

Open BIM for Infrastructure – mit OKSTRA und IFC Alignment zur internationalen Standardisierung des Datenaustauschs

Julian Amann, André Borrmann
Lehrstuhl für Computergestützte Modellierung und Simulation
Leonhard Obermeyer Center
Technische Universität München

Einleitung: Building Information Modeling

Building Information Modeling (BIM) steht für die durchgängige Nutzung hochwertiger digitaler Daten eines Bauwerks über dessen gesamten Lebenszyklus – angefangen beim Entwurf über die Ausführung und den Betrieb bis hin zum Rückbau (Eastman et al. 2008). Auf diese Weise können Informationsbrüche und die heute häufig vorkommenden manuellen Neueingaben von Daten und Fehler vermieden werden. Im Zentrum von BIM steht das digitale Bauwerksmodell als umfassende digitale Repräsentation des realen Bauwerks, das neben der detaillierten 3D-Geometrie der Bauteile vor allem auch alphanumerische Informationen, bspw. zu Materialien, Bauteiltypen, Kosten usw. beinhaltet.

BIM bietet erhebliche Vorteile in vielen Bereichen der Planung und Ausführung (Borrmann et al. 2015). Das Arbeiten mit einem 3D-Modell sorgt dafür, dass die abgeleiteten Ansichten und Schnitte immer konsistent zueinander sind. BIM verbessert die Koordination der verschiedenen Gewerke und ermöglicht es, Kollisionen frühzeitig zu erkennen und zu beheben. Mengen, die aus einem digitalen Bauwerksmodell ermittelt werden, bieten eine verlässliche Basis für die Ausschreibung, Vergabe und Abrechnung. Für die Vorbereitung der Ausführung kann das 3D-BIM mit dem zeitlichen Bauablauf kombiniert werden, sodass ein 4D-BIM entsteht, das die Verifizierung der Abläufe sowie die Planung und Abwicklung der Baustellenlogistik ermöglicht. Wird dem Bauherrn nach Abschluss des Bauvorhabens ein digitales Bauwerksmodell übergeben, kann er es unmittelbar für das Facility Management nutzen.

Im Hochbau ist die Verbreitung von BIM bereits weit vorangeschritten. Große öffentliche Bauherren in den USA, wie bspw. die General Service Administration oder das US Army Corps of Engineers, setzen bereits seit einiger Zeit auf die modellgestützte Planung. Ähnliches gilt auch für die skandinavischen Länder, allen voran Finnland und Norwegen. In Großbritannien wird ab April 2016 die BIM-basierte Planung für alle Bauvorhaben der öffentlichen Hand verbindlich vorgeschrieben.

Auch in Deutschland besteht zunehmendes Interesse seitens privater und öffentlicher Bauherren, BIM für die Abwicklung von Bauvorhaben einzusetzen. Ein Hauptaugenmerk der öffentlichen Hand liegt dabei auf der Nutzung von BIM für den Infrastrukturbau. In einer viel beachteten Rede hat Alexander Dobrindt, Bundesminister für Verkehr und digitale Infrastruktur, im April 2014 dazu die folgende Stellungnahme abgegeben: „Die Digitalisierung des Bauens bietet Chancen, große Bauprojekte im Zeit- und Kostenrahmen zu realisieren. Bessere Datengrundlagen für alle am Bauprojekt Beteiligten sorgen für Transparenz und Vernetzung. Dadurch können Zeitpläne, Kosten und Risiken früher und präziser ermittelt werden. Modernes Bauen heißt: erst virtuell und dann real bauen.“ Inzwischen wurden vom BMVI die ersten BIM-Pilotprojekte ins Leben gerufen.

Eine wichtige Rolle für den Erfolg von BIM spielt die Verfügbarkeit von offenen Schnittstellen zum verlustfreien Austausch von hochwertigen Bauwerksinformationen zwischen den Software-Applikationen verschiedener Hersteller. Mit den Industry Foundation Classes (IFC) wurde von der internationalen Organisation buildingSMART ein standardisiertes Datenmodell geschaffen, das diesen Anforderungen gerecht wird und heute von einer Vielzahl von BIM-Applikationen unterstützt wird (Borrmann et al. 2015). Das Datenformat wurde als ISO 16739 normiert und wird über die europäische Normung auch Eingang in die DIN-Standardisierung finden. Werden herstellernerneutrale, offene Datenformate zur Abwicklung von BIM-Projekten eingesetzt, spricht man von „Open BIM“.

Bis zur aktuellen Version 4 unterstützen die IFC ausschließlich die Modellierung von Gebäuden. Infolge der stark gestiegenen Bedeutung von „BIM for Infrastructure“ in der ganzen Welt sind jedoch für das nächste große Release IFC 5 umfassende Erweiterungen geplant, die die Beschreibung von Bauwerken des Infrastrukturbaus wie Straßen, Brücken und Tunnel ermöglichen.

Als erster Schritt zur Entwicklung von IFC-Infrastructure wurde von April 2014 bis März 2015 das IFC-Alignment-Projekt durchgeführt, dessen Ziel die Entwicklung eines Datenmodells zur Beschreibung einer Trassierung war. Am Projekt haben Forscher der TU München und des französischen CSTB sowie der Firmen AEC3 Deutschland und Bentley Systems mitgewirkt. Das Projekt wurde von den staatlichen Organisationen Trafikverket in Schweden und Rijkswaterstaat in den Niederlanden finanziert. Zur Absicherung der weltweiten Akzeptanz des zukünftigen Standards wurde ein international besetztes Experten-Panel eingerichtet, das in einer Reihe von Workshops die Ausarbeitung des Datenmodells begleitet hat. Im Februar 2015 hat das Datenmodell den Status eines „Candidate Standards“ erreicht und erfolgreich den öffentlichen Review-Prozess durchlaufen. Nach der formalen Annahme durch das buildingSMART Standards Committee wird das Datenmodell in den Rang eines offiziellen Standards (Final Standard) gehoben.

