

Analyse zur Erstellung eines 3D-BIM-konformen Straßenentwurfs auf Basis von IFC-Alignment aus einem OKSTRA konformen 2D-Straßenentwurf unter Einsatz der TUM Open Infra Platform in Verbindung mit der OKSTRA-Klassenbibliothek

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

Im Auftrag der Bundesanstalt für Straßenwesen, Bergisch Gladbach

Vorwort

Der Inhalt dieser Analyse wurde unter dem Titel „Open BIM for Infrastructure – mit OKSTRA und IFC-Alignment zur internationalen Standardisierung des Datenaustauschs“ im Rahmen des 6. OKSTRA – Symposiums (20./21. Mai 2015) in Köln präsentiert.

Einleitung: Building Information Modeling

Building Information Modeling (BIM) steht für die durchgängige Nutzung hochwertiger digitaler Daten eines Bauwerks über dessen gesamten Lebenszyklus hinweg – 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 abgeleitete 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, 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, welche 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. In diesem 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.

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.

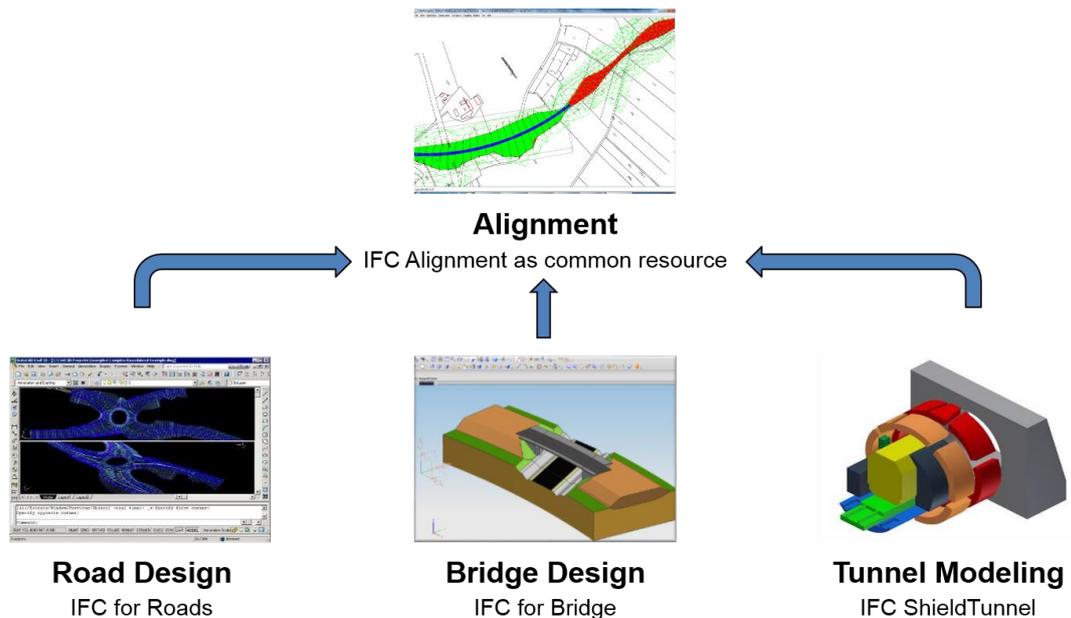


Abbildung 1: Das Datenmodell IFC-Alignment beschreibt die Trassierung und bildet die Grundlage für die zukünftigen Standards IFC-Road, IFC-Bridge und IFC-Tunnel.

Das 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.

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 gemeinsamen 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.

