Multi-scale geometric-semantic modeling of shield tunnels for GIS and BIM applications

A. Borrmann, T. H. Kolbe, A. Donaubauer, H. Steuer, J. R. Jubierre, M. Flurl

Leonhard Obermeyer Center of Digital Methods for the Built Environment
Technische Universität München,
Arcisstraße 21, 80290 Munich, Germany

Published in *Computer-Aided Civil and Infrastructure Engineering, Wiley*
DOI: 10.1111/mice.12090

Abstract: *The planning of large infrastructure facilities such as inner-city subway tracks requires the consideration of widely differing scales, ranging from the kilometer scale for the general routing of the track down to the centimeter scale for detailed design of connection points. On the one hand this implies the utilization of both, Geographic Information Systems (GIS) as well as Building Information Modeling (BIM) tools, for performing the required analysis, modeling, and visualization tasks. On the other hand, a sound foundation of handling multi-scale representations is required. While multi-scale modeling is already well established in the GIS field, there are no corresponding approaches in Infrastructure BIM so far. However, multi-scale concepts are also much needed in the BIM context, as the planning process typically provides only rough information in the early stages and increasingly detailed and fine-grained information in later stages. To meet this demand, this paper presents a comprehensive concept for incorporating multi-scale representations with building information models, with a particular focus on the geometric-semantic modeling of shield tunnels. Based on a detailed analysis of the data modeling methods used in CityGML and the requirements present in the context of infrastructure planning projects, we discuss potential extensions to the BIM data model Industry Foundation Classes (IFC) for incorporating multi-scale representations of shield tunnels. Particular emphasis is put on providing means for preserving the consistency of the representation across the different Levels-of-Detail (LoD), while taking into account both, semantics and geometry. For realizing consistency preservation mechanisms, we propose to apply a procedural geometry description which makes it possible to define explicit dependencies between geometric entities on different LoDs. The modification of an object on a coarse level consequently results in an automated update of all dependent objects on the finer levels. Finally, we discuss the transformation of the IFC-based multi-scale tunnel model into a CityGML compliant tunnel representation.*

1 INTRODUCTION

The design and engineering of infrastructure facilities is a highly complex task, as numerous constraints and boundary conditions have to be taken into account. This includes the connection with the existing transport network as well as the technical characteristics of the infrastructure facility itself. As a consequence, a large number of specialists are involved which requires intensive and continuous coordination. At the same time widely differing scales have to be considered – ranging from the kilometer scale for the general routing of the carriageway down to the centimeter scale for the detailed planning of individual track nodes.

Since humans started to create geographic maps, they have made use of scale-dependent representations for providing cartographic information on a suitable level of abstraction. This helps to reduce the complexity of the maps' content, improves their readability and allows the viewer to concentrate on the relevant information. This approach has consequently been adopted by digital cartographic methods and integrated into the respective data models and standards.

Construction planning also 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, today's data models for representing and exchanging planning data support multi-scale modeling only to a very limited extent.

Another important aspect of modern computer-aided track planning is the increasing demand for integrating computer-aided design (CAD) with geographic information systems (GIS). While the former is required to perform the actual design process, the latter is used for assessing the resulting track design with respect to different criteria, such as environmental impact, traffic connections etc. This concept – which is often referred to as "geo-design" – has received increased attention in recent years (Steinitz 2012).

In this paper we describe an approach for transferring the concept of multi-scale representations from the field of 3D city modeling to the field of infrastructure planning, more precisely the planning of tunneled carriageways. A major challenge lies in the diverging characteristics of the application scenarios with respect to the data dynamics. While geographic and cartographic information is rather static and rarely subject to modifications, the design process involved with the planning of carriageways is highly dynamic and data updates occur in high frequency. The conventional approach taken by geographic data models relying on maintaining independent representations on the different levels-of-detail thus, is not appropriate here, because the risk of losing consistency is too high.

In order to overcome this issue, we are proposing an approach which is based on the explicit modeling of dependencies between the individual levels of detail (LoD) to achieve automated consistency preservation. This is realized by means of a procedural description of the geometric model, i.e. the geometry is not described through an explicit boundary representation but by means of a procedural model comprising the individual construction steps conducted. In addition, methods of parametric modeling are employed to define dependencies between the geometric entities of the different levels of detail. Modifying the representation on a coarse LoD thus results in an automated update of all dependent objects on the finer LoDs – hence realizing automated cross-LoD consistency preservation.

An important objective of our multi-scale data modeling activities is the maintenance of spatio-semantic coherence (Stadler & Kolbe 2007; Clementini 2010): The semantic description must be aligned with the geometric representation. We discuss in detail how this aspect is addressed by the proposed multi-scale model.

For the purpose of GIS-based route assessment, its visualization in a geographical context, or the update of the spatial database after completion of the planned object, the procedurally created model has to be embedded in the geographic context. In order to achieve this, the semantic and geometric data of the model needs to be transformed into an explicit representation, e.g. into the CityGML format. We also discuss this aspect.

The paper is organized as follows: Section 2 discusses background and related work of our research. Section 3 gives an overview on the multi-scale facilities of the geospatial standard CityGML, in particular with respect to the modeling of tunnels. Section 4 describes the concepts and techniques of parametric and procedural modeling which provide the basis for dynamic multi-scale modeling with automated cross-LoD consistency preservation. Based on this, Section 5 presents a multi-scale product model for shield tunnels based on the data model Industry Foundation Classes (IFC) and discusses its integration with a procedural geometry representation. Section 6 discusses a prototypical implementation on the basis of the parametric CAD system Autodesk Inventor. Section 7 describes the mapping of the product model to the geospatial model CityGML and its use in the context of geo-analysis. Section 8 provides a real-world study proving the suitability of the presented concept, and Section 9 concludes the paper and summarizes its main findings.

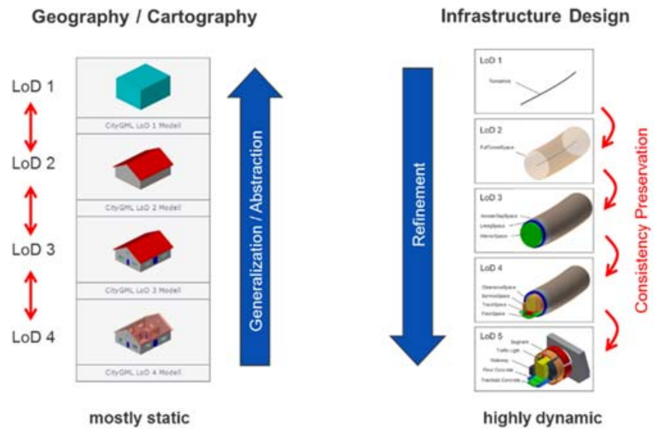
2 BACKGROUND / RELATED WORK

The concept of describing buildings and infrastructure facilities using multiple levels of detail is well established in the GIS domain (van Oosterom & Schenkelaars 1995, Sester & Brenner 2009). In cartography, mostly a bottom-up approach is followed, i.e. coarser representations are abstracted from detailed data by a process called 'generalization' (Forberg 2007; Meng & Forberg 2007). Often, though, models on different LoDs are independently generated due to decoupled acquisition processes. In consequence, the available data models for representing and exchanging 3D city models rely on an independent representation of the geometry on the individual LoDs. For example, CityGML, a comprehensive data model for representing 3D city models, provides five different levels-of-detail (Kolbe 2008), as discussed in more detail in Section 3. An alternative approach to LoD creation and management is presented in Döllner and Buchholz (2005) and Ohori et al. (2013), who propose the dynamic generation of coarse representations from finer ones and replace finite LoDs by continuous ones.

Planning processes, however, implement a top-down approach starting from a coarse representation (e.g. the general course of a tunnel represented by 3D curve) and advancing by continuously elaborating the design until reaching the fully detailed model required for production and/or construction (Figure 1). At the same time, planning processes have an inherently iterative character, meaning that design problems may become apparent not before a certain level of detail is reached, but then imply the need for modifications on coarser levels of detail. A good example in the context of designing shield tunnels is the identification of clashes with existing infrastructure facilities, which is only possible after the tunnel diameter has been determined. The tunnel diameter, however, is derived from the interior configuration which requires a very detailed model. Detecting a clash will usually force the planner to adapt the alignment of the tunnel axis, which requires the planner to go back to the coarsest LoD. This iterative design cycle is only poorly supported by tunnel design tools available today, meaning that changes in the alignment force for a complete manual re-elaboration of the tunnel design and the associated 2D drawings. To overcome these limitations, this paper introduces the concept of explicit LoDs in tunnel design by adopting and extending concepts from the GIS domain. Most importantly, it presents a methodology which allows the definition of dependencies among the different LoDs enabling an automated update of the finer LoDs when coarser ones are modified. The methodology relies on the application of parametric modeling techniques which are explained in detail in Section 4.

The concepts underlying parametric modeling were developed in the 1990s and subsequently implemented in mature commercial CAD systems, including Autodesk Inventor, Dassault CATIA and Siemens NX. Today, these systems are used mainly in the mechanical engineering domain, but there is also increasing adoption in the AEC industry (Lee et al. 2006). In particular, parametric modeling concepts have recently been applied to create highly flexible and adaptable models of infrastructure facilities, such as bridges and roadways (Ji et al. 2011a, Ji et al. 2011b, Obergriesser et al. 2011, Ji et al. 2013). The application of parametric modeling concepts makes it possible to embed domain knowledge in the generation of the model, thus fostering re-usability (Lee at al. 2006). However, so far parametric modeling has not been applied for the creation of consistency-preserving multi-scale models.

As planning large infrastructure facilities requires intense collaboration among the numerous experts involved, a periodic handover of planning information has to be performed as soon as well-defined planning stages are reached. In today's industrial practice, this information handover is realized by exchanging 2D plans, which neither transport 3D geometry nor semantics, resulting in an immense effort on the receiving site for interpretation and re-use. For building design, this issue has been addressed by developing the concept of Building Information Modeling (BIM), which relies on the creation and exchange of comprehensive digital representations of buildings (Eastman et al. 2008). The concept is implemented by means of the standardized data model Industry Foundation Classes (IFC) (Liebich et al 2013). However, so far the IFC standard does not support the representation of tunnels. It does also not explicitly define LoDs nor provide means for a procedural description of geometry. This paper discusses the necessary extensions to fill these gaps in Section 5.



3 MULTI-SCALE MODELING OF TUNNELS IN CITYGML

Multi-scale modeling is an integral part of the CityGML standard and is implemented through five well-defined Levels of Detail. This concept makes it possible for an object to be represented in different LoDs simultaneously. In this way, results from differing data collection methods or models optimized according to differing application requirements can be integrated in a single dataset (Kolbe & Gröger 2003). Additionally, LoDs enable efficient data analysis and visualization.

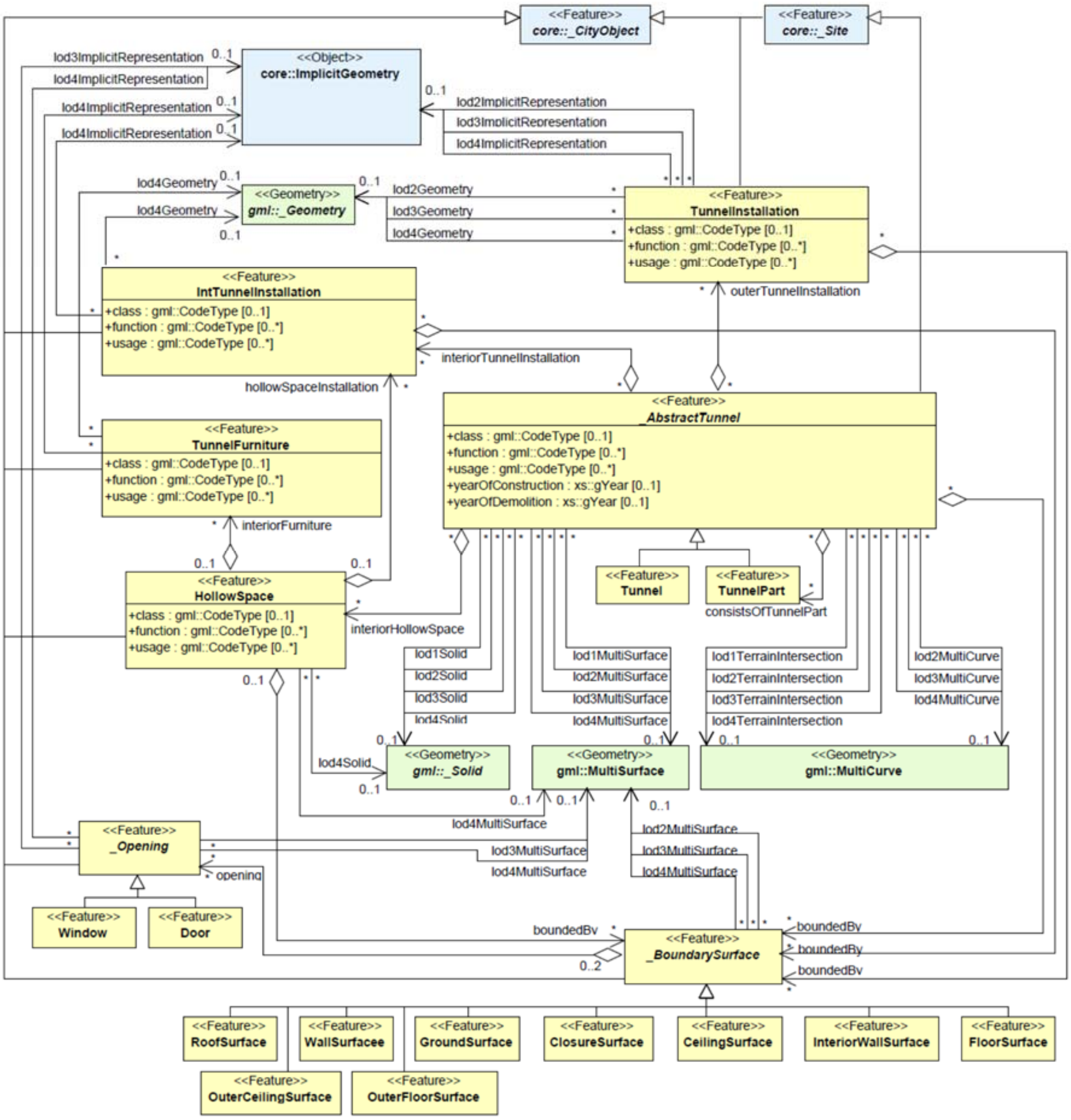
CityGML supports the concept of cartographic model generalization by allowing to aggregate a set of objects which are too small to visualize on a specific scale into a joint representation (McMaster & Shea 1992) by providing an explicit generalization association between city objects. For example, several city objects of a high level of detail may be represented by a single object on a lower level of detail.

