An alignment meta-model for the comparison of alignment product models

J. Amann, J.R. Jubierre & A. Borrmann
Chair of Computational Modeling and Simulation

M. Flurl
Chair for Computation in Engineering

Leonhard Obermeyer Center, Technische Universität München, Germany

ABSTRACT: In this paper, an alignment meta-model is presented that can be used as a common basis to compare arbitrary alignment models. To this end, we describe how existing alignment product models can be mapped to the proposed meta-alignment model. In addition, we define quality metrics of a well-performing alignment model, which can be used to evaluate different design alternatives for alignment models.

1 INTRODUCTION & MOTIVATION

Typical infrastructure projects such as road, bridge or tunnel buildings are driven by an alignment model, which describes the course of a carriageway and, thus, represents the highest level of abstraction in the planning process of this kind of infrastructure projects. If the alignment is changed, all elements depending on the alignment must also be modified accordingly, e.g. the superstructure and substructure of a bridge. Therefore, the correct definition of alignment data is crucial for any open data standard representing information about infrastructures.

Throughout the lifecycle of the facility, the availability of a standardized infrastructure asset data is an imported key to a higher efficiency in planning, constructing and operating of infrastructure projects. A comprehensive neutral data model capable to present both semantic and geometric aspects is necessary for enabling data exchange and opening data access in the context of planning, realization and maintenance of road and rail infrastructure. Without a neutral data standard an efficient workflow between different project partners such as contractors, engineers, structure analysis experts and other stakeholders in the planning process and lifecycle of infrastructure projects is nearly impossible. Thus, a neutral standard would offer many advantages.

The Industry Foundation Classes (IFC) (steered by the buildingSMART organization) provide a standardized product model for the design and construction of buildings that had been highly adopted by the industry. Something similar is missing for the infrastructure sector. Since IFC is well established and provides a solid framework supported by many software applications it makes sense to build an alignment model based on IFC.

There are already some existing (pseudo) standards for storing and exchanging alignment data such as LandXML, OKSTRA, or RoadXML. Besides this some proposals (e.g. Amann et al. 2013) were made for alignment data models. Because of different issues in the existing approaches we suggest an IFC 4 based alignment model for infrastructure design as proposed in Amann et al. 2013.

2 CONTRIBUTION

In this paper, a novel alignment meta-model is presented, which facilitates the comparison of different alignment models in a qualitative manner. First, we describe how arbitrary alignment product models can be mapped to our proposed alignment meta-model and secondly, how our proposed meta-model can be used to evaluate the different design alternatives for alignment models.

3 RECENT WORK

3.1 *Existing (Pseudo) Standards for Alignment Data Models*

The development of alignment models started in the sixties of the last century (Rebolj et al. 2008). Since then, many different alignment models have been developed. Today the most frequently used data model is *LandXML* (Rebolj et al. 2008) focusing on the data exchange of alignments between different experts. Nevertheless, LandXML is quite broken and has several issues: Minimally documented, syntax errors in

LandXML 1.2 Schema, Weak Point Typing, Case Inconsistencies, Name Optionality Inconsistency, etc. (Scarponcini 2013). The development on LandXML has suddenly stopped in 2009, while after 4 years of complete silence, in December 2013 suddenly an update on the LandXML.org webpage appeared. Right now it is not clear if the development of LandXML will be continued. A major problem is that the LandXML data format is currently not supported by a standard organization that guarantees its longevity. Besides this, some legal and organizational issues have to be clarified. For instance, it is not clear who owns the copyright of LandXML or who is the official maintainer of LandXML.

The OpenSpatialGroup considered LandXML to integrate into their existing standards. Because of the many different problems described above (see also Scarponcini 2013) they decided against the integration of the LandXML standard. Instead they plan to create InfraGML, a standard that targets the same use cases (alignment, roads, drainage, parcels, etc.) as LandXML does. Right now *InfraGML* is in the requirement analysis phase and up to now there is no data model available.

