TECHNISCHE UNIVERSITÄT MÜNCHEN
Chair of Computational Modeling and Simulation

Technical Documentation

A multi-scale product model for shield tunnels based on the Industry Foundation Classes

Javier Ramos Jubierre
André Borrmann

Technische Universität München

Version: 5.09.2014

Changes

Version	Data	Writer(s)	Adds to the previous version
0.0.1	11/06/2014	Javier Jubierre	First draft
0.0.2	15/07/2014	Javier Jubierre	Second draft
1.0.0	05/09/2014	Javier Jubierre	First Version

Section 1. Shield Tunnel Product Model

1.1. Introduction

This document present a proposal for a shield tunnel product model based on the standardized data model Industry Foundation Classes (IFC), as introduced in (Borrmann et al. 2014). In particular, the proposed extension introduces the integration of the concept of multiple levels-of-detail into the IFC standard.

Currently the IFC does not support the exchange of models with different levels-of-detail. However, the IFC model implements the important principle of a strict separation between the semantic description of the building and its geometric description, which allow a semantic object to be associated with multiple geometric representations such as Boundary Representation (BRep), Constructive Solid Geometry (CSG) or extrusion and sweep based geometry descriptions (Figure 1) (Zhang et al. 2014). Although, this would, in principle, facilitate the integration of the multi-scale modeling approach in the geometric part, the integration of this concept in the semantic structure and the explicit definition of refinement relationships is lacking so far.

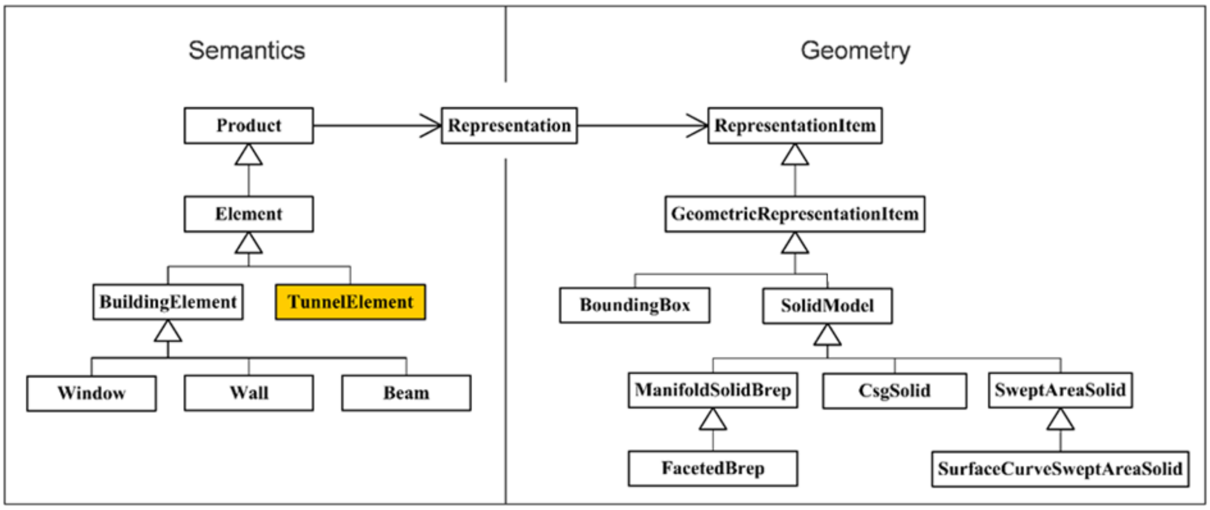


Figure 1 Separation of geometry and semantics in the IFC data model. Left: Small section of the semantic model; Right: Subset of the different geometry representations; Semantics and geometry can be combined through a flexible linkage mechanism using the mechanism using the Representation entity. The highlighted entity represents the proposed extension. The prefix ‘Ifc’ has been omitted.

Despite these limitations, multi-scale approaches are much needed to properly support the design and engineering of track-based infrastructure facilities, such as tunnels. To overcome this issue, we present a comprehensive approach for soundly integrating multi-scale modeling into an IFC-based tunnel model. We follow the principle of “minimal intervention”, i.e. only minimal modifications and extensions to the existing data model are proposed. Our approach respects the important boundary condition that applications which do not support multi-scale approaches should also be able to access and display the model correctly.

This document is structured as follows: In the remaining part of this section a sound description of our extension is presented, providing as well, the basic concepts on the IFC model to understand the structure of our example files. Section 2 discusses different levels of extending the IFC data model and presents corresponding example files.

1.2. Motivation for multi-scale modeling

Construction planning relies heavily on the use of different scales for representing geometric information on a suitable level of detail. The produced drawings range from general site layout plans, which provide an overview of the entire project, down to detailed workshop drawings presenting the precise design of individual components, connection points etc.

Employing a multi-scale representation is particularly important in the context of planning tunneling projects as they typically have a very large extent (several kilometers) and at the same are subject to design decisions in the range of only a few centimeters in order to provide the desired connections and avoid spatial conflicts. Despite the multi-scale characteristics inherent to the planning of tunnels, the current IFC data model provides only very limited support for multi-scale modeling.

Multi-scale representations are however well established in geography and cartography. The underlying concepts have been adopted in the development of the corresponding digital data models. Among them is CityGML, the standard for representing 3D city models, which provides five dedicated levels-of-detail (LoD). Also digital representations of buildings can benefit significantly from storing and exchanging semantic and geometric information on different levels of detail. However, the introduction of multi-scale concepts into Building Information Models (BIM) requires careful consideration of the highly dynamic planning processes which result in frequent modifications of the data stored in the BIM.

To overcome this limitation, we present a comprehensive approach for soundly integrating multi-scale modeling into an IFC-based tunnel model. We make describe in our proposal the content that might be contained in each scale. We make an intensive use of the concept of spaces and products to complete describe the physical objects which define a tunnel. Our approach is implemented in a way that applications which do not support multi-scale still are able to access and display the complete model correctly.