Since in this paper we are focusing on the representation of tunnel constructions, we will discuss the LoD features by means of the tunnel model which has been introduced to CityGML since version 2.0 (Gröger et al. 2012).

On each of the LoDs 1-4, a tunnel object can be represented by distinct geometries. Since CityGML employs a consistent LoD concept for all its thematic modules, the LoDs 1-3 describe the outer shell only; more precisely the tunnel's boundary surface adjacent to the surroundings. LoD4 adds the modeling of the interior of a tunnel.

Figure 2 depicts the data model in UML notation. A tunnel model in LoD1 consists of a boundary representation of the tunnel volume without any further semantic classification. Its geometry must be a solid resulting from a vertical extrusion. Additionally it is possible to model the intersection of tunnel and terrain using gml::MultiCurve.

In LoD2 a tunnel may be modeled in greater detail using additional gml::MultiSurface or gml::MultiCurve objects.



Additionally, the structure of a tunnel can be differentiated semantically.

Thus, boundary surfaces can be classified as wall, roof, ground plate, outer floor, outer ceiling or ClosureSurfaces. Furthermore, tunnel elements which strongly affect the outer appearance, e.g. stairs, can be modeled using the class TunnelInstallation which is additionally described by its attributes class, function and usage.

LoD3 adds openings like doors and windows represented as thematic objects to the tunnel model. LoD4 provides the most detailed modeling capabilities by allowing the modeling of the interior structure of a tunnel. The free space inside of a tunnel can be subdivided into several (potentially overlapping) semantic objects called HollowSpace. According to the CityGML standard a HollowSpace "should be uniquely related to exactly one tunnel or tunnel part object" (Gröger et al. 2012). It may be classified by its attributes class, function and usage where class denotes a general classification, e.g. commercial or private rooms, and function and usage describe the designated and actual usage respectively.

Using the grouping concept provided by CityGML, hollow spaces may be aggregated according to arbitrary, user defined criteria. These groups can be semantically enriched by providing group names, specific attributes and role names of the participating objects.

The boundary surfaces of a HollowSpace may be modeled as specialized semantic objects like FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface.

Furthermore, interior objects which cannot be moved can be represented in LoD4 by the class IntTunnelInstallation. Tunnel Installations can either be associated with HollowSpaces (e.g. ventilator, signals) or with the _AbstractTunnel (e.g. pipes or cable trench).

However, the standard states that “it will be within the responsibility of the user or application to make sure that objects in different LoDs refer to the same real-world object” (Gröger et al. 2012). This implies furthermore that the consistency of representations of objects on differing LoDs has to be ensured by the user or application.

The multi-scale modeling approach of tunnels in CityGML has three major limitations in the context of the planning of tunneled carriageways:

(1) A single object can be represented by differing, unconnected geometric representations on the different LoDs. The consistency of these differing representations cannot be enforced by CityGML.

(2) For the planning process of a tunnel, a LoD model of the interior is needed. In CityGML the interior is modeled on LoD4 only.

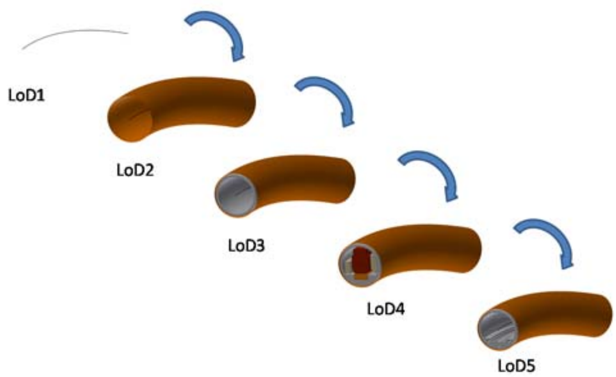
(3) The LoD concept of CityGML reflects the process of acquiring geospatial data by reconstructing existing real world entities from observations and measurements. It is not specifically adapted to the tunnel design process.

To sum up, a more specialized LoD approach is needed for the highly dynamic phases of planning processes. In the following sections we introduce a novel approach which allows to maintain the consistency of representations on different LoDs in an automated manner.

4 MULTI-SCALE GEOMETRIC MODELING FOR THE DYNAMIC PLANNING OF TUNNELS

This paper presents a new methodology for creating and storing multi-scale geometric models for shield tunnels which relies on the explicit definition of dependencies between the individual levels-of-detail. These explicit dependencies allow for automated consistency checks and even automated consistency preservation.

The proposed methodology is based on a step-wise development of tunnel models evolving from a coarse level of detail to the finer ones, which precisely reflects the well-established best practice in tunnel design. Conventionally, fundamental modifications on a coarse level in a late planning phase, such as the modification of the principal tunnel axis, force the planners to completely re-elaborate all related models and plans, e.g. the detailed tunnel geometry. Applying the methodology presented in this paper ensures that modifications on a coarse LoD are automatically propagated to all finer LoDs, thus providing a means for the automated preservation of consistency and, at the same time, significantly reducing the effort required for re-elaboration.

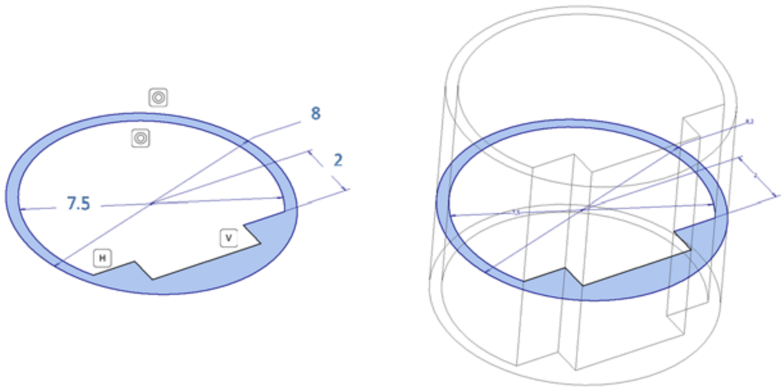


The definition of the dependencies between the different LoDs is realized by applying technologies provided by parametric CAD systems. The core concept is not to store the final outcome of the construction process, i.e. an explicit geometric model, but instead the history of the individual construction operations. Such models are referred to as *procedural models* or *construction history models* (Mun et al. 2003, Stiteler 2004, Pratt et al. 2005, Koch & Firmenich 2011).

In the proposed concept, the LoDs can be flexibly defined by the planning team according to the requirements of the tunneling project under consideration. During the modeling process, switches between one LoD and another are explicitly triggered by the designing engineer who in this way decides which geometric elements belong to which LoD. As an example, Figure 3 illustrates the five different LoDs defined for the design of a tunneled carriageway. These LoDs correspond to well-defined planning phases in current industry practice.

Applying procedural technologies for multi-scale modeling provides the possibility for a stringent definition of dependencies between individual geometric elements on different levels of detail. Thus, the LoDs of the model are not isolated from each other, but inter-related by means of the construction history (Figure 4). Accordingly, the resulting multi-scale model is inherently consistent and preserves a high degree of flexibility. Modifications of elements of a coarse LoD, such as the principal axis of the tunnel are automatically propagated to all dependent objects on the finer LoDs. This includes the tunnel’s shell comprising the individual ring segments as well as the position and orientation of all interior installations.

However, there are limitations to the degree of modifications on coarse levels which can be propagated to finer ones. These limitations are mainly driven by operations in the construction history which only produce results if certain conditions are fulfilled by their operands. A typical example is the Boolean intersect operation which only generates a valid vol-



ume object if the operands do overlap. If their position is determined by earlier operations, the Boolean operation might fail, resulting in a non-evaluable procedural model. The user must be informed about such issues to take appropiate actions.

As stated above, the creation and maintenance of the multi-scale model is achieved through the application of parametric CAD systems. They typically implement a twofold approach, comprising the definition of 2D sketches, including dimensional and geometric constraints on the one hand, and the subsequent procedural definition of 3D volumes through the sequential use of geometric operations such as extrusion, transformation and Boolean operations on the other hand (Figure 5) (Bettig & Shah, 2001). The realization of the proposed multi-scale approach makes use of both principles for defining dependencies between geometric elements.

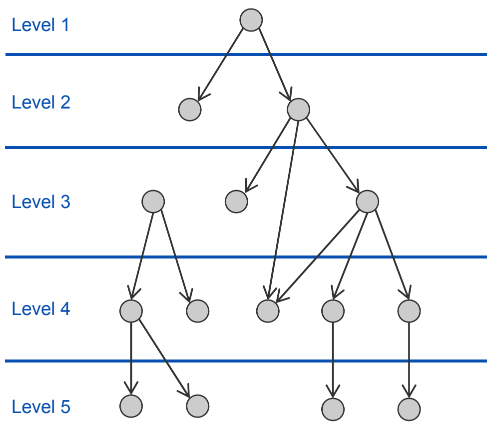
For the composition of a parametric sketch, the user can apply geometric constraints to pairs of geometric elements (points, lines, arcs), thus specifying their relative position. Additionally, dimensional constraints can be used to restrict the size or the position of a geometric element. For defining dimensions, parameters can be used and their values can be interrelated to each other by means of arithmetic expressions. These two types of constraints allow the generation of complex 2D designs capturing geometric rules and providing a high degree of flexibility.

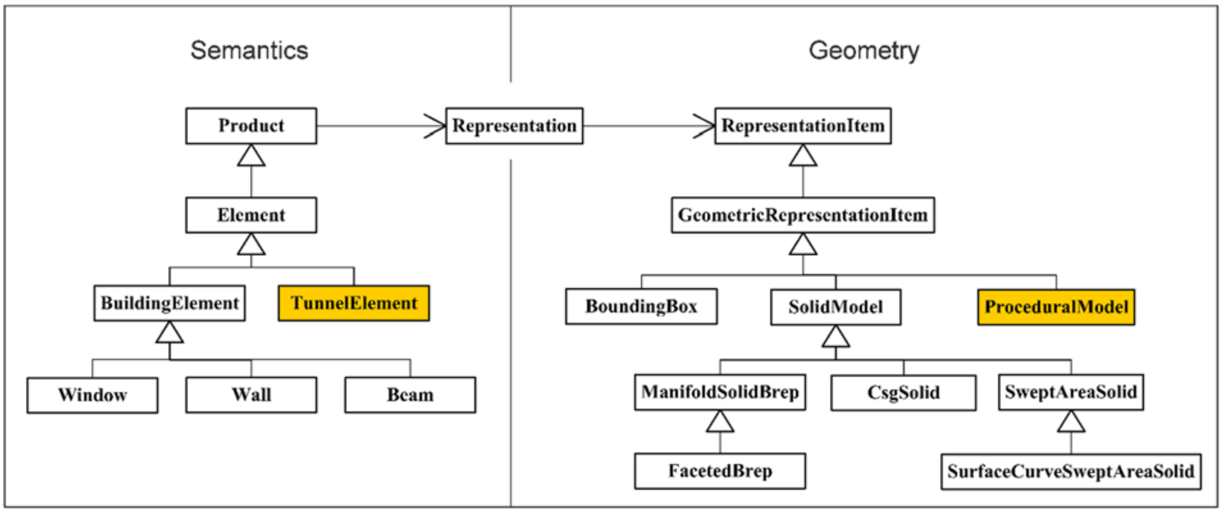
The second important concept provided by parametric CAD systems is the explicitly available construction history. The system records each single construction operation and displays the resulting list as part of the user interface. All operations are parameterized – e.g. the height of an extrusion is an explicitly available parameter. The maintenance of the construction history stands in strong contrast to conventional systems which only store the result of the construction operations, usually be means of an explicit boundary representation. The procedural approach provides the user of the system with the possibility to easily modify an existing model by going back in the construction history and adapting the corresponding parameter, such as a dimension in a sketch or the path applied in an extrusion operation.

The construction operations provided by parametric 3D CAD systems include operations which create volumetric objects from parametric sketches (e.g. sweeping, extrusion etc.). On the resulting volumetric objects further 3D operations may be applied, such as union, intersection, chamfering etc. The combination of these different aspects of parametric design makes it possible to create highly flexible and complex 3D models.

