beschrieben.

IfcDistributionSystem ist ähnlich wie *IfcZone* von *IfcSystem* (und *IfcGroup*) abgeleitet und wird daher nicht weiter beschrieben.

Die Information, dass es sich bei dem *IfcDistributionSystem*-Eintrag um eine Haltung handelt, wurde durch die Bezeichnung des *ObjectType* mit „*Reach*“ festgelegt.

Zusätzlich wurde das Attribut *PredefinedType* mit dem *ifc-DistributionSystemEnum*-Aufzählungswert *.SEWAGE*. belegt, damit erkennbar ist, dass es sich um ein Abwassersystem handelt. Denkbar sind hier auch weitere mögliche vom IFC-EXPRESS-Schema vorgegebene Aufzählungswerte wie *.DRAINAGE*., *.RAINWATER*., *.STORMWATER*. oder *.WASTEWATER*.

Das Attribut *Name* wurde für die Haltungsbezeichnung – hier „*160.1-160*“ – verwendet.

Zuordnung von Einzelrohren zu einer Haltung

Die Zuordnung von Einzelrohren zu einer Haltung wurde in Zeile #274 ebenfalls mit dem *ifcRelAssignsToGroup*-Typ abgebildet. Die Information, dass es sich bei dem *ifcRelAssignsToGroup*-Eintrag um eine Zuordnung von Einzelrohren zu einer Haltung handelt, wurde durch die Bezeichnung des Attributes *Name* mit „*PipeToReach*“ festgelegt.

Die Zuordnung mit *ifcRelAssignsToGroup* für *IfcZone* und *IfcDistributionSystem* ist deshalb möglich, weil sowohl *IfcZone* als auch *IfcDistributionSystem* beide von *IfcGroup* abgeleitet sind.

GloballyUniqueId Globallid='1mCXZ13Xz7GgvT_fd_wZ0V'

Label Name='160.1-160'

Label ObjectType='Reach'

PredefinedType=SEWAGE

```
#273=ifcDistributionSystem ('1mCXZ13Xz7GgvT_fd_wZ0V', $, '160.1-160', $, 'Reach', $, .SEWAGE.);
```

ifcDistributionSystem	Inheritance of ifcDistributionSystem:	OwnerHistory _OwnerHistory=\$(null)	Text Description=\$(null)	Label LongName=\$(null)
1 GloballyUniqueId Globallid	[of class ifc.Root], ifcRoot			
2 OwnerHistory _OwnerHistory	[of class ifc.Root], ifcObjectDefinition			
3 Label Name	[of class ifc.Root], ifcObject			
4 Text Description	[of class ifc.Root], ifcGroup			
5 Label ObjectType	[of class ifc.Object], ifcSystem			
6 Label LongName	[of class ifc.DistributionSystem], ifcDistributionSystem			
7 Enum PredefinedType	[of class ifc.DistributionSystem]			

Bild 5: Verwendung des *IfcDistributionSystem*-Elementes zur Abbildung von Haltungen