The authors are well aware that it will take a significant time until multi-scale concepts will find their way into standardized product models. However they hope to support progress in that direction with the development of the data models presented in this document.

1.3. Semantic Model

Based on preliminary work by (Yakubi et al. 2007, 2013) we are presenting a product model for shield tunnels which fulfills the demands of data exchange in the context of the design and engineering of large infrastructure projects. Like the IFC model, the proposed tunnel product model provides a clear separation between semantic objects and the associated geometry. In the presented concept, the semantic entities are associated with a particular LoD, which helps to achieve and maintain the semantic-geometric coherence of the overall model.

In order to maintain downwards compatibility with the current IFC standard, we make extensive use of the space structure concept provided by the IFC to model refinement relationships across the LoDs. In the IFC standard, the concept is applied to provide a hierarchical aggregation structure for buildings, using *Site*, *Building* and *BuildingStorey* objects and organizing them by means of the relationship *Aggregates* (Figure 2). In the proposed data model for shield tunnels we apply the space structure concept and introduce corresponding spatial containers. More importantly and as explained in detail below, we make use of the space structure concept for modeling cross-LoD refinement relationships.

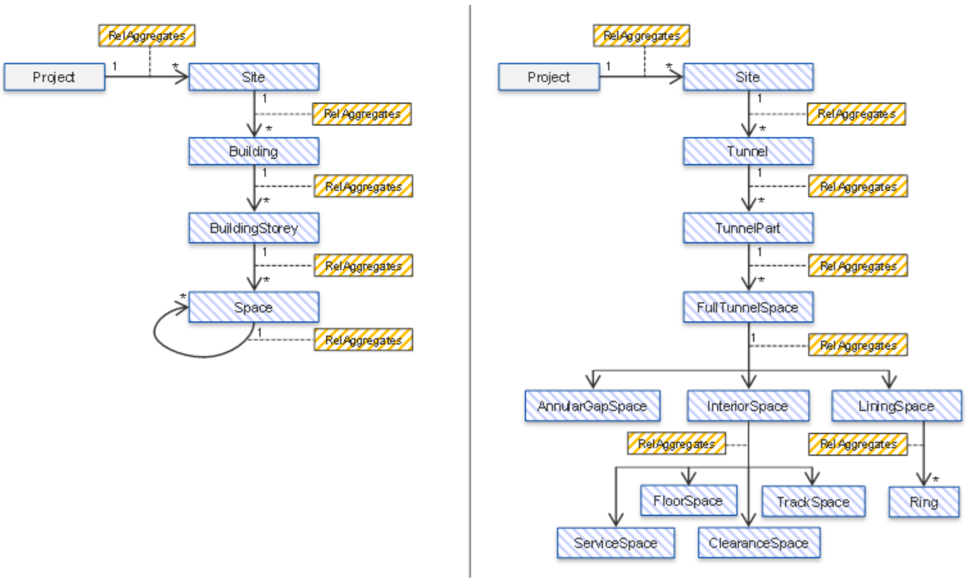


Figure 2 Left: Modeling of space aggregation hierarchies in the IFC standard, Right: Usage of the space aggregation concepts in the proposed extension. The prefix ‘Ifc’ is omitted.

Figure 3 depicts the main components of the tunnel model as 2D cross-sections, while Figure 4 provides a number of 3D views depicting the different LoDs. Figure 5 displays the proposed extensions of the IFC data model to capture shield-tunnel specific elements and provide means for multi-LoD representations. In alignment with the IFC model, the proposed tunnel model extensions consist of space objects and physical objects. Figure 6 provides an instance diagram illustrating how these objects are used and how the relationships between them a set up.

In order to group and provide access to all elements at a certain level of detail, we make use of a new class of relationship objects, which we name *LoD*. These objects aggregate all spatial and physical objects at the corresponding level. At the same time, we maintain the aggregation relationships across the different LoDs in order to explicitly model a refinement hierarchy. This is realized by the newly introduced relationship class *IsRefinedBy*, a subclass of *Aggregates*.

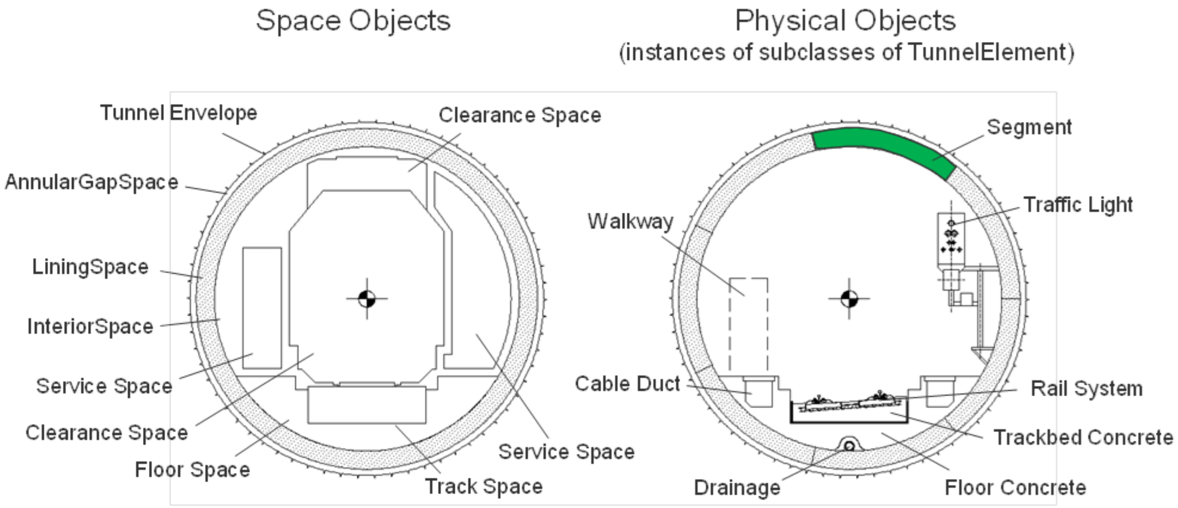


Figure 3 A tunnel cross-section depicting the individual spaces (left) and physical elements (right) of the proposed multi-scale product model.