For the implementation of the methodology presented in Section 3 we make use of both aspects, parametric sketches and the construction history, to define dependencies between the geometric elements of different LoDs. In order to pre-

```
1.  CREATE SPLINE          LoD1
2.  CREATE SKETCH          LoD1
------------------------------------
3.  CREATE SWEEP           LoD2
4.  CREATE SKETCH          LoD2
------------------------------------
5.  CREATE SWEEP           LoD1
6.  BOOLEAN UNION          LoD1
------------------------------------
7.  CREATE SKETCH          LoD2
8.  DEFINE CONSTRAINT      LoD2
9.  CREATE SWEEP           LoD2
------------------------------------
10. CREATE SWEEP           LoD3
11. BOOLEAN DIFFERENCE     LoD3
12. CREATE EXTRUSION       LoD3
13. ...
```





serve the flexibility of the multi-scale model we do not maintain the resulting explicit geometry of the representations on the different LoDs but instead store the underlying sketches and construction operations. The resulting geometry description is denoted as a procedural representation (ISO, 2005).

5. A MULTI-SCALE SHIELD-TUNNEL PRODUCT MODEL

5.1 Overview

Within the AEC industry, the data exchange between different stakeholders is of crucial importance. The use of neutral, open data formats has proven to be the most suitable approach to realize this data exchange. Here we aim to provide a neutral data model which provides the possibility to share a procedural description of multi-scale models in order to transmit the dependencies between the different LoDs and which makes it possible to maintain the flexibility and inherent consistency of the model.

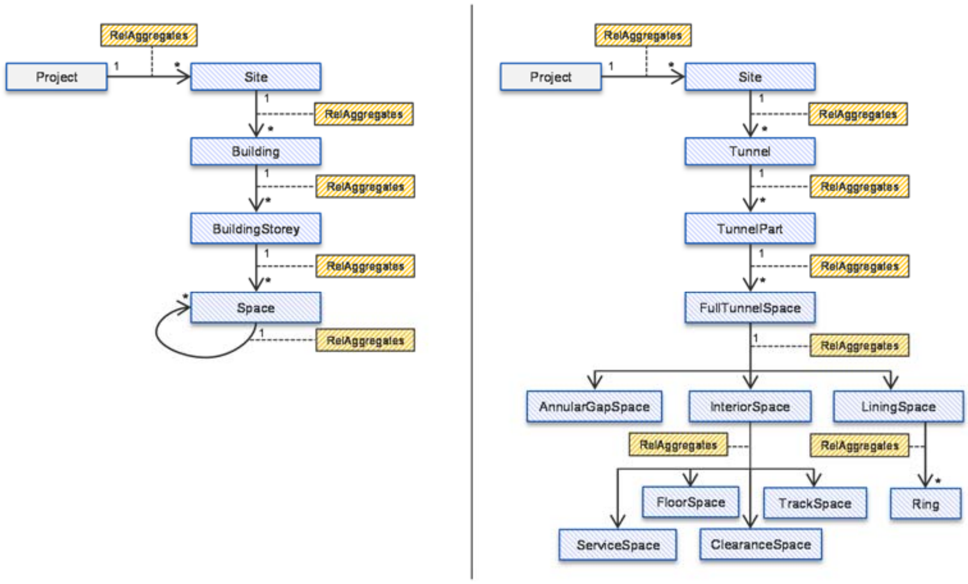
For supporting data exchange in the domain of building design, engineering and construction, the comprehensive data model IFC has been developed over the last decade. As opposed to the CityGML standard, the IFC standard is not defined as an UML or XML schema. Instead, the model is specified using the data modeling language EXPRESS, which forms part of the ISO standard 10303 “STEP – Standard for the exchange of product model data”. Other import particularities of the IFC model are the comprehensive use of objectified relationships and inverse attributes (Eastman 1999, Yang & Eastman 2007, Zhang et al. 2014). The model is very fine-grained and provides more than 600 entities for the detailed description of the semantics and the geometry of buildings and building components.

Only a few researchers have addressed the extension of the IFC model to also cover tunnel facilities. The most important contributions are those by Yabuki et al. who propose an IFC-based model for the description of shield tunnels (Yabuki et al. 2007, Yabuki et al. 2013). However, so far they have not been adopted by official IFC standardization activities.

The main purpose of the IFC model is to support data exchange between different building design and engineering applications, in particular the seamless integration of various simulation and analysis tools. For this reason, large parts of the IFC standard are dedicated to extensive geometry representation capabilities, including different versions of Boundary Representation (BRep), Constructive Solid Geometry (CSG) as well as extrusion and sweep based geometry descriptions.

Like the CityGML model, the IFC data model implements the important principle of a strict separation between the semantic description of the building (space objects, building elements, and their relationships) and its geometric description (Figure 6). Thus, a semantic object can be associated with multiple geometric representations (2D, 3D, BRep, CSG, etc.) (Zhang et al. 2014). This would, in principle, facilitate multi-scale modeling providing different levels-of-detail in the geometric part. However, the integration of the multi-scale concept in the semantic part and the explicit definition of refinement relationships is lacking so far.

For these reasons the current IFC model allows only very limited support for multi-scale modeling. However, as discussed earlier, 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.

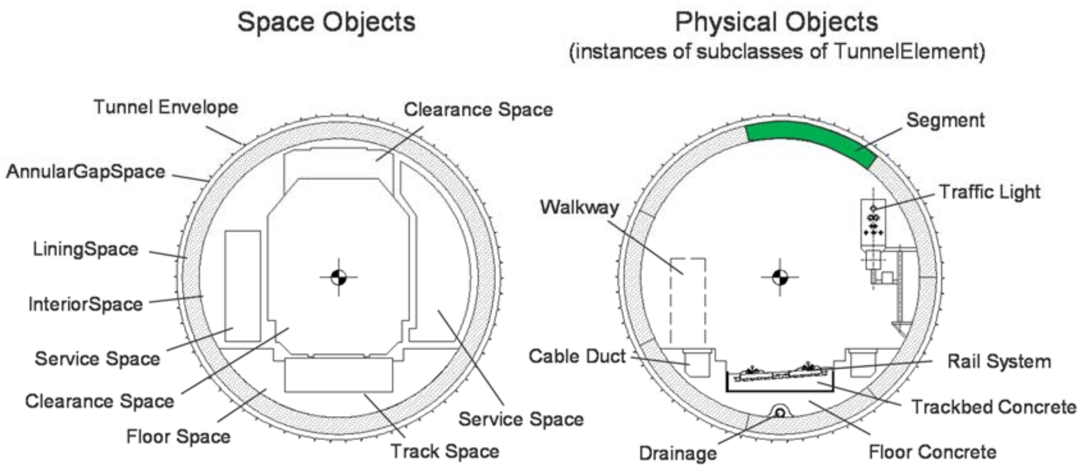


We discuss our approach by extending a product model for shield tunnels, i.e. tunnels which are built by means of Tunnel Boring Machines (TBM), by multi-scale capabilities. However, the presented approach is general and can be applied to other linear infrastructure facilities in the same manner.

5.2 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 and borrows some concepts from CityGML (e.g. Tunnel, TunnelPart, TunnelInstallation). Like the IFC model, the proposed tunnel product model provides a clear separation between semantic objects and the associated geometry. The integration of the semantic model with the procedural geometry description will be discussed in Section 5.4. In this section, we focus on realizing the multi-scale approach for the semantic part. 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 (Stadler & Kolbe 2007; Clementini 2010).

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 7). 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.



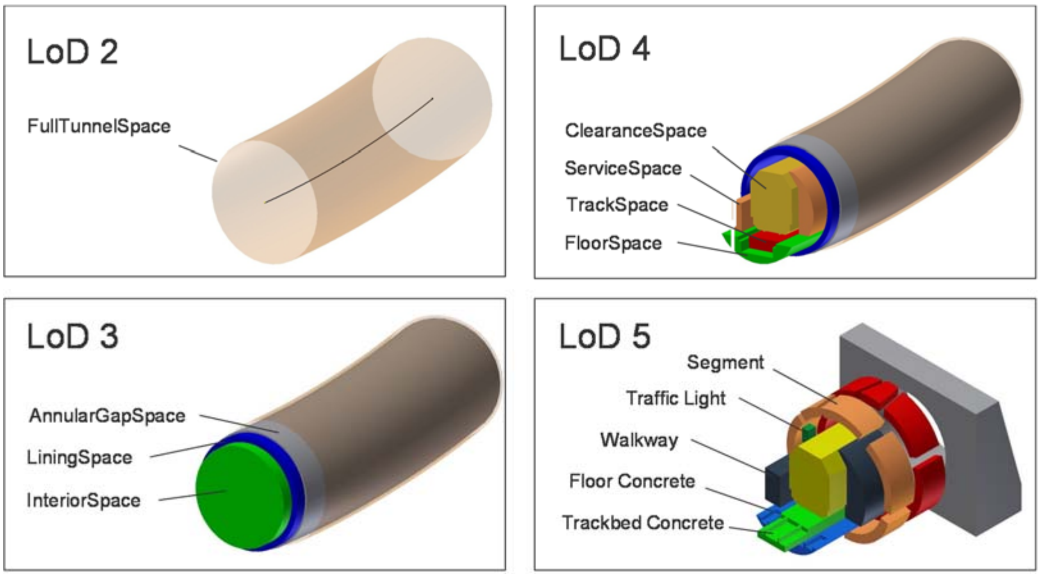
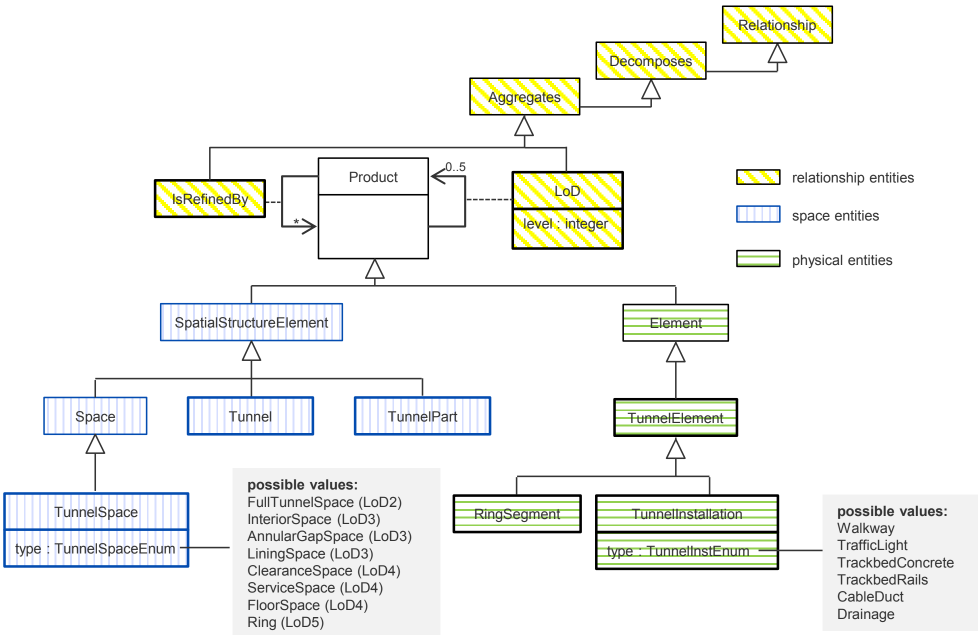
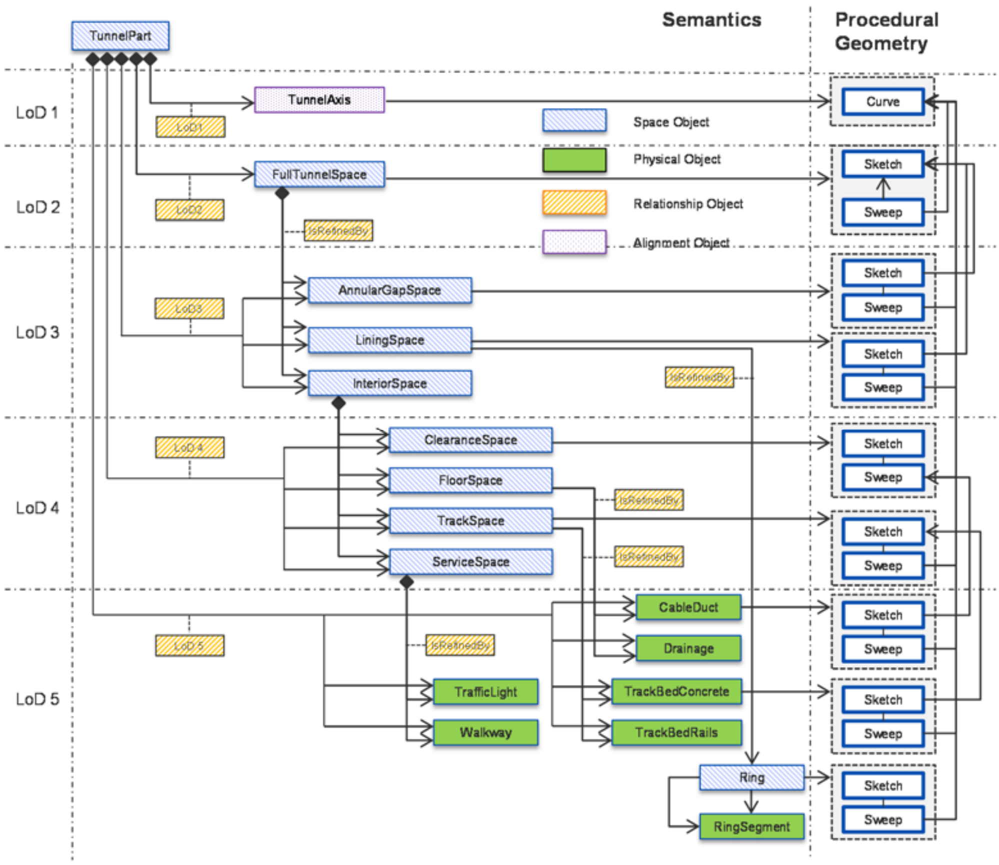


Figure 8 depicts the main components of the tunnel model as 2D cross-sections, while Figure 9 provides a number of 3D views depicting the different LoDs. Figure 10 displays the proposed extensions of the IFC data model to capture shield-tunnel specific elements and provide means for multi-LoD representations. In alignment with the IFC model, the proposed tunnel model extensions consist of space objects and physical objects. Figure 11 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*.