The German standard *OKSTRA* (**O**bjekt**k**atalog für das **Stra**ßen- und Verkehrswesen) unifies the data description of objects from traffic engineering and also contains a data model for alignment data. It is owned by BASt (German Federal Highway Research Institute) and distributed under a free license. The main design goal of OKSTRA is the simple data exchange between different applications that implement the standard. The current OKSTRA standard (version 2.016) has been published in January 2014. In the past the OKSTRA standard has been delivered as an EXPRESS schema along with and XSD schema, while in the current version the data model is solely described as a XSD. Also the CTE data format (STEP based data format to store EXPRESS based instance files) was retired. In detail, the OKSTRA developers moved from EXPRESS to XSD, from NAIM Diagrams (similar to EXPRESS-G Diagram) to UML and from STEP to store instance files (also called CTE files within the OKSTRA standard) to XML.

A similar transition from EXPRESS/EXPRESS-G/STEP to XSD/UML/XML will also appear probable in the IFC standard, since UML, XSD and XML are better supported by tools than EXPRESS, STEP and EXPRESS-G (since 2001 ifcXML is available). OKSTRA uses several packages of the ISO harmonized model (maintained by ISO/TC 211 Geographic information/Geomatics). Table 1 gives an overview of the different used standards.

Table 1. Overview of the different standards used by OKSTRA

Standard name
ISO 00639 Language Codes
ISO 03166 Country Codes
ISO 19103:2005 Schema Language
ISO 19107:2003 Spatial Schema
ISO 19108:2006 Temporal Schema
ISO 19109 Application Schema
ISO 19111 Referencing by Coordinates
ISO 19115:2006 Metadata (Corrigendum)
ISO 19136 GML
ISO 19139 Metadata – XML Implementation
ISO 19148 Linear Referencing
ISO 19156:2011 Observations and Measurements

The major problem of the OKSTRA standard is that it only targets the German market. In the provided XSD schema only identifiers in German language are used and the documentation is also only provided in German language hindering this standard to become a good candidate for an international standard.

RoadXML (current version 2.4) is an open file format for the logical description of road networks (Chaplier et al. 2010) and targeted for vehicle driving simulations. It is based on XML and contains also and alignment data model. In contrast to OKSTRA it is a very lightweight standard (see Table 2). Its main goal is to enhance the interoperability between different driving simulators and traffic engineering applications. Currently RoadXML is maintained by the RoadXML Board which encompasses INRETS (French National Institute for Transport and Safety Research), OKTAL (French company that develops simulation software and systems for vehicles), PSA Peugeot Citroën (French vehicle manufacturer), Renault S.A. (French vehicle manufacturer), and the Thales Group (French provider for electrical systems and services for aerospace, defense, transportation and security).

Table 2 gives an overview of the complexity of different described standards. Of course *LandXML* and OKSTRA are not only limited to alignments and have also other use cases like land acquisition or accident documentation. Lines of Code are not the best measurement for complexity, but they show at least a hint of how complex these data models are.

Table 2. Complexity in Lines of Code

(Pseudo) Standard	XSD Lines of Code	Last Update
LandXML 1.2	4821	2008
OKSTRA 2.016	28400	2013
RoadXML 2.4	594	2013

Road product models have a long history. There are many more data models for roads such as *The Road Shape Model Kernel* (RSMK) which has been developed by the Building and Construction Research group in the Netherlands (TNO Institute). The Swedish company *EuroSTEP* has developed a road product model for the Swedish National Road Administration based on STEP. Furthermore, *TransXML* has been developed by the US National Cooperative Highway Research Program for the interchange of transportation data. More details to the evolution of product data models for road design can be found in Rebolj et al. (2008). Nevertheless, the last updates to the existing (pseudo) standards were made to LandXML, OKSTRA and RoadXML. It seems that RSMK, EuroSTEP and TransXML are not further developed and the last updates to them were made before 2007.

3.2 *Proposals for Alignment Data Models*

Recently, several proposals for the integration of an alignment model into IFC were made. Amann et al. 2013 describe an approach that is similar to the one used by LandXML. It supports the description of an alignment by using a 2D approach based on a horizontal and a vertical alignment. The alignment model described in Amann et al. 2013 has a few shortcomings. First of all, it does not describe in detail how transition curve types besides clothoids can be described. Additionally, the model does not support multiple vertical alignments for one specific horizontal alignment, a feature requested by many stakeholders. Furthermore, it ignores the approach of modeling roads as string lines as proposed by the Finnish Inframodel (Hyvärinen 2011), which must be considered to satisfy the needs of an international standard.

