

Synchronous collaborative tunnel design based on consistency-preserving multi-scale models

André Borrmann^a, Matthias Flurl^b, Javier Ramos Jubierre^a, Ralf-Peter Mundani^b, Ernst Rank^b

^aChair of Computational Modeling and Simulation, Leonhard Obermeyer Center, Technische Universität München, Arcisstraße 21, 80290 Munich, Germany

^bChair of Computation in Engineering, Leonhard Obermeyer Center, Technische Universität München, Arcisstraße 21, 80290 Munich, Germany

Corresponding Author: André Borrmann, andre.borrmann@tum.de, Phone: +49 89 289 23047

Please cite as: Borrmann, A.; Flurl, M.; Jubierre, J.R.; Mundani, R.-P.; Rank, E.,: Synchronous collaborative tunnel design based on consistency-preserving multi-scale models, *Advanced Engineering Informatics*, 2014

DOI: [10.1016/j.aei.2014.07.005](https://doi.org/10.1016/j.aei.2014.07.005)

Abstract

The planning of large infrastructure projects such as inner-city subway tracks is a highly collaborative process in which numerous experts from different domains are involved. While performing the planning task, widely differing scales have to be taken into consideration, ranging from the kilometer scale for the general routing of the track down to the centimeter scale for the detailed design of connection points. Currently there is no technology available which supports both the collaborative as well as the multi-scale aspect in an adequate manner. To fill this technological gap and better support the collaborative design and engineering activities involved with infrastructure planning, this paper introduces a new methodology which allows engineers to simultaneously manipulate a shared multi-scale tunnel model. This methodology comprises two main aspects. The first aspect is a multi-scale model for shield tunnels, which provides five different levels of detail (LoD) representing the different levels of abstraction required throughout the planning progress. The second aspect is a conceived collaboration platform, which enables simultaneous modifications of the multi-scale model by multiple users. In existing multi-scale approaches, where the individual representations are stored independently from each other, there is a high risk of creating inconsistencies, in particular in the highly dynamic collaborative planning context. To overcome this issue, the concept presented in this paper makes use of procedural modeling techniques for creating explicit dependencies between the geometric entities on the different LoDs. This results in a highly flexible, yet inherently consistent multi-scale model where the manipulation of elements on coarser LoDs results in an automated update of all dependent elements on finer LoDs. The proposed multi-scale model forms a well-suited basis for realizing the collaboration concept, which allows several experts to simultaneously manipulate a shared infrastructure model on various scales while using the different design tools they are accustomed to. The paper discusses in detail the principles and advantages of the proposed multi-scale modeling approach as well as its application in the context of collaborative tunnel design. The paper concludes with a case study of a large infrastructure project: a new inner-city subway tunnel in Munich, Germany.

Keywords: Collaborative design, Product Modeling, Level of Detail, Multi-scale modeling, Infrastructure, Shield Tunnel

1 Introduction

The built infrastructure – especially for transportation – is of crucial importance for today's highly developed societies, since it guarantees the mobility of its population and is a prerequisite for the constant stream of goods provided to industry and private households. This particularly applies to the transport networks of large cities, comprising rail-based public transport as well as a complex road networks. Due to the continuous growth of the population in the world's conurbations, the built infrastructure facilities are constantly being developed and extended.

The design and engineering of inner-city 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 collaboration.

Collaboration is usually organized in two different modi [1]: In the *asynchronous* form, information is exchanged between the different stakeholders without immediate feedback, i.e. the collaborative work is temporally decoupled. In the *synchronous* form, however, the participants work simultaneously and provide direct responses to proposed design modifications. While from a duration point of view, asynchronous collaboration is dominating the planning process, the synchronous phases play a more significant role: Here occurring problems involving the different parties are discussed and joint decisions are taken to solve them. For this reason, we focus on synchronous collaboration in this paper and present novel computational methods supporting it in the context of tunnel design.

An important peculiarity of infrastructure design is that 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.

Today, these aspects are supported only to a very limited extent by currently available software tools for the planning of infrastructure projects. The majority of the projects still employ the conventional planning approach based on 2D technical drawings. This has a number of significant issues, including:

- The consistency between the different 2D plans (top views, cross-sections etc.) must be preserved manually. As these 2D plans must be produced at multiple scales and on different levels of detail, their mutual consistency again has to be preserved manually.
- In consequence, all plans have to be manually checked and updated, when modifications are made.

These issues mean that the planning process is both laborious and error-prone, and that the engineers involved are forced to spend a disproportionate amount of time in dealing with minor administrative tasks and consistency preservation instead of being able to focus on the core engineering tasks.

This paper presents a comprehensive methodological approach for improving computer support for the planning of infrastructure projects that has the potential to overcome these limitations. In our investigations, we have focused on the following aspects:

- holistic application of 3D modeling techniques for the geometric design of the infrastructure project,
- development of a formal method for multi-scale modeling which supports automated consistency preservation between the different scales,
- coherent coupling of semantic descriptions with multi-scale geometric models,
- techniques for supporting synchronous collaborative work on the basis of shared multi-scale models, including the development of locking mechanisms which allow engineers to work concurrently without disturbing one another and avoid to violate the consistency of the overall model.

The remainder of this paper is organized as follows: In Section 2 we introduce a new methodology for the inherently consistent multi-scale modeling of infrastructure projects which relies on the application of parametric modeling techniques for establishing dependencies between the different scales. Based on this methodology, we describe a comprehensive data model in Section 3 with which it is possible to describe the primary track model, the multi-scale geometric model as well as the associated semantics in a three-fold data structure. This data model forms the basis for an infrastructure design collaboration platform which we describe in detail in Section 4. The paper concludes with a real-world case study: the second main subway track in Munich, Germany, which is currently in planning. We have applied our methodology to this project in order to prove its general suitability.

2 Inherently consistent multi-scale modeling

2.1 Overview

This paper presents a new methodology for creating, exchanging and storing multi-scale geometric models for infrastructure projects which explicitly defines dependencies between the individual levels of detail (LoDs). These explicit dependencies support automated consistency checks and even automated consistency preservation. The methodology relies on parametric modeling technologies [2], including the use of dimensional and geometric constraints for defining flexible 2D sketches, as well as the procedural definition of complex 3D models through the sequential use of geometric operations such as extrusion, transformation and Boolean operations.

Parametric modeling techniques facilitate a step-wise development of infrastructure models that evolve from a coarse level of detail to successively finer LoDs, which precisely reflects well-established practice in infrastructure planning. Conventionally, when fundamental modifications at a coarse level are made at a late planning phase, such as the modification of the principal tunnel axis, the planners are forced to completely re-elaborate all related models and plans, e.g. the detailed tunnel geometry. By applying the methodology presented in this paper, modifications at a coarse LoD are automatically propagated to all finer LoDs, thus providing a means for an automated preservation of consistency and, at the same time, significantly reducing the effort required for re-elaboration.

2.2 Related Work

The concept of multiple geometric representations on different scales is well known from the domains of Cartography and Geographic Information Systems (GIS). For example CityGML, an open standard for the storage and exchange of 3D city models based on GML, provides 5 different levels of detail [3]. The LoD concept in these application areas relies on the independent storage of individual geometric models on each level of detail (Figure 1). As the dependency between the individual levels is not explicitly represented, inconsistency can easily arise. Nevertheless, for geographic applications the concept of independent LoD representations is well suited since GIS applications rely on rather static data sets, which are rarely subject to modifications.

Another important difference between the cartography/GIS domain and the infrastructure design domain considered here is the way multi-scale models are generated: In cartography, mostly a bottom-up approach is followed, i.e. detailed data is captured and abstracted to generate coarser representations [4,5]. In design processes, however, a top-down approach is followed starting from a coarse representation (e.g. the general course of a tunnel) and adding more and more details to create finer representations.

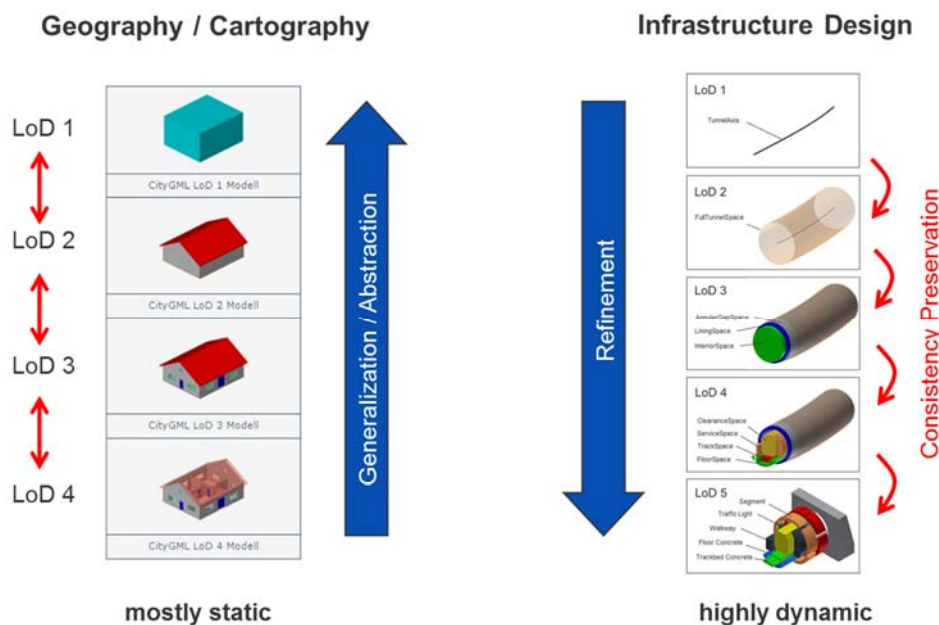


Figure 1: In the established multi-scale approaches of geography and cartography applications, the representations of the individual levels of detail are maintained independently from each other. This is well-suited for rather static application scenarios such as those commonly found in the field of Geographic Information Systems. However, in the highly dynamic context of infrastructure design the risk of inconsistencies arising is too high. A more robust approach is required that allows for an automated consistency preservation.

Taking these characteristics of planning processes into account, i.e. the strong dynamics regarding frequent model updates and the top-down design procedure, we present an approach to multi-scale modeling in infrastructure design, which provides both flexibility and robustness. To realize this, we propose the definition of explicit dependencies between the different levels of detail during the creation of the multi-scale model.

It is important to distinguish the *level of detail* concept elaborated in this paper from the *Level of Development* approach which has been introduced recently by the American Institute of Architects [6]. Though both concepts share the same acronym, there are important

differences in the underlying semantics. The Level of Development approach is used to define the content, maturity and reliability of information provided by a building information model and serves as a basis for contractual specifications [7]. By contrast, the LoD concept introduced here provides different *levels of abstraction* for a linear infrastructure facility following the well-defined multi-scale concepts of the cartography / GIS domain. Most importantly, the proposed multi-scale model contains all the different levels of detail at once, allowing the planner to dynamically switch between them and choose the appropriate level of abstraction for a particular planning task. The Level of Development concept, however, defines the required/provided information for one singular model and focuses on the evolution of this model over time, meaning that the coarser representations get lost when the development of the model progresses.

2.3 Cross-LoD consistency preservation using procedural geometry representations

The proposed methodology for the creation and management of multi-scale geometric models relies on an explicit definition of dependencies between the individual levels of detail. These explicitly available dependencies are the basis for the automated preservation of the consistency of the multi-scale model.

The definition of the dependencies is realized by applying technologies provided by parametric Computer Aided Design (CAD) systems [2,8]. 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, which are referred to as procedural models or construction history models, combine the use of dimensional and geometric constraints for defining flexible 2D sketches, with the concept of a procedural definition of complex 3D models through the successive use of geometric operations such as extrusion, rotation and Boolean operations [9–12]. Parametric modeling concepts have recently been applied to model infrastructure facilities, such as bridges and roadways [13–16].

These techniques facilitate the step-wise development of infrastructure models that evolve from a coarse level of detail to successively finer LoDs. In the proposed concept, the LoDs can be flexibly defined by the planning team according to the requirements of the infrastructure project under consideration. During the modeling process, the 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 2 illustrates the five different levels of detail defined for the design of a roadway tunnel.

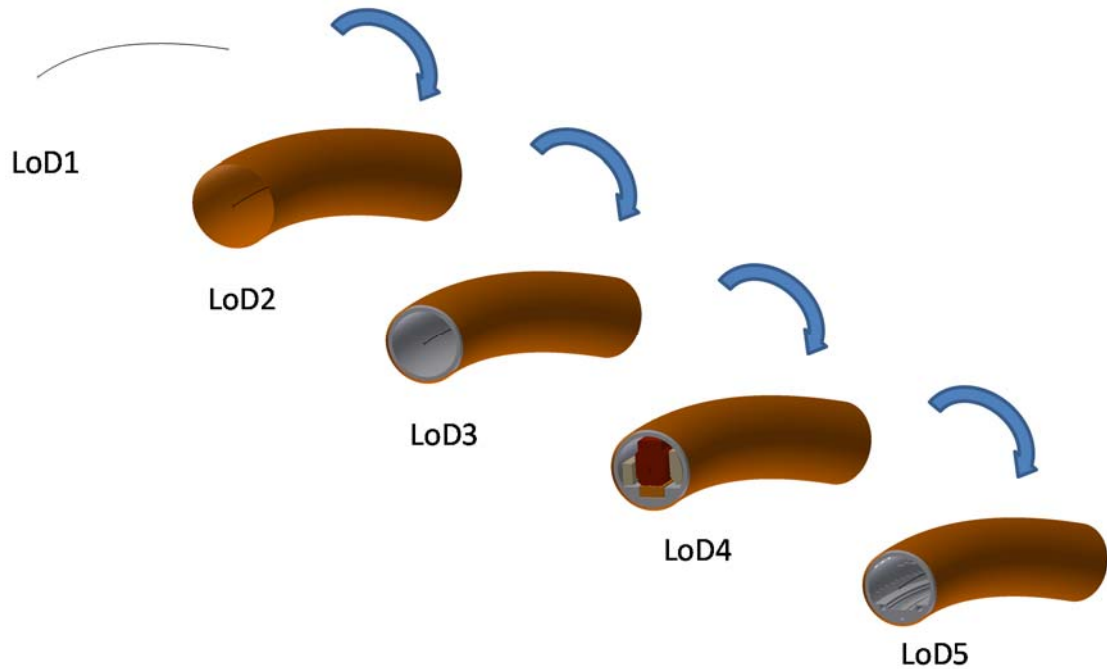


Figure 2: Illustration of the five different levels of detail defined for railway tunnel design. The concept of procedural modeling makes it possible to explicitly define dependencies between geometric elements on different levels of detail.