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 comprising the corresponding ring segments which are physical objects 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*.

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. While 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 pre-defined enumeration. This allows for easy maintenance and extendibility.

Following this paradigm we model the diverse spaces depicted in Figure 11, not as individual classes but subsumed by the class *TunnelSpace* which in turn provides a *type* attribute to select from a number of pre-defined 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 pre-defined element types (*Trackbed-Concrete, 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 10 also illustrates the introduction of the level of detail concept into the class model. As discussed above, a dedicated relationship class *LoD* 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 11. Secondly, the relationship class *IsRefinedBy* has been integrated for modeling the refinement relationships as shown in Figure 11.

As described in Section 4, an important part of our multi-scale concept for dynamic modeling processes relies on the use of a procedural geometry description in order to explicitly define dependencies between the geometric entities on the individual levels of detail and thus facilitate automated updating in the case of modifications. The integration of the procedural geometry with the model is described in the next section.

5.3 Procedural Geometry Description

For capturing a procedural model we developed a dedicated data model which consists of two main parts. The first part provides the possibility to describe parametric sketches which include geometric elements as well as the applied dimensional and geometrical constraints. This part has been published in (Ji et al. 2011). The second part enables the storing of the construction history of the geometric model.

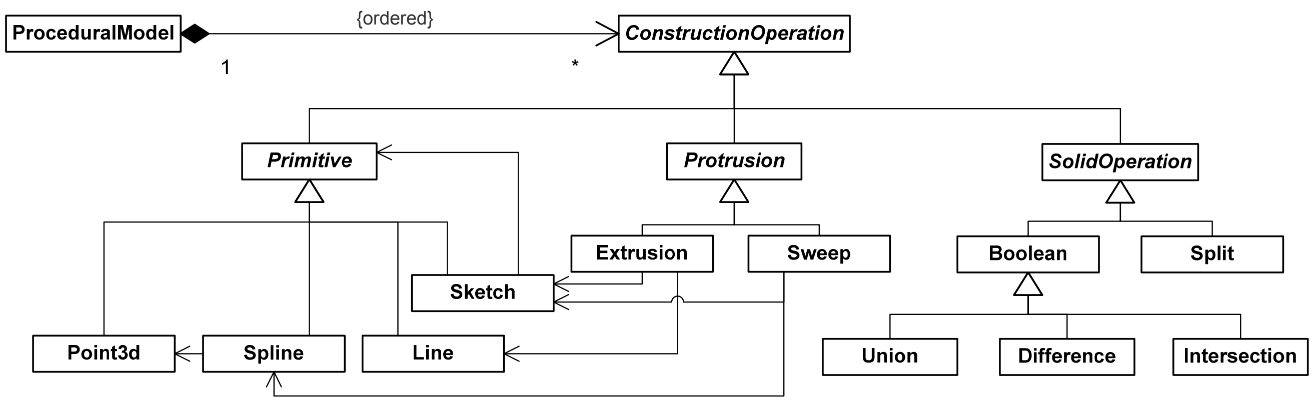
Figure 12 shows a section of the data model developed. The Procedural Model consists of an ordered list of *Construction Operations*. For *Construction Operation* there are a number of defined subclasses, among others *Primitive*, *Sketch*, and *Solid*. Primitive objects are *Points*, *Lines*, *Splines* and *Arcs*, for example. A *Sketch* object comprises primitive objects as well as dimensional and geometric constraints. *Solid* is the superclass of all operations which generate or modify a solid, such as the different protrusions, the Boolean operations, or specific split operations. *Protrusion* is subclassed by *Extrusion* and *Sweep*. Both operations take a sketch as the first argument, while *Extrusion* uses a simple direction for the second argument, and *Sweep* uses a spline as the extrusion path. In both case, the third argument is the protrusion distance.

Using these classes the most important parts of a procedural model can be captured. However not all construction operations provided by modern feature-based CAD systems are included. This applies to more specific construction operations such as chamfering, for example. This is due to the fact that those operations are of minor importance in infrastructure design, as opposed to the design in mechanical engineering.

By defining references from one procedural operation to another (e.g. from the extrusion operation to the extrusion path), the dependencies between the geometric entities in the resulting procedural model are explicitly modeled. This makes it possible to realize the automated update mechanism described in detail in Section 4.

5.4 Combining the multi-scale semantic model with the procedural geometry description

In order to realize a coherent multi-scale product model, the multi-scale semantic model presented in Section 5.2 has to be properly integrated with the procedural geometry description introduced above. The instance diagram provided by Figure 11 illustrates how this has been realized.



For the geometry representation of the individual elements (spaces and physical objects), a procedural description is used. The individual operations of the procedural description can refer to operations or geometric entities used on lower levels. One example is the tunnel axis which acts as the LoD1 representation: the assigned curve is used as the path for creating the extrusion geometry of all longitudinal objects on the finer LoDs. Another example is the sketch-based creation of space profiles on finer levels from coarser ones using offset operations applied to the sketch elements.

6. PROTOTYPICAL IMPLEMENTATION

The developed concept of multi-scale tunnel modeling and cross-LoD consistency preservation has been implemented on the basis of the parametric CAD system Autodesk Inventor. The prototype system provides the possibility to create multi-scale shield tunnel models using the pre-defined semantic objects introduced in Section 4. In addition it allows to explicitly define dependencies between the geometric entities on different LoDs by means of the built-in parametric capabilities. As soon as a modfication is performed, all depending objects are automatically updated. Figure 13 shows four screenshots of the developed system depciting different LoDs.

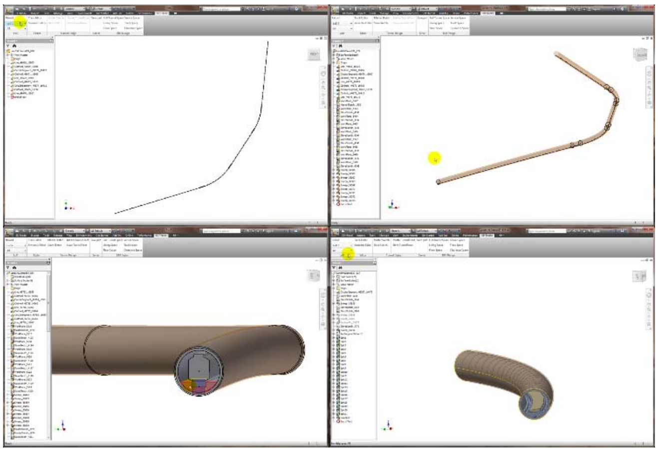
Product Model for Shield Tunnels		CityGML	
LoD	Concept	LoD	Concept
-	TunnelPart	0,2,4	Tunnel / TunnelPart
1	TunnelAxis	0	TransportationComplex with LoD0Network (gml::GeometricComplex)
2	FullTunnelSpace	2	_AbstractTunnel with LoD2Solid (gml::_Solid), and/or WallSurface, GroundSurface, Roofsurface (gml::MultiSurface)
3	InteriorSpace	4	HollowSpace with LoD4Solid InteriorWallSurface, CeilingSurface, Floorsurface
3	LiningSpace, AnnularGapSpace	-	-
4	ClearanceSpace, ServiceSpace	4	HollowSpace with LoD4Solid (gml::_Solid)
4	FloorSpace, TrackSpace	-	-
5	Walkway	4	HollowSpace with LoD4Solid (gml::_Solid)
5	TrafficLight	4	IntTunnelInstallation (gml::_Geometry)
5	FloorConcrete, TrackbedConcrete	4	FloorSurface, TrafficArea both with Lod4MultiSurface (gml::MultiSurface)
5	CableDuct, Drainage	4	IntTunnelInstallation (gml::_Geometry)
5	Ring	-	-
5	RingSegment	4	InteriorWallSurface, CeilingSurface, Floorsurface for surfaces pointing to the interior of the tunnel, WallSurface, GroundSurface, Roofsurface for surfaces pointing to the exterior of the tunnel (gml::MultiSurface)

Table 1. Mapping entities of the proposed IFC-based product model for shield tunnells to the corresponding CityGML entities

The suitability of the developed extensions of the IFC data model was proved by developing an export module for the Inventor-based prototype system and an corresponding import modules for the parametric CAD system Siemens NX. Using these modules, it was possible to successfully transfer multi-scale tunnel models between these two systems inluding the defined dependencies among their entities. In addition, the developed product model was successfully employed in simulating the tunneling process (Stascheid et al. 2013).

7. GEOMETRIC AND SEMANTIC MAPPING BETWEEN THE IFC-BASED MULTI-SCALE MODEL AND CITYGML

In several phases of the planning process, it is necessary to integrate the 3D planning model into its geographic context, the latter being provided by a Geographic Information Systems (GIS). Examples are visualizing the model for the purpose of stakeholder involvement, environmental impact studies, testing collisions with existing infrastructure and updating a topographic information system after the completion of the structure (Fedra 2002, Lopes & Dias 2006, Carozza et al. 2014). Integrating the 3D model created by a parametric



CAD system into a GIS means to transform both the geometry and the semantics into a data structure which can be interpreted by a GIS. We show that this transformation is feasible using CityGML as the GIS data structure.

On the one hand, the procedural geometry model has to be transformed into an explicit (BRep) representation. This is due to the fact that geometry models following the generative modeling paradigm as it is applied by the procedural geometry model, are not supported by GIS or spatial database management systems due to various well-known reasons (cf. Kolbe, Plümer 2004), e.g. the lack of spatial indexes for geometries of such kind. Using CityGML as a target schema, this transformation process consists of two major steps. First, the procedural geometry has to be transformed into a BRep representation which can be achieved using geometric modeling kernels like Parasolid or OpenCASCADE. – of course a loss of precision has to be observed due to the sampling of the procedural models. While in many 3D GIS applications the consistency of 3D solid geometries is difficult to ensure (see Ledoux 2013), in our case this does not state a problem, because the 3D BRep representations are always automatically derived from the procedural and parametric descriptions.

Second, the BRep solids resulting from transforming the volumetric objects have to be converted to surfaces which is not trivial as the surfaces of one and the same solid might be associated with different semantic objects of the CityGML schema, e.g. the solid resulting from a RingSegment object of the procedural model must be converted to surfaces associated with InteriorWallSurface, CeilingSurface, Floorsurface for surfaces pointing to the interior of the tunnel and WallSurface, GroundSurface, Roofsurface for surfaces pointing to the exterior of the tunnel.

On the other hand, a semantic transformation between our product model for shield tunnels and the respective concepts provided by CityGML must be carried out. The following table describes a semantic mapping between the concepts of the IFC-based product model and CityGML for the different LoD definitions.

Table 1 shows that concepts from LoD3-5 from the product model are mapped to CityGML LoD4 concepts. Although CityGML permits the modeling of free space inside a tunnel using the *HollowSpace* concept, only those types of space concepts from the product model are mapped to CityGML which are not occupied by physical objects. Therefore the *ClearanceSpace* and *ServiceSpace* is mapped whereas the *FloorSpace*, *TrackSpace, Ring, AnnularGapSpace* and *LiningSpace* are not. During the conversion of the geometry from procedural geometry to BRep, in some cases, the geometry types have to be changed from solids into multi-surfaces, e.g. when transforming an *InteriorSpace* object into *InteriorWallSurface* objects. In other cases only parts of the geometry can be mapped, e.g. the surface of *FloorConcrete* and *TrackbedConcrete* is mapped to *FloorSurface*. Table 1 shows the geometry classes of CityGML which are used for the geometry mapping. LoD3 is not relevant as the product model currently does not allow for specifying openings in the outer shell of the tunnel.

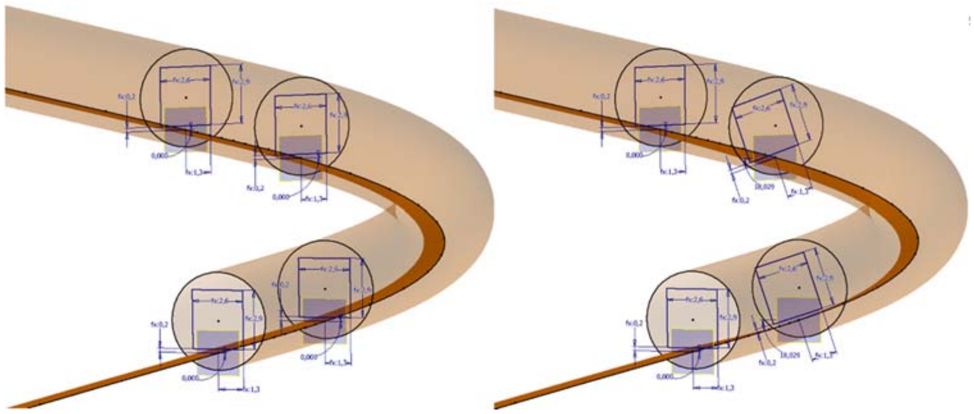
Since a more differentiated level of detail concept for indoor environments is under discussion for the next version of CityGML, it may be possible in future to also map *FloorSpace* and *TrackSpace*.