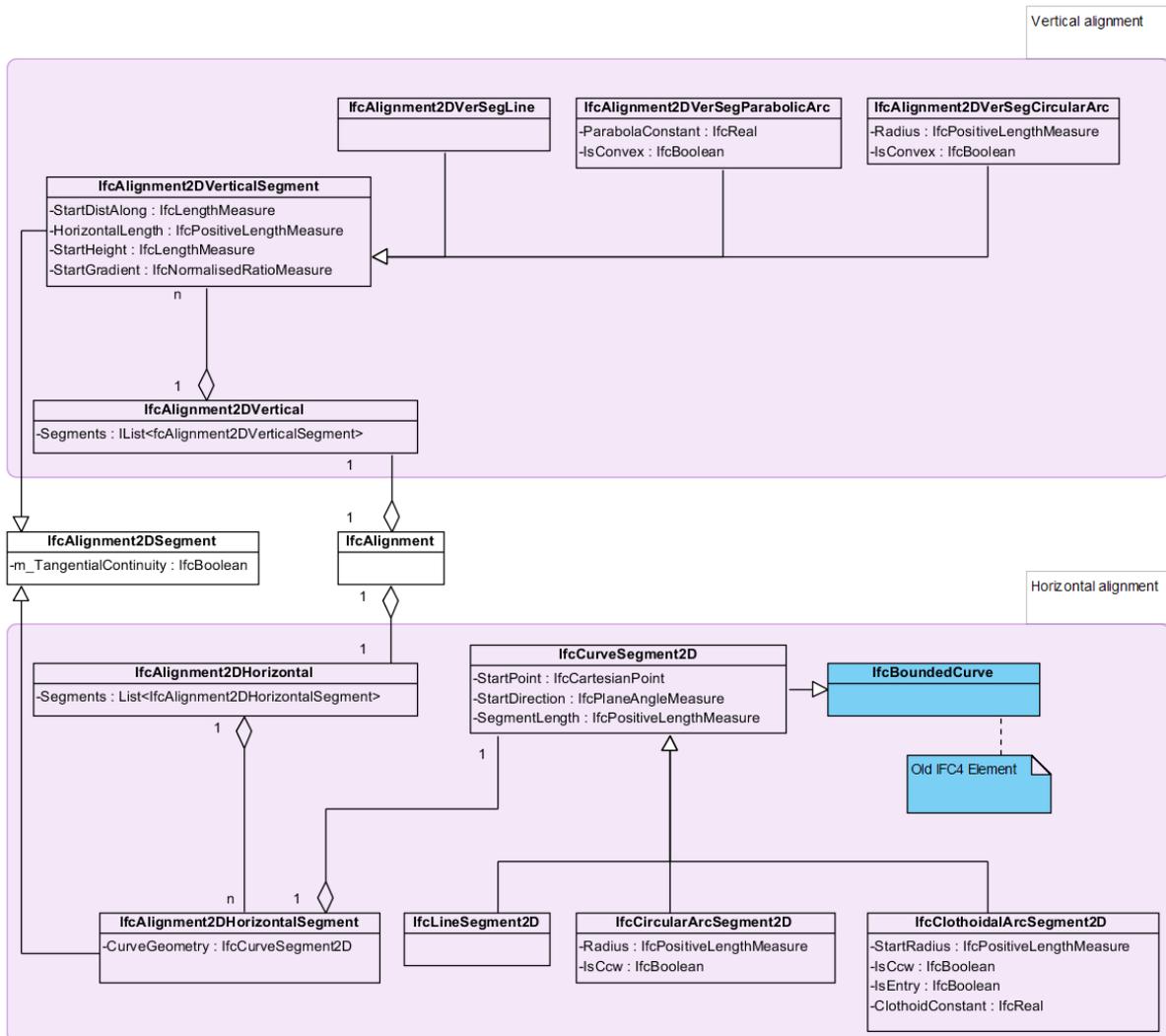


Abbildung 2: UML-Klassendiagramm, das die wichtigsten Neuerungen der IFC-Alignment-Erweiterung zeigt

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*).

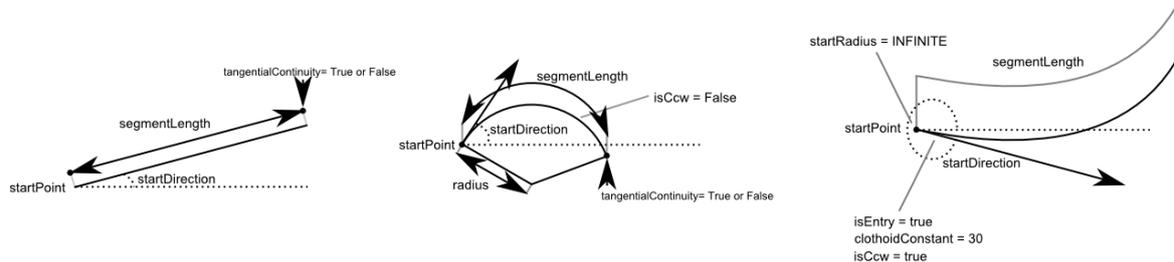


Abbildung 3: Übersicht über die unterschiedlichen Parameter der Trassierungselemente für den Lageplan; von links nach rechts: *IfcLineSegment2D*, *IfcCircularArcSegment2D* und *IfcClothoidalArcSegment2D*

Eine Trassierung (*IfcAlignment*) ist immer lückenfrei. Um eine 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*).

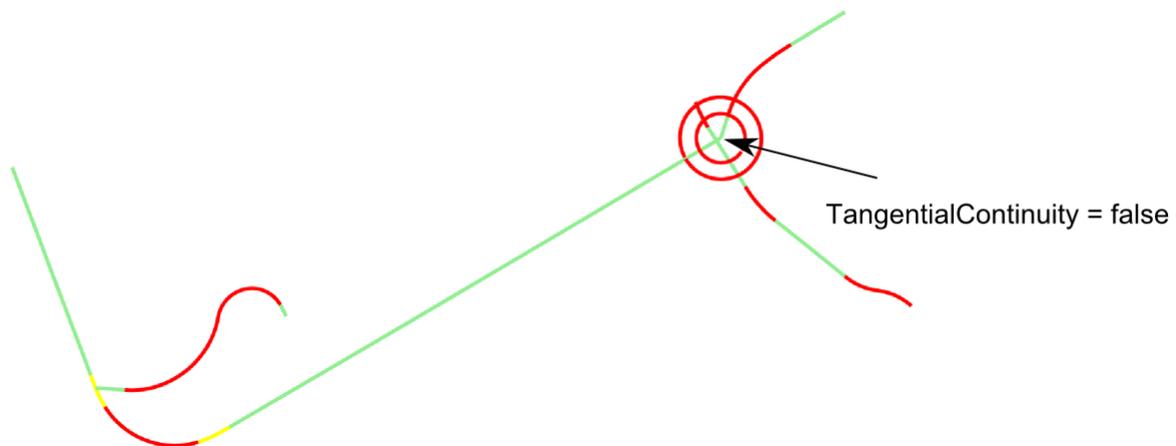


Abbildung 4: Trassierungselemente müssen nicht zwangsläufig tangential kontinuierlich zueinander sein.

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 Startgradienten wird durch das Attribut *StartGradient* beschrieben und entspricht im gezeigten Fall der Steigung der Geraden

von Punkt *AE6* 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.

