IfcTunnel - A proposal for a multi-scale extension of the IFC data model for shield tunnels under consideration of downward compatibility aspects

S.Vilgertshofer, J. R. Jubierre & A. Borrmann
Chair of Computational Modeling and Simulation, Leonhard Obermeyer Center
Technical University of Munich, Germany

ABSTRACT: The Industry Foundation Classes (IFC) provide a comprehensive, standardized and neutral data format to enable the exchange of digital building models. However, the current version of IFC lacks the ability to comprehensively describe infrastructure facilities such as roads, bridges, railways or tunnels in detail. This paper shows the general concept for a space oriented approach to describe shield tunnel models by extending the IFC and the integration of multiple levels of detail into the IFC standard in the scope of considering downward compatibility aspects. The proposal therefore introduces three consecutive levels of extension. Thus, we enable any IFC-viewer supporting IFC4 to visualize the exemplary instance files created in the first level by using proxy objects. The higher levels extend the standard IFC4 schema by tunnel-specific semantic elements. They also integrate an approach aiming at the representation of multi-scale models by integrating multiple levels-of-detail.

1 INTRODUCTION

The Industry Foundation Classes (IFC) provide a comprehensive, standardized and neutral data model to enable the exchange of digital building models.

Although IFC models are able to represent a wide variety of buildings, they are not explicitly well suited for exchanging product data models of infrastructure constructions such as roads, bridges or tunnels. Additionally, IFC does not support the exchange of models implementing the concept of different levels of detail (LoD), which is widely used in the GIS domain and particularly useful for the display of a product model's geometry in varying scales. This multi-scale representation is particularly important in the context of planning infrastructure projects like tunnels, as they typically extend over long distances. For the routing of the alignment, a kilometer scale is required, while a centimeter scale needs to be considered for the detailed design of connection points and to avoid spatial conflicts.

This paper presents a proposal for a shield tunnel product model based on the IFC under consideration of preliminary work by Yabuki (Yabuki et al., 2007; Yabuki et al., 2013). It is based on a formerly published conceptual proposal (Borrmann & Jubierre, 2013; Borrmann et al., 2014a, 2014b) and further enhances this approach in the scope of considering downward compatibility aspects. It shows the general concept for this space oriented approach to describe shield tunnel models by extending the IFC and the integration of multiple levels of detail into the IFC standard. Additionally, we introduce three consecutive levels of extension. Thus, we enable any IFC-viewer supporting IFC4 to visualize the exemplary instance files created in the lowest level. The higher levels then extend the standard IFC4 schema by tunnel-specific semantic elements (Figure 1).

The presented product model for shield tunnels fulfills the requirements on data exchange in the context of the design and engineering of large infrastructure projects. As demanded by IFC modeling guidelines, our proposal provides a clear separation between semantic objects and the corresponding geometry. The concept also implements the association of semantic entities with a particular LoD in order to enable LoD-dependent model views.

In order to maintain downward compatibility with the current IFC standard, we make use of the space structure concept provided by the IFC to model refinement relationships across the different LoDs. In the IFC standard, this concept is applied to provide a hierarchical aggregation structure for buildings (using *IfcSite*, *IfcBuilding*, etc.) and to organize them by means of the relationship *IfcRelAggregates*. In the proposed data model, this space structure concept is used to introduce corresponding spatial containers which describe different spaces of a tunnel. Additionally, we make use of this space structure concept for modeling cross-LoD refinement relationships.

Level of Extension		Tunnel Model	Levels of Detail
Level of Extension	1	tunnel entities modeled as proxy objects	not available
	2	integration of designated tunnel entities	not available
	3	integration of designated tunnel entities	integration of aggregation entities to include LoDs

Figure1. The three levels of extension. Models generated in Level 1 can be imported by any IFC4-capable viewer.

A key aspect of our approach is that the refinement hierarchy is created with the help of space objects, while physical objects are modeled on the finest level only. It allows us to use spaces as placeholders on coarser levels which prevents overlapping of physical objects. It differs from the multi-scale concept of CityGML, which allows the description of physical objects on every LoD (Borrmann et al., 2014b; Kolbe, 2009). Additionally, the explicit dependencies defined by the refinement relations allow the execution of cross-LoD consistency checks.