Applying procedural technologies for multi-scale modeling makes it possible to stringently define dependencies between individual geometric elements on different levels of detail. As a result, the levels of detail of the model are not isolated from each other, but inter-related by means of the construction history. Accordingly, the resulting multi-scale model is inherently consistent and preserves a high degree of flexibility. Modifications of elements at a coarse LoD, such as the principal axis of the tunnel are automatically propagated to all dependent objects on the finer LoDs.

However, there are limits to the degree of modifications made at coarse levels which can be propagated to finer ones. These limits 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 volume 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.

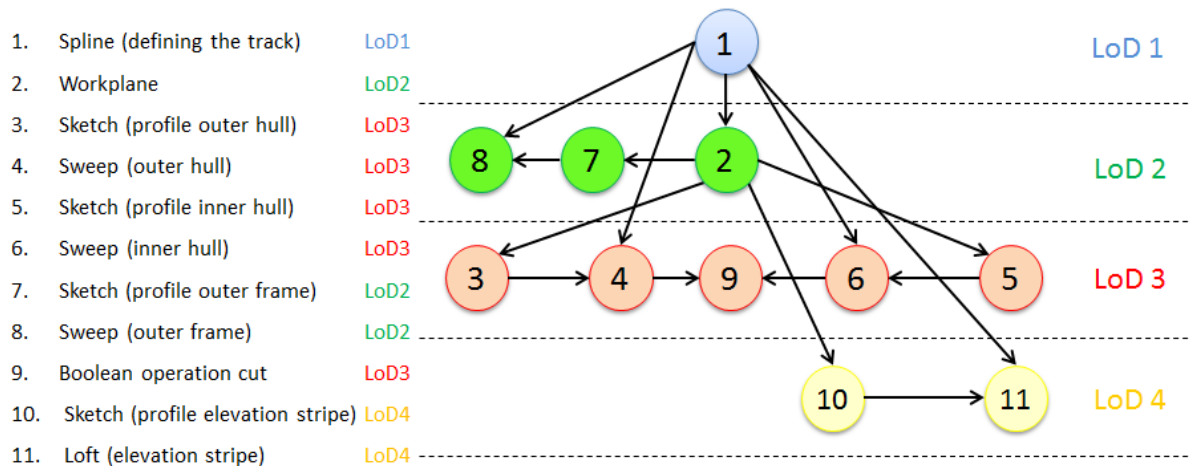


Figure 3: Left-hand side: Illustration of a construction history captured by a procedural model. The switches between the individual LoDs are explicitly triggered by the user. Right-hand side: The cross-LoD dependency graph is a directed acyclic graph (DAG) resulting from referencing entities on lower levels for construction operations on higher levels. The edges are directed towards the dependent object.

The proposed methodology for creating inherently consistent multi-scale models relies on the use of parametric modeling technologies [2] for defining dependencies between the geometric elements of the different LoDs (Figure 3). The concepts that underlie parametric modeling were developed in the 1990's and subsequently implemented in mature commercial CAD systems, including Autodesk Inventor, Dassault CATIA, Dassault SolidWorks and Siemens NX. Today, these systems are used mainly in the mechanical engineering domain, but there is increasing adoption in the Architecture Engineering and Construction (AEC) industry too [16–18].

It is important to distinguish these general-purpose parametric modelers from BIM authoring systems (such as Autodesk Revit, Graphisoft ArchiCAD, etc.). While the former allow the creation of adaptable geometric models in a very generic and flexible manner, the latter rely on the use of pre-defined object types (doors, walls, etc.) which are tailored to the specific needs of the AEC domain and provide additional non-geometric (semantic) information. Some BIM authoring tools incorporate aspects of parametric modeling, but in a much more restricted fashion than the general-purpose parametric CAD systems applied here [18].

The majority of the available parametric CAD systems implement a twofold approach, comprising the definition of 2D sketches including dimensional and geometric constraints on the one hand (Figure 5) 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 6) [19]. 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. Figure 4 depicts some of the geometric constraints available in major parametric CAD systems. 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. This is typically achieved through the integration of a

geometric constraint solver which computes a feasible solution to the given set of constraints [20]. The user is informed if the sketch is over-constrained (too many constraints) or under-constrained (insufficient constraints). If it is well-constrained, the valid solution is immediately displayed (Figure 5).

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 by means of an explicit boundary representation. The procedural approach provides the user of the system the possibility to easily modify an existing model by going back in the construction history and adapting the corresponding parameter of the construction operation.

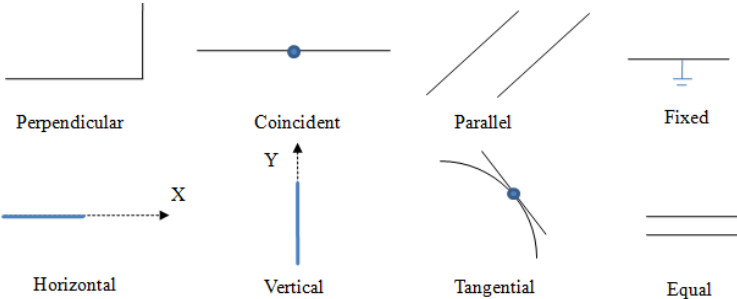


Figure 4: Geometric constraints typical provided by parametric CAD systems

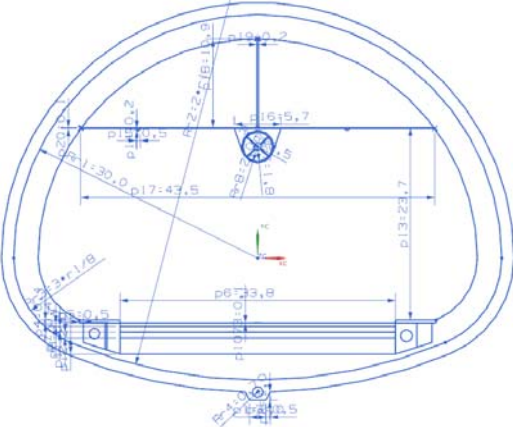


Figure 5: A complex parametric sketch defining the cross-section of a tunnel

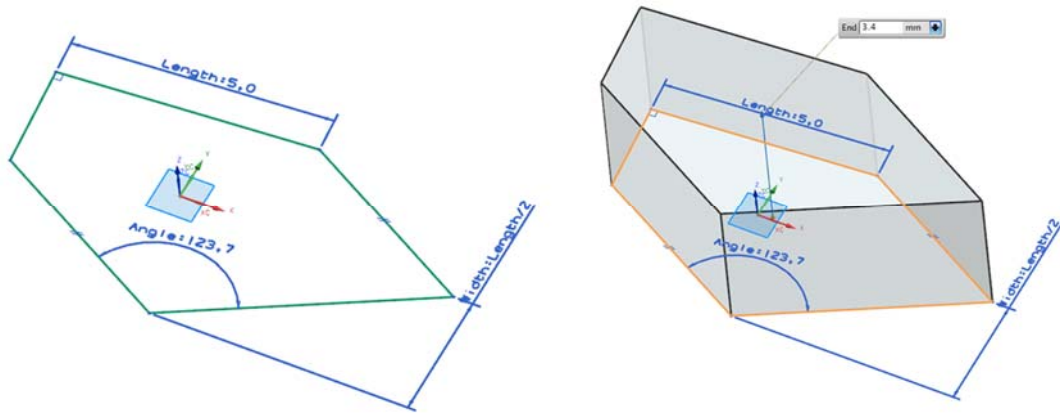


Figure 6: A sketch definition and the subsequent application of an extrusion operation

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

Whenever a construction operation operates on the outcome of a preceding operation, there is a dependency relationship between these operations. A typical example is an extrusion operation which uses a 2D sketch as basis. In this case, the extrusion operation is dependent on the sketch, which means that it cannot be performed without the existence of the sketch, or more precisely the completion of the *create sketch* operation. The entirety of all dependency relationships can be represented by a directed acyclic graph (DAG), such as that shown in Figure 3. Please note that in the context of this paper, the edges are depicted as being directed towards the dependent object. The dependency relationship graph is an important basis for handling concurrent modifications by multiple users, as discussed in more detail in Section 4.

To implement our concept of inherently consistent multi-scale models we make extensive use of parametric modeling techniques, including parametric sketches and the construction history, to define dependencies between the geometric elements across different levels of detail. The next section describes a neutral data model which is able to capture a procedural model comprising the construction operations as well as the dependencies among them and thus facilitates the exchange of multi-scale model with embedded consistency preservation rules.

3 A data model for the exchange of multi-scale infrastructure models

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 facilitating this data exchange [21–23]. A neutral data model that makes it possible to share a procedural description of multi-scale models is able to transmit the dependencies between different LoDs and means that the flexibility and the inherent consistency of the model can be maintained [24,25].

To meet the complex demands of multi-scale modeling, the data model we have developed consists of three main parts:

- a track alignment model capturing the primary alignment parameters,
- a procedural geometric model describing the complete geometric representation of the project at different scales as well as the dependencies between the different scales,
- a semantic model capturing the semantics of the geometric objects as well as the relationships among them.

All three parts are interrelated with each other. Together they form a comprehensive multi-scale description of the infrastructure facility.

3.1 Track Alignment Model

All linear infrastructure facilities are defined by an axis, known as the track or alignment of the infrastructure [26]. The infrastructure alignment is the curve that defines the trajectory between an origin and a destination, and which also adapts vertically to follow the terrain. In today's well-established practice, the alignment is designed by means of two 2D curves, the horizontal and the vertical alignment, which are superimposed to form the resulting 3D curve [27–29].

The horizontal alignment defines the course of the track in the X-Y plane and is composed of three elements, namely straight lines, arcs and spirals. Spirals are used to connect the straight elements with the arcs to create smooth transitions between elements. The vertical alignment (also known as gradient or profile) defines the corresponding Z coordinates for every point along the track. This alignment is usually defined by straight lines connected by crest and sag curves determined by parabolic or circular functions.

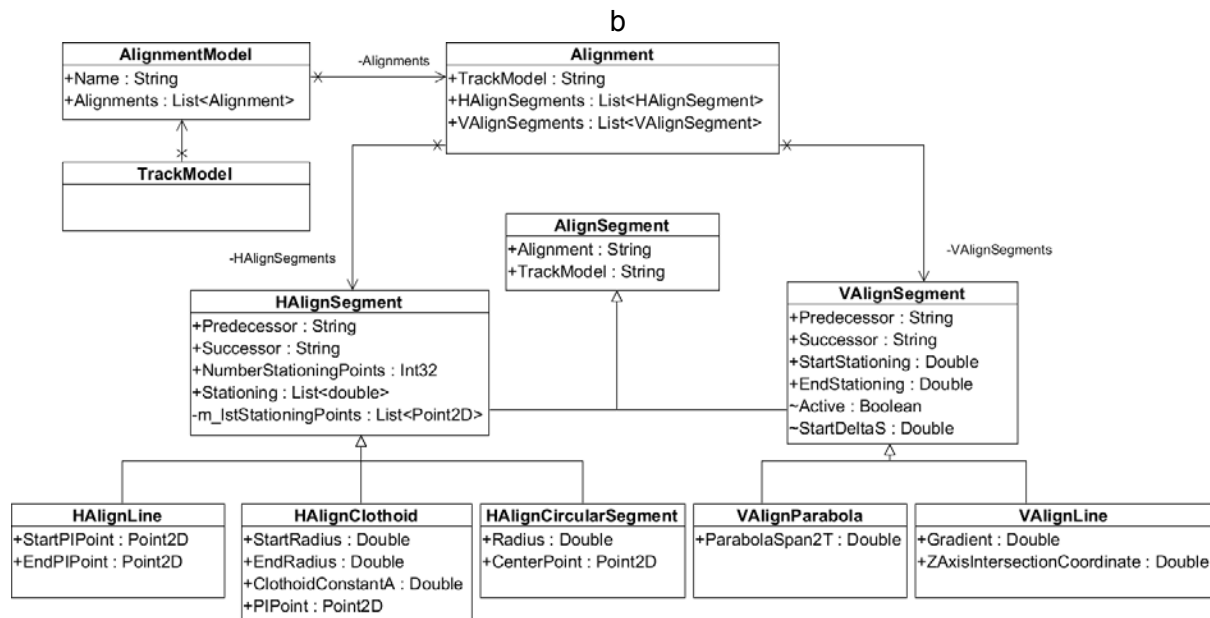


Figure 7: UML class diagram representing the track alignment model

To capture the alignment information, a dedicated alignment model has been developed (Figure 7). The track alignment model can be populated by importing alignment information created by genuine track design software tools such as Autodesk Civil3D, or Bentley InRoads, etc. via the neutral data format LandXML [30]. These track design tools are well suited for the modeling of an alignment, but do not allow the subsequent 3D modeling of the tunnel tube, the escape shafts, stations, etc. The introduced alignment model meets the

specific requirements of the collaboration platform presented in Section 4, which is the interactive modification of the alignment during the collaborative session. To this end, the data model representing the alignment was streamlined (in comparison with LandXML), providing only the main track parameters that are required to perform the intended modifications. Any derived data was excluded from the model, as it is dynamically computed by the client applications whenever changes occur. Any changes in the alignment can be performed directly on the basis of the presented track alignment model.

The track alignment model is connected with the procedural operations (Section 3.2) which define the track curve as explicit 3D geometry. On the one hand, this approach allows the direct manipulation of the primary alignment parameters, e.g. radius, clothoid's constant, etc. On the other hand, the automated propagation of these changes into the procedural model becomes possible.

In addition, the genuine alignment information provided by the track alignment model permits the calculation of dependent geometric information that would be used by the procedural model at different levels of detail. A straightforward example is the calculation of the 3D curves of the track on LoD 1. A second example can be found in the cant or super-elevation of the track, a horizontal inclination of the track defined primarily by the speed and the width between rails.