In buildingSMART MVD 2011 an IFC Model View Definition for a LandXML v1.2 has been introduced. The main idea of this approach is to simply embed a well-defined IDM/MVD for LandXML into IFC. This way, LandXML can be used within the IFC environment. The authors themselves consider this approach only as an interim solution (since it suffers from the similar flaws as LandXML) until a standard for infrastructure based on IFC is developed.

OpenBrIM is a XML file format for bridges, but includes also an alignment model. The description of the horizontal and vertical alignment elements differ from the ones used by Amann et al. 2013 and the Finnish Inframodel.

Besides the above mentioned model some other proposals were made (Kim 2013, Liebich 2013, Bentley Systems 2013, Scarponcini 2013) as well as some older proposals, such as the IfcBridge Draft (Lebegue et al. 2012), which also contains an alignment model.

4 A META ALIGNMENT MODEL

Comparing different alignment models is quite a complex task. One approach is to define a meta-model that describes the underlying semantics of the alignment data model. This alignment meta-model can be used to convert different file formats for alignment data originating from different models.

4.1 *A proposal for an alignment meta-model*

The following meta-model is not intended for implementation, but only targets the description concerning the semantics of different objects used in an alignment model. A specific implementation of an alignment model can and will of course be different. The alignment meta-model distinguishes between storing two-dimensional based and three-dimensional based alignment models. An alignment can be a 3D curve or a 2D alignment consisting of a horizontal and a vertical alignment. Figure 1 depicts an UML view of this alignment meta-model.

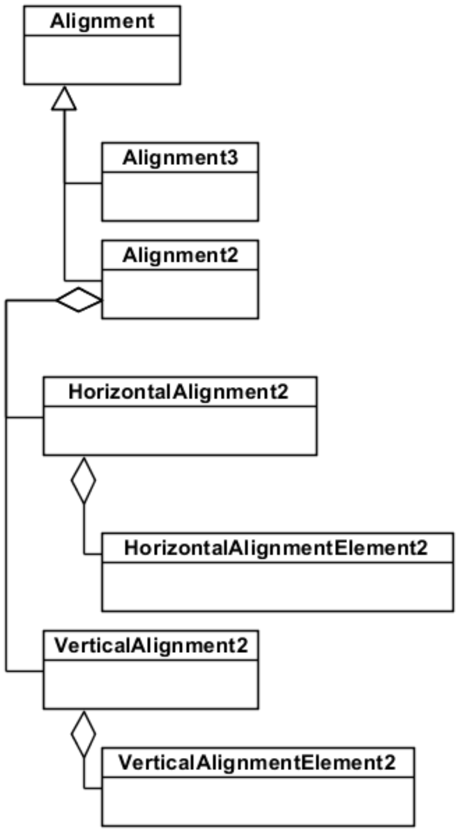

Figure 1. Top level view of the meta-alignment model.

This meta-model is used as a basis for a common ontology used to analyze different alignment product models. *Alignment3* implies a 3D based alignment model, while Alignment2 represents an alignment based on a two-dimensional approach.

A horizontal alignment consists of horizontal alignment elements, such as straight line segments, circle segments (arcs), and transition curves (depicted in Figure 2). An example for a transition curve is a clothoid.

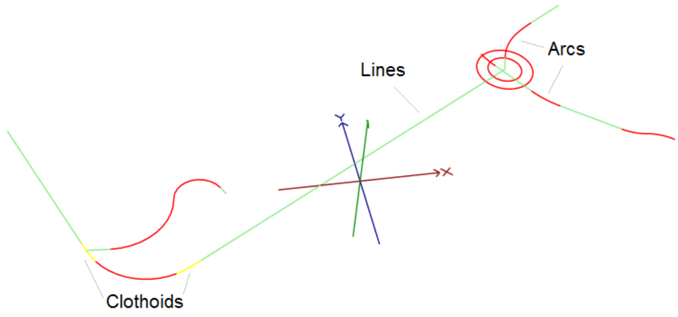

Figure 2. Usually the horizontal alignment consists of lines, arcs and clothoids.