The presented approach introduces five levels of detail. In LoD 1 the tunnel is geometrically represented by a curve describing the main axis. For the levels 2 to 4 a strict containment hierarchy is employed: The spaces on a finer level are fully included in a space provided by the coarser level. In the 5th LoD, each physical object is placed in one of the spaces of the coarser LoDs. Figure 2 depicts the main space objects and physical objects described by this extension schema.

The feasibility of our approach is demonstrated by a collection of example files, describing an exemplary tunnel in different LoDs by using various methods of geometric representations that the IFC supports. The example files are available online and can be downloaded at: *www.cms.bgu.tum.de/ifctunnel*.

In the first level of extension we create these models by employing the standard IFC4 schema and use *IfcProxy* objects to model the tunnel spaces. The models created this way can be displayed by any IFC viewer capable to read IFC4 files. In a second and third level of integration we introduce the specific tunnel entities and LOD-related information as described in the proposed tunnel extension. These models can only be displayed in an IFC viewer that implements the proposed extension, though.

This paper is structured as follows: In Section 2 previous and related works as well as the theoretical background of this proposal are outlined. Section 3 gives a description of the shield tunnel product model in terms of semantics and geometry. Section 4 shows the three levels of extension, which ensure downward compatibility. The final Section concludes the paper and summarizes its main findings.

2 RELATED WORK AND THEORETICAL BACKGROUND

The Industry Foundation Classes (IFC) is a comprehensive data model for the exchange of information in the domain of building design, engineering and construction (Liebich et al., 2013), which has been developed over the last decade by the international organization buildingSMART. IFC aims at being a vendor-independent open data format to support the interoperability in the AEC industry. The IFC data model is defined by the data modeling language EXPRESS, which is a part of the ISO standard 10303 “STEP – Standard for the exchange of product model data”(Laakso & Kiviniemi, 2012). An important feature of the IFC model is the use of objectified relationships and inverse attributes. The IFC data model consists of several hundred entities, which enable the

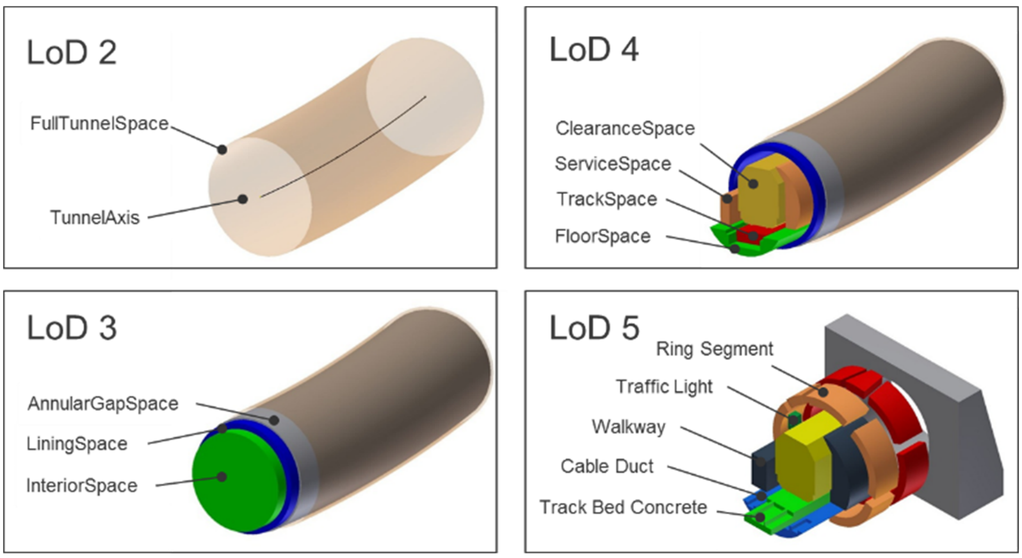

Figure 2. A 3D representation of the different LoDs of the multi-scale tunnel product model.

description of the semantics and geometry of buildings and building parts.

There is only little research that addresses the extension of the IFC model to enable the description of tunnel facilities. The first contributions date back almost 10 years to a fundamental study on the development of a shield tunnel product model (Yabuki et al. 2007). This proposal was further developed (Yabuki et al. 2013) but has not been integrated into the IFC standard as of yet. Another IFC-based date scheme for the representation of NATM tunnels has been developed by Lee et al. (2015). In Stascheit et al. (2013) a holistic IFC-based tunnel product model is discussed. It includes a ground data model and a tunnel boring machine model as well as a tunnel product model. This tunnel product model is based on the following approach by Borrmann et al.

