EXTENSION OF THE UPCOMING IFC ALIGNMENT STANDARD WITH CROSS SECTIONS FOR ROAD DESIGN

Julian Amann[1], Dominic Singer[1], André Borrmann[2]

1) Ph.D. student, Chair of Computational Modeling and Simulation, Technische Universität München, Munich, Germany.
2) Dr.-Ing., Prof., Chair of Computational Modeling and Simulation, Technische Universität München, Munich, Germany.

Abstract: In the field of structural engineering, the data exchange of product models is mainly achieved using the Industry Foundation Classes (IFC). Until now, IFC is not very suitable for sharing product data models of infrastructure constructions, such as roads, tunnels or bridges. Important alignment information, such as horizontal or vertical alignment, is missing. Besides this, it is also not possible to describe the cross slope, superelavation and cross sections of roads with the help of IFC. As a consequence, buildingSMART, the organization that maintains IFC, started a project called “Infrastructure Alignment & Spatial Reference System” (known for short simply as “P6”) for developing an alignment and reference system. Based on the first version of the finalized conceptual model (“V 1.0”) developed by the P6 group we show how cross sections for roads can be integrated into this model. This demonstrates how IFC can be extended for use in road design applications.

Keywords: Infrastructure, data exchange, alignment, IFC, product model, road design

1. INTRODUCTION & MOTIVATION

In the field of structural engineering, the data exchange of product models is mainly achieved using the Industry Foundation Classes (IFC) (Eastman et al. 2011). Until now, IFC is not very suitable for sharing product data models of infrastructure constructions, such as roads, tunnels or bridges. Important alignment information, such as horizontal or vertical alignment is missing. Besides this, it is also not possible to describe the cross slope, superelavation and cross sections of roads with the help of IFC. As a consequence, buildingSMART, the organization that maintains IFC, started a project called “Infrastructure Alignment & Spatial Reference System” (known for short simply as “P6”) for developing an alignment and reference system. The goal of the project is to deliver a facility for storing alignments as part of the next upcoming release of IFC (IFC 4.x or 5), which can serve as a basis for all other alignment-based infrastructure constructions such as roads, tunnels or bridges.

In order to benefit from a neutral data standard in civil engineering in a manner similar to what already exists in structural engineering, we need an open, well accepted standard within the next few years. Since the P6 group only considers the alignment and does not currently consider the description of roads, we want to describe how road cross sections can be integrated in the upcoming IFC alignment model within this paper.

2. RELATED WORK

Several IFC based alignment models have already been proposed. The first one was the alignment model that was introduced in IFC-Bridge, an extension of the IFC Standard for bridge construction, which is still in ongoing development (Yakubi et al. 2006; Lebegue et al. 2012). The IFC Bridge extension introduces an *IfcReferenceCurveAlignment2D* element, which references a horizontal and a vertical alignment curve. For the horizontal as well as the vertical alignment, they use an *IfcCurve* element. Since *IfcCurve* is very general, it accommodates many different types of curve descriptions, which presents a challenge for software application implementers to handle all types and combinations of curve types. For instance editing the start or end radius of a clothoid is also difficult if it is described by arbitrary curve elements.

(Amann et al. 2013) describes a generalized IFC 4 based alignment model that can be used in the field of infrastructure to describe road, tunnel and bridge alignments. The model supports a 3D space curve (*IfcReferenceCurve3D*) as well as the traditional approach of horizontal and vertical alignments (*IfcReferenceAlignment2D*). The *IfcReferenceAlignment2D* consist of a gap and junction free horizontal (*IfcHorizontalAlignment*) and vertical alignment (*IfcVerticalAlignment*). The *IfcHorizontalAlignment* consist of an ordered list of *IfcHorizontalAlignmentSegments*. An *IfcHorizontalAlignmentSegment* is a superclass of *IfcHorizontalAlignmentLine* for line segments, *IfcHorizontalAlignmentCircularSegment* for circle segments and *IfcHorizontalAlignmentTransitionCurve* for transition curves. The only supported transition curve is the *IfcHorizontalAlignmentClothoid* for a clothoid. The vertical alignment consists of an ordered list of *IfcVerticalAlignmentSegments* such as *IfcVerticalAlignmentPointVerticalIntersection* and *IfcVertical-AlignmentRounding*. An *IfcVerticalAlignmentRounding* has only one subclass (*IfcVerticalAlignmentParabola*).
Instead of introducing new geometry representations for elements like straight lines or arcs, it references new geometry using existing geometry representations from the IFC. In particular, the extension contains the semantic elements *IfcLine* and *IfcCircle* that reference an *IfcTrimmedCurve* object to describe straight-line

segments and arcs. The semantic line and circle object do not introduce new geometric representations to avoid duplication of geometric descriptions: the IFC already contains many different options to describe straight lines and arcs. Similarly, a clothoid element is described with a trimmed curve. Since the standard IFC does not support clothoids, an *IfcHorizontalAlignmentClothoid* has been introduced to hold some specific data of the clothoid such as the clothoid constant.