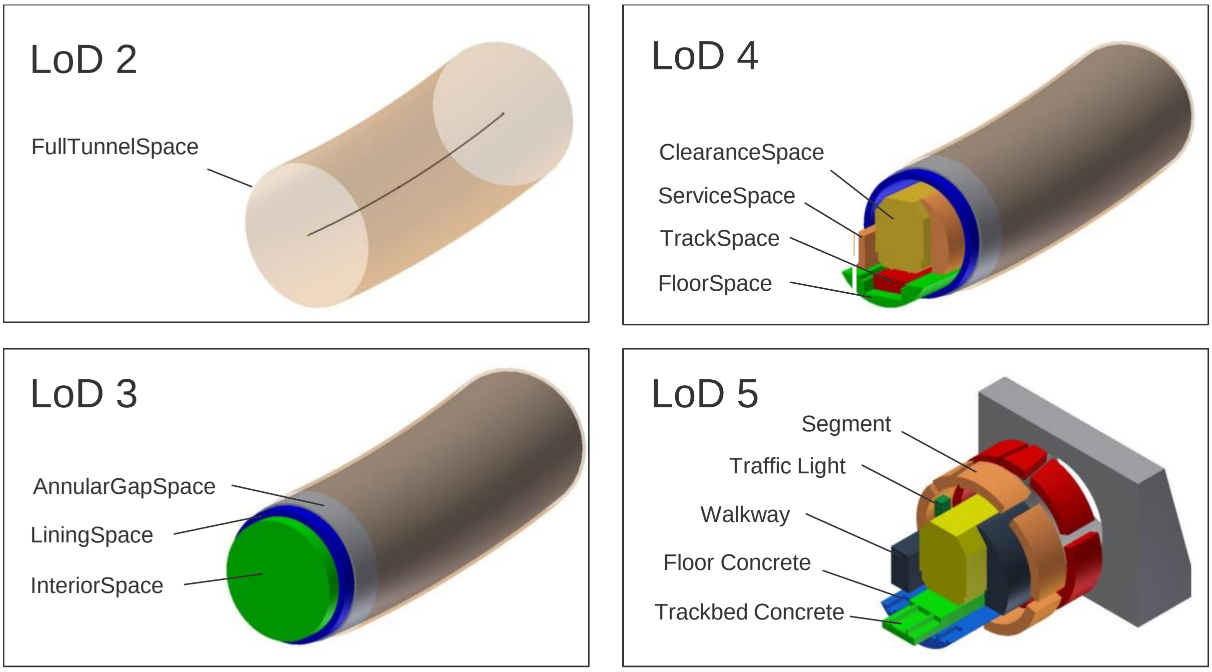


Figure 4: 3D representation of the different LoDs of the multi-scale tunnel product model.

One of the key aspects of our approach is that the refinement hierarchy is created with the help of space objects, while physical objects form part of the finest level only. This allows us to use spaces as placeholders on coarser levels, thus avoiding overlapping physical objects (which could be erroneously interpreted as clashes) and hence providing full compliance with the standard IFC approach. This is different from the LoD concept of CityGML where on each level physical objects can be described.

On LoD 1, the tunnel is represented geometrically by a curve representing the main axis. To this end, the tunnel object is associated with a *TunnelAxis* object which in turn refers to the underlying alignment. Since the alignment plays a key role in the design and engineering of tunnels, it is essential to provide the genuine alignment objects such as lines, arc segments and clothoids as part of the product model (Amann et al. 2013).

For the levels 2 to 5 we employ a strict containment hierarchy. We call this approach the Matyroshka principle: In analogy to the Russian dolls the spaces on a finer level are fully included in a space provided by the coarser level. Physical objects are present only on the finest level, LoD 5. A typical example is the ring space which is a LoD 4 space object representing a complete ring. It comprises the corresponding ring segments which are physical object belonging to LoD 5.

Except for the ring space, all space objects represent longitudinal spaces along the entire *TunnelPart*. The Ring space, however, has the length of a single ring segment only. The relations between the semantic objects rely on the space structure concept, modeling aggregation relationships between the site, the tunnel, the tunnel parts, the longitudinal spaces, and the rings.

On LoD 2, the space object *FullTunnelSpace* is used to provide a semantic object representing the entirety of the tunnel. This space object is further refined on LoD 3 by three distinct (non-overlapping) space objects: *AnnularGapSpace*, *LiningSpace* and *InteriorSpace*. On LoD 4, the interior space is refined by the space objects *ClearanceSpace*, *FloorSpace*, *TrackSpace* and *ServiceSpace*.