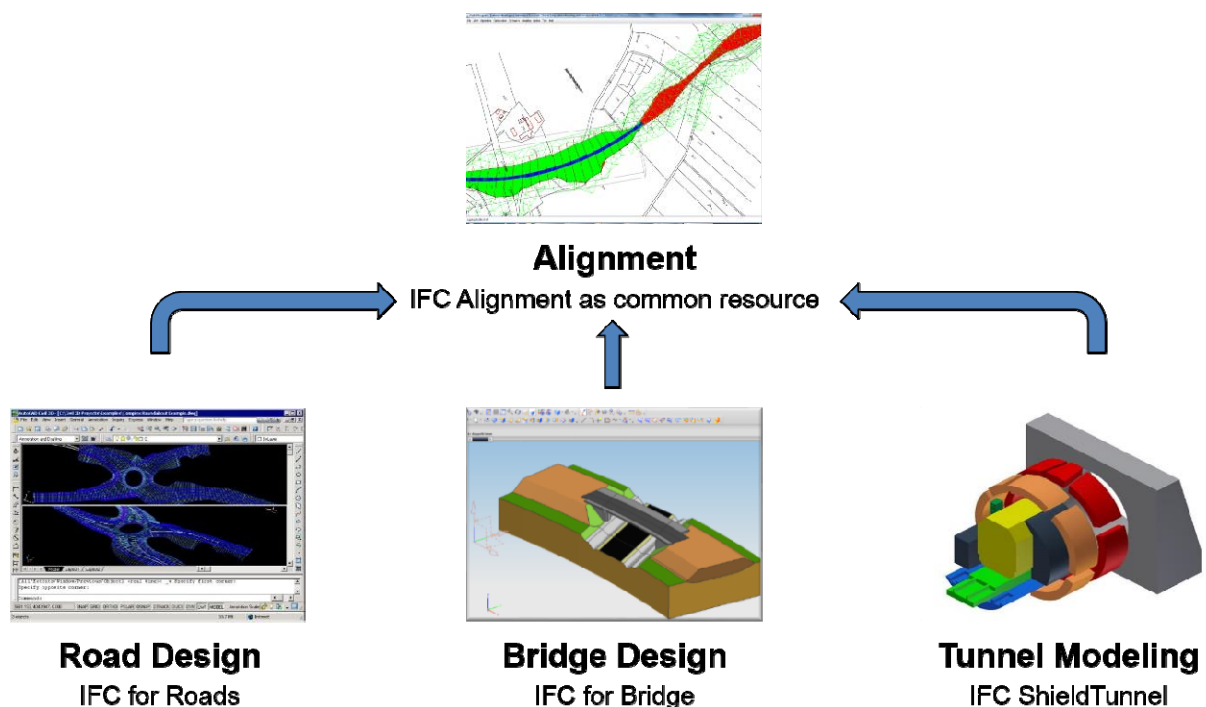


Abbildung 1: Das Datenmodell IFC Alignment beschreibt die Trassierung und bildet die Grundlage für die zukünftigen Standards IFC-Road, IFC-Bridge und IFC-Tunnel.

Das IFC-Alignment-Datenmodell wird die Grundlage für die Datenmodelle IFC-Road, IFC-Bridge und IFC-Tunnel bilden (Abbildung 1). Momentan ist buildingSMART noch auf der Suche nach Geldgebern zur Finanzierung dieser wichtigen Standardisierungsarbeiten, die in Kürze beginnen sollen.

Um Kompatibilität zwischen dem in Deutschland weit verbreiteten und gut etablierten Datenformat-Standard OKSTRA und dem zukünftigen internationalen Standard IFC-Alignment herzustellen, wurde der Lehrstuhl für Computergestützte Modellierung und Simulation der TU München von der Bundesanstalt für Straßenwesen (BASt) beauftragt, ein Verfahren zur Konvertierung zwischen diesen beiden Formaten zu entwickeln. Der vorliegende Beitrag stellt die Datenformate im Detail vor, geht auf Unterschiede und Besonderheiten ein und beschreibt die Herangehensweise bei der Entwicklung der Konvertierungsfunktionalitäten.

Das Datenmodell IFC-Alignment

IFC-Alignment dient zur Beschreibung einer Straßen- bzw. Bahntrassierung. Das Modell basiert auf der etablierten Herangehensweise, die einen Entwurf in Lageplan und Höhenplan (Gradiente) vorsieht. Das zugrundeliegende konzeptionelle Modell (Amann et al. 2014) wurde in Zusammenarbeit mit dem OpenGIS Consortium (OGC) entwickelt. Basierend auf diesem Modell wurde IFC-Alignment entwickelt. Im Rahmen von OGC soll zukünftig ein Standard namens InfraGML entwickelt werden, der auf der Geography Markup Language (GML) beruht. Dabei dient ebenfalls das von buildingSMART und OGC gemeinsam entwickelte konzeptionelle Modell als Grundlage. Der Entwurf von OGC sieht allerdings neben der reinen Trassierung auch noch weitere Anwendungsfälle vor, beispielsweise die Landvermessung oder Liegenschaftsverwaltung, und wird im Unterschied zu IFC-Alignment für die Abbildung des Bestands in GIS-System verwendet werden. Durch die Zusammenarbeit von BuildingSMART und OGC, die das gemeinsame konzeptionelle Modell auf Basis von UML entwickelt haben, sollen die Harmonisierung zwischen beiden Standardisierungsvorhaben gesichert und eine vielversprechende Verbindung zwischen der BIM- und GIS-Welt geschaffen werden.