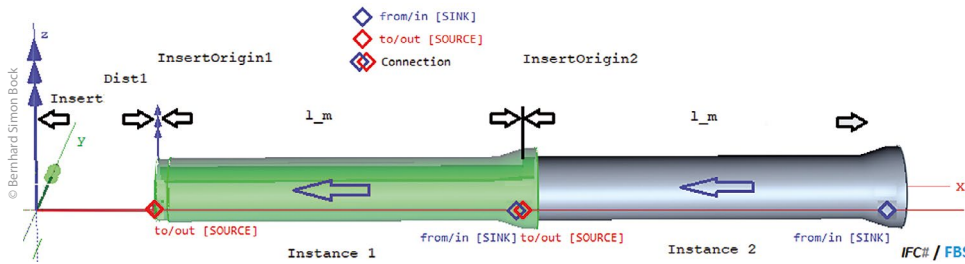


Bild 6: Kommentar-Grafik oberhalb Zeile #275

Abbildung von Einzelrohren

In [1] wurde bereits die mögliche Abbildung eines einzelnen Rohres in IFC erläutert und soll hier nicht wiederholt werden. Neu ist aber, dass die hier verwendeten Rohre nur einmal als Typ definiert wurden und dann nur noch eingefügt werden. In Zeile #283 wird das erste Rohr, welches als Typ in Zeile #95 (nicht in Bild 3 enthalten) definiert wurde, nun im lokalen Koordinatensystem der Haltung in Zeile #282 mit *ifcLocalPlacement* eingefügt. Das erste Attribut *PlacementRelTo* von *ifcLocalPlacement* in Zeile #282 bezieht sich auf die lokale Platzierung, die in #279 definiert ist und das zweite Attribut *RelativePlacement* auf den lokalen Einfügekpunkt mit zwei 3D-Achsansgaben *ifcAxis2Placement3D* in Zeile #281.

Das erste Attribut *Location* von *ifcAxis2Placement3D* in Zeile #281 enthält im Verweis auf Zeile #280 den Abstand vom Haltungsanfang (hier Schachtmitte) mit 0,5 m. Die Platzierung des zweiten 3 m langen Rohres (in Bild 3 nicht enthalten) unterscheidet sich von der Platzierung des ersten Rohres nur noch durch den Abstand zum Haltungsanfang mit einem Wert von 3, 5 m (0,5 m Abstand Schachtmitte zzgl. 3 m vorheriges Rohr).

Das zweite Attribut *Axis* von *ifcAxis2Placement3D* in Zeile #281 enthält den Verweis auf die lokale Z-Achse über Zeile #4 = *ifcDirection(0,0,1)* (in Bild 3 nicht enthalten). Das dritte Attribut *RefDirection* enthält den Verweis auf die lokale X-Achse über Zeile #2 = *ifcDirection(1,0,0)* (in Bild 3 nicht enthalten).

Die o. g. lokale Platzierung *ifcLocalPlacement* in Zeile #279 bezieht sich mit dem einzigen verwendeten zweiten Attribut *RelativePlacement* nur noch auf den lokalen Einfügekpunkt mit zwei 3D-Achsansgaben *ifcAxis2Placement3D* in Zeile #278.

Der *ifcAxis2Placement3D*-Eintrag in Zeile #278 verweist mit dem ersten Attribut *Location* auf die Zeile #275, wo sich die Koordinate des Haltungsanfangs (Schachtmitte) befindet. Das in Zeile #275 verwendete Koordinatensystem ist ein lokales Koordinatensystem des vorliegenden IFC-Modells, welches eine Zuordnung zum globalen UTM-Koordinatensystem besitzt. Die Zuordnung zum UTM-Koordinatensystem wurde bereits in [1] beschrieben.

Das zweite Attribut *Axis* von *ifcAxis2Placement3D* in Zeile #278 verweist auf die Zeile #276. Dort ist die Richtung der Z-Achse des lokalen Haltungskoordinatensystems in den globalen Koordinaten dieses IFC-Modells definiert. Die Z-Achse ist entsprechend dem Haltungsgefälle geneigt. Durch das geringe Gefälle ist die Z-Komponente (3.Wert) groß gegenüber den X- und Y-Komponenten.

Das dritte Attribut *RefDirection* von *ifcAxis2Placement3D* in Zeile #278 verweist auf die Zeile #277. Dort ist die Richtung der X-Achse des lokalen Haltungskoordinatensystems in den globalen Koordinaten dieses IFC-Modells definiert. Auch die X-Achse ist entsprechend dem Haltungsgefälle geneigt. Durch das geringe Gefälle ist die Z-Komponente klein gegenüber den X- und Y-Komponenten, welche die ebene Richtung der Haltung angeben. Die noch fehlende