Beside existing works that focus on alignment models, there are also several works that focus on topics that depend on alignment models, for example the proposed IFC-Tunnel product model, an extension of the IFC Standard which provides data structures for tunnel buildings. These efforts are mainly driven by the German IFC Tunneling Project (Hegemann et al. 2012), and by the Japanese Shield-Tunnel Project (Yabuki et al. 2007, Yabuki 2008).

3. CONTRIBUTION

We introduce the alignment model developed by P6. Furthermore, we show how the alignment model proposed by P6 can be extended with cross sections to describe a road body. Finally, to validate our approach we show a prototypical implementation of our extended alignment model.

3. INTEGRATION OF CROSS SECTIONS

3.1 Overview of the upcoming IFC Alignment model

Figure 1 shows an overview of the IFC alignment model proposed by the P6 project members (Liebich 2014). Note that this is only a draft that may change in future. The model can be roughly divided in two parts. The first part describes the horizontal alignment, the second the vertical alignment.

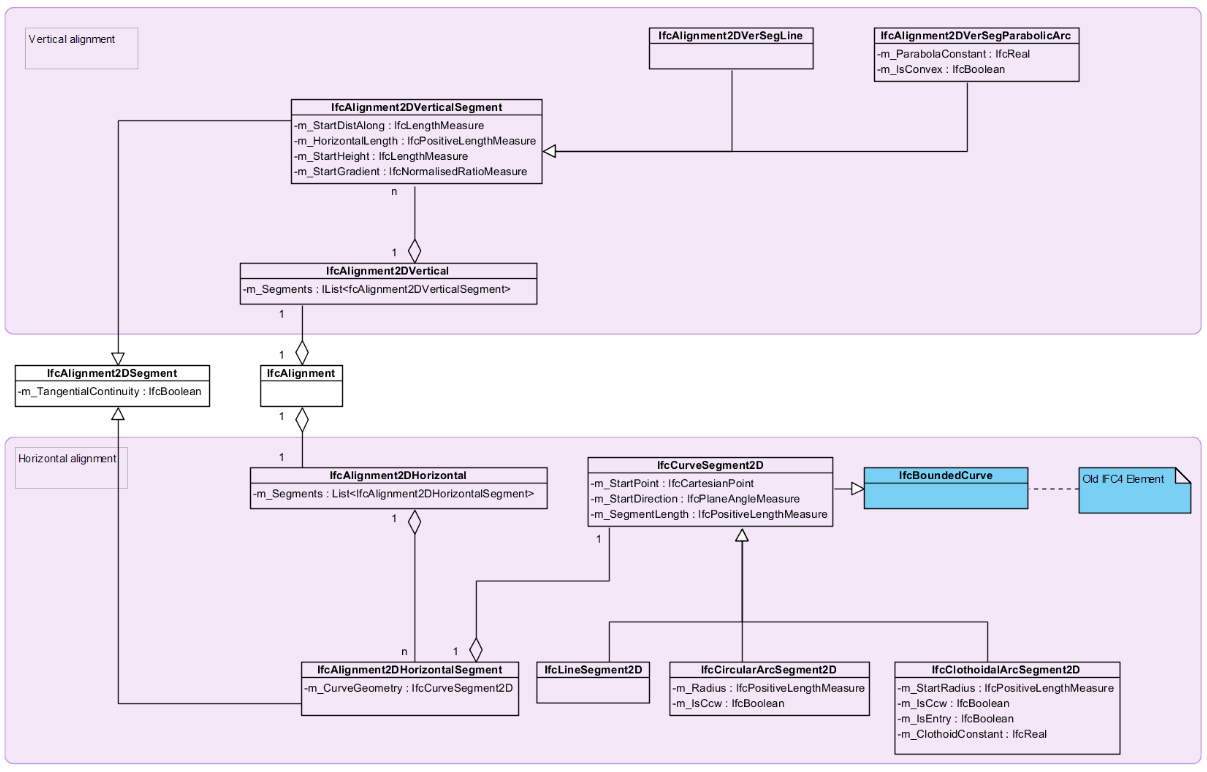
Vertical alignment
IfcAlignment2DVerSegLine
IfcAlignment2DVerSegParabolicArc
-m_ParabolaConstant : IfcReal
-m_IsConvex : IfcBoolean
IfcAlignment2DVerticalSegment
-m_StartDistAlong : IfcLengthMeasure
-m_HorizontalLength : IfcPositiveLengthMeasure
-m_StartHeight : IfcLengthMeasure
-m_StartGradient : IfcNormalisedRatioMeasure
n
1
IfcAlignment2DVertical
-m_Segments : IList<fcAlignment2DVerticalSegment>
1
1
IfcAlignment2DSegment
-m_TangentialContinuity : IfcBoolean
IfcAlignment
1
1
Horizontal alignment
IfcAlignment2DHorizontal
-m_Segments : List<IfcAlignment2DHorizontalSegment>
1
IfcCurveSegment2D
-m_StartPoint : IfcCartesianPoint
-m_StartDirection : IfcPlaneAngleMeasure
-m_SegmentLength : IfcPositiveLengthMeasure
IfcBoundedCurve
Old IFC4 Element
1
n
1
IfcAlignment2DHorizontalSegment
-m_CurveGeometry : IfcCurveSegment2D
IfcLineSegment2D
IfcCircularArcSegment2D
-m_Radius : IfcPositiveLengthMeasure
-m_IsCcw : IfcBoolean
IfcClothoidalArcSegment2D
-m_StartRadius : IfcPositiveLengthMeasure
-m_IsCcw : IfcBoolean
-m_IsEntry : IfcBoolean
-m_ClothoidConstant : IfcReal