The presented alignment model is crucial for a better support of the collaborative planning process as discussed in Section 4, in particular for improving the collaboration between alignment experts and tunneling engineers.

3.2 Procedural Geometry Model

As described in Section 2.3, the proposed methodology for consistency preservation of multi-scale models relies on an explicit definition of dependencies between the individual levels of detail. To create these dependencies we make use of parametric CAD systems, which provide parametric sketches as well as a construction history-based approach for defining flexible geometric models. Hence, to allow the exchange of inherently consistent multi-scale models, a neutral data model had to be developed which is able to capture procedural models. The model was conceived in a way that allows its application in a synchronous modeling process via a collaboration platform.

The data model we developed for capturing procedural models consists of two main parts. The first part (Figure 8) provides the possibility to describe fundamental geometric operations such as splines, work planes, sweeps, Boolean operations, as well as parametric sketches comprising both geometric elements and the dimensional and geometrical constraints applied [16]. The individual operations are stored in an ordered list, reflecting the sequence in which they are applied to generate the desired geometry. The preservation of the original order ensures the consistency of the model and facilitates a simple step-by-step reconstruction.

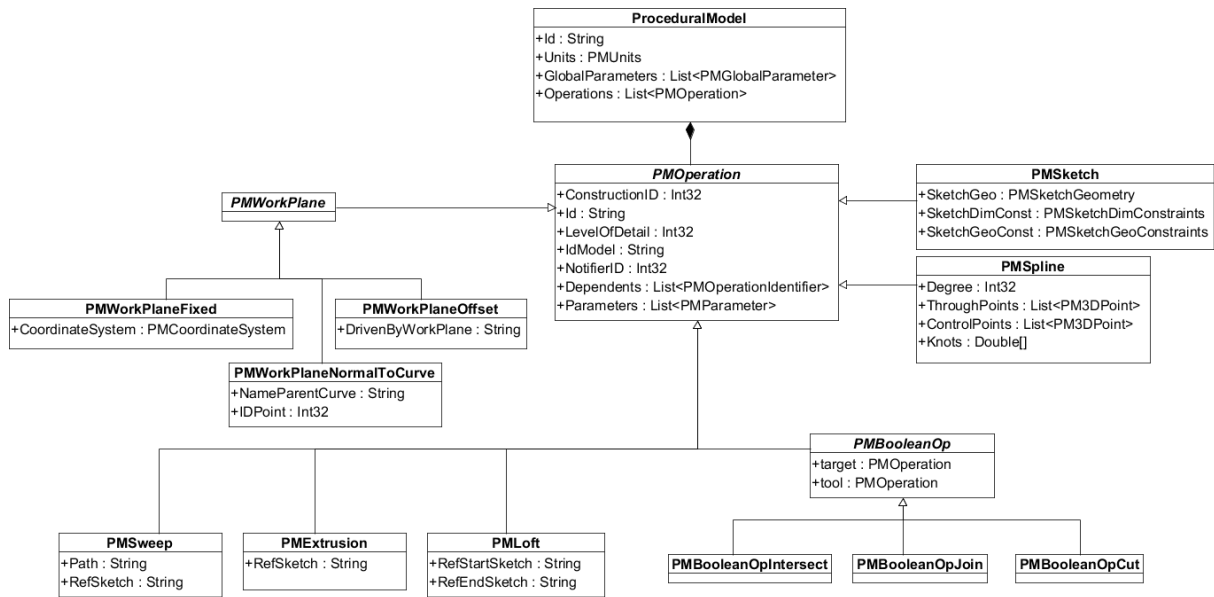


Figure 8: Section of the developed data model for capturing a procedural model (UML diagram)

In contrast to the rather generic approach taken by Part 55 of the Standard for the Exchange of Product model data (STEP) [31–34], the available construction operations are explicitly represented by the data model. By means of the defined operations, the most important parts of a procedural model can be captured. However, not all construction operations provided by modern parametric CAD systems are included. In the current state of the implementation, we excluded more specific construction operations such as chamfering or filleting, as these operations are of minor importance in infrastructure design.

The second part of the data model (Figure 9) makes it possible to explicitly represent the dependencies between different operations. These dependencies produce a directed acyclic dependency graph which is used for preserving the consistency of the procedural geometry model. These dependencies are used to perform an automatic update of dependent operations in case of changes. In addition, these dependencies provide a basis for locking mechanisms that are necessary for supporting synchronous manipulation of this model, i.e. different users operating simultaneously on the procedural model (see Section 4).

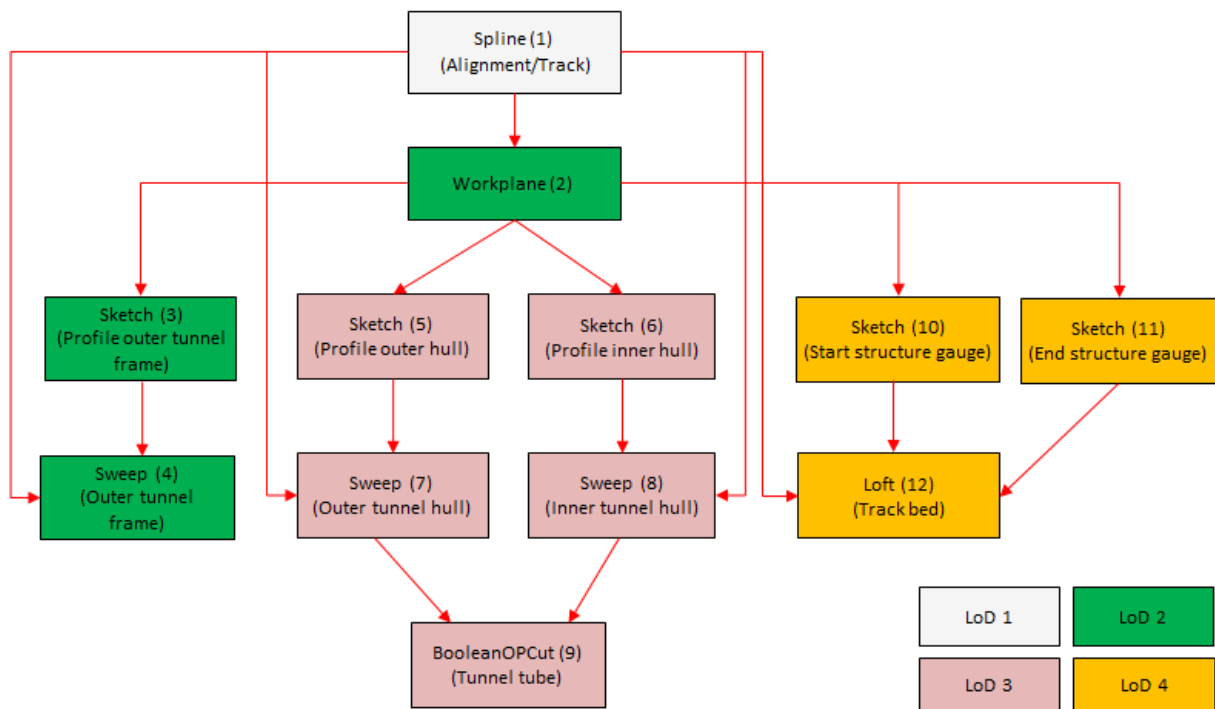


Figure 9: Dependencies between different operations produce an acyclic dependency feature graph, where the operations are the nodes and the dependencies the edges. Different LoDs are depicted in different colors.

To realize the multi-scale concept, each operation is assigned an appropriate level of detail during the creation process of the procedural model. By defining dependencies across different LoDs, it becomes feasible for the system to automatically issue all necessary updates and thus achieve the desired automated cross-LoD consistency preservation.

3.3 Master-replication concept

As mentioned above, the proposed procedural model covers only a subset of the commonly used modeling commands. This confinement is determined by the modeling context, the planning of inner-city subway tracks. At the same time, the planning context considered here implies adding high-level modeling operations into the procedural model, which encapsulate a number of low level operations and specifically fulfill the requirements of tunnel design.

One of these high-level operations is the master-replication concept, which meets the demand of repetitive application of design patterns. For example, the concept is applied for modeling the geometry of the tunnel cross-sections.

A base sketch, the so-called master-sketch defines the geometry, constraints and parameters of the cross-section. The replication sketches copies this information, but is able to assign differing values to the parameters. Usually, the parameters of the replicated sketch are defined through an arithmetic expression, which links them to the corresponding value of the master-sketch. In consequence, replication sketches provide a topological identical but morphological deviating copy of the master sketch.

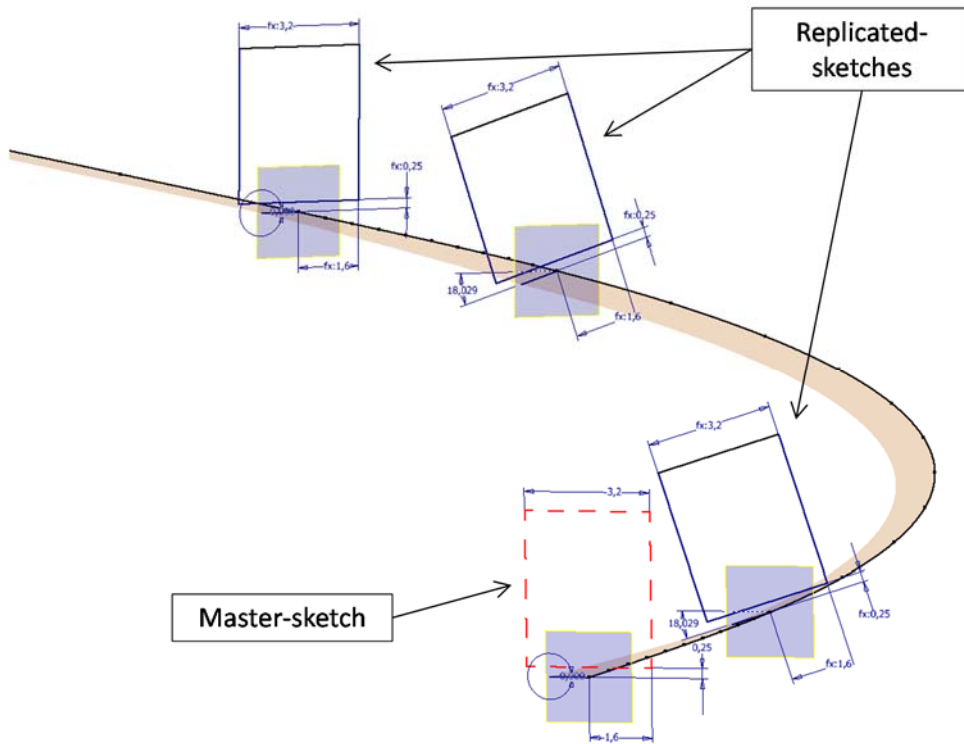


Figure 10: A master-sketch (in dashed red color) and its replicated sketches (in continuous blue color) defines the railway clearance gauge. All the replicated parameters are related to the master-sketch except the one which defines the slope of the super-elevation stripe.

The master-replication concept for sketches is an ideal basis to reduce repetitive tasks to improve the modeling process: It allows e.g. defining the geometry of the tunnel cross-section, since the principle cross-section geometry does not change along the track, yet it only changes its inclination according to the cant. When applying this concept, the designing engineer has to define the cross-section only once in the master sketch, while replicated sketches adapt it to a certain cant value, caused e.g. by different curvature of the alignment segments (Figure 10). The data model used to represent the relationships between master and replication sketches is depicted in Figure 11.

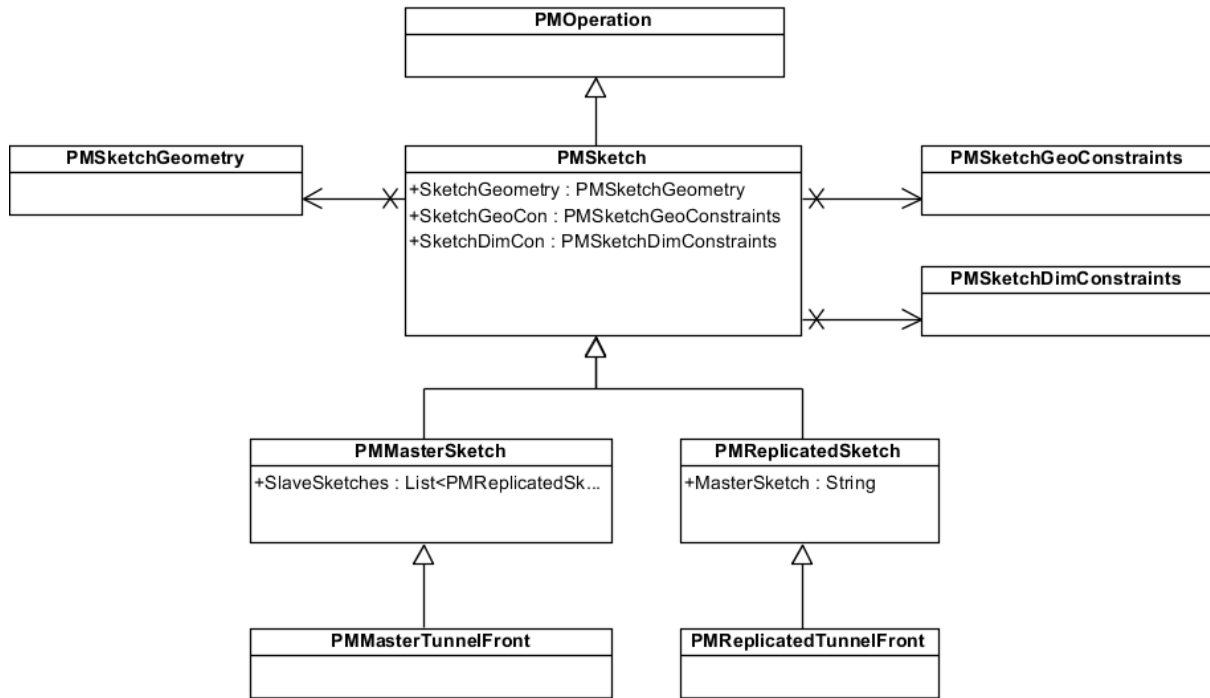


Figure 11: The master/replication sketch concept forms part of the proposed procedural model

3.4 A multi-scale product model combining geometry and semantics

For a comprehensive use of the proposed model throughout the entire design and engineering process, it is necessary to incorporate semantics. Models that comprise both a semantic and geometric description are usually referred to as product models [23]. Examples are the Industry Foundation Classes (IFC) for building design [35] and the CIS/2 model for structural steel projects [36]. These models are based on object-oriented principles and provide typing, inheritance, attributes and relationships, resulting in powerful mechanisms for describing semantics. Consequently, product models form a sound foundation for ensuring interoperability between different software products and between different stages of the construction project.