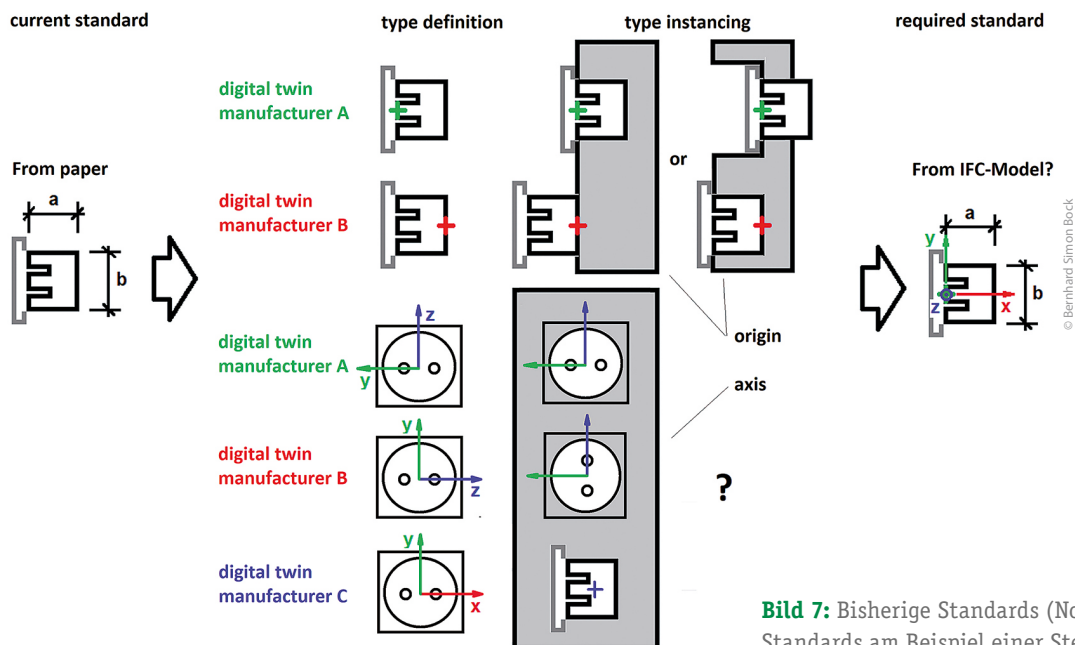


Bild 7: Bisherige Standards (Normen) und erforderliche Standards am Beispiel einer Steckdose [2], [3]

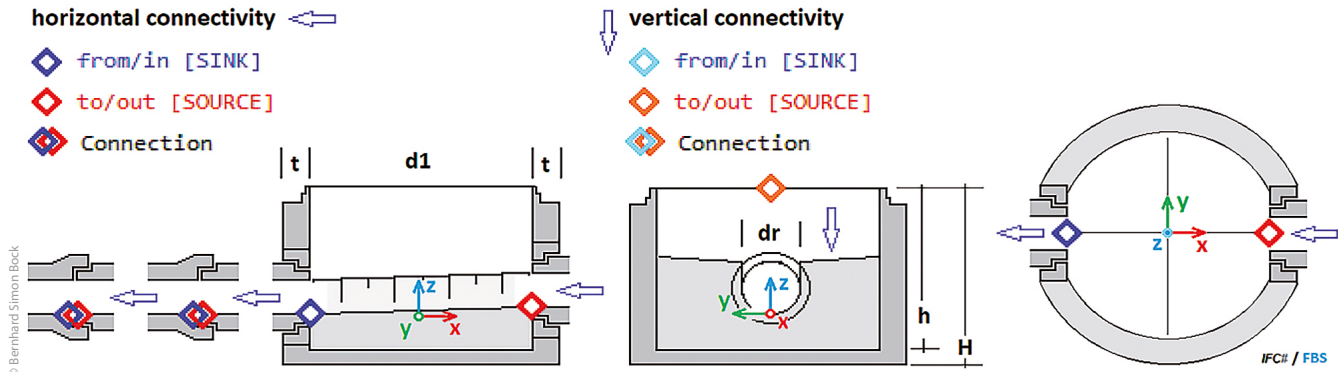


Bild 8: Auszug der vom FBS bereits festgelegten Standards zu Einfügepunkt, Ausrichtung und Konnektivität [4], [5], [6]

Y-Achse des lokalen Haltungskoordinatensystems ergibt sich durch das Kreuzprodukt von Z- und X-Achse.

Oberhalb der Zeile #275 befindet sich ein Link auf eine Erläuterungsgrafik, die in **Bild 6** dargestellt ist. In der Original-IFC-Datei ist dieser Link als Kommentar eingefügt [12].