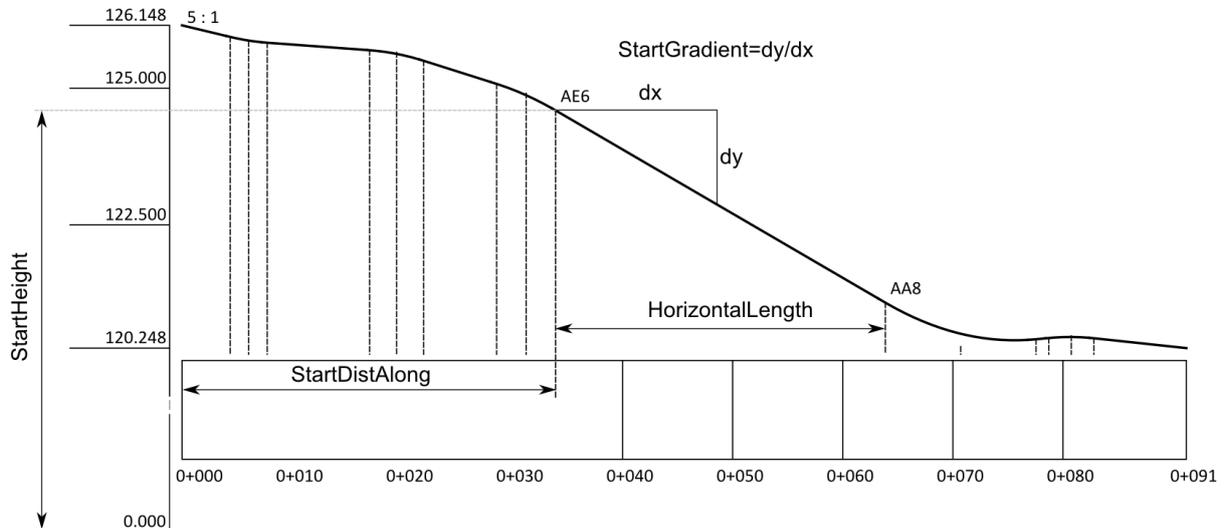


Abbildung 5: Gemeinsame Attribute des Höhenplans

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

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

Das OKSTRA-Datenmodell

Im OKSTRA-Datenmodell wird eine Trasse durch die Klasse *Trasse* beschrieben (siehe Abbildung 6).

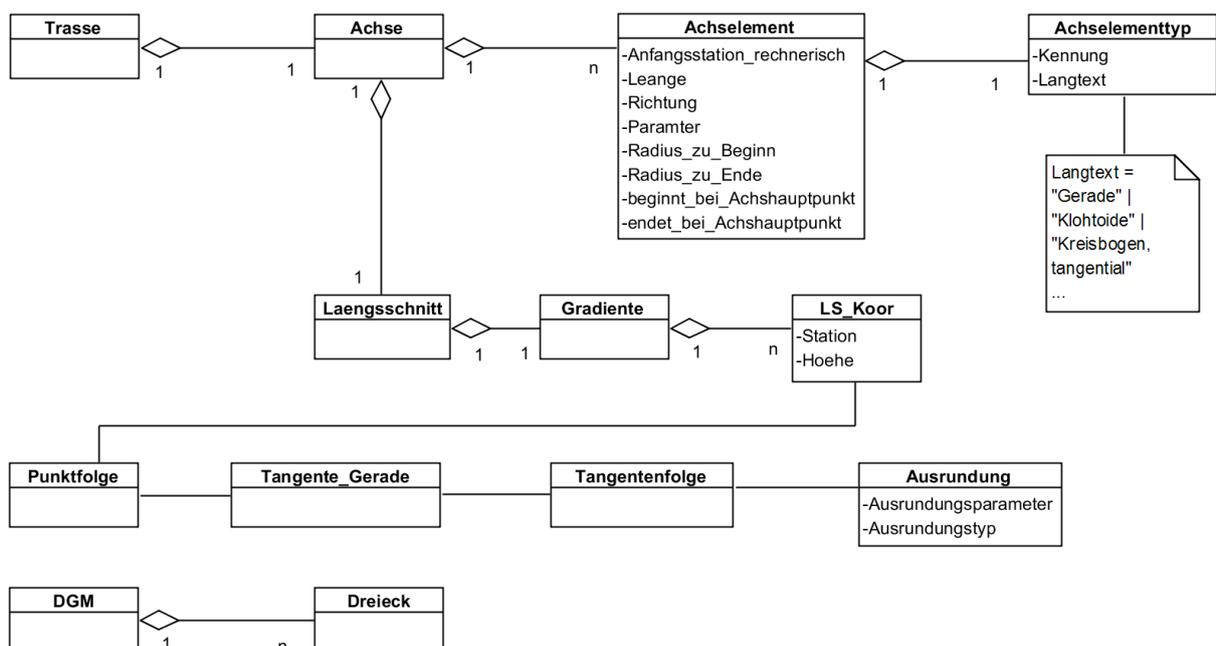
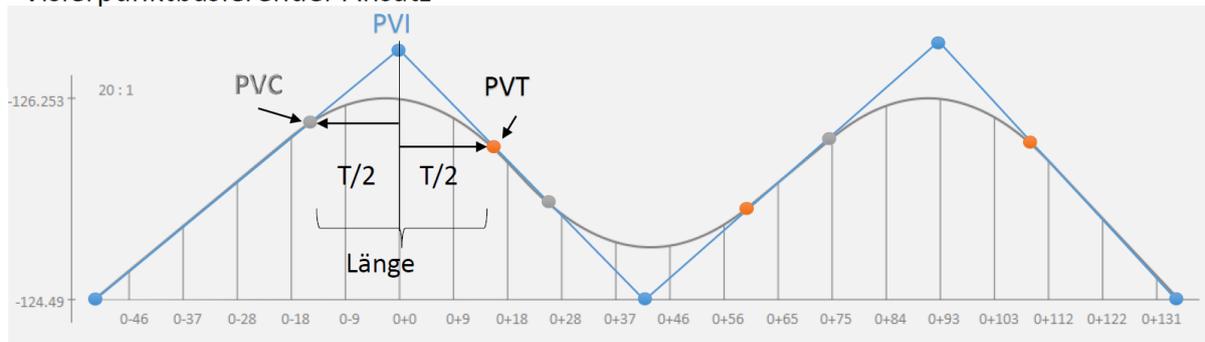


Abbildung 6: OKSTRA-Datenmodell für Trassen

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“).