Based on preliminary work by [37,38] we are introducing a comprehensive product model for shield tunnels which fulfills the demands of the design and engineering of large infrastructure projects. The main emphasis is placed on integrating the semantic description with the multi-scale geometry approach discussed above. In the presented concept, the multi-scale approach also forms part of the semantic model, i.e. specific entities are only available at a particular LoD. The major challenge is then to achieve and maintain semantic-geometric coherence in the model [39,40], which means that geometric elements at a certain LoD are assigned to correct semantic elements on the same level.

The development of the multi-scale product model is described in three steps: Departing from a single-scale product model for shield tunnels, we first introduce multiple levels of detail into the model but use isolated geometry representation for each LoD, before we finally integrate a procedural geometry description to realize the desired cross-LoD consistency preservation.

3.4.1 Point of departure: A single-scale product model for shield tunnels

The point of departure is the development of a “single-scale” product model for shield tunnels. Although a first draft for a shield tunnel product model was provided in (Yabuki et al. 2007), it had to be adapted to the specific needs of our research, in particular with respect to the multi-scale modeling approach. Figure 12 shows an overview of the resulting model. The semantic model presented is aligned with the Industry Foundation Classes (IFC), a comprehensive, standardized product model for buildings. In particular, we make extensive use of the space structure concept. As explained in detail below, it significantly simplifies the integration of multi-scale concepts into the model. Like the IFC model, the proposed tunnel product model provides a clear separation between semantic objects and the associated geometry.

The left-hand side of Figure 12 shows the semantic part of the model. Please note that, like in the IFC, we distinguish space objects (depicted in blue) from physical objects (depicted in green). The meaning of the individual entities is illustrated in Figure 13. Except for the ring space, all space objects represent longitudinal spaces along the entire *TunnelPart*. The *Ring* space, however, has the length of a 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.

The associated geometry representations are depicted on the right-hand side of Figure 12. The tunnel object is associated with a dedicated Alignment object. 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. This is realized by means of the track alignment model presented in Section 3.1.

A *TunnelPart* represents a stretch of the tunnel with unvaried characteristics. The denomination *TunnelPart* was chosen in favor of *TunnelSection* to avoid confusion with cross-sections. The start and end of a *TunnelPart* are defined through specifying the corresponding abscissas (chainage values) of the underlying alignment curve. The following objects are continuous along the entire *TunnelPart*: all space entities, except for the *Ring* element, and the physical entities *Cable Duct*, *Drainage*, *Trackbed Concrete*, *Trackbed Rails*, *Walkway*. The *Traffic Light* and *Ring Segment* entities represent discrete (non-continuous) objects.

For describing the geometry of the continuous objects (both the space objects and the physical objects), we rely on the swept area geometry representation (*IfcFixedReferenceSweptAreaSolid*) provided by the available IFC standard to define volumetric geometry by means of cross-sections extruded along a given axis. For describing the geometry of the discrete objects, straightforward boundary representations are applied. The position of non-continuous elements along the axis is defined by reference to the corresponding abscissa (chainage value) of the underlying alignment curve.

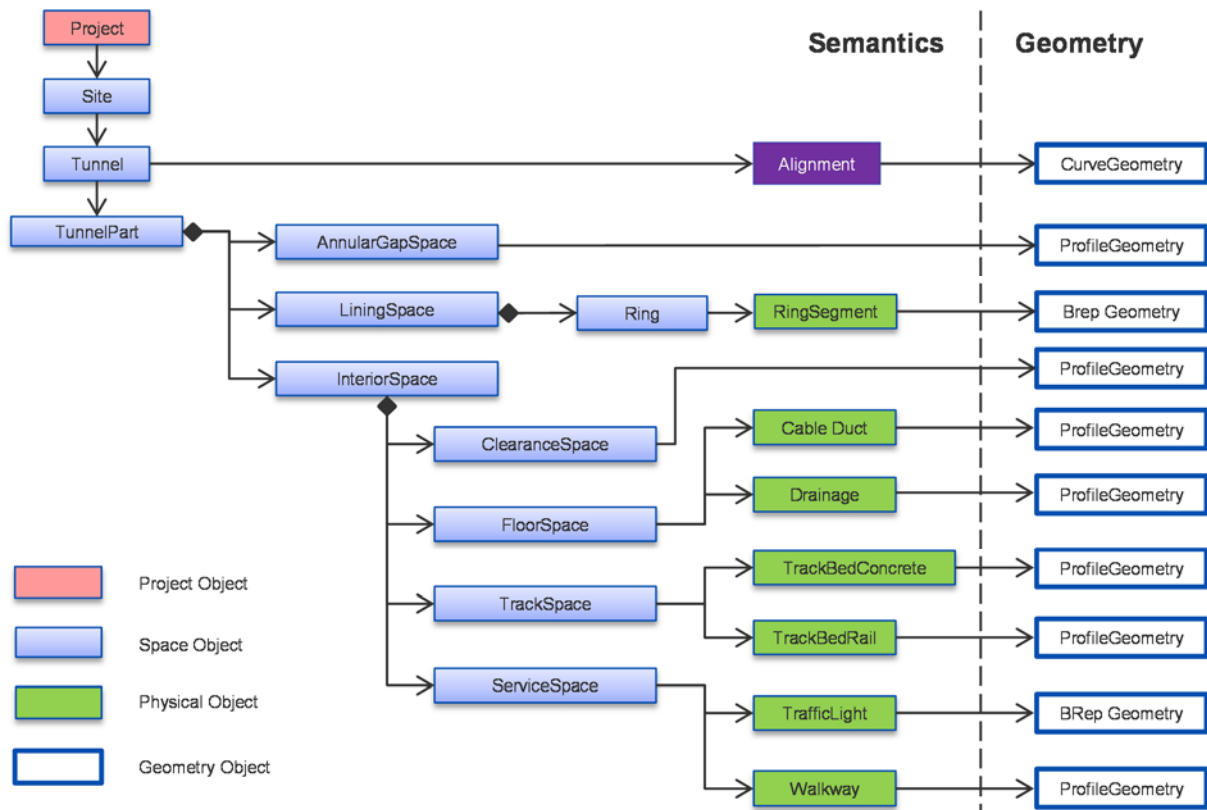


Figure 12: A single-scale shield-tunnel product model (UML object diagram)

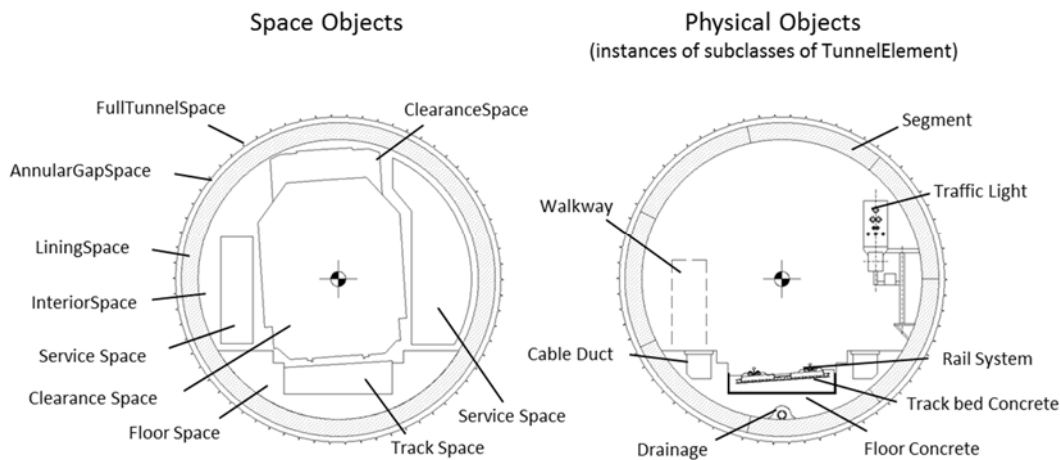


Figure 13: A tunnel cross-section depicting the individual spaces (left) and elements (right) of the product model

3.4.2 A multi-scale product model using isolated geometry descriptions

Based on the single scale product model introduced above, we have developed concepts for integrating multi-scale approaches. The main difference between our approach and the one followed by GIS standards, such as CityGML, is the scale-aware sub-division of the semantic part of the model. While the GIS standards allow the association of multiple geometric representations for the individual levels with one semantic object, but keep the semantic object structure fixed across the different LoDs, we propose to explicitly represent refinement relationships in the semantic part of the model, thus providing a much higher degree of semantic-geometrical coherence of the multi-scale model.

The resulting multi-scale product model is depicted in Figure 14. In order to group and provide access to all elements at a certain level of detail, we introduce dedicated LoD objects. 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. 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 providing full compliance with standard product modeling approaches for space-element aggregation structures.

The geometry representation is basically identical to that of the model defined above. Figure 13 provides a 2D graphical illustration of the representations at the different LoDs, while Figure 15 provides a 3D illustration. Please note that at LoD1 the tunnel is represented by its axis only. On LoD2 the additional space object *FullTunnelSpace* has been introduced to provide a semantic object representing the entirety of the tunnel. The Ring space objects belong to the finest level of detail, LoD5, since their definition happens at a more advanced stage of the planning process. Each *Ring* object contains the *RingSegments* which belong to it.

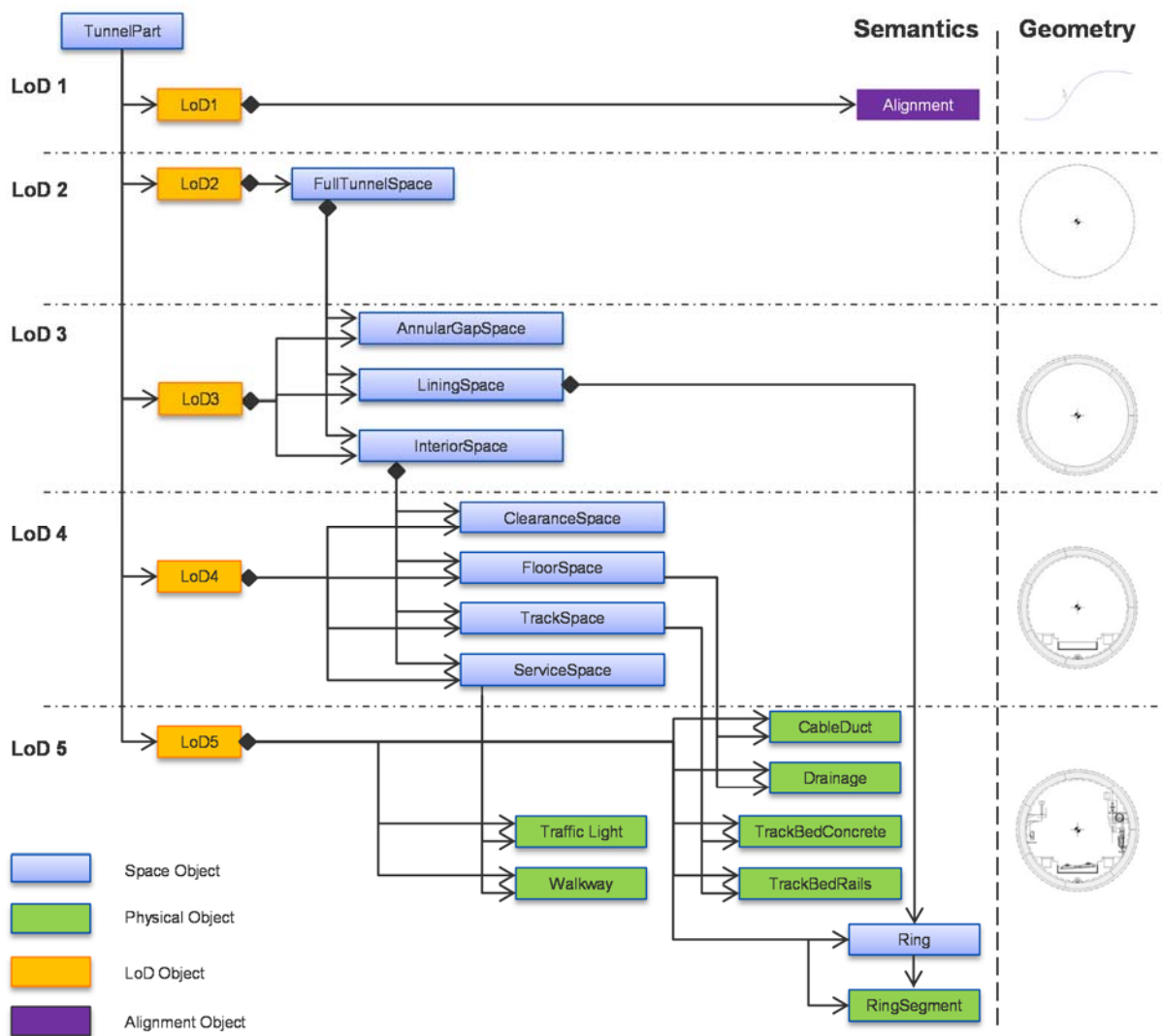


Figure 14: The proposed shield tunnel product model incorporating a coherent multi-scale representation of semantics and geometry (UML object diagram)