Under consideration of the preliminary work by Yabuki et al., a concept for a shield tunnel product model has been developed (Borrmann & Jubierre, 2013; Borrmann et al., 2014b). This concept aims to fulfill the demands of data exchange in the context of the design and engineering of large infrastructure projects and forms the basis of the product model that is described in Section 3 in detail. As it is not yet part of the IFC this paper demonstrates its applicability by introducing a low level extension that can be used with current IFC viewers.

An important aspect of the proposal by Borrmann et al. is the integration of multi-scale modeling into an infrastructure product model, which is a concept well established in the GIS domain. An important example is CityGML, an XML-based data model for the representation of 3D city models, which comprises five LoDs (Kolbe, 2009).

Multi-scale models provide different geometric representations of a semantic object in each LoD. These representations are then used to visualize the modeled buildings or infrastructure facilities in different scales.

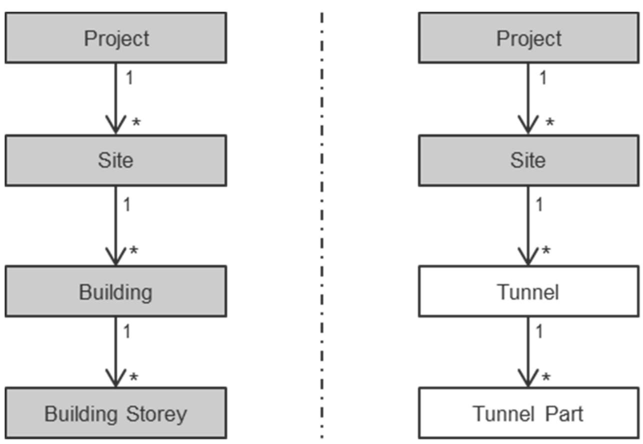

Figure 3. Left: Hierarchical space structure in the IFC standard. Right: Space aggregation structure in the proposed extension. The prefix 'Ifc' is omitted.

3 DESIGN OF THE SEMANTIC SHIELD TUNNEL PRODUCT MODEL

The following section gives an overview of the design of the semantic and geometric parts of the proposed shield tunnel product model that was introduced by Borrmann et al. It is the basis for the three level extension approach that is shown in Section 4.

The presented product model for shield tunnels aims to comply with the requirements of data exchange necessary for the design and engineering of large linear infrastructure projects, i.e. shield tunnels.

In compliance with the IFC standard, the tunnel product model is based on a clear separation between semantic objects and the geometric representation of those objects. A key point is the association of each semantic entity with a particular LoD to achieve semantic-geometric coherence of the overall model.

The IFC standard provides a space structure concept to provide a hierarchical aggregation structure for buildings (Figure 3). It uses *IfcSite*, *IfcBuilding* and *IfcBuildingStorey* objects and orders them by employing the relationship *IfcRelAggregates.* This space structure concept is used to model refinement relationships across the different LoDs in the proposed product model. Doing so, downward compatibility with the current IFC standard is maintained.

Figure 2 depicts the defined LoDs 2-5 of the tunnel model as 3D views. The extensions that need to be added to the IFC data model to describe shield tunnel specific elements and enable multi-scale representations are listed in Figure 5.

The extensions necessary to implement the proposed tunnel model consist of space objects and physical objects in compliance with the IFC model. To group and to select all elements belonging to a certain level of detail, a new relationship entity is