The mapping is necessary in order to be able to apply available GIS analysis tools for evaluating the tunnel design with respect to geographical criteria. To this end, an advanced geo web service was developed and made available from the design systems discussed in Section 6. It allows to initiate GIS analysis workflows by converting the IFC based tunnel model into the corresponding CityGML representation and sending it to the analysis web service. The result of the analysis can be visualized using either the CAD system or a GIS (Steuer et al. 2013). This gives the planner a direct access to GIS analysis functionality from the geometric modeling system at use.

Using the functionality of advanced geo web services, it is possible to perform complex analysis workflows. A good example is the technical specification of interoperability relating to 'safety in railway tunnels' in the trans-European conventional and high-speed rail system as demanded by the European Communities. Corresponding workflows will consider more criteria like ownership of land at the ground surface, distance to safe areas and accessibility for rescue teams.

8. CASE STUDY: SECOND MAIN SUBWAY TRACK IN MUNICH, GERMANY

In order to prove the suitability of the developed approach we conducted a real-world case study. As case study project we chose the second main subway track in Munich which is currently under planning. On the basis of conventional 2D plans of the project, we developed a multi-scale model of the shield tunnels to be constructed in the frame of this project.



The planning of this second main subway track in Munich started ten years ago and is almost finished now. The construction of the second track is announced to start in 2014 and planned to be finished in 2019. This second track is designed as a twin shield tunnel with a length of about seven kilometers, which connects the two most important inner-city train stations Hauptbahnhof and Ostbahnhof. The already existing subway lines confront the engineers with a highly complex planning task, which culminates in three new stations, which are to be built about 40m below ground.

Figure 13 depicts the tunnel in the different LoDs defined. Figure 14 depicts a LoD 4 model of the shield tunnel connecting the stations Hauptbahnhof and Marienhof. Due to its length of about two kilometers, an escape shaft is placed in its middle.

The project was originally planned by means of a conventional 2D drawing-based approach. The resulting drawings were provided by the engineering planning office in charge and used to re-model major parts of the complete project by means of the multi-scale approach presented in Section 4. The resulting 3D parametric, multi-scale model shows significant advantages compared to the static 2D representation. This includes:

- full 3D representation: clash detection can be performed, consistent 2D plans can be derived
- multi-scale representation: the model can be visualized and modified on different levels of abstraction
- flexibility: modifications on coarser levels are directly propagated into all finer levels

The shield tunnels were modeled using the LoD approach presented in Section 4 using the software prototype introduced in Section 6. Due to large extent of the shield tunnels, the model has been subdivided into several parts (submodels). To maintain the geometric consistency across these submodels, dedicated consistency preservation mechanisms (Jubierre and Borrmann, 2013) have been applied.

The LoD approach allows engineers to adapt the track of the tunnel and the escape shafts in a very flexible and dynamic manner. A prominent use case is that the planned train speed determines the cant of the tunnel interior geometry, in particular the cant of the super-elevation stripe and the loading railway gauge. If the engineer changes the parameter regarding the permitted train top velocity the model automatically adapts to this changed parameter (Figure 15). Additionally, this provides a direct clash detection mechanism, since an overvalued velocity produces a visible intersection of the loading railway gauge and the inner tunnel hull.

The 3D city model of Munich was used as planning context by importing parts of the corresponding CityGML models into the prototype modeling system. Using advanced geo web services the model of a planned escape shaft can be analyzed with regards to a surrounding city model. The result of the analysis, i.e. the affected buildings, streets and parts of the sewer system, are visualized using both the CAD system as well as a GIS (Steuer et al. 2013). This allows the planner a direct access to GIS analysis functionality from the geometric modeling system at use.

9. CONCLUSIONS

The planning of large infrastructure facilities such as inner-city subway tracks requires the consideration of widely differing scales, ranging from the kilometer scale for the general routing of the track down to the centimeter scale for detailed design of connection points.

Multi-scale representations are 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, so-called building information models, 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 requires careful consideration of the highly dynamic planning processes which result in frequent modifications of the data stored in the BIM. The approach implemented by CityGML, which relies on storing an independent representation for each LoD, is not applicable for the design phase, since it bears the high risk of introducing inconsistencies when modifications are not simultaneously performed for all LoDs.

The paper has presented a methodology which enables the creation of multi-scale models where the individual levels of detail are inherently consistent with one another. The core concept is the definition of dependencies between geometry objects on different LoDs by making use of procedural geometry representations. The concept is based on the application of parametric modeling techniques. Applications that are capable of interpreting and processing procedural geometry are able to automatically preserve the consistency of the multi-scale model by propagating changes on geometric objects to all dependent representations and updating them accordingly.

To illustrate the integration of this concept with an IFC-based data model we discussed the development of a multi-scale product model for shield tunnels. The semantic part of the model implements the multi-scale approach by providing explicit LoD objects and making use of the space aggregation hierarchy for modeling refinement relationships. The geometric part associates the individual semantic objects with a procedural description which is defined across multiple LoDs. The resulting product model provides geometric-semantic coherence and at the same time mechanisms for automated consistency preservation. It thus responds to the particular demands of multi-scale representations in the context of highly dynamic planning processes.

Although we focused on applying the LoD concept to shield tunnels in this paper, it is generic and applicable to a wide range of linear infrastructure facilities, including cut-and-cover tunnels and tunnels created through drilling and blasting. A challenge to be tackled in the future is the consistent application of the LoD concept on non-linear infrastructure facilities, such as underground stations and switch boxes. Another open issue is the manual definition of dependencies between the geometric entities on different LoDs which is very flexible, but laborious and error-prone. We plan to develop design automation techniques based on graph replacement methods to achieve a higher degree of automation while preserving the desired flexibility. This will enable to capture engineering knowledge in a comprehensive manner, including aspects of functional coherence.

In a modern planning process, it is desirable to closely interrelate planning and impact analysis. This way, the design model can be analyzed in the context of its spatial environment, e.g. in an existing 3D city model, in order to study the environmental impact or to test collisions with existing infrastructure. Therefore, it is necessary to convert the procedural model into an explicit representation (BRep with absolute world coordinates), preferably in an automatic way and under preservation of the geometric and semantic information. We have shown that this process can be achieved using geometric modeling kernels.

Regarding the semantic mapping of the concepts introduced within the product model to CityGML we have shown that most of the concepts can be mapped. Therefore it is possible to use CityGML to integrate the tunnel design model with the environment model within one consistent framework

In order to avoid complex geometric operations during the transformation process described in section 6, future research should focus on extending CityGML with regard to volumetric objects. .

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

Bettig, B., Shah, J. (2001). Derivation of a standard set of geometric constraints for parametric modeling and data exchange, Computer-Aided Design, 33 (1), pp. 17–33

Carozza, L., Tingdahl, D., Bosché, F., van Gool, L. (2014): Markerless Vision-Based Augmented Reality for Urban Planning, Computer-Aided Civil and Infrastructure Engineering 29 (1), 2–17

Clementini, E. (2010): Ontological impedance in 3D semantic data modeling, 5th International 3D GeoInfo Conference, Berlin, Germany

Döllner, J., Buchholz, H. (2005): Continuous level-of-detail modeling of buildings in 3D city models, Proc. of the 13th annual ACM Int. Workshop on Geographic information systems.

Eastman, C.M (1999): Building Product Models: Computer Environments Supporting Design and Construction: CRC Press.

Eastman, C., Teicholz, P., Sacks, R. and Liston, K. (2008) BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, John Wiley & Sons, Inc., New Jersey.

Fedra, K., Winkelbauer, L. (2002): A Hybrid Expert System, GIS, and Simulation Modeling for Environmental and Technological Risk Management, Computer-Aided Civil and Infrastructure Engineering 17 (2), 131–146

Forberg, A. (2007): Generalization of 3D building data based on a scale-space approach. ISPRS Journal of Photogrammetry & Remote Sensing 62 (2007) 104–111

Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H. (2012), OGC City Geography Markup Language (CityGML) Encoding Standard, v2.0. OGC Doc. No. 12-019. http://www.opengeospatial.org/standards/citygml (06.05.2013)

ISO (2005). International Organization for Standardization (ISO): STEP Part 55: Integrated generic resource: Procedural and hybrid representation. ISO 10303-55:2005.

Ji, Y., Beetz, J., Bonsma, P., Bisbet, N., Katz, C., Borrmann, A. (2011a). Integration of Parametric Geometry into IFC-Bridge. In: Proc. of the 23th Forum Bauinformatik, Cork, Ireland

Ji, Y., Borrmann, A., Obergrießer, M. (2011b): Towards the Exchange of Parametric 3D Bridge Models Using a Neutral Data Format. In: Proc. of the ASCE Int. Workshop on Computing in Civil Engineering. Miami, USA

Ji, Y., Borrmann, A., Beetz, J., Obergrießer, M. (2013): Exchange of Parametric Bridge Models using a Neutral Data Format. ASCE Journal of Computing in Civil Engineering 27 (6), 593-606

Jubierre, J. R., Borrmann, A. (2013): Cross-submodel consistency preservation in multi-scale engineering models, In: Proc. of the 14th Int. Conf. on Civil, Structural and Environmental Engineering Computing (CSC2013), Cagliari, Italy

Koch, C., Firmenich, B. (2011): An approach to distributed building modeling on the basis of versions and changes. Advanced Engineering Informatics, 25 (2), pp. 297-310

Kolbe, T. H. (2008): Representing and Exchanging 3D City Models with CityGML, In: Proceedings of the 3rd International Workshop on 3D Geo-Information, Seoul, Korea

Kolbe, T. H.; Gröger, G. (2003): Towards unified 3D city models. In: Proceedings of the Joint ISPRS Commission IV Workshop on Challenges in Geospatial Analysis, Integration and Visualization II in Stuttgart.

Kolbe, T. H; Plümer, L. (2004): Bridging the Gap between GIS and CAAD, In: GIM International, No. 7

Ledoux, H., 2013. On the Validation of Solids Represented with the International Standards for Geographic Information. Computer-Aided Civil and Infrastructure Engineering, 28(9), 693–706

Lee, G., Sacks, R., Eastman, C. M. (2006): Specifying parametric building object behavior (BOB) for a building information modelling system. Automation in Construction 15(6) 758-776

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 Support Group).*

Lopes, A.C., Dias, J.M.S. (2006): Integration of Geo-Referenced Data for Visual Simulation in Location-Based Mobile Computing. Computer-Aided Civil and Infrastructure Engineering, 21(7), 514–529

McMaster, R. B., Shea, K. S. (1992). Generalization in digital cartography. Washington, DC: Association of American Geographers.

Meng, L., Forberg, A. (2007) 3D Building Generalisation. In: Mackaness, W., Ruas, A. and Sarjakoski, T. (Eds): Challenges in the Portrayal of Geographic Information: Issues of Generalisation and Multi Scale Representation. Chapter 11, Elsevier Science Ltd

Mun, D., Han, S., Kim, J., Oh, Y. (2003): A set of standard modeling commands for the history-based parametric approach. Computer-Aided Design, 35 (13), pp. 1171–1179

Obergriesser, M., Euringer, T., Borrmann, A., Rank, E. (2011): Integration of geotechnical design and analysis processes using a parametric and 3D-model based approach. In: Proc. of the 2011 ASCE International Workshop on Computing in Civil Engineering. Miami, FL, USA.

van Oosterom, P., Schenkelaars, V. (1995): The development of an interactive multi-scale GIS. International Journal of Geographical Information Systems 9(5): 489-507.

Ohori, A., Biljecki, K. F., Stoter, J. and Ledoux, H. (2013), Manipulating higher dimensional spatial information, In: Proc. of the 16th AGILE Conference on Geographic Information Science, Leuven, Belgium

Pratt, M. J., Anderson, B. D., Ranger, T. (2005): Towards the standardized exchange of parameterized feature-based CAD models. Computer-Aided Design, 37 (12), pp. 1251–1265

Sester, M., Brenner, C. (2009): A vocabulary for a multiscale process description for fast transmission and continuous visualization of spatial data, Computers & Geosciences, 35(11), pp. 2177–2184

Steuer, H., Donaubauer, A., Kolbe, T. H., Flurl, M., Mundani, R.-P., Rank, E. (2013): Planning Inner-City-Railway-Tracks: Dynamic Integration of Geospatial Web Services in a Collaborative Multi-Scale Geometry Modelling Environment, In: Proc. of the EG-ICE Workshop on Intelligent Computing in Engineering, Wien, Austria, 2013

Stadler, A., Kolbe, T. H. (2007): Spatio-semantic coherence in the integration of 3D city models In: Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede, The Netherlands

Stascheit, J., Koch, C., Hegemann, F., König, M., Meschke, G. (2013): Process-oriented numerical simulation of mechanized tunneling using an IFC-based tunnel product model, Proc. of 13th Int. Conf. on Construction Applications of Virtual Reality, London, UK

Steinitz, C. (2012): A Framework for Geodesign: Changing Geography by Design, ESRI Press, Redlands, CA.

