A critical analysis of linear placement in IFC models

A critical analysis of linear placement in IFC models

Š. Jaud, S. Esser, A. Borrmann
Chair of Computational Modelling and Simulation, Technical University of Munich, Munich, Germany

L. Wikström
Triona SE, Stockholm, Sweden

S. Muhič
buildingSMART International Ltd, Kings House, Station Road, Kings Langley, Hertfordshire, UK

J. Mirtschin
Geometry Gym Pty Ltd., Port Fairy, Australia

ABSTRACT: The open data exchange standard Industry Foundation Classes (IFC) has been recently significantly extended to also cover infrastructure facilities such as roads and railways. The results of these activities form part of version 4.3 of the standard. Linear placement of objects is one of the most important concepts in infrastructure asset modelling. As such, Release Candidate 1 of IFC 4.3 has been critically analysed. In this paper, we address several issues that were identified together with the participants of the IFC Infrastructure Extension Deployment and IFC Rail Phase 2 projects. We present an improved model removing unnecessary doubling of concepts and reusing many already established entities. We showcase the new model on two example scenarios from one of the projects and determine better adherence with the IFC legacy. The proposed improvements have been adopted in Release Candidate 2 of the IFC 4.3.

1 INTRODUCTION

1.1 *Background*

Building Information Modelling (BIM) is being increasingly implemented in the infrastructure sector within the Architecture, Engineering, and Construction (AEC) domain (Bradley et al. 2016). This calls for the peculiarities common to the infrastructure domain to be introduced to the established workflows, processes, and data models previously only focusing on the building sector (Borrmann et al. 2019).

One of the most important pieces of information about any AEC object is its location and orientation in the three-dimensional (3D) space – the placement of its coordinate system (CS) within some geometric context. This is usually modelled by providing the object's Cartesian coordinates and rotation angles in the geospatial context or relatively to another object's placement (Jaud et al. 2020a).

Infrastructure assets are not residing on small parcels but rather span multiple kilometres connecting cities and industry across the globe. As such, the notion of linear placement has been introduced to specify the position of objects along a linear axis, as opposed to providing the Cartesian orthogonal coordinates. Such an axis is usually defined in the global CS and is declared a (curvilinear) coordinate axis with the stationing coordinate uniquely denoting locations along it. This unique concept has been standardized by the International Standardization Organization (ISO) in ISO 19148 (2012).

For example, consider the two-dimensional (2D) placement of the stations' local CSs $(x,y)_i$ along the railway line as shown in Figure 1. These CSs can be expressed by specifying the Cartesian position of each origin and axes' rotations in the global CS (X,Y) as is usually the case in the building domain.

However, more naturally to the infrastructure domain, first the railway's main curve is defined within the global CS (X,Y). This curve in turn defines the curvilinear axis of the linear CS. The stations' CSs $(x,y)_i$ can then be placed in the curvilinear CS with (i) their stationing coordinate representing the distance along the curvilinear axis, (ii) offsets from the main curve, and (iii) rotation of the axes defined. Note that the example is shown in the 2D plane, but it is similarly valid in the 3D space.

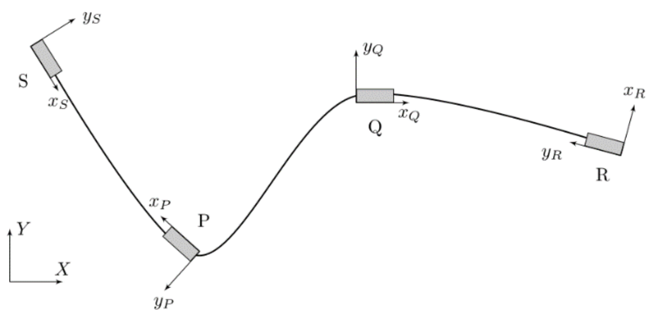

Figure 1. Railway track (black line) defines a curvilinear CS. The four stations (SPQR) have their CSs defined in this CS with their stationing coordinate (distance along the line), their offsets from the line, and rotations of CS's axes (Jaud et al. 2020a).

1.2 *Problem Statement*

It is clear that data models need to integrate the concept of linear placement to support infrastructure workflows. In this paper, we take a closer look at one of the most prominent vendor-neutral data schemas Industry Foundation Classes (IFC) defined in ISO 16739 (2018). We critically look at how this concept has been introduced in the frame of the recent extensions and explore alternatives.

1.3 *Methodology*

We follow the Design Science Research methodology (Peffers et al. 2007).

1. Problem identification and motivation (Section 1.1).
2. Definition of objectives for a solution (Section 4.1).
3. Design and development of artefacts solving the problem (Section 4.2).
4. Demonstration of suitability of developed artefacts to solve the problem (Section 5).
5. Evaluation of the solution by comparing the objectives and obtained results (Section 6).
6. Communication of the obtained results (this paper).

1.4 *Structure of the Paper*

The paper is structured as follows. Section 1 presents our motivation, the problem statement, and our methodology. Section 2 briefly summarizes related work. Section 3 presents the necessary theoretical background, describes the IFC standard, and the modelling of placement concepts for the problem at hand. Section 4 lists the issues of the current IFC model and describes a new solution. Section 5 showcases its application to two case studies. Section 6 concludes the paper with a brief discussion.

2 RELATED WORKS

The concept of linear placement has been thoroughly addressed by the Open Geospatial Consortium (OGC) and standardized in ISO 19148 (2012). It specifies a conceptual schema for locations relative to a one-dimensional (1D) object as measurement along (and optionally offset from) that object. The specification is implemented in the Geography Markup Language 3.3 (GML) standard (OGC, 2012).