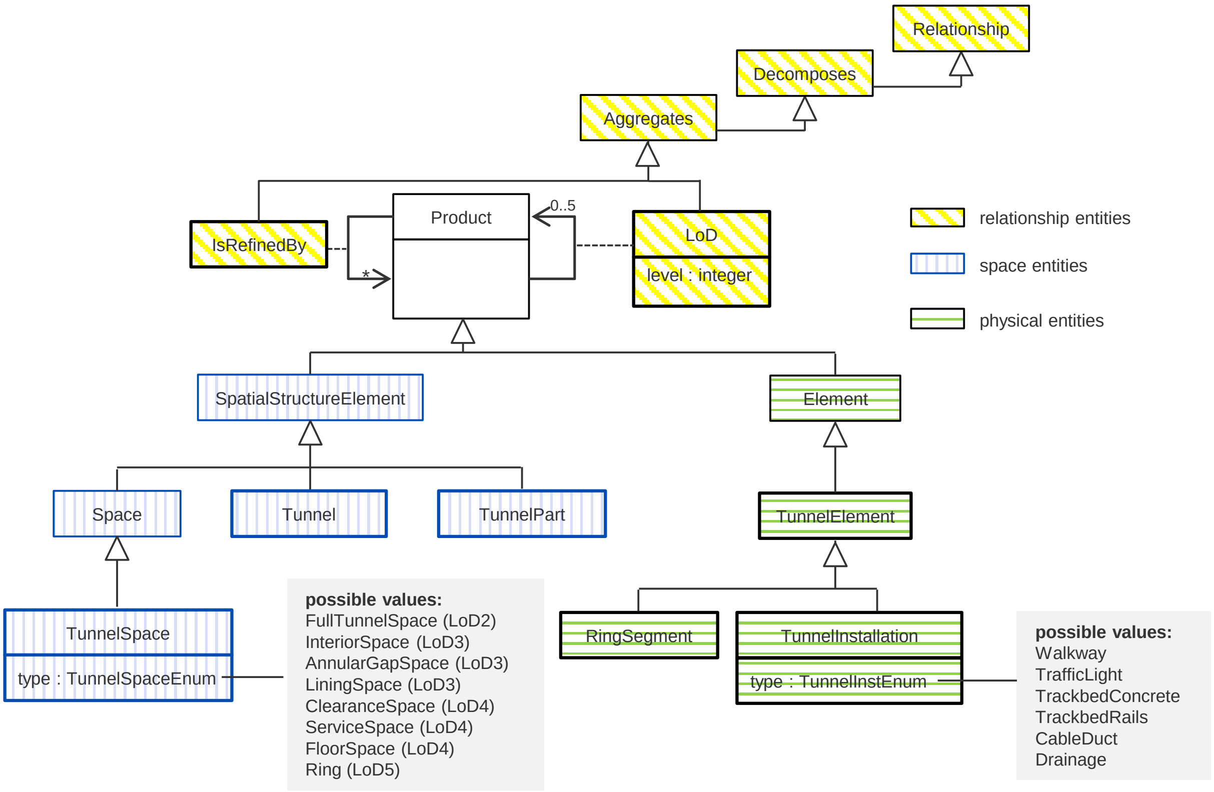


Figure 5 UML Class Diagram depicting the introduced relationship classes *IsRefinedBy* and *LoD*, as well as the classes *TunnelSpace* and *TunnelInstallation* which are used to model tunnel-specific spaces and installations. Classes depicted in blue are subclasses of *SpatialStructureElement*, classes depicted in green are subclasses of *Element* and represent physical objects. Relationship classes are depicted in yellow.

LoD 5 provides the physical objects of the tunnel model. All physical objects are assigned to a respective space via the *ContainedInSpatialStructure* relationship: The objects *TrackBedConcrete* and *TrackBedRails* belong to the *TrackBedSpace*, *CableDuct* and *Drainage* belong to the *FloorSpace*, and *TrafficLight* and *Walkway* objects are embedded in the *ServiceSpace*.

In addition, the *LiningSpace* defined on LoD3 is refined into a number of Ring space objects on LoD 5. Although the *LiningSpace* is a longitudinal object stretching along the entirety of the tunnel, the *Ring* space represents only one ring of ring segments. *Ring* space objects belong to the finest level of detail, since their definition happens at a very advanced stage of the planning process. Each *Ring* space contains the RingSegments it comprises.

In compliance with the principles of object-oriented modeling in general and the IFC modeling guidelines in particular, we decided against a fine-grained class structure where each and every space or component type is represented by a class of its own. Instead, we make use of more general classes and provide them with a type attribute representing a predefined enumeration. This allows for easy maintenance and extendibility.

Following this paradigm we model the diverse spaces depicted in Figure 6, not as individual classes but subsumed by the class *TunnelSpace* which in turn provides a type attribute to select from a number of predefined space types (*FullTunnelSpace*, *InteriorSpace*, etc.). The same approach is applied to the physical tunnel objects which are subsumed by the class *TunnelInstallation*. Here the type attribute is used to select from predefined element types (*TrackbedConcrete*, *CableDuct* etc.). Only *RingSegment* is modeled by means of a dedicated class due to its importance and particular characteristics.

Consequently, the entities depicted have to be interpreted as instances of *TunnelSpace* or *TunnelElement*, respectively, and not as instances of specific classes.