Figure 1: Overview of the IFC alignment model proposed by the P6 project members.

In the current draft the horizontal alignment consists of straight lines (*IfcLineSegment2D*), arcs (*IfcCircularArcSegment2D*) and clothoids (*IfcClothoidalArcSegment2D*). It is expected that other transition curve types besides clothoids will be supported in an upcoming new revision of the current draft. All horizontal alignment elements (*IfcLineSegment2D, IfcCircularArcSegment2D* and *IfcClothoidalArcSegment2D*) have the same common properties: Start point (*m_StartPoint*), start direction (*m_StartDirection*) and segment length (*m_SegmentLength*) which are inherited from *IfcCurveSegement2D*. Additionally the *IfcCircularArcSegment2D* provides a radius (*m_Radius*) and an orientation of the circular arc (*m_IsCcw*). The *IfcClothoidalArcSegment2D* also provides a start radius (*m_StartRadius*) as the radius of the clothoidal arc at the start point. If the radius is not provided by a value, i.e. is “NIL”, it is interpreted as being INFINITE, which means that the curvature at the start point is zero. Additionally, the *m_IsCcw* attribute denotes the orientation of the clothoidal arc with Boolean="true" being counter-clockwise, or “to the left”, and Boolean="false" being clockwise, or “to the right”. The attribute *m_isEntry* defines if the curvature is increasing or decreasing towards the end point. If it is increasing, the value of *m_isEntry* is "true", otherwise it is "false". Finally, the clothoid constant A (not A squared), that determines the rate of curvature change along the clothoid, is stored in the attribute *m_Clothoid*. Figure 2 shows a graphical overview of the parameters of the different horizontal alignment elements.

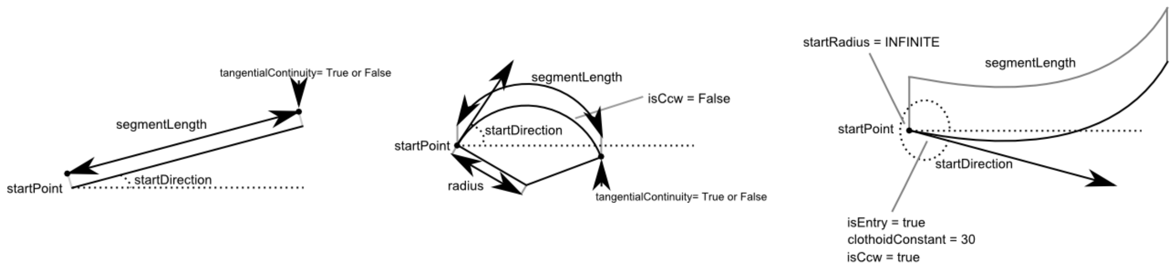

Figure 2: Overview of the parameters of the different horizontal alignment elements. From left to right: IfcLineSegement2D, IfcCircularArcSegment2D and IfcClothoidalArcSegment2D

Different horizontal alignment segments can be concatenated to a horizontal alignment. A horizontal alignment is realized as a list of *IfcAlignment2DHorizontalSegment* elements. This list (*m_Segments*) is managed by the *IfcAlignment2DHorizontal* entity. A horizontal alignment is assumed to be gap free. End points are not stored in the horizontal alignment element as they can be computed from the start point, segment length and the other given properties. Being gap-less, the start point of a subsequent segment matches the calculated end point of the previous segment. The connectivity between the continuous segments does not necessarily have to be tangential. The attribute *m_TangentialContinuity* defines whether two segments are tangential or not. Figure 3 shows an alignment were the tangential continuity is not fulfilled. The tangential continuity flag makes it possible to check whether the calculated end direction of the previous segment matches the provided start direction of the current segment.