Im Folgenden soll das Datenmodell IFC-Alignment genauer betrachtet werden. Abbildung 2 zeigt einen Überblick über die neu eingeführten Entitäten in das IFC4-Schema. Als zentrales Element wurde die Klasse *IfcAlignment* eingeführt. Diese referenziert den Lageplan (*IfcAlignment2DHorizontal*) und den Höhenplan (*IfcAlignment2DVertical*). Der Lageplan selbst besteht aus den bekannten Trassierungselementen Gerade (*IfcLineSegment2D*), Kreisbogen (*IfcCircularArcSegment2D*) und Klothoide (*IfcClothoidalArcSegment2D*). Neben der Klothoide gibt es derzeit keine weiteren Übergangskurven im Standard, jedoch werden in den geplanten Erweiterungen IFC-Road bzw. IFC-Rail zukünftig weitere Übergangskurven wie z.B. der Blossbogen definiert werden. Gerade, Kreisbogen und Übergangskurve besitzen eine Untermenge von gleichen Datenattributen. Daher wurde eine gemeinsame Basisklasse namens *IfcCurveSegment2D* eingeführt, die gemeinsame Datenattribute wie Startposition/-richtung und Segmentlänge beinhaltet. Die Klasse *IfcAlignment2DHorizontal* selbst besteht aus einer geordneten Liste von *IfcAlignment2DHorizontalSegment*-Elementen, die jeweils auf ein konkretes Trassierungselement verweisen (Gerade, Kreisbogen oder Klothoide).

Für den Höhenplan wurde ähnlich vorgegangen. Im Höhenplan finden sich Parabeln (*IfcAlignment2DVerSegCircularArc*) und Kreisbögen (*IfcAlignment2DVerSegParabolicArc*) für Ausrundungen und daneben auch Geraden (*IfcAlignment2DVerSegLine*). Die drei genannten Elemente sind wiederum von der Klasse *IfcAlignment2DVerticalSegment* abgeleitet, da es auch hier gemeinsame Eigenschaften wie beispielsweise die Startgradienten eines Elements gibt. Eine geordnete Liste der genannten Trassierungselemente im Höhenplan wird durch die

Klasse *IfcAlignment2DVertical* erzielt. Diese wird wiederum von der Klasse *IfcAlignment* referenziert.

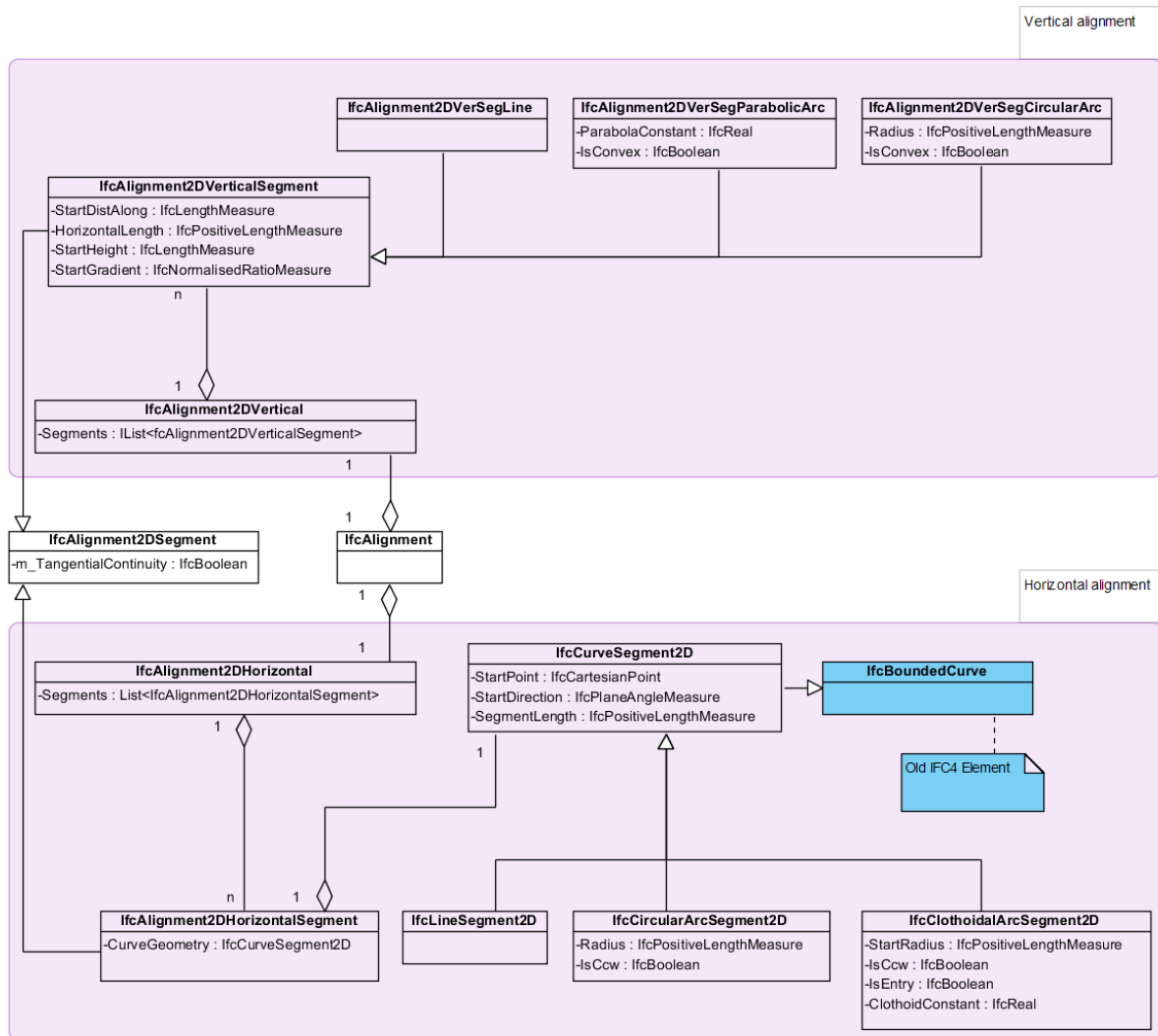


Abbildung 2: UML-Klassendiagramm, das die wichtigsten Neuerungen der IFC-Alignment-Erweiterung zeigt