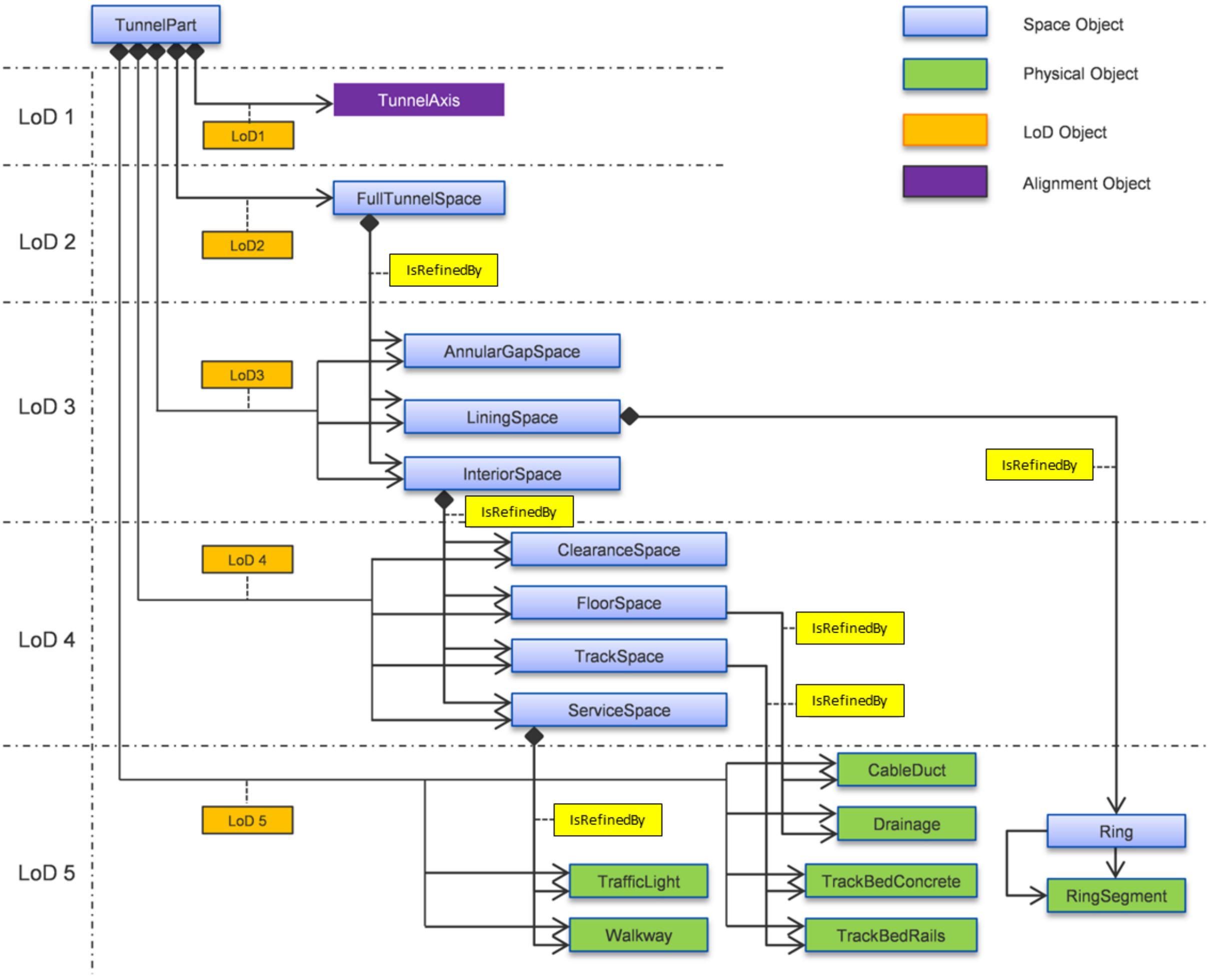


Figure 6 UML Instance diagram depicting the semantic part of the proposed shield tunnel product model incorporating a multi-scale representation. The TunnelPart object is associated with the different representations via dedicated LoD objects. The proposed tunnel models consists of space objects (depicted in blue) and physical objects (depicted in green). Refinement relationships are explicitly modeled across the LoDs. Implementing the Matyroshka principle, the spaces on a finer level are fully included in the corresponding space on the coarser level. Physical objects are modeled only on the finest level.

Figure 5 also illustrates the introduction of the level of detail concept into the class model. As discussed above, a dedicated relationship class *LevelOfDetail* has been integrated as a subclass of the existing relationship class *Aggregates*. This relationship is used to relate instances of subclasses of *Product* to a given level of detail as illustrated in Figure 6. Secondly, the relationship class *IsRefinedBy* has been integrated for modeling the refinement relationships as shown in Figure 6.

1.4. Geometric Representations

As already introduced the IFC standard provides a clear separation between the semantic definition and its geometric representations. This clear separation allows a single semantic model to be represented by multiple shape representations within the same instantiated file. Moreover, when this connection – based on *IfcProductRepresentation* – is established, a space placement of the semantic object must be additional defined.

In the following sub-section points a brief introduction to the placement structure and the different shape representations in IFC is done.

1.4.1. Placement Structure

All objects contained in the semantic description are a subtype of *IfcProduct*, which holds an attribute for its local placement in space. The main reason to define a geometric location in the semantic structure remains in the fact that one single product can be defined by different geometrical representations. Thus, done in this way, all local coordinate systems contained in the different geometric representations of one product object will point to the same coordinate system.

Additionally, at least one global coordinate system must be assigned for every representation context and defined independently of the local placement contained in the space and element objects. The *IfcGeometricRepresentationContext* entity defines this global placement and some additional optional attributes such as *TrueNorth* or *Precision*.

Figure 7 show the placement objects, which are attached parallel to the spatial structure and the element containment. The main difference between the two placement structures relies on the fact that the placement objects attached to the spatial structure re-creates the hierarchy of spaces in the project, whereas the placement objects attached to the physical products usually are linked to its spatial element by a relative placement. (Liebich 2009)