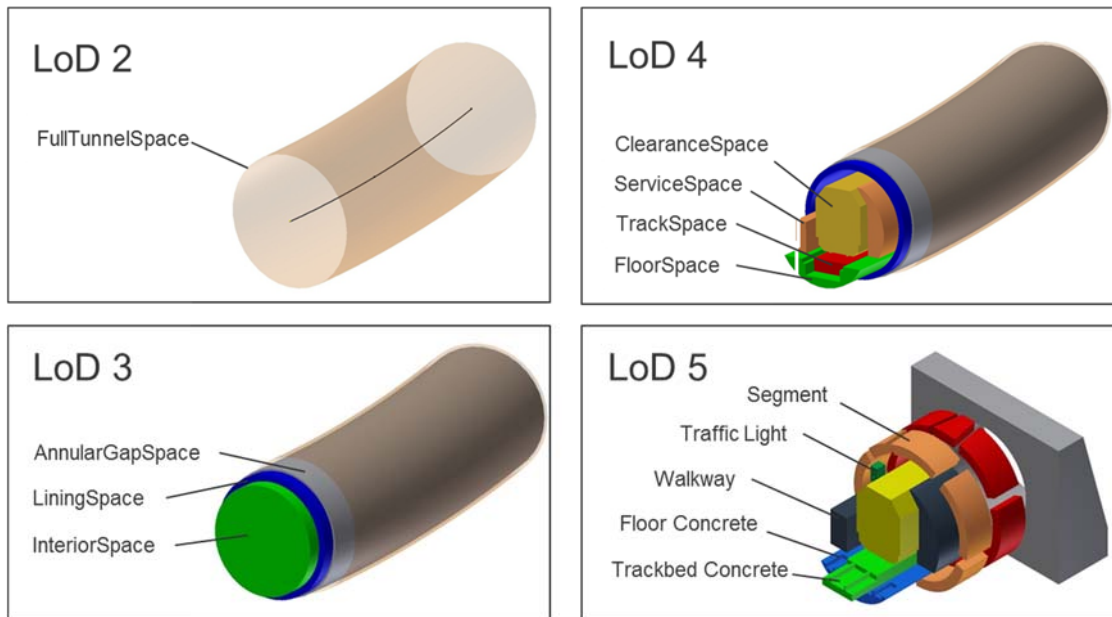


Figure 15: A 3D representation of the different LoDs of the multi-scale shield-tunnel product model

3.4.3 A multi-scale product model using procedural geometry descriptions

The multi-scale model introduced above provides a coherent representation of semantics and associated geometry. However, it does not yet provide a means of preserving the consistency between the different LoDs. This is caused by the fact that the geometry representations of the individual LoDs are independent of each other. Inconsistencies can arise, for example, when a modification is performed on one level, but not propagated to the other levels. To overcome this deficiency we propose making use of a procedural geometry description as described in Section 3.2, which allows us to explicitly define dependencies between individual geometric objects and thus provides a means for automatic consistency preservation.

Figure 16 depicts how the procedural geometry representation is integrated with the multi-scale semantic model. The explicit geometry representation of individual elements of the model is replaced by a procedural geometry description, as introduced in Section 3.2, linking the geometry of higher-level entities to that of lower-level ones.

The geometry of the longitudinal elements on LoD2-5 (FullTunnelSpace, AnnularGapSpace, Floor Concrete etc.) is described by means of sweep operations, which refer to the 3D curve representing the alignment LoD1 and use it as sweeping path. Accordingly, their geometry representation is bound to that of the alignment and automatically updated in case of changes.

For describing the geometry of the discrete objects, conventional boundary representations are applied. The position of discrete (non-continuous) elements is defined by reference to the underlying alignment curve. The longitudinal position is specified through defining the corresponding abscissa (chainage) or a distance (along the curve) to a given reference point along. In the transversal (cross-sectional) view, the position is specified using the parametric

techniques including dimensional and geometric constraints. Doing so, the discrete elements stay in their position relative to the tunnel axis when modifications are made.

Thanks to the integration of the procedural geometry description, the cross-LoD consistency preservation mechanisms introduced in Section 3.2 are embedded in the neutral product model. This enables that during the data exchange, all consistency rules are preserved and the flexibility of the model is maintained. The resulting model combines semantic information with the procedural geometry and allows the exchange of geometric-semantic multi-scale models of shield tunnels. In contrast to STEP Part 55 “Procedural and hybrid representation” [33], which takes a generic approach by allowing any STEP entity to act as an operation in a procedural geometry description, the construction operations are explicitly defined by the proposed data model. In addition, we achieve a strong coherence between the semantic and geometric information by introducing the LoD concepts in both parts and providing corresponding linking mechanisms.

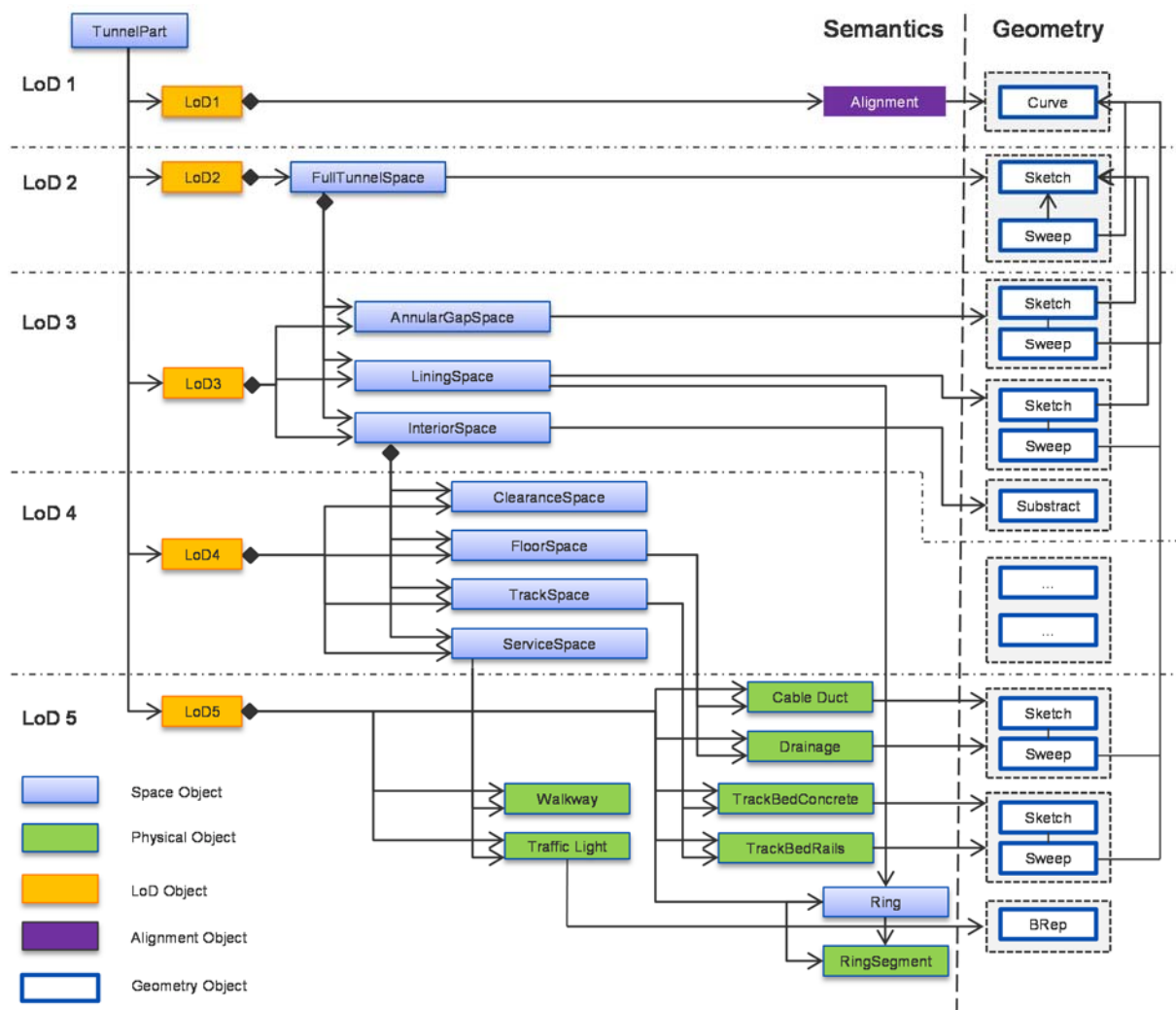


Figure 16: The shield-tunnel product model incorporating a multi-scale semantic representation and a procedural geometry representation. The model makes it possible to define dependencies between the geometric representations at the different LoDs, thus providing a means for preserving consistency across the levels.

3.5 Implementation of the data model

The overall data model for the exchange of multi-scale tunnel models combines the track alignment model, the procedural description of the geometry and the semantic description of the individual parts of a shield tunnel.

The data model has been implemented as an XML schema, where dedicated sections correspond with the individual parts described in the sections above. The corresponding instance files allow the exchange of multi-scale shield tunnel models using a procedural geometry representation. Thus multi-scale models including the cross-LoD consistency rules can be transmitted from one design system to another.

The suitability of the data model was proved by developing corresponding import and export modules for the parametric CAD systems Autodesk Inventor and Siemens NX, followed by a successful transfer of multi-scale tunnel models between these two systems. In addition, the developed product model was successfully employed in simulating the tunneling process [41].

4 Synchronous collaborative infrastructure design

In the previous sections, we presented novel concepts for the tunnel design based on multi-scale infrastructure models that address – amongst others – aspects of consistency. These concepts form the basis for realizing synchronous collaboration in multi-scale design processes, where several experts are able to work on *one shared model on different levels of detail at the same time*. Obviously, further considerations concerning consistency among the local copies of the shared model become inevitable.

4.1 Overview and related work

Collaboration plays a decisive role in infrastructure planning as in any other complex engineering process. Two types can be distinguished, asynchronous and synchronous collaboration [42]. Both types are necessarily included in most planning processes, very often they alternate and thus have to be coordinated.

For both of them, a large number of concepts and supporting software environments have been developed during the last decades. Synchronous collaboration takes place when a team works together in project meetings, i.e. at the same time, although not necessarily at the same location. It is the preferred type of collaboration whenever planning progress can be achieved by direct discussion, in fast, yet often small steps, without long-term individual design or planning phases, and whenever decision-making in a joint, multi-view and often multi-disciplinary team session is the central goal. Generic computer-supported collaborative work (CSCW) tools include application sharing, instant messaging, whiteboards etc.

To maintain the consistency of the shared model in synchronous collaboration, concurrency control strategies have to be applied. In this regard, optimistic concurrency control has to be distinguished from pessimistic concurrency control [43–45]. In the first case, different users are allowed to perform modifications affecting identical model entities (local copies) at the same time. This may result in modification conflicts, which have to be (preferable automatically) resolved subsequent to the modification phase through a possibly complex merging procedure – nevertheless, in many cases manually user interaction becomes necessary. In pessimistic concurrency control, contradictory modeling steps are prevented a

priori by means of locks, i.e. concurrent accesses to the same resource have to be mutual exclusive (i.e. entailing a sequential processing) and are typically guarded by locks realized with mutexes, semaphores, or monitors [46]. The granularity of these locks determines the degree of concurrency [47]. If the locks affect large parts of the shared model, this results in long phases where all other users are actively blocked (busy waiting) by the one who is performing the modification [46]. For optimistic as well as for pessimistic concurrency control suitable *transaction* mechanisms have to be applied. A transaction is defined as a set of steps taking a model from one consistent state to another. A broad overview about transactions in general, long transactions, and consistency concerning transactions particularly in the context of engineering design processes is given by Haerder & Reuter [48], Kutay & Eastman [49], Barghouti & Kaiser [50], and Eastman [51]. Sheth & Rusinkiewicz [52] and Rusinkiewicz & Sheth [53] discuss the application of transactional concepts to so-called transactional-workflows specifying these kinds of workflows and investigating issues involved during their execution.

Asynchronous collaboration is necessary in all cases where the work of the collaborators can or must be performed at different times. Phases of asynchronous collaboration become necessary when model information – for instance from third parties – is not accessible at all times (due to local offline modeling phases) and should be merged with the central model at later stages. Asynchronous work takes place in individual, temporarily independent planning threads, before a planning stage is reached, which can be discussed and synchronized with the project partners. Tools for asynchronous collaboration range from simple emails over bulletin boards to groupware and versioning systems. Surveys on generic supportive systems for synchronous and asynchronous collaboration can be found e.g. in [54,55].

Planning processes in civil engineering involve many specific tasks, which require computer supported collaboration with capabilities far beyond those of generic CSCW tools mentioned above. Asynchronous collaboration in the civil engineering context is discussed for instance in [10,56,57].

Only a limited number of researchers have investigated geometry-related synchronous collaboration in engineering until now. [58] provided a consistency model for real-time cooperative editing systems, thereby providing a convergence scheme for causality and intention preservation. [59] presented the Alibre Design 2D/3D CAD software. In a client-server structure called webSPIFF a central server hosts the geometric model, while engineers using a webSPIFF thin-client can work on this model synchronously. Though this approach facilitates synchronous work, it does not support the usage of common CAD software systems.

[60] researched and classified conflicting situations in the field of collaborative modeling and suggested strategies to solve these conflicts. [61] presented the CoCoS platform, which allows synchronous model-based collaboration incorporating computational simulations. However, the platform supports only very simple manipulations of the geometric model. [62] developed an approach for more advanced synchronous modeling using neutral modeling commands. Vendor specific geometric modeling commands are translated into neutral commands, sent to a central collaboration server that forwards them to the other clients. These clients then translate the neutral commands back into their own vendor specific commands. The principle idea of this approach is similar to the one presented here. However, it lacks the support of a flexible synchronous collaboration, since only one planer –

using a floor control mechanism – is allowed to modify the central model at a time, and also lacks the support of different levels of detail which are crucial in our approach.

[63] investigate a flexible concurrency structure in the field of synchronous geometric modeling. [64] present a similar dependency-based automatic locking scheme to facilitate real-time collaborative programming work using a peer-to-peer architecture. Thereby, a collaborative feature dependency graph allows locking strategies to ensure model consistency. Although the basic ideas in these locking schemes are very natural, they are not applicable to our scenario of level of detail modeling.