Die Platzierung der Rohre entsprechend **Bild 6** ist jedoch nur möglich, wenn es eine Festlegung zu Ursprung und Ausrichtung des Bauteilkoordinatensystems (hier: Rohr) gibt, was nachfolgend beschrieben wird.

Standardisierung von openBIM-Bauteilbibliotheken

Vergleich mit einer Steckdose

Am Beispiel einer Steckdose wird illustriert, was passiert, wenn verschiedene Hersteller beim Austausch von Bauteilen in einem Gesamtmodell auf Grund fehlender Festlegungen in aktuellen Normen unterschiedliche Koordinatenursprünge und Ausrichtungen verwenden. In **Bild 7** ist links die derzeitige geometrische Beschreibung von Bauteilen am Beispiel einer Steckdose abgebildet. Würde man sich nur auf diese geometrischen Angaben beziehen, ergibt sich ein sehr großer Interpretationsspielraum, der in der Mitte der Grafik dargestellt ist. Bauteile unterschiedlicher Hersteller lassen sich dann in einem Modell nicht austauschen. Erst wenn es Festlegungen zu Ursprung und Ausrichtung – wie in **Bild 7** rechts dargestellt – gibt, ist ein Austausch von gleichartigen Produkten in einem Modell möglich. Gleiches gilt auch für Rohre, was auch schon in [1] beschrieben wurde.

Für maschinenlesbare Normen ist also anstelle einer Darstellung mit Maßketten auf Papier idealerweise ein herstellerneutrales IFC-Modell geeignet, in dem Ursprung und Ausrichtung festgelegt sind.

Festlegungen der FBS-Fachgruppe BIM

Derzeit gibt es noch keinen Standard für Ursprung und Ausrichtung von digitalen Bauteilzwillingen. Aus diesem Grund wurde von der FBS-Fachgruppe BIM [5] ein Standard ausgearbeitet, der noch veröffentlicht werden soll. Er enthält Festlegungen zu Ursprung und Ausrichtung von Entwässerungsbauteilen. Die Begründung der Festlegungen wurde ebenfalls dokumentiert. **Bild 8** zeigt die Festlegungen des FBS für Schachtunterteile und Rohre.

Nachfolgend wird auf die Festlegung für Ursprung und Ausrichtung von Rohren genauer eingegangen. Neben schematischen Abbildungen, wie in **Bild 8**, sind auch detailliertere Betrachtungen erforderlich, die in **Bild 9** dargestellt sind.

Für Abwasserrohre wurde festgelegt:

- Die X-Achse verläuft parallel zur Rohrachse entgegen der Fließrichtung,
- Der Nullpunkt/Ursprung liegt am Spitzende auf Sohlhöhe,
- Bei Böschungsstücken liegt der Nullpunkt am Beginn der Sohle,
- Die Z-Achse verläuft senkrecht zur Rohrachse in der Vertikalen nach oben,
- Die Y-Achse ergibt sich aus den Festlegungen der X- und Z-Achse (Kreuzprodukt).

Begründungen für die getroffenen Festlegungen:

Das Spitzende auf Sohlhöhe der Auslaufseite wurde als Koordinatenursprung gewählt, da alle Rohre ein Spitzende haben, aber nicht unbedingt eine Muffe. Das Spitzende ist der Anschlusspunkt an das nächste Bauteil.

Die Z-Achsen-Festlegung wurde gewählt, da die Sohlhöhe in Kanalplanung und Bestandsdokumentation als Bezugsgröße verwendet wird und in der Bauausführung die Bezugsgröße ist. Die Sohlhöhe ist bei allen Querschnittsformen (Eiprofil, Trockenwetterrinne, Maulprofil usw.) am einfachsten zu bestimmen.