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

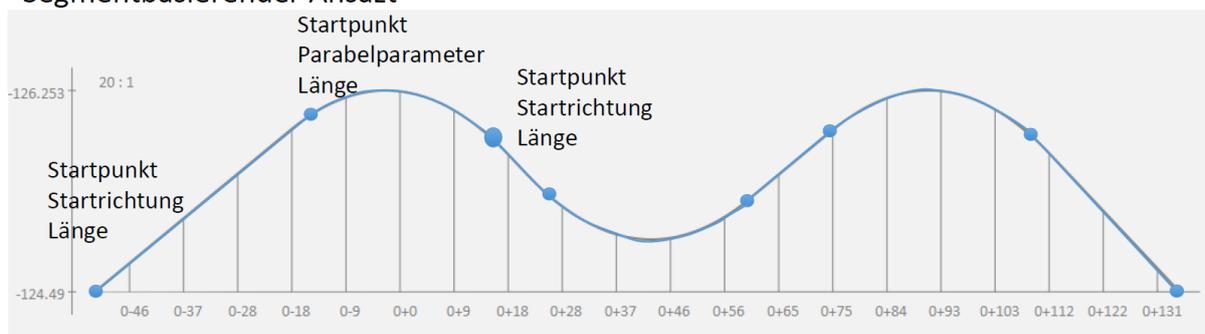


Abbildung 7: Visierpunkt- und segmentbasierender Ansatz im Vergleich

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*, *Tangente*, *Gerade*, *Tangentenfolge*).

Ein *Digitales-Gelände-Modell* (DGM) kann 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 Lageplans zwischen dem IFC-Alignment-Standard und dem OKSTRA-Standard ist unkompliziert, da beide segmentbasierend 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 Lageplans 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_End
IfcClothoidalArcSegment2D.ClothoidalConstant	Achselement.Parameter

Die Umwandlung des Höhenplans gestaltet sich etwas komplizierter. Der Algorithmus dazu ist in Tabelle 2 in Pseudocode formuliert (dieser ist für OKSTRA-Version 1.014 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 lies Attribut „hat_Tangente_Gerade“ vom Typ „Tangente_Gerade“ ii. Besuche diese Tangente und lies das Attribut „folgt_auf_Tangente_Gerade“ vom Typ „Tangentenfolge“ (falls vorhanden) iii. Besuche diese Tangentenfolge und lies das Attribut „hat_Ausrundung“ vom Typ „Ausrundung“ (falls vorhanden) iv. Besuche diese Ausrundung und lies das Attribut „Ausrundungsparameter“ und „Ausrundungstyp“ (falls vorhanden) 2. Berechne Anfangs- und Endsteigung sowie Parameter 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 Parameter 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.

In der OKSTRA Version 2.016 gab es im Höhenplan einige Veränderungen im Vergleich zur Version 1.014. Insbesondere wurde das Laengsschnitt-Objekt entfernt und LS_Koor wurde durch das Objekt Grad_Koors ersetzt. Durch die Änderungen gestaltet sich der Zugriff auf die

Parabelparameter einfacher. Tabelle 3 zeigt exemplarisch die Abbildungen einiger IFC-Alignment-Elemente des Höhenplans auf das OKSTRA (2.016) Produktdatenmodell.

Tabelle 3: Konvertierung des Höhenplans von IFC-Alignment nach OKSTRA 2.016

IFC-Alignment	OKSTRA
IfcAlignment2DVerticalSegment .StartDistAlong	Gradiente.hat_Grad_Koor.Station
IfcAlignment2DVerticalSegment .HorizontalLength	Gradiente.hat_Grad_Koor.Station[index+1] - Gradiente.hat_Grad_Koor.Station[index]
IfcAlignment2DVerticalSegment .StartHeight	Gradiente.hat_Grad_Koor.Hoehe
IfcAlignment2DVerticalSegment .StartGradient	$\frac{\text{Gradiente.hat}_{\text{GradKoor}}.\text{Hoehe}[\text{index} + 1] - \text{Gradiente.hat}_{\text{GradKoor}}.\text{Hoehe}[\text{index}]}{\text{Gradiente.hat}_{\text{GradKoor}}.\text{Station}[\text{index} + 1] - \text{Gradiente.hat}_{\text{GradKoor}}.\text{Station}[\text{index}]}$
IfcAlignment2DVerSegParabolicArc .ParabolaConstant	= abs(FocalLength * 2) FocalLength = Abstand zwischen Brenn- und Scheitelpunkt = 1.0 / (4.0 * a) mit a = 0.5 * ((SteigungAmEnde – steigungAmAnfang) / (endStation - anfangsStation)) Anfangs- und Endstation lassen sich über Gradiente.hat_Grad_Koor.Station ermitteln. Steigung am Anfang: siehe IfcAlignment2DVerticalSegment .StartGradient Steigung am Ende: Analog zu IfcAlignment2DVerticalSegment .StartGradient
IfcAlignment2DVerSegParabolicArc .IsConvex	Wenn FocalLength * 2.0 < 0 dann ist IsConvex = false andernfalls ist es true.
IfcAlignment2DVerSegCircularArc .Radius	Kreisbögen werden im Höhenplan von OKSTRA nicht unterstützt
IfcAlignment2DVerSegCircularArc .IsConvex	Kreisbögen werden im Höhenplan von OKSTRA nicht unterstützt

Neben der 2D-basierten Variante, bestehend aus Lage- und Höhenplan, gibt es auch die Möglichkeit, im Rahmen von IFC-Alignment Achsen mithilfe von 3D-Punkten zu beschreiben. Die präferierte Variante ist aber, die 2D-Variante zu verwenden. Außerdem wird die 3D-Variante nicht von OKSTRA unterstützt.