In contrast to all discussed collaboration concepts, we will concentrate on synchronous *multi-scale* modeling processes and take advantage of the procedural model presented in Section 3. We exploit the hierarchy of this model in a locking concept, which can enable different users to work synchronously on different levels of detail according to the directed acyclic dependency graph introduced in Section 2.3 – without blocking them and, thus, hindering collaboration.

A typical scenario in our application field is a (virtual) team meeting discussing as consequence of a modification of the horizontal or vertical track alignment (LOD 1) leading to possible interference with existing underground infrastructure (LOD 4). Even if all involved planners work during such a meeting on the same model, they may want to use – without blocking the ongoing work of their partners – different CAD tools, will need different logical views, different details, and possibly different background information in order to achieve a joint decision for this planning step.

4.2 System architecture

The general structure of the proposed collaboration concept [65] follows a classical client-server architecture. A central collaboration server hosts the shared procedural model and provides a distinguished point of access for the different clients to this model, which contains all construction steps together with their inter-dependencies. To provide the system neutrality of this model, the construction steps are stored as system independent procedural model operations (PMO), which abstract system specific operations (SSO) such as an extrusion or a Boolean operation of proprietary CAD software tools. If a client joins a collaborative session, it receives (a copy of) the shared procedural model comprising all PMOs from the server. These PMOs are processed in the original construction order and translated into a sequence of SSOs, which then are executed one after another by the respective client CAD tool. As soon as this translation and execution process is finished the user can actively participate in the collaborative modeling process. As depicted in Figure 17 to this end, every SSO done by a user (1) is immediately translated into a PMO and sent to the central server (2). The server incorporates this PMO into the shared model and instantaneously forwards it to all other participating clients (3), which translate it into a proper SSO (4) and execute it.

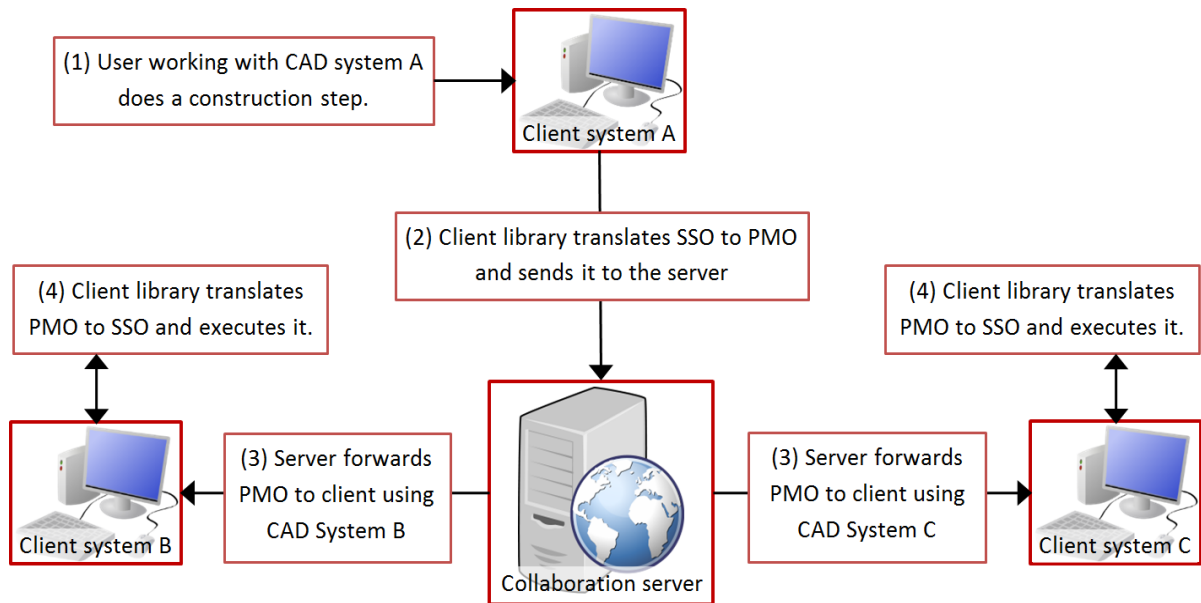


Figure 17: Synchronous collaborative modeling using the system independent procedural model via the collaboration platform

For implementation purposes, a translation process of a SSO into a PMO and vice versa has to be performed by using system specific libraries, integrated via the Application Programming Interfaces (APIs) provided by the different CAD systems. As proof of concept, prototypical libraries have been developed for the CAD tools Autodesk Inventor, Siemens NX, and Creo Elements (Wildfire Pro/ENGINEER).

4.3 Synchronous modeling based on efficient locking strategies

In a synchronous collaborative geometric modeling process possibly contradictory activities of different users, (e.g. two users modifying the same geometric entity at a time) endanger the models consistency or at least lead to results confusing the different participants [63,64,66,67]. Strategies to *cure* or *prevent* these situations are subject to the research field of concurrency control mechanisms [43]. In the most optimistic concurrency control approach, conflicting situations are generally allowed relying on the user's and/or system's abilities to correct possible inconsistencies, for example by using (often complex) merge or versioning mechanisms in order to recover the consistency between different inconsistent local models [10,56,68]. In the most pessimistic approach only one user is allowed to perform a geometric modeling step at a time. This approach obviously excludes inconsistent local model states due to contradictory modeling steps [59,62]. For clarity it should be noted, that even a consistent model might be invalid due to insufficient engineering or modeling knowledge of a user.

From a mathematical point of view, the process of procedural modeling can be described using bipartite graphs, in which the set of vertices comprises the construction operators and their operands, namely the geometric objects, while the edges represent the dependencies between those two constituents. From this bipartite graph the above presented acyclic dependency graph – exemplarily depicted in Figure 9 – can be derived (for a detailed explanation of this derivation process see [69]).

In this section, a concurrency control approach is presented that allows several users to actively contribute to a collaborative modeling session simultaneously by means of multi-

lateral synchronization, i.e. using a locking strategy based on the acyclic dependency graph. In particular, it provides different users the possibility to modify the central shared model at a time, while the system ensures the consistency of this model and the different local copies. To explain this approach in detail as well as the incorporation of the concept of different levels of detail into the collaborative modeling process in Section 4.5, some notations are introduced first.

As explained above, the described multi-scale procedural model PM can be seen as a partially ordered set of the different procedural modeling operations o_i it comprises

$$PM = \{o_1, o_2, \dots, o_n\}.$$

For the execution of a certain PMO o_j possibly several other operations $o_{j_1}, o_{j_2}, \dots, o_{j_m}$ as well as the results of their execution might be explicitly referenced as input parameters. In this case, we say that the operation o_j depends on these operations $o_{j_1}, o_{j_2}, \dots, o_{j_m}$. Subsequently, let the dependency of one operation o_j on another operation o_i be denoted by $o_i \rightarrow o_j$. Clearly, root operations do not depend on other operations.

For example, a Boolean cut operation o_C might use two extrusion operations o_{E1} and o_{E2} as input parameters resulting in the two dependencies $o_{E1} \rightarrow o_C$ and $o_{E2} \rightarrow o_C$, resp., while the two extrusion operations themselves depend on two sketch operations o_{SK1} and o_{SK2} , i.e. $o_{SK1} \rightarrow o_{E1}$ and $o_{SK2} \rightarrow o_{E2}$.

Finally, by $dep(o_i)$ we denote the operations that depend directly or indirectly on the operation o_i , i.e.

$$dep(o_i) = \{o_j | o_i \rightarrow o_j\} \cup \{o_j | \exists_{j_1, j_2, \dots, j_m} o_i \rightarrow o_{j_1} \rightarrow o_{j_2} \rightarrow \dots \rightarrow o_{j_m} \rightarrow o_j\}.$$

Rule 1: A modification of a procedural operation o_i only can have an impact on its dependent operations $dep(o_i)$ and trivially on o_i itself.

Obviously, the simultaneous modification of the same procedural operation by two different users immediately leads to inconsistent models. To prevent this conflicting situation, as soon as one user starts a modification of an operation, the collaboration platform locks this operation, i.e. prohibits a concurrent modification by a different user. This approach is a first step, but does not prevent inconsistencies due to situations where two users modify two operations that depend on each other or modify two operations that have common dependent operations (Rule 1). To prevent this conflicting situation, we analyze the acyclic dependency graph explained above by not only locking the corresponding operation but also its dependent operations. Thus, a modification process comprises the following steps:

- As soon as a user starts a modification of an operation o_i , a lock request for this operation is sent to the collaboration server.
- On the server side, the set $dep(o_i)$ is determined by depth-first-search, i.e. recursively descending the acyclic dependency graph.
- The set $dep(o_i) \cup \{o_i\}$ is compared with the set $lock = \{o_j | o_j \text{ is locked}\}$ of already locked operations.

- If the intersection $(dep(o_i) \cup \{o_i\}) \cap lock$ is empty, then the set $dep(o_i) \cup \{o_i\}$ is added to the set $lock$ and the lock for operation o_i is granted to the user. If this intersection is not empty, the lock for the operation o_i will be denied.
- The user executes his modification step and sends the result to the server.
- The server incorporates this modification into the central shared model, forwards this modification to the other clients and forces them to incorporate it into their local copies. (In particular, this incorporation automatically forces a recalculation process of the dependent elements by the client CAD tools.)
- Finally, the set $dep(o_i) \cup \{o_i\}$ is deleted from the set $lock$ and the user who originated the modification step is informed about the success of the modification process.

This locking strategy ensures strict consistency (Hammond et al. 2004) of the shared data, i.e. in our context an update of an operation o_i is immediately visible to all participants. Figure 18 depicts the principle workflow of the explained modification process.

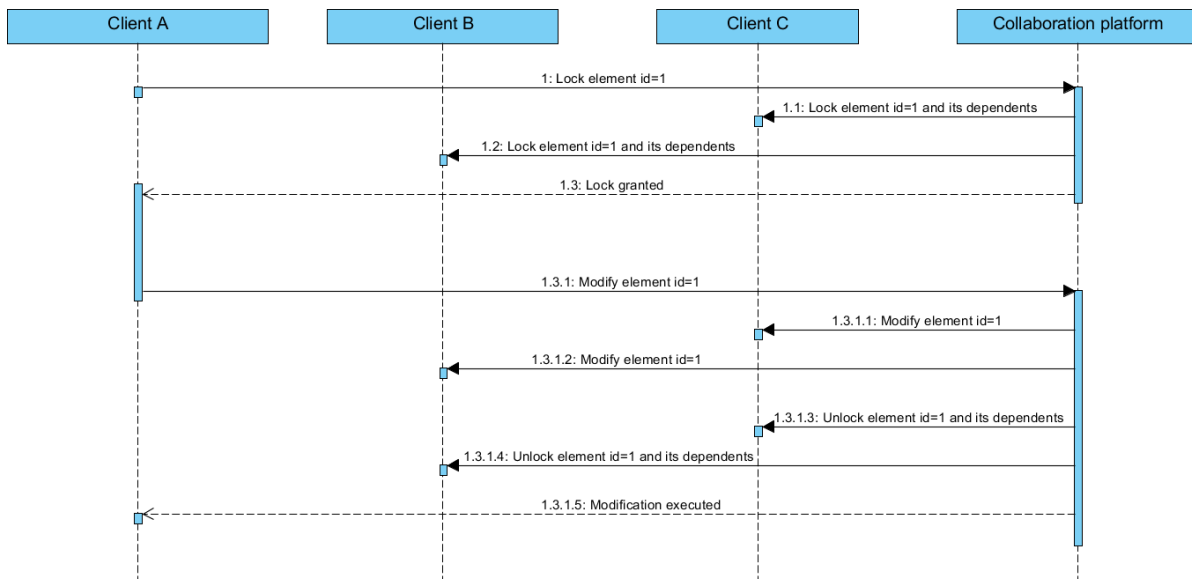


Figure 18: Locking and modifying an operation via the collaboration platform

4.4 Combining Track Alignment Model and Procedural Geometry Model

As described in Section 3.1, the Track Alignment Model comprises all track specific data imported from LandXML files while the Procedural Geometry Model consists of the construction steps and their dependencies that make up the geometric model. Since the alignment data is used to generate the curve defining the principle tunnel course, both models must be available at all times and, thus, are permanently connected within the collaboration platform as explained in the following section.

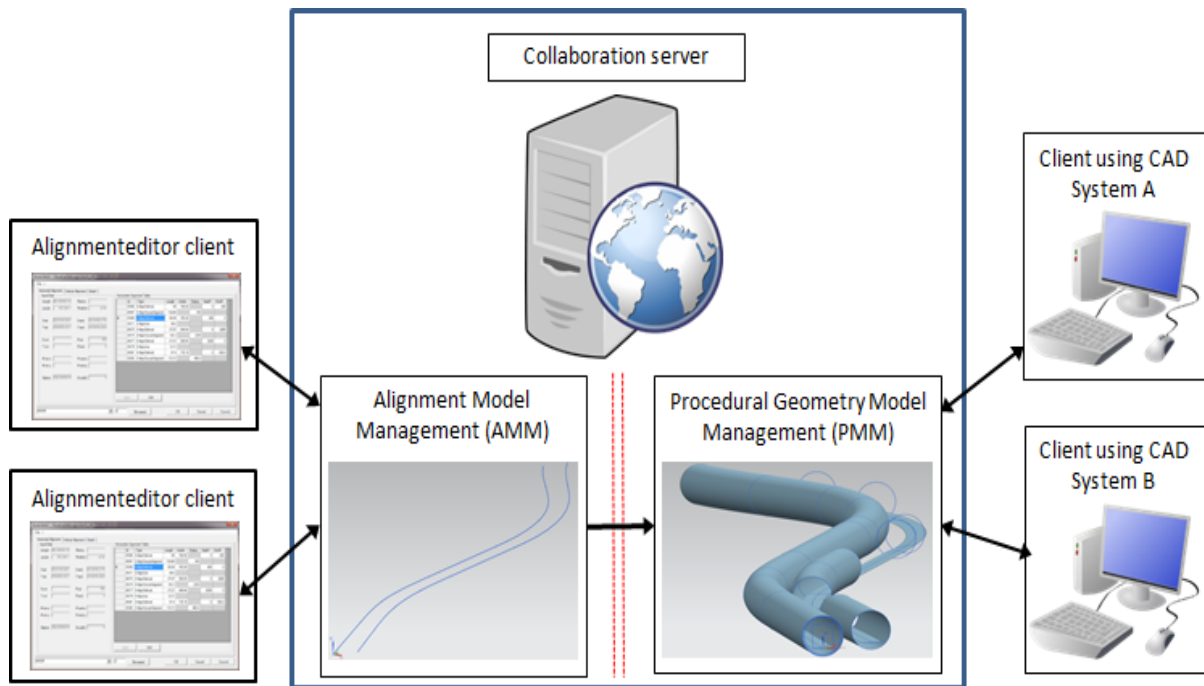


Figure 19: Interaction between Alignment Model Management and Procedural Model Management

The alignment object structure consists of two different 2D alignments – namely the horizontal and the vertical one. Both 2D alignments superposed describe a 3D curve – the railway track generated by the Alignment Model Management. This generation step is performed when alignment or track data is imported and every time when a track planner modifies the alignment(s) using the alignment editor client. The Alignment Model Management – which is directly connected to the Procedural Geometry Model by the collaboration server – then forwards any new curve data to the Procedural Model Management for making it accessible from the geometric model.