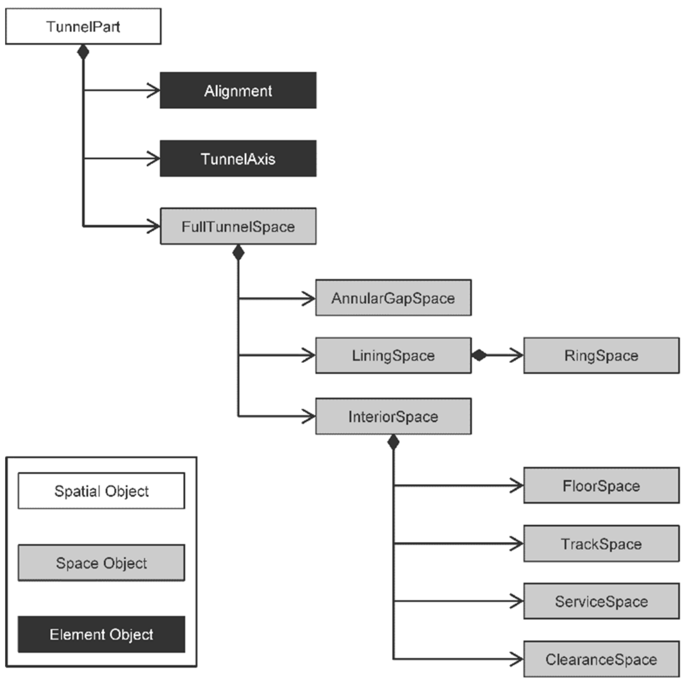

Figure 4. Space aggregation structure of a TunnelPart in the proposed extension. The prefix 'Ifc' is omitted.

used: *IfcLoD*. Instances of this entity aggregate all spatial and physical objects at a certain LoD. The LoD is specified by the entity's attribute *Level*. To explicitly model the refinement hierarchy, aggregation relationships across the different LoDs are maintained. This is realized by the newly introduced relationship entity *IfcIsRefinedBy*, which inherits from *IfcRelAggregates*.

The definition of the refinement hierarchy is realized by the use of space objects as placeholders. The physical objects are modeled only on the highest level (Figure 2). This use of spaces as placeholders on coarser levels prevents an overlapping of physical objects, which could be wrongly interpreted as clashes, thus being fully compliant with the general IFC approach. On the contrary, the LoD concept of CityGML allows the description of physical objects on any LoD (Kolbe, 2009).

On LoD 1, the tunnel is represented geometrically by a curve representing the alignment. To this end, the respective *IfcTunnelPart* object is associated with an *IfcTunnelAxis*. This object then refers to its geometric representation in form of a curve representing the underlying alignment. The alignment is of major importance in the design and engineering of infrastructure facilities such as tunnels and it is therefore essential for a product model to include alignment objects e.g. lines, arc segments and clothoids (Amann et al. 2013).

The containment hierarchy of the levels 2 to 5 defines that spaces on a finer level are completely included in a space of the coarser level. For example, all the spaces defined in LoD 4 are refining the *InteriorSpace* of LoD3. The relations between these semantic objects is realized by the space structure concept and thereby creates the containment hierarchy (Figure 4).

In the following listing the spaces of the LoDs 2-4 are presented:

- LoD 2: The *FullTunnelSpace* is used to provide an object that represents the complete outer bounding of the tunnel.
- LoD 3: The three non-overlapping space objects *AnnularGapSpace*, *LiningSpace* and *InteriorSpace* refine the *FullTunnelSpace*
- LoD 4: The *InteriorSpace* is refined by the space objects *ClearanceSpace*, *FloorSpace*, *TrackSpace* and *ServiceSpace*.

In LoD 5 any physical object of the tunnel can be modeled and is also assigned to a certain space by the *IfcContainedInSpatialStructure* relationship.

The proposed model makes use of very general entities and provides them with an attribute to declare the specific type of an object based on a predefined enumeration. It is thereby avoided to define a particular entity to represents every element component, which is the case in the approaches described by Yabuki et al. and Lee et al. This general concept follows the principles of object-oriented modeling and the IFC modeling guidelines and allows easy maintenance and extendibility (Borrmann et al., 2014b).

By this paradigm the diverse tunnel spaces are not modeled as individual entities. Instead they are combined in the entity *IfcTunnelSpace*, which provides a type attribute that explicitly defines the space types of the object (FullTunnelSpace, InteriorSpace, etc.). The