There are three central concepts for specifying a linear position defined: (a) the linear element being measured along, (b) the measure values (captured by a distance expression) specifying the distance along and optionally offset from the linear element, and (c) the method of measurement. Furthermore, in the case where the linear element is a curve in the 2D or 3D Cartesian space, the standard defines the relationship between a linear position and a point in the Cartesian space of the curve with the interface *LR_ISpatial*. This interface defines two functions that allow the transformation between the linear position and the Cartesian position, which must be implemented by the linear element (ISO 19148, 2012).

During recent years, projects conducted by buildingSMART International (bSI) have expanded the well-established IFC data model for infrastructure facilities. The concept of linear placement has been tackled by the IFC Alignment project and a first proposal was included in version IFC4x1 (Liebich et al. 2017). These definitions were reused and built upon by subsequent projects, e.g. IFC-Bridge (Borrmann et al. 2019). The definitions were successfully used by Esser & Borrmann (2019), who converted data from PlanPro and RailML data models into IFC and place railway signals along a railway axis.

The currently running project "IFC Infrastructure Extension Deployment" (Jaud et al. 2020b) has brought inconsistencies in the modelling of linear placement within IFC to light. Additionally, multiple issues have been reported by the software vendors participating in the project. These issues and their resolutions are addressed in this paper.

3 THEORETICAL BACKGROUND

Since the AEC industry operates in the 3D world, we limit our consideration to the 3D Euclidean space $\mathbb{R}^3$. Additionally, we only consider right-handed, orthogonal CSs. We assume that the engineering CS of the project has an already defined relationship with the geospatial context as described by Jaud et al. (2020a). Thus, there exists a global engineering CS (X,Y,Z) within which all other placements reside.

3.1 *Placement*

The main purpose of placement is to specify the position and orientation of a CS relative to another CS, i.e. to define the relationship between the *origin* and *target* CSs. In this sense, the *origin* CS describes the local CS of the object which is placed within a global CS. For easier notation, we define (x,y,z) as the coordinate axes of the origin CS and (u,v,w) as the coordinate axes of the target CS. The transformation function f connecting these CSs needs a clear definition:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) . \tag{1}$$

Following Equation 1, $[u_O,v_O,w_O]^T = f([0,0,0]^T)$ are the coordinates of origin CS's Point of Origin (PoO) in the target CS.

Additionally, we define $g(t)$ as a C^0 continuous curve in the target CS:

$$g(t) = a + bu(t) + cv(t) + dw(t)\,, \tag{2}$$

with parameter $t \in \mathbb{R}$ uniquely determining a point on the curve and $\{a, b, c, d\} \in \mathbb{R}$.

We explore three major possibilities of placement as used in the AEC domains in the next subsections.

3.1.1 *Orthogonal Placement*

The so-called 7-parameter Helmert transformation $\{u_O, v_O, w_O, \alpha, \beta, \gamma, \lambda\}$ connects two Cartesian CSs as shown in Figure 2 (Jaud et al. 2020a):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u_O \\ v_O \\ w_O \end{bmatrix} + \lambda R(\alpha, \beta, \gamma) \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{3}$$

where λ denotes the factor between the Unit of Measurements (UoM) of both CSs, and R defined as:

$$R(\alpha,\beta,\gamma) = \begin{bmatrix} c\gamma\, c\beta & c\gamma\, s\beta\, s\alpha + s\gamma\, c\alpha & -c\gamma\, s\beta\, c\alpha + s\gamma\, s\alpha \\ -s\gamma\, c\beta & -s\gamma\, s\beta\, s\alpha + c\gamma\, c\alpha & s\gamma\, s\beta\, c\alpha + c\gamma\, s\alpha \\ s\beta & -c\beta\, s\alpha & c\beta\, c\alpha \end{bmatrix} \tag{4}$$

where *c* and *s* stand for *cos* and *sin*, respectively.

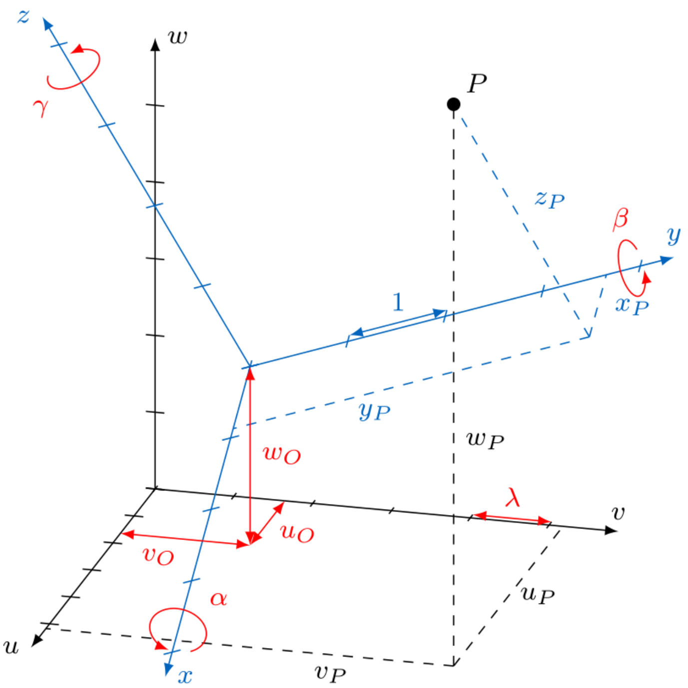

Figure 2. Visualization of the Helmert transformation between two Cartesian CSs from Equation 3 (Jaud et al. 2020a). The 7 parameters are shown in red.

3.1.2 *Grid Placement*

The grid placement is usually used in architecture, where the columns are placed on a grid-like arrangement throughout the building as shown in Figure 3.

The grid is represented with an array of non-colinear curves $g_i(t)$. A chosen intersection of two curves $g_i(t)$ and $g_j(t)$ with $i \neq j$ defines the origin CS's PoO and the transformation follows Equation 3.