Considering different horizontal alignment elements we frequently examine different properties, such as start- and end points, length, or other data like curvature. For instance, considering a straight line the start- and corresponding endpoint, the length, or direction are of common interest. A circle segment (arc) can be described by its start, end, and center point. Additionally, the rotation order (clockwise or counter-clockwise) is part of the meta-model. For clothoids, the start and end point as well as the point of intersection are of interest, among other things. Figure 3 represents an overview of line, arc and transition curve. Additionally, it can be observed that the alignment meta-model assumes some base types, such as *vector2*, which describes a two-dimensional vector, *real*, which describes a floating point number and a *positive real* refers to a positive floating point number. The digit 2 at the end of the descriptors *Base2*, *Line2*, *Arc2*, etc. emphasizes that we are facing a two-dimensional element.

It should be mentioned, that some parameters can be computed by other parameters. For instance, the radius of an arc can be gained from its curvature. This information can be noted using the UML Object Constraint Language. Clearly, also other constraint can be noted this way (like radius >= 0):

```
context Arc2 inv:
self.radius = 1/self.curvature
context Arc2 inv:
self.radius >= 0
```

Concerning the alignment meta-model, redundant data does not have a negative impact; on the contrary, it is desired to have all the different variants and possibilities to describe a certain alignment element contained in the model in order to compare different models. Thus, the basic idea is to assign unique semantics to each alignment element. For instance, there are different possible ways to describe a line. One model might describe it by start and end point, while another one might describe it by the start point, direction vector and a length value. In fact, both descriptions include different representations of the same information. Thus, the comparison based on attributes is a difficult task.

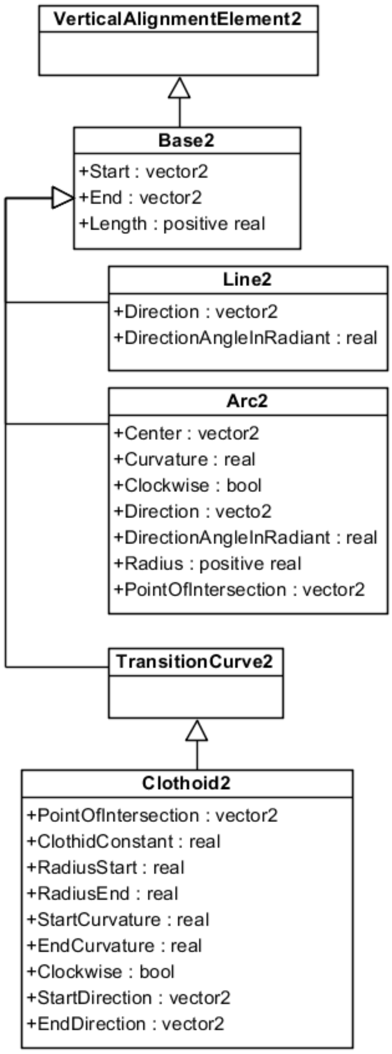

Figure 3. UML class diagram that shows different horizontal alignment elements of the alignment meta-model.

It is more convenient and expedient to find a mapping to the alignment meta-model and compare the models referring to a common denominator. Nevertheless, there might be a national standard like OKSTRA that hardly can be understood by non-German speakers. Besides, the alignment meta-model enables non experts to get some insight in various alignment product models without being an adept in every specific domain. For instance if you are not familiar with IFC/EXPRESS it might be hard to understand a model that is based on this concepts. The alignment meta-model facilitates a meaningful comparison between different alignment data models.

4.2 *Mapping between LandXML and the alignment meta model*

Table 2 represents a mapping of LandXML 1.2 to the alignment meta-model. For the mapping XQuery is utilized (we assume here that $x = LandXML/Alignments/Alignment holds).