Die X-Achsenfestlegung wurde gewählt, da die Richtung parallel zur Rohrachse Standard bei Planung und Bauausführung ist. Die

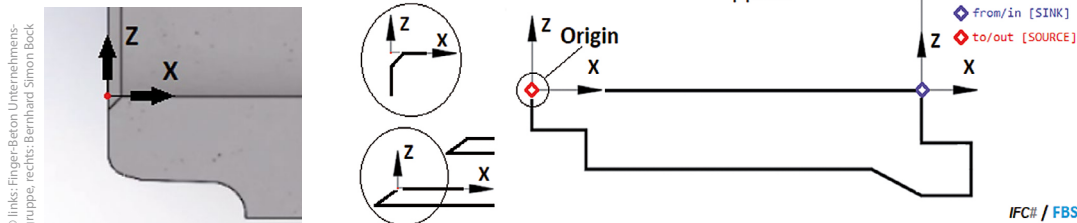
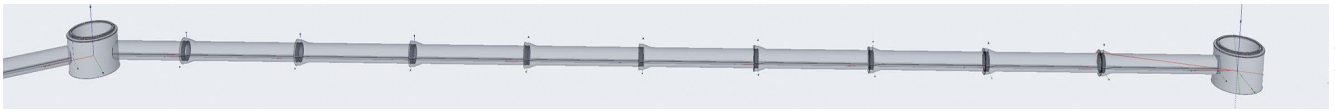


Bild 9: Detaillierter Auszug zu Einfügepunkt, Ausrichtung bei Rohren, links [14], rechts [4], [5], [6]



© Bernhard Simon-Bock

Bild 10: Visualisierung des IFC-Modells aus Bild 3 mit IFC++ [16]

Richtung entgegen der Fließrichtung wurde gewählt, um mit positiven Werten zu arbeiten, da die Verlegung im Normalfall in die gleiche Richtung erfolgt.

Aufbereitung von Herstellerdaten für den Datenaustausch

Um einen Datenaustausch von Herstellerbauteilen im IFC-Standard entsprechend den vorgenannten Festlegungen zu ermöglichen, sind ggf. noch folgende Anpassungen eines exportierten IFC-Modell erforderlich:

1. Anpassungen von Ursprung und Ausrichtung des Koordinatensystems, was aber i. d. R. mit wenigen gezielten Eingriffen im IFC-Modell möglich ist.
2. Umstellung von IFC-Instanztypen (hier: *IfcPipeSegment*) zu IFC-Bauteil-Typen (hier: *IfcPipeSegmentType*), damit diese im Zielformat entsprechend Zeile #283 (Bild 3) eingefügt werden können.
3. Anpassung oder Ergänzung von Typ-Eigenschaften (*IfcProperty*). Typ-Eigenschaftssätze (*IfcPropertySet*) sind in IFC anders zugeordnet als Instanz-Eigenschaftssätze und müssen bei bisheriger Verwendung von Instanz-Eigenschaftssätzen umgestellt werden. Vergabe von Eigenschaftsschablonen für Instanz-Eigenschaften (*IfcPropertySetTemplate* und *IfcPropertyTemplate*), wie zum Beispiel „Einbaudatum“.

Erst wenn das Bauteilmodell die vorgenannten Anforderungen erfüllt, ist in einem Gesamtmodell ein Austausch von herstellernneutralen Produkten mit herstellereigenen Produkten möglich.

Baggersteuerung auf Basis eines IFC-Modells und Anwendung von IDS

Das in Bild 3 beschriebene und in Bild 10 dargestellte IFC-Modell kann auch dazu verwendet werden, den Datensatz für eine Baggersteuerung zu befüllen (Bild 11).

Hierzu wird das IFC-Modell zunächst nach Einträgen von *IfcDistributionSystem* durchsucht, deren Attribut *ObjectType* den Wert „Reach“ hat. Innerhalb jedes der o.g. *IfcDistributionSystem*-Einträge wird nach allen *IfcPipeSegment*-Einträgen gesucht und der Einfügepunkt ermittelt. Aus dem *IfcPipeSegmentType*, auf den *IfcPipeSegment* jeweils verweist, wird das *IfcPropertySet*, dessen Attribut *Name* den Wert „DIN“ hat, ermittelt. Innerhalb des *DIN-IfcPropertySet* werden alle angeschlossenen *IfcPropertySingleValue* mit dem Attribut *Name* ausgewertet, die den Wert „l“ für Länge oder „di“ für Innendurchmesser haben und deren Wert aus dem Attribut *NominalValue* ausgelesen.