3.1.3 *Linear Placement*

The linear placement is a commonly used method for specifying the location and orientation of objects within the realm of *long* infrastructure assets such as roads and railways. Here, objects are placed relative to the main alignment of the road or railway, which represents a curvilinear coordinate axis as shown on Figure 4. This axis fulfils the requirements for a linear element as defined in ISO 19148 (2012).

The origin CS's PoO is defined in a chosen point along the curve with the transformation following Equations 2 and 3.

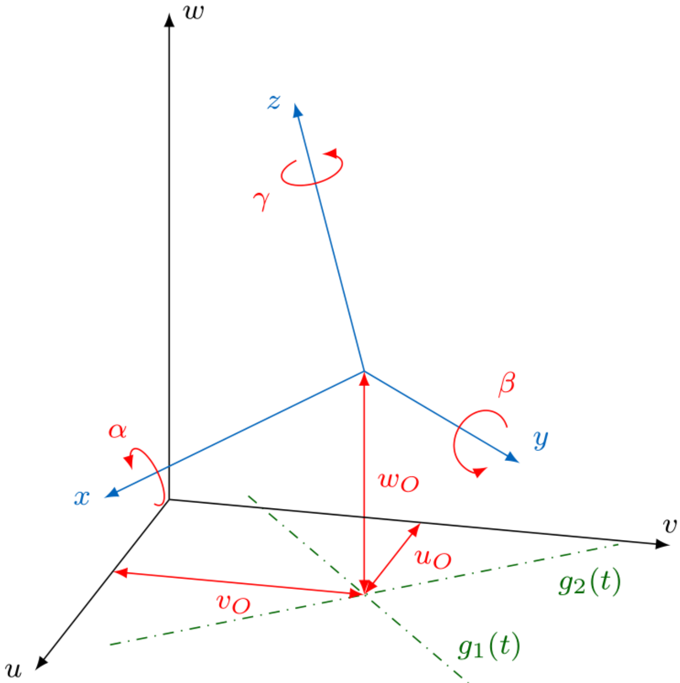

Figure 3. Visualization of the parameters of grid placement as usually defined in AEC domain: the grid axes are defined in $w = 0$ plane, i.e. $w_1(t) = w_2(t) = 0$, with w_O being defined separately to define 3D location.

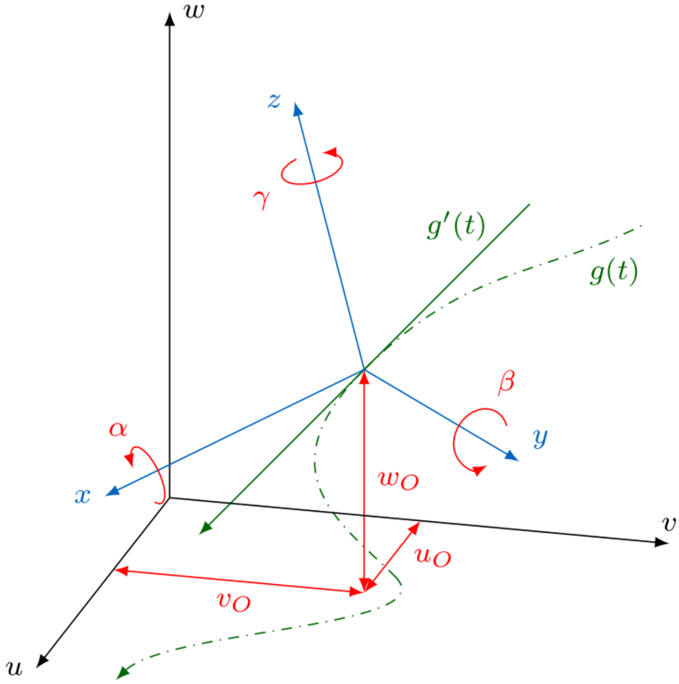

Figure 4. Visualization of the parameters of linear placement: $g(t)$ represents the curvilinear axis in 3D space along which a location can be uniquely defined with parameter t. Additionally, the tangent $g'(t)$ at the chosen t is exemplary shown.

3.2 *IFC Data Model*

This section presents the concepts from Section 3.1 as they are incorporated in the *IFC 4.3 Release Candidate 1* (IFC4x3_RC1) published by bSI (bSI, 2020a). This version has been superseded by the *Release Candidate 2* (IFC4x3_RC2), which includes the changes as reported in this paper (bSI, 2020b).

For easier reading, we adopt *CamelCase* notation for IFC entities. The diagrams are drawn with EXPRESS-G notation as specified in ISO 10303-11, Annex D (ISO 10303-11, 2004).

3.2.1 *Geometric Representation*

Figure 5 shows a diagram of the most important geometric entities used for implementing the concept of placement in IFC. Base entity for any geometry in IFC is the *IfcGeometricRepresentationItem.* From it, entities for points, curves, surfaces, and solids are derived as *IfcPoint*, *IfcCurve*, *IfcSurface*, and *IfcSolid*, respectively, with their subtypes.