Abbildung 3 zeigt eine Übersicht über die unterschiedlichen Parameter der Trassierungselemente für den Lageplan. Wie bereits beschrieben, besitzen alle Trassierungselemente des Lageplans einen Startpunkt (*StartPoint*), eine Startrichtung (*StartDirection*) und eine Länge (*SegmentLength*). Darüber hinaus besitzt der Kreisbogen einen Radius (*Radius*) und ein Attribut, das die Orientierung des Kreisbogens angibt (*isCcw*). CCW ist dabei die Abkürzung für *counterclockwise*, den englischen Begriff für „gegen den Uhrzeigersinn“. Das Attribut ist entsprechend „*true*“, falls der Kreisbogen gegen den Uhrzeigersinn orientiert ist, andernfalls ist das Attribut „*false*“. Bei der Klothoide gibt es einen Startradius (*StartRadius*), der den Radius der Klothoide am Startpunkt beschreibt. Ist die Krümmung am Startpunkt 0, so ist der Startradius unendlich. In diesem Fall wird für den Startradius kein Wert gespeichert. Das Attribut *isCcw* der Klothoide gibt genauso wie beim Kreisbogen die Orientierung der Klothoide an. Das Attribut *isEntry* definiert, ob die Krümmung von Start- zum Endpunkt der Klothoide ab- oder zunimmt. Falls die Krümmung zunimmt, ist der Wert von *isEntry* auf „*true*“ gesetzt, andernfalls ist er „*false*“. Als letztes Attribut besitzt die Klothoide noch die Klothoidenkonstante (*ClothoidConstant*).

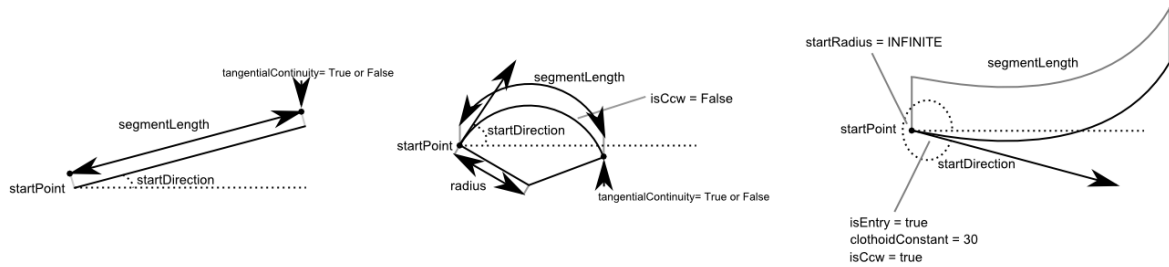


Abbildung 3: Übersicht über die unterschiedlichen Parameter der Trassierungselemente für den Lageplan. Von links nach rechts: *IfcLineSegment2D*, *IfcCircularArcSegment2D* und *IfcClothoidalArcSegment2D*

Eine Trassierung (*IfcAlignment*) ist immer lückenfrei. Um einen Lücke zwischen zwei Segmenten beschreiben zu können, müssen zwei getrennte *IfcAlignment*-Elemente angelegt werden, d. h. ein *IfcAlignment*-Segment besteht immer aus einer zusammenhängenden Abfolge von horizontalen oder vertikalen Trassierungselementen (horizontal: Gerade, Kreisbogen, Klothoide; vertikal: Gerade, Parabelbogen, Kreisbogen). Die einzelnen horizontalen bzw. vertikalen Trassierungselemente müssen dabei nicht zwangsläufig tangential kontinuierlich zueinander sein. Abbildung 4 verdeutlicht dies. Die Klasse *IfcAlignment2DSegment* besitzt ein Attribut *TangentialContinuity*, mit dem festgelegt werden kann, ob das folgende Trassierungselement tangential kontinuierlich ist (*TangentialContinuity = true*) oder eben nicht (*TangentialContinuity = false*).

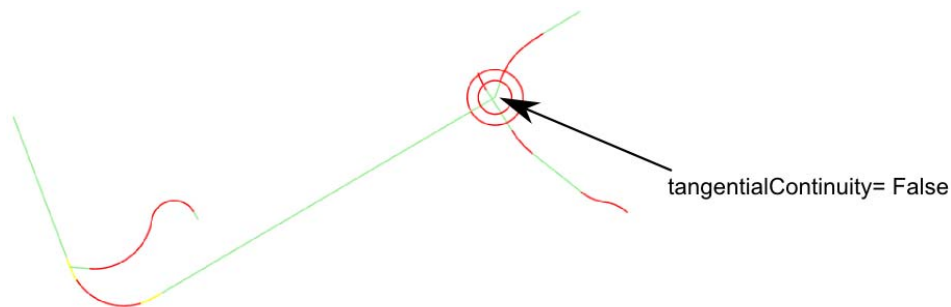


Abbildung 4: Trassierungselemente müssen nicht zwangsläufig tangential kontinuierlich zueinander sein

IfcAlignment2DHorizontalSegment und *IfcAlignment2DVerticalSegment* erben von dieser Klasse und können entsprechend die Eigenschaft des Elements setzen.