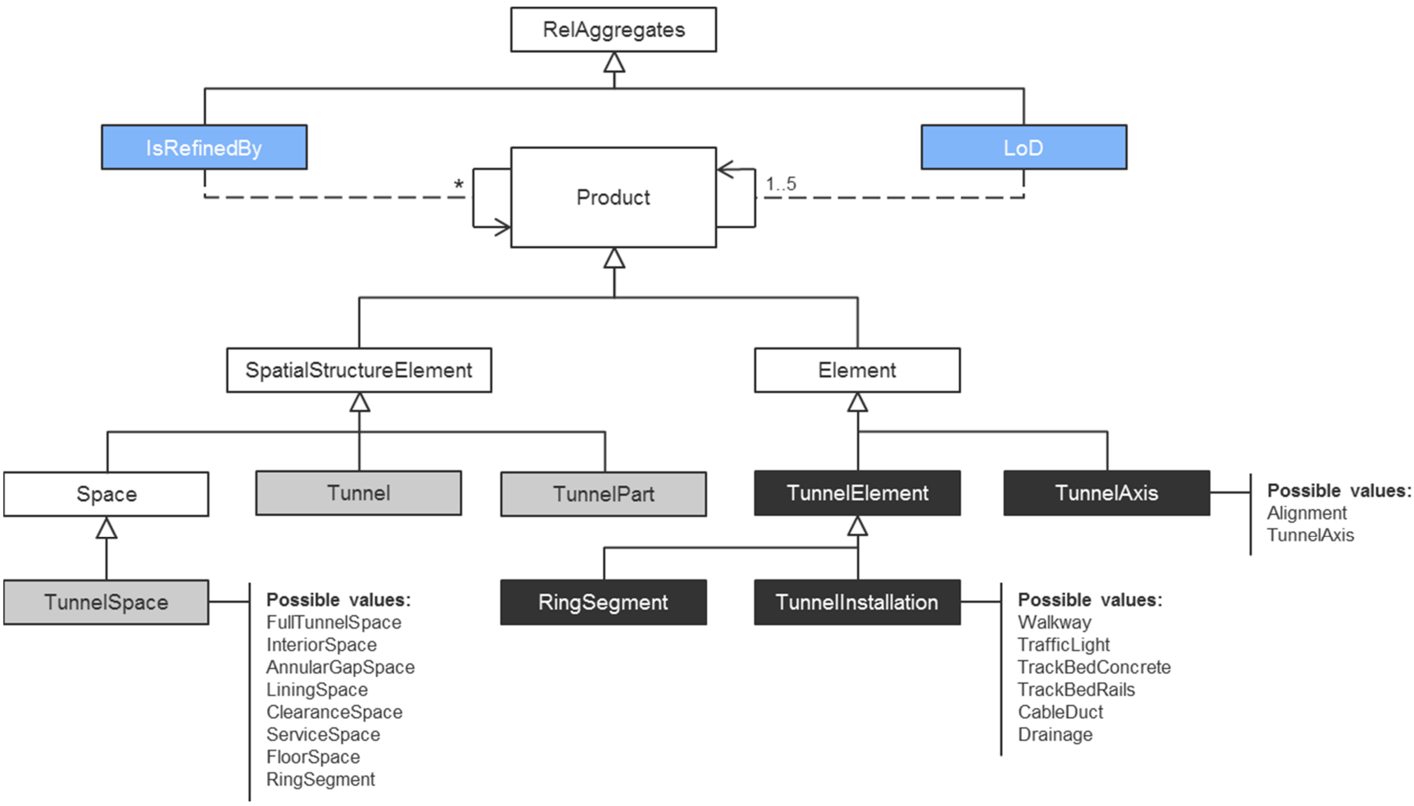

Figure 5. UML class diagram depicting the introduced relationship and object classes.

same applies to the physical tunnel objects. The only objects that are modeled by a dedicated entity due to its importance and particular characteristics are the RingSegments. That means most objects have to be interpreted as instances of TunnelSpace or TunnelElement, instead of as instances of a specific entity.

The level of detail concept is introduced into the class model with the help of a dedicated relationship entity *IfcLoD*, which is a subclass of the existing relationship entity *IfcRelAggregates*. This relationship is used to assign instances of subclasses of *IfcProduct* to a given level of detail. Additionally, the relationship entity *IfcIsRefinedBy* has been integrated for modeling the refinement relationships.

4 ENABLING DOWNWARD COMPATIBILITY

The concept described in Section 3 and the other approaches of creating a shield tunnel product model, which were listed in Section 2, are not yet implemented in the IFC standard. Even up-to-date IFC-viewers are not able to interpret the described models, as neither the extension of the IFC data model by semantic shield tunnel objects nor the introduction of multi-LoD concepts into IFC are close to standardization. Thus, the possibilities to demonstrate the use of the developed data models are severely limited.