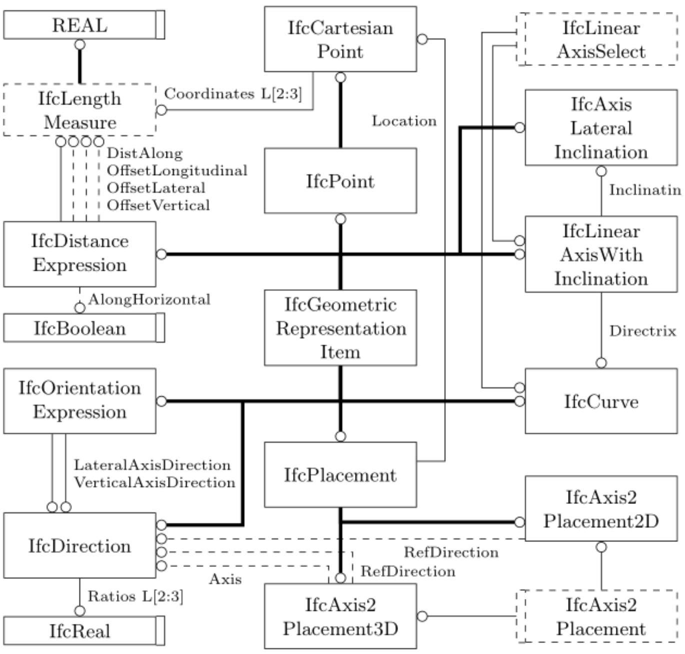

Figure 5. EXPRESS-G diagram of geometric representation entities from IFC4x3_RC1 that are important to our study.

The abstract super class *IfcPlacement* encapsulates the semantics of placement as stated in Section 3.1. It provides (a) the coordinates of the PoO in its *Location* attribute, and (b) the orientation of the CS axes through its derived classes' attributes *RefDirection* and *Axis* (Figure 5, bottom).

In order to support the notion of specifying a point along a curve, *IfcDistanceExpression* was introduced in IFC4x1 (see requirement (b) from ISO 19148 described above). The entity also enables to optionally offset the point from the curve with longitudinal, lateral and vertical offsets as presented in Figure 6.

Additionally, *IfcOrientationExpression* was introduced to define the orientation of a CS relative to a curve with attributes *LateralAxisDirection* and *VerticalAxisDirection* (Figure 5, left).

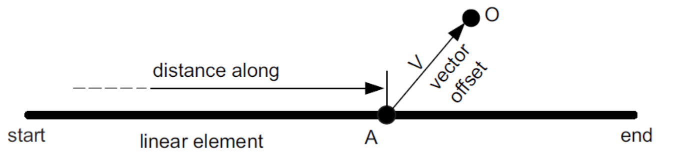

Figure 6. One way of determining a point along a linear element with distance along and an offset vector (ISO, 2012).

As it is usually required for railway engineering, the main axis incorporates lateral inclination in its definition (also called cant). This is modelled in IFC with *IfcLinearAxisWithInclination* that references a curve *IfcCurve* together with the inclination profile *IfcAxisLateralInclination* (see Figure 5, right).

3.2.2 *Object Placement*

Each *IfcProduct*'s geometry (e.g. walls, columns, courses, etc.) can be placed in the geometric context by specifying its *ObjectPlacement* attribute. For this, an entity of type *IfcObjectPlacement* must be used (see Figure 7 and Figure 8, top right).

There are three deriving entities, each encapsulating one of the placement possibilities as described in subsections 3.1.1-3.1.3. A placement using Cartesian coordinates as shown in Figure 2 is modelled as *IfcLocalPlacement*, which allows for 2D or 3D local placement (Figure 7, right). Its attribute *RelativePlacement* conveys the coordinates of the origin CS's PoO as well as its orientation in space. The scale factor remains 1 as all *IfcLengthMeasure* instances within one IFC file have the same UoM specified globally.

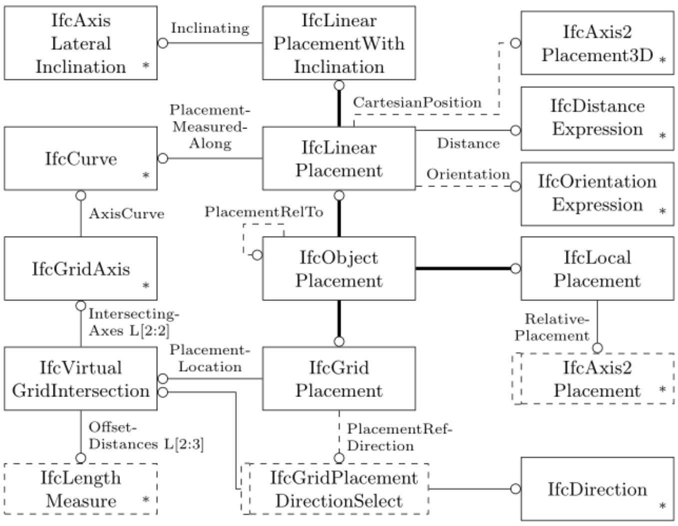

Figure 7. EXPRESS-G diagram of object placement entities from IFC4x3_RC1 that are important to our study.

The grid placement is modelled as *IfcGridPlacement* (Figure 7, bottom) which allows for specifying the 3D location, but only one rotation, i.e. around z-axis ($\alpha = \beta = 0$). The PoO is specified by an intersection point (*IfcVirtualGridIntersection*) of two grid axes (*IfcGridAxis*). The x-axis of the origin CS is defined with the direction from the PoO to another intersection point or a direction in the grid's context.

Linear placement is modelled with *IfcLinearPlacement* (Figure 7, top) which combines an *IfcCurve* as the curvilinear axis together with an *IfcDistanceExpression* to obtain a unique point on the axis with optional offsets from the curve resulting in the PoO. The orientation of the CS is determined using an attribute of type *IfcOrientationExpression*. The deriving *IfcLinearPlacementWithInclination* allows to account for the lateral inclination (i.e., cant) of the curvilinear axis supporting the railway domain.

All placements have an optional attribute *PlacementRelTo*, which allows to chain placements and thus establish their intertwined dependencies.

3.2.3 *Positioning Element*

IFC models the geometric context of positioning elements with the entity *IfcPositioningElement* and its derivates (Figure 8, top center). The data model includes a constraint (a WHERE rule) on the inherited optional attribute *ObjectPlacement* that makes it non-optional. Thus, it is ensured that a positioning element always has an assigned placement within the geometric context of the model.