This makes it possible to synthesize different data where the Alignment Model Management is responsible for the logic behind the alignment data while the Procedural Model Management is in charge of creating geometric data from this logic. In this way, we can achieve a perfect encapsulation of data, as the latter must not be aware of the existence of the Alignment Model Management, which only uses the common public interface to add and modify geometry. The advantage of this concept is that it allows keeping the Model Management part very generic. The Alignment Model Management only needs to understand alignment specific data while the Procedural Model Management solely needs to know about procedural geometry data. Thus, the Procedural Model Management is applicable to any engineering context provided by separated modules – in our case tunnel specifics provided by the Alignment Model Management. The interaction between these two modules is depicted in Figure 19.

4.5 Different levels of detail in the collaborative modeling process

As explained in Sections 2 and 3, the procedural model supports the concept of different levels of detail (LoD). In this section, it will be shown how this concept can be used in the collaborative modeling process (described in section 4.2) in order to give the different experts participating the possibility to work on the level of abstraction their specialized task requires. Therefore, as an example, the alignment specialist works on the principle track course (LoD 1), while not being concerned with the precise structure of the tunnel front (LoD 3).

Subsequently, we denote the association of a certain PMO o_i to the LoD it is assigned to by $lod(o_i)$. For further considerations, we have to ensure that the design of the procedural model follows the following rule.

Rule 2: $o_i \rightarrow o_j$ implies $lod(o_i) \leq lod(o_j)$.

In order to work in a certain or up to a certain LoD n , the user specifies this LoD, when he joins a collaborative session. On server side, all operations $\{o_i | lod(o_i) \leq n\} \subseteq PM$ from the shared model are selected and sent to the client. Since operations only depend on operations in the same or lower LoDs (Rule 2), a valid (partial) model can be constructed out of this subset, while the client is completely unaware of all operations o_i with $lod(o_i) > n$.

According to the schema explained in section 4.3, as soon as a planner starts a modification of an operation o_i , a lock is requested from the collaboration server for this certain operation o_i . The set $dep(o_i)$ is determined on the server side since it might contain operations o_j with $lod(o_i) < lod(o_j)$, of which the user is completely unaware. Thus, the locking strategy in section 4.3 ensures consistency and holds for all steps described in the algorithm 4.3. In particular, after forwarding the modified operation o_i to other users, all dependent operations o_j – in particular operations o_j with $lod(o_i) < lod(o_j)$ – are recalculated automatically on the other participants' sides. Thus, even if a user is only aware of operations up to the LoD he is working on, the server guarantees the consistency of the entire model also comprising higher LoDs (Figure 20).

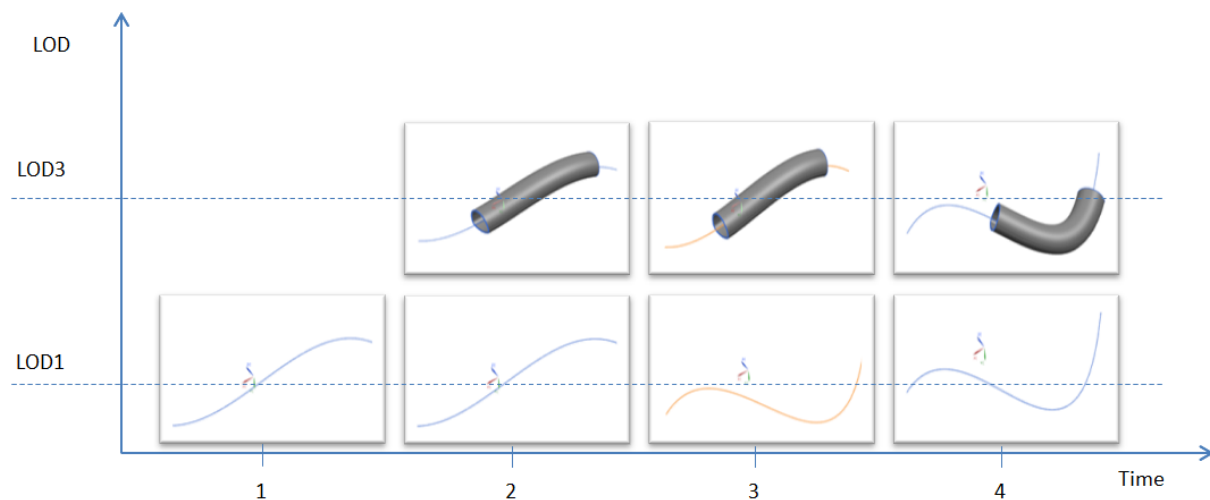


Figure 20: Working synchronously in two different levels of detail

Summarizing, in our approach presented in the section 4.2–4.5 we can see crucial differences to established collaboration concepts in the field of synchronous geometric modeling:

- Using the new multi-scale procedural model and its system independent format of storing the procedural operations, the collaboration platform facilitates a synchronous modeling process, where the different planners can use the CAD modeling tools they are accustomed to. This approach differs from most approaches where only one distinguished modeling tool is allowed.

- The presented locking mechanism allows several users to contribute to a collaborative session concurrently by locking only those sections of a model that are concerned by modifications. This concept is a major improvement to other approaches where only one user at a time is allowed to actively participate in a collaborative geometric modeling session.
- In the field of alignment based infrastructure projects, the presented concept of the integration of alignment data into the collaborative planning process allows the different experts, i.e. the alignment and the tunnel experts, to work concurrently on the same geometric model.
- The integration of the different levels of detail into the modeling process allows the different experts to work in the level of abstraction their specialized task is assigned to.

5 Case Study: Second main subway track in Munich, Germany

To prove the suitability of the developed approach we conducted a real-world case study. For the case study project, we chose the second main subway track in Munich which is currently in planning. Based on conventional 2D plans of the project, we developed a multi-scale model of the shield tunnels including the track model, the procedural geometry model and the semantic model, as discussed in Section 3. The model was made available on the collaboration platform to perform tests on synchronous design activities.

The construction of the Munich suburban city train (S-Bahn) started in the 1960s; the inner-city section, completed in 1972, was built completely below ground. Originally, the S-Bahn was constructed to transport 250,000 passengers a day. Due to the growth of the city population and its economic relevance, the use of the S-Bahn has increased dramatically and today some 800,000 people use the S-Bahn every day. As doubling the frequency of the trains was not able to solve this problem, the discussion arose to construct a second track about 20 years ago. The specific planning of this second track started 10 years ago and is now almost finished. The construction of the second track is scheduled to start in 2014 and should be finished in 2019. This second track is planned as a twin tunnel with a length of about seven kilometers, which connects the two most important inner-city train stations, Hauptbahnhof and Ostbahnhof. The already existing S-Bahn and 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.

The new S-Bahn is designed to be a fast connection between two ends of the city center and reduce the travelling time between the city and the airport significantly. This fast connection is achieved by inserting only one additional station between the two terminal points, a fact that extends the usual distance between stations. In order to reduce the distance that passengers have to walk in case of emergency, escape shafts have been planned each kilometer along the track. In Figure 21 the tunnel model is shown, which visualizes the connection between the station Hauptbahnhof and the brand new station Marienhof. Due to its length of about two kilometers, an escape shaft will be placed in its middle.



Figure 21: A model of an escape shaft between the two tunnel tubes

The project was originally planned using a conventional 2D drawing-based approach. These 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 3. 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 at different levels of abstraction
- flexibility: modifications made at coarser levels are directly propagated to all finer levels



Figure 22: Integration of the multi-scale shield tunnel model with the 3D city model of Munich

Figure 22 shows the integration of the tunnel model with the station Hauptbahnhof and the city center of Munich. To display 3D city models as planning context we integrated

functionality for accessing CityGML models using the collaborative planning platform (Steuer et al. 2013).

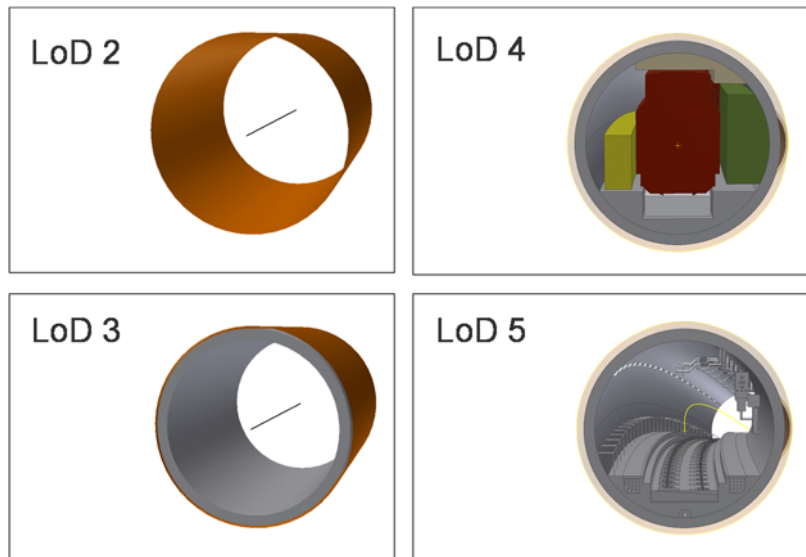


Figure 23: LoD methodology applied to the shield tunnel model of the case study

The LoD approach allows engineers to adapt the track of the tunnel and the escape shaft in a very flexible and dynamic manner. As explained in Section 3.3, the speed of the train 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 determining the permitted train maximum velocity, the model automatically adapts to this changed parameter. Additionally, this provides a clash detection mechanism, since a too high value for velocity results in a visible intersection of the loading railway gauge and the inner tunnel hull. The automatic adaption of the super-elevation stripe according to the train speed is shown in Figure 24.

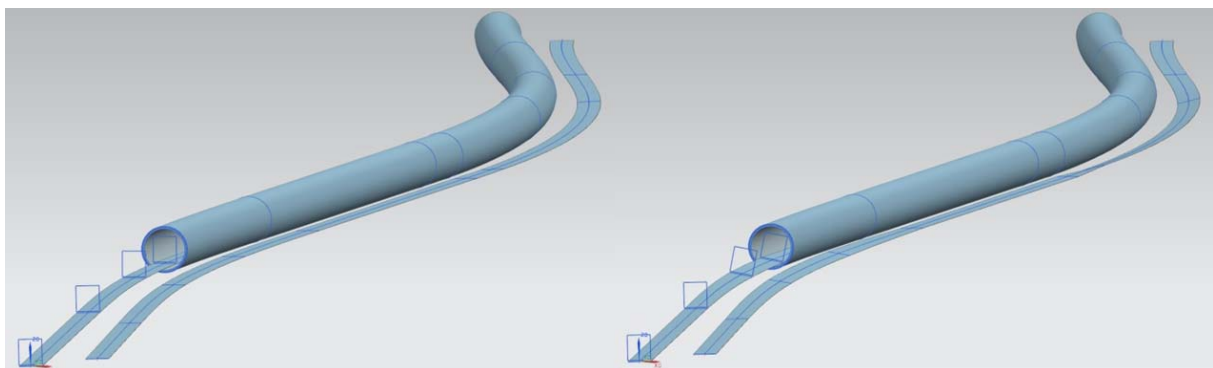


Figure 24: The super-elevation stripe and the loading railway gauge are automatically recalculated when the train speed is modified.

The collaboration platform we have developed allows different experts to work synchronously in this planning process. The different modeling specialists are able to use the modeling tools they are accustomed to. An example where two clients work synchronously using two different CAD systems – Autodesk Inventor and Siemens NX – is shown in Figure 25.

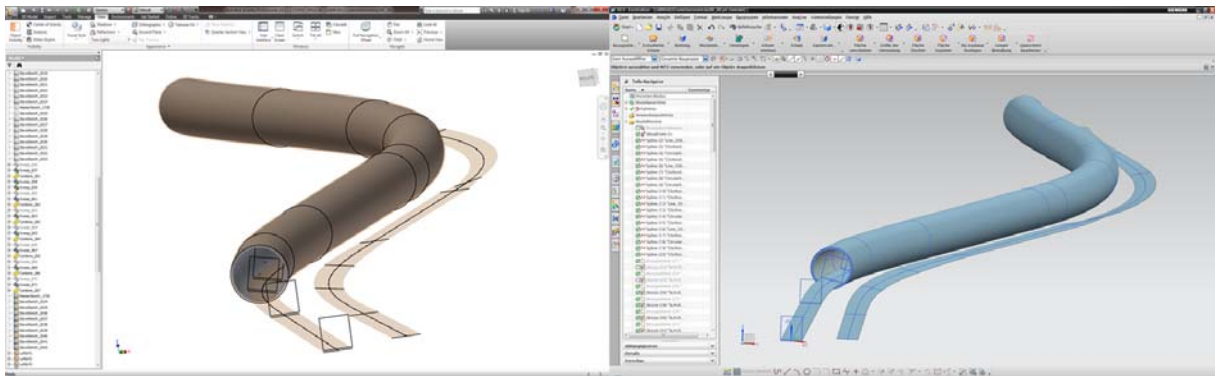


Figure 25: A Siemens NX and an Autodesk Inventor client used for synchronously modifying a tunnel model

6 Conclusions and future work

In this paper, a new methodology was introduced that makes it possible to create inherently consistent multi-scale models and to use them for synchronous engineering collaboration. The core concept is the definition of dependencies between geometry objects on different LoDs by using procedural geometry representations. The implementation of the concept is based on the application of parametric modeling techniques. The methodology is general and applicable to a wide range of infrastructure project types.