Based on the product model presented in Section 3, we created a set of schemas and corresponding instance files, to show the applicability of this approach. These schemas and instance files are provided in three different consecutive levels of extension and therefore enable downward compatibility with the current IFC4 standard (Figure 6).

At *www.cms.bgu.tum.de/ifctunnel* we present example shield tunnel models in different levels of detail, which were created with the proposed product model extension. As only models of the first extension level can be displayed by IFC-viewers, we can only present images of these models. The IFC files are imported and displayed with the FZK Viewer developed at the Karlsruhe Institute of Technology ("FZKViewer," 2016).

4.1 *Overview*

On the lowest level of extension, we use the standard IFC4 schema without any tunnel-specific extensions. Instead, we make use of the *IfcProxy* entity for representing tunnel spaces and objects. The semantic description is provided by the *Name* attribute of the *IfcProxy* objects. The instance files created in this level of extension can be interpreted and visualized by any IFC4-capable viewer.

On the second level the schema extends the standard IFC4 schema by tunnel-specific semantic elements, e.g. *IfcTunnel*, *IfcTunnelPart* and *IfcTunnelSpace*. This extension (or a variation) is proposed to form part of a concept used in a future IFC for infrastructure data model.

IfcProxy: schema		**LoD1**	**LoD2**	**LoD3**	**LoD4**	**LoD5**	**All LoDs**
IfcFacetedBrep			📷 ◆	📷 ◆	📷 ◆	Work in progress	◆
IfcAdvancedBrep (NURBS)			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	

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

Figure 6. Example IFC-files available at *www.cms.bgu.tum.de/ifctunnel*

The third level of extension expands the second level schema by entities that allow the explicit representation of the levels of detail. On this level we introduce the entities *IfcLevelOfDetail* and *IfcRelIsRefinedBy*. The entity *IfcLOD* defines the level of detail that a tunnel object belongs to *IfcRelIsRefinedBy* is used to describe the refinement relationships among the objects belonging to different LoDs.

For each level, we generated examples with different geometry representations. Which type of representation is used when a model is generated should always depends on the geometric properties of an object. This means that as far as possible implicit representations should be preferred, while explicit representations should only be used when no implicit representation is possible or available.

The entities that are used for geometry representation purposes are listed as follows:

- *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. By definition of the entity, the extrusion must be a straight path. Therefore, this geometry representation can only be used as an approximation, if the real geometry is based on a curved axis (linear approximation by segmentation).
- *IfcSweptDiskSolid*: The geometry is created by sweeping a circular disk along a given axis. We created 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 LoD 4 and LoD 5 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 methods as listed in the previous paragraph are used. As this representation supports the definition of arbitrary geometry, all levels of detail are modeled.

4.2 *Level 1*

We use the standard IFC4 schema without any tunnel-specific extensions for the first level of extension. Accordingly, we model the *Tunnel* and *TunnelPart* objects by using the *IfcProxy* entity. The tunnel spaces (*IfcFullTunnelSpace*, *IfcLiningSpace*, etc.) as well as the physical objects (*IfcTunnelElement*) are also modeled as instances of the *IfcProxy* entity. The relations between those objects is realized with *IfcRelAggregates*.

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 reduced manner, as the *IfcProxy* entity is applied. In order to associate the tunnel specific semantic information with the objects, we make use of the attribute *Name* of *IfcProxy*, which labels whether the object is a Tunnel, TunnelPart, a certain type of tunnel space or a tunnel element. The attribute *ProxyType* is left as *Notdefined*.

On the first level, there is not yet an explicit representation of the different levels of detail, as an integration of this concept is not supported by the current IFC4 schema.

4.3 *Level 2*

For the second level of integration we introduce the tunnel entities described in the proposed model extension. Hence, our examples start with the compulsory *IfcProject* and *IfcSite* objects. Then we incorporate the new *IfcTunnel* and *IfcTunnelPart* objects as is shown in Figure 3.

We do not use the *IfcProxy* object to represent these objects and the different spaces, but the *IfcTunnel*-spaces as defined in the extension model. Thus, we are able to model the complete tunnel-specific semantic information by containing it in the attributes of the introduced entities. Although we do not introduce the LoD concept yet, we structure the spaces under the same hierarchy we introduced in Figure 3 and 4 by means of *IfcRelAggregate*.

The examples on the second level are based on a customary extension of the IFC product model, which is not yet part of the standard. Therefore, the resulting examples cannot be interpreted by any of the currently available IFC-viewers.