The *IfcGrid* comprises 2 or 3 lists of unique grid axes used for positioning in *IfcGridPlacement*: *UAxes*, *VAxes*, and optional *WAxes*. These are then referenced by the placement entities as described in Section 3.2.2.

The *IfcReferent* can be used a placeholder for additional information along a linear element. For example, a stationing jump can be modelled using the optional *RestartDistance* attribute.

The *IfcLinearPositioningElement* (and derived *IfcAlignment*) points through its attribute *Axis* to the curve along which one can linearly position other objects.

The relationship *IfcRelPositions* models a connection between products and a positioning element (Figure 8, top left). This is useful if the exact placement of those products (and thus the dependency) cannot yet be expressed with *IfcObjectPlacement*.

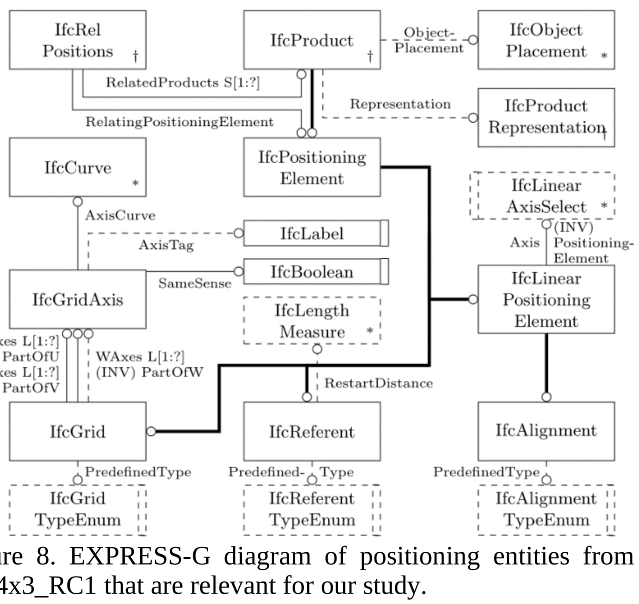

Figure 8. EXPRESS-G diagram of positioning entities from IFC4x3_RC1 that are relevant for our study.

4 NEW PROPOSAL

In the course of the deployment activities of IFC Infrastructure Extension Deployment and IFC Rail Phase 2 projects (Jaud et al. 2020b), a multitude of problems and issues were reported to the project teams. Among these were questions and concerns about the linear placement concept, which have been included in our study.

This section describes the problems identified in the model and explores the solution as included in the IFC4x3_RC2 standard (bSI, 2020b).

4.1 *Problem Identification*

We identified four major data model issues and additional requirements with the linear placement concept as described in Section 3.2.

4.1.1 *Reuse of Existing Concepts*

Borrmann et al. (2017) mandate reuse of existing IFC entities wherever possible when expanding IFC for infrastructure. This prohibits unnecessary duplication of concepts, which would result in redundancy. To put it differently, it omits an increased burden on software vendors when updating their existing IFC interface implementations.

However, clear parallels can be observed in Figure 5 between the left-most and central columns. *IfcDistanceExpression* defines a point similarly to *IfcCartesianPoint*, however it is not inherited from *IfcPoint*. Similarly, *IfcOrientationExpression* defines the orientation of the CS axes but it is not inherited from *IfcPlacement*. The introduction of both entities is violating the recommendations provided by Borrmann et al. (2017).

4.1.2 *Method of Measurement*

ISO 19148 requires a linear placement to define (a) a linear element, (b) a measure value, and (c) a method of measurement (see Section 2). While requirements (a) and (b) are successfully addressed by the attributes of *IfcLinearPlacement*, (c) is implicitly assumed to be an *IfcLengthMeasure*.

The IFC standard foresees that each *IfcCurve* has its parameterization clearly defined and successfully uses this principle in entities such as *IfcPointOnCurve* and *IfcTrimmedCurve*. To uniquely specify the position, *IfcParameterValue* is used which bases the method of measurement on the type of the underlying curve (e.g. [0,1] for straight lines and $[0,2\pi]$ for circles). As such, the implicit assumption mentioned above violates the already established norm in IFC.

4.1.3 *Tangent Reference*

The curve used for linear placement is required to be C^0 continuous (Equation 2). This means that a tangent in places without C^1 continuity cannot be uniquely determined and as such the directions of offsets from *IfcDistanceExpression* remain undefined.

Additionally, the CS in which *IfcOrientationExpression* is expressed is unclear. The directions provided do not specify which axes they define, e.g. does the *LateralAxisDirection* attribute define the *x*, *y* or *z* axis direction of the origin's CS?

4.1.4 *Chaining of Placements*

IfcObjectPlacement allows for chaining of placements using the *ObjectPlacement* attribute. The implementation is clear for the case when an instance of *IfcLocalPlacement* is placed relative to another *IfcLocalPlacement* (see Figure 2 and Equation 3).

However, what does it mean, if an *IfcLinearPlacement* or an *IfcGridPlacement* is positioned relative to another *IfcObjectPlacement*? This is especially critical when considering that both mentioned placements need to account for the geometric context of the *IfcPositioningElement* their curves belong to.

4.2 *Solution*

This section presents the developed solution to the problems specified in Section 4.1. The solution has been adopted by bSI and is implemented in the Release Candidate 2 of the 4.3 version (IFC4x3_RC2). Figure 9 shows an overview of the changes, where changes are marked with red colour and deprecated entities omitted.