Stiteler, M. (2004): Construction History and Parametrics: Improving affordability through intelligent CAD data exchange. Tech. rep., CHAPS Program Final Report, Advanced Technology Institute, 5300 International Boulevard, North Charleston, SC 29418, USA

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

Yang, D., Eastman C.M. (2007): "A Rule-Based Subset Generation Method for Product Data Models." Computer-Aided Civil and Infrastructure Engineering 22 (2), 133–148

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

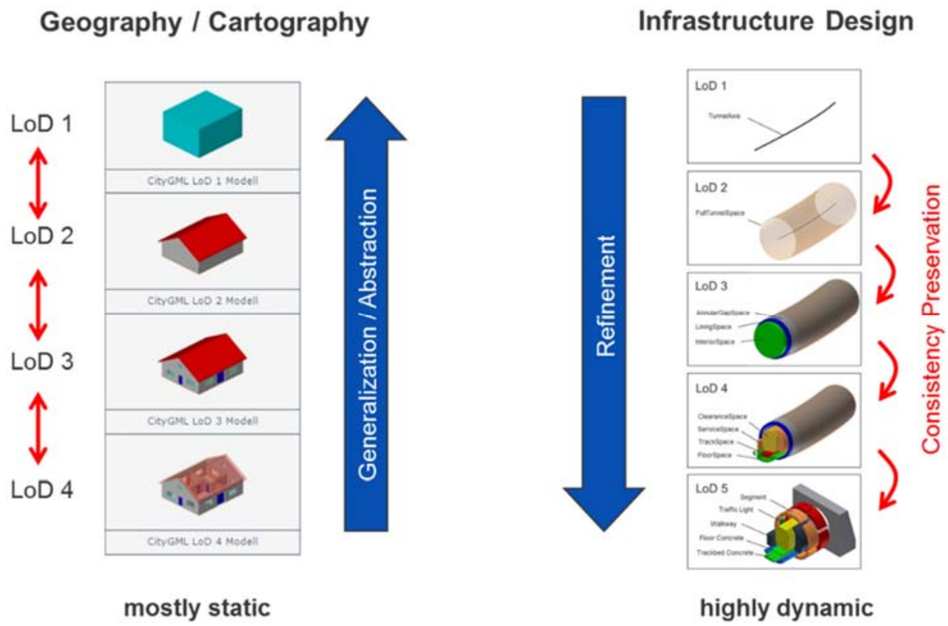


Figure 1: There are important differences in the LoD approaches taken by GIS applications and infrastructure design. In cartography, usually fine-grained data is captured and generalized to create coarser levels of detail (bottom-up approach), whereas in infrastructure design, planners start with a coarse representation and add more and more details (top-down approach).

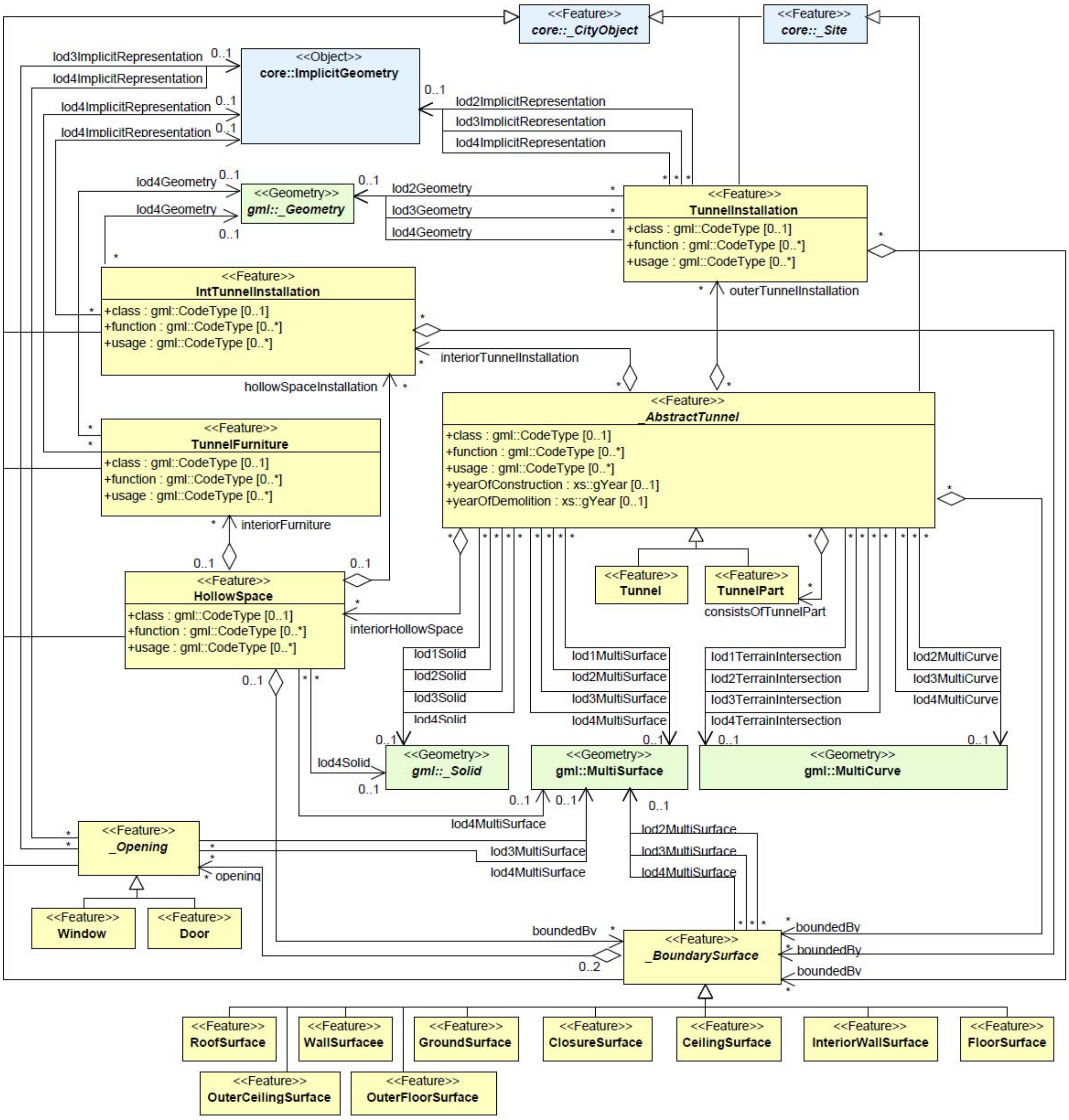


Figure 2: The CityGML data model for the multi-scale representation of tunnels (Source: (Gröger et al. 2012))

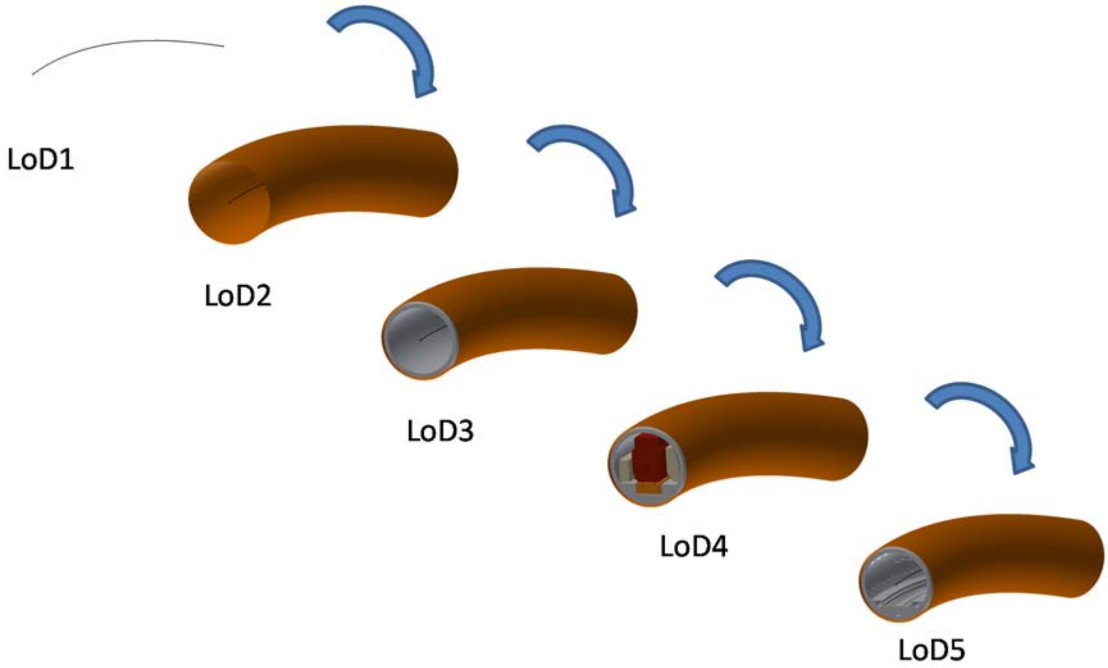


Figure 3: Illustration of the five different levels of detail defined for designing a tunneled carriageway. The concept of procedural modeling permits the explicit definition of dependencies between geometric elements on different levels of detail

```
1.   CREATE SPLINE          LoD1
2.   CREATE SKETCH          LoD1
------------------------------------
3.   CREATE SWEEP           LoD2
4.   CREATE SKETCH          LoD2
------------------------------------
5.   CREATE SWEEP           LoD1
6.   BOOLEAN UNION          LoD1
------------------------------------
7.   CREATE SKETCH          LoD2
8.   DEFINE CONSTRAINT      LoD2
9.   CREATE SWEEP           LoD2
------------------------------------
10.  CREATE SWEEP           LoD3
11.  BOOLEAN DIFFERENCE     LoD3
12.  CREATE EXTRUSION       LoD3
13.  ...
```

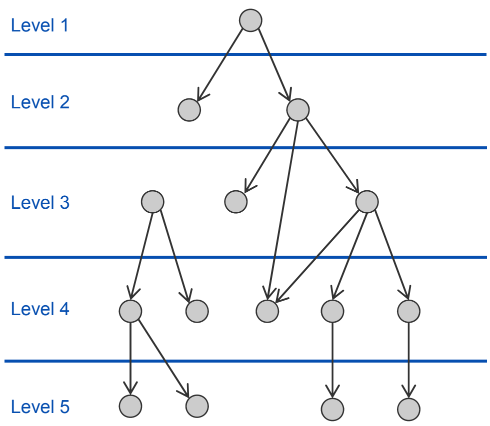


Figure 4: Left hand side: Illustration of a construction history captured by a procedural model. The switches between the individual LoDs are explicitly triggered by the user. Right hand side: The cross-LoD dependency graph resulting from referencing entities on lower levels for construction operations on higher levels.

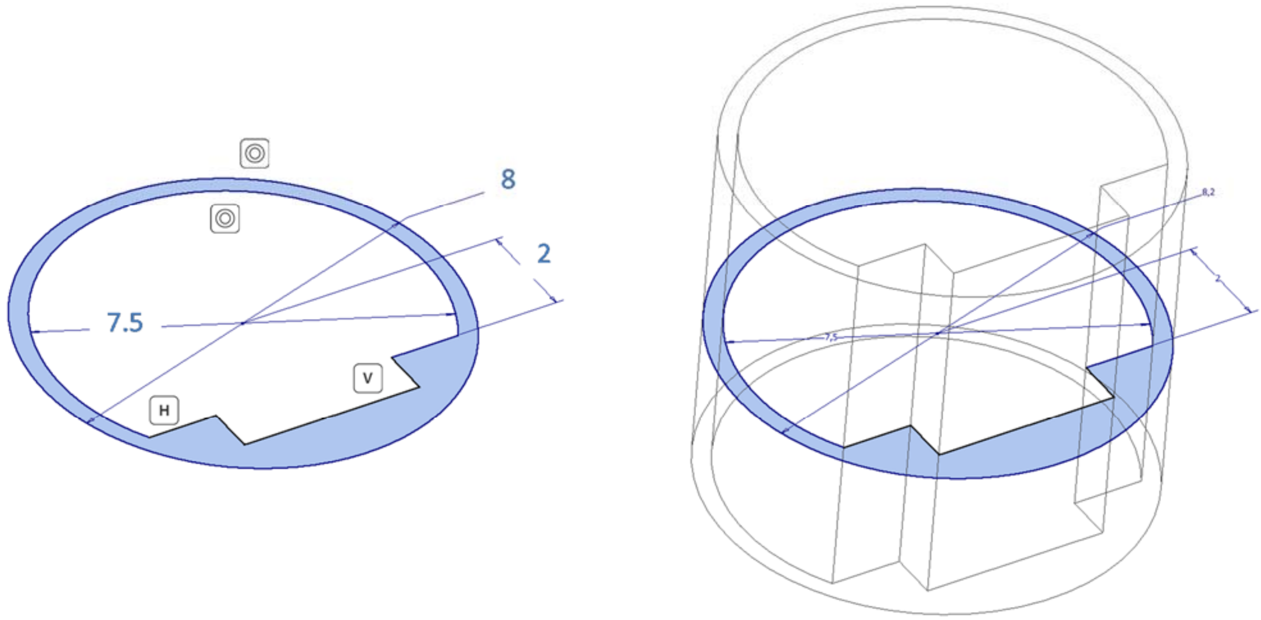


Figure 5: A sketch definition and the subsequent application of an extrusion operation

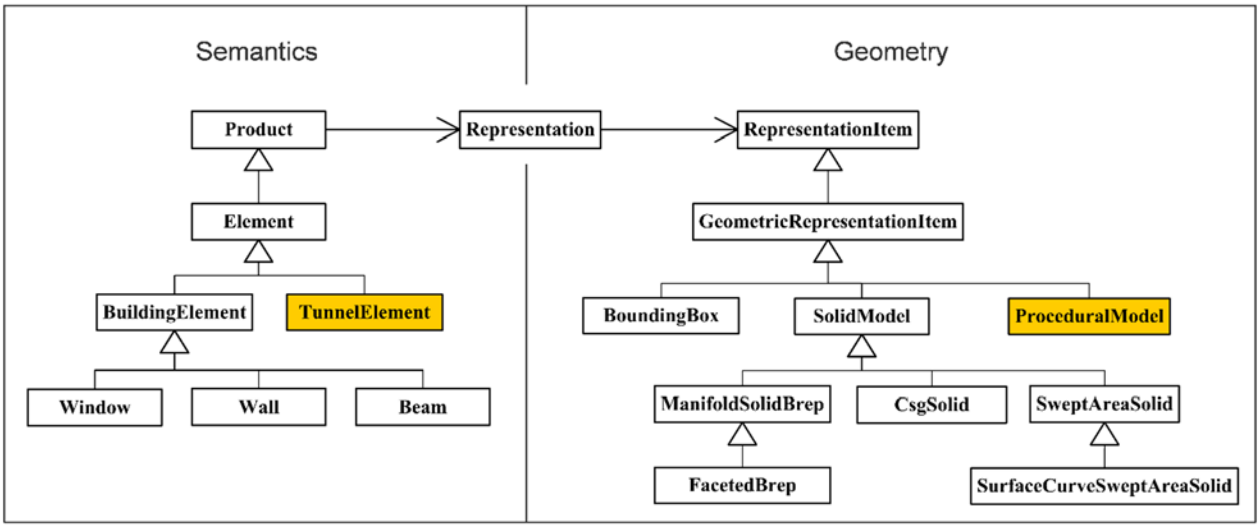


Figure 6: Separation of geometry and semantics in the IFC data model. Left: Small section of the semantic model; Right: Subset of the different geometry representations; Semantics and geometry can be combined through a flexible linkage mechanism using the Representation entity. The highlighted entities represent the proposed extensions. The prefix 'Ifc' has been omitted.