Table 2. Mapping of LandXML 1.2 to Meta alignment model

Meta Alignment Model	LandXML 12 XQuery
Line2	$x/CoordGeom/Line
Line2.Start	$x/CoordGeom/Line/Start
Line2.End	$x/CoordGeom/Line/End
Line2.Lenght	$x/CoordGeom/Line/@length
Line2.DirectionAngleInRadiant	$x/CoordGeom/Line/@dir
Arc2.Start	$x/CoordGeom/Curve/Start
Arc2.Center	$x/CoordGeom/Curve/Center
Arc2.End	$x/CoordGeom/Curve/End
Arc2.Clockwise	$x/CoordGeom/Curve/@rot
Arc2.Radius	$x/CoordGeom/Curve/@radius
Arc2.Length	$x/CoordGeom/Curve/@length
…	…

This mapping process supports the detection where different data is stored in a LandXML instance file. A similar process can be applied for other XML based alignment models.

4.3 *Quality metrics of an alignment model*

The first stated propose of the alignment meta-model is to have a common denominator, since normally not everyone is an expert on every alignment product model, maybe because of language problems, technology problems or just because of the complexity of an alignment model.

Besides, the alignment meta-model supports the evaluation of the quality of a specific alignment model. Thus, in a first step the basic qualities of a good alignment model have to be defined. One important quality is to **avoid redundant data**, since redundant data can lead to data inconsistency. For instance, a Line2 in the alignment meta-model is uniquely described by a start and end point. In the context of the alignment meta-model we can define sets of properties sufficient to describe certain alignment elements unambiguously. For instance, for a line the properties *{Line2.Start, Line2.End}* are sufficient. Another possibility is to describe a line by the properties *{Line2.Start, Line2.Direction, Line2.Length}*. But in one and the same model a Line2 should never be described in more than one way in order to avoid redundancy.

Another quality to take into consideration is **query complexity**. It expresses the difficulties for implementation experts while querying the model for certain pieces of information. For instance, to get the end point of a line segment in LandXML we just have to query the corresponding XML element `LandXML/Alignents/Aligment/CoordGeom/Line/End`. In the alignment model proposed by Amann et al. 2013 it gets a bit more complicated. Here an *IfcTrimmedCurve* is used to store a line segment. *IfcTrimmedCurve* derives from a basis curve. For this basis curve an *IfcLine* is chosen. The *IfcLine* is defined by the start point, a direction vector and a magnitude value. Furthermore, an *IfcTrimmedCurve* contains two trim points that determine a line segment. Trim points can be expressed as Cartesian points or as real values. This example clarifies that in different alignment models an end point query can be more or less complex. Similar problems occur for other elements, such as arc segments. Since IFC provides different possibilities to describe certain elements and a user is not restricted to use a specific variant all different possibilities have to be considered. This means that it is more difficult to query some data – so the proposed *IfcAlignment* model has a higher query complexity than the LandXML model.

Additionally, one should not underestimate the corresponding **tool/technology support** of a certain model. The best product model is worth nothing in case of an insufficient tool support. For example, there are more tools available for XML than for STEP. Even this is not a core aspect of a model it also has to be considered. On the other hand, STEP files are much more easier to parse than XML files. XML is very heavyweight and the EXPRESS approach is very lightweight.

Most alignment models support either **relative stationing** or **absolute stationing**. Relative stationing implies that an alignment element is described referring to a specific element. Again, a line can be observed as a simple example. A line can be defined by its start point, its direction and length. This way, the end point must be computed. Moreover, the computed end point can be used as start point for the next horizontal alignment element. Instead of explicitly storing the start and end points of the corresponding alignment elements (as done in an absolute stationing model) the position can be implicitly computed from the relative movement.

Another design criterion is the so-called **domain mapping**, which indicates the models quality concerning the mapping of the domain. Does it use domain specific classes and terms like vertical alignment, horizontal alignment or curve types? For instance, we can assess how good the model does map to road construction.

Moreover, **internationalization support** should be considered. This comprises to consider if the concepts map only for one specific country or only support specific use cases, such as railway construction. Furthermore, the language describing the model is importance. For example, identifiers or documentation written in German language are useless in an international context.

The above described design guidelines can be used to assess and improve existing alignment models. Figure 4 shows an overview of the different design qualities.