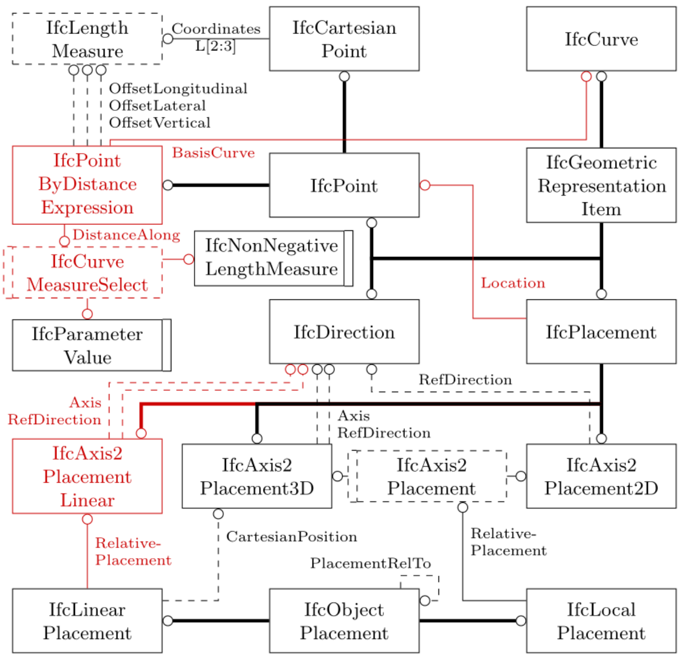

Figure 9. EXPRESS-G diagram of changed entities from Figures 5 & 7 marked in red. *IfcGridPlacement* and its attributes not shown for brevity, since nothing changed.

Firstly, the *IfcDistanceExpression* was renamed to *IfcPointByDistanceExpression* and is now inherited from the abstract *IfcPoint* entity. A new attribute *BasisCurve* was introduced to convey along which curve the location is to be measured along. This resulted in the removal of the *PlacementMeasuredAlong* attribute of *IfcLinearPlacement*. The offsets are clearly defined relative to the tangent of the curve at the location specified. This partly addresses the issues from Sections 4.1.1 and 4.1.3.

Additionally, the type of *DistanceAlong* attribute was changed to a new select type *IfcCurveMeasureSelect*. This allows for specifying the measurement method according to (a) the established *IfcParameterValue* with the IFC model or (b) with an absolute length from the beginning of the curve *IfcNonNegativeLengthMeasure* as is usually the case with infrastructure assets. This solves the problem emphasized in Section 4.1.2.

Secondly, a new entity *IfcAxis2PlacementLinear* replaces *IfcOrientationExpression* and derives from *IfcPlacement*. The *Location* attribute of *IfcPlacement* previously pointing to an instance of *IfcCartesianPoint* now allows to reference any *IfcPoint*, thus ensuring that *IfcAxis2PlacementLinear* entities can reference the introduced *IfcPointByDistanceExpression* mentioned above. This is enforced by using a special WHERE rule on *IfcAxis2PlacementLinear*. Consequently, *IfcLinearPlacement* is provided with the necessary semantics to establish a linear placement as explained in Section 2. This partly addresses the issues from Section 4.1.1.

Thirdly, the definition of *Axis* and *RefDirection* attributes of *IfcAxis2PlacementLinear* state that they are defined relative to the tangent of the curve at the specified location. This ensures unambiguity as demanded by Section 4.1.3.

Lastly, a restriction on *IfcObjectPlacement* was introduced, where an instance of *IfcLinearPlacement* cannot be placed relative to another *IfcLinearPlacement*. Rather, only the *IfcObjectPlacement* used by the corresponding *IfcLinearPositioningElement* can be referenced in *PlacementRelTo* attribute of *IfcLinearPlacement*. This addresses the issue described in Section 4.1.4. Consequently, the retrieval of the context of an *IfcGridPlacement* has been significantly simplified. It is not anymore necessary to navigate through *IfcGridAxis* inverse attributes *PartOfU*, *PartOfV* and *PartOfW* to obtain the geometric context, as it is specified in *PlacementRelTo* attribute.

In consequence of the changes described, the entities defining inclination were removed as the changes to *IfcPlacement* now allow the specification of inclination angles directly in the context of an *IfcCurve*. The entities *IfcAxisLateralInclination*, *IfcLinearAxisWithInclination* and *IfcLinearPlacementWithInclination* are deprecated in IFC4x3_RC2 with planned removal in the final version of the standard.

Note that other elements not explicitly mentioned in this section from Figures 5, 7 & 8 were left unchanged (like *IfcGridPlacement*). The only exception is the *Axis* attribute of *IfcLinearPositioningElement* whose type got reversed to be *IfcCurve* as it was defined in IFC4x1 by Liebich et al. (2017).

5 CASE STUDIES

We tested our proposal on two unit-test sample files made available by the IFC Infrastructure Extensions Deployment project (Jaud et al. 2020b). The Step Physical Files (SPF) following ISO 10303-21 (2016) with supporting documentation and screen dumps can be obtained from the project's official repository: www.github.com/bSI-InfraRoom/IFC-infra-unit-test

5.1 *Railway Sleepers*

The first example consists of ten railway sleepers linearly aligned along and rotating about the axis as seen on Figure 10. Each is slightly more rotated in a clockwise direction than the previous. This example showcases both (i) the location along the alignment as well as (ii) the modelling of different orientation contexts. The SPF file consists of three major blocks: (a) project context with default units, (b) an alignment whose axis consists of a single linear segment, and (c) multiple instances of *IfcBuiltElement* modelling the sleepers with their location and orientation specified w.r.t. the given alignment axis.