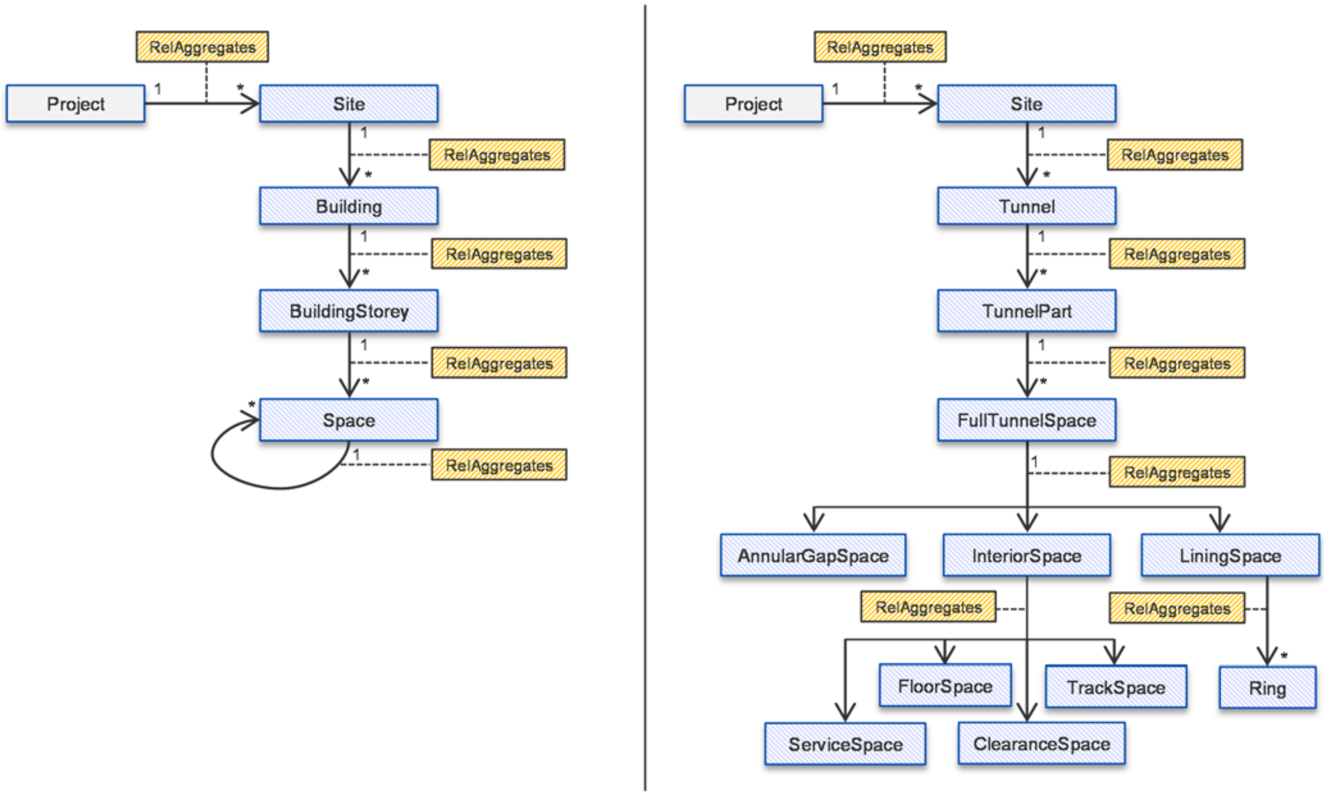


Figure 7. Left: Modeling of space aggregation hierarchies in the IFC standard, Right: Usage of the space aggregation concepts in the proposed extension. The prefix 'Ifc' is omitted.

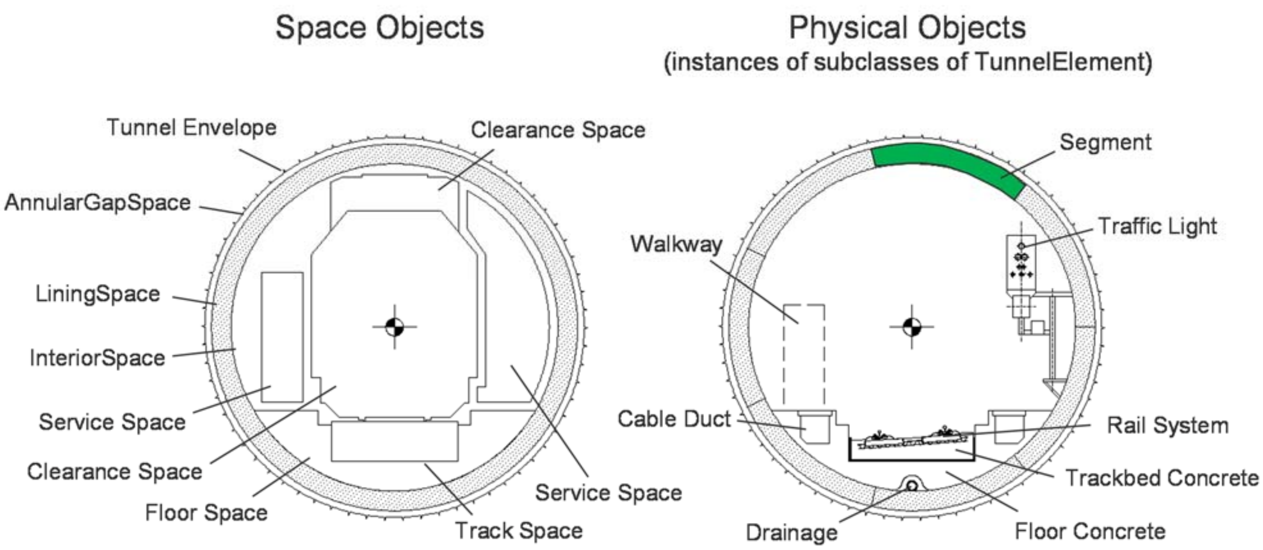


Figure 8. A tunnel cross-section depicting the individual spaces (left) and physical elements (right) of the proposed multi-scale product model

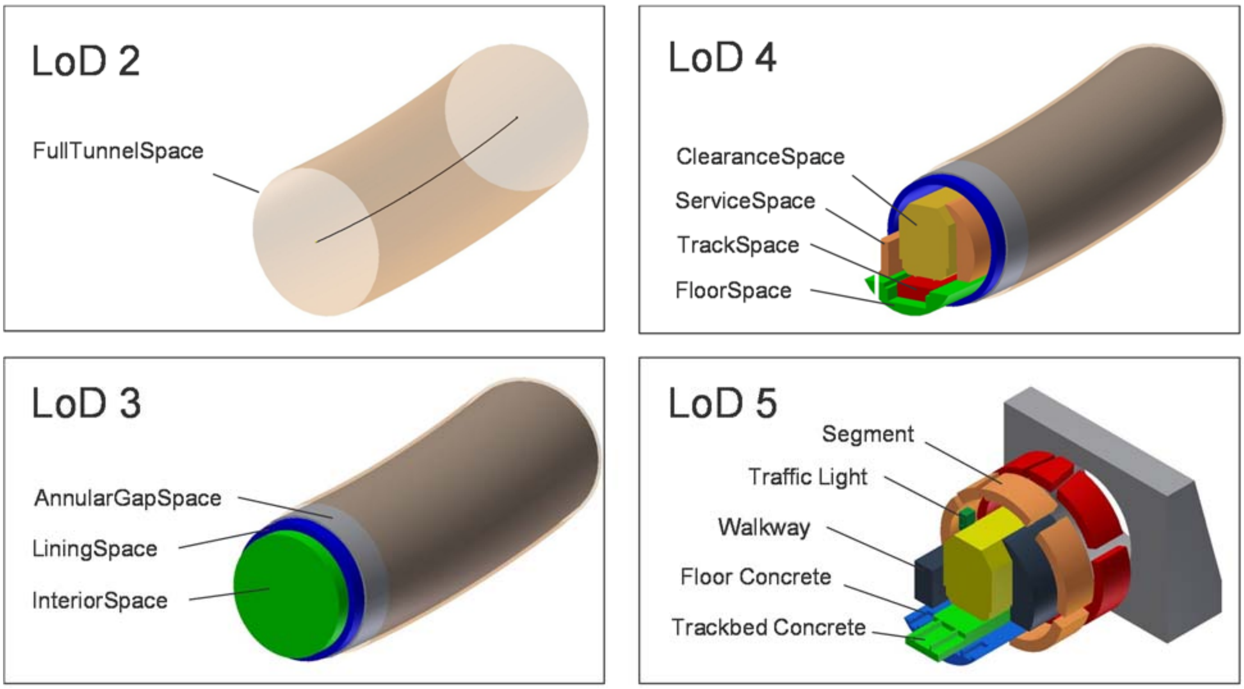


Figure 9. A 3D representation of the different LoDs of the multi-scale tunnel product model

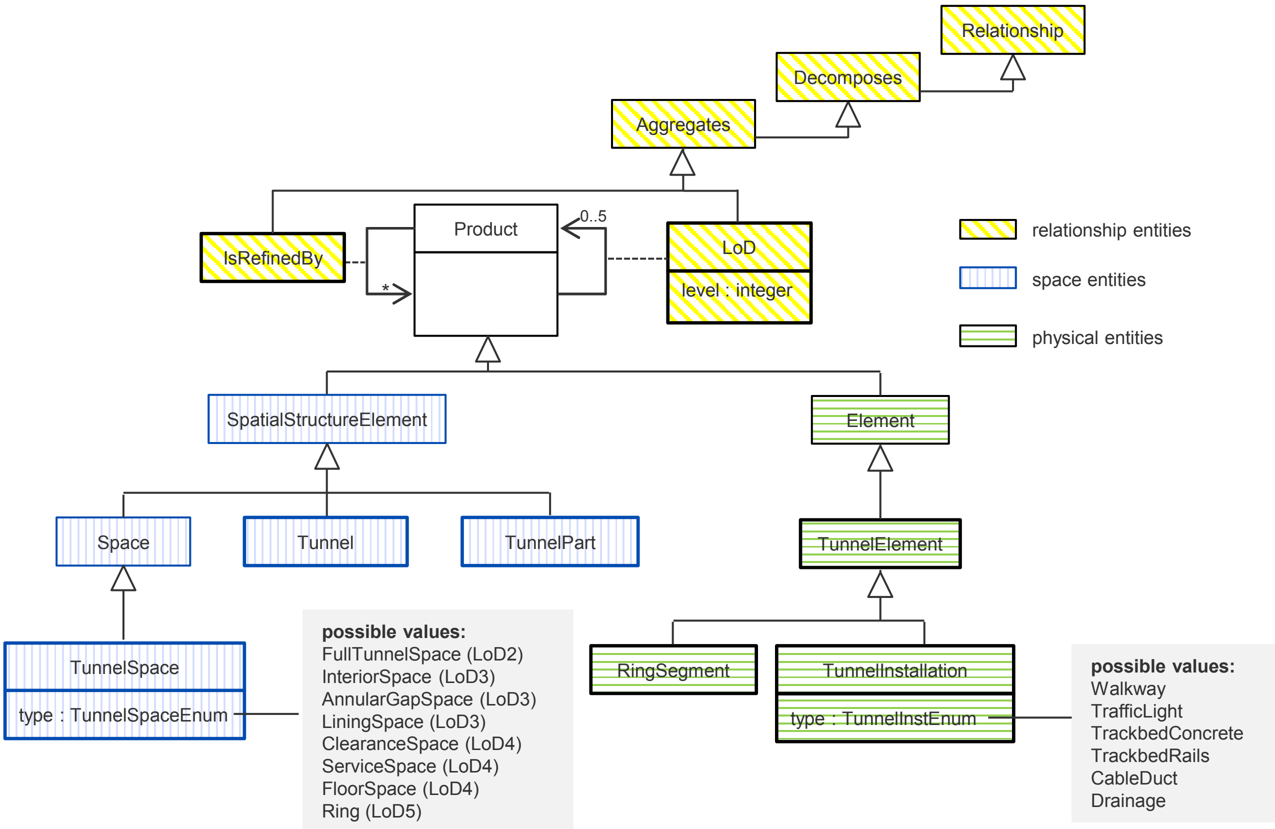


Figure 10: UML Class Diagram depicting the introduced relationship classes *IsRefinedBy* and *LoD*, as well as the classes *TunnelSpace* and *TunnelInstallation* which are used to model tunnel-specific spaces and installations. Classes

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.