4.4 *Level 3*

Only on the third and highest level of extension, we introduce the aggregation entities *IfcLod* and *IfcRelIsRefinedBy*, which substitute *IfcRelAggregates* used in the previous levels. The aggregation *IfcLod* is used to connect the different spaces and elements with *IfcTunnelPart*. This way, a capable viewer can filter the model based on the levels of detail and thereby show only relevant information. *IfcIsRefinedBy* 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 *IfcRelContainedInSpatialStructure* 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.

The examples generated within this extension level are also based on an extension of the IFC product model. Therefore, they cannot be interpreted by any of the currently available IFC viewers.

5 CONCLUSION

The current version of the IFC standard is not well suited for representing and exchanging product models of infrastructure facilities.

This paper gives an overview of the concept for a future extension of the IFC standard in order to enable the detailed modeling and exchange of shield tunnels product models. The planning of such large infrastructure facilities requires the consideration of differing scales. Therefore, the proposed extension introduces the concept of multi-scale modeling by enabling the representation of objects in different levels of detail.

Based on the presentation of the product model, the paper focuses on the issue of compatibility between the proposed model extension and current applications for interpreting IFC-files. As the definition is not yet included in the IFC standard, there are no IFC-viewers capable of importing IFC-files that use shield tunnel specific entities.

Therefore, we present the shield tunnel product model in three consecutive levels of extension. This gradual approach provides a low level implementation that does not require an extension of the existing IFC 4 standard, but introduces the concept of the presented product model by using the *IfcProxy* entity. This downward compatibility allows existing IFC-viewers to interpret example files generated in scope of this research. Only on the second and third level of extension, new IFC entities are defined to model tunnel- or LoD-specific objects and relations.

By the introduction of the shield tunnel product model in these three levels of extension we aim at demonstrating the use of the developed data model. Thereby, we present the advantages of the product model and show how the application of the proposed approach can be applied.

ACKNOLEDGEMENTS

We gratefully acknowledge the support of the German Research Foundation (DFG) for funding the project under grant FOR 1546.

REFERENCES

Borrmann, A., Flurl, M., Jubierre, J.R., Mundani, R.-P., Rank, E., 2014a. Synchronous collaborative tunnel design based on consistency-preserving multi-scale models. Advanced Engineering Informatics 28, 499–517.

Borrmann, A., Jubierre, J.R., 2013. A multi-scale tunnel product model providing coherent geometry and semantics. In: Proc. of the 2013 ASCE International Workshop on Computing in Civil Engineering. Los Angeles, pp. 291–298.

Borrmann, A., Kolbe, T.H., Donaubauer, A., Steuer, H., Jubierre, J.R., Flurl, M., 2014b. Multi-Scale Geometric-Semantic Modeling of Shield Tunnels for GIS and BIM Applications. Computer-Aided Civil and Infrastructure Engineering 30, 263–281.

FZKViewer [WWW Document], 2016. URL http://www.iai.fzk.de/www-extern/index.php?id=1931 (accessed 4.14.16).

Kolbe, T.H., 2009. Representing and Exchanging 3D City Models with CityGML. In: Lee, J., Zlatanova, S. (Eds.), Proceedings of the 3rd International Workshop on 3D Geo-Information, Seoul, Korea, Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg.

Laakso, M., Kiviniemi, A., 2012. The IFC Standard - A Review of History, Development and Standardization. ITcon Journal of Information Technology in Construction 17, 134–161.

Lee, S.H., Park, S.I., Park, J., 2015. Development of an IFC-Based data schema for the design information representation of the NATM tunnel. KSCE Journal of Civil Engineering 00, 1–12.

Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Richter, S., Chipman, T., Weise, M., Wix, J., 2013. Industry Foundation Classes: Version 4. BuildingSmart International (Model SupportGroup).

Stascheit, J., Meschke, G., Koch, C., Hegeman, F., König, M., 2013. Processoriented numerical simulation of mechanized tunneling using an IFC-based tunnel product model. In: Proceedings of the 13th International Conference on Construction Applications of Virtual Reality. London, UK.

Yabuki, N., Aruga, T., Furuya, H., 2013. Development and application of a product model for shield tunnels. In: Proceedings of the 30th ISARC. Montréal.

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, 261–268.