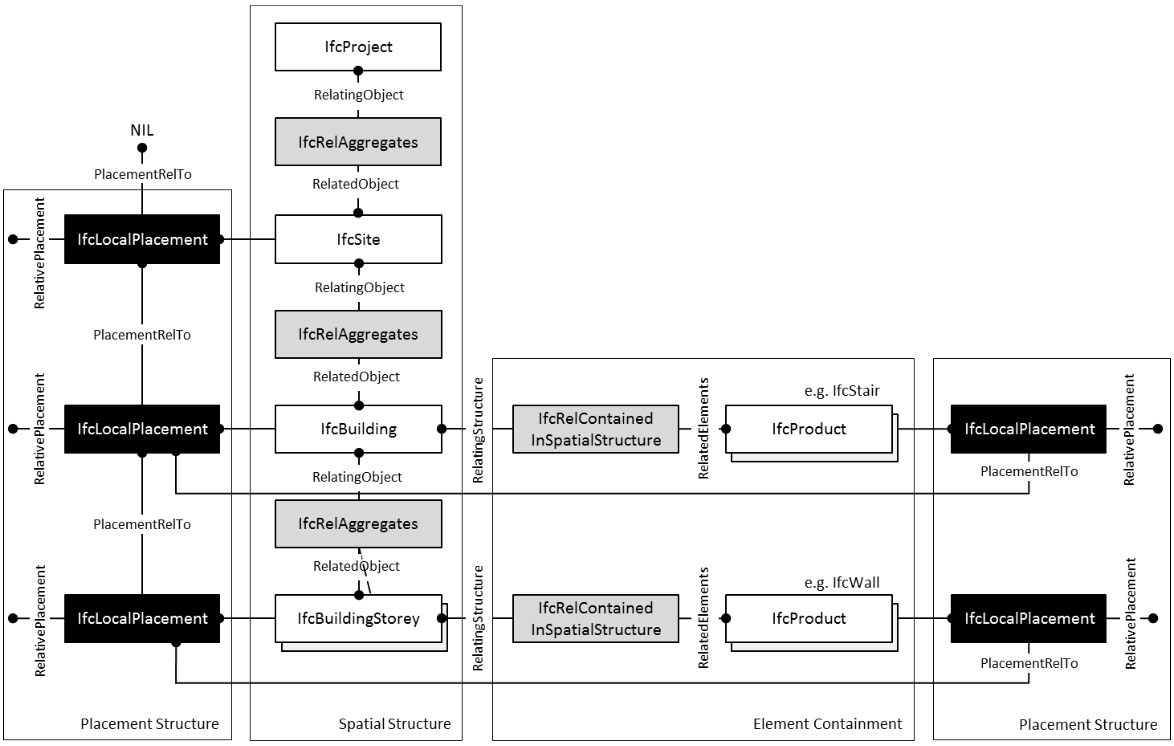


Figure 7: The standard IFC semantic model based on spatial structure and element containment. *IfcLocalPlacement* objects are attached to the *IfcProducts* to provide the basic placement and orientation in space. Source: BuildingSmart

1.4.2. Product Shape Representations

The semantic definition of a product object is connected to the different geometric representation through the *IfcShapeRepresentation* object, who integrates the different geometrical representations.

The following list summarizes the most used geometric representations. For the complete description of all the possible representations the reader is forwarded to (buildingSMART 2014):

- **BoundingBox**: Simplistic 3D representation where a (minimal) box is used to surround the geometry
- **SufaceModel**:
    - Face and Shell based: The geometry is divided in a set of faces or shells depending on the representation selected.
    - Tessellation: The surfaces of the geometry are tiled in planar triangles.
- **SolidModel**:
    - Brep: The surfaces of the geometry are represented by Brep entities.
    - AdvancedBrep: The surfaces of the geometry are represented by Bspline entities.
    - SweptSolid: The geometry is created by swept volumes such as extrusion or revolution. Excluded from those geometries are the tapered volumes.

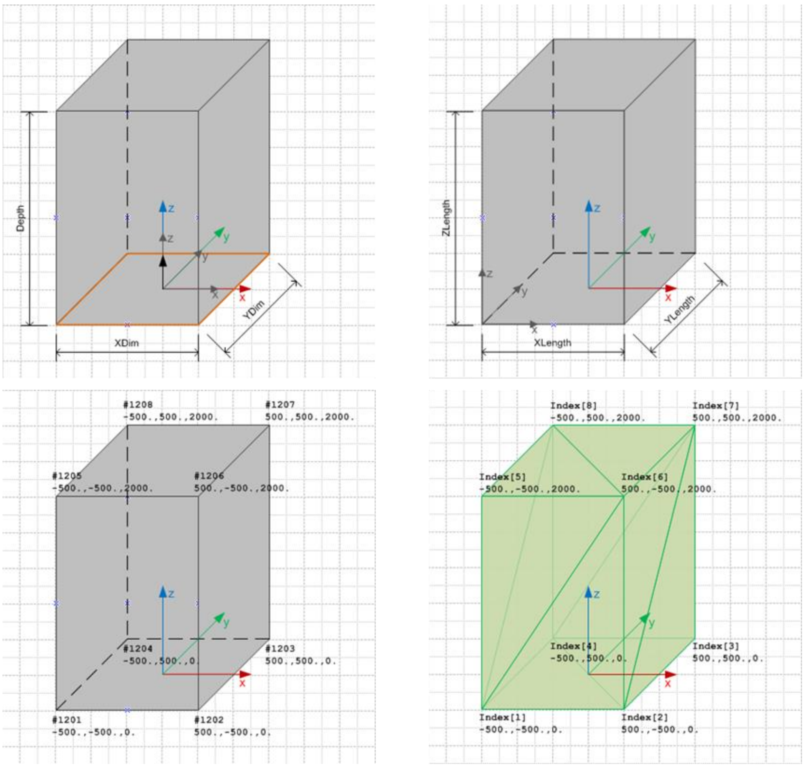


Figure 8: Four different geometric representations of the same model, namely by a (top right) Bounding Box, by a (top left) SweptSolid, by a (bottom left) Brep and by a (bottom right) Tessellation. Source: BuildingSmart

Section 2. EXPRESS schemas and instance files

2.1. Overview

For demonstrating the use of the developed data model, we created a set of schemas and corresponding instance files. As neither the extension of the IFC data model by semantic shield tunnel objects, nor the introduction of multi-LoD concepts into IFC is close to standardization, we provide three different levels of extension.

- Level 1: On the lowest level, we use the standard IFC4 schema without any tunnel-specific extensions. We make use of proxy objects for representing tunnel objects in the semantic part. The respective instance files can be interpreted and visualized by any IFC4-capable viewer.
- Level 2: The schema extends the standard IFC4 schema by tunnel-specific semantic elements, such as *IfcTunnel, IfcTunnelPart, IfcTunnelSpace* and *IfcTunnelElement*. This extension (or a minor variation) is expected to form part of the future IFC-Infrastructure data model.
- Level 3: The schema extends the level 2 schema by entities which allow the explicit representation of the levels-of-detail by introducing the entity *IfcLevelOfDetail* as well as the description of the refinement relationships among the elements by means of the elements *IfcRelIsRefinedBy*,

For each level, we provide examples with different geometry representations:

- IfcFacetedBrep: A triangle-based explicit representation of the elements' geometry.
- IfcAdvancedBrep: A NURBS-based explicit representation of the elements' geometry. We make use of the respective geometry entities introduced in IFC4. NURBS representations are particularly advantageous in the context of tunnels, as their elements possess a high number of curved surfaces.
- IfcExtrudedAreaSolid: An extrusion of the tunnel profile along a straight axis. As by definition of the entity, the extrusion must be a straight path, this geometry representation can only be used to approximate the real geometry in case of a curved axis (linear approximation by segmentation).
- IfcSweptDiskSolid: The geometry is created by sweeping a circular disk along a given axis. We provide different examples which use either an IfcCompositeCurve (a composition of linear and arc segments) or an IfcBSplineCurve as sweeping axis. As this representation supports only the sweeping of a circular disc, models on LoD4 and LoD5 cannot be modeled.
- IfcFixedReferenceSweptAreaSolid: The geometry is created by sweeping an arbitrary closed profile along a given path. For the definition of the path the same examples introduced in the previous geometry are used. As this representation supports the definition of arbitrary geometry, all Levels-of-Detail are modeled.