Die Entscheidung für eine primäre 2D-Repräsentation im IFC-Alignment-Standard wurde getroffen, da so die originären ingenieurtechnische Parameter des Trassenentwurfs (wie Krümmungsradius, Steigung etc.) direkt im Datenmodell abgelegt werden können. Damit bleiben diese Informationen direkt zugänglich, prüfbar und leichter manipulierbar. Aus den 2D-Informationen lässt sich immer eine 3D-Trassierung berechnen, umgedreht wäre dies aber nur sehr eingeschränkt möglich. Der IFC-Alignment-Entwurf sieht vor, dass zusätzlich eine 3D-Trassierung hinterlegt werden kann. Damit werden Viewer ohne Trassierungsverarbeitung in die Lage versetzt, die Trasse korrekt anzuzeigen. Einschränkend ist jedoch anzumerken, dass bei redundanter Speicherung von 2D- und 3D-Informationen leicht Inkonsistenzen auftreten können. IFC-Alignment bildet u.a. die Grundlage für IFC-Bridge, IFC-Tunnel usw. Die dort definierten Bauteile werden auf jeden Fall primär als 3D-Geometrie modelliert werden.

Querprofile bzw. ein Deckenbuch werden nicht vom IFC-Alignment erfasst. Der Fokus von IFC-Alignment liegt auf der Beschreibung der Trassierung/Achse bestehend aus Höhen- und Lageplan. Allerdings wurden hierzu schon einige Vorschläge ausgearbeitet wie z. B. in (Amann et al. 2015 oder (Singer et al. 2014). Der Vorschlag zur Erweiterung von IFC, der in (Amann et al. 2015) beschreiben wird, ist in Abbildung 8 dargestellt.

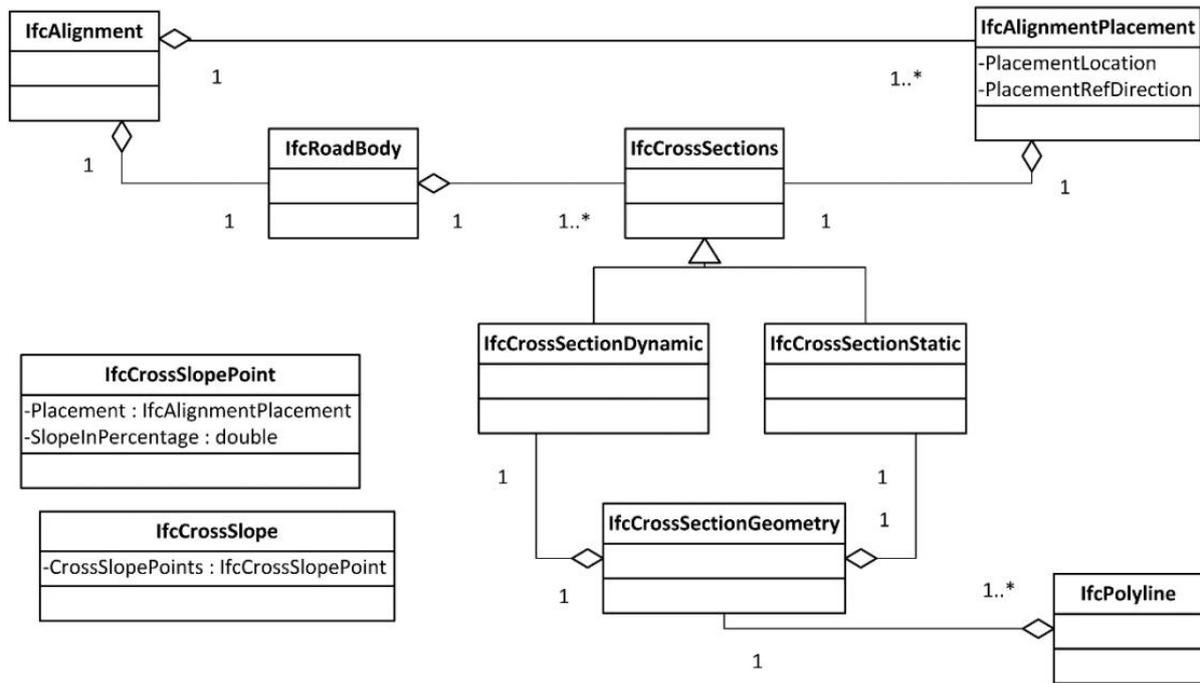


Abbildung 8: Vorschlag zur Erweiterung von IFC-Alignment um Querschnitte

Nähere Informationen dazu sind den Veröffentlichungen (Amann et al. 2015) und (Singer et al. 2014) zu entnehmen und sollen an dieser Stelle nicht wiederholt werden.

Das Deckenbuch bzw. Querprofile für Straßen werden von buildingSMART im Rahmen des IFC-Road Projektes umgesetzt werden. Im Rahmen des INFRA Room von buildingSMART wird derzeit auch über einen sogenannten Line-String-Ansatz (siehe dazu auch (Inframodel.fi 2014)) diskutiert, der im finnischen Straßen-Standard verwendet wird. Welcher Ansatz sich am Ende durchsetzen wird oder ob es einen hybriden Ansatz geben wird, ist derzeit noch nicht klar.

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 9 zeigt die möglichen Konvertierungsoptionen innerhalb der TUM Open Infra Platform. Insbesondere kann die OIP dazu verwendet werden, OKSTRA-Daten in das IFC-Alignment-Format zu konvertieren. Auch die umgekehrte Richtung ist möglich.

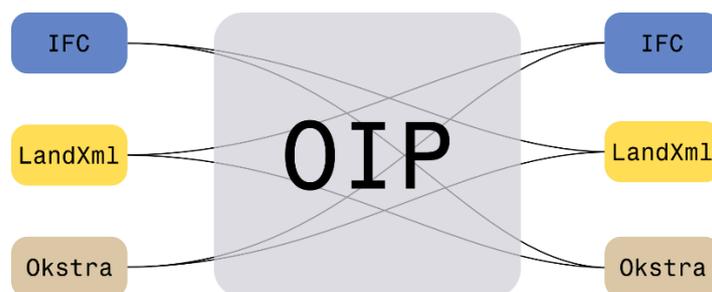


Abbildung 9: Konvertierungsmöglichkeiten innerhalb der TUM Open Infra Platform

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 10 ist ein Screenshot der TUM Open Infra Platform zu sehen. Neben der 3D-Ansicht im oberen Drittel der Anwendung werden auch das Krümmungsband (mittlerer Teil) sowie der Höhenplan visualisiert (unterer Teil).