Die Vorgehensweise bei der Auswertung des IFC-Modells zeigt, dass erwartet wird, dass bestimmte IFC-Entities und Attributwerte vorhanden sein müssen, um das Modell für diesen Anwendungsfall verwenden zu können. Die Anforderung an das IFC-Modell kann man zwar in Anlehnung an die Beschreibung in diesem Kapitel als Freitext formulieren, womit dann diese Anforderung aber nicht maschinenlesbar wäre. Um Anforderungen an Modelle maschinenlesbar zu formulieren, gibt es von buildingSMART die „Information Delivery Specification“ – IDS [7], die derzeit noch in



© MTS-Schrode AG

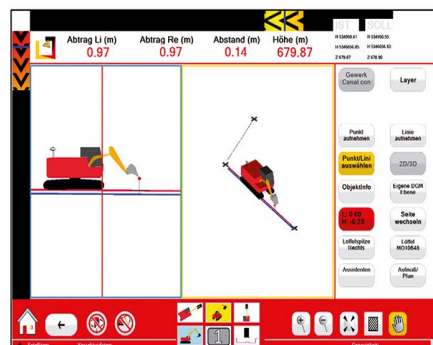
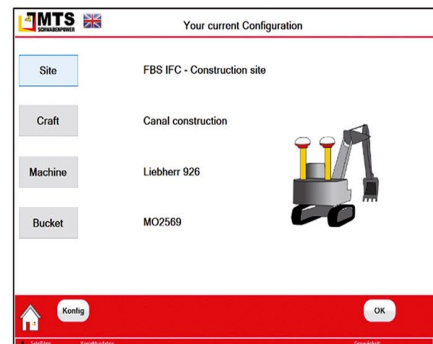


Bild 11: Bagger mit GNSS-Steuerung und hinterlegter Leitungsplanung auf Basis eines IFC-Modells [6], [15]

Entwicklung ist. Mit IDS werden die Anforderungen an ein Modell in einem standardisierten XML-Dokument beschrieben, was dann eine softwarebasierte Auswertung von Modellen ermöglicht, ob sie für den jeweiligen Anwendungsfall geeignet sind. D. h. in einem Modell können beliebige weitere Angaben und Modellteile enthalten sein. Wesentlich ist nur, dass die Angaben für den jeweiligen Anwendungsfall auch enthalten sind.

Zusammenfassung

Die openBIM-IFC-Modell-Systematik kann nicht nur Bauwerke und Bauteile mikroskopisch abbilden, sondern auch eine makroskopische Abbildung von z. B. ganzen Kläranlageneinzugsgebieten in einem Gesamtmodell.

Für eine standardisierte Abbildung von Bauteilen der Entwässerung, aber auch von Einzugsgebietenkennwerten und deren Beziehung untereinander sind weitergehende Standards erforderlich, die über die Beschreibung der IFC-Modell-Systematik hinaus gehen. Hierzu gehören:

1. Geometrische Festlegungen bezüglich Ursprung und Ausrichtung von Bauteilen,
2. Festlegung von spezialisierten Typ-Eigenschaftssätzen und Instanz-Eigenschaftssätzen,
3. Festlegungen zu verwendeten IFC-Typen und Bezeichnungen für Objekttypen (z. B. „Reach“ bei Attribut *ObjectType* in *IfcDistributionSystem* als Festlegung für eine Haltung).

Die ersten beiden Punkte werden derzeit vom FBS [5] im Bereich der Abwasserableitungsbauteile festgelegt. Für Festlegungen zu anderen Bauteilarten bieten sich die jeweiligen Fachverbände an.