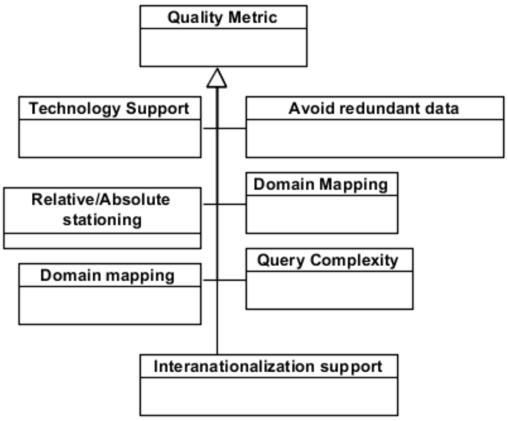

Figure 4. Overview of the different quality metrics.

4.4 *Problems of existing approaches*

Table 3 shows a small excerpt of a comparison table of different alignment product models that is used within the Building Smart P6 project which is responsible for development of IfcAlignment. The original table has many more alignment models included and compares many more items. Nevertheless the table shows that sometimes it is not clear which element from one standard can be mapped to which other elements, thus making it hard to compare different models and to makes the communication between different domain experts complicated.

Table 3. Overview of different alignment product models.

LandXML	Highway Object Model	OpenBrIM
Horizontal Alignment	Horizontal Alignment	Horizontal Alignment
Line	LineString	Straight (Line)
start station	startPoint,	StartStation
length	endPoint	Length
dir(ection)	-	StartAzimuth
(start,end)	-	
Easting/	-	
northing	-	
Spiral (clothoid)	(Spiral)	(Spiral) Curve
start station	-	StartStation
length	-	Length
-	start/end angle	StartAzimuth
radiusStart	start radius	-
radiusEnd	end radius	-
constant		-
(start, PI, end)	-	-
easting	startPoint,	-
northing	endpoint	-
rot(ation)		Direction

5 APPLYING QUALITY METRICS TO IMPROVE AN ALIGNMENT MODEL

5.1 *Reducing query complexity*

In (Amann et al. 2013) an IFC based alignment model has been proposed. Figure 5 represents an excerpt of the supposed description of horizontal alignment segments.

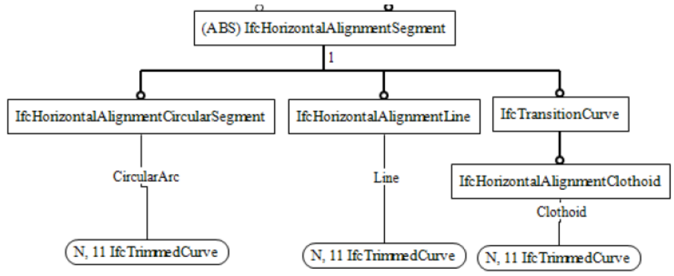

Figure 5. Horizontal alignment segments use and *IfcTrimmedCurve* (part of standard IFC 4) to describe its geometry

As described in Section 4.3 the query complexity is higher than in LandXML. To simplify the users' query process, the description of the different elements can be put directly into the corresponding elements. The horizontal alignment elements described in Amann et al. 2013 are shown in the following EXPRSS code segment:

```
ENTITY IfcHorizontalAlignmentLine
 SUBTYPE OF (IfcHorizontalAlignmentSegment);
  Line : IfcTrimmedCurve;
END_ENTITY;

ENTITY IfcHorizontalAlignmentCircularSegment
 SUBTYPE OF (IfcHorizontalAlignmentSegment);
  CircularArc : IfcTrimmedCurve;
END_ENTITY;

ENTITY IfcClothoid
 SUBTYPE OF (IfcCurve);
  Clothoid : IfcTrimmedCurve;
END_ENTITY;
```

To reduce query complexity the following changes can be made:

```
ENTITY IfcHorizontalAlignmentLine
 SUBTYPE OF (IfcHorizontalAlignmentSegment);
  Start : IfcCartesianPoint;
  End : IfcCartesianPoint;
END_ENTITY

ENTITY IfcHorizontalAlignmentCircularSegment
 SUBTYPE OF (IfcHorizontalAlignmentSegment);
  CircularArc : IfcTrimmedCurve;
END_ENTITY;

ENTITY IfcClothoid
 SUBTYPE OF (IfcCurve);
  Start : IfcCartesianPoint;
  Direction : IfcLengthMeasure;
  StartCurvature : IfcLengthMeasure;
  EndCurvature : IfcLengthMeasure;
  Length : IfcLengthMeasure;
END_ENTITY;
```