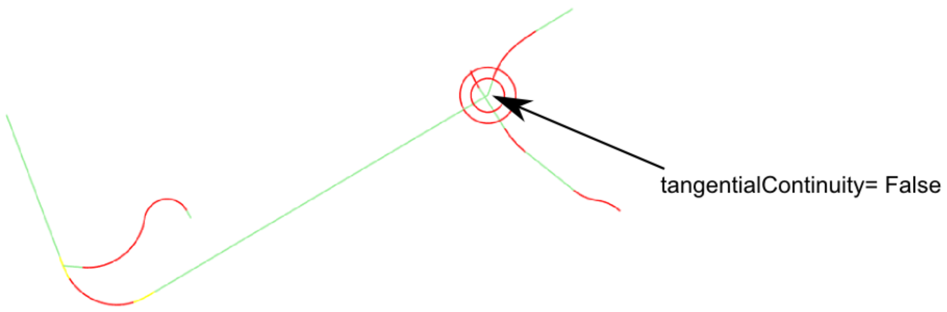

Figure 3: Tangential continuity.

On top of the model there is *IfcAlignment*, which references a horizontal alignment and a vertical alignment. The vertical alignment (*IfcAlignment2DVertical*) is also composed of elements. The vertical alignment elements are defined in xy-space in which x describes the current station (distance along the horizontal alignment) and y describes the corresponding height (elevation) value at this station. Figure 4 shows how the horizontal and vertical alignment can be combined to derive a 3D alignment form it.

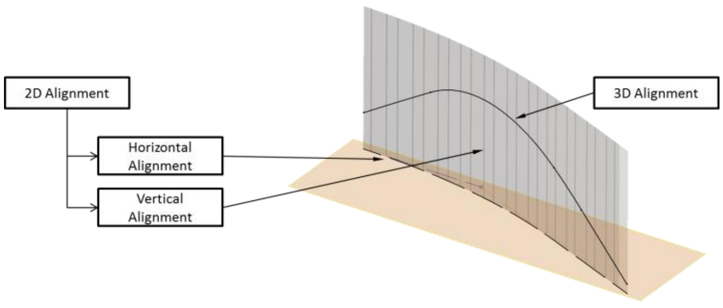

Figure 4: If the horizontal and vertical alignment is combined we can derive a 3D alignment from it.

Horizontal alignment elements (*IfcAlignmentSegment2D*) can be straight line segments (*IfcAlignment2DVerSegLine*), arcs (*IfcAlignment2DVerSegParabolicArc*) and parabolas. The vertical alignment does not have to be tangential. Again, the “tangential continuity” flag can be used to enforce it (or not). The vertical alignment definition also supports so-called “unsymmetrical parabolic arcs”. Unsymmetrical parabolic arcs can be realized using two connected parabolic arcs with different parabola constants. Each vertical alignment segment provides a start point (station value/height coordinate), a start gradient (in percentage, with horizontal being 0, uphill positive, and downhill negative), a length value (as horizontal length along the distance, not the curve segment length) and additional curve parameters.

