Cross-submodel consistency preservation in multi-scale engineering models

J.R. Jubierre, A. Borrmann
Chair of Computational Modeling and Simulation,
Technische Universität München, Germany

Abstract

In recent years a tendency toward the adoption of parametric modeling capabilities has been observed in the AEC industry. These capabilities facilitate the design and enable engineers to capture design intent and reduce errors that would otherwise occur when manual changes are performed. In the modeling of large infrastructures engineers usually divide their models / projects into a set of submodels to improve access and performance. However, maintaining the consistency across these submodels is a major issue. Even though the actual parametric CAD systems are able to establish implicit dependencies between the operations produced in the modeling process, these dependencies are not enough when the model / project is split into several submodels. This paper introduces two novel methods, which enable automated consistency preservation across the split submodels. The first method links scalar values, used in orientation vectors or positioning coordinates, by means of global parameters. The second method links topological information a using a master/slave approach. To assess and compare the suitability of the methods they are applied in a real-world case study pertaining to a subway tunnel project in Munich, Germany. We discuss the drawbacks and benefits of both methods.

Keywords: Infrastructure modeling, multi-scale modeling, model consistency.

1 Introduction

In recent years, considerable research efforts have been focusing on moving the AEC industry from 2D drawing and planning into the 3D modeling world. While major progress has been achieved in the field of building construction – particularly through the widespread adoption of the building information modeling (BIM) technology – in infrastructure design and engineering, conventional planning techniques based on 2D drawings are still dominant. However, if applied effectively, 3D modeling approaches can significantly speed-up infrastructure planning projects and lead to fewer errors and increased efficiency (Borrmann et al. [1], Obergriesser et al. [6], Ji et al. [5]).

One of the specific challenges in applying this technology lies in the different scales in which civil engineers have to work - ranging from the kilometre scale for the general routing down to the centimetre scale for the detailed planning of individual nodes. Despite the multi-scale characteristics inherent in the planning of infrastructures, today's planning software does not support multi-scale modeling. The research unit "3DTracks" funded by the German Research Foundation is tackling this issue by developing a methodological basis for introducing multi-scale modeling into civil engineering projects.

In order to enable engineers to work flexibly on different scales, it is absolutely imperative that CAD software preserves consistency between the scales. A suitable approach to solving this problem relies on the use of parametric CAD systems for explicitly defining dependencies between the representations on the different scales (Borrmann et al. [2]). This is realized by means of a procedural geometry description and enables the consistency across multiple levels of abstraction to be maintained. More specifically, procedural models allow engineers to demonstrate the logic applied while generating the model and thus facilitate consistent multi-scale modeling.

Large-scale infrastructure projects frequently have to be split into different submodels, to allow different engineers to work on different parts of the model at the same time. Unfortunately, a procedural model only guarantees consistency inside the submodel and cannot define automatic dependencies between the different submodels of the complete project. Moreover, in a conventional modeling process it is not possible to establish these dependencies between two elements that share a particular geometry. To fill this technological gap, this paper presents a novel concept for linking the different sections of the model based on advanced parametric and geometric relationships.

2 Motivation

The application of procedural modeling aims to create inherently consistent multi-scale models based on the use of parametric modeling technologies (Shah & Mäntyla [7]). The concepts underlying parametric modeling were developed in the 1990's and subsequently implemented in mature commercial CAD systems. The majority of the parametric CAD systems available are based on so-called features. Features create 3D volumes through the sequential use of geometric operations such as extrusion, rotation or Boolean operations (Bettig & Shah [3]). Features and parametric sketches can be subsumed under the term "procedural operations".

In this way, the model remains a single entity while simultaneously providing the desired consistency. Despite that fact, a set of initial features in large infrastructure projects can be restricted to different parts of the project, thus creating different submodels within the project. This modeling strategy may therefore result in destroying the model's consistency, since the modifications will remain local and the designing engineer must be aware that they have to update the dependent submodels manually, which is time-consuming and error-prone.