In this study, we have focused on applying the multi-scale methodology for modeling shield tunnels. We presented how procedural geometry can be used to create an inherently consistent multi-scale tunnel model and discussed how it can be integrated with a corresponding semantic data model.

We proposed a collaboration platform, which enables different planners to participate concurrently in the planning process. The procedural models' neutral data format facilitates the usage of different proprietary modeling tools that the involved planners are accustomed to, while the concept of different levels of detail enables the planners to undertake specialized modeling tasks at their own specific level of abstraction. To enable different planners to work simultaneously on the shared multi-scale model we proposed a concurrency approach based on a locking strategy using the model inherent acyclic dependency graph.

We applied the methodology in a real-world case study – the second main subway track in Munich, Germany. The case study conducted proves the general feasibility of the approach.

An open question that has yet been not tackled is the possibility to propagate modifications made at finer LoDs to coarser ones. This requires the bi-directional modeling of dependencies and will be the subject of future research.

Acknowledgements

We gratefully acknowledge the support of the German Research Foundation (DFG) for funding the project under grant FOR 1546. We are also very thankful for the support of the engineering consultancy company Obermeyer Planen und Beraten for providing valuable

insight in current planning practices and data for the presented real-world case study. We also thank Deutsche Bahn and Landeshauptstadt München for their support.

References

- [1] R. Johansen, *Groupware: Computer Support for Business Teams*, The Free Press New York, USA, 1988.
- [2] J.J. Shah, M. Mäntylä, *Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications*, Wiley Press Inc., New York, 1995.
- [3] T.H. Kolbe, *Representing and Exchanging 3D City Models with CityGML*, in: *Proc. of the 3rd International Workshop on 3D Geo-Information*, Seoul, Korea, 2008.
- [4] L. Meng, A. Forberg, *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, 2007.
- [5] A. Forberg, *Generalization of 3D building data based on a scale-space approach*, *ISPRS Journal of Photogrammetry & Remote Sensing* (2007) 104–111.
- [6] AIA, *AIA G202-2013 Building Information Modelling Protocol Form*, American Institute of Architects (2013).
- [7] J. Bedrick, *A Level of Development Specification for BIM Processes*, available at http://www.aecbytes.com/viewpoint/2013/issue_68.html (accessed on May 16, 2013).
- [8] D. Roller, *An approach to computer-aided parametric design*, *Computer Aided Design* (1991) 385–391.
- [9] M.J. Pratt, B.D. Anderson, T. Ranger, *Towards the standardized exchange of parameterized feature-based CAD models*, *Computer-Aided Design* 37 (2005) 1251–1265.
- [10] C. Koch, B. Firmenich, *An approach to distributed building modeling on the basis of versions and changes*, *Advanced Engineering Informatics* (2011) 297–310.
- [11] D. Mun, S. Han, J. Kim, Y. Oh, *A set of standard modeling commands for the history-based parametric approach*, *Computer-Aided Design* 35 (2003) 1171–1179.
- [12] M. Stiteler, *Construction History and Parametrics: Improving affordability through intelligent CAD data exchange*. CHAPS Program Final Report, Advanced Technology Institute, North Charleston, USA, 2004.
- [13] M. Obergrießer, T. Euringer, T. Horenburg, W. Günthner, *CAD-Modellierung im Bauwesen - Integrierte 3D-Planung von Brückenbauwerken*, Hochschule Regensburg, Germany, 2011.
- [14] Y. Ji, A. Borrmann, J. Beetz, M. Obergrießer, *Exchange of Parametric bridge Models using a Neutral Data Format*, *Journal of Computing in Civil Engineering ASCE* (2013) 593–606.
- [15] Y. Ji, J. Beetz, P. Bonsma, N. Nisbet, C. Katz, A. Borrmann, *Integration of Parametric Geometry into IFC-Bridge*, in: *23rd Forum Bauinformatik*, Cork, Ireland, 2011.
- [16] Y. Ji, A. Borrmann, M. Obergrießer, *Towards the Exchange of Parametric 3D Bridge Models Using a Neutral Data Format*, in: *Proc. of the ASCE International Workshop on Computing in Civil Engineering*, Miami, USA, 2011.
- [17] M. Obergrießer, T. Euringer, A. Borrmann, E. Rank, *Integration of geotechnical design and analysis processes using a parametric and 3D-model based approach*, in: *Proc. of the ASCE International Workshop on Computing in Civil Engineering*, Miami, Florida, USA, 2011.
- [18] G. Lee, R. Sacks, C.M. Eastman, *Specifying parametric building object behavior (BOB) for a building information modeling system*, *Automation in Construction* 15 (2006) 758–776.
- [19] B. Bittig, J. Shah, *Derivation of a standard set of geometric constraints for parametric modeling and data exchange*, *Computer-Aided Design* (2001) 17–33.
- [20] R. Anderl, R. Mendgen, *Modeling with constraints: theoretical foundation and application*, *Computer-Aided Design*, vol. 28(3), (1996) 155–168.
- [21] C.M. Eastman, P. Teicholz, R. Sacks, K. Liston, *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*, 2nd ed., Wiley, Hoboken, 2011.
- [22] K. Lee, S. Chin, J. Kim, *A core system for design information management using Industry Foundation Classes*, *Computer-Aided Civil and Infrastructure Engineering* (2003) 286–298.
- [23] C. Eastman, *Building Product Models: Computer Environments Supporting Design and Construction*, CRC Press, 1999.
- [24] J.R. Jubierre, A. Borrmann, *Cross-submodel consistency preservation in multi-scale engineering models*, in: *Proc. of the 14th International Conference on Civil, Structural and Environmental Engineering Computing*, Sardinia, Italy, 2013.

- [25] A. Borrmann, Y. Ji, J.R. Jubierre, Multi-scale Geometry in Civil Engineering Models: Consistency Preservation through Procedural Representations, in: Proc. of the International Conference on Computing in Civil and Building Engineering, Moscow, Russia, 2012
- [26] Lichtberger B., Track Compendium: Substructure; Track System; Maintenance; Economics, 2nd ed., Eurailpress in DVV Media Group, 2011.
- [27] Illinois Department of Transportation, Horizontal Alignment, 2010.
- [28] Regelwerk Technik Eisenbahn, Fahrbahnpraxis Meter- und Spezialspur: D RTE 22540, 2007.
- [29] Ministerio de Fomento Espanol, Norma 3.1-IC Trazado – Instrucción de Carreteras, Spain, 1999.
- [30] LandXML, LandXML Data Specifications: LandXML 1.2, available at <http://www.landxml.org/spec.htm>.
- [31] K. Kim, M.L. Pratt, R.G. Lyer, R.D. Sriram, Standardized data exchange of CAD models with design intent, Computer-Aided Design (2008) 760–777.
- [32] M.L. Pratt, Exchanging history-based parametric CAD models using ISO 10303, International Journal of Product Lifecycle Management, vol. 4 (2010) 321-316.
- [33] ISO - International Organization for Standardization, ISO 10303 - 55.
- [34] G.-H. Choi, D. Mun, S. Han, Exchange of CAD part models based on the macro-parametric approach, International Journal of CAD/CAM, (2002) 13-21.
- [35] ISO - International Organization for Standardization, ISO 16739:2013, Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries.
- [36] R.R. Lipman, Mapping Between the CIMsteel Integration Standards and Industry Foundation Classes Product Models for Structural Steel, in: Proc. of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montréal, Canada, 2006.
- [37] N. Yabuki, T. Aruga, H. Furuya, Development and application of a product model for shield tunnels, in: Proc. of the 30th International Symposium on Automation and Robotics in Construction, Montréal, Canada, 2013.
- [38] N. Yabuki, Fundamental Study on Development of a Shield Tunnel Product Model. Journal of Applied Computing in Civil Engineering, (2007) 261-268.
- [39] A. Sandler, T.H. Kolbe, Spatio-semantic coherence in the integration of 3D city models, in: Proc. of the 5th International Symposium on Spatial Data Quality, Enschede, Netherlands, 2007.
- [40] E. Clementini, Ontological impedance in 3D semantic data modeling, in: Proc. of the 5th International 3D GeoInfo Conference, Berlin Germany, 2010.
- [41] J. Stascheit, C. Koch, F. Hegemann, M. König, G. Meschke, Process-oriented numerical simulation of mechanized tunneling using an IFC-based tunnel product model, in: Proc. of the 13th International Conference on Construction Applications of Virtual Reality, London, UK, 2013.
- [42] J. Vogel, W. Geyer, L.T. Cheng, M. Muller, Consistency Control for Synchronous and Asynchronous Collaboration Based on Shared Objects and Activities. In: Computer Supported Cooperative Work (2005) 573-602.
- [43] J. Munson, D. Dewan, A concurrency control framework for collaborative systems. In: Proc. of the ACM conference on Computer supported cooperative work, (1996) 278-287.
- [44] C.A. Ellis, S.J. Gibbs, Concurrency control in groupware systems, in: Proc. of the ACM SIGMOD International Conference on Management of Data, (1989) 399–407.
- [45] D.A. Menasce, N. Nakanishi, Optimistic versus pessimistic concurrency control mechanisms in data base management systems, Information Systems vol. 7 (1982) 13-27.
- [46] I. Foster, Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering, Addison-Wesley, 1995.
- [47] M. Schorr, A. Borrmann, M. Obergrießer, Y. Ji, W. Günthner, T. Euringer, E. Rank, Using Data Management Systems for Civil Engineering Projects - Potentials and Obstacles, Journal of Computing in Civil Engineering (2011) 430–441.
- [48] T. Haerder, A. Reuter, Principles of Transaction-Oriented Database Recovery, ACM Computing Surveys (1983) 287–317.
- [49] A.R. Kutay, C. Eastman (Eds.), Transaction Management in Engineering Databases: Proceedings of Annual Meeting of Database Week; Engineering Design Applications, IEEE Computer Society Press, 1983, pp. 73-80.
- [50] N.S. Barghouti, G.S. Kaiser (Eds.), Concurrency Control in Advanced Database Applications, ACM Computer Surveys, 1991, pp. 269–317.
- [51] C. Eastman, Managing Integrity in Design Information Flows. Computer Aided Design, (May 1996) 551-561.
- [52] A. Sheth, M. Rusinkiewicz, On Transaction Workflows, Data Engineering Bulletin, (1993).
- [53] M. Rusinkiewicz, A. Sheth, Specification and Execution of Transactional Workflows, Addison-Wesley, 1994.

- [54] J. Xu, J. Zhang, T. Harvey, J. Young, A Survey of Asynchronous Collaboration Tools, *Information Technology Journal* (2008) 1182–1187.
- [55] J. Xu, J. Zhang, A Survey of Synchronous Collaboration Tool, *Information Technology Journal* (2008) 1049–1054.
- [56] C. Koch, B. Firmenich, Experiences from the Application of Processing-oriented Building Information in a CAD-based Environment, in: *Proc. of the International Conference on Computing in Civil and Building Engineering*, Nottingham, UK, 2010.
- [57] B. Firmenich, *CAD im Bauplanungsprozess: Verteilte Bearbeitung einer strukturierten Menge von Objektversionen*, Shaker, 2002.
- [58] C. Sun, Y. Yang, Y. Zhang, C. Chen, A Consistency Model and Supporting Schemes for Real-time Cooperative Editing Systems, in: *Proc. of the 19th Australian Computer Science Conference*, Melbourne, Australia, 1996.
- [59] R. Bidarra, E. Van de Berg, W.F. Bronsvoort, A Collaborative Feature Modeling System, In: *Journal of Computing and Information in Engineering*, vol. 2(3), (2002) 192–198.
- [60] M. Tang, S.-C. Chou, J.-X. Dong, Conflicts classification and solving for collaborative feature modeling, *Advanced Engineering Informatics* 21 (2007) 211–219.
- [61] A. Borrmann, P. Wenisch, Ch. van Treeck, E. Rank, Collaborative computational steering: Principles and application in HVAC layout, *Integrated Computer-Aided Engineering* 13 (2006) 361-376.
- [62] M. Li, S. Gao, C. Wang, Real-Time Collaborative Design With Heterogeneous CAD Systems Based on Neutral Modeling Commands, *Journal of Computing and Information Science in Engineering* (2007) 113–125.
- [63] X. Cai, Li, X., He, F., S. Han, X. Chen, Flexible Concurrency Control For Legacy CAD to Construct Collaborative CAD Environment, *Journal of Advanced Mechanical Design, Systems, and Manufacturing* (2012) 324–339.
- [64] H. Fan, C. Sun, Dependency-Based Automatic Locking for Semantic Conflict Prevention in Real-Time Collaborative Programming, in: *SAC, 2012*, Riva del Garda, Italy, 2012.
- [65] M. Flurl, R.-P. Mundani, A. Borrmann, E. Rank, A collaborative multi-scale planning platform: Concept and implementation approach, in: *Proc. of the International Conference on Computing in Civil and Building Engineering*, Moscow, Russia, 2012.
- [66] H. Lui, X. Li, F. He, Z. Huang, A Less Constraint Concurrency Control and Consistency Maintenance in Collaborative CAD System, in: *Proc. of the 14th International Conference on Computer Supported Cooperative Work in Design*, Shanghai, China, 2010.
- [67] C. Sun, Optional and responsive fine-grain locking in internet-based collaborative systems, in: *IEEE Transactions on Parallel and Distributed Systems*, vol. 13(9), (2002), 994-1008.
- [68] T. Berlage, A. Genau, A Framework for Shared Applications with a Replicated Architecture, in: *Proc. of the 6th annual ACM symposium on user interface software and technology*, (1993) 249–257.
- [69] M. Flurl, R.-P. Mundani, E. Rank, Graph-Based Concurrency Control for Multi-Scale Procedural Models, in: *Proc. of the European Conference on Product and Process Modelling*, Wien, Austria, 2014.