3.2 Extension with Cross Sections for Road Design

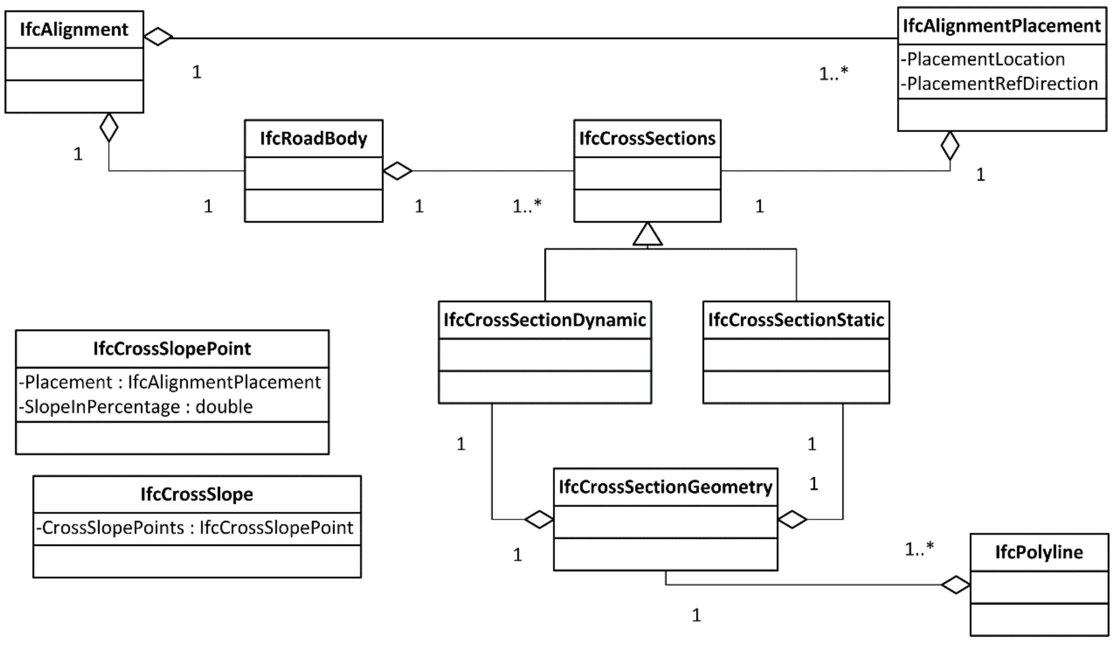

Figure 5: Curvature String and Cross Slope String in road design

Figure 5 shows the extension of the P6 IFC Alignment proposal for cross sections. The supplemented entities are defined in a UML-diagram. First, a cross fall string (*IfcCrossSlope*) extends the entity IfcReferenceCurveAlignment2D. The cross fall string is represented as a sequence of points of *IfcCrossSlopePoints*. Each of these points stores a station (*Station*) and the corresponding cross slope (*SlopeInPercentage*). *IfcCrossSlope* manages a list of consecutive *IfcCrossSlopePoints* pairs consisting of the values of the respective station and *SlopeInPercentage*. To determine the cross slope for a station that is not explicitly stored, it can be derived from a simple linear interpolation between the corresponding *IfcCrossSlopePoints*. The *IfcCrossSlope* is optional. Thus, alignments can be stored which do not include cross slope information.