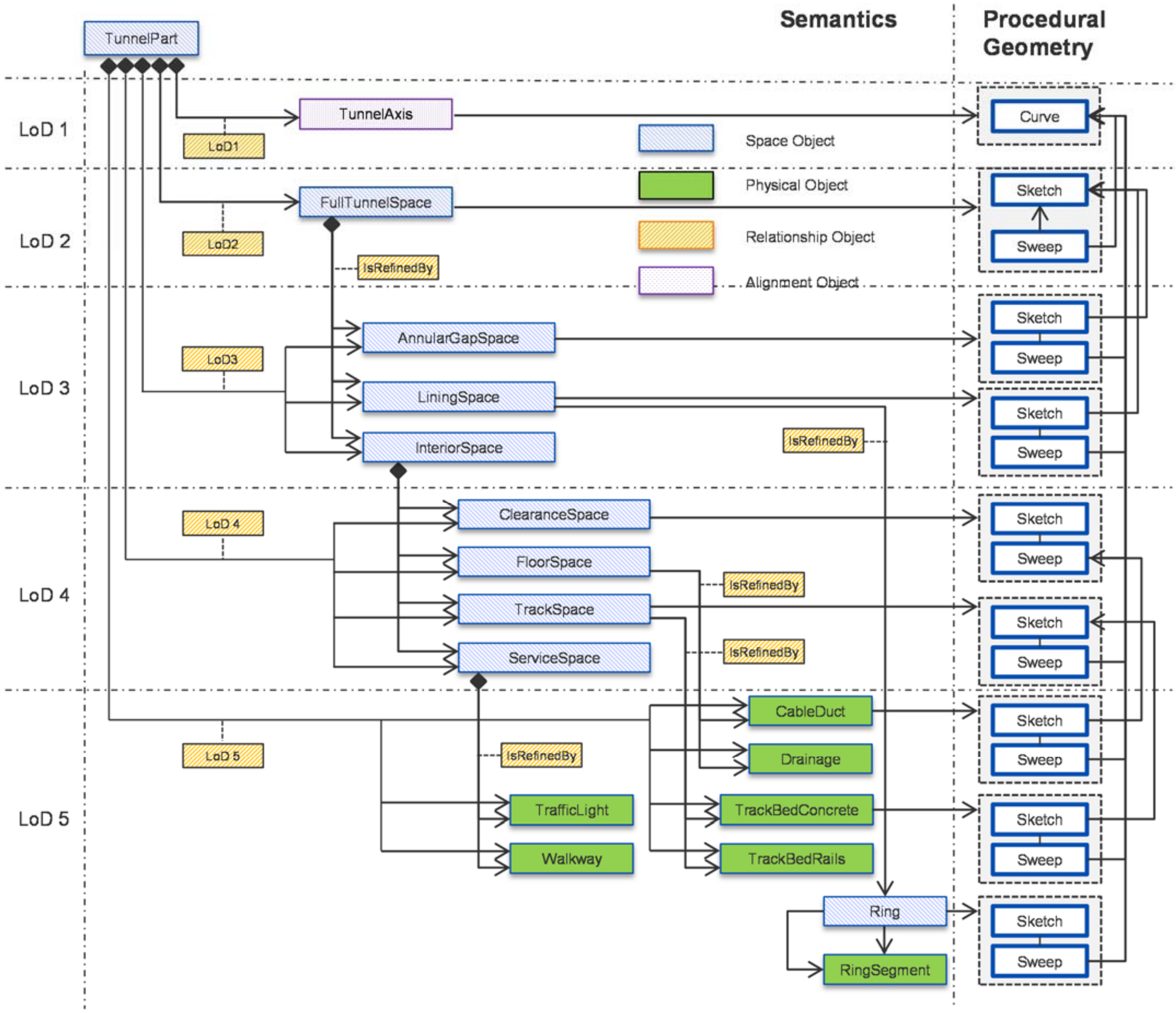


Figure 11. UML Instance diagram depicting the semantic and the geometric part of the proposed shield tunnel product model incorporating a multi-scale representation. The TunnelPart object is associated with the different representations via dedicated LoD objects. The proposed tunnel models consists of 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. The right-hand side depicts the integration of the procedural geometry representation. The procedural model defines dependencies between the geometric representations at the different LoDs, thus providing a means for preserving consistency across the levels.

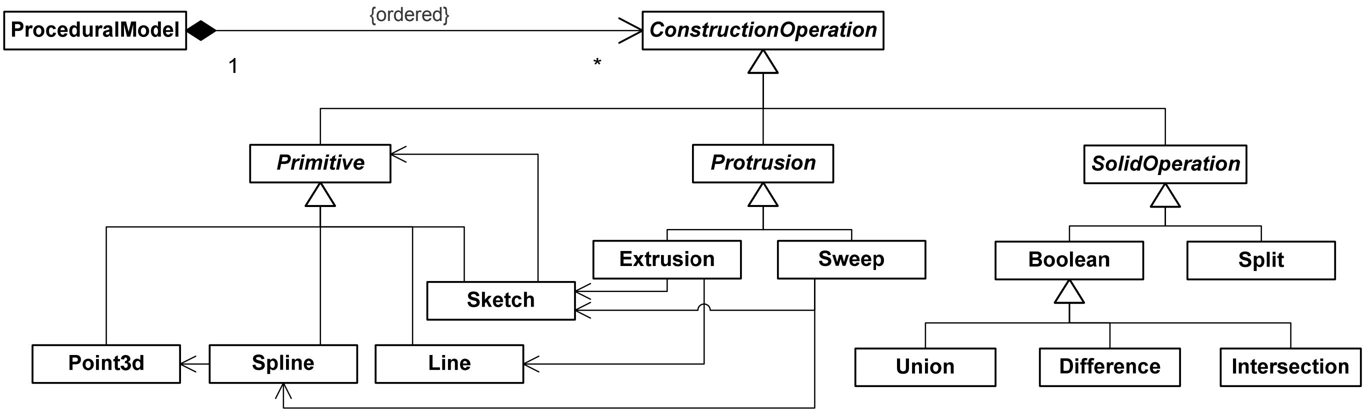


Figure 12: Section of the developed data model for capturing a procedural model (UML diagram)

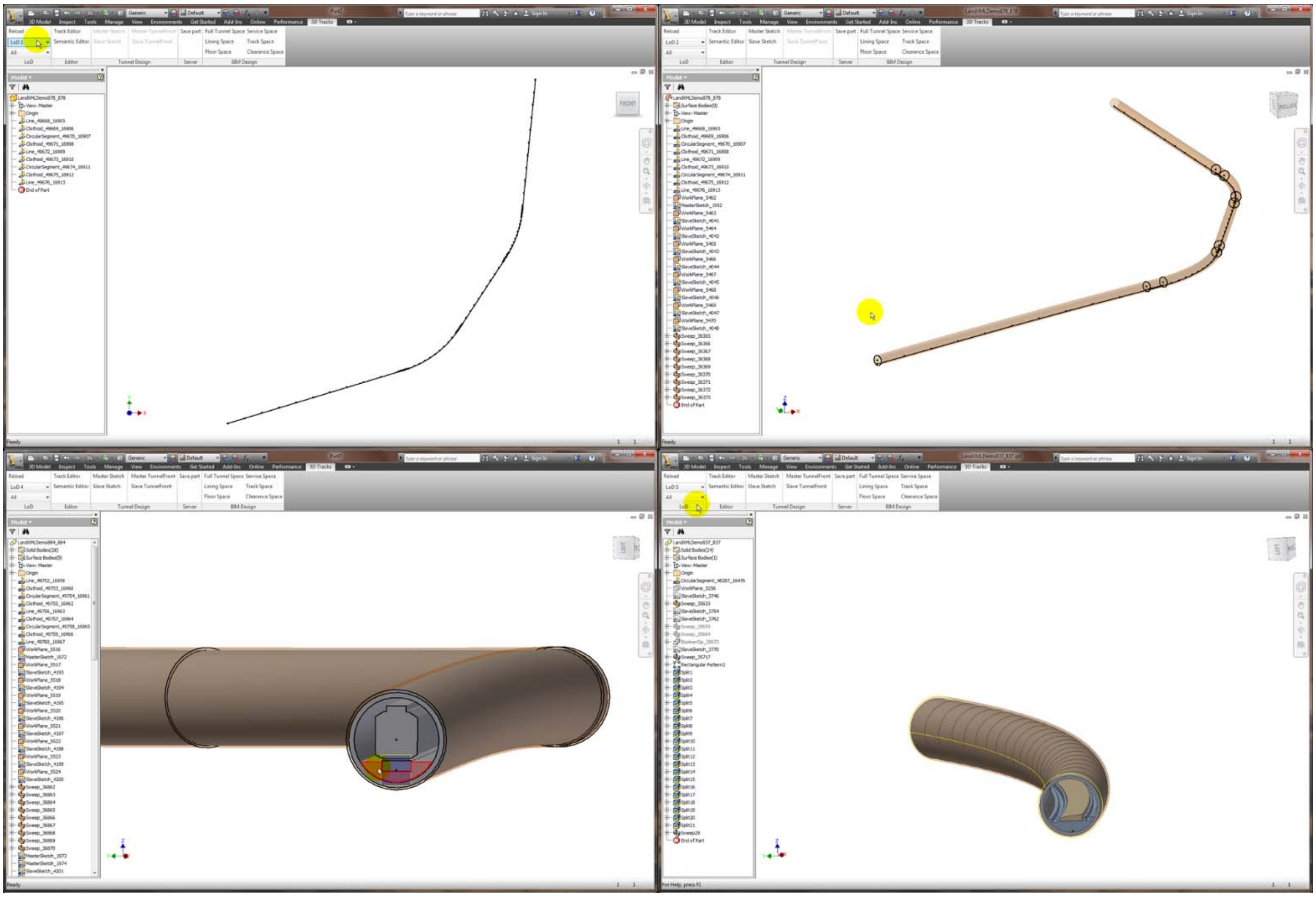


Figure 13: Four screenshots of the developed prototype system depicting the modeling process on the LoDs 1-4, respectively.

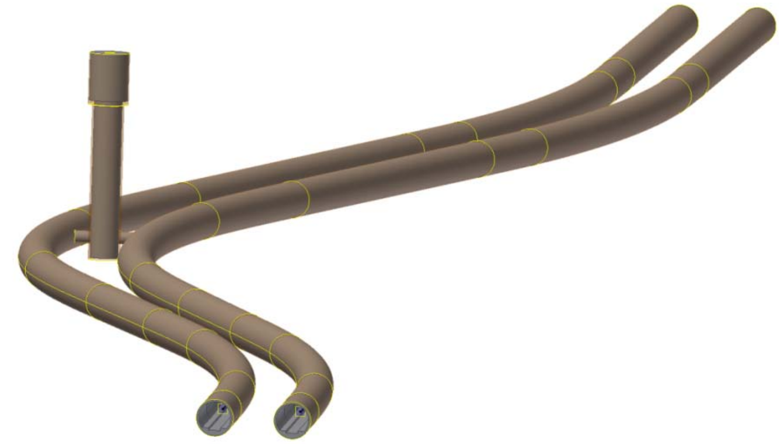


Figure 14: The LoD4 model of the shield tunnels and an escape shaft connceted to them

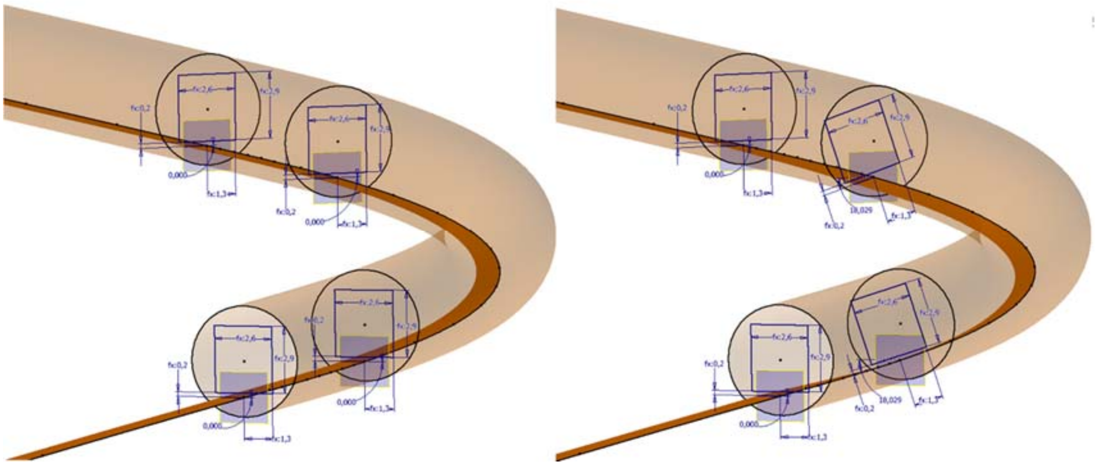


Figure 15: The super-elevation stripe and the loading railway gauge are automatically recalculated when the train velocity is adapted.