Figure 1: Tunnel model split into two submodels.

As the consistency of the model is one key point in the modeling of large infrastructure projects, we cannot rely on manual consistency preservation and need to provide an approach which preserves the integrity of the model in an automated fashion. This approach has to link the submodels in an explicit way. These links can subsequently be used to automatically update all submodels that share information with the one being modified. This functionality makes it possible to achieve cross-submodel consistency preservation.

3 Parametric Modeling

In recent years a tendency toward the adoption of parametric capabilities has been observed in the AEC industry. These capabilities facilitate the design and enable engineers to capture design intent and reduce errors that would otherwise occur when manual changes are performed.

The majority of parametric CAD systems available 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 (Bettig & Shah [3]).

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 or the dimensions in a sketch, are explicitly available parameters. 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.

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 2 depicts some of the geometric constraints available in major parametric CAD systems. In addition, 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.

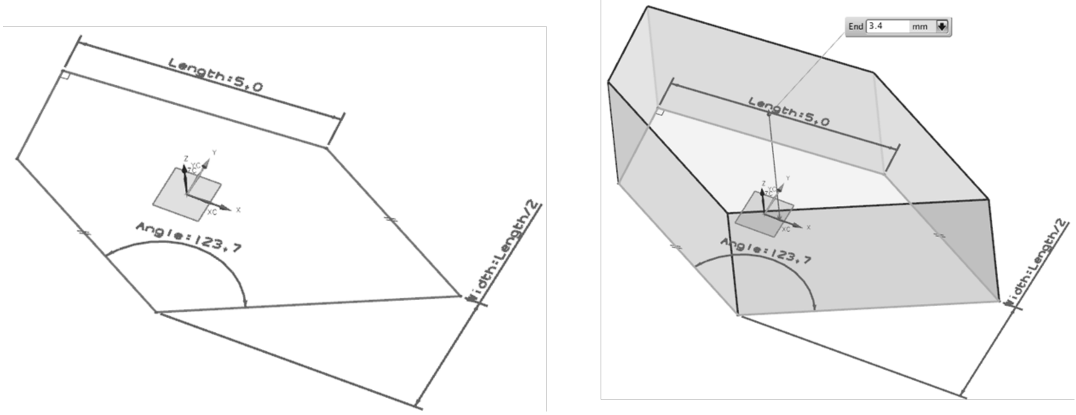

Figure 2: A sketch definition and the subsequent application of an extrusion operation.

To assess the increased productivity made available by parametric CAD systems, Cambashi Limited (Cambasi [4]) conducted extensive modeling tests using the CAD system AutoCAD 2007 and AutoCAD 2010. While the former is a standard non-parametric CAD system, a parametric module was added in the latter version. As the study reports, the time invested to create the same model was decreased by 35 per cent using the system with parametric capabilities.

3.1 Implicit dependencies in procedural models

In our approach, every procedural operation is modelled as a separate unit that contains the information needed to modify the existing solid. As depicted in Figure3, these units contain three types of information. The dependencies describe the relations between procedural operations. To illustrate this point using an example, it would not be possible to accomplish an extrusion without the definition of a sketch, so the extrusion container will automatically create a link between the extrusion and the sketch.

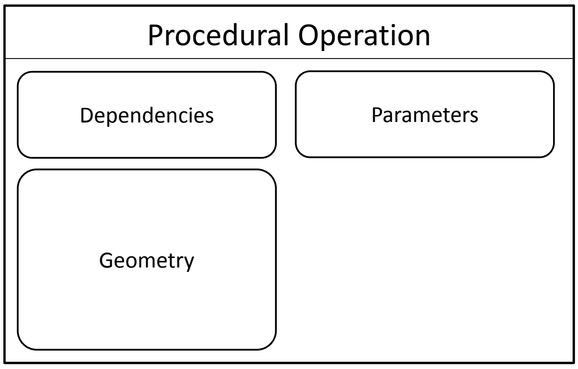

Figure 3: Container for a procedural operation.