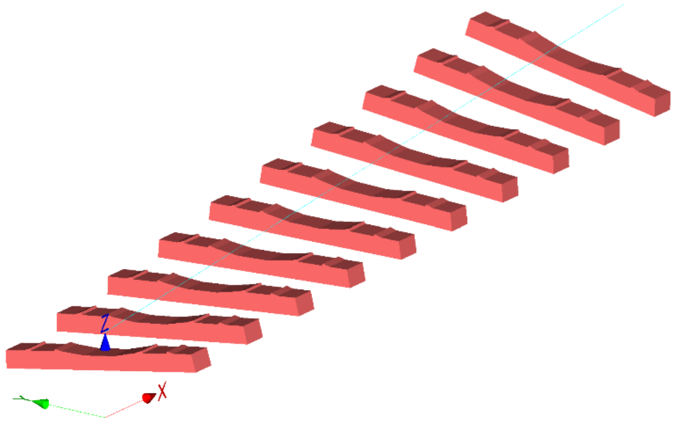

Figure 10. Railway sleepers linearly placed along a straight line with changing inclination values.

We provide a closer look at a linear placement of an individual sleeper in Algorithms 1-3. Algorithm 1 shows an *IfcBuiltElement* instance modelling a sleeper. Each sleeper references the SPF reference *#277* to define its geometrical representation. and is placed in its correct location with the SPF reference *#40* (see also Figure 8, top right).

Algorithm 1. An excerpt from an SPF defining a semantic object for a sleeper together with its geometry from Figure 10. The placement reference *#40* is modelled in Algorithms 2 & 3.

```
#38=IFCBUILTELEMENT('0QLu06Q0LBIfiMIDl4KKna',
     #1002,'linear positioned: 1',$,$,#46,#40,$,$);
#40=IFCPRODUCTDEFINITIONSHAPE($,$,(#277));
#277= IFCSHAPEREPRESENTATION(#15,'Body','Brep',
      (#651));
#651= IFCFACETEDBREP(#647);
#647= IFCCLOSEDSHELL((…)); //shortened
```

The Algorithms 2 & 3 show the most important lines concerning the linear placement of the sleeper. They showcase the proposed changes between the definitions from Sections 3.2 and 4.2 (i.e. IFC4x3_RC1 and IFC4x3_RC2). In both cases, the SPF reference *#20* points to the instance of the alignment axis. The SPF reference *#46* is consumed by an instance of *IfcBuiltElement* as its *ObjectPlacement* (see Algorithm 1).

Algorithm 2. An excerpt from an SPF of a linear placement for a sleeper from Figure 10 following the definitions from Section 3.2 (i.e. following IFC4x3_RC1).

```
#42=IFCDISTANCEEXPRESSION(1.,0.,-0.2,0.,.T.);
#43=IFCDIRECTION((0.,0.1361,0.9907));
#44=IFCDIRECTION((1.,0.,0.));
#45=IFCORIENTATIONEXPRESSION(#44,#43);
#46=IFCLINEARPLACEMENT(#20,#42,#45,$);
```

Algorithm 3. An excerpt from an SPF of a linear placement for a sleeper from Figure 10 following the definitions from Section 4.2 (i.e. following IFC4x3_RC2).

```
#42=IFCPOINTBYDISTANCEEXPRESSION(IFCNON
     NEGATIVELENGTHMEASURE(1.),0.,-0.2,0.,#20);
#43=IFCDIRECTION((0.,0.1361,0.9907));
#44=IFCDIRECTION((1.,0.,0.));
#45=IFCAXIS2PLACEMENTLINEAR(#42,#43,#44);
#46=IFCLINEARPLACEMENT($,$,$,#45,$);
```

5.2 *Road Signs*

The second example models chevron signs positioned on the outsides of horizontal curves of a typical road axis as shown on Figure 11. While the first example showcases rotation of an element about the main axis, the second example models the placement of an object relative to the axis, being arbitrarily rotated and offset from the axis.

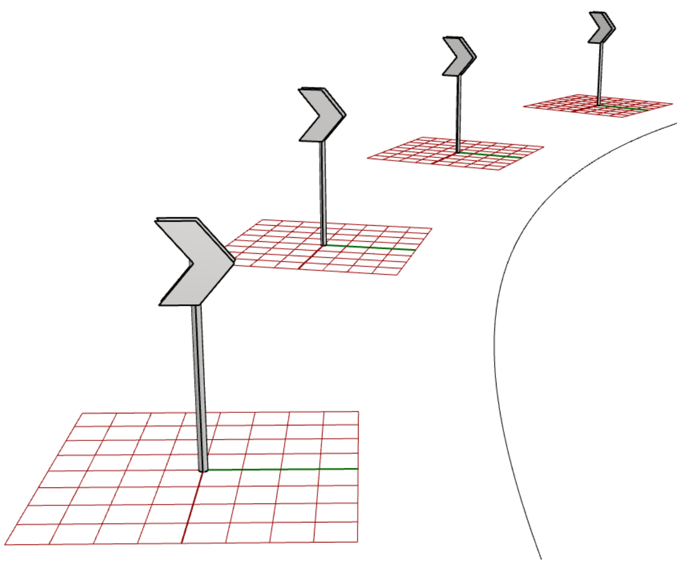

Figure 11. Road signs linearly placed along and perpendicular to the main axis in a curved segment. The horizontal (x,y) planes of their individual local CSs are shown as grids.

The individual signs are modelled as *IfcElementAssembly* instances containing the post and chevron parts positioned locally in the context of the sign (see Algorithm 4). Their geometries are defined once and reused for each sign using *IfcMappedItem*. The position of an individual sign is modelled with *IfcLinearPlacement* with an optional additional *IfcLocalPlacement* instance to rotate the sign to always point towards the curve when positioned on the outside of the axis' curves.