Abbildung 5 zeigt ein Beispiel für einen Höhenplan. Die gemeinsame Basisklasse aller vertikalen Trassierungselemente ist die Klasse *IfcAlignment2DVerticalSegment*. Diese besitzt verschiedene Attribute, die für die Beschreibung der einzelnen Segmente des Höhenplans verwendet werden. Das Attribut *StartDistAlong* gibt die Startstationierung des entsprechenden Trassierungselements an. In der Abbildung beginnt hier eine Gerade (*IfcAlignment2D-VerSegLine*) am Punkt AA6 und endet am Punkt AA8. Die Startgradient wird durch das Attribut *StartGradient* beschrieben und entspricht im gezeigten Fall der Steigung der Geraden von Punkt AA6 nach AA8. Das Attribut *StartHeight* gibt die Höhe des Startpunkts des entsprechenden vertikalen Trassierungssegments an. Die horizontale Länge eines vertikalen Trassierungselements wird durch das Attribut *HorizontalLength* beschrieben. Dabei handelt es sich nicht um die reale Länge des Segments, sondern um die horizontale Länge im Höhenplan, was der Länge im Lageplan entspricht.

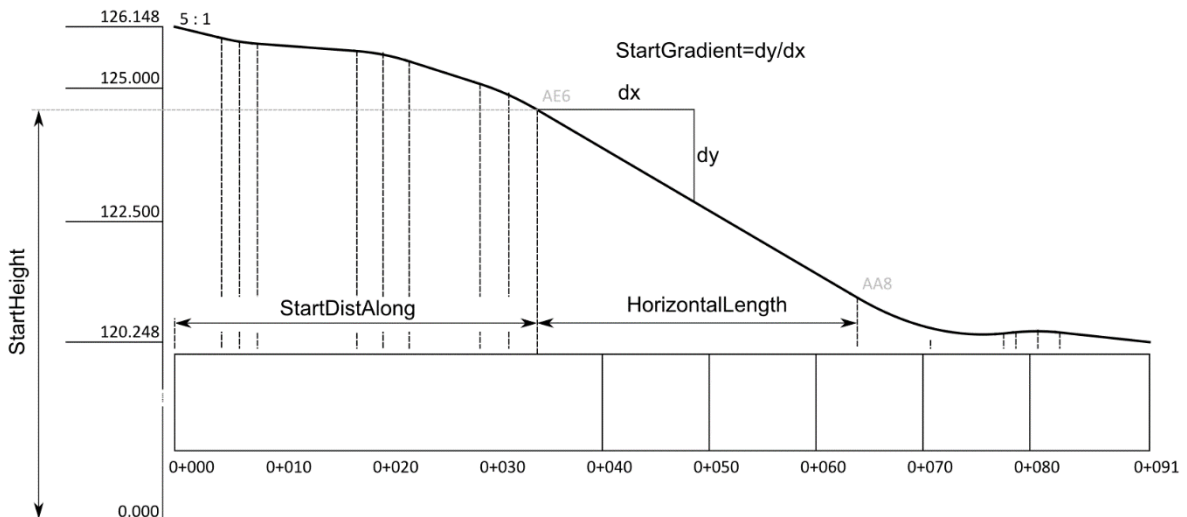


Abbildung 5: Gemeinsame Attribute des Höhenplans

Mehr Details zum IFC-Alignment Datenmodell sind in (Liebich 2014) zu finden.

Das OKSTRA-Datenmodell

Im OKSTRA-Datenmodell wird eine Trasse durch die Klasse *Trasse* beschrieben (siehe Abbildung 6). Die Klasse *Trasse* referenziert dabei wiederum selbst ein Element der Klasse *Achse*. Die Klasse *Achse* verweist auf eine geordnete Liste von Achselementen vom Typ *Achselement*. Diese beschreiben horizontale Trassierungselemente, also Trassierungselemente des Lageplans. Hierbei kann durch einen Achselementtyp der Typ eines Achselements festgelegt werden (z. B. „Gerade“, „Klothoide“ oder „Kreisbogen, tangential“).