The second type of information concerns the parameters. Parameters are scalar values which can be expressed as a number or as an equation connecting their own parameters or parameters employed in foreign operations. The last type of information refers to the geometry. This section contains all the geometric information including dimensions and constraints in the sketch or the height dimension in the extrusion.

Within the modeling process and the creation of the construction tree, dependencies are automatically generated between operations. These relations are known as implicit dependencies and allow the parametric CAD system to properly update the related operations when a modification is performed. We call these dependencies *implicit* as the CAD system generates them in a transparent manner to the end user.

Nevertheless, in the modeling process of complex infrastructures (see Figure 4), even if the operations are inserted into the construction tree consecutively, no relations are established between submodels. This main drawback is responsible for the generation of discontinuities in the model when local changes are performed.

1. Create Sketch3D
2. Create Workplane
3. Create Sketch2D
4. Create Sweep
5. Create Sketch3D
6. Create Workplane
7. Create Sketch2D
8. Create Sweep

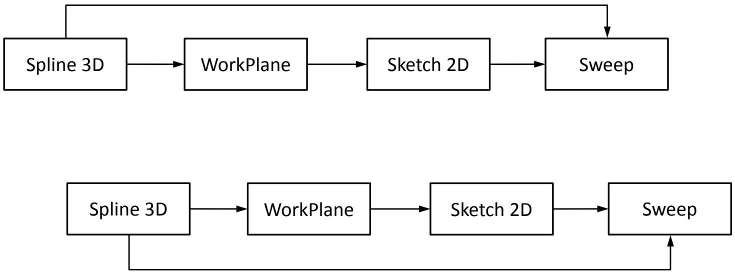

Figure 4: Construction tree and implicit dependencies of a tunnel split into two submodels.

4 Explicit dependencies in procedural models

In order to detect the needs of explicit dependencies in infrastructure models, we began by looking for sources of inconsistency in the infrastructure models. The example we chose was a subway tunnel designed to link two underground stations. Here we found two types of predefined models, namely one static model, depicting the two train stations, and one dynamic model representing the tunnel connecting those stations which was the subject of our study.

If we take a closer look at the connection between the underground stations and the subway tunnel, we may realize that both models must be connected by both the tunnel axis and the geometry of the tunnel leading to the station. If the axis of the tunnel goes deeper in the earth or the cross section of the tunnel changes, both models have to be updated accordingly. Another example is to be found in long tunnels where the tunnel is split to fit in the exit through a rescue shaft.

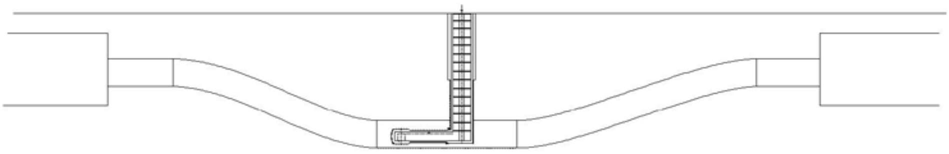

Figure 5: Different submodels of a tunnel which connects two train stations.

After a close study of this example, we discovered three kinds of inconsistencies. The first inconsistency is produced by the positioning of the axis, which causes discontinuities in the track of the infrastructure. The second inconsistency is generated by the orientation of the cross-section, which produces empty spaces / gaps between the models. Finally, there is an inconsistency in the accuracy of the cross-section between models, which creates fractional models over the axis or track of the infrastructure.

As the implicit dependencies cannot create automatic links between the models, it is necessary to define a new set of explicit dependencies. This new type of relationships has to use elements that already exist in the definition of parametric models and are employed at the time the model is created. As result of our research, we created two novel methods to link the submodels by means of global parameters and reference geometry. Their definition and usage is defined in the following subsections.

4.1 Global Parameters

Positioning and orientation in 3D space is defined as one of the first needs when connecting different submodels. Such connections are defined by scalar values and can therefore be described by parameters. The positioning of one submodel is defined by its coordinates and the orientation of its normal vector, both of which can be decomposed in a set of three parameters.