Glossar	
BIM	Building Information Modeling, Bauwerks- Informations-Modellierung
openBIM	BIM-Informationsaustausch mit offenen Standards
IFC	Industry Foundation Classes, offenes openBIM-Datenmodell von buildingSMART International
IFC-STEP	„Standard for the Exchange of Product model data“, Produktdatenaustauschformat, bei dem jede Zeile mit einer Raute mit nachfolgender Zahl eingeleitet wird, hier für IFC
XML, ifcXML	„Extensible Markup Language“, Auszeichnungssprache und Datenformat, welches Begriffe mit eckigen Klammern umschließt, hier für IFC z.B. mit <ifcXML> und </ifcXML>
SQL	Structured Query Language (strukturierte Abfragesprache, die in relationalen Datenbanken eingesetzt wird)
EXPRESS-Schema	Meta-Sprache gemäß ISO 10303-11 zur Datenmodellierung, mit der sämtliche Sprachbestandteile definiert werden, hier das IFC-Datenmodell
Attribut	Merkmal oder Eigenschaft, aber auch Sprachelement in Auszeichnungssprachen wie z. B. HTML oder XML
GNSS	global navigation satellite system (globales Navigationssatellitensystem)

Literatur

Der Abruf der nachfolgend genannten Links erfolgte am 29. April 2023.

- [1] Building Information Modeling in der Abwasserableitung mit open-BIM, Bernhard Bock und Eberhard Michaelis, Wasser und Abfall 5/2019
- [2] Online-Veranstaltung: „Erforderliche Normungsanpassungen für BIM – Wenn aus der Unterputzsteckdose plötzlich eine Überputzsteckdose wird“ | buildingSMART Deutschland, 24.06.2021, online
- [3] The Power outlet problem example, Bernhard Simon Bock, General Assembly of Implementers buildingSMART International, Boston (USA) September 2022 / online, <https://www.buildingsmart.org/general-assembly-of-implementers/>
- [4] IFC#, <https://github.com/IfcSharp>
- [5] Bundesfachverband Betonkanalsysteme e.V. (FBS), <https://fbsrohre.de/>
- [6] Avoidance of information loss by implementing an IFC-based process in the life cycle of sewage systems, InfraBIM open 2023, Tampere (Finnland), FBS-Fachgruppe BIM
- [7] Information Delivery Specification IDS, buildingSMART International <https://technical.buildingsmart.org/projects/information-delivery-specification-ids/>
- [8] IFC 4.3.x Documentation, buildingSMART International, <https://ifc43-docs.standards.buildingsmart.org/>
- [9] IFC2x Edition 3 Model Implementation Guide, Version 2.0, Thomas Liebich, 18.05.2009, https://standards.buildingsmart.org/documents/Implementation/IFC2x_Model_Implementation_Guide_V2-0b.pdf,
- [10] ifcSQL, <https://github.com/IfcSharp/ifcSQL>
- [11] Building Information Modeling, Technologische Grundlagen und industrielle Praxis, 2., aktualisierte Auflage, Borrmann, A., König, M., Koch, C., Beetz, J. (Hrsg.), Springer Vieweg, 2021
- [12] IFC in Water Management by Bernhard Simon Bock, General Assembly of Implementers February 2022, online, <https://www.youtube.com/watch?v=HhQKsRinkY&t=6619s>
- [13] BERDING BETON GmbH, <https://www.berdingbeton.de>
- [14] Finger-Beton Unternehmensgruppe, <https://www.finger-beton.de/>
- [15] MTS Schrode AG, <https://www.mts-online.de>
- [16] IFC+- Viewer, <http://www.ifcquery.com/>
- [17] https://de.wikipedia.org/wiki/Universally_Unique_Identifier#Globally_Unique_Identifier
- [18] <https://technical.buildingsmart.org/standards/ifc/ifc-formats/>
- [19] <https://technical.buildingsmart.org/resources/software-implementations/>

Autor

Dipl.-Ing. Bernhard Simon Bock

Hans-Geiger-Str. 12
67434 Neustadt an der Weinstraße
bsbock@bock-nw.de

Hinweis

Der Beitrag ist die Fortführung eines Beitrags in WASSER UND ABFALL vom Mai 2019 [1]. Dort wurde u. a. gezeigt, wie ein einzelnes Rohr mit den Industry Foundation Classes (IFC) von buildingSMART beschrieben werden kann.

Hinweis

Um eine bessere Verständlichkeit für den Leser zu ermöglichen, wurden in diesem Beitrag abweichend von den redaktionellen Vorgaben der WASSER UND ABFALL ausnahmsweise unterschiedliche farbige Hervorhebungen im Text verwendet.