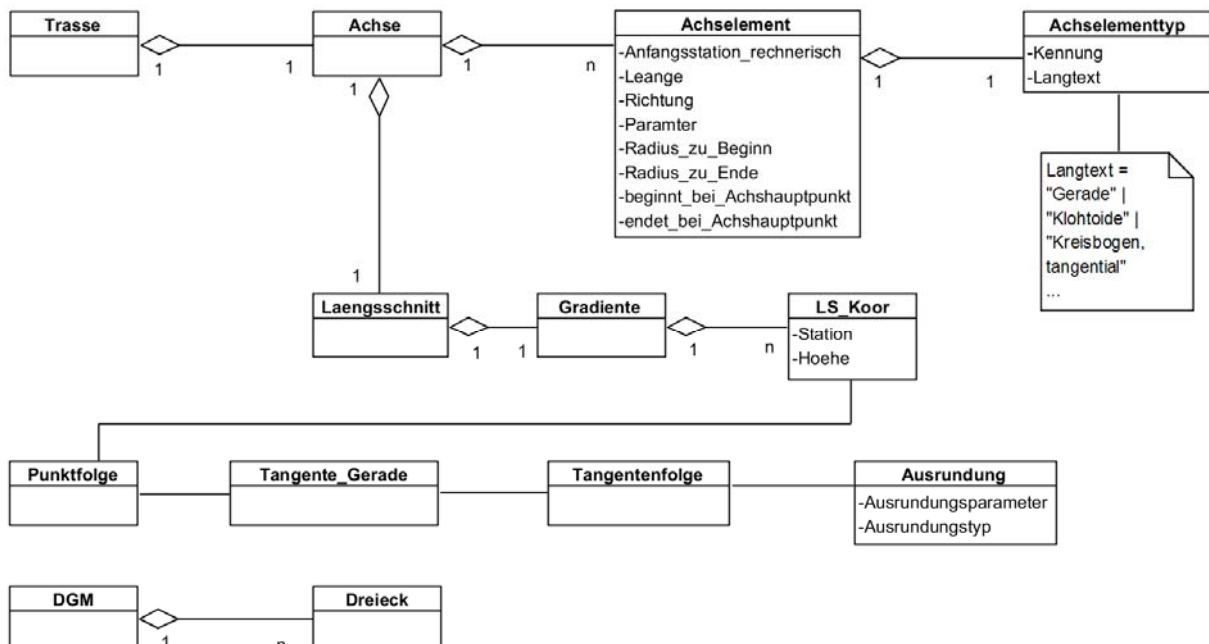
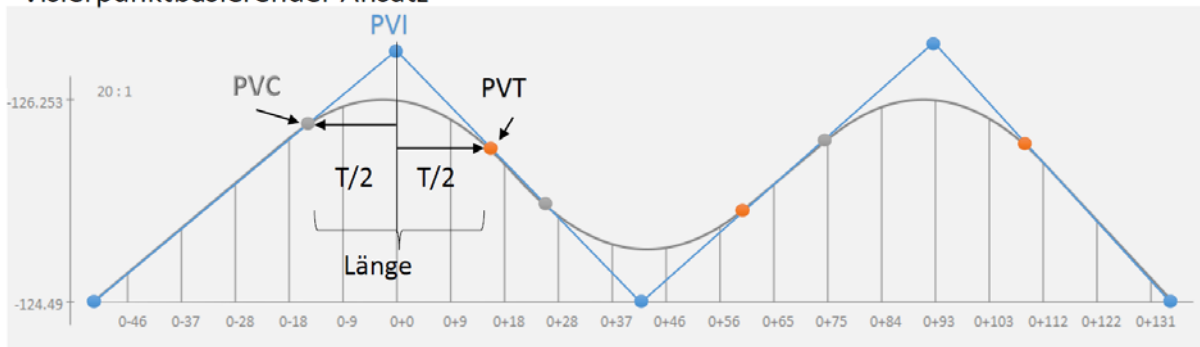


Abbildung 6: OKSTRA-Datenmodell für Trassen

Der Höhenplan wird im OKSTRA-Standard anders als im IFC-Alignment-Standard nicht segmentbasiert, sondern visierpunktbasiert gespeichert. Beim segmentbasierten Ansatz werden die Informationen der einzelnen Segmente gespeichert. Beispielsweise werden für jede Gerade und jede Parabelausrundung der Start- und Endpunkt des entsprechenden

Segments gespeichert. Beim visierpunktbasierten Ansatz werden nur die Visierpunkte und die Ausrundungsradien gespeichert. Die Start- und Endpunkte der einzelnen Segmente müssen hier explizit berechnet werden. Abbildung 7 zeigt den visierpunktbasierten Ansatz im Vergleich zum segmentbasierten Ansatz. Beide Varianten können verlustfrei ineinander umgerechnet werden. Im Rahmen des IFC-Alignment-Projekts hat man sich nach intensiver Diskussion mit der internationalen Community für die segmentbasierte Darstellungsweise entschieden. Diese Variante wurde gewählt, da auch der Lageplan innerhalb des IFC Alignment-Standards bereits segmentbasiert beschrieben wird.

Visierpunktbasierender Ansatz



Segmentbasierender Ansatz



Abbildung 7: Visierpunkt- und segmentbasierender Ansatz im Vergleich

Der Höhenplan ist in der Klasse *Laengsschnitt* hinterlegt. Dieser wird von dem Objekt *Achse* referenziert. Der Längsschnitt verweist wiederum auf eine *Gradiente* (vom Typ *Gradiente*). Die *Gradiente* besteht aus Visierpunkten mit gegebenenfalls einem Ausrundungsradius und Ausrundungstyp. Um an die entsprechenden Attribute zu gelangen, muss durch eine Reihe von Assoziationen navigiert werden (*Punktfolge*, *Tangent_Gerade*, *Tangentfolge*).

Ein Digitales-Gelände-Modell (DGM) kann ebenfalls in Form einer Dreiecksbeschreibung abgelegt werden. Dazu dient die Klasse *DGM*, die eine Liste von Dreiecken vom Datentyp *Dreieck* speichern kann.

Konvertierung

Die Umwandlung des Höhenplans zwischen dem IFC Alignment-Standard und dem OKSTRA-Standard ist unkompliziert, da beide segmentbasierend sind und relativ ähnlich aufgebaut sind. Beispielsweise lässt sich über die Attribute *beginnt_bei_Achshauptpunkt* der Startpunkt für das IFC-Alignment ermitteln. Das Mapping zwischen den verschiedenen Attributen der beiden Standards ist einfach. Die folgende Tabelle (Tabelle 1) zeigt exemplarisch einige Abbildungen:

Tabelle 1: Konvertierung des Höhenplans von IFC Alignment nach OKSTRA

IFC Alignment	OKSTRA
IfcLineSegment2D.StartPoint	Achselement.beginnt_bei_Achshauptpunkt
IfcLineSegment2D.StartDirection	Achselement.Richtung
IfcLineSegment2D.SegmentLength	= distance(Achselement.beginnt_bei_Achshauptpunkt Achselement.endet_bei_Achshauptpunkt) oder Achselement.Laenge
IfcCircularArcSegment2D.Radius	Achselement.Radius_zu_Beginn
IfcClothoidalArcSegment2D.StartRadius	Achselement.Radius_zu_Beginn
IfcClothoidalArcSegment2D.IsEntry	1/Achselement.Radius_am_Beginn < 1/Achselement.Radius_am_Ende
IfcClothoidalArcSegment2D.ClothoidalConstant	Achselement.Parameter

Komplizierter als die Umwandlung des Höhenplans gestaltet sich die Umwandlung des vertikalen Alignments. Der Algorithmus dazu ist in Tabelle 2 in Pseudocode formuliert.