As shown in Figure 3, every procedural operation contains the definition of its local parameters where this information can be recorded. Nevertheless, storing shared parameters in a local operation may raise inconsistencies when the operation is deleted. So it is necessary to create a central space for recording the parameters where they can be called up regardless of the modeling process.

To overcome this problem, we extended the procedural model by inserting a *GlobalParameters* section before the list of procedural operations. In this way, we allow local parameters to be linked with global parameters while the scalar information remains independent of the workflow during the different procedural operations. Moreover, every time one of the global parameters is updated, all the associated submodels will be automatically updated, too.

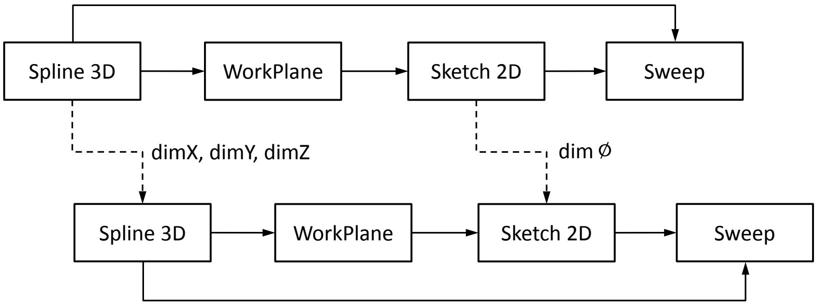

Figure 6: Global parameters added to the operation dependency diagram.

4.2 Reference Geometry

As the complexity of the model increases, the need for topological relations becomes a must. When two submodels share the same geometry, the scalar information stored in the global parameters proves to be insufficient. Therefore a central place to store atomic geometric elements, following the problematic dealt with the scalar values, shines as a proper solution. However, geometric elements are not just defined by its intrinsic geometric information, but also by its relations with other elements or the position and orientation in space.

In order to allow geometry to be shared while keeping the set of submodels consistent, we decided to introduce the concept of Master/Slave in the definition of geometric elements when sketching the procedural operations. Using this methodology, the first procedural operation that defines a shared component, defines its geometry as the Master. All the subsequent operations linked to this geometry, will copy the geometric content, applying the definition Slave, and every time the master geometry is updated, all the relevant copies are automatically updated, as well.

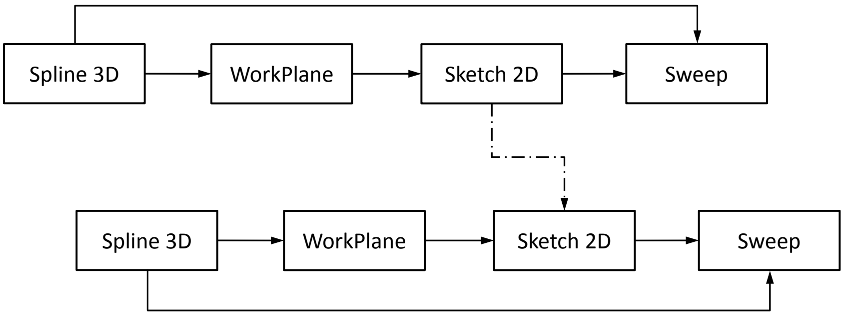

Figure 7: Reference geometry dependencies added to the operation dependency diagram.

At the current stage of our research we do not allow master and slave geometry definitions to be mixed during the same procedural operation. Nor did we find any example where a mixture of both definitions might be useful, during the first phase of our research. We are nevertheless still working on this field of research and expect to extend this methodology to other case studies.