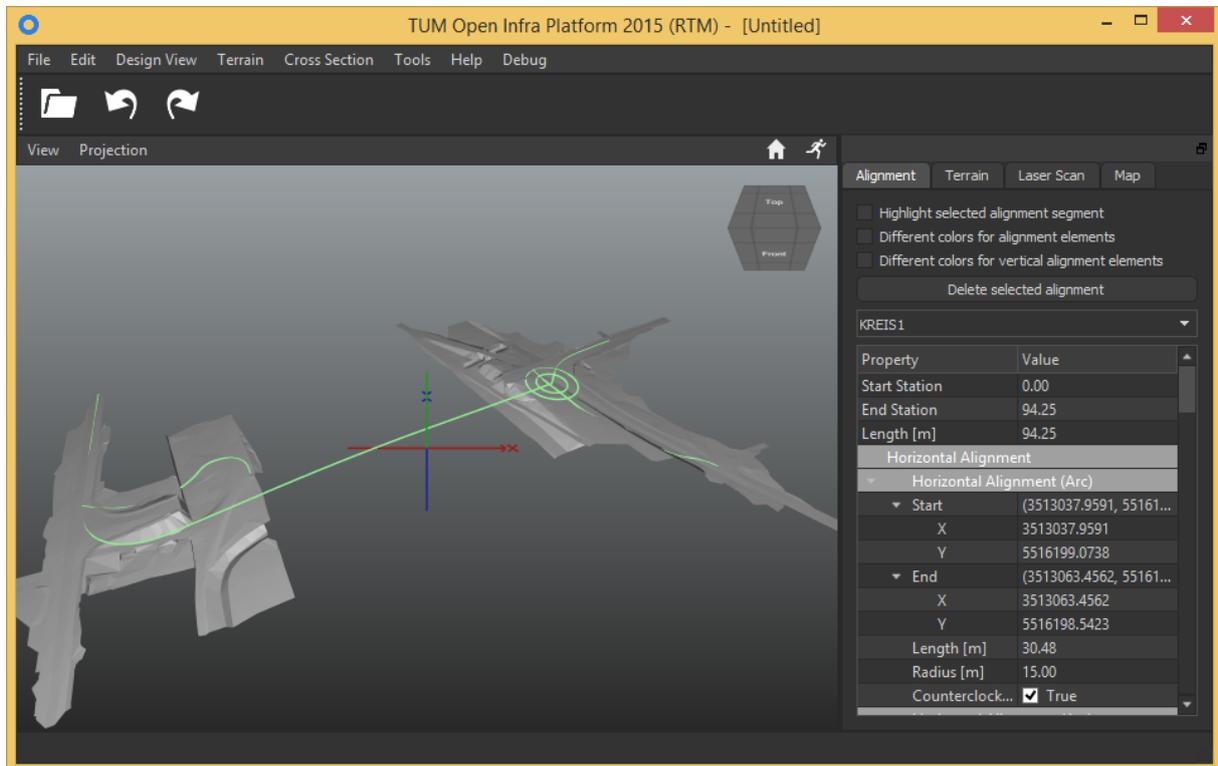


Abbildung 10: Screenshot der TUM Open Infra Platform

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 11).

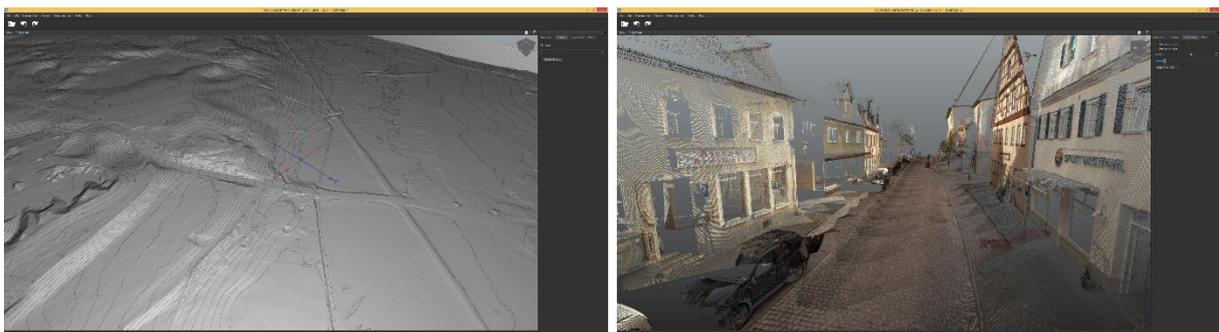


Abbildung 11: Links: XYZ-Import; rechts: LAS-Import

Durchgeführte Tests

Im Rahmen unserer Softwaretests wurden 50 Testdateien im IFC-Alignment-Dateiformat erzeugt. Diese wurden mithilfe der TUM Open Infra Platform in das OKSTRA-Dateiformat konvertiert und es wurde ein visueller Abgleich durchgeführt. Entsprechend wurden auch OKSTRA-Dateien in das IFC-Alignment-Format überführt und visuelle Vergleiche erstellt. Abbildung 12 zeigt einen Ausschnitt der rein visuellen Tests. In der ersten Spalte sind jeweils die IFC-Alignment-Dateien zu sehen und in der zweiten Spalte befinden sich die entsprechend

OKSTRA-Dateien. Die generierten OKSTRA-Dateien wurden außerdem mithilfe des OKSTRA-Werkzeugs (Version 1.3.0.25) validiert.