Tabelle 2: Algorithmus zur Umwandlung des Höhenplans von OKSTRA nach IFC Alignment

<ol style="list-style-type: none"> 1. Alle „LS_Koor“ der Gradiente besuchen <ol style="list-style-type: none"> a. Station und Höhe des Visierpunkts ermitteln (Attribute Station und Hoehe auslesen) b. Besuche Attribut „folgt_auf_LS_Koor“ vom Typ „Punktfolge“ falls vorhanden <ol style="list-style-type: none"> i. Besuche diese Punktfolge (falls vorhanden) und lese Attribut „hat_Tangente_Gerade“ vom Typ „Tangente_Gerade“ ii. Besuche diese Tangente und lese das Attribut „folgt_auf_Tangente_Gerade“ vom Typ „Tangentenfolge“ (falls vorhanden) iii. Besuche diese Tangentenfolge und lese das Attribut „hat_Ausrundung“ vom Typ „Ausrundung“ (falls vorhanden) iv. Besuche diese Ausrundung und lese das Attribut „Ausrundungsparameter“ und „Ausrundungstyp“ (falls vorhanden) 2. Berechne Anfangs- und Endsteigung sowie Paramater T/2 3. Berechne Anfangs- und Endpunkt der vertikalen Trassierungselemente

Der Algorithmus besucht zunächst alle Visierpunkte und sammelt alle verfügbaren Informationen ein. Dazu gehören die Position (Stationierung, Höhe) und gegebenenfalls ein Ausrundungsparameter. Im zweiten Schritt werden die Anfangs- und Endsteigung sowie der Paramater T/2 (siehe Abbildung 7) berechnet. Im letzten Schritt werden die Anfangs- und Endpunkte der vertikalen Trassierungselemente berechnet und die IFC spezifischen Alignment-Segmente erzeugt.

Das digitale Geländemodell wird von der Geometrieseite auf ein *IfcTriangulatedFaceSet* abgebildet und mit der Semantik *IfcGeographicElement* verknüpft. Die *IfcGeographicElement*-Entität wird genutzt, um Terraindaten auszuzeichnen.

Die TUM Open Infra Platform

Die TUM Open Infra Platform (OIP) wurde vom Lehrstuhl für Computergestützte Modellierung und Simulation der Technische Universität München entwickelt. Dabei handelt es sich um ein Softwareprogramm, das genutzt werden kann, um Trassierungsdaten in verschiedenen

Dateiformaten zu betrachten und zwischen verschiedenen Dateiformaten zu konvertieren. Derzeit werden die Dateiformate IFC-Alignment, OKSTRA und LandXML unterstützt. Abbildung 8 zeigt die möglichen Konvertierungsoptionen innerhalb der TUM Open Infra Platform. Insbesondere kann OIP dazu verwendet werden, OKSTRA-Daten in das IFC-Alignment-Format zu konvertieren. Auch die umgekehrte Richtung ist möglich.

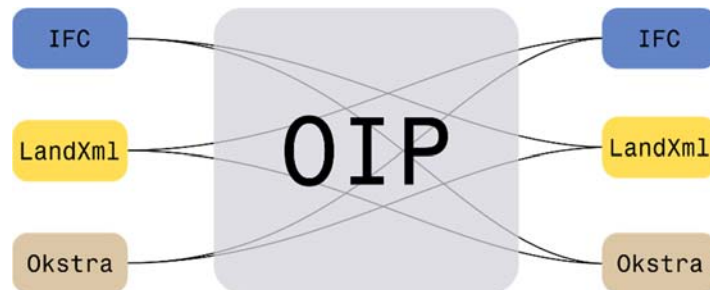


Abbildung 8: Konvertierungsmöglichkeiten innerhalb der TUM Open Infra Platform

Die Software ist kostenlos von der Webseite <https://www.cms.bgu.tum.de/oip> downloadbar. Auf der Webseite findet man ebenfalls Informationen zu den Systemanforderungen, der Installation und eine Dokumentation zur Software. In Abbildung 9 ist ein Screenshot der TUM Open Infra Platform zu sehen. Neben der 3D-Ansicht im oberen Drittel der Anwendung wird auch das Krümmungsband (mittlerer Teil) sowie der Höhenplan visualisiert (unterer Teil).