5 Case Study

Our techniques for cross-submodel consistency preservation have been validated in a real-world case study, namely the second city tunnel in Munich (2. S-Bahn-Stammstrecke München)[1]. The infrastructure of this second subway tunnel is designed to balance the traffic through the city and consists of three new stations and two completely new double tunnels. For our study, we selected the first tunnel which is about two kilometres long and connects the new module at Munich central station with the new –Marienhof underground station. Due to the length of the tunnel, a rescue shaft has been planned in the middle of the tunnel, which remains independent of the tunnel design and allows us to test our new methodology.

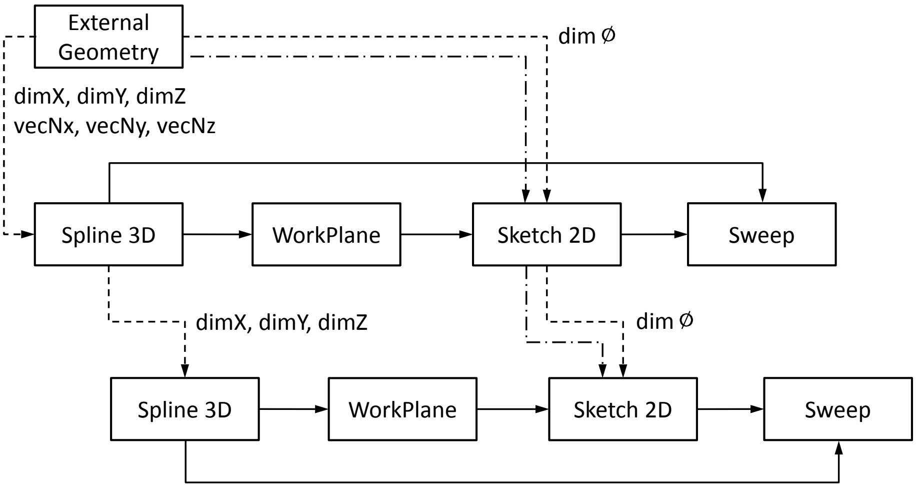

Figure 8: Procedural operation dependency diagram for an inner-city tunnel and a train station.

The modeling of the tunnel allows us to test the methodology of dependencies based on global parameters. As can be seen in Figure 5, the tunnel was split into five submodels; two submodels represent the tunnel's entrance and exit, one the connection with the rescue shaft and the last two, just the tunnels connecting them. To preserve the cross-submodel consistency between these elements, we used global parameters at the axis of the line of the tunnel to obtain the right spatial position. In this case, the normal axis was not needed as the cross-section is defined as being perpendicular to the tangent of the curve at this point. We did require both the location and the orientation of the connection between the submodel of the tunnel and the station, however.