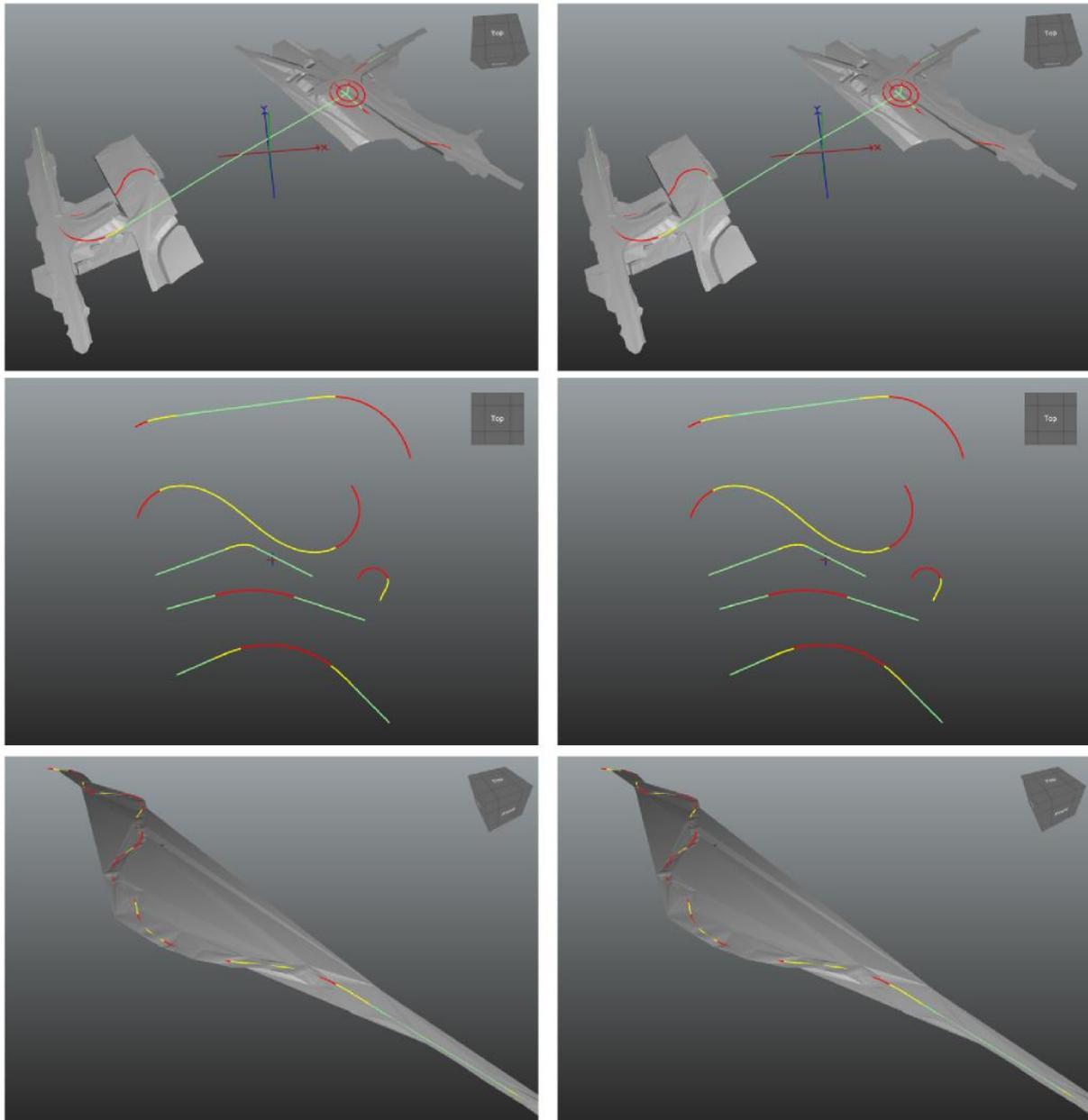


Abbildung 12: Visueller Vergleich zwischen IFC-Alignment und OKSTRA

Die TUM Open Infra Platform verwendet Fließkommazahlen einfacher (Float32) und doppelter Genauigkeit (Float64) nach IEEE 754. Diese besitzen nur eine begrenzte Genauigkeit, wie das Code-Snippet 1 zeigt.

```
#include <iostream>

int main() {
    if (0.362 * 100.0 != 36.2)
        std::cout << "verschieden" << std::endl;

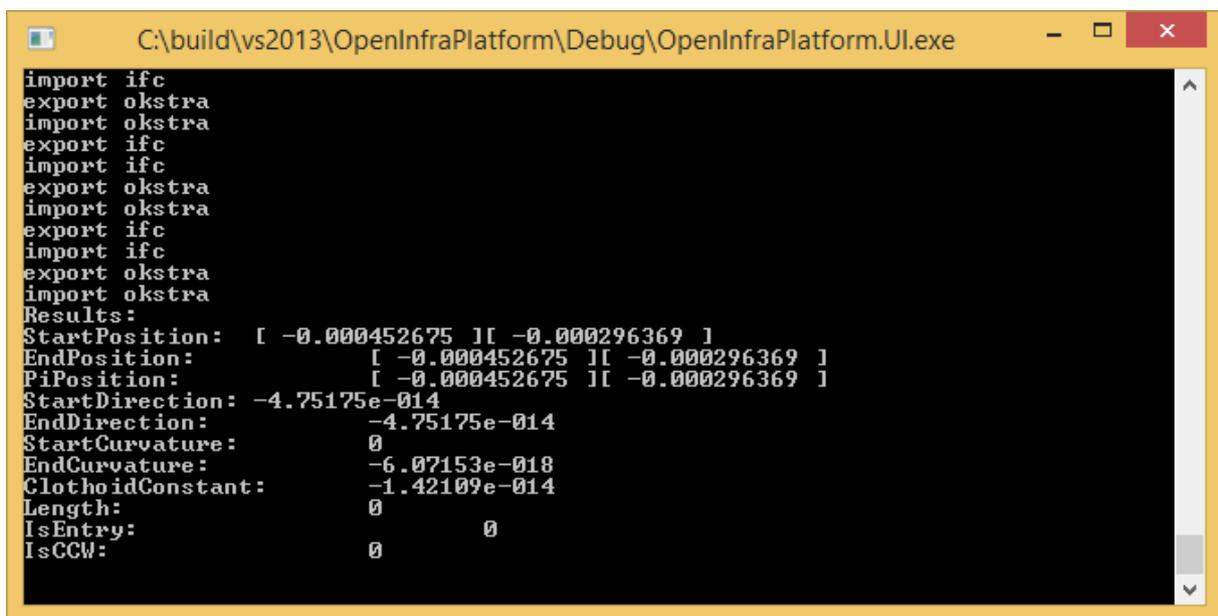
    if (0.123 * 100.0 / 100.0 != 0.123)
        std::cout << "auch verschieden" << std::endl;

    return 0;
}
```

Code-Snippet 1: Ein C++ Programm, das die Ausgabe "verschieden" und "auch verschieden" erzeugt.