5.2 *Relative und absolute stationing*

RoadXML supports relative stationing. In the following example the horizontal alignment starts at the position (0, 20). The direction value contains the initial orientation. The line segment is 17.4926 units long, thus ends at the point (17.4926). This line segment is then followed by a clothoid transition curve and a circle arc.

```
<XYCurve direction="0" x="0" y="20">
  &lt;Segment length="17.4926"/>
  <ClothoArc endCurvature="0.00129562" length="50"
  startCurvature="-0.00285714"/>
  <CircleArc curvature="-0.0172478" length="25"/>
</XYCurve>
```

In order to extract the end position the corresponding end point of the specific clothoid has to be computed. The computed end point is then used as a start point for the circle arc. If there are many alignment elements numerical inaccuracies can lead to small differences in the computed coordinates of end point in different software applications. Thus, some people prefer absolute stationing, in which the end point is stored instead of the length and direction value. This way, the problem is shifted to the length parameter. Considering this example the length parameter would be computed and could have a small shifted value due to numerical inaccuracies.

Considering these two cases, one always has to ponder if the length value or the end point is more important. First of all, we could use an algebraic system in order to prevent these numerical inaccuracies, but a simpler solution is to support both approaches in one single model.

6 CONCLUSION

The qualities (design guidelines) described in this paper are not only restricted to alignment product models. The same or similar qualities can also be applied to other product models.

Besides, the alignment meta-model can be used for conversion between different data models. Assuming we have three different alignment models A, B, and C and want to be able to convert every model in to each other (A to B, A to C, B to C, B to A, C to A and C to B) we have to implement a specific converter for every possible conversion and, thus, for every converter, a domain expert is needed who is familiar with both alignment models. Writing a converter from A to the alignment meta-model and vice versa and repeating this for each alignment model we will end up with six converter implementations. The alignment meta-model pays off when we have more than 3 different alignment models since the implementation effort increases linear with the help of the meta-alignment model. Obviously, without meta-model the implementation effort increases dramatically. As demonstrated, the alignment meta-model in conjunction with the quality metrics can help to improve and evaluate different design variants concerning alignment product models.

The shown alignment meta-model is not finished right now. It needs to be extended to support further transition curve types. In addition, the vertical alignment needs to be included in the meta-model. Furthermore, a database of common constraints that describes how data can be automatically converted (i.e. radius = 1/curvature) between the different product models needs to be created.

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 MVD for LandXML v1.2 DRAFT 0.5, 2011. http://cic.vtt.fi/bSI_LandXML12_MVD/index.htm

buildingSMART and InfraFINBIM Nordic Workshop, Helsinki 30.11.2011

Chaplier, J., That, T. N., Hewatt, M., & Gallée, G. 2010. *Toward a standard: RoadXML, the road network database format* At the proceedings of the Driving Simulation Conference, pp. 211-220.

Hyvärinen J., 2011. Finnish InfraBIM Activities & Finnish Inframodel http://www.rts.fi/workshop20120507/Classifications_and_BIM_workshop_InfraBIM_Inframodel_20120507_JH.pdf

Liebich, T. 2013. *Alignment Data Schema* (openINFRA proposal - Alignment Data Schema, received 07/05)

OpenBrIM, http://openbrim.org/Schema2.aspx

Kim, H. 2013. *Highway Object Model & Object Based 3D Intelligent Model for Construction Planning/Simulation in a Highway Construction*

Rebolj, D., Tibaut, A., Čuš-Babič, N., Magdič, A., & Podbreznik, P. 2008. *Development and application of a road product model*. In: Automation in Construction, Volume 17, pp. 719-728.

Scarponcini, P. 2013. *InfraGML Proposal (13-121)*, OGC Land and Infrastructure DWG/SWG.

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.*