---

[1] http://www.2.stammstrecke-muenchen.de/

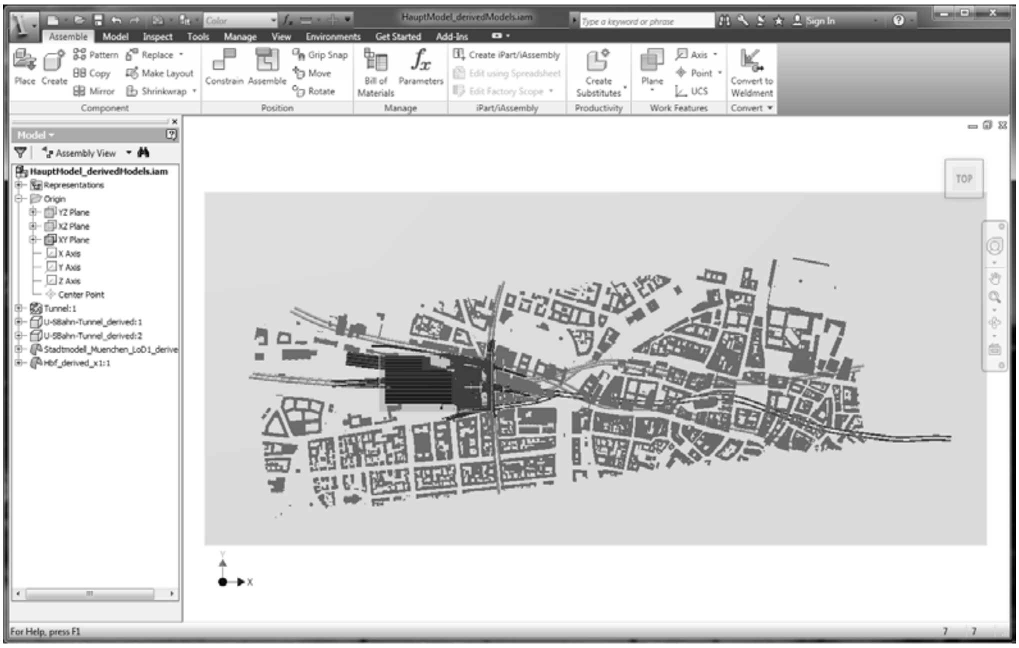

Figure 9: Model showing the course of the new underground tunnel in Munich.

In order to test the reference geometry method, we defined the cross-section of the tunnel in the first submodel, the one in which the train will enter, as the *Master* sketch. The other submodels establish the sketch defining the cross-section as a *Slave* of the first one, so it was not possible to make any modifications to them.

6 Conclusions

This paper presents some of the results of our ongoing research project 3DTracks, which aims to introduce consistency-preserving multi-scale modeling of infrastructure facilities on the basis of parametric CAD systems and procedural models. Large infrastructure models are usually divided into a number of submodels to allow a team of engineers to work simultaneously on the model. Another reason for working with submodels is the need to integrate models which originate from different CAD systems.

Even though the actual parametric CAD systems are able to establish implicit dependencies between the operations produced in the modeling process, these dependencies are not enough when the model / project is split into several submodels. For this reason, it is necessary to automate the definition of explicit dependencies in order to maintain consistency across the submodels.

In this paper we presented two new methods to establish explicit dependencies between submodels which preserve consistency in multi-scale engineering models. These two methods are designed for different purposes, depending on the modeling information to be shared. –The first one, the global parameter method, shows a very sound method of sharing scalar information, like the position or spatial orientation. The second type of explicit dependency that we introduced is the reference geometry method, which allows the user to connect submodels by means of geometry. The disadvantage of the second method is that the complete operation is defined as a master or slave and no mixed definitions are permitted in the geometry.

A first case study, conducted for the modeling of a tunneled subway track, proves the general feasibility of our approach. An open question not tackled so far is the possibility of isolating the referenced geometry defined as a master in a separate section within the procedural geometry, as we did for the global parameters, or how to reclassify geometry defined as a copy into the master, when the master element will be deleted by the user.

Acknowledgments

The research presented in this paper has been sponsored by the German Research Foundation (DFG) within the framework of the 3DTracks research unit. This support is gratefully acknowledged.

References

[1] A. Borrmann, Y. Ji, I-C. Wu, M. Obergrießer, E. Rank, C. Klaubert, W. Günthner: "ForBAU - The virtual construction site project", In: Proc. of the 24th CIB-W78 Conference on Managing IT in Construction. Istanbul, Turkey, 2009.

[2] A. Borrmann, Y. Ji, J.R. Jubierre: "Multi-scale geometry in civil engineering models: Consistency preservation through procedural representations", In: Proc. of the 14th Int. Conf. on Computing in Civil and Building Engineering. Moscow, Russia, 2012.

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

[4] Cambashi Limited: "AutoCAD 2010 Productivity Study", 2009 http://www.cambashi.com/white-papers, accessed March 2013.

[5] Y. Ji, A. Borrmann, J. Beetz, M. Obergrießer: "Exchange of Parametric Bridge Models using a Neutral Data Format", Journal of Computing in Civil Engineering (in press), DOI: 10.1061/(ASCE)CP.1943-5487.0000286, 2013

[6] 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.

[7] J.J. Shah, M. Mäntylä: "Parametric and Feature-based CAD/CAM - Concepts, Techniques, Applications", Wiley Press Inc., 1995.