Da bei der Konvertierung von IFC-Alignment nach OKSTRA ebenfalls arithmetische Operationen auf Basis von doppelter Genauigkeit durchgeführt werden, kann es zu kleinen Abweichungen bei der Umwandlung kommen.

Zur Bewertung der Abweichungen, die bei der Umwandlung von IFC-Alignment nach OKSTRA und umgekehrt entstehen können, haben wir einen Test implementiert. Dieser überführt eine IFC-Alignment-Datei (original.ifc) in eine OKSTRA-Datei. Anschließend wird diese OKSTRA-Datei wieder zurück in eine IFC-Alignment-Datei umgewandelt (konvertiert.ifc). Dieses Vorgehen entspricht genau einem Umwandlungszyklus in unserem Test. Diesen Umwandlungszyklus haben wir zehnmal wiederholt. Dabei ergaben sich die Abweichungen, die in Abbildung 13 zu sehen sind. Die Abweichung der Endkrümmung liegt bei $6,0 \cdot 10^{-18}$ und ist damit praktisch 0. Bei der Start- und Endposition kommen größere Abweichungen zustande, die jedoch ebenfalls in praktisch vertretbaren Grenzen liegen.



```
import ifc
export okstra
import okstra
export ifc
import ifc
export okstra
import okstra
export ifc
import ifc
export okstra
import okstra
Results:
StartPosition: [ -0.000452675 ][ -0.000296369 ]
EndPosition:   [ -0.000452675 ][ -0.000296369 ]
PiPosition:   [ -0.000452675 ][ -0.000296369 ]
StartDirection: -4.75175e-014
EndDirection:  -4.75175e-014
StartCurvature: 0
EndCurvature:  -6.07153e-018
ClothoidConstant: -1.42109e-014
Length: 0
IsEntry: 0
IsCCW: 0
```

Abbildung 13: Abweichungen bei zehnmalem Umwandlungszyklus

Bezüglich der numerischen Stabilität unserer Umwandlungsalgorithmen sind sicherlich noch Optimierungen möglich, jedoch sind die erzielten Abweichungen durchaus akzeptabel.

Ausblick mit Vorschlägen für weitere Forschungsarbeiten

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.

Folgende Projekte der buildingSMART International (bSI) befinden sich derzeit in Vorbereitung: IFC-Rail, IFC-Road sowie IFC-Bridge. Alle drei Projekte werden auf IFC-Alignment aufbauen. Zudem ist eine Ergänzung des IFC-Alignment-Modells geplant, die den Namen IFC-Alignment 1.1 trägt.

Im Rahmen des IFC-Alignment 1.1 Projekts soll IFC-Alignment und damit IFC 4 um weitere Übergangskurven, wie z.B. den Bloss-Bogen, erweitert werden. Außerdem soll eine einheitliche Möglichkeit für die Referenzierung bezüglich der Trassierung (dem Alignment) geschaffen werden. Dadurch soll es ermöglicht werden, beispielsweise Querprofile von Straßen (wie z. B. RQ 26) an eine bestimmte Stationierung zu binden. Abschließend sollen im Rahmen des IFC-Alignment Projekts 1.1 Offset-Alignments (Nebenachsen) eingeführt werden. Die Finanzierung des IFC-Alignment 1.1 Projekts ist derzeit noch nicht endgültig geklärt. Es zeichnet sich zwar die Beteiligung der bisherigen Geldgeber (Trafikverket und Rijkswaterstaat) ab, darüber hinaus ist die Beteiligung weiterer staatlicher Organisationen anderer Länder gewünscht.

Deutsche Institutionen sollten sich hier nach Möglichkeit an diesem und weiteren Infrastrukturerweiterungen des IFC-Standards beteiligen, sowohl mit eigenen Forschungsprojekten als auch als beratende Institution, um auf diese Weise deutsche Interessen bei der Schaffung europäischer und internationaler Standards zu vertreten.

Gleiches gilt auch für das IFC-Road-Projekt, das die Erweiterung von IFC 4 für den Straßenbau zum Ziel hat. Auch hier wurden schon umfangreiche Vorarbeiten von dritter Seite (maßgeblich Korea und China) geleistet. Es wird die ausdrückliche Empfehlung ausgesprochen, dass sich deutsche Verwaltungen und Softwarehersteller am IFC-Road-Projekt beteiligen und damit die umfassenden Erfahrungen, die im Rahmen der Entwicklung von OKSTRA gesammelt wurden, in die internationale Standardisierung einfließen lassen. Gleichzeitig sollte versucht werden, eine möglichst hohe Kompatibilität zwischen dem OKSTRA-Standard und den Infrastrukturerweiterungen des IFC-Standards zu erreichen.

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., Lebeque, 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

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

Inframodel.fi (2014). Application schema, version 3.0.1, <http://www.inframodel.fi/en/>

Lebeque, E., Fiès, 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

Singer, D.; Amann, J.: Erweiterung von IFC Alignment um Straßenquerschnitte In: Proc. of the 26th Forum Bauinformatik, Darmstadt, Deutschland, 2014

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.