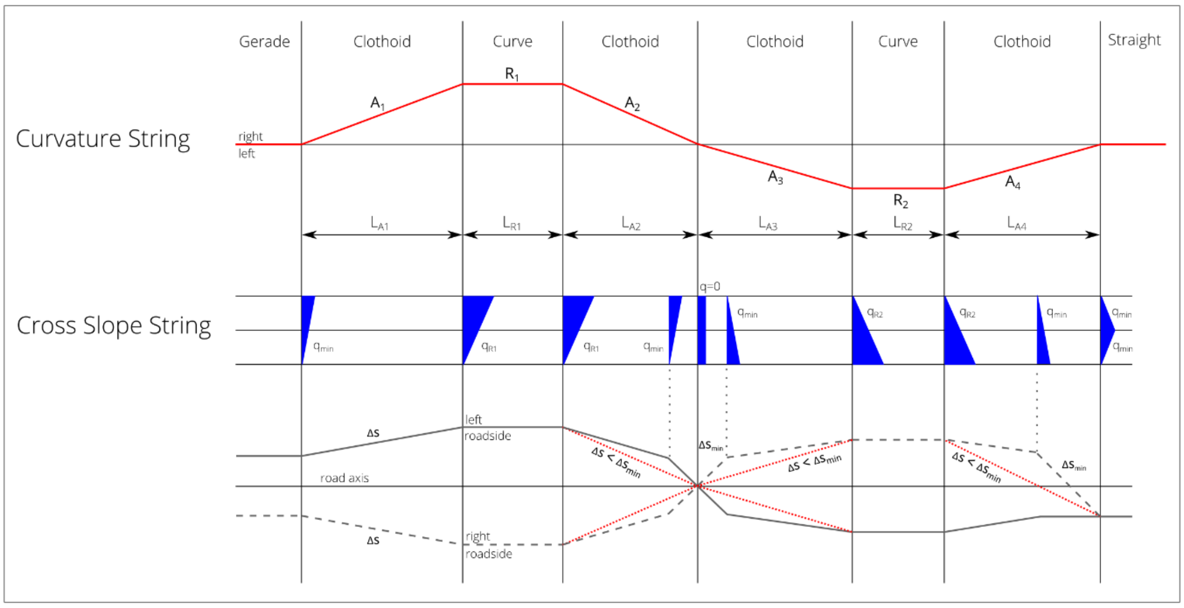

Figure 6: Extension of the P6 IfcAlignment draft

To store the *IfcCrossSection* elements, the entity *IfcRoadBody* is introduced. An *IfcRoadBody* entity references one *IfcAlignment* object. *IfcCrossSection* inherits two classes, namely *IfcCrossSectionStatic* and *IfcCrossSectionDynamic*. *IfcCrossSectionStatic* defines the cross section at a station. This deployment is stored in the *IfcAlignmentPlacement* attribute of the station *IfcCrossSection*. The *IfcCrossSectionStatic* is referenced even on an *IfcCrossSectionGeometry* element which maps the road cross section with a list of *IfcPolyline* objects. A 2D profile is therefore used to describe a cross section. The origin of the cross section is the associated 3D axis point of the current station of the referenced *IfcAlignmentPlacement* attribute *PlacementLocation*. The cross section is defined perpendicular to the tangent at the current station point (*PlacementRefDirection*). The static transverse profile (*IfcCrossSectionStatic*) is independent of the cross slope string. Figure 6 shows the familiar two-dimensional description of the curvature-, cross slope- and ramp string of a path.

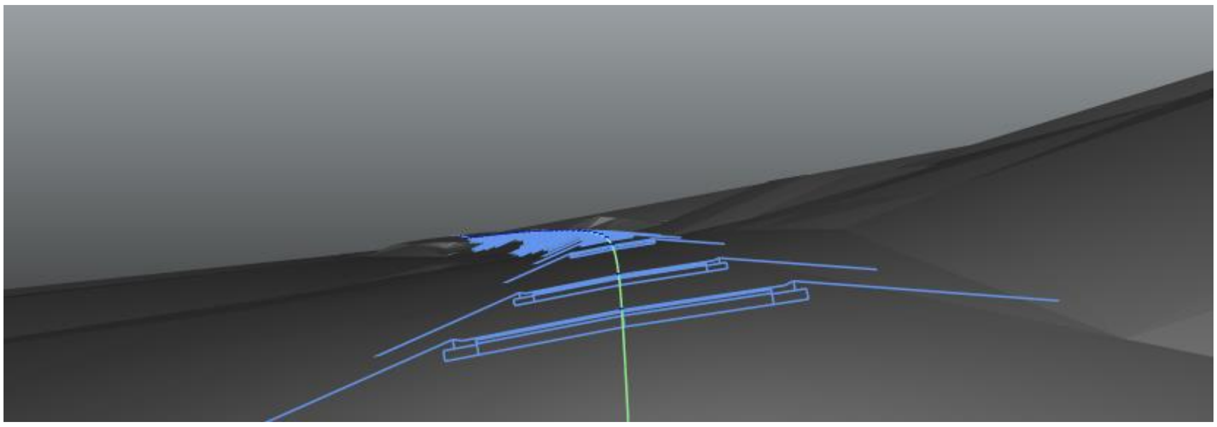

Figure 7: Cross section of the road body based on IfcCrossSection

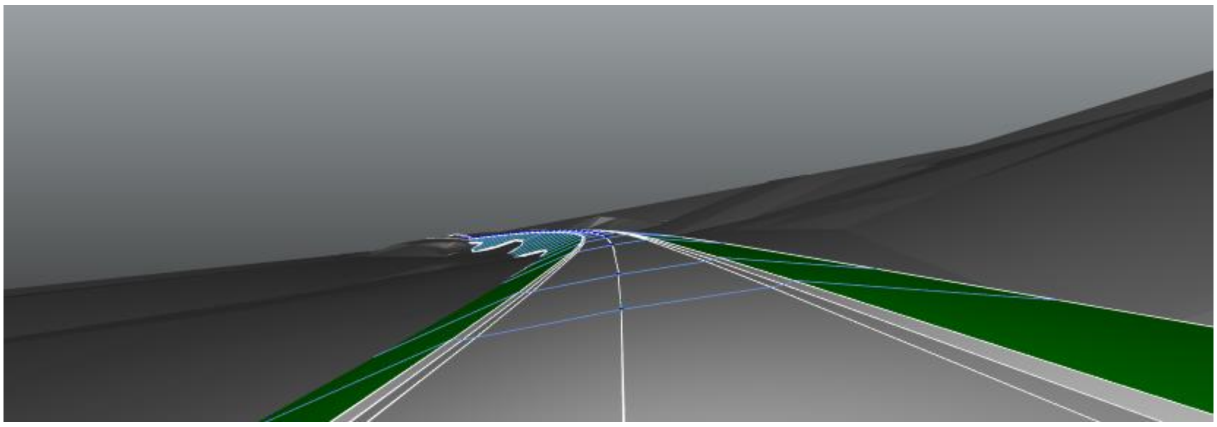
Figure 8: Road body based on the IfcCrossSection