All variants of the schema as well as the corresponding examples can be downloaded from the following website:

https://www.cms.bgu.tum.de/de/forschung/projekte/31-forschung/projekte/415-ifctunnel.html

2.2. Level 1: Tunnel spaces and objects defined by proxy objects

2.2.1. Description

For the first level of extension we use the standard IFC4 schema without any tunnel-specific extensions. Accordingly, we use the spatial structure entities *Building* and *BuidingStorey* to model the *Tunnel* and *TunnelPart* objects. The tunnel spaces (*IfcFullTunnelSpace, IfcLiningSpace,* etc.) as well as the physical objects (*IfcTunnelElement*) are modeled by IfcProxy objects.

As the schema employed on the Level 1 is the standard IFC4 schema, any IFC viewer capable to read IFC4 files is able to display the model correctly. However, the tunnel-specific semantic information can only be represented in a much reduced manner, as proxy objects are applied. In order to associate the semantic information with the *Building* and *BuidingStorey* objects, we make use of the attribute *Name*, which labels whether the object is a *Tunnel* or *TunnelPart*. Similarly, the attribute *Tag* on the *Proxy* object is used to describe the type of tunnel space or tunnel element, while the *ProxyType* is left as *Notdefined*.

On Level 1, there is no explicit representation of the different levels of detail, since this concept is not supported by the IFC4 schema.

2.2.2. STEP-P21 example files

For demonstrating the use of the Level 1 tunnel model, we are providing a set of instance examples enumerated in Table 1. This work is not yet finished, and some of the examples are currently under development.

IfcProxy: schema		**LoD1**	**LoD2**	**LoD3**	**LoD4**	**LoD5**
IfcFacetedBrep		Work in progress	Work in progress	Work in progress	Work in progress	Work in progress
IfcAdvancedBrep (NURBS)		Work in progress	Work in progress	Work in progress	Work in progress	Work in progress
IfcExtrudedAreaSolid + IfcArbitraryClosedProfileDefinition + IfcCircle + IfcBooleanResult			📷 ❖	📷 ❖	📷 ❖	Work in progress
IfcSweptDiskSolid	IfcCompositeCurve	📷 ❖	📷 ❖	📷 ❖		
	IfcBSplineCurve	❖	❖	❖		
IfcFixedReferenceSweptAreaSolid	IfcCompositeCurve	📷 ❖	❖	❖	❖	Work in progress
	IfcBSplineCurve	❖	❖	❖	❖	Work in progress

Table 1 EXPRESS files modeled based on *Proxy* entities.

For the Level 1 of integration, we provide the following geometry representation examples:

- IfcExtrudedAreaSolid: As the extrusion of the tunnel profile is usually done along a straight axis and defined by a *height* parameter, no model is provided for the LoD1. For the LoD2, LoD3 and LoD4 a combination of different IFC entities is used to represent the cross-sections of the different spaces. Thus, as example, *IfcCircle* and *IfcBooleanResult* are used for the modeling of the lining space and *IfcArbitraryClosedProfileDefinition* for the modeling of the clearance space.

- IfcSweptDiskSolid: As previously mentioned, the geometry is created by sweeping a circular disk along a given axis. We provide two different examples for the definition of the axis based on *IfcCompositeCurve* (a composition of linear and arc segments) and *IfcBSplineCurve*. As models on LoD4 and LoD5 cannot be modeled by a disk, those Levels-of-Detail are left empty.

- IfcFixedReferenceSweptAreaSolid: The geometry is created by sweeping an arbitrary closed profile along a given path. For the definition of the path the same examples introduced in the previous geometry are used. For the definition of the different cross-sections the same IFC entities used for the extruded geometry is used.

2.3. Level 2: Tunnel model defined without LoD

2.3.1. Description

For the second level of integration we introduce the tunnel entities described in the proposed model extension. Hence, our examples start with the compulsory *Project* and *Site* objects, and then we incorporate the new *Tunnel* and *TunnelPart* objects as is shown in Figure 2.

In addition, we do not use the *IfcProxy* object to represent the different spaces, but the *IfcTunnelSpace* as defined in our extension model. Thus, contained in the properties of these objects we are able to introduce the complete tunnel-specific semantic information. Even more, although we do not introduce yet the concept of Level-of-Detail, we structure the spaces under the same hierarchy we introduced in Figure 6 by means of *IfcRelAggregate*.

As these examples are based on an extension of the IFC product model, which is not yet part of the standard, the resulting examples cannot be interpreted by any of the currently available IFC viewers.

2.3.2. STEP-P21 example files

For the files contained on this level, we extended the Level 1 examples incorporating the entities *Tunnel*, *TunnelPart* and *TunnelSpace*. Table 2shows the different examples developed for this level.