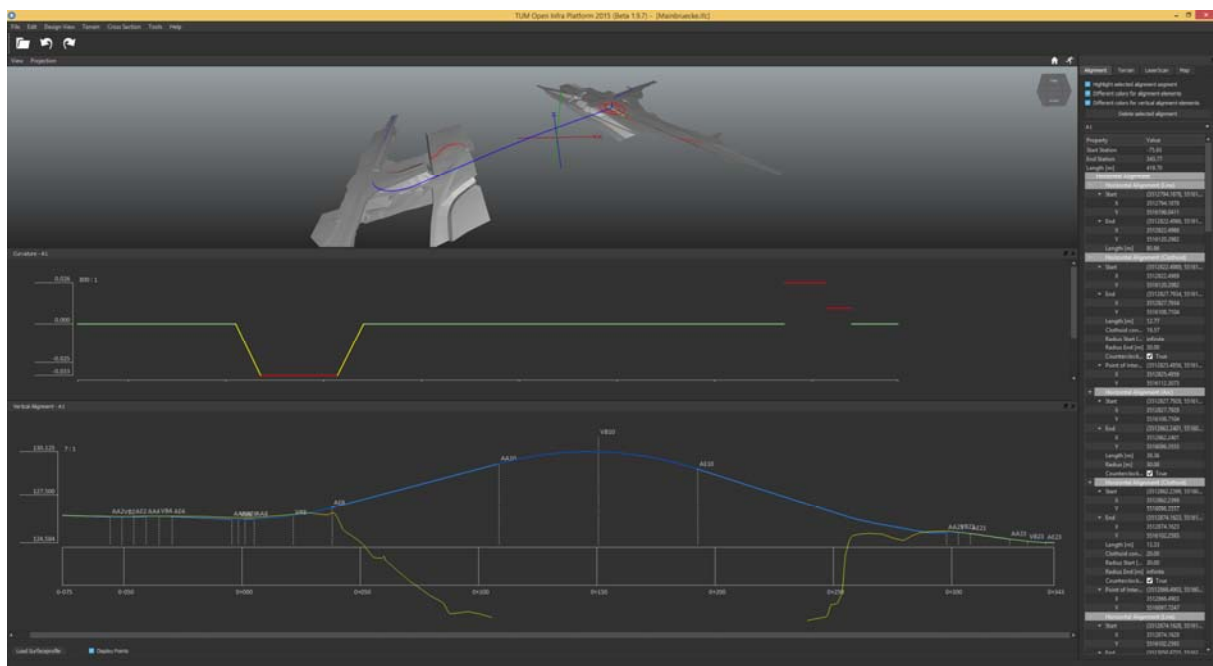


Abbildung 9: Screenshot der TUM Open Infra Platform

OIP bietet auch noch weitere Funktionen. So können beispielsweise Geländedaten im XYZ-Format importiert oder Laser-Scan-Daten im LAS-Format betrachtet werden (siehe Abbildung 10).

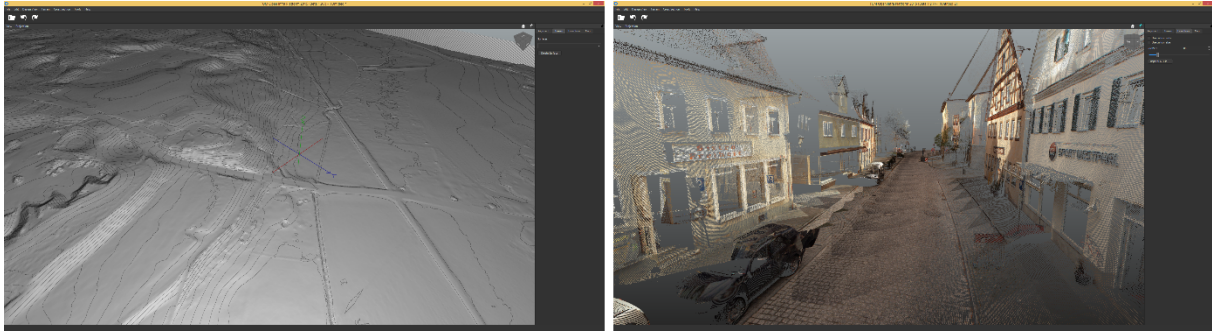


Abbildung 10: Links: XYZ-Import; Rechts: LAS-Import

Ausblick

Mit der IFC-Alignment-Erweiterung wurde der erste Schritt für die Unterstützung von Infrastrukturprojekten innerhalb der IFC-Welt unternommen. Noch fehlen im IFC-Datenmodell Möglichkeiten, Straßenquerschnitte oder Tunnel- bzw. Brückenbauwerke zu beschreiben. Auch für den Gleisbau fehlen noch entsprechende Elemente. Jedoch wurden hierfür bereits zahlreiche Vorschläge gemacht (Yabuki et al. 2007, Yabuki et al. 2008, Lebegue et al 2012, Amann et al. 2013, Amann et al. 2015). Auf Basis dieser Vorschläge sollen in naher Zukunft offizielle buildingSMART-Projekte entstehen, die diese Standardisierungsvorhaben weiter vorantreiben, damit auf internationaler Ebene ein starker Standard zur umfassenden Beschreibung von Infrastrukturbauwerken etabliert wird.

Referenzen

- Amann, J.; Borrmann, A.; Hegemann, F.; Jubierre, J.R.; Flurl, M.; Koch, C.; König, M.: A Refined Product Model for Shield Tunnels Based on a Generalized Approach for Alignment Representation, In: Proc. of the ICCBEI 2013, Tokyo, Japan, 2013
- Amann, J., Borrmann, A., Chipman, T., Lebegue, E., Liebich, T., Scarponcini, P., P6 IFC Alignment – Conceptual Model, <http://www.buildingsmart-tech.org/downloads/ifc/ifc5-extension-projects/ifc-alignment/ifcalignment-conceptualmodel-cs>, 2014, buildingSMART
- Amann, J.; Singer, D.; Borrmann, A.: Extension of the upcoming IFC alignment standard with cross sections for road design, In: Proc. of the ICCBEI 2015, Tokyo, Japan, 2015
- Lebegue, E., Fiés, B. and Gual, J., (2012). IFC-Bridge V3 Data Model – IFC 4, Edition R1
- Liebich, T., IFC Alignment Schema, <http://www.buildingsmart-tech.org/downloads/ifc/ifc5-extension-projects/ifc-alignment/ifcalignment-ifcextension-cs>, 2014, buildingSMART
- Borrmann, A., König, M., Koch, C., Beetz, J. (Hrsg.): Building Information Modeling – Technologische Grundlagen und industrielle Praxis, Springer Vieweg, 2015
- Eastman, C., Teicholz, P., Sacks, R., Liston, K., BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, Wiley Publishing, 2008
- Yabuki, N., Azumaya, Y., Akiyama, M., Kawanai Y., Miya, T. (2007): Fundamental Study on Development of a Shield Tunnel Product Model. Journal of Civil Engineering Information Application Technology 16, pp. 261-268
- Yabuki, N. (2008). Representation of caves in a shield tunnel product modeling. In Proc. of the 7th European Conference on product and Process Modeling. Sophia Antipolis, France.