The dynamic cross section (*IfcCrossSectionDynamic*) depends on the ramp string, unlike the static cross section. It stores its geometry in the same way as the static cross section. However, the geometry serves as a kind of template, which describes the cross section in a non-rotated state. With the help of the *CrossSectionDynamic*, a cross section can also be determined for other non-explicit stations, in which case the *IfcCrossSlopePoints* have to be considered. If you want to calculate a cross section at station s, you just have to identify the cross slope at this station s via the *IfcCrossSlopePoints* and twist the template accordingly.

4. VALIDATION

To validate the proposed approach, the extended alignment model has been implemented in the TUM Open Infra Platfom (see figure 9). To achieve this the IFC Alignment EXPRESS Schema has been extended. Afterwards, a late binding has been generated for it. Figure 10 shows a snippet of an IFC Alignment STEP file extended with our cross section proposal. Additionally the software can convert LandXML files (see LandXML.org 2014) which contain cross sections to our proposed IFC Alignment extension and vice versa.

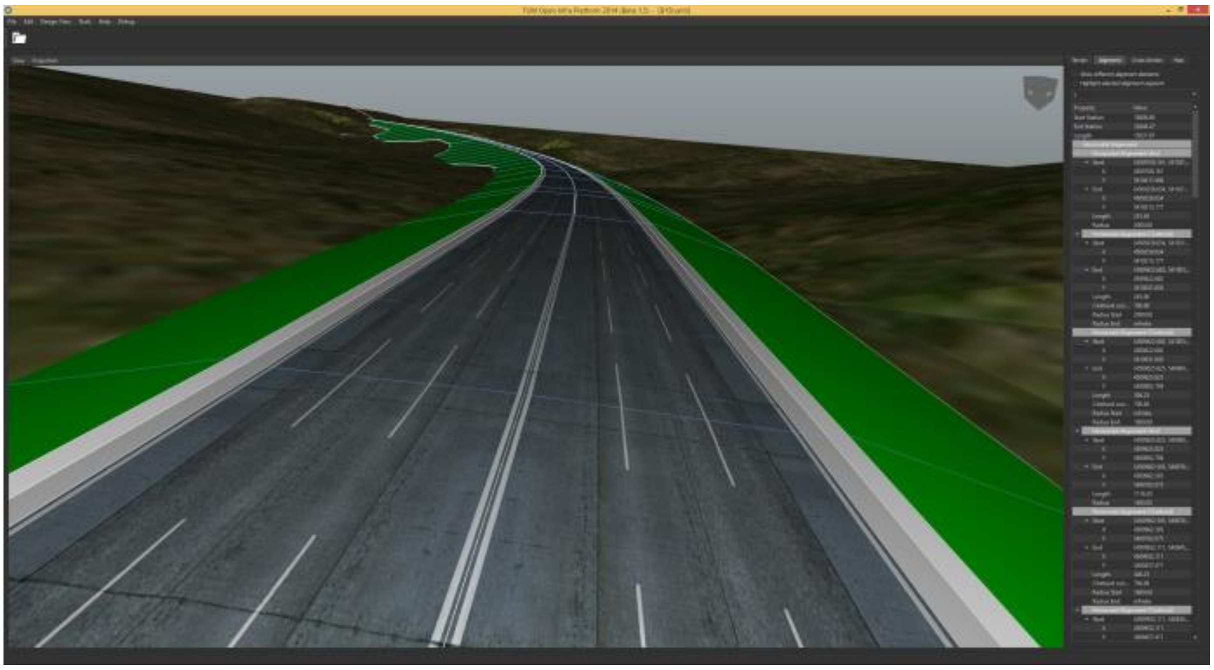
Figure 9: Integration of the data model into the TUM Open Infra Platform