Algorithm 4. A tree view of IFC elements and their attributes representing a sign from Figure 11 (cropped on the right side).

```
⊟ #280= IFCELEMENTASSEMBLY('3ZsRKiJhpmC0lFpJHy_xn3',$,$,$,$,#281,$,
  ⊟ Placement : #281= IFCLOCALPLACEMENT(#279,#178);
    ⊟ PlacementRelTo : #279= IFCLINEARPLACEMENT(#4,$,$,#278,$);
      ⊞ PlacementRelTo : #4= IFCLOCALPLACEMENT($,#5);
      ⊟ RelativePlacement : #278= IFCAXIS2PLACEMENTLINEAR(#277,#1
        ⊟ Location : #277= IFCPOINTBYDISTANCEEXPRESSION(IFCNON
          ⊞ BasisCurve : #110= IFCGRADIENTCURVE(#14,(#111,#115,#1
        └ Axis : #155= IFCDIRECTION((0.0,0.0,1.0));
    ⊟ RelativePlacement : #178= IFCAXIS2PLACEMENT3D(#164,#155,#17
      ├ Location : #164= IFCCARTESIANPOINT((0.0,0.0,0.0));
      ├ Axis : #155= IFCDIRECTION((0.0,0.0,1.0));
      └ RefDirection : #174= IFCDIRECTION((-1.0,0.0,0.0));
  ⊟ IsDecomposedBy(3)
    ⊞ #283= IFCMEMBER('3HDs_AqgUGMyOGXeseK_OG',$,$,$,$,#282,#18
    ⊞ #286= IFCSIGN('1EMCzoWtuwgvPd0tNVSEdl',$,$,$,$,#285,#187,$,$)
    ⊞ #288= IFCSIGN('0cmjprzXqvhcLurRfHz0Xt',$,$,$,$,#287,#187,$,$);
```

6 CONCLUSION

We have critically evaluated a part of the recent candidate standard extension of IFC (IFC4x3_RC1) as published by bSI (2020a). With the help of participants of the IFC Infrastructure Extension Deployment and IFC Rail Phase 2 projects (Jaud et al. 2020b), several issues about the linear placement concept have been identified as presented in Section 4.1. The proposed simplified model from Section 4.2 addresses the requirements listed while respecting the guidelines provided by Borrmann et al. (2017). The improved model has been adopted as RC2 of IFC4.3.

We showcase the new model on two example scenarios from the projects mentioned above. We argue that the proposed model shown in Figure 9 enables every foreseeable constellation of placement as occurring in the AEC industry – in the building as well as infrastructure sectors. One can observe that the newly introduced inheritance from *IfcPoint* and *IfcPlacement* for *IfcPointByDistanceExpression* and *IfcAxis2PlacementLinear*, respectively, allow for modular software architecture.

The reuse of existing concepts ensures little-to-no effort required by software vendors already supporting these concept templates of the IFC schema. We call for fast adoption of the newly developed IFC4x3_RC2 standard by the industry and active participation of all IFC implementors with the deployment activities.

REFERENCES

Borrmann, A., Amann, J., Chipman, T., Hyvärinen, J., Liebich, T., Muhič, S., Mol, L., Plume, J. & Scarponcini, P. 2017. *IFC Infra Overall Architecture Project: Documentation and Guidelines.* Technical Report. buildingSMART International.

Borrmann, A, Muhič, S., Hyvärinen, J., Chipman, T., Jaud, Š., Castaing, C., Dumoulin, C., Liebich, T., & Mol, L. 2019. *The IFC-Bridge Project – Extending the IFC Standard to Enable High-Quality Exchange of Bridge Information Models.* 2019 European Conference on Computing in Construction, Chania, Crete, Greece.

Bradley, A.; Li, H.; Lark, R.; & Dunn, S. 2016. BIM for infrastructure: An overall review and constructor perspective. *Automation in Construction*. 71, 139–152. DOI: 10.1016/j.autcon.2016.08.019.

bSI 2020a. *Industry Foundation Classes: 4x3 candidate standard 1 documentation*. Online, URL: https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC1/HTML/, accessed 2020-12-22.

bSI 2020b. *Industry Foundation Classes: 4x3 candidate standard 2 documentation*. Online, URL: https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC2/HTML/, accessed 2020-12-22.

Esser, S.; Borrmann, A. 2019 *Integrating Railway Subdomain-Specific Data Standards into a common IFC-based Data Model.* In: Proc. of the 26th International Workshop on Intelligent Computing in Engineering, Leuven, Belgium.

ISO 2004. *ISO 10303-11:2004: Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual.* Standard International Organization for Standardization, Geneva, Switzerland.

ISO 2012. *ISO 19148:2012: Geographic information – Linear referencing.* Standard International Organization for Standardization, Geneva, Switzerland.

ISO 2016. *ISO 10303-21:2016: Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure.* Standard International Organization for Standardization, Geneva, Switzerland.

ISO 2018. *ISO 16739:2018-1: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. Standard International Organization for Standardization, Geneva, Switzerland.

Jaud, Š., Donaubauer, A., Heunecke, O. & Borrmann, A. 2020a. Georeferencing in the context of building information modelling. *Automation in Construction, 118(103211).* DOI: 10.1016/j.autcon.2020.103211

Jaud, Š., Esser, S., Muhič, S. & Borrmann, A. 2020b. Development of IFC Schema for Infrastructure. In: *Proceedings of 6th international conference siBIM: Structured data is the new gold*, pp.27-35. Online.

Liebich, T., Amann, J., Borrmann, A., Chipman, T., Hyvärinen, J., Muhič, S., Mol, L., Plume, J., Scarponcini, P. 2017. *IFC Alignment 1.1 Project: IFC Schema Extension Proposal.* Technical Report. buildingSMART International.

OGC 2012. *Geography Markup Language (GML) — Extended schemas and encoding rules*. OpenGIS Implementation Standard. OGC 10-129r1.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. 2007. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.