IfcTunnel without LoD: schema		**LoD1**	**LoD2**	**LoD3**	**LoD4**	**LoD5**
IfcFacetedBrep		Work in progress	Work in progress	Work in progress	Work in progress	Work in progress
IfcAdvancedBrep (NURBS)		Work in progress	Work in progress	Work in progress	Work in progress	Work in progress
IfcExtrudedAreaSolid + IfcArbitraryClosedProfileDefinition + IfcCircle + IfcBooleanResult			❖	❖	❖	Work in progress
IfcSweptDiskSolid	IfcCompositeCurve	❖	❖	❖		
	IfcBSplineCurve	❖	❖	❖		
IfcFixedReferenceSweptAreaSolid	IfcCompositeCurve	❖	❖	❖	❖	Work in progress
	IfcBSplineCurve	❖	❖	❖	❖	Work in progress

Table 2 EXPRESS files modeled based on Tunnel entities, but without LoD.

2.4. Level 3: Tunnel model defined with LoD

2.4.1. Description

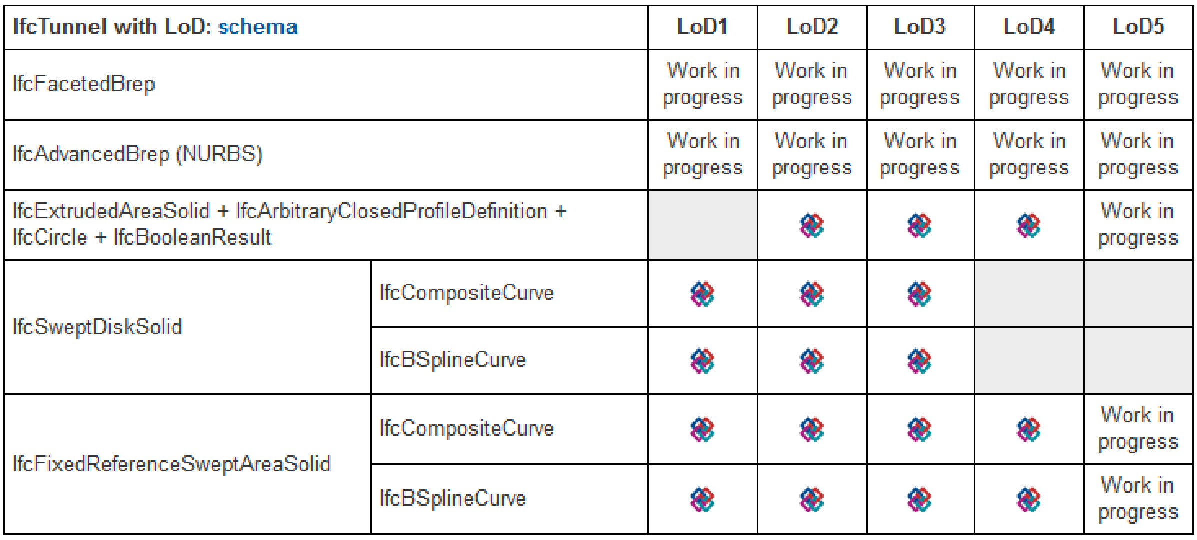
On the third and highest extension level, we introduce the aggregation entities *LoD* and *IsRefinedBy*, which substitute the *IfcRelAggregates* used in the previous levels of integration. The aggregation *LoD* is, as shown in Figure 6, used to connect the different spaces and elements with the object TunnelPart. Done in this way, a capable viewer can filter the model based on the Level-of-Detail, and therefore shown only the relevant information.

Differently the aggregation *IsRefinedBy* is used to reproduce the hierarchical structure of spaces and physical elements. Moreover, when the aggregation is done between a space and an element, the aggregation *IfcRelContainedInSpaceStructure* is maintained. This allows the standard IFC viewers to recognize the relation between the spatial structure and the element containment independently of the Level-of-Detail.

As the following examples are based on an extension of the IFC product model, which is not yet part of the standard, the resulting examples cannot be interpreted by any of the currently available IFC viewers.

2.4.2. STEP P-21 example files

For the files contained on this level of integration, we extended the examples contained in the second level of integration incorporating the aggregation entities *LoD* and *IsRefinedBy*. Table show the different examples developed at this level.



IfcTunnel with LoD: schema		LoD1	LoD2	LoD3	LoD4	LoD5
IfcFacetedBrep		Work in progress	Work in progress	Work in progress	Work in progress	Work in progress
IfcAdvancedBrep (NURBS)		Work in progress	Work in progress	Work in progress	Work in progress	Work in progress
IfcExtrudedAreaSolid + IfcArbitraryClosedProfileDefinition + IfcCircle + IfcBooleanResult			◆	◆	◆	Work in progress
IfcSweptDiskSolid	IfcCompositeCurve	◆	◆	◆		
	IfcBSplineCurve	◆	◆	◆		
IfcFixedReferenceSweptAreaSolid	IfcCompositeCurve	◆	◆	◆	◆	Work in progress
	IfcBSplineCurve	◆	◆	◆	◆	Work in progress

Table 3 EXPRESS files based on Tunnel entities and LoD aggregations.

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 1st Int. Conf. on Civil and Building Engineering Informatics, Tokyo, Japan

Borrmann, A., Kolbe, T.H., Donaubauer, A., Steuer, H., Jubierre, J.R., Flurl, M. (2014): Multi-Scale Geometric-Semantic Modeling of Shield Tunnels for GIS and BIM Applications, Computer-Aided Civil and Infrastructure Engineering, in press. DOI: 10.1111/mice.12090

buildingSMART (2014) IfcShapeRepresentation. Available at: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcrepresentationresource/lexical/ifcshaperepresentation.htm (Accessed: 15th July 2014)

Liebich, T. (2009): IFC 2x Edition 3 – Model Implementation Guide (version 2.0), buildingSMART International Modeling Support Group

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., Takashi, A., Hiroshi, F. (2013): Development and application of a product model for shield tunnels. Proc. of the 30th Int. Symposium on Automation and Robotics in Construction, Montréal, Canada

Zhang, J., Yu. F, Li, D., Hu, Z. (2014), Development and Implementation of an Industry Foundation Classes-Based Graphic Information Model for Virtual Construction, Computer-Aided Civil and Infrastructure Engineering 29 (1), 60–74