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('IFC4'),'2;1');
FILE_NAME('IfcAlignment-export.ifc','2014-07-17T23:38:32',(''),('',''),'','IfcAlign
ment','');
FILE_SCHEMA(('IFC4'));
ENDSEC;
DATA;
..
#35=IFCREFERENCECURVEALIGNMENT2D($,$,$,$,$,$,$,#36,#96,$);
..
#135=IFCROAD($,$,$,$,$,$,$,$,$,#35,(#136,#139,#142,#145,#148,#151,#154,#157,#160,#1
63,#166,#169,#172,#175,#178,#181,#184,#187,#190,#193,#196,#199,#202,#205,#208,#211,
#214,#217,#220,#223,#226,#229,#232,#235));
#136=IFCCROSSSECTIONSTATIC(240,#137);
#137=IFCCROSSSECTIONGEOMETRY((#138));
#138=IFCPOLYLINE((...));
#139=IFCCROSSSECTIONSTATIC(245,#140);
#140=IFCCROSSSECTIONGEOMETRY((#141));
#141=IFCPOLYLINE((...));
#142=IFCCROSSSECTIONSTATIC(250,#143);
#143=IFCCROSSSECTIONGEOMETRY((#144));
#144=IFCPOLYLINE((...));
#145=IFCCROSSSECTIONSTATIC(253.30000000000001,#146);
#146=IFCCROSSSECTIONGEOMETRY((#147));
#147=IFCPOLYLINE((...));
#148=IFCCROSSSECTIONSTATIC(254.30000000000001,#149);
#149=IFCCROSSSECTIONGEOMETRY((#150));
#150=IFCPOLYLINE((...));
#151=IFCCROSSSECTIONSTATIC(255,#152);
#152=IFCCROSSSECTIONGEOMETRY((#153));
#153=IFCPOLYLINE((...));
#154=IFCCROSSSECTIONSTATIC(260,#155);
#155=IFCCROSSSECTIONGEOMETRY((#156));
#156=IFCPOLYLINE((...));
#157=IFCCROSSSECTIONSTATIC(260.65600000000001,#158);
#158=IFCCROSSSECTIONGEOMETRY((#159));
#159=IFCPOLYLINE((...));
#160=IFCCROSSSECTIONSTATIC(260.80000000000001,#161);
#161=IFCCROSSSECTIONGEOMETRY((#162));
#162=IFCPOLYLINE((...));
```

Figure 10: Snippet of an IFC Alignment STEP file extended with our cross section proposal

5. FUTURE WORK AND CONCLUSION

Within the scope of this work we have demonstrated how the upcoming IFC alignment schema developed by buildingSMART can be extended with cross sections. Until now, we only considered junction free road sections. In future work we will also consider junctions. To solve this issue we plan to introduce an *IfcRoadJunction* element, which should serve as a docking point for different alignment segments to connect different road alignments.

In addition, we await the release of the development concept of IfcRoad Extension (Hyounseok 2014) that is being driven by the Korea Institute of Construction Technology. The IfcRoad Extension looks very promising but has not yet been released. Future work may therefore also take into account the ideas proposed in this extension.

This paper reflects the current status of our work in the development of a cross section extension and as such should not be considered an end state. If anything, it should be considered as a starting point. This draft will need further discussion with an expert panel and verification before internationally accepted.

REFERENCES

Amann, J.; Borrmann, A.; Hegemann, F.; Jubierre, J.R.; Flurl, M.; Koch, C.; König, M. (2013). A Refined Product Model for Shield Tunnels Based on a Generalized Approach for Alignment Representation In: Proc. of the ICCBEI 2013, Tokyo, Japan

buildingSMART. (2013). Industry Fondation Classes IFC4 Official Release, buildingSMART http://www.buildingsmart-tech.org

Eastman C., Teicholz P., Sacks R., Liston K (2011) BIM Handbook: A Guide to Building Information Modeling for Owners (Second Editon), Managers, Designers, Engineers and Contractors, Wiley Publishing, ISBN: 0470541377

Hegemann, F., Lehner K., König, M. (2012), IFC-based product modeling for tunnel boring machines, Proceedings of the 9th European Conference on Product and Process Modeling 2012, Reykjavik, Iceland, 289-296

Hyounseok, M. (2014), Development Concept of IfcRoad Extension, Presentation Slides of Infra Room meeting in Stockholm, http://iug.buildingsmart.org/resources/itm-and-iug-meetings-2014-stockholm/infra-room-meeting/development-concept-of-ifcroad-extension-in-korea/at_download/file

LandXML.org (2014). LandXML 2.0 (Working Draft) Schema Announced, http://landxml.org/

Lebegue, E., Fiês, B. and Gual, J., (2012). IFC-Bridge V3 Data Model – IFC 4, Edition R1.

Liebich, T (2014). IFC-Alignment IFC Extension Summary, P6 "IFC Alignment" project, budilingSMART. http://www.buildingsmart-tech.org/infrastructure/projects/alignment

Yabuki, N., Lebeque, E., Gual, J., Shitani, T. and Li, Z. T., (2006). International Collaboration for Developing the Bridge Product Model IFC-Bridge, *In Proc. Of the International Conference on Computing and Decision Making in Civil and Building Engineering.* pp.1927-1936